

UNIVERSITÄT DER BUNDESWEHR MÜNCHEN  
Fakultät für Elektrotechnik  
(Energie- und Informationstechnik)

# Eine asynchrone Sender- und Empfängerstruktur zur Übertragung von seriellen Datenströmen höchster Datenraten

Philipp Börker

Vorsitzender des Promotionsausschusses: Prof. Dr.-Ing. Ulrich Appel  
1. Berichterstatter: Prof. Dr.-Ing. Kurt Hoffmann  
2. Berichterstatter: Prof. Dr.-Ing. Berthold Lankl

Tag der Prüfung 29.7.2004

Mit der Promotion erlangter akademischer Grad:  
Doktor-Ingenieur  
(Dr.-Ing.)

Berlin, den 5. Oktober 2004

## Zusammenfassung

Diese Dissertation stellt neuartige Schaltungskonzepte zum Senden und Empfangen von seriellen Datenströmen höchster Datenraten vor. Ziel ist es, eine Anordnung zu finden, die für eine gegebene CMOS-Fertigungstechnik die maximale Übertragungsdatenrate ermöglicht. Ihre Machbarkeit wird anhand einer Beispielimplementierung für eine Datenrate von 10 GBit/s in einem 120 nm-CMOS-Prozess überprüft.

Im ersten Teil der Arbeit werden die Anforderungen an ein Datenübertragungssystem aus Sicht des Schaltungsentwicklers erörtert. Dann wird ein neues Konzept für das Senden von Daten vorgestellt, das als einzig bekanntes eine Senderate ermöglicht, die um ein Vielfaches über der Betriebstaktfrequenz des Senders liegt. Es werden Schaltungen präsentiert, die die Funktion des Konzeptes auf einem Testchip belegt haben. In der Folge wird ein neues Verfahren zum Datenempfang eingeführt. Es erlaubt, analog zum Sender, den Betrieb bei einer gegenüber der Datenrate um ein Vielfaches herabgesetzten Taktfrequenz. Systemsimulationen dienen zur Erläuterung seiner Funktionsweise.

Der Sender/Empfänger vereint große Vorteile gegenüber dem Stand der Technik auf sich:

- symmetrisches Layout
- Einfachheit von Entwurf und Übertragbarkeit auf künftige Fertigungstechnologien
- geringe Leistungsaufnahme
- Einsatz nur eines Oszillators sowohl im Master- als auch im Slave-Modus
- minimale Anzahl schneller Komponenten
- Wegfall eines hochfrequenten Taktbaumes
- Begrenzung der Datenrate nur durch die Bandweite der Ein- und Ausgangsstufen

# Inhaltsverzeichnis

<b>Vorwort</b>	<b>5</b>
Einleitung . . . . .	5
Gliederung . . . . .	6
<b>1 Systemübersicht</b>	<b>7</b>
1.1 Die optische Übertragungsstrecke . . . . .	7
1.2 Netzwerke . . . . .	9
1.3 Ansatzpunkte dieser Arbeit . . . . .	11
<b>2 Anforderungen und Zielsetzung</b>	<b>13</b>
2.1 Allgemeine Anforderungen . . . . .	13
2.2 Spezielle Anforderungen . . . . .	15
2.3 Sonstige Vorbetrachtungen . . . . .	19
2.3.1 Taktfrequenz . . . . .	19
2.3.2 Signalhub $\Delta U$ . . . . .	20
2.4 Zielsetzung . . . . .	21
<b>3 Ein asynchroner Multiplexer mit niedriger Leistungsaufnahme</b>	<b>24</b>
3.1 Vorbetrachtungen . . . . .	24
3.1.1 Einleitung . . . . .	24
3.1.2 Das Schieberegister . . . . .	24
3.1.3 Die binäre Baumstruktur . . . . .	25
3.1.4 Vergleich synchroner Teiler – asynchroner Teiler . . . . .	27
3.1.5 Typische 2:1-Multiplexerschaltungen . . . . .	29
3.2 Die neue Multiplexeranordnung . . . . .	31
3.2.1 Die neue 2:1-Multiplexer-Grundzelle . . . . .	31
3.2.2 Die Vorkodierung der Daten . . . . .	33
3.2.3 Vor- und Nachteile des neuen Multiplexers . . . . .	37
3.2.4 Die Funktionsweise mit herabgesetztem Oszillatortakt . . . . .	38
3.2.5 Ein verbesserter Kodierer . . . . .	40
3.3 Blockdiagramm . . . . .	42
3.4 Schaltungen . . . . .	45
3.4.1 Überblick . . . . .	45

3.4.2	Die Schnittstelle zum Digitalkern <code>in_buf_wtx</code> . . . . .	47
3.4.3	Der 32:16-Multiplexer <code>Mux_32_16</code> . . . . .	50
3.4.4	Der 16:2-Multiplexer <code>new_mux_16_2</code> . . . . .	52
3.4.5	Die Komponenten des 16:2-Multiplexers . . . . .	55
3.5	Meßergebnisse . . . . .	66
<b>4</b>	<b>Ein asynchroner Empfänger</b> . . . . .	<b>70</b>
4.1	Klassifizierung von Empfängeranordnungen . . . . .	70
4.1.1	Open-Loop Architekturen . . . . .	71
4.1.2	Phasenregelkreise . . . . .	71
4.2	Zur Kodierung der Daten . . . . .	73
4.3	Ausgewählte Beispiele aus der Literatur . . . . .	75
4.3.1	Digitale Schleifenfilter . . . . .	75
4.3.2	Phasendetektoren . . . . .	78
4.3.3	Die Frequenzdetektion . . . . .	80
4.3.4	Zusammenfassung der Vorbetrachtungen . . . . .	85
4.4	Die neue Empfängeranordnung . . . . .	86
4.4.1	Pegel- und Flankendarstellung eines NRZ-kodierten Signales . . . . .	86
4.4.2	Die neue Demultiplexergrundzelle . . . . .	86
4.4.3	Blockschaltbild . . . . .	88
4.4.4	Die Datenrückgewinnung . . . . .	91
4.4.5	Das Auslesen des asynchronen Registersatzes . . . . .	96
4.4.6	Die Taktrückgewinnung . . . . .	99
4.4.7	Der Phasenregelkreis . . . . .	105
4.4.8	Die asynchrone Betriebsweise . . . . .	117
<b>5</b>	<b>Fazit</b> . . . . .	<b>121</b>

# Vorwort

## Einleitung

Der immer größer werdende Bedarf an Bandbreite zur Bereitstellung von neuen Informationsdiensten erfordert einen stetig wachsenden Aufwand bei der Weiterentwicklung der Übertragungstechnologien. In nicht allzu ferner Zukunft sollen Privathaushalte flächendeckend Zugang zu Datenraten in der Größenordnung von hundert Megabit pro Sekunde erhalten.

Da sich mediale Angebote wie „Video-on-demand“ nur durchsetzen werden, wenn ein geringer Endkundenpreis für die Zugangstechnologie erreicht wird, stehen der Entwicklung geeigneter Lösungen neben den inhärenten technischen Schwierigkeiten noch marktwirtschaftliche Anforderungen erschwerend entgegen.

So sind marktfähige Preise nur zu erwarten, wenn die günstigste Fertigungstechnologie der Halbleitertechnik benutzt wird, die CMOS-Technologie. Dabei steht außer Frage, daß diese von den drei wichtigen Fertigungstechnologien, der CMOS-, Bipolar- und GaAs-Halbleitertechnologie, unter rein technischen Aspekten die für Hochgeschwindigkeitsdatenübertragung ungeeignetste ist. Da aber für die Wirtschaftlichkeit der Technik insbesondere die Integration der eigentlichen Übertragungseinheiten mit weiteren Funktionen der Datenverarbeitung auf einem Chip entscheidend sein wird, kommt aus bekannten Gründen nur die CMOS-Technologie in Betracht.

Ihre geringere Geschwindigkeit muß also infolge wirtschaftlicher Überlegungen in Kauf genommen und überwunden werden. Dabei gilt es zwangsläufig Strukturen zu wählen, die diesen Nachteil durch den großen Vorteil der CMOS-Technologie aufwiegen können: die Verfügbarkeit hunderttausender Schaltelemente auf einem Chip. Deshalb sind insbesondere parallelisierte Ansätze zu berücksichtigen.

Diese Dissertation stellt eine neuartige Architektur zur Übertragung höchster Datenraten vor. Es werden für einige wichtige Aspekte sowohl des Sendens als auch des Empfangens neue Lösungen eingeführt.

Da ein großer Teil der Arbeit im Rahmen eines industriellen Projektes bei der INFINEON TECHNOLOGIES AG, München, erfolgt ist, werden die vorgestellten allgemeingültigen

Lösungen auf den konkreten Fall einer 10 GBit/s-SDH<sup>1</sup>/SONET<sup>2</sup>/OPTICAL-ETHERNET-Übertragungsstrecke bezogen. Darüber hinaus ist Ziel dieser Arbeit, Lösungen zu finden, die der theoretischen maximalen Übertragungsgeschwindigkeit für eine gegebene Fertigungstechnologie nahekommen. Da die Ansätze auch auf künftige schnellere Technologien übertragbar sein werden, stellen die aus den angeführten technischen Standards folgenden strengen Vorgaben jedoch weniger eine feste Zielsetzung als vielmehr nur einen Maßstab für die Eignung der vorgestellten Techniken dar.

Die allen Betrachtungen zugrundeliegende Fertigungstechnologie war die neueste CMOS-Technologie der INFINEON TECHNOLOGIES AG, ein 120 nm-Prozeß.

## Gliederung

Diese Arbeit gliedert sich in drei Teile. Deren erster im Anschluß an das Vorwort gibt eine Einführung über optische Datenübertragungsnetze und die durch die verschiedenen Standards an sie gestellten Anforderungen. Auf diese Weise werden das Arbeitsziel erläutert sowie Maßstäbe zur Überprüfung des Entwicklungserfolges bereitgestellt.

Der zweite Teil ab Seite 23 stellt eine neue Multiplexeranordnung für die Übertragung von Daten mit einer Rate von 10 GBit/s vor, deren Funktion inzwischen durch einen Testchip nachgewiesen wurde — trotz erheblicher Schwierigkeiten bei der Fertigung funktionstüchtiger Chips in der damals noch in der Entwicklung befindlichen 120 nm-Technologie von INFINEON. Es werden alle neuen Prinzipien dargelegt, die Schaltungen durch Simulationsergebnisse erläutert und abschließend einige Meßergebnisse präsentiert, die die Funktion des resultierenden Halbleiterchips belegen.

Der dritte Teil ab Seite 69 gilt einer neuen komplexen Empfängerstruktur mit Taktrückgewinnung. Diese Architektur weist einige große Vorteile gegenüber traditionellen Ansätzen auf; dennoch wurde sie im Rahmen des industriellen Projektes nicht berücksichtigt, weil ihre Eignung für das Entwicklungsziel nicht gesichert erschien und tiefere Untersuchungen im engen Zeitplan des Projektes unmöglich waren.

Da die schaltungstechnische Umsetzung einer kompletten Empfängerstruktur den Rahmen einer Dissertation sprengen würde, mußte hier auf eine Realisierung als Chip verzichtet werden. Es werden jedoch technische Rückschlüsse aus dem bei der Entwicklung des industriellen Chips verfolgten Ansatz in die systemtechnischen Untersuchungen einfließen, um die Zuverlässigkeit der gezogenen Schlußfolgerungen sicherzustellen.

---

<sup>1</sup>„SDH“ steht für „synchronous digital hierarchy“

<sup>2</sup>„SONET“ steht für „synchronous optical network“

# 1 Systemübersicht

## 1.1 Die optische Übertragungsstrecke

Die Datenübertragung bei einer Geschwindigkeit von 10 GBit/s und höher kann über längere Strecken nur auf optischem Wege erfolgen. Gegenwärtig werden zwar vielerorts noch kabelgebundene Lösungen bevorzugt, weil entsprechende Infrastrukturen bereits vorhanden sind, doch werden jenseits der Grenze von 1 GBit/s auch im Nahbereich Glasfasern die Kupferleitungen vorhandener lokaler Netze nach und nach ersetzen.

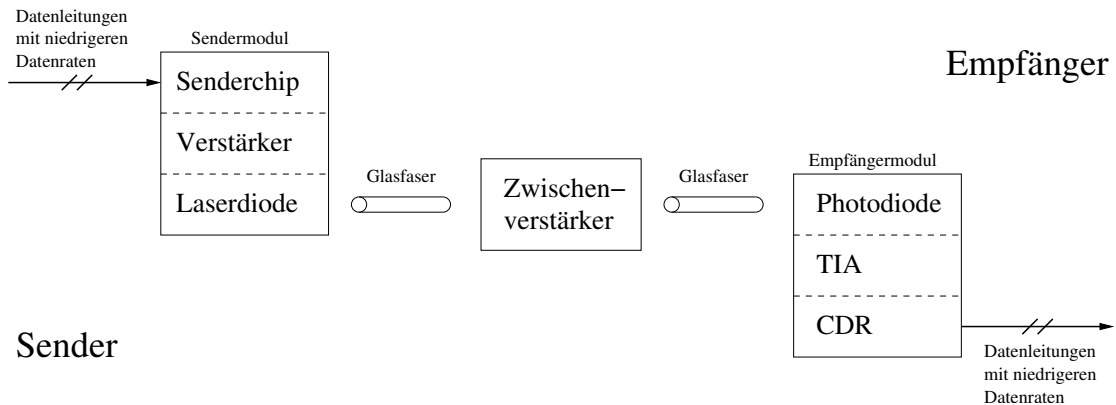
Im Folgenden wird ein kurzer Überblick über eine typische Glasfaserstrecke zwischen zwei Punkten gegeben, damit der genaue Ansatzpunkt der hier vorgestellten Arbeit lokalisiert werden kann. Abbildung 1.1 zeigt ein Diagramm einer solchen optischen Datenübertragungsstrecke.

Die Daten fließen vom Sendermodul („Transmitter“) auf der linken zum Empfängermodul („Receiver“) auf der rechten Seite. Normalerweise werden beide Module an jedem Ende der Datenstrecke zu einem einzigen zusammengefaßt, so daß zwei sogenannte „Transceiver“ bidirektional über zwei Glasfasern miteinander kommunizieren.

Die zu sendenden Daten liegen auf mehreren Leitungen niedriger Datenrate an und werden innerhalb des als „Senderchip“ bezeichneten Blockes auf nur eine Leitung mit 10 GBit/s gemultiplext. Außerhalb des Chips werden sie noch einmal verstärkt, bevor die Laserdiode die Umwandlung des elektrischen Signales in ein optisches vornimmt. Die Daten werden trotz der immanenten Nachteile im NRZ-Format, also als einfache Serie gleichlanger Nullen und Einsen, gesendet. Auf diese Problematik wird auf Seite 70 im einleitenden Abschnitt über Empfängerarchitekturen ausführlicher eingegangen.

Die begrenzenden Größen für die Länge der optischen Übertragungsstrecke sind Pulsdämpfung und -dispersion. Die Dämpfung entsteht durch die optische Abschwächung der Amplitude des Pulses durch das Medium, während die Dispersion durch die optisch unterschiedlich langen Wege der nicht absolut parallel in die Glasfaser eintretenden Quanten durch Reflexion an der Faserwand hervorgerufen wird. Die Dispersion bewirkt, daß sich ein Puls, ähnlich einem elektrischen Puls durch Einwirken eines Kapazitätsbelages, über einen längeren Zeitraum verteilt und dabei die bekannte abgerundete Glockenform einnimmt. Hierdurch kann es im Störfall zu *Inter-Signal Interferenzen* (ISI) kommen.

Der Dispersionseffekt kann durch Verwendung sogenannter Monomode-Fasern gemildert werden, so daß in der praktischen Anwendung die Signaldämpfung zur begrenzenden Grö-



**Abbildung 1.1:** Übersicht über eine typische optische Datenübertragungsstrecke

ße für die Länge der optischen Übertragungsabschnitte wird. Aus diesem Grund müssen in längeren Strecken Zwischenverstärker eingefügt werden, die die Pulse des Datenstromes auffrischen.

Diese Zwischenverstärkung wird im elektrischen Bereich durchgeführt, da sich noch keine brauchbare Methode gefunden hat, eine Verstärkung ohne optoelektrische Wandlung zu verwirklichen. Dies wäre jedoch wünschenswert, da die Wandlung vom optischen ins elektrische Signal und umgekehrt sowie die sonstigen notwendigen Schritte zur Auffrischung des Signales Latenzzeiten in die Übertragungsstrecke einbringen.

Auf der Empfängerseite wird wiederum durch eine Photodiode das empfangene Signal in ein elektrisches umgewandelt. Ein diskreter sog. *Transimpedance Amplifier* (TIA) oder Transimpedanzverstärker verstärkt es dann auf einen Pegel, der im folgenden Empfängerchip verarbeitet werden kann. Der Transimpedanzverstärker muß eine hohe Eingangsempfindlichkeit aufweisen, um das schwache Signal der Photodiode verstärken zu können. Außerdem muß er besonders rausch- und verzerrungsarm sein, da Störungen an diesem Punkt des Gesamtsystems die gravierendste Wirkung hinsichtlich des Signal/Rausch-Verhältnisses und damit der Bitfehlerrate hätten.

Die für das Empfangen vorgesehenen Chipblöcke werden gewöhnlich als *Clock-and-Data-Recovery* (CDR), also Takt-/Datenrückgewinnung, bezeichnet. Der Name dieses Teiles der Übertragungsstrecke nennt bereits seinen Zweck: die Rückgewinnung der übertragenen Daten sowie die Extraktion des ihnen inhärenten Taktes. Außerdem werden die Daten gedemultiplext, also die enthaltene Information auf mehrere Datenleitungen niedrigerer Geschwindigkeit verteilt.



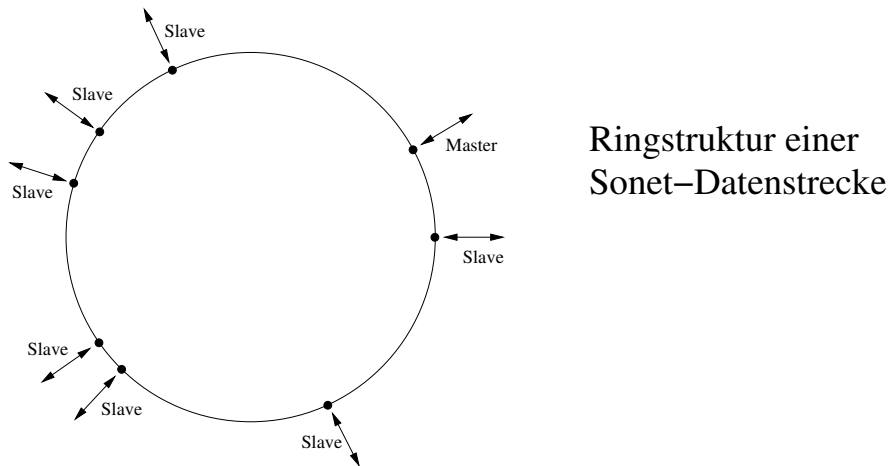


Abbildung 1.2: Ein als Ring ausgeführtes Datennetz

## 1.2 Netzwerke

Mehrere optische Datenverbindungen werden zu einem Netzwerk verbunden, wobei auch elektrische Verbindungen einbezogen sein können. Diese sind dann jedoch gewöhnlich Anschlüsse von Subnetzen mit niedrigerem Datendurchsatz. Eine übliche Netzstruktur ist die des Ringes. Abbildung 1.2 zeigt ein Schema eines solchen Netzwerkringes.

Bei einer Ringstruktur werden mehrere Datenknoten auf einem Ring platziert. Bei einem Ring gemäß dem SDH/SONET-Standard bildet jeder Datenknoten einen Add-/Drop-Multiplexer, d.h. an jedem Knoten wird der für das vom Knoten ausgehende Subnetz bestimmte Datenstrom aus dem Ring in das Subnetz geleitet und der aus dem Subnetz stammende Datenstrom wiederum an dessen Stelle in den Datenring eingespeist.

Die SONET-Netzstruktur sieht vor, daß jede Teilstrecke des Gesamtnetzes ein ganzzahliges Vielfaches eines Datendurchsatzes von 51,85 MBit/s aufweist. Auf diese Weise kann das gesamte Netzwerk synchron, also von einer einzigen Taktquelle gesteuert, arbeiten. Die Geschwindigkeit von 10 GBit/s entspricht OC-192. Dabei steht „OC“ für *optical carrier*, während „192“ den oben erwähnten Faktor bedeutet, der mit 51,85 multipliziert werden muß, um auf den Datendurchsatz in MBit/s zu schließen.

Der SDH-Standard ist dem etwas älteren SONET-Standard sehr ähnlich. So entspricht die SONET-Datenrate OC-3 (155,55 MBit/s) genau dem Durchsatz des SDH-1, OC-192 folglich der Rate des SDH-64. In einigen Punkten hinsichtlich der physikalischen Parameter des Systems ist der SDH-Standard etwas strenger, weshalb der SONET-Standard als Untermenge des SDH-Standards angesehen werden kann. Folglich erfüllt ein System, das

die Vorgaben des SDH-Standards erfüllt, automatisch auch die des SONET-Standards. Optische Netze nach dem SONET-Standard sind in Nordamerika und Japan, solche nach dem SDH-Standard in Europa verbreitet. Die Forschungsliteratur nimmt hauptsächlich Bezug auf den SONET-Standard, weshalb er auch im Folgenden als Maßstab benutzt wird.

Die synchrone Funktionsweise eines Datennetzes hat viele Vorteile. So kann an keinem Knoten die Notwendigkeit entstehen, Datenpakete zwischenspeichern, weil etwa die empfangene Datenmenge — und sei es auch nur kurzfristig — die Sendekapazität überstiege. Dadurch werden die verwendeten Protokolle einfacher, da keine aufwendigen Rückmeldungen bzw. Empfangsbestätigungen gebraucht werden.

Für den Aufbau eines synchronen Netzwerkes ist es unerlässlich, im gesamten Netzwerk nur eine einzige Taktquelle einzusetzen. Denn wie gut auch immer zwei Uhren aufeinander abgestimmt wären, käme es doch stets nach einer endlichen Betriebsdauer zu zeitlichen Verschiebungen zwischen den einzelnen Takten. Das geringfügig langsamer getaktete Subnetz würde zwingend einen mit der Zeit wachsenden Datenrückstau im Verbindungsknoten von der schneller getakteten Netzhälfte zur Folge haben.

Wenn aber nur eine Taktquelle vorhanden ist und der Takt, um Bandbreite zu sparen, nicht übertragen wird, müssen alle anderen Netzknoten die Taktinformation aus den empfangenen Datenströmen zurückgewinnen und den extrahierten Takt ihrerseits zum Senden verwenden. Diese Knoten arbeiten im sog. Slave-Modus, da sie abhängig von den Knoten arbeiten, mit denen sie kommunizieren.

An nur einem einzigen Knoten wird aus einer externen Taktreferenz der Systemtakt eingespeist. Dabei ist dieser Takt gewöhnlich ein Takt relativ niedriger Frequenz, aus dem der eigentliche Datentakt durch Multiplikation gewonnen wird. Üblich sind 155 MHz oder 620 MHz. Der Knoten, an dem der Netzwerktakt eingespeist wird, arbeitet im Master-Modus. Da sehr viel weniger Knoten im Master-Modus betrieben werden als im Slave-Modus, werden für diese keine eigenen Produkte entwickelt. Vielmehr muß ein Netzwerkmodul nach Wahl des Anwenders in beiden Betriebsmodi arbeiten können. In der Abbildung 1.2 sind die einzelnen Knoten entsprechend ihrem Betriebsmodus beschriftet.

Weil der Takt an jedem Netzknoten aus dem Datenstrom extrahiert und wiederum für das Senden benutzt werden muß, verschlechtert er sich durch Störeinflüsse und technische Limitierungen von Stufe zu Stufe. Offensichtlich kann die fehlerfreie Übertragung von Daten für das gesamte Netz nur dann gewährleistet werden, wenn die Taktqualität nicht zu schlecht wird. Da eine sehr niedrige Bitfehlerrate und eine große Zahl von Netzknoten wünschenswert sind, stellen die technischen Standards sehr hohe Ansprüche an die Qualität der Taktrückgewinnung. Sie ist deshalb auch die größte Herausforderung bei der Entwicklung eines Transceivers für höchste Datenraten.

Die Beschränkung auf nur eine einzige Taktquelle stellt ein theoretisches Ideal dar. Natürlich kann wegen der von jedem Knoten eingebrachten Störung des immer wieder ex-

trahierten Taktes nur eine endliche Zahl von Knoten auf einem optischen Datenring platziert werden. Da aber in einem weltumspannenden Netz wie dem Internet oder dem Telefonnetz eine enorme Zahl von Knoten miteinander kommunizieren können muß, müssen lokale Unternetze, die jedes über eine Taktquelle verfügen, miteinander verbunden werden. Dabei kommt es zum oben erläuterten Problem geringer Verschiebungen der einzelnen Taktquellen zueinander.

Neben asynchroner und synchroner Betriebsweise wird deshalb noch als Zwischenbereich der plesiosynchrone Betrieb definiert. Streng genommen handelt es sich bei diesem um einen asynchronen Betrieb, jedoch wird garantiert, daß die beiden Takte höchstens um ein geringes spezifiziertes Maß voneinander abweichen. Die Datenstruktur der optischen Standards enthält in jedem Datenblock ein leeres Byte, so daß an einer plesiosynchronen Schnittstelle empfangene Daten einer leicht höheren Datenrate vollständig über eine etwas langsamere Datenverbindung gesendet werden können, indem regelmäßig ein Byte des Datenstromes in das freie Byte eingefügt wird [1]. Im umgekehrten Fall kann ein weiteres Leerbyte in den Datenrahmen eingefügt werden. Da die Datenraten nur um ein sehr geringes Maß abweichen dürfen, reicht der zusätzliche Raum in den Datenblöcken in jedem Falle aus.

### 1.3 Ansatzpunkte dieser Arbeit

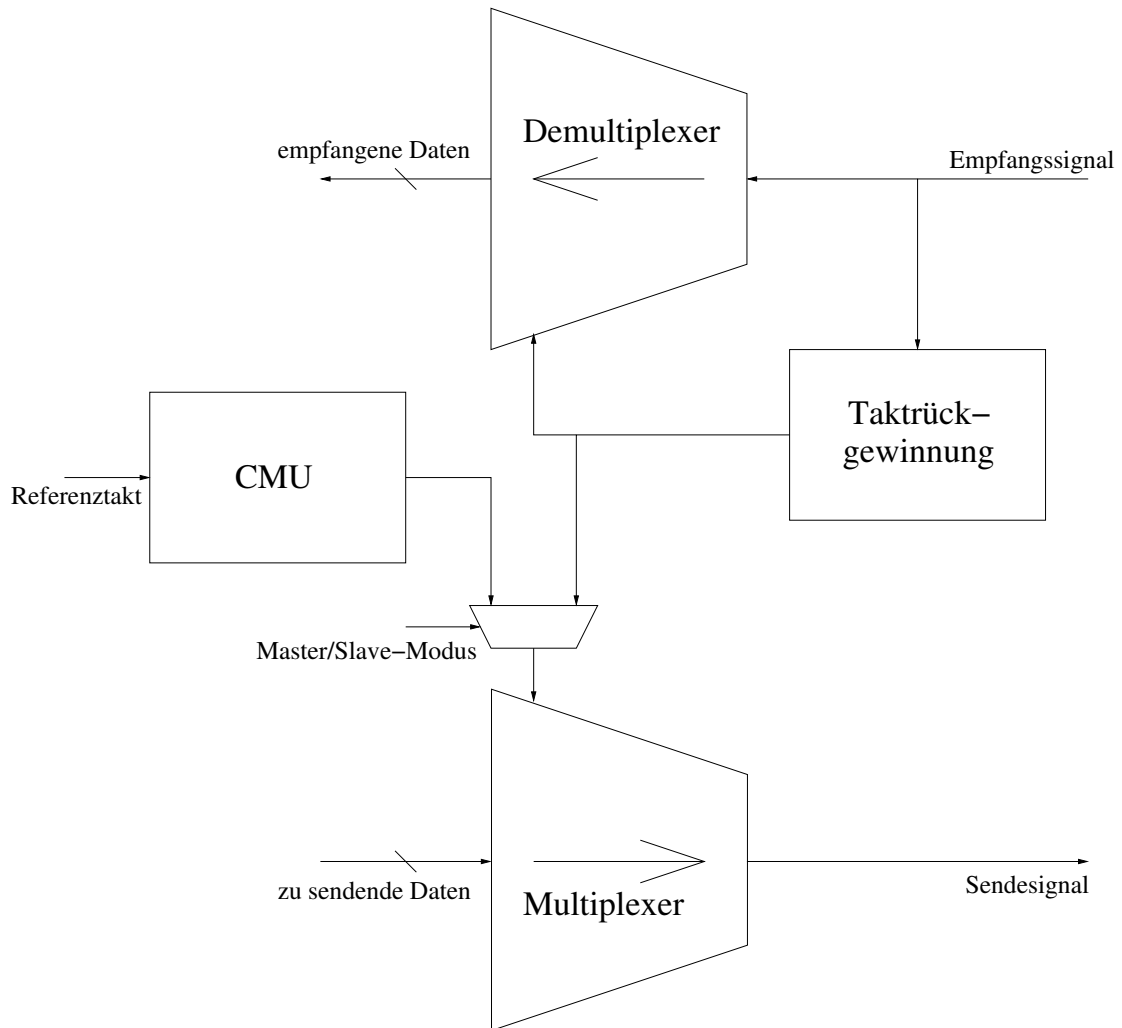
In dieser Arbeit werden ausschließlich Ansätze vorgestellt, die innerhalb der Funktionalität des eigentlichen Transceiver-Chips liegen. Er umfaßt den Multiplexer, den Demultiplexer, die Taktrückgewinnung und die Taktgeneration durch einen Taktmultiplizierer (*Clock Multiplier Unit* CMU). Diese Blöcke sind in Abbildung 1.3 in einer Übersicht dargestellt.

Die CMU wird nur benötigt, wenn der Chip im Master-Modus betrieben wird. Gewöhnlich wird sie als Phasenregelkreis bzw. *Phase-locked Loop* (PLL) konzipiert. Wenn auch die schaltungstechnische Entwicklung einer CMU für die für das gesetzte Ziel benötigte Geschwindigkeit äußerst schwierig ist, so sind doch die grundlegenden Prinzipien allgemein bekannt, so daß sich eine umfangreiche Arbeit über diesen Teil erübrigt.

Von den verbleibenden Teilen läßt sich der Multiplexer als einzelner Block aus dem Ganzen herauslösen, da seine Funktionalität nicht von den anderen Blöcken abhängt. Als einzige Schnittstelle zu ihnen benötigt er lediglich einen Takt, je nach Betriebsmodus wahlweise den durch die CMU generierten oder den aus den empfangenen Daten zurückgewonnenen, sowie selbstredend die Eingangsdaten. Ein neuartiger Multiplexer wird getrennt von den sonstigen Untersuchungen im ersten Hauptteil ab Seite 23 vorgestellt.

Demultiplexer und Taktrückgewinnung lassen sich im allgemeinen funktional nicht so klar voneinander trennen, wie es das Blockdiagramm suggeriert. So muß in jedem Falle der

rückgewonnene Takt zum Abtasten und Demultiplexen der empfangenen Daten benutzt werden, während gleichzeitig die Abtastung der empfangenen Daten die benötigte Information über die Phasenlage zwischen einlaufenden Daten und dem lokal reproduzierten Takt liefert. Folglich werden diese beiden Blöcke gemeinsam im zweiten Hauptteil der Arbeit ab Seite 69 untersucht. In dem dort vorgestellten Ansatz sind Taktrückgewinnung und Datenabtastung aufgrund des völlig neuartigen Prinzips untrennbar miteinander verbunden.



**Abbildung 1.3:** Übersicht über die Funktionsblöcke eines typischen Transceiverchips

## 2 Anforderungen und Zielsetzung

### 2.1 Allgemeine Anforderungen

Es gilt, technische Anforderungen für die Realisation von Hochgeschwindigkeitsdatenübertragungsstrecken zu definieren und die wichtigsten Probleme der Umsetzung zu identifizieren. Die so gewonnene Einsicht soll benutzt werden, um geeignete Vorgaben für eine neue Sender/Empfängerstruktur zu entwickeln. Zuerst muß hierzu jedoch die Einbindung des Transceiver-Chips in sein Umfeld betrachtet werden.

Bei einer optischen Übertragungsstrecke hat der Sender/Empfänger-Chip lediglich Hochgeschwindigkeitsschnittstellen zur Laserdiode für das Senden, und zur Photodiode, bzw. dem ihr nachgeschalteten TIA, für das Empfangen. Je nach Ausführung kann die Laserdiode ungetaktet oder getaktet sein. Die getaktete stellt dabei die höheren Anforderungen an das Gesamtsystem, weil sie einen mit den Sendedaten synchronen Takt benötigt. Im Fall eines 10 GBit/s-Übertragungssystems, müßte der CMOS-Chip folglich einen 10 GHz-Takt elektrisch treiben können, was die technische Machbarkeit an diesem Punkt mindestens in Frage stellt.

Wenn die Aufgabenstellung lautet, eine Empfängerarchitektur zu entwickeln, die sich der theoretisch höchsten für einen gegebenen CMOS-Prozeß erreichbaren Geschwindigkeit annähert, dann stellt der gegenüber den Daten doppelte Bandweitenbedarf des Taktsignales eine unnötige Begrenzung für die Erhöhung der Datenrate dar. Ungetaktete Laserdioden sind deshalb aus Sicht des Systementwicklers in jedem Falle zu bevorzugen. Der in dieser Arbeit vorgestellte Sender/Empfänger sucht Datenraten zu erreichen, die die Bereitstellung eines zugehörigen Taktes nicht mehr erlauben. Deshalb wird der Einsatz einer ungetakteten Laserdiode vorausgesetzt.

Die Schnittstelle zum meist als bipolarer Schaltkreis ausgeführten TIA gestaltet sich einfach; sie beschränkt sich auf das durch ihn verstärkte Empfangssignal. Ehrgeizige Pläne sehen jedoch auch die Integration des TIA auf dem CMOS-Chip vor. Die Hauptschwierigkeiten werden dabei die geringe Eingangsempfindlichkeit von CMOS-Transistoren sowie die Einstreuung von deterministischem Rauschen aus den getakteten Chipteilen darstellen.

Das erste Problem liegt in der Natur der CMOS-Transistoren begründet, deren Source-Drain-Strom nur quadratisch von der Gate-Source-Spannung abhängt im Gegensatz zur exponentiellen Abhängigkeit des Kollektor-Emitter-Stromes des Bipolartransistors von

der Basis-Emitter-Spannung. Eine mögliche Lösung könnte der Einsatz eines BiCMOS-Prozesses sein, mit dem einige kleinere Funktionsteile des Gesamtsystems in Bipolar-schaltungstechnik auf demselben CMOS-Chip verwirklicht werden könnten. Dieser Idee stünde jedoch der Wunsch eines billig in Massen produzierbaren Chips wegen des aufwendigeren Produktionsprozesses zumindest bedingt entgegen.

Die Probleme der Einstreuung von Störsignalen aus anderen Chipteilen können künftig nicht mehr allein durch klassische Mittel wie die sog. Guard-Ringe u. ä. gelöst werden und müssen deshalb ebenfalls auf produktionstechnischer Ebene angegangen werden. So wird hier der Trend zu *Silicon-on-Insulator*-Prozessen (SOI) wegen der wesentlich geringeren Substratkopplung sehr willkommen sein. Zusätzlich könnten Techniken wie um die störempfindlichen Bauteile in das Substrat geätzte Gräben zum Einsatz kommen.

Die Störmechanismen verschiedener Chipelemente untereinander bedürfen generell noch weitreichender Grundlagenforschung. Noch gibt es keine brauchbaren Modelle, die andere Kopplungsmechanismen als den rein kapazitiven berücksichtigen und somit zur Simulation im Schaltungsentwurf eingesetzt werden könnten. Aus den aufgeführten Gründen können noch nicht einmal Schätzungen über die technische Machbarkeit einer Integration des TIA in den Transceiver-Chip innerhalb eines im Maßstab der Halbleitertechnik überschaubaren Zeitraumes getroffen werden. Der TIA wird auch in näherer Zukunft als separater Bipolarbaustein verwirklicht werden müssen, weshalb er nicht Gegenstand dieser Arbeit ist.

Die vom TIA bereitgestellten Pegel weisen im ungünstigsten Fall nur etwa 50 mV single-ended Signalhub am Eingangsverstärker des CMOS-Chips auf, weshalb das Signal auch im Weiteren so störungsfrei wie möglich verarbeitet werden muß. Differenzielle Signale sind dabei selbstverständlich.

Um jedoch ein so schwaches Signal auf ein Maß verstärken zu können, das sich ohne Schwierigkeiten abtasten läßt, muß die kapazitive Last, die dem Verstärker folgt, minimiert werden, weil der für die Verstärkung zu betreibende Aufwand bei hoher Signalbandbreite direkt abhängig von dieser Last ist. So haben erste Simulationen gezeigt, daß bei einer Bandbreite von ca. 8 GHz, die für ein NRZ-kodiertes Signal von 10 GBit/s minimal benötigt wird, jede Stufe in einer Kette von Verstärkern nur eine Verstärkung von etwa 1,2 bewirken kann! Da aber das empfangene Signal mit jeder Stufe, die es passieren muß, auch gestört wird, sind der Anzahl möglicher Stufen Grenzen gesetzt. Eine neue Empfängeranordnung mit geringer Eingangslast wäre folglich von großem Vorteil.

Gewöhnlich werden Sender/Empfänger bzw. die Module, in denen sie eingesetzt werden, in größeren Netzwerkknoten verschaltet, so daß auf kleinem Raum eine Vielzahl solcher Chips zusammenkommen können. Da diese Netzknoten möglichst wartungsunanfällig sein sollen, ist es wünschenswert, daß die einzelnen Module eine geringe Leistungsaufnahme aufweisen, weil so unzuverlässige mechanische Komponenten zur aktiven Kühlung wegfallen können, die Stromversorgung vereinfacht wird und die Betriebskosten sinken.

Deswegen lautet eine weitere wichtige Anforderung an eine neue Empfängerarchitektur, möglichst wenig Leistung zu verbrauchen. Neben der Hochintegrierbarkeit von CMOS-Chips ist dies ein weiterer Grund für den Einsatz dieser Schaltungstechnik, da die Aussicht besteht, daß sich weniger leistungsintensive Lösungen als in Bipolar- oder GaAs-Technologie entwickeln lassen.

Die letzte allgemeine Anforderung folgt wiederum aus wirtschaftlichen Gesichtspunkten implizit aus der Wahl des Werkmaterials: der Hauptvorteil der CMOS-Technologie, die Möglichkeit des Hochintegrierens zusammen mit weniger kritischen Komponenten der Datenverarbeitung auf einem Chip, läßt sich nur dann wirtschaftlich nutzen, wenn die Ausbeute der  $6\sigma$ -Vorgabe folgt. Damit dies für den gesamten Transceiverchip gilt, muß jedoch jede einzelne Funktionseinheit, also auch die Hochgeschwindigkeitsschnittstelle, die  $6\sigma$ -Bedingung erfüllen.

Als Grundlage für diese Vorgabe wird angenommen, daß die verschiedenen Prozeßparameter einer Gauß'schen Verteilung unterliegen. Alle Funktionsgruppen auf einem Chip müssen nun bei einer Variation aller Prozeßparameter bis zum Sechsfachen der Standardabweichung  $\sigma$  in beliebiger Kombination mit den verschiedenen spezifizierten Betriebsparametern zuverlässig arbeiten.

In Zahlen ausgedrückt bedeutet die Ausbeutevorgabe, daß von einer Million gefertigter Teile trotz der Prozeßschwankungen lediglich 3,4 die Funktion versagen dürfen. Der Entwurf der Hochgeschwindigkeitsmodule ist durch die Ausbeutevorgabe sehr stark erschwert.

Zusammenfassend seien die allgemeinen Anforderungen an eine neue Sender/Empfängerarchitektur noch einmal aufgezählt:

1. empfängerseitig geringe Eingangskapazität,
2. geringer Leistungsverbrauch,
3. zuverlässige Funktion trotz weiter Schwankung der Fertigungs- und Betriebsparameter.

## 2.2 Spezielle Anforderungen

Ein SONET-Netzwerkchip muß, wie im Abschnitt 1.2 auf Seite 10 erläutert, zwei Betriebsmodi beherrschen, den Master- und den Slave-Modus. Hierbei stellt der häufiger benötigte Slave-Modus den Entwickler vor wesentlich größere Schwierigkeiten als der Master-Modus. Vor allem muß eine zuverlässige Methode gefunden werden, einen Takt hinreichender Genauigkeit aus den empfangenen Daten zu extrahieren.

Diese Genauigkeit wird durch die optischen Standards anhand verschiedener Maßzahlen vorgegeben, die jede einzelne Komponente des Netzes garantieren muß, damit die Funktion des gesamten Netzwerkes gesichert ist. Zum gegenwärtigen Zeitpunkt stehen jedoch nur aus den korrespondierenden Standards für niedrigere Datenraten abgeleitete vorläufige Zahlen zur Verfügung [2], weil die Standardisierung für die Geschwindigkeit von 10 GBit/s noch nicht abgeschlossen ist.

Die einzige neben der Datenrate für die Anwendung relevante Maßzahl eines Datennetzes ist die Bitfehlerrate bei der Datenübertragung. Sie sollte in einer Größenordnung von  $10^{-12}$  liegen. Insgesamt wird die Bitfehlerrate von der Interaktion vieler verschiedenartiger Komponenten verschiedener Hersteller abhängen, weshalb sie im Test nur für höchstens einige konkrete Fälle gemessen werden könnte. Deshalb müssen andere Vorgaben gefunden werden, die bei beliebigen Netzaufbauten eine niedrige Bitfehlerrate garantieren. Die durch den SONET-Standard vorgegebenen Maßzahlen und ihre Meßweisen werden im folgenden kurz erläutert.

### Jitter-Generation

Die wichtigsten beiden Maßzahlen betreffen den Jitter des Sendesignales und damit implizit den Jitter des — je nach Betriebsmodus — extrahierten oder generierten Sendetaktes. Gemessen wird die Schwankung des Zeitpunktes der Nulldurchgänge<sup>1</sup> der steigenden und fallenden Flanken des Sendesignales. Dabei wird erstens der *absolute Jitter* und zweitens der sog. *rms-Jitter* gemessen.

Beim absoluten Jitter werden die Extrema der zeitlichen Abweichung der Nulldurchgänge vom Ideal bestimmt, während der rms-Jitter den rms-Wert<sup>2</sup> dieser zeitlichen Abweichungen vom Ideal angibt. Der erlaubte Wert für den absoluten Jitter des Sendesignals beträgt  $0,1 \text{ UI}_{pp}$ <sup>3</sup> oder  $10 \text{ ps}_{pp}$ , der Wert für den rms-Jitter  $1 \text{ ps}_{rms}$ . Bei der Messung wird der niederfrequente Jitter unterhalb von 4 MHz aus dem Datensignal herausgefiltert. Die 4 Mhz-Grenze ergibt sich aus der durch die Standards vorgeschriebenen Tolerierung von niederfrequentem Jitter auf der Empfängerseite. Vgl. hierzu den Abschnitt zur Jitter-Tolerance.

Der Wert für den absoluten Jitter erscheint zwar deutlich höher und damit weniger streng als der für den rms-Jitter, doch stellt er gewöhnlich die härteren Anforderungen an das Gesamtsystem, da in einem Digitalchip nicht von einem rein stochastischen Rauschen

---

<sup>1</sup>Bei einem NRZ-Signal dürfte eigentlich nicht von einem „Nulldurchgang“ gesprochen werden, da üblicherweise die Null einem der beiden Pegel zugewiesen ist. Da aber allgemein differentielle Signale verwendet werden, soll der Begriff hier um der Einfachheit willen verwendet werden.

<sup>2</sup>„rms“ steht für „root-mean-square“ und bedeutet die Quadratwurzel des Mittels der Quadrate der Meßgröße

<sup>3</sup>„UI“ steht für „Unit Intervals“ und bedeutet die Dauer eines Bits bei der angestrebten Datenrate, bei 10 GBit/s also 100 ps



ausgegangen werden kann. Wegen der deterministischen Natur des Rauschens in einer getakteten Umgebung kann es eher zu kurzzeitigen größeren Abweichungen kommen, die wegen der Mittelwertbestimmung bei der Berechnung des rms-Jitters auf dessen Wert keine besondere Wirkung haben, während sie bei der Bestimmung des absoluten Jitters sehr wahrscheinlich die Maximalwerte stellen.

Da der gemessene Wert des absoluten Jitters mit der Meßzeit steigt, ist eine Vergleichbarkeit verschiedener Messungen nur gegeben, wenn diese über den gleichen Meßzeitraum vorgenommen wurden. Gewöhnlich wird die Messung des absoluten Jitters über 60 Sekunden durchgeführt.

### Jitter-Transfer

Eine weitere wichtige Vorgabe betrifft die Übertragung des Jitters des empfangenen Datensignales auf das Sendesignal bei Betrieb im Slave-Modus. Die Taktextraktion in der CDR-Einheit muß den empfangenen Jitter filtern, so daß effektiv der extrahierte Takt gegenüber dem den empfangenen Daten innewohnenden verbessert wird. Der SONET-Standard schreibt eine Tiefpaßcharakteristik erster Ordnung mit einer Grenzfrequenz von 120 kHz für die Filterung des Eingangsjitters vor. Es wird also vom Jitter des empfangenen Datensignales nur der tieferfrequente, gewöhnlich engl. „*wander*“ bezeichnete, Teil auf das Sendesignal übertragen.

Die Grenzfrequenz liegt sehr niedrig, so daß sehr viel Jitter aus dem empfangenen Datensignal bei der Erzeugung des Sendetaktes unterdrückt wird. Die Jitter-Transfer-Charakteristik ist allerdings mit der Problematik der Jitter-Generation verzahnt, da die üblicherweise zur Takterzeugung eingesetzten PLLs das Phasenrauschen ihres Oszillators mit einer Hochpaßcharakteristik mit der gleichen Grenzfrequenz übertragen, wie sie das Phasenrauschen aus dem Referenzeingang tiefpaßfiltert. In anderen Worten: je mehr die PLL das Eingangsrauschen unterdrückt, desto schwieriger wird es, die Anforderungen der Jitter-Generation einzuhalten, weil immer mehr lokal erzeugtes Phasenrauschen auf den Ausgang übertragen wird [3].

Die für die Systementwicklung wichtige Folgerung aus der durch die SONET-Standards vorgegebene Grenzfrequenz für den Jitter-Transfer ist, daß auch die Grenzfrequenz der zur Taktrückgewinnung eingesetzten lokalen PLL 120 kHz betragen muß.

### Jitter-Peaking

Es muß hier in Erinnerung gerufen werden, daß Sender/Empfänger gewöhnlich in Ringen angeordnet werden. Da jedes Glied dieses Ringes selbst Rauschen erzeugt und dieses nur bedingt durch die beim Jitter-Transfer definierte Filteroperation unterdrückt wird, summiert sich der Rauschbeitrag aller Sender/Empfänger in deren Durchlaßbereich.

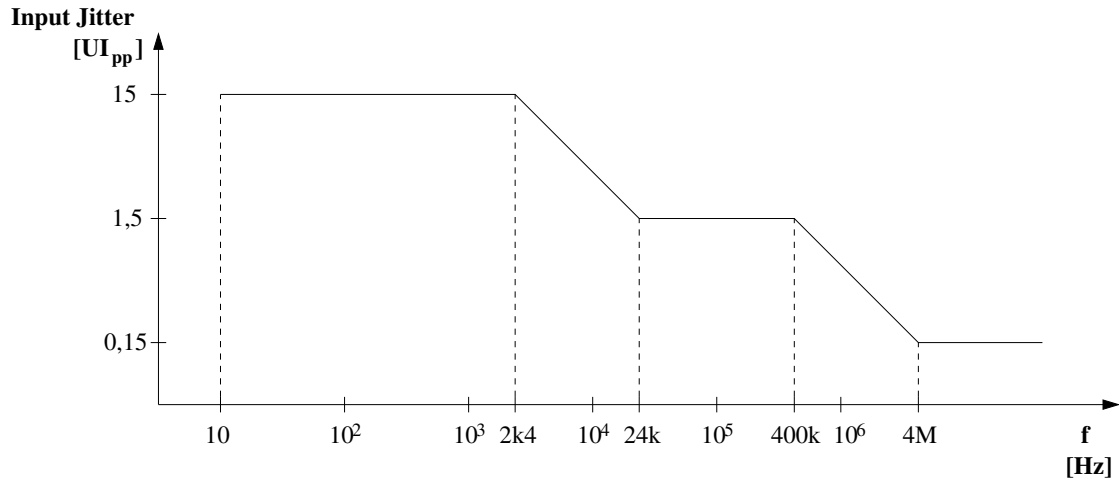


Abbildung 2.1: Maske für den Jitter-Tolerance Test

Weist der Durchlaßbereich eine Welligkeit auf, kann es in einzelnen Frequenzbereichen zu einer Verstärkung des Rauschens kommen, das unter Umständen so groß würde, daß der zuverlässige Betrieb des Datenringes nicht mehr möglich wäre. Deshalb darf die Jitter-Transfer-Charakteristik der PLL keinen Bereich mit einer Verstärkung größer als 0,1 dB aufweisen.

### Jitter-Tolerance

Die Jitter-Tolerance macht Vorgaben darüber, wie robust die Taktrückgewinnung gegenüber dem Jitter sein muß, der dem empfangenen Signal innewohnt. Es wird eine Maske vorgegeben, die Frequenz und Pegel des Eingangsjitters, den die Schaltung tolerieren können muß, beschreibt. Abbildung 2.1 zeigt den Frequenzverlauf mit den vorgegebenen Eckwerten.

Für die Messung wird angenommen, daß der Jitter eine sinusförmige Charakteristik aufweist. Seine Amplitude wird in Einheitsintervallen angegeben, für den Fall einer Datenrate von 10 GBit/s folglich in Einheiten von 100 ps. Die Amplitude kann, wie in der Graphik zu sehen ist, bis zum Fünfzehnfachen einer Bitlänge betragen. Offensichtlich muß der zur Datenabtastung eingesetzte lokale Takt dieser Verschiebung folgen, da andernfalls der Abtastzeitpunkt außerhalb des abzutastenden Bits fallen würde.

Naturgemäß kann die Taktrückgewinnung jedoch nicht jeder beliebigen Verschiebung des Datensignales folgen. Da oberhalb einer Jitterfrequenz von ungefähr 4 MHz die Amplitude desselben ausreichend kleiner ist als die Bitlänge, genügt es, wenn die Taktrückgewin-

nung dem Jitter bis zu dieser Grenzfrequenz folgen kann. Diese Schlußfolgerung bedeutet aber auch, daß die Bandweite der zur Taktrückgewinnung eingesetzten PLL wenigstens 4 MHz betragen muß.

Diese Forderung an die PLL widerspricht offensichtlich der durch den Jitter-Transfer vorgegebenen Schleifenbandweite von nur 120 kHz. In der Praxis werden deshalb gewöhnlich zwei komplette PLLs vorgesehen, deren erste mit einer Bandweite von 4 MHz die eigentliche Takt- und Datenrückgewinnung durchführt. Die zweite PLL mit einer Schleifenbandweite von 120 kHz filtert den zurückgewonnenen Takt, so daß er die durch den Jitter-Transfer gestellten Forderungen an die Taktreinheit erfüllt.

All dies bedeutet einen erheblich gesteigerten schaltungstechnischen Aufwand, der sich auch im Leistungsbedarf des gesamten Sender/Empfänger-Modules deutlich negativ niederschlägt. Verschlimmernd kommt hinzu, daß sich zwei auf dem gleichen Chip eingesetzte Oszillatoren über die Substratkopplung stören. Dieser Effekt wird Kreuzmodulation genannt und erhöht das Phasenrauschen beider Oszillatoren. In dieser Arbeit wird deshalb ein vollkommen neuartiger Ansatz vorgestellt, der die beiden widersprüchlichen Vorgaben zur Schleifenbandweite der Taktrückgewinnung in einer einzigen PLL vereint.

## 2.3 Sonstige Vorbetrachtungen

### 2.3.1 Taktfrequenz

In klassischen Ansätzen zur Takt-/Datenrückgewinnung wird normalerweise der Datentakt in seiner vollen Geschwindigkeit zurückgewonnen. Die Notwendigkeit dazu leitet sich vor allem aus dem Betrieb im Slave-Modus ab, bei dem der rückgewonnene Takt benutzt wird, um Daten zu senden. Da es sehr schwierig ist, in einem CMOS-Prozeß Oszillatoren mit sehr hoher Mittenfrequenz herzustellen, besteht der Wunsch nach Architekturen, die niedrigere Taktgeschwindigkeiten bei gleichbleibend hohen Datenraten voraussetzen.

Ein weitverbreiteter Ansatz ist der sogenannte „half-rate“-Ansatz nach [4], bei dem der Oszillator mit der halben Frequenz der Datenrate schwingt, bei 10 GBit/s folglich mit 5 GHz. In Literatur und Anwendung finden sich inzwischen zahlreiche Varianten dieses Ansatzes. Allen ist gemein, daß für den Betrieb sowohl die steigenden wie fallenden Flanken des Oszillator-Signals genutzt werden, um die empfangenen Daten abzutasten bzw. um die zu sendenden Daten zu multiplexen.

Während hierdurch die Frequenz des Oszillators halbiert und damit neben der vereinfachten Realisierung als CMOS-Chip auch die Leistungsaufnahme gesenkt werden kann, kommt nun als zusätzliche kritische Komponente die Asymmetrie der Halbtakte des Oszillatorsignales hinzu, „duty-cycle distortion“ genannt. Denn wenn aufgrund beliebiger

Gründe die beiden Halbtakte ungleich lang sind, werden jeweils zwei aufeinanderfolgende Bits im Sendesignal unterschiedlich lang ausfallen. Dies wäre gleichbedeutend mit einer erheblichen Beeinträchtigung des Sendesignals durch Jitter.

Auch der im Projekt bei der INFINEON TECHNOLOGIES AG verfolgte Ansatz reiht sich in die Gruppe der „half-rate“-Architekturen ein. Deshalb setzen die zum Multiplexer ab Seite 23 gemachten Ausführungen einen Oszillatortakt von 5 GHz voraus.

Bisher ist noch keine Sender/Empfängerarchitektur in der Literatur vorgestellt worden, die eine stärkere Herabsetzung des Oszillatortaktes als um einen Faktor 2 bei gleichbleibender Datenrate zu leisten vermag. Dabei fehlen insbesondere für das Senden entsprechende Lösungen. Hier wird indes eine Architektur vorgestellt, die theoretisch beliebige Faktoren der Form  $N = 2^n$  mit  $n \in \mathbb{N}$  zulässt, dabei aber die volle Funktionalität eines Transceiver-Chips zur Verfügung stellt. Sowohl das Senden als auch das Empfangen von Hochgeschwindigkeitsdaten in Slave- und Master-Modus sind mit einem niedrigen Oszillatortakt möglich.

### 2.3.2 Signalhub $\Delta U$

Das grundlegende Gesetz des Ladestromes  $i_c$  einer elektrischen Kapazität  $C$  zeigt die physikalischen Schranken auf, die dem Entwurf von Komponenten, die bei 5 GHz und höher funktionieren sollen, Grenzen setzen:

$$i_c = C \cdot \frac{dU}{dt} \approx C \cdot \frac{\Delta U}{\Delta t} \quad (2.1)$$

Im verwendeten CMOS-Prozeß liegen die parasitären Kapazitäten an einem Knoten zwischen zwei benachbarten Chipelementen im ungünstigsten Fall bei etwa 15 fF. Dieser Wert berücksichtigt neben den Diffusionskapazitäten auch Verdrahtungskapazitäten für eine minimale Strecke, beispielsweise zwischen zwei benachbarten Latches. Für eine Funktionsgeschwindigkeit von 5 GHz muß diese kapazitive Last in maximal einem Drittel einer Taktperiode umgeladen werden, um brauchbare Flankensteilheiten  $\frac{\Delta U}{\Delta t}$  zu erhalten und das Signal weiterverarbeiten zu können.

Setzt man die Randwerte in die angeführte Gleichung ein und setzt einen CMOS-typischen Signalhub gleich der Versorgungsspannung von 1,3 V voraus, so ergäbe sich ein Strom von ungefähr 3 mA! Da ein Gatter oder Latch zusätzlich noch parasitäre Kapazitäten an internen Knoten besitzt, für die dieselbe Gesetzmäßigkeit gilt, läge der benötigte Strom in jedem Falle deutlich über dem Akzeptablem.

Da sich für eine gegebene Fertigungstechnologie als einzige Gleichungsgröße der Signalhub  $\Delta U$  ändern läßt, muß er offensichtlich gesenkt werden. Die untere Grenze stellt dabei die Schwellspannung der CMOS-Transistoren dar, da eine Unterscheidung der logischen

Zustände Null und Eins nur dann möglich ist, wenn ein Transistor durch das Signal vom sperrenden in den leitenden Zustand und umgekehrt geschaltet werden kann.

Wird der Signalhub auf ca. 400 mV gesenkt, was der Schwellspannung plus einer Marge für die Störsicherheit entspricht, ergibt sich nur noch ein Ladestrom von etwa 1 mA. Dazu muß jedoch eine Schaltungsfamilie zum Einsatz kommen, die mit so geringen Spannungshüben zuverlässig funktioniert. Aus der Bipolarschaltungstechnik sind die Emitter-Coupled- und die eng verwandte Current-Mode-Logic bekannt (ECL- bzw. CML-Logik), die mit den für bipolare Schaltkreise typischen niedrigen Logikpegeln arbeiten [5].

Diese Logikfamilien sind differenziell aufgebaut, was ihre Störanfälligkeit erhöht. Eine weitere wichtige Eigenschaft dieser Logikfamilien ist, daß sie einen konstanten Querstrom aufweisen. Ist diese Eigenschaft bei großen Digitalschaltungen unerwünscht, birgt sie doch einen für die hier verfolgte Aufgabenstellung wichtigen Vorteil: Da aufgrund des konstanten Querstromes eines CML-Gatters der Substratstrom des Hochgeschwindigkeitsmoduls ebenfalls konstant ist, kommt es zu deutlich weniger Störeinkopplungen über das Chipsubstrat auf hinsichtlich der Jitter-Generation kritische Komponenten wie etwa den Oszillator.

Für schnelle Schaltkreise in CMOS-Technik wird deshalb allgemein auf das Prinzip der CML-Logik zurückgegriffen und diese mit CMOS-Transistoren umgesetzt. Auch im Projekt der INFINEON TECHNOLOGIES AG wurde eine der CML-Logik ähnelnde Schaltkreisfamilie eigens entwickelt. Diese Aufgabe, ohne die die hier vorgestellten Untersuchungen schon im Ansatz unmöglich gewesen wären, hat Herr Dipl.-Ing. VOLKMAR REBMANN ausgeführt. Leider unterliegen die Schaltungen dem Betriebsgeheimnis von INFINEON TECHNOLOGIES, weshalb hier keine Darstellung erfolgt.

Die für übergeordnete Untersuchungen maßgebliche Kenngröße eines Hochgeschwindigkeits-Flip-Flops ist die Leistungsaufnahme. Sie liegt bei rund 4 mA bei 1,3 V Versorgungsspannung, also etwa 5 mW, für eine vollständige Master/Slave-Anordnung der eingesetzten Variante der CML-Logik.

### 2.4 Zielsetzung

Ziel der vorliegenden Arbeit sind neue Ansätze für Sender und Empfänger eines Transceiver-Chips, die die im Abschnitt 2.1 ab Seite 13 definierten allgemeinen Wünsche an eine neue Architektur derartig erfüllen, daß eine spürbare Verbesserung gegenüber dem Stand der Technik erreicht wird. Für die Empfängerseite wird in Abschnitt 4.3 ab Seite 75 eine Betrachtung ausgewählter Beispiele aus der Literatur vorgenommen. Dabei werden deren zu übernehmenden Vor- und zu lösenden Nachteile identifiziert. Die Eignung der Entwicklung soll dabei an den im Abschnitt 2.2 definierten Vorgaben gemessen werden.

Das ultimative Ziel bei der Entwicklung einer Sender/Empfängerarchitektur muß sein, einen Ansatz zu präsentieren, der die höchste theoretisch für eine gegebene Fertigungstechnologie mögliche Datenrate erreicht. In dieser Arbeit wird ein Ansatz für eine Takt/Datenrückgewinnung vorgestellt, die dieses Ziel zu erreichen sucht. Im Text wird auf Seite 122 der neue Ansatz an diesem Ziel geprüft.

# Der Multiplexer

## 3 Ein asynchroner Multiplexer mit niedriger Leistungsaufnahme

### 3.1 Vorbetrachtungen

#### 3.1.1 Einleitung

Bei der Entwicklung eines Hochgeschwindigkeitsmultiplexers ist neben der immer geltenden Ausbeutevorgabe die wichtigste der in Kapitel 2 aufgeführten Randbedingungen die der geringen Leistungsaufnahme. Die Jitter-Generation ist hier nur bedingt einzubeziehen, da sie nahezu allein von der letzten Stufe, dem Ausgangstreiber, abhängt, in der der Sendetakt das Ausgangssignal erzeugt.

Um eine möglichst geringe Leistungsaufnahme zu erreichen, sollen zuerst die möglichen Architekturvarianten untersucht werden. Dabei wird als Maßstab die Zahl der benötigten Hochgeschwindigkeits-Flip-Flops herangezogen. Bei einer überschlägig angestrebten Leistungsaufnahme von 500 mW bei 1,3 V Betriebsspannung für den gesamten Hochgeschwindigkeitsteil und unter Berücksichtigung der auf Seite 21 erläuterten Leistungsaufnahme eines einzelnen Flip-Flops wird deutlich, daß jedes eingesparte Flip-Flop einen wahrnehmbaren Gewinn in der Leistungsbilanz darstellt. Vernachlässigt man sogar Funktionseinheiten mit hoher Leistungsaufnahme wie den Oszillator oder den Aufwand für einen 5 GHz Taktbaum, wären trotzdem nur maximal 100 Flip-Flops im beschränkten Leistungsbudget des Hochgeschwindigkeitsteiles möglich.

#### 3.1.2 Das Schieberegister

Abbildung 3.1 zeigt ein Blockdiagramm eines als Schieberegister ausgeführten Multiplexers. Ein synchroner Taktteiler bestimmt, wann die Flip-Flops der Kette geladen werden. Die geladenen Werte werden durch die Kette nach rechts heraus „geschoben“ und so gemultiplext. Da alle Flip-Flops in dieser Anordnung bei vollem Hochgeschwindigkeitstakt betrieben werden, ergibt sich für die Schieberegisteranordnung ein Aufwand von:

$$A = (N - 1) + (\log_2 N + S) \quad (3.1)$$



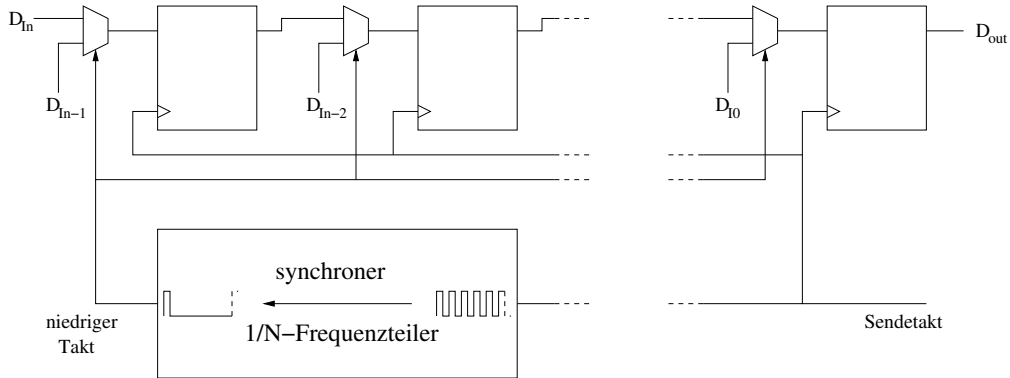


Abbildung 3.1: Blockdiagramm eines als Schieberegister ausgeführten Multiplexers

Der erste Term beschreibt die Anzahl der benötigten schnellen Flip-Flops für die Registerkette, der zweite den Aufwand für den Taktteiler.  $N$  ist dabei das Multiplexverhältnis,  $S$  bedeutet einen nicht näher spezifizierten Aufschlag für die Notwendigkeit eines synchronen Zählers. Ein synchroner Zähler benötigt prinzipiell zusätzlichen Aufwand für ebenfalls schnelle und somit leistungsintensive Logikgatter.

Für das hier angestrebte Multiplexverhältnis von 1:16, also von 622 MBit/s auf 10 GBit/s, würde sich ein Wert von etwa  $15 + (4 + 1)$  ergeben. Für diese verhältnismäßige einfache Funktionseinheit würden also schon 20% des Leistungsbudgets des gesamten Hochgeschwindigkeitsteiles verloren gehen! Das Schieberegister stellt zwar einen typischen Ansatz für langsame Digitalerschaltungen dar, doch verbietet es sich hier aufgrund seiner immensen Leistungsaufnahme.

### 3.1.3 Die binäre Baumstruktur

Verkürzt man das Schieberegister auf ein einziges Flip-Flop, so ergibt sich ein 2:1-Multiplexer, der sich in einer binären Baumstruktur gemäß Abbildung 3.2 anordnen lässt. In jeder Ebene des Baumes wird die Datenrate verdoppelt, bis die Senderate erreicht ist. Die Anzahl der Ebenen ist  $\log_2 N$ ; in jeder Ebene werden  $2^{i-1}$  2:1-Multiplexer benötigt, wobei  $i$  die Ebene im Baum angibt. Der benötigte Aufwand berechnet sich also nach folgender Formel:

$$A = \sum_{i=1}^{\log_2 N} 2^{i-1} + (\log_2 N + S) \quad (3.2)$$

Der erste Term bedeutet wiederum den Aufwand für den eigentlichen Multiplexer, der

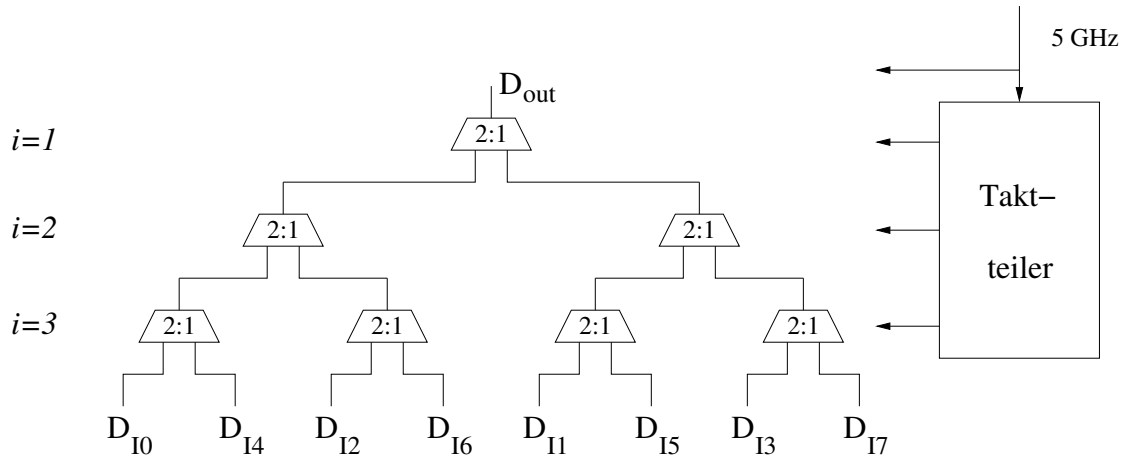


Abbildung 3.2: Blockdiagramm eines als Binärbaum ausgeführten Multiplexers

zweite den für den synchronen Taktteiler. Bei einem Multiplexverhältnis von  $N = 16$  ergibt sich ein Wert von  $A = (2^3 + 2^2 + 2^1 + 1) + (4 + 1) = 20$ . Es folgt das gleiche Ergebnis wie für das Schieberegister nach Formel 3.1.

Betrachtet man den Term  $\sum_{i=1}^{\log_2 N} 2^{i-1}$  der Formel 3.2 genauer, so erkennt man, daß er sich über folgende Umformungsschritte vereinfachen läßt:

$$\sum_{i=1}^{\log_2 N} 2^{i-1} = \sum_{i=0}^{\log_2 N - 1} 2^i = 2^{(\log_2 N - 1) + 1} - 1 = 2^{\log_2 N} - 1 = N - 1 \quad (3.3)$$

Gleichungen 3.1 und 3.2 sind identisch! Das bedeutet, daß beide Multiplexerarchitekturen den gleichen Aufwand an Flip-Flops aufweisen und somit beide mit Hinsicht auf ihre Leistungsaufnahme ungeeignet sind. Dies ist jedoch nur scheinbar richtig, denn ein wichtiger Punkt wurde bei obenstehender Betrachtung nicht berücksichtigt: jede Ebene arbeitet nur mit der halben Geschwindigkeit der ihr übergeordneten, weshalb jeweils auch die doppelte Zeit  $\Delta t$  für das Umladen der logischen Zustände zur Verfügung steht. Nach Formel 2.1 von Seite 20 wird deshalb auch nur der halbe Ladestrom  $i_c$  benötigt, woraus wiederum folgt, daß stromsparendere Varianten von Flip-Flops eingesetzt werden können.

Weil jede Ebene *halb* so schnell arbeitet wie die ihr übergeordnete, aber *doppelt* so viele 2:1-Multiplexer enthält, ergibt sich für alle Ebenen der gleiche Strombedarf. Da die höchste Ebene aus nur einem Flip-Flop besteht, das bei der höchsten Geschwindigkeit arbeitet, berechnet sich der Aufwand für den Binärbaum richtig:

$$A = \log_2 N + (\log_2 N + S) \quad (3.4)$$

$\log_2 N$  ist wieder die Zahl der Ebenen des Binärbaumes, der zweite Term der Aufwand für den synchronen Taktteiler. Für  $N = 16$  ergibt sich somit nur noch ein Aufwand von  $A = 4 + (4 + 1) = 9$ . Der Vergleich der Formeln 3.1 und 3.4 ergibt: für alle  $N \geq 2$  ist  $N - 1 \geq \log_2 N$ . Da nur Multiplexverhältnisse  $\geq 2$  sinnvoll sind, ist die binäre Baumstruktur der Schieberegisteranordnung unter dem Gesichtspunkt der Leistungsaufnahme in jedem Fall überlegen.

Ein Nachteil der Baumstruktur liegt jedoch in der Tatsache, daß der Taktteiler entsprechende Takte für jede Ebene des Baumes zur Verfügung stellen muß, also die Ausgänge aller Flip-Flops des als Zähler aufgebauten Taktteilers nach außen geführt werden müssen.

### 3.1.4 Vergleich synchroner Teiler – asynchroner Teiler

Für den Leistungsverbrauch eines synchronen Teilers gilt Vergleichbares wie für das Schieberegister: alle  $\log_2 N$  Flip-Flops des Zählers (sowie einige Logikgatter) müssen bei voller Geschwindigkeit arbeiten können. Der asynchrone Teiler dagegen besteht aus einer Kette von wiederum  $\log_2 N$  als  $\div 2$ -Teiler geschalteten Flip-Flops. Er bietet, analog zum Multiplexer nach Binärbaumstruktur, den Vorteil, daß der Stromverbrauch jedes Flip-Flops entsprechend seiner Arbeitsgeschwindigkeit angepaßt werden kann. Außerdem fällt der Aufwand für Logikgatter weg.

Der Leistungsverbrauch eines asynchronen Teilers kann nach folgender Formel abgeschätzt werden:

$$A = \sum_{i=1}^{\log_2 N} \frac{1}{i} = 2 - \frac{1}{\log_2 N} \approx 2 \quad (3.5)$$

Die für einen Multiplexer benötigte Leistung kann durch den Einsatz eines asynchronen Teilers weiter gesenkt werden. Für das angenommene Multiplexverhältnis von  $N = 16$  sinkt der Aufwand für den Teiler von 5 auf 2. Für eine Baumstruktur mit asynchronem Taktteiler sinkt der Gesamtaufwand folglich auf insgesamt 6 Hochgeschwindigkeits-Flip-Flops, was gegenüber dem ersten Ansatz eines Schieberegisters, der den Einsatz eines asynchronen Teilers nicht zuläßt, eine Verbesserung um 70% bedeutet. Der Anteil des Multiplexers am gesamten Leistungsbudget des Hochgeschwindigkeitsmodules ist auf ungefähr 6% geschrumpft.

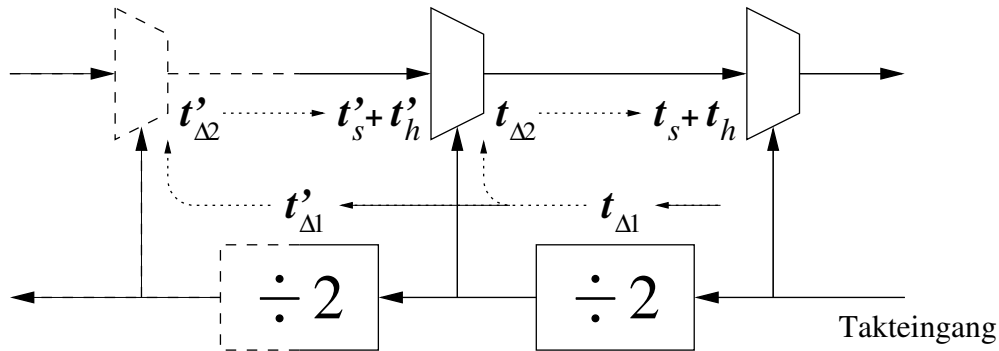


Abbildung 3.3: Timing im Multiplexer bei Einsatz eines asynchronen Teilers

Gleichwohl hat der Einsatz eines asynchronen Teilers gravierende Nachteile, die es nach Möglichkeit zu kompensieren gilt. Analysiert man die zeitlichen Gegebenheiten im Multiplexer, so erschwert der asynchrone Teiler den Entwurf erheblich, da durch seinen Einsatz die zur Verfügung stehenden Zykluszeiten weiter sinken. Abbildung 3.3 dient zur Verdeutlichung dieses Sachverhaltes.

In der Abbildung ist im unteren Teil ein asynchroner Teiler als Kette von  $\div 2$ -Teilern dargestellt, im oberen eine Reihe von 2:1-Multiplexern symbolisch angedeutet. Für zwei Maschen wurden die zwischen den Signalen entstehenden Verzögerung eingetragen.

Es ist zu beachten, daß der Signalfluß durch die Multiplexer von links nach rechts verläuft, der durch die Teilerkette jedoch in Gegenrichtung. Dadurch summieren sich alle anfallenden Verzögerungen jeder Masche. Der zeitliche Bezugspunkt für die Betrachtung ist das Taktsignal am Eingang des asynchronen Teilers. Es ergibt sich  $t_{ges} = t_{\Delta 1} + t_{\Delta 2} + t_s + t_h$ . Dabei bedeutet  $t_{\Delta 1}$  die Verzögerung, die zwischen dem Auftreten der steigenden Taktflanke am Takteingang des  $\div 2$ -Teilers und seinem Ausgang auftritt („Clock-to-Q“).  $t_{\Delta 2}$  ist die Clock-to-Q-Verzögerung im 2:1-Multiplexer. Hinzu kommen schließlich noch die Setup- und Hold-Zeiten der folgenden Multiplexer-Stufe.

Die Aufrechnung fällt für einen synchronen Teiler zunächst identisch aus, ändert sich jedoch bei der nächsten Ebene des Multiplexer-Baumes mit asynchronem Teiler. Dort ist das zuvor schon einmal heruntergeteilte Signal bereits mit einer Verzögerungszeit  $t_{\Delta 1}$  belastet, so daß zur Summe der Verzögerungszeiten für diese Schleife ein weiterer Term tritt. Dies setzt sich für die weiter unten liegenden Ebenen entsprechend fort, so daß auch in den langsameren Multiplexerteilen der Entwurf schwierig sein kann, zumal dort ja auf langsame aber stromsparende Full-swing-Flip-Flops zurückgegriffen werden soll.

Beim Entwurf sehr schneller Schaltungen ist es meist unvermeidbar, die an einzelnen Punkten entstehenden Verzögerungen in anderen Signalwegen nachzubilden, so daß sich

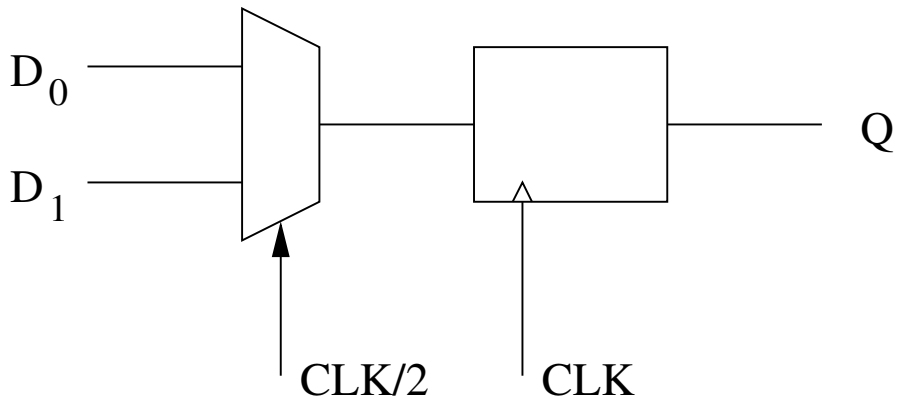


Abbildung 3.4: Eine typische 2:1-Multiplexeranordnung

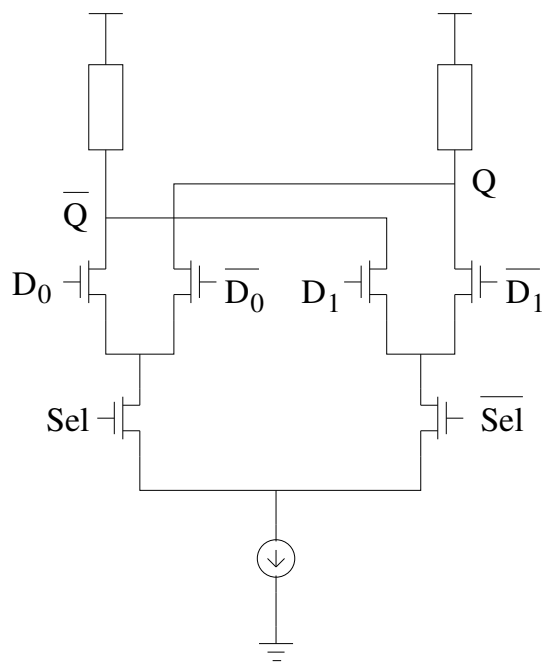
beide Verzögerungen gegenseitig kompensieren. Bei Taktfrequenzen, wie sie in dieser Arbeit auftreten, ist jedoch auch dieses Vorgehen äußerst schwierig, denn nicht nur die Durchlaufzeiten einzelner Funktionselemente liegen bereits in der Größenordnung der Taktdauer, sondern auch ihre Schwankung aufgrund der verschiedenen Fertigungs- und Betriebsparameter.

So weist z.B. das eingesetzte CML-Flip-Flop eine typische „Clock-to-Q“-Zeit von etwa 25 ps auf. Diese Zeit sinkt jedoch bei schnellstmöglicher Parameterkombination zusammen mit niedriger Betriebstemperatur und hoher Betriebsspannung auf ca. 15 ps, steigt hingegen beim langsamsten Parametersatz, hoher Temperatur und niedriger Betriebsspannung auf 40 ps. Die Differenz zwischen den einzelnen Zeiten ist mit 25 ps gleich groß wie die typische Durchlaufzeit!

Ähnliches gilt auch für Takttreiber, Logikgatter und andere Elemente in CML-Schaltungstechnik. Die hohe Schwankungsbreite der Verzögerungszeiten erschwert es besonders, Verzögerungen zu kompensieren, wenn sie in Schaltungen verschiedener Schaltungstechnik entstehen. So weist ein CMOS-Inverter ein völlig anderes Verhältnis zwischen maximaler und minimaler Verzögerungszeit auf, weshalb die Verzögerungen zwischen einem CML- und einem full-swing-CMOS-Signal niemals für alle Parameterkombinationen ausgeglichen werden kann. Damit wird insbesondere die Schnittstelle zwischen dem Teil des Multiplexers, der in klassischer CMOS-Schaltungstechnik verwirklicht ist, und dem CML-Teil schwierig.

### 3.1.5 Typische 2:1-Multiplexerschaltungen

Abbildung 3.4 zeigt einen typischen Ansatz für einen 2:1-Multiplexergrundblock, der in einer Baumstruktur angeordnet werden könnte. Seine Elemente sind ein einfacher 2:1-



**Abbildung 3.5:** Eine Multiplexerschaltung in CMOS-CML-Schaltungstechnik

Multiplexer mit Select-Eingang, sowie ein Flip-Flop. Am Select-Eingang des Multiplexers liegt das halbierte Taktsignal an und schaltet abwechselnd einen der beiden Dateneingänge auf den Ausgang durch. Anschließend werden die nun doppelt so schnell anliegenden Daten mit der vollen Taktrate in ein Flip-Flop eingelesen.

Abbildung 3.5 zeigt eine Multiplexerschaltung in einer mit CMOS-Transistoren umgesetzten CML-Schaltungstechnik. Diese Anordnung weist jedoch den Nachteil auf, daß das Signal am ersten Dateneingang  $D_0$  bereits früh anliegen muß, um die Setup-Zeit für das abschließende Flip-Flop einzuhalten; der andere Dateneingang wird dagegen erst im folgenden Halbtakt des Select-Taktes durch den Multiplexer durchgeschaltet und muß solange stabil anliegen, bis das Flip-Flop auch den zweiten Wert sicher eingelesen hat. Für die gesamte Anordnung gelten deshalb die zeitlichen Anforderungen des schnellen Taktes.

[6] stellt eine 2:1-Multiplexer-Anordnung vor, die diesen Nachteil entschärft, indem in den Pfad des zweiten Dateneinganges ein Latch eingefügt wird, das den anliegenden Wert einliest und während des zweiten Halbtaktes zwischenspeichert. Außerdem wird dort die Anordnung, die sich in der Baumstruktur regelmäßig wiederholt, anders aufgeteilt, so daß einzelne Multiplexergrundzellen entstehen, die nur einen Takt benutzen. Abbildung 3.6 gibt die Anordnung der zitierten Quelle sinngemäß wieder.

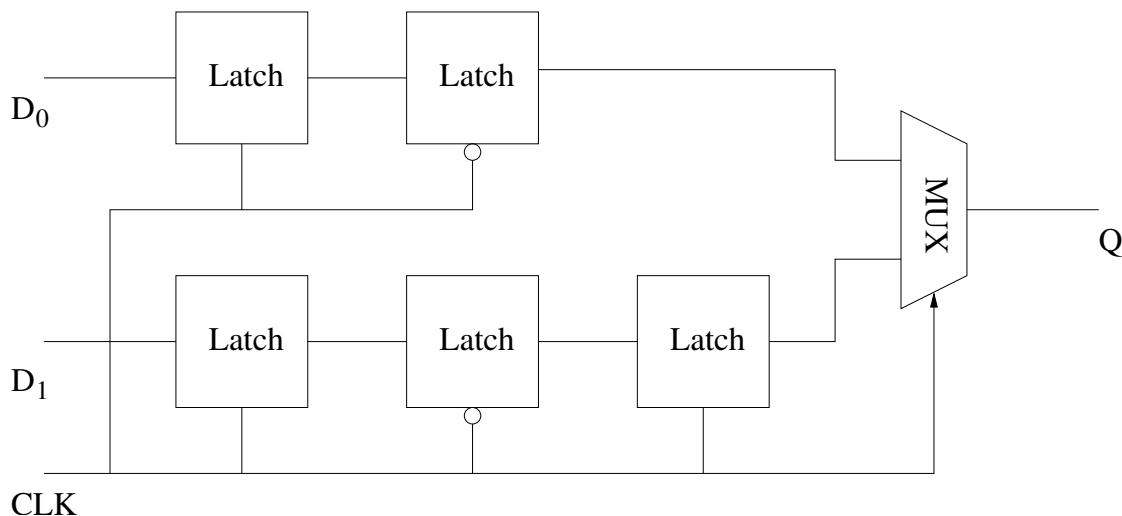


Abbildung 3.6: Blockdiagramm des Multiplexers nach [6]

Obwohl die Quelle noch nicht zur Verfügung stand, als die im folgenden Abschnitt vorgestellte Schaltung entwickelt wurde, sind die beiden Lösungsansätze schaltungstechnisch ähnlich, weisen aber unterschiedliche Vor- und Nachteile auf. Beim hier bevorzugten Ansatz überwogen die in Abschnitt 3.2.3 auf Seite 37 erläuterten Gründe.

## 3.2 Die neue Multiplexeranordnung

### 3.2.1 Die neue 2:1-Multiplexer-Grundzelle

Abbildung 3.7 zeigt die neue 2:1-Multiplexer-Grundzelle. Der Unterschied zur in der vorhergehenden Abbildung dargestellten Schaltung besteht in dem verwendeten Exklusiv-Oder-Gatter anstelle eines Multiplexers mit Select-Eingang.

Betrachtet man die Struktur eines XOR-Gatters in CML-Schaltungstechnik, so erkennt man, daß sie mit der eines 2:1-Multiplexers identisch ist. Lediglich die Ein- und Ausgänge werden gemäß Tabelle 3.1 umbenannt. Der Ausgang des letzten Latches vor dem Eingang  $B/\bar{B}$  des XOR-Gatters wird nun doppelt belastet, was aber nicht schwer wiegt, denn zum einen ist die Belastung der Ausgänge der Latches gering im Vergleich zu der des Taktbaumes, zum anderen kann man die Belastung beider Datenwege symmetrisieren. Dies geschieht, indem man die XOR-Struktur in halber Stärke doppelt auslegt, die Eingänge der zweiten Struktur vertauscht und die Ausgänge beider kurzschließt.

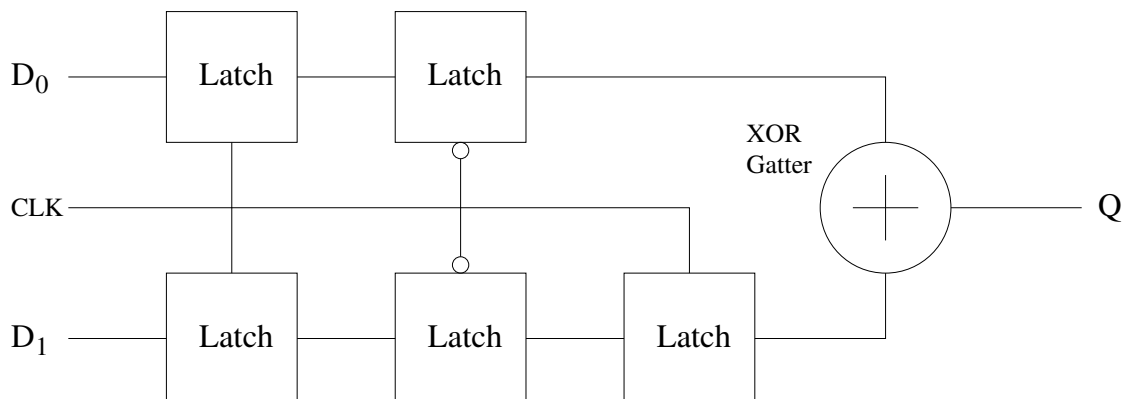


Abbildung 3.7: Schaltung der Multiplexer-Grundzelle

MUX	XOR
A	B
$\overline{A}$	$\overline{B}$
B	B
$\overline{B}$	$\overline{B}$
Sel	A
$\overline{\text{Sel}}$	$\overline{A}$

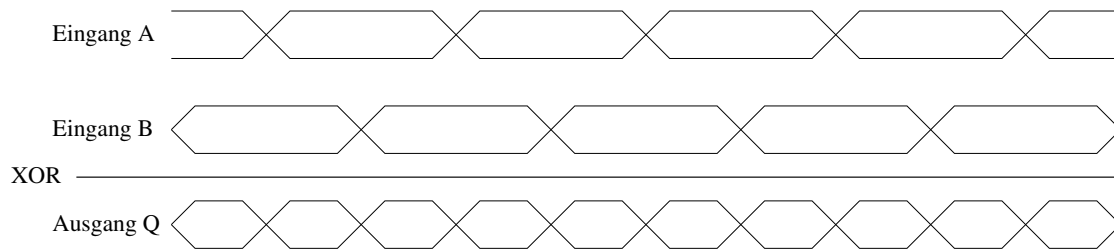
Tabelle 3.1: Umbenennung der Ein- und Ausgänge des Multiplexers

Die Eignung der vorgestellten Schaltung als Multiplexer soll zuerst bildlich veranschaulicht werden. Abbildung 3.8 illustriert hierzu einen wichtigen Aspekt des XOR-Gatters. Sie zeigt, daß mit dem Wechsel genau eines der beiden Eingangssignale zwingend ein Wechsel des Ausgangssignales assoziiert ist. Es ist somit möglich, aus zwei um eine halbe Bitlänge gegeneinander verschobenen Eingangssignalen gegebener Datenrate vermittels eines XOR-Gatters ein Ausgangssignal der doppelten Datenrate zu erzeugen.

Anstatt der XOR-Struktur hätte hier auch die Äquivalenz-Funktion zum Einsatz kommen können, da auch sie über die wichtige Eigenschaft verfügt, daß sich aus der Kenntnis des Ausgangswertes und eines Eingangswertes für jede Kombination von Eingangswerten auf den zweiten Eingangswert rückschließen läßt, also die logische Operation umkehrbar ist. Diese Bedingung muß erfüllt sein, damit sich die logische Verknüpfung der Daten an anderer Stelle wieder aufheben läßt.

Da beim Entwurf des Systems nicht die Freiheit besteht, die Kodierung der Daten auf der Übertragungsstrecke zu bestimmen, müssen die Daten trotz der logischen Verknüpfung durch das XOR-Gatter unmodifiziert gesendet werden. Deshalb muß ein Weg gefunden





**Abbildung 3.8:** Logische Verknüpfung der beiden Datensignale in der Multiplexerzelle

werden, sie senderseitig so vorzukodieren, daß die logische Verknüpfung der einzelnen Datenströme im Multiplexer an seinem Ausgang genau wieder die zu sendenden Daten ergibt.

Der Ausgangswert des Multiplexers kann frei bestimmt werden und hängt im Falle des 2:1-Multiplexers von der logischen Kombination zweier aufeinanderfolgender Bits der beiden Eingangsdatenströme ab. Jedes Bit der Eingangsdatenströme wirkt sich dabei auf den Wert von zwei Ausgangsbits aus, so daß die Vorkodierung eine zeitliche Komponente berücksichtigen muß.

Noch komplizierter werden diese Zusammenhänge bei Betrachtung einer Baumstruktur wie in Abbildung 3.9. Darin sind für jede Multiplexer-Grundzelle die für einen Datenstrom im Verhältnis zum anderen durch das dritte Latch entstehenden Verzögerungen eingetragen. Dabei muß beachtet werden, daß in jeder Ebene die Verzögerungen doppelt so lang sind wie die der ihr übergeordneten Ebene. Verfolgt man die einzelnen Wege von den Eingängen zum Ausgang der Schaltung und summiert dabei die anfallenden Verzögerungen als Vielfache der Ausgangsbitlänge, dann ergibt sich für jeden Weg durch den Baum ein anderer Wert. Diese Werte sind in der Abbildung an den entsprechenden Eingängen vermerkt; es ergeben sich alle Werte zwischen 0 und  $N - 1$ .

Da die Baumstruktur als  $N$ -fach Exklusiv-Oder angesehen werden kann, beeinflusst jeder Eingangswert einen Satz von  $N$  aufeinanderfolgenden Ausgangswerten. Die vom Baum durchgeführte Operation kann als kontinuierliche Paritäts-Berechnung von jeweils  $N$  Eingangswerten verstanden werden. Die Verknüpfung einer zeitlichen und einer logischen Ebene läßt die Lösung des Problems der Vorkodierung sehr kompliziert erscheinen. Bei einer anderen Betrachtungsweise zeigt sich jedoch das Gegenteil, wie im folgenden Abschnitt dargelegt.

### 3.2.2 Die Vorkodierung der Daten

Jedes NRZ-Signal, also jede einfache Folge von gleichlangen logischen Einsen und Nullen, kann auch als Folge von steigenden und fallenden Flanken angesehen werden. Abbil-

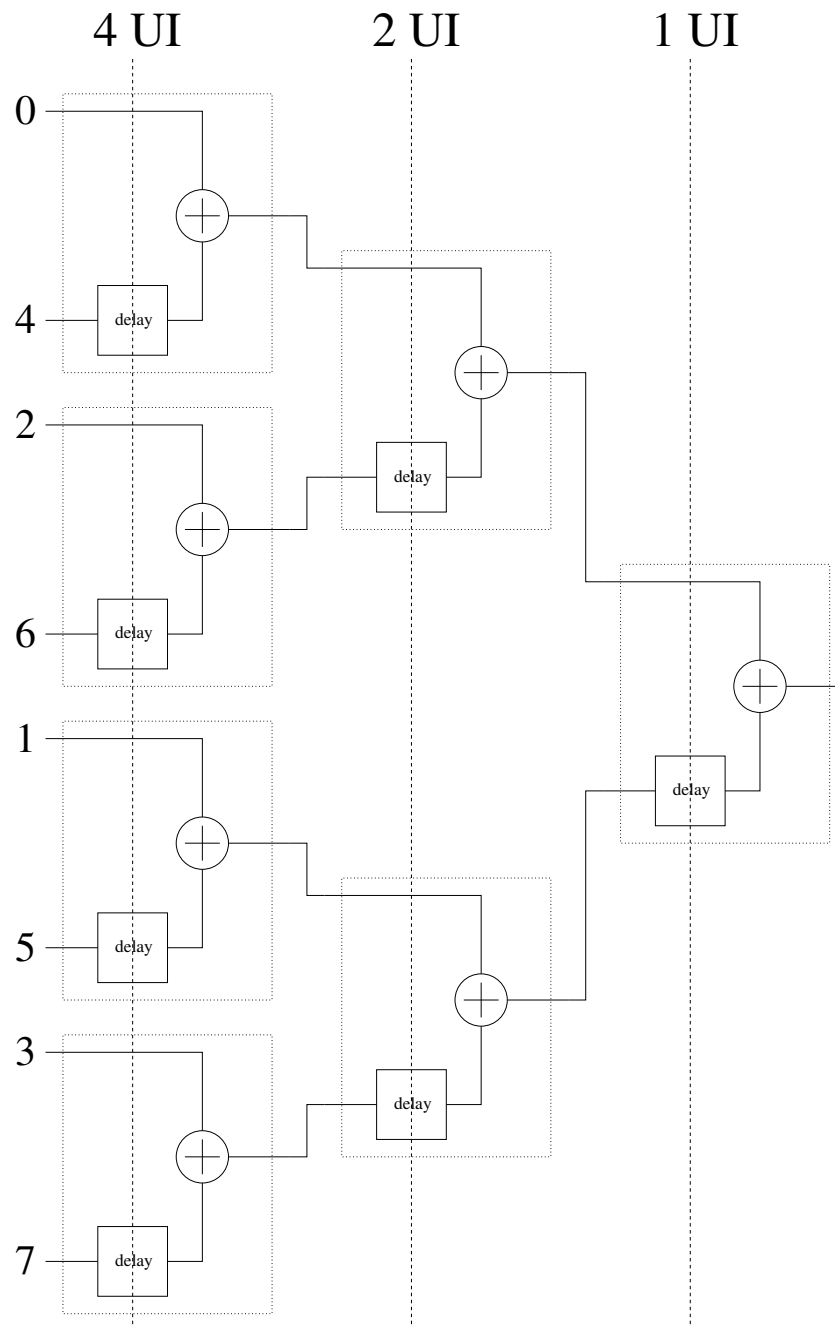
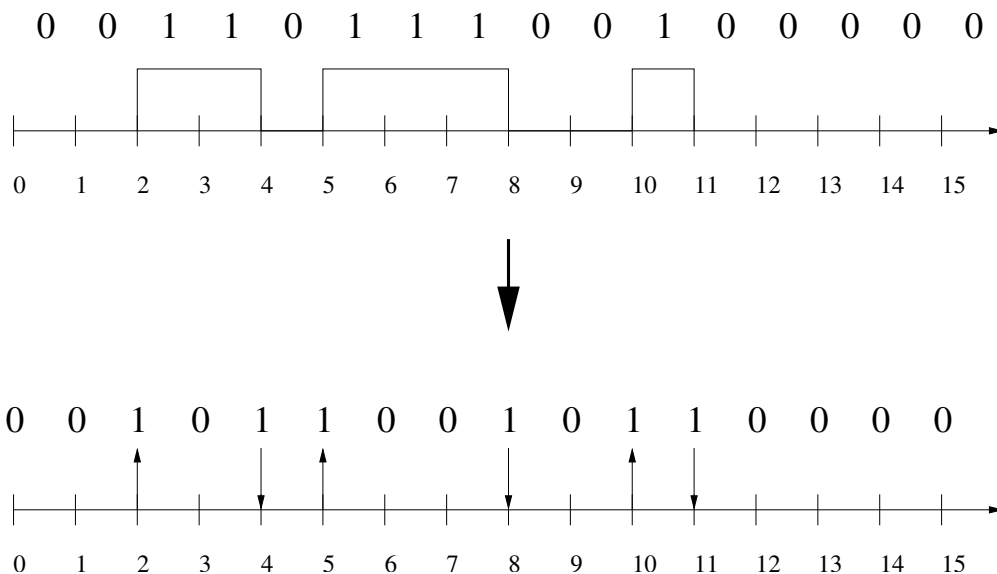


Abbildung 3.9: Die Multiplexerzellen zu einem binären Baum angeordnet



**Abbildung 3.10:** Umkodierung eines pegelkodierten Signales in ein flankenkodiertes

Abbildung 3.10 zeigt die Umkodierung eines NRZ-Signales in eine Repräsentation derselben Daten als eine Folge von Flanken. Dabei bedeutet jede Eins einen Wechsel von Eins nach Null oder umgekehrt im pegelkodierten Signal, jede Null zwei aufeinander folgende gleiche Werte. Die Umkodierung vom pegel- zum flankenkodierten Signal wird durch Exklusiv-Oder-Verknüpfung jeweils benachbarter Werte erreicht.

Das flankenkodierte Signal gibt Aufschluß darüber, zu welchen Zeitpunkten Wechsel im Ausgangssignal vorliegen bzw. erzeugt werden müssen. Die wichtigste Folgerung aus den Betrachtungen zu den Abbildungen 3.8 und 3.9 war, daß ein Wechsel an einem Eingang des Multiplexerbaumes einen Wechsel des Ausgangssignales zu einem bestimmten, dem Eingang zugeordneten Zeitpunkt bewirkt. Demzufolge ist jedem Eingang des Multiplexers ein bestimmter Zeitpunkt im Ausgangssignal zugeordnet. Soll zu diesem Zeitpunkt eine Flanke erzeugt werden, muß lediglich der entsprechende Eingang des Multiplexers invertiert werden.

Aus dieser einfachen Feststellung heraus läßt sich die Schaltung des Vorkodierers ableiten. Abbildung 3.11 zeigt das Blockdiagramm. Ein Satz von  $N$  Eingangsbits wird von der Pegel- in die Flankenkodierung überführt. Hierzu muß das letzte Bit des vorhergehenden Datensatzes zwischengespeichert werden. An jedem Eingang des Multiplexerbaumes ist jeweils ein sog. Toggle-Flip-Flop vorgesehen. Ergibt der umkodierte Datensatz eine Flanke, dann bewirkt die resultierende Eins am Eingang des zugeordneten Toggle-Flip-Flops eine Invertierung seines Wertes. Soll hingegen keine Flanke erzeugt werden – es liegt eine

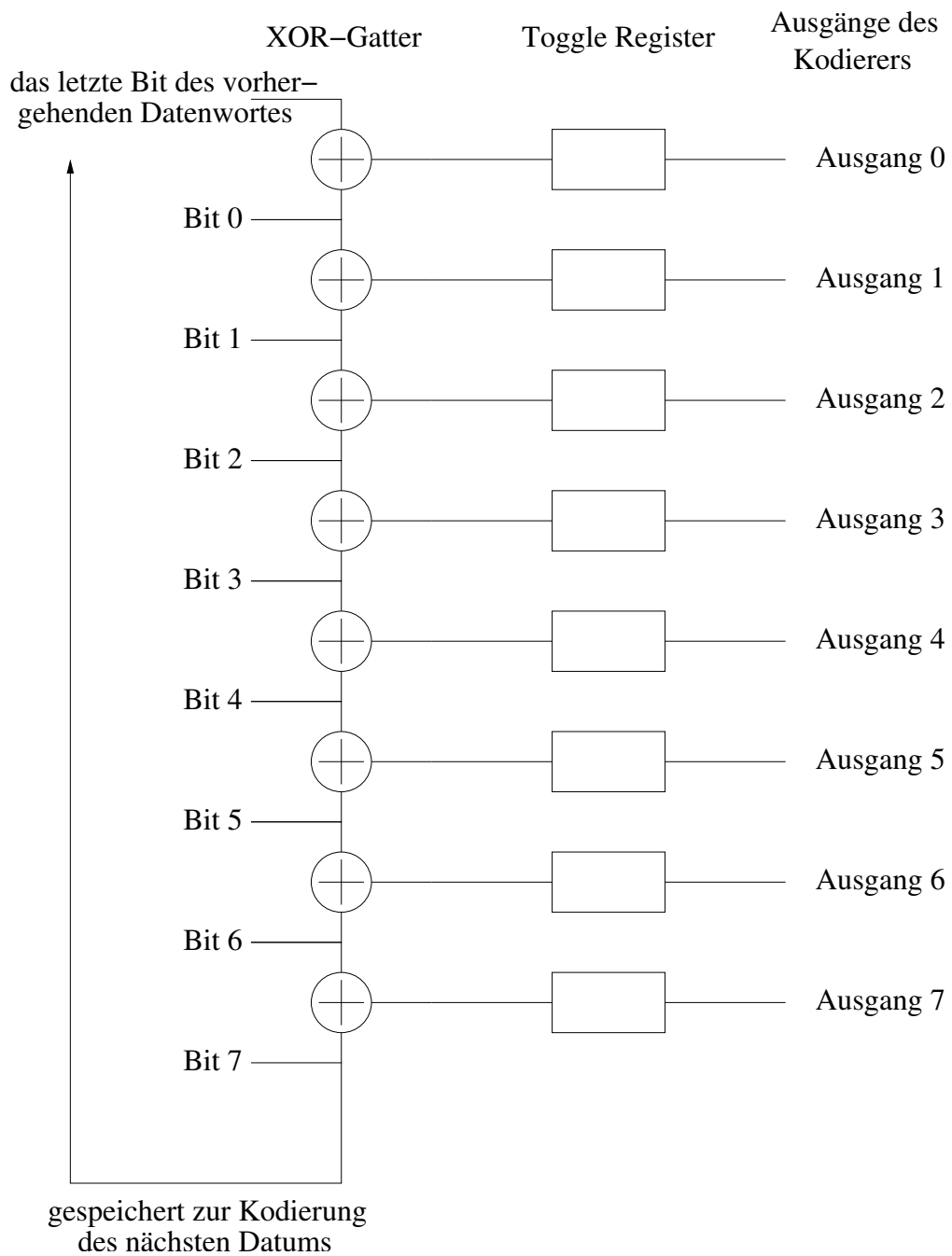


Abbildung 3.11: Prinzipschaltbild des Vorkodierers

Null am Eingang des Toggle-Flip-Flops an – speichert es unverändert seinen vorherigen Wert.

Es zeigt sich, daß die Vorkodierschaltung mit geringem schaltungstechnischem Aufwand umgesetzt werden kann. Der Vorkodierer arbeitet bei der niedrigen Geschwindigkeit der parallelen Eingangsdaten und läßt sich ohne weiteres in einfacher und stromsparender CMOS-Logik aufbauen. Dieser zusätzliche Aufwand für die neue Multiplexer-Anordnung ist deshalb vernachlässigbar.

### 3.2.3 Vor- und Nachteile des neuen Multiplexers

Die neue Multiplexeranordnung zeigt im Vergleich zu der in [6] vorgestellten Lösung Vor- und Nachteile. Der Schaltungsentwickler muß nach den Anforderungen seines Entwicklungszieles entscheiden, welche Eigenschaften der Schaltungen für ihn ausschlaggebend sind. Die Gesichtspunkte, die zur Entwicklung des neuen Multiplexers geführt haben, werden im Folgenden erläutert.

Die Belastung der Taktleitung durch einen Multiplexer gemäß Abbildung 3.5 ist gleich groß wie die durch ein CML-Latch verursachte. Da höhere Belastungen der Taktleitungen einen größeren Aufwand bei der Verteilung der Takte und, nicht zuletzt, einen größeren Leistungsverbrauch bedingen, senkt der neue Multiplexer die kapazitive Belastung der Taktleitungen um ein Sechstel allein durch die wegfallenden Gate-Kapazitäten des CML-Multiplexers.

Einzubeziehen ist jedoch, daß die Funktion sowohl des XOR-Gatters als auch des Multiplexers in der Schaltung nach Abbildung 3.6 in das erste Latch der folgenden Stufe integriert wird, was in der CML-Logik ohne besonderen Aufwand möglich ist. Je nach den Gegebenheiten des Layouts muß deshalb in der Variante nach [6] die Taktleitung bis zur nächsten Stufe des Baumes weitergeführt werden, was eine überproportionale Steigerung ihrer kapazitiven Belastung durch die Verdrahtungskapazitäten bedeuten kann. Tatsächlich zeigte sich für die verwendete 120 nm-Technologie, daß Verdrahtungskapazitäten den überwiegenden Anteil der Belastung des Taktbaumes ausmachen. Der Einsatz des XOR-Gatters senkt deshalb die Belastung des Taktbaumes um mehr als das rechnerische Sechstel.

Ein weiterer Vorteil der neuen Anordnung ist, daß es an keinem Punkt notwendig ist, zwei Takte gleichzeitig einzusetzen, wie das bei Nutzung eines CML-Multiplexers der Fall ist. Dort werden sowohl das Taktsignal für das erste Latch jedes Datenweges als auch das heruntergeteilte Select-Signal für den in das Latch integrierten Multiplexer benötigt. Wegen der asynchronen Funktionsweise der gesamten Schaltung kann es schwieriger sein, vier Signale (Daten, Select-Signal und Takt) zeitlich zu koordinieren, als nur drei (Daten und Takt).

Der einzige Nachteil der neuen 2:1-Multiplexer-Grundzelle liegt im etwas schlechteren Timing. Bei der Schaltung in Abbildung 3.6 auf Seite 31 schaltet das Taktsignal den Multiplexer als letztes Element und bestimmt somit an diesem Punkt die zeitliche Koordination des Ausgangssignales mit der nächsten Stufe. Bei der hier vorgestellten Lösung geschieht dies in den jeweils letzten Latches der beiden Datenpfade, die darauf durch die Durchlaufzeit des Exklusiv-Oder-Gatters anfallende Verzögerung muß von der zur Verfügung stehenden Zykluszeit abgezogen werden.

Dieser Nachteil wird jedoch abgemildert. Beim Ansatz nach [6] schaltet das lokale Taktsignal, das kapazitiv bereits erheblich belastet ist und folglich flache Flankensteigungen aufweist, die Datensignale auf den Multiplexerausgang durch. Wegen der schwachen Flankensteigung wird der entsprechende Transistor erst später voll durchgeschaltet, was das Erscheinen des Eingangsdatums auf dem Ausgang verzögert.

Beim Exklusiv-Oder-Gatter hingegen wird der Wechsel des Ausgangswertes direkt durch den Wechsel eines Eingangswertes ausgelöst. Da die kapazitive Last auf den Ausgängen der vorangehenden Latches niedrig ist, werden die Eingänge des Gatters entsprechend früher umgeladen und voll durchgeschaltet, was die Durchlaufzeit durch das XOR-Gatter um ca. 10 ps verkürzt.

#### 3.2.4 Die Funktionsweise mit herabgesetztem Oszillatortakt

Der größte Vorteil dieser neuen Multiplexerarchitektur, den keine andere bekannte Anordnung gestattet, ist der eines herabgesetzten Oszillatortaktes bei gleichbleibend hoher Ausgangsdatenrate. Da der Multiplexer innerhalb eines Projektes entwickelt wurde, das aufgrund anderer Anforderungen einen 5 GHz-Oszillator voraussetzte, wurde seine schaltungstechnische Umsetzung, die in Abschnitt 3.4 beschrieben wird, darauf abgestimmt. D.h., er wurde als „half-rate“-Architektur ausgelegt. Grundsätzlich kann der neue Ansatz jedoch auch mit niedrigeren Oszillatortakten funktionieren.

Man kann die Elemente der einzelnen Zweige des Multiplexerbaumes der Abbildung 3.9 auf Seite 34 auch anders anordnen. In Abbildung 3.12 wurden für die einzelnen Wege von den Eingängen zu den Ausgängen die entstehenden Verzögerungen summiert und in jeweils einem Block am Eingang zusammengefaßt. Für den ersten Eingang ergibt sich keine Verzögerung, die weiteren Eingänge verzögern das zugeordnete Eingangssignal um jeweils einen Teil einer Taktperiode des niedrigen Eingangstaktes. Der Eingang, der beispielsweise die 5. Position einer möglichen Flanke im Ausgangssignal kontrolliert, weist eine Verzögerung von  $\frac{5}{8}$  UI auf, der für die 6. Flankenposition eine von  $\frac{6}{8}$  UI usw. Abbildung 3.13 verdeutlicht, wie schon Abb. 3.8 für eine 2:1-Multiplexerzelle, noch einmal die Verknüpfung der acht Eingangssignale zu einem Ausgangssignal achtfacher Datenrate.

Durch die Umordnung wird der Multiplexerbaum zu einem in drei Stufen ausgeführten achtfachen Exklusiv-Oder. Da alle Eingänge eines Exklusiv-Oder-Gatters gleichwertig

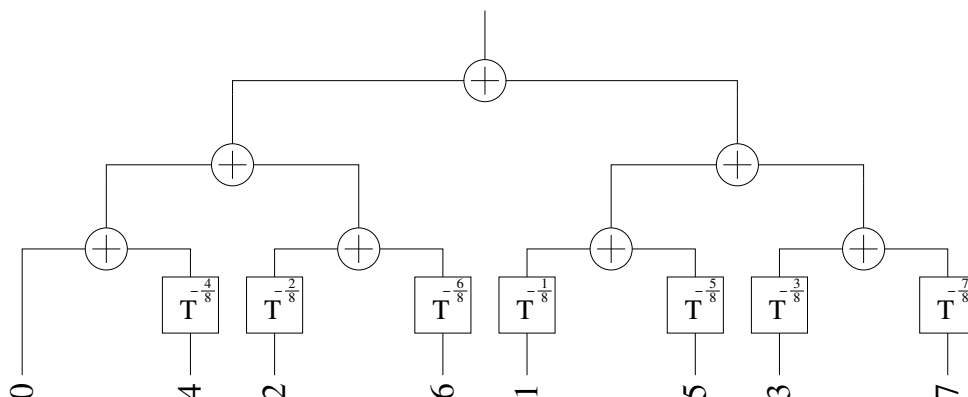


Abbildung 3.12: Umgeordneter Multiplexer-Baum nach Abb. 3.9

sind, könnten die Verzögerungen an den Eingängen beliebig angeordnet werden. Die Anordnung aus Abbildung 3.9 wurde aber übernommen, ist sie doch implementationstechnisch die günstigste, weil in jedem XOR-Gatter zwei Daten verknüpft werden, die um eine halbe Bitlänge gegeneinander verschoben sind. Dies bedeutet die niedrigste Anforderung an die spektrale Bandweite für jede Stufe des Multiplexerbaumes.

Die verschiedenen Verzögerungen, die alle Vielfache der Ausgangsbitlänge sind, lassen sich mit Hilfe eines Oszillators erzeugen, dessen Frequenz nur  $\frac{1}{N}$  der Sendedatenrate beträgt, wobei  $N$  wiederum das Multiplexverhältnis bezeichnet. Die Bedingung hierfür lautet, daß der Oszillator  $N$  Phasenausgänge mit einer Phasendifferenz von  $\frac{1}{N}$  der Taktperiode zur Verfügung stellt. Den Eingängen des Exklusiv-Oder-Gatters werden einfach Latches vorgeschaltet, die mit den entsprechenden Phasensignalen des Oszillators getaktet werden, so daß das Erscheinen der Eingangsdaten auf den Eingängen des XOR-Gatters entsprechend ihrer Reihenfolge verzögert wird.

Die resultierende Datenrate berechnet sich als Produkt der Taktfrequenz des eingesetzten Oszillators und der Anzahl seiner Phasenausgänge. Im Projekt der INFINEON TECHNOLOGIES AG wurden LC-Oszillatoren entwickelt, die vier bzw. acht solcher Phasensignale bei 5 GHz Taktfrequenz zur Verfügung stellen. Alle weiteren praktischen Vorbehalte außer acht gelassen, ließen sich mit diesen Oszillatoren also theoretisch Datenströme mit 20 respektive 40 GBit/s erzeugen!

Generell gilt jedoch festzustellen, daß die Anforderungen an das Phasenrauschen proportional zur Datenrate steigen. Ein gewöhnlicher aus CMOS-Invertern bestehender 600 MHz Ringoszillator mit 16 Phasenausgängen wird niemals die Ansprüche erfüllen können, die die grundlegenden Standards für 10 GBit/s-Datenstrecken stellen. Noch höhere Datenraten werden erst dann in CMOS möglich sein, wenn die Fortschritte in der Technologie auch die Entwicklung noch besserer Oszillatoren erlauben.

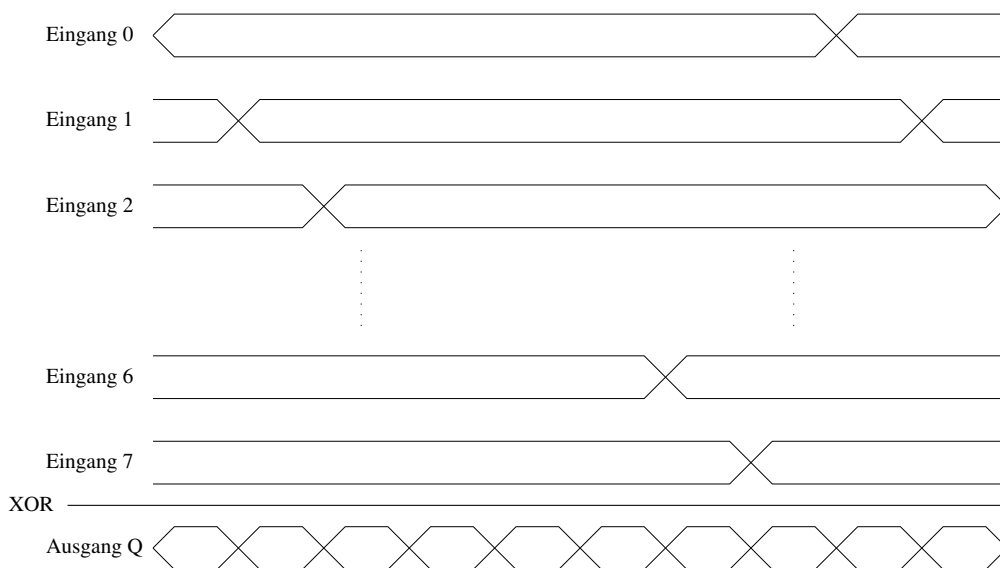


Abbildung 3.13: Darstellung von acht Signalen gemäß Abb. 3.8

### 3.2.5 Ein verbesserter Kodierer

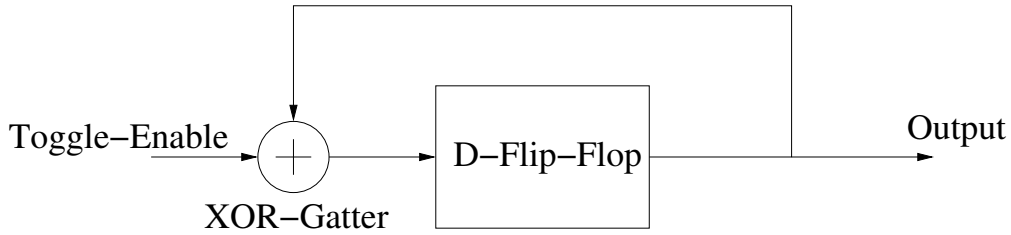
Die in Abschnitt 3.2.2 erläuterte Vorkodierung der Eingangsdaten sei eingangs zusammengefaßt, um auf einen schwerwiegenden Mangel der Schaltung aus Abbildung 3.11 hinzuweisen und dessen Lösung vorzustellen.

Der Kodierer überführt durch Exklusiv-Oder-Bildung benachbarte Bits in einem ersten Schritt von der gewöhnlichen Pegelkodierung in eine flankenkodierte Darstellung. Dadurch wird festgestellt, an welchen Positionen eine Pegelveränderung des Ausgangsstromes verursacht werden muß. Eine Reihe von sogenannten Toggle-Flip-Flops löst diese Veränderung dann aus. Wurde eine Flanke festgestellt, das heißt, ist der Ausgang eines XOR-Gatters logisch Eins, so wechselt das zugehörige Toggle-Flip-Flop seinen Wert. Diese Invertierung des gespeicherten Wertes bewirkt dann den benötigten Wechsel des Ausgangspegels des Multiplexers an der entsprechenden Position im Ausgangssignal.

Diese Form des Vorkodierens der Daten ist notwendig für die Funktion des Multiplexers. Sie hat jedoch einen großen Nachteil: wechselt aufgrund irgendeiner Störung (Einschlag eines  $\alpha$ -Teilchens, Einkopplung etc.) eines der Toggle-Flip-Flop seinen gespeicherten Zustand, dann werden *alle* nachfolgenden Bits am Ausgang des Multiplexers invertiert; der Multiplexer liefert also fortan ausschließlich falsche Ausgangsdaten!

Dieses Problem läßt sich glücklicherweise einfach beheben. Um die Herleitung zu verstehen, wird in Abbildung 3.14 eine typische Realisation eines Toggle-Flip-Flops gezeigt.





**Abbildung 3.14:** Typische schaltungstechnische Realisierung eines Toggle-Flip-Flops

Die Übertragungsfunktion dieser Schaltung ergibt sich zu:

$$Q^n = Q^{n-1} \oplus Z^n \quad (3.6)$$

Dabei bedeuten  $\oplus$  eine Exklusiv-oder-Verknüpfung,  $Q$  den Ausgang und  $Z$  den Toggle-Enable-Eingang der Schaltung. Die Superskripte bezeichnen die zeitliche Ordnung.

Diese Formel entspricht einer Reihe, die sich wie folgt entwickelt:

$$\begin{aligned} Q^1 &= Z^1 \\ Q^2 &= Z^2 \oplus Z^1 \\ Q^3 &= Z^3 \oplus Z^2 \oplus Z^1 \\ &\text{etc.} \end{aligned} \quad (3.7)$$

Bezieht man nun ein, daß in der Kodierschaltung die Toggle-Enable-Eingänge durch eine Exklusiv-oder-Verknüpfung aus jeweils benachbarten Datenbits  $D_x$  (der Index gibt die Bitposition im Datenwort an) gewonnen werden, so erhält man über diesen Zwischenschritt:

$$\begin{aligned} Z_x^n &= D_x^n \oplus D_{x-1}^n \\ Q_x^1 &= D_x^1 \oplus D_{x-1}^1 \\ Q_x^2 &= (D_x^2 \oplus D_{x-1}^2) \oplus (D_x^1 \oplus D_{x-1}^1) \\ Q_x^3 &= (D_x^3 \oplus D_{x-1}^3) \oplus (D_x^2 \oplus D_{x-1}^2) \oplus (D_x^1 \oplus D_{x-1}^1) \\ &\text{etc.} \end{aligned} \quad (3.8)$$

Da die Exklusiv-oder-Operation kommutativ ist, kann man diese Entwicklung umstellen:

$$\begin{aligned}
 Q_x^1 &= D_x^1 \oplus D_{x-1}^1 \\
 Q_x^2 &= (D_x^2 \oplus D_x^1) \oplus (D_{x-1}^2 \oplus D_{x-1}^1) \\
 Q_x^3 &= (D_x^3 \oplus D_x^2 \oplus D_x^1) \oplus (D_{x-1}^3 \oplus D_{x-1}^2 \oplus D_{x-1}^1) \\
 &\text{etc.}
 \end{aligned}
 \tag{3.9}$$

Führt man diese Formel nun in eine Schaltungsstruktur zurück, so liegt die in Abbildung 3.15 gezeigte Variante der Kodiererschaltung von Seite 36 nahe.

Hierbei entsprechen die  $Y_x^n$  den Reihen  $Y_x^{n-1} \oplus D_x^n$ , also z.B.  $Y_x^3 = D_x^3 \oplus D_x^2 \oplus D_x^1$  etc. Nicht in der Abbildung dargestellt ist die wiederum notwendige Speicherung des Ausgangsbits  $Y_{x_{max}}$  des höchstwertigen Toggle-Flip-Flops. Dieses Bit wird zur Erzeugung des nächsten Ausgangswertes  $Q_0$  des Kodierers benötigt.

Die Ausgänge  $Q_x$  liefern dieselbe logische Kombination der Eingangsbits  $D_x^n$  wie in der vorhergehenden Schaltung. Diese quasi „umgedrehte“ Schaltungsvariante hat jedoch einen grundlegenden Vorteil: speichert, durch eine beliebige Störung verursacht, eines der Toggle-Flip-Flops einen falschen Wert, so werden davon jetzt zwei Ausgänge des Kodierers gleichzeitig betroffen. Das bedeutet, daß die Störung genau *zwei* benachbarte, falsche Flanken<sup>1</sup> verursacht und somit lediglich ein einziges falsches Ausgangsbit durch einen Fehler in einem Toggle-Flip-Flop erzeugt wird! Im vorhergehenden Fall, hätte ein Fehler *alle* nachfolgenden Bits invertiert, also verfälscht.

In der früheren Schaltungsvariante wäre es im Falle einer geraden Anzahl von gleichzeitig auftretenden Fehlern in den Toggle-Flip-Flops ebenfalls nur zu einer vorübergehenden Verfälschung des Ausgangsdatenstromes gekommen; allein eine ungerade Zahl von gleichzeitig auftretenden Fehlern hätte den Ausgangsdatenstrom für alle künftigen zu sendenden Bits verfälscht. In der neuen Variante wird jedoch in jedem Fall eine gerade Anzahl an Fehlern an den Ausgängen des Kodierers erzeugt, weshalb der Multiplexer auch nach dem Auftreten einer beliebigen Fehlerkombination in den Toggle-Flip-Flops noch richtig funktioniert.

### 3.3 Blockdiagramm

Abbildung 3.16 stellt die im Projekt der INFINEON TECHNOLOGIES AG nach dem neuen Prinzip realisierte Multiplexervariante als Blockdiagramm dar. Der 16:1-Multiplexer

---

<sup>1</sup>„Falsche Flanke“ soll heißen, eine Flanke erscheint im Ausgangsdatenstrom, wo keine hätte erscheinen sollen, oder umgekehrt.

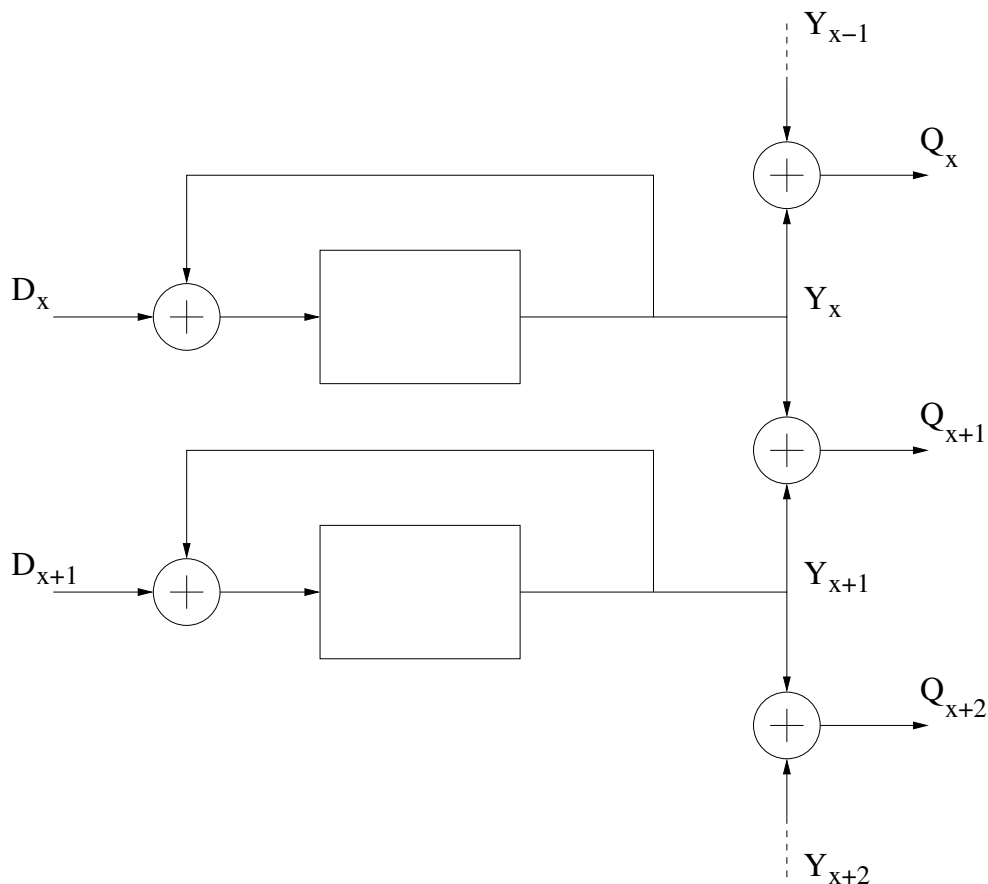


Abbildung 3.15: Alternative Ausführung des Kodierers

zerfällt hier in zwei 8:1-Multiplexer. Die beiden 5 GBit/s-Datenströme werden im Ausgangstreiber des Chips gleichzeitig verstärkt und gemultiplext, so daß der Multiplexer selbst nur ein Multiplexverhältnis von 16:2 realisieren muß. Die beiden Hälften des Multiplexers sind zu beiden Seiten des asynchronen Taktteilers angeordnet, so daß vollständige Symmetrie herrscht.

Vor den Eingängen der beiden Multiplexerbäume befinden sich die Vorkodierer, sie können jedoch im tatsächlichen Chiplayout an anderer Stelle liegen, da ihre Funktion in den langsamen Digitalteil des Netzwerkchips integriert werden kann. Hinter den beiden Multiplexerausgängen stehen einige asymmetrisch verteilte Latches. So weist der eine Ausgang nur ein solches Latch auf, der andere jedoch deren zwei. Das zweite Latch im unteren Zweig verzögert das zugehörige Ausgangsdatum um einen halben Takt, was die Realisierung des Ausgangstreibers/-multiplexers entsprechend vereinfacht.

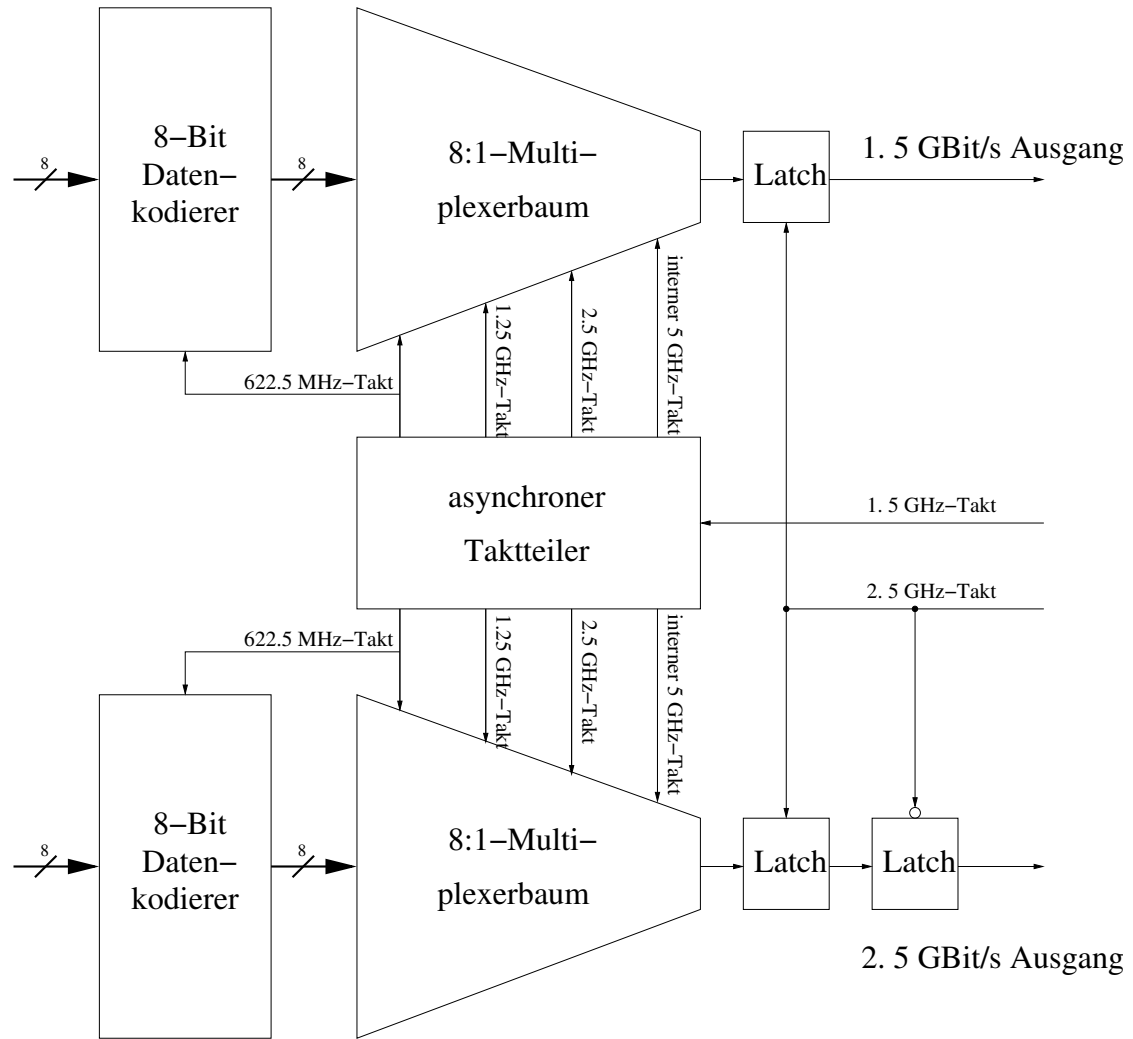


Abbildung 3.16: Blockdiagramm des Multiplexers

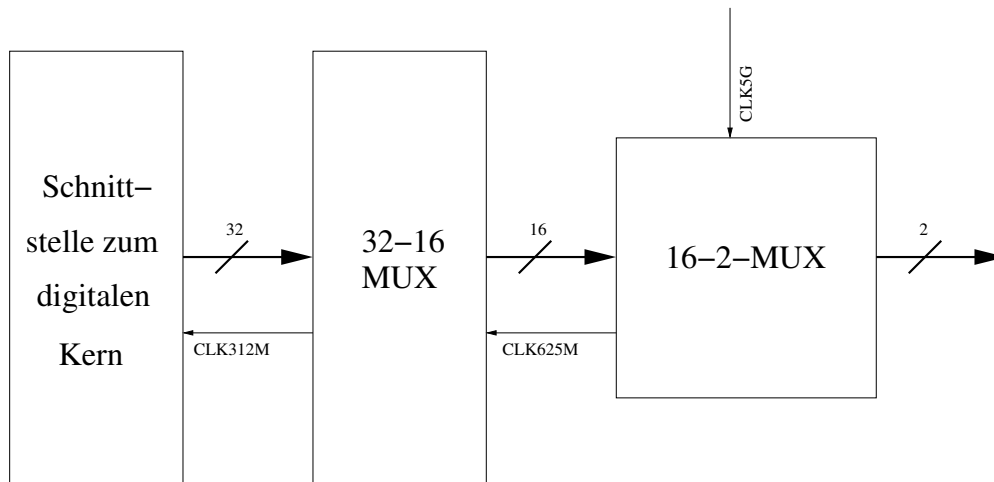


Abbildung 3.17: Überblick über die logischen Blöcke des Multiplexers

Ebenfalls im Blockdiagramm eingetragen sind die zum Betrieb der Schaltung notwendigen verschiedenen Takte. Neben zwei 5 GHz-Takten handelt es sich um die vom asynchronen Taktteiler heruntergeteilten Takte, die in den Multiplexerbäumen benötigt werden. Die zwei 5 GHz-Takte sind verschiedene Phasen des eingesetzten mehrstufigen Oszillators. Durch ihren gleichzeitigen Einsatz kann die Last auf dem Hochgeschwindigkeitstaktbaum auf zwei verschiedene Taktsignale verteilt und der Aufwand für die Verstärkung der Taktsignale seitens des Oszillators gesenkt werden.

## 3.4 Schaltungen

### 3.4.1 Überblick

Der Multiplexer wurde für ein Multiplexverhältnis von 32:2 ausgelegt. Dabei wurde eine erste 32:16-Stufe nach klassischen Verfahren aufgebaut und nur der nachgeordnete 16:2-Multiplexer nach dem beschriebenen neuartigen Multiplexerprinzip ausgeführt. Die Eingangsdaten liegen mit einer Datenrate von 312,5 MBit/s pro Leitung an und werden auf zwei Datenleitungen mit je 5 GBit/s hochgesetzt. Der Ausgangstreiber, der von Herrn Dipl.-Ing. DAVID MÜLLER von INFINEON TECHNOLOGIES entwickelt wurde, führt schließlich die beiden Datenströme zu einem zusammen.

Parametersatz	$I_{D_{rms}}$	$\cdot$	$V_{DD}$	$=$	$P$
fast-ff	100,1 mA	$\cdot$	1,65 V	$=$	165,2 mW
nominal	76,8 mA	$\cdot$	1,5 V	$=$	115,2 mW
slow-ss	59,7 mA	$\cdot$	1,35 V	$=$	80,6 mW

**Tabelle 3.2:** Der Leistungsbedarf des Multiplexers für verschiedene Betriebsparameter

Abbildung 3.17 zeigt die Blöcke des gesamten Multiplexers. Diese sind von links nach rechts

- die Schnittstelle zum digitalen Kern des Transceivers
- ein 32:16 Multiplexer
- der neuartige 16:2-Multiplexer

Die Daten fließen von links nach rechts, während die notwendigen Takte, CLK5G, CLK625M und CLK312M, in Gegenrichtung ausgehend vom 5 GHz-Oszillator bereitgestellt werden. Der 32:16-Multiplexer liegt zusammen mit dem Schnittstellenblock direkt beim digitalen Kern des Transceivers, der mit 312 MHz betrieben wird. Sein Sinn liegt darin, die Busweite für den Transport der Daten über den Chip zur Hochgeschwindigkeitsschnittstelle und den damit verbundenen Flächenbedarf zu senken.

Der gesamte Multiplexer mit allen Takttreibern nimmt eine Chipfläche von ungefähr  $107.000 \mu\text{m}^2$  ein. Der wichtigste Gesichtspunkt bei der Entwicklung des Multiplexers war, eine niedrige Leistungsaufnahme zu erzielen. Tabelle 3.2 zeigt die simulierte Leistungsaufnahme des Multiplexers für die drei wichtigsten Parametersätze. „Fast-ff“ bezeichnet hier einen „schnellen“ Chip, also einen mit niedrigen Einsatzspannungen  $V_{ij}$  und geringen parasitären Kapazitäten, bei niedrigen Temperaturen und hoher Betriebsspannung, „nominal“ einen durchschnittlichen Chip bei normalen Betriebsbedingungen sowie „slow-ss“ einen, der wegen hoher Einsatzspannungen, großer parasitärer Kapazitäten, hoher Betriebstemperatur und niedriger -spannung besonders langsam ist. Es zeigt sich, daß zwischen „fast-ff“ und „slow-ss“ ein Faktor zwei bei der Leistungsaufnahme liegt. Im Nominalfall verbraucht der Multiplexer immerhin etwa 23 % des geplanten Leistungsbudgets. Eine Überarbeitung des Multiplexers ergab ein Verbesserungspotential von 30 - 50 mW, so daß sich sein Anteil an der Gesamtleistung auf ca. 15 % reduzieren läßt. Diese verbesserte Schaltung wurde jedoch noch nicht auf einem Testchip verwirklicht, so daß im Folgenden die ältere Version erläutert wird.

Bei 5 GHz kann ein Signal höchstens ein Logikelement passieren, bevor es wieder in ein Flip-Flop eingelesen wird. Das größte Problem beim Entwurf von komplexeren Hochgeschwindigkeitsblöcken ist die Tatsache, daß der variable Anteil der Durchlaufzeit durch

Flip-Flops und Logikschaltungen in derselben Größenordnung wie die Taktperiode selbst liegt. Deswegen weisen die Bitflanken der 5 GBit/s-Ausgangsdatenströme des Multiplexers eine Verschiebung für die verschiedenen Parametersätze auf. Abbildung 3.18 zeigt ein Augendiagramm für die drei wichtigsten Fälle. Bei „fast-ff“-Parametern beträgt die Verzögerung zwischen Takt- und Datenflanke nur ungefähr 31 ps. Im Nominalfall beläuft sie sich bereits auf  $\approx 42$  ps und steigt für den „slow-ss“-Fall auf 69 ps. Die Verzögerung variiert also um 38 ps, was ungefähr dem Nominalwert entspricht. Im langsamsten Fall ist die Verzögerung größer als eine Dritteltaktperiode.

#### 3.4.2 Die Schnittstelle zum Digitalkern `in_buf_wtx`

Schnittstelle und 32:16-Multiplexer enthalten keinerlei schaltungstechnische Überraschungen. Sie wurden aus Zellen der INFINEON-internen STARLIB-Bibliothek aufgebaut. Der Sinn des Schnittstellenblocks liegt allein darin, definierte Zeitverhältnisse bei der Übergabe der Daten vom ebenfalls aus STARLIB-Zellen bestehenden synthetisierten Digitalkern des Transceivers herzustellen.

Der Block hat die folgenden Eingänge:

- `D<31:0>`
- `RN`
- `ClkIn`
- `inv_clock`
- `gpr_xaui_rloop_force<1:0>`

Sowie die Ausgänge:

- `Q<31:0>`
- `OutRxClk`

`D<31:0>` sind die Sendedaten. Sie werden intern im Register `FF0<31:0>` gespeichert, das durch den Takt `ClkIn` gesteuert wird. `RN` ist der Null-aktive Reset, der im Block `reset_n_sync`, unten links im Schaltplan, zu `ClkIn` synchronisiert wird. Das Signal `inv_clock` wurde vorgesehen, um den internen Takt zu invertieren, sollten Timing-Probleme zwischen Digitalkern und Multiplexer im Test entstehen. Wenn `inv_clock` Null ist, bleibt der Takt unverändert. Über `gpr_xaui_rloop_force` wird der Betriebsmodus

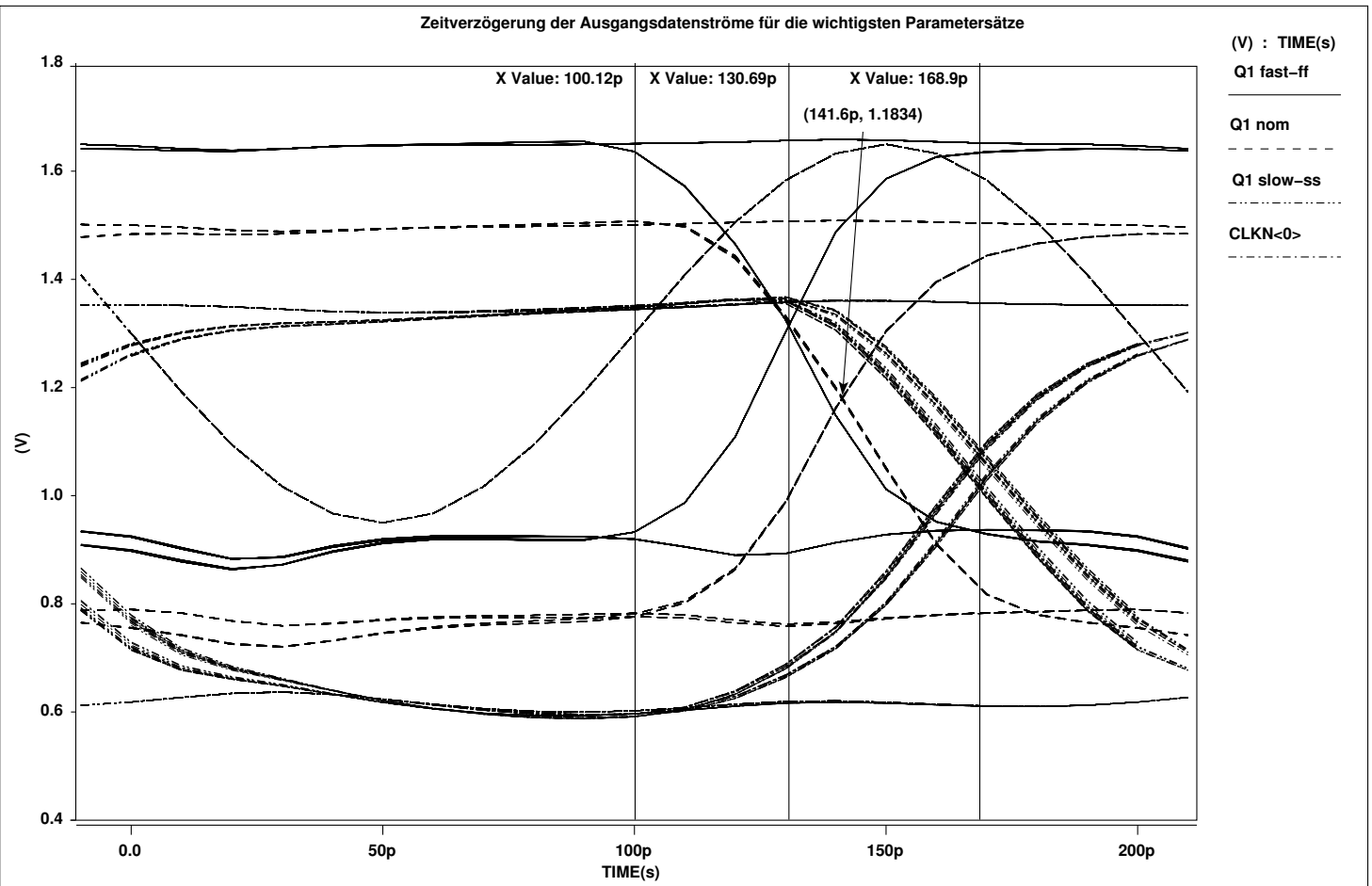


Abbildung 3.18: Augendiagramm für die verschiedenen Parametersätze





ausgewählt. Neben dem normalen Betriebsmodus gibt es drei Testfälle. `grp_xaui_rloop_force` bezeichnet eine vorzeichenlose Binärzahl, die den gewünschten Modus nach der folgenden Tabelle auswählt:

<code>grp_xaui_rloop_force</code>	Beschreibung
00	D<31:0> wird auf die Ausgängen Q<31:0> gegeben
01	Alle Bits werden auf „1“ gesetzt
10	Alle Bits werden auf „0“ gesetzt
11	Gerade Bits werden auf „0“, ungerade auf „1“ gesetzt

Die Auswahl der Quelle wird durch Multiplexer aus der STARLIB bewirkt. Die Ausgänge sind selbsterklärend benannt: Q<31:0> bezeichnet die Datenausgänge, OutRxClk ist der Ausgangstakt, durch den das Timing am Eingang zum digitalen Kern des Transceivers bestimmt wird.

### 3.4.3 Der 32:16-Multiplexer Mux\_32\_16

Der 32:16-Multiplexer besteht ebenfalls aus STARLIB Zellen. Er befindet sich auf dem Chip direkt hinter dem Schnittstellenblock, wo er die Datenwortweite vor der Übermittlung der Daten über den Chip halbiert. Die Schaltung wird in Abbildung 3.20 gezeigt.

Bits mit niedrigeren Indices werden zuerst gesendet. Diese Regel wird im gesamten Multiplexer befolgt. Im Falle des 32:16-Multiplexers bedeutet das, daß die unteren 16 Bits Din<15:0> zuerst durchgeschaltet werden, anschließend die Bits Din<31:16>.

Die Liste der Eingänge:

- Din<31:0>
- clk\_625
- a\_resetn

Die Ausgänge sind:

- Q<15:0>
- clk\_312

Der 625 MHz-Takt `clk_625` liest zum einen die Daten in das Ausgaberegister FF2<15:0> auf der rechten Seite der Schaltung, zum anderen wird aus ihm das Taktsignal `clk_312` der halben Frequenz erzeugt. `clk_312` wird benutzt, um das zu sendende Datenwort auszuwählen, und wird dem Schnittstellenblock zur Synchronisation mit dem digitalen Kern des Sender/Übertragers zur Verfügung gestellt.

### 3 Ein asynchroner Multiplexer mit niedriger Leistungsaufnahme

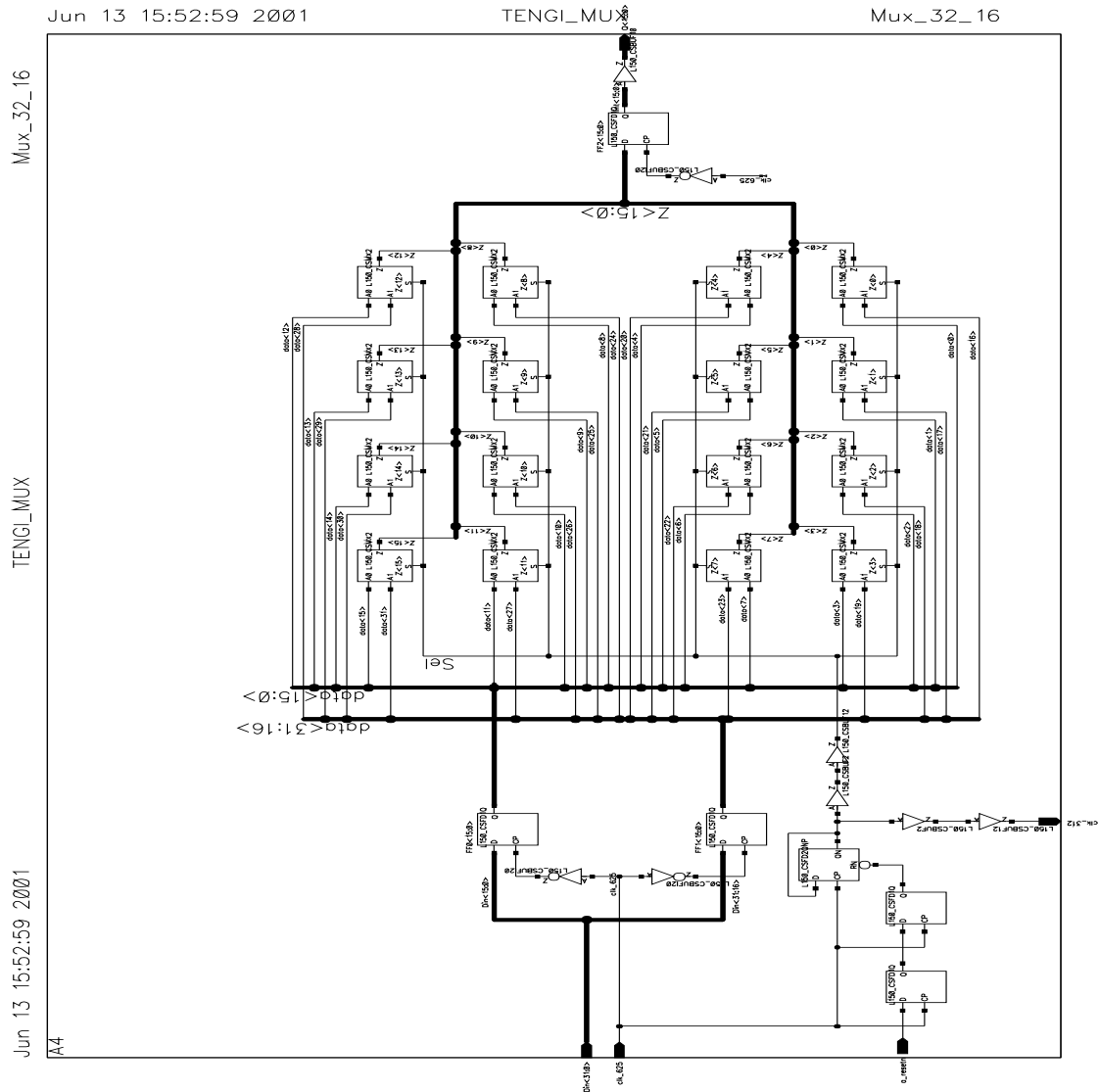


Abbildung 3.20: Die Schaltung des 32:16-Multiplexers

### 3.4.4 Der 16:2-Multiplexer `new_mux_16_2`

Eingänge:

- `DIN<15:0>`
- `reset_n`
- `pd`
- `iref_320u`
- `clkp<3:0>`
- `clkcn<3:0>`

Ausgänge:

- `Q1, Q1N`
- `Q2, Q2N`
- `clk625M`

Das Schaltbild dieses wichtigsten Blockes ist in der Abbildung 3.21 zu sehen. Bei genauere Betrachtung fällt die strukturelle Ähnlichkeit mit dem in Abbildung 3.16 auf Seite 44 gezeigten Blockdiagramm auf. Die Daten fließen von unten nach oben, symmetrisch zur Längsachse sind die beiden größten Unterblöcke, die 8:1-Multiplexer, angeordnet. Sie beziehen ihre jeweils acht Bit breiten Datensätze von dem Vorkodiererblock ganz unten, der zwischen den Dateneingängen und den beiden Multiplexerhälften liegt. Relativ klein dargestellt ist der asynchrone Teiler, der sich rechts des linken 8:1-Multiplexers befindet. Rechts von diesem, mittig in der oberen Hälfte der Schaltung, ist ein länglicher Bias-Block zu sehen, der Biasströme für die in den verschiedenen Blöcken enthaltenen differentiellen Schaltungen bereitstellt. Ein weiterer Bias-Block ist der quadratische Block oben rechts. Durch das `pd`-Signal können beide Bias-Blöcke abgeschaltet werden, wodurch die leistungsintensiven differentiellen Flip-Flops und Treiber in einen stromsparenden Auszustand geschaltet werden. Ist der `pd`-Eingang logisch Eins, dann werden die differentiellen Schaltungsteile abgeschaltet. In allen Schaltplänen, in denen differentielle Funktionseinheiten enthalten sind, finden sich neben deren `refn`-Eingängen als Dioden geschaltete Transistoren, die aus den von den Biasblöcken bereitgestellten Strömen die richtige Biasspannung für die Stromquellentransistoren innerhalb der CML-Schaltungen erzeugen.

### 3 Ein asynchroner Multiplexer mit niedriger Leistungsaufnahme

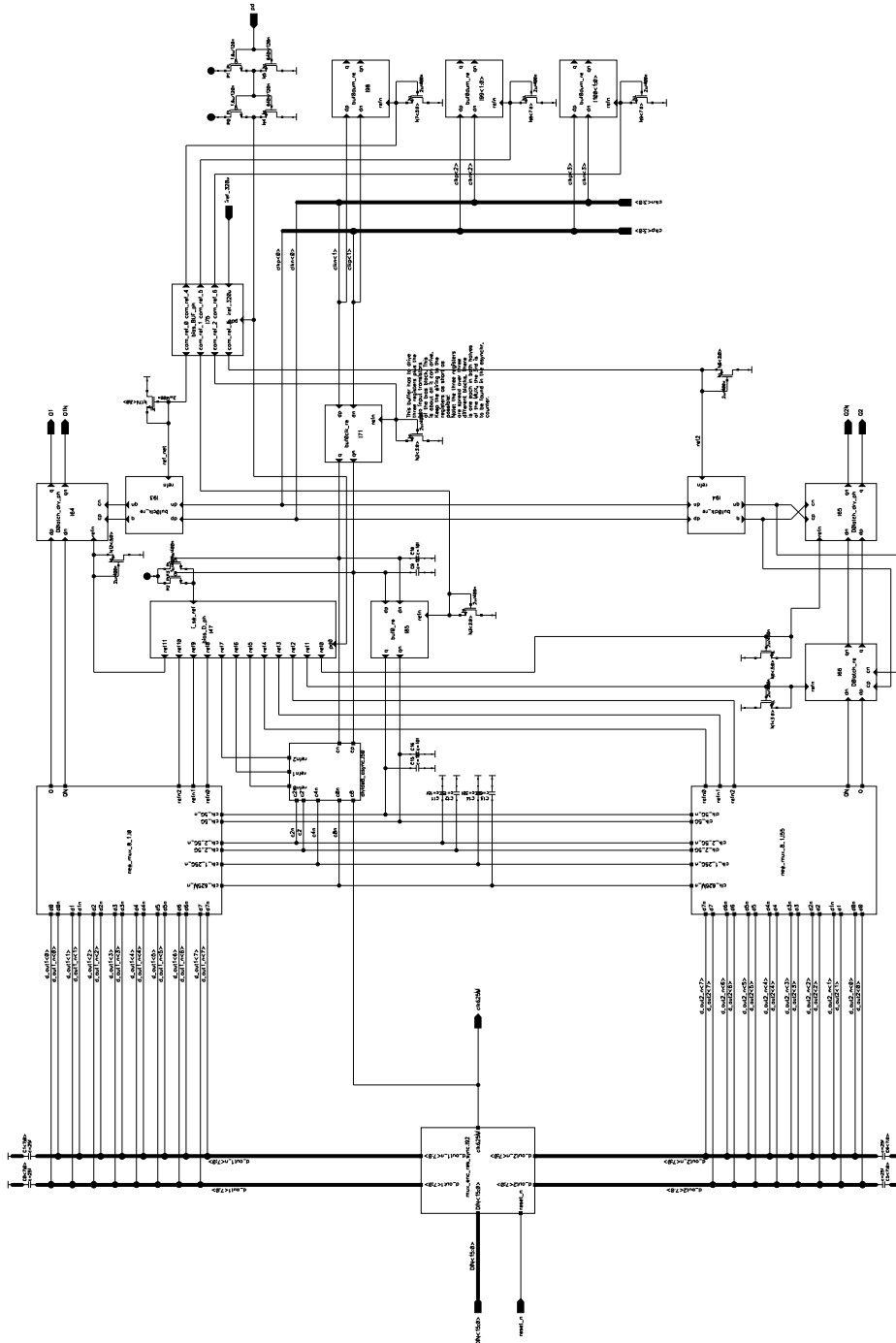


Abbildung 3.21: Die Schaltung des 16:2-Multiplexers

Ganz rechts in der Schaltung sind drei identische Blöcke zu sehen, die sogenannte „Dummies“ sind und zur symmetrischen Terminierung der vier Taktphasen vorgesehen wurden. Sie stellen sicher, daß alle vier Taktphasen gleich stark kapazitiv belastet werden. Die Taktphase  $\text{clkp}\langle 3:2 \rangle$  bzw.  $\text{clk}\langle 3:2 \rangle$  werden im Multiplexer nicht benötigt,  $\text{clkp}\langle 0 \rangle / \text{clk}\langle 0 \rangle$  hingegen durch zwei,  $\text{clkp}\langle 1 \rangle / \text{clk}\langle 1 \rangle$  durch einen differentiellen Treiber belastet. Deshalb wurden für jede Phase so viele Dummy-Treiber vorgesehen, daß jedes Phasensignal durch genau zwei Buffer belastet werden, für  $\text{clkp}\langle 3:2 \rangle / \text{clk}\langle 3:2 \rangle$  also zwei, für  $\text{clkp}\langle 1 \rangle / \text{clk}\langle 1 \rangle$  nur einer und für  $\text{clkp}\langle 0 \rangle / \text{clk}\langle 0 \rangle$  keiner.

Der 16:2-Multiplexer ist direkt mit dem Ausgangstreiber des Chips verbunden, welcher als 2:1-Multiplexer aufgebaut ist. Um ein Signal einer so hohen Datenrate über den Gehäusepin und eine gewisse Strecke auf der Platine treiben zu können, muß der Ausgangstreiber aus sehr großen Transistoren aufgebaut sein, die ihrerseits eine hohe kapazitive Belastung für die vorhergehende Stufe bedeuten. Damit der 16:2-Multiplexer den Ausgangsmultiplexer treiben kann, müssen an seinen Ausgängen Treiber vorgesehen werden. Da in reinen Datenpuffern zusätzliche temperatur- und parameterabhängige Verzögerungen entstehen, werden hier an ihrer Stelle Latches vorgesehen. Dadurch wird an jeder Stufe der Pufferkette der zeitliche Bezug zum Takt des Ausgangstreibers sichergestellt. Die Latches, die die Ausgänge des 16:2-Multiplexers treiben, wurden dreimal so stark ausgelegt wie die sonstigen eingesetzten differentiellen Latches.

Der Ausgangstreiber erwartet die beiden Datenströme vom 16:2-Multiplexer mit einer Verschiebung einer halben Bitlänge zueinander. Deshalb enthält der Pfad zum zweiten Ausgang ein zusätzliches Latch, während der erste nur das Treiberlatch aufweist. Die Latches werden von einer anderen Taktphase gesteuert als der asynchrone Zähler und die beiden 8:1-Multiplexer. Auf diese Weise wird die kapazitive Last auf mehrere Taktphasen verteilt und somit die Bereitstellung des 5 GHz-Taktes vereinfacht.

Es wurde vor dem Entwurf festgelegt, daß die Ausgangsdaten synchron<sup>2</sup> zum Takt  $\text{clkp}\langle 0 \rangle / \text{clk}\langle 0 \rangle$  bereitgestellt werden sollen. Um eine robuste zeitliche Abstimmung zwischen den beiden mit unterschiedlichen Taktphasen betriebenen Schaltungsteilen sicherzustellen, müssen die Taktflanken des Taktes, der Multiplexer und asynchrone Zähler steuert, etwas später erscheinen als die im Takt der Ausgangslatches. Deshalb werden Zähler und Multiplexer mit dem Signal  $\text{clkp}\langle 1 \rangle / \text{clk}\langle 1 \rangle$  getaktet.

In der Schaltung befinden sich neben den Treibern für die beiden Ausgangslatches zwei weitere; sie sind in Serie geschaltet und zentral in der Schaltung zu sehen. Der erste verstärkt den Takt  $\text{clkp}\langle 1 \rangle / \text{clk}\langle 1 \rangle$  und treibt so den asynchrone Zähler und den zweiten Treiber. Der zweite Treiber ist zum einen notwendig, um Last vom ersten zu nehmen, zum anderen, um die Verzögerung zu kompensieren, die im asynchrone Teiler auf dem 2,5 GHz-Takt durch einen dort benötigten Treiber anfällt.

---

<sup>2</sup>„Synchron“ bedeutet hier „mit nur zwei Gatterlaufzeiten belastet“. Diese entstehen durch den Einsatz eines Taktpuffers vor den Takteingängen der Ausgangslatches und das Ausgangslatch selbst.

### 3.4.5 Die Komponenten des 16:2-Multiplexers

#### Der 8:1-Multiplexer

- `d0-7, d0n-7n`
- `clk_5G, clk_5G_n`
- `clk_2_5G, clk_2_5G_n`
- `clk_1_25G_n`
- `clk_625M_n`
- `refn[0-2]`

Ausgänge:

- `Q, QN`

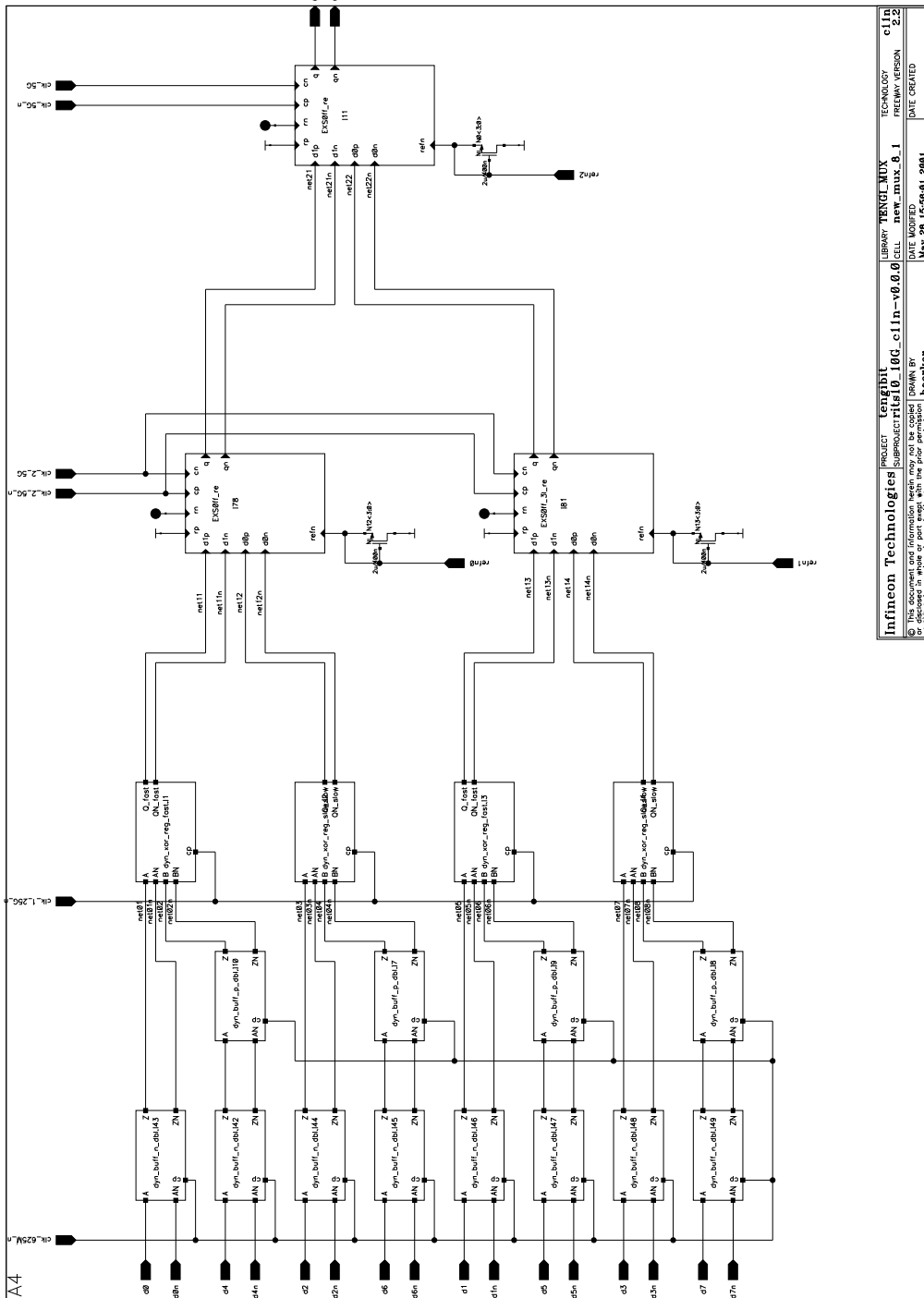
Die Dateneingänge sind differentiell ausgelegt, wobei ein „n“ am Ende des Signalnamens das invertierte Signal bezeichnet. „d0“ bis „d7“ sind die acht Dateneingänge. Die vom asynchronen Zähler erzeugten Takte sind an die `clk_XXX`-Eingänge der 8:1-Multiplexer angeschlossen. Während die 5 und 2,5 GHz-Takte differentiell vorliegen, laufen die langsameren Teile des Multiplexers mit einfachen Takten, hier `clk_1_25G_n` und `clk_625M_n`.

Die Baumstruktur ist in Abbildung 3.22 deutlich zu erkennen. Insgesamt besteht der Multiplexerbaum aus fünf verschiedenen Unterblöcken. Links beginnend sind dies die Puffer `dyn_buff_n_dbl` und `dyn_buff_p_dbl`, die full-swing XOR-Multiplexer `dyn_xor_reg_fast` und `dyn_xor_reg_slow` sowie der differentielle XOR-Multiplexer `EXS0ff_re`.

Die dynamischen Komponenten wurden in der sog. NORA-Logik [7] aufgebaut. Stellvertretend werden zwei der Blöcke gezeigt und kurz erläutert. Abbildung 3.23 zeigt einen Ausschnitt aus der Schaltung des `dyn_buff_p_dbl`, einem Block, der ein Bit einen halben Takt lang zwischenspeichert; die Schaltung ist doppelt ausgeführt, da neben dem Eingangswert `A` auch seine Inverse `AN` zwischengespeichert werden soll.

Dynamische Schaltungen kennen zwei Taktphasen, eine „Precharge“-Phase, in der ein dynamischer Knoten vorgeladen und der Ausgangswert isoliert wird, sowie die Auswertphase, in der die Ladung abhängig vom Eingangswert entweder erhalten oder umgeladen wird. Bedingung für den Betrieb einer dynamischen Schaltung nach dem NORA-Prinzip ist, daß der logische Eingangswert der Schaltung während der gesamten Auswertphase

### 3 Ein asynchroner Multiplexer mit niedriger Leistungsaufnahme



Infineon Technologies	PROJECT	ten@bit	LIBRARY	10Q_c1n-v0.0.0	TECHNOLOGY	c1n
	SUBPROJECT	10_10Q_c1n-v0.0.0	CELL	new_mux_0_1	PRESYN VERSION	2.23
© This document and information herein may not be copied, drawn, or reproduced without the prior written permission of Infineon Technologies AG. All rights reserved.						
	DATE MODIFIED	May 20 15:56:01 2001	DATE CREATED			
	DRAWN BY	boerker				

Abbildung 3.22: Der 8:1-Multiplexer



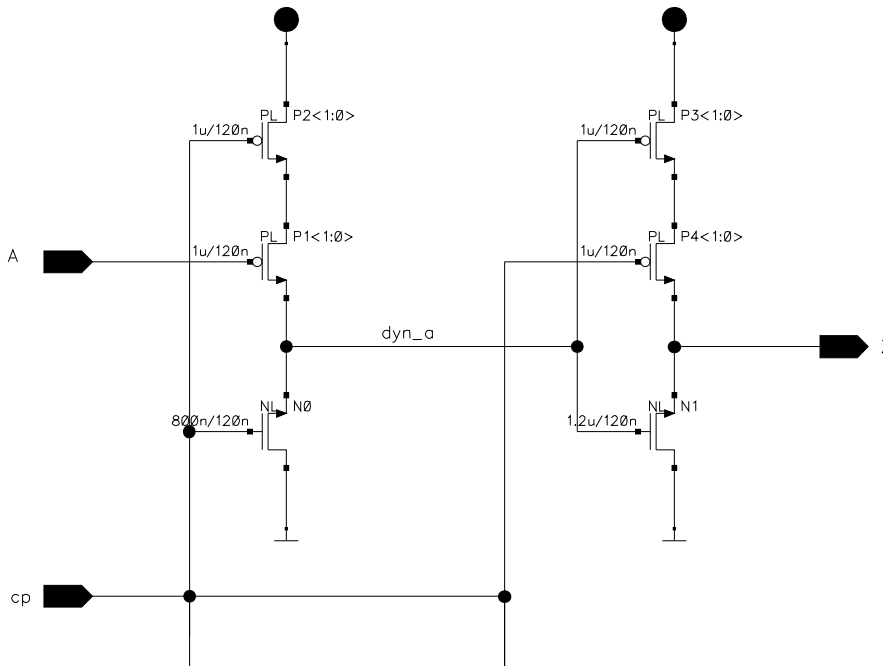


Abbildung 3.23: Eine Hälfte des dynamischen Buffers in PMOS-Ausführung

stabil anliegen muß. Während der Auswertphase wird aber ihr Ausgangswert aktualisiert, ändert sich also gegebenenfalls, weshalb niemals zwei gleichartige Blöcke aufeinanderfolgen können, weil dann die Bedingung für die nachfolgende Stufe verletzt würde. Deshalb folgt auf jeden Block in dynamischer NMOS-Logik einer in dynamischer PMOS-Logik und umgekehrt. PMOS und NMOS-Blöcke unterscheiden sich darin, daß erstere die Auswertphase während der „high“-Phase des Taktes, letztere während der „low“-Phase durchführen.

In der gezeigten PMOS-Variante der NORA-Logik wird der dynamische Knoten `dyn_a` entladen, wenn `cp` logisch 1 ist, wechselt `cp` seinen Wert, dann wird `dyn_a` abhängig vom Eingang `A` umgeladen. Ist `A` logisch 0, dann wird `dyn_a` auf  $V_{DD}$  aufgeladen, ist `A` logisch 1, dann bleibt die vorgeladene logische Null auf `dyn_a` erhalten. In diesem Fall wird der Ausgang `Z` auf Eins gezogen, andernfalls wird `Z` über den Transistor `N1` entladen.

Der Block `dyn_buff_n_dbl` ist die NMOS-Variante des Zwischenspeichers; sie wird nicht extra erklärt. Die beiden Blöcke `dyn_xor_reg_fast` und `dyn_xor_reg_slow` sind Realisierungen einer XOR-Funktion in NORA-Logik. Sie beide unterscheiden sich lediglich

voneinander darin, daß die **slow**-Variante das Ergebnis der XOR-Funktion einen zusätzlichen halben Takt zwischenspeichert, bevor es ausgegeben wird. Dadurch wird die nach dem neuen Multiplexerprinzip notwendige Verschiebung des einen Eingangsdatums um einen halben Takt bewirkt.

Die differentielle Umsetzung eines Flip-Flops mit XOR-Verknüpfung zweier Eingangssignale **EXSOff\_re** wurde von Herrn Dipl.-Ing. VOLKMAR REBMANN von INFINEON TECHNOLOGIES entwickelt. Aus patentrechtlichen Gründen kann die Schaltung hier ebensowenig gezeigt werden, wie die ihrer Grundform, des differentiellen Master-Slave-Flip-Flops.

#### Der asynchrone Taktteiler `divide8_async`

Die Liste der Eingänge:

- `cp, cn`
- `refn0, refn1, refn2`

Ausgänge:

- `c2, c2n`
- `c4n`
- `c8, c8n`

Dieser Block ist in Abb. 3.24 zu sehen. Die Eingänge `cp` und `cn` bezeichnen den differentiellen 5 GHz-Takt, mit dem der asynchrone Zähler betrieben wird. `refn[0-2]` sind Bias-Eingänge. Das Biaskonzept wird im Abschnitt 3.4.5 erläutert. Die Ausgänge `c2` bis `c8n` sind die verschiedenen Taktausgänge für den Betrieb des Multiplexers. Das Verhältnis ihrer Frequenz zu der des Haupttaktes ist durch die Ziffer im Namen des jeweiligen Taktsignales angegeben; so besitzt z.B. `c2` eine Frequenz von 2,5 GHz.

Der asynchrone Taktteiler ist das Zentralstück der Multiplexerschaltung; in ihm entscheiden sich die zeitlichen Bezüge zwischen den verschiedenen Takten und damit die korrekte Datenübergabe zwischen den einzelnen Stufen des Multiplexerbaumes. Das Grundprinzip des Taktteilers ist sehr einfach, besteht er doch lediglich aus einer Kette mehrerer als Taktteiler geschalteter D-Flip-Flops. Die Flip-Flops, die die Glieder der Taktteilerkette bilden, wurden nach den jeweils geltenden Anforderungen an ihre Geschwindigkeit ausgewählt. Deswegen bestehen die ersten beiden Teilerstufen aus schnellen differentiellen Flip-Flops, während für die letzte Stufe ein leistungssparendes dynamisches Flip-Flop

### 3 Ein asynchroner Multiplexer mit niedriger Leistungsaufnahme

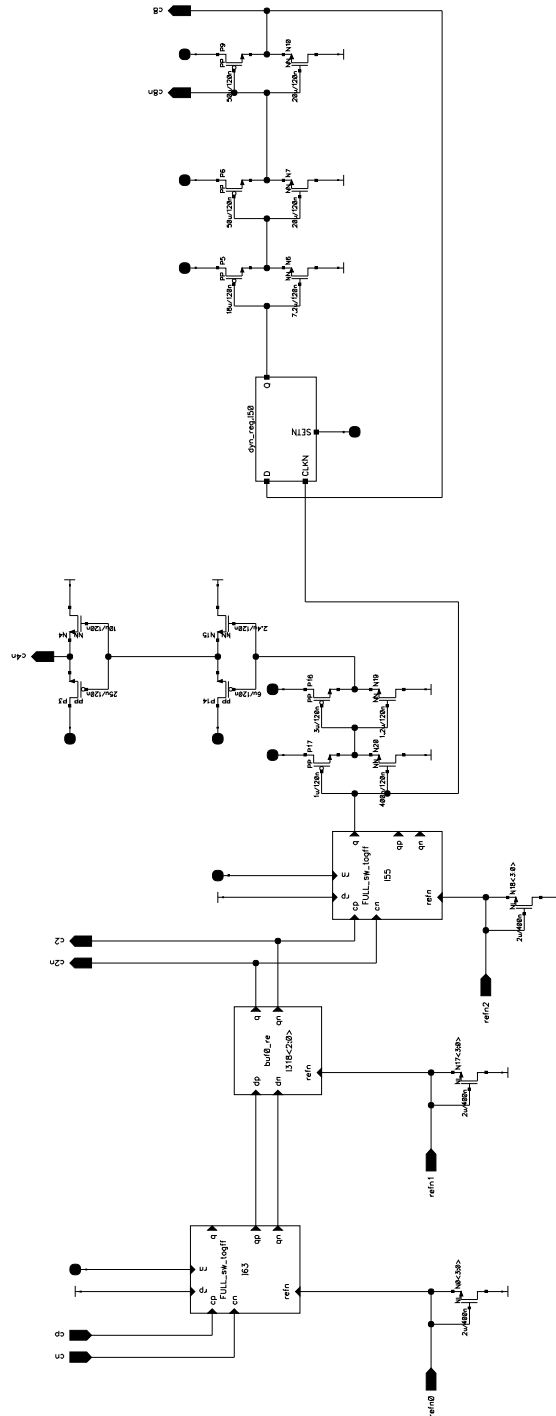


Abbildung 3.24: Der asynchrone Zähler

verwendet wurde. Die Ausgangstreiber der unteren beiden Stufen sind einfache CMOS-Inverter, die der bekannten 1 :  $e$ -Lastanpassungsoptimierung [8] folgen.

Wie schon im Abschnitt 3.1.4 ab Seite 27ff. erklärt wurde, liegt die größte Herausforderung beim Entwurf des Multiplexers in der Balancierung der Verzögerungen zwischen den einzelnen vom Taktteiler erzeugten Takten. Beim asynchronen Zähler summieren sich die Verzögerungen jedes Gliedes der Kette, so daß auch der Entwurf der langsameren Teile schwierig werden kann. Um für jede mögliche Prozeß- und Betriebsparametervariation funktionierende zeitliche Verhältnisse zwischen den einzelnen Takten garantieren zu können, mußten sehr sorgfältige Simulationen unter Einbeziehung der Layout-Daten durchgeführt werden. Mittels solcher Simulationen konnte dann bestimmt werden, ob beispielsweise ein weiterer Takttreiber in die Pfade geschaltet werden mußte oder aber sogar ein Ausgangstakt invertiert zur Verfügung gestellt werden sollte.

Dieser Aufwand, der sich bei Einsatz eines asynchronen Taktteilers nicht vermeiden läßt, ist ein großer Nachteil. Er wird jedoch mehr als aufgewogen, weil der asynchrone Zähler selbst eine geringe Last für den 5 GHz-Takt darstellt, der nur ein Hochgeschwindigkeits-Flip-Flop treiben muß. Deshalb ist nur ein Zwischenverstärker notwendig, um den 5 GHz-Takt dem Taktteiler zuzuführen. Die Schwierigkeit des Entwurfs wird vorteilhaft von jenen Schaltungsteilen, die bei 5 GHz laufen, zu langsameren hin verschoben.

#### Das dynamische Flip-Flop `dyn_reg`

Eingänge:

- D
- SETN
- CLKN

Ausgang:

- Q

Die Schaltung des dynamischen Flip-Flops ist in Abb. 3.25 zu sehen. Es besitzt zwei dynamische Knoten und zählt zu der Klasse der „true single-edge triggered dynamic flip-flops“, also der flankengesteuerten dynamischen Flip-Flops. Dies unterscheidet es von z.B. der NORA-Logik, bei der das Eingangsdatum frühestens nach einer halben Taktphase auf dem Ausgang erscheint, also erst nach Auftreten einer zweiten, entgegengesetzten Flanke im Taktsignal. Der erste dynamische Knoten wird in der Vorladephase auf  $V_{DD}$

### 3 Ein asynchroner Multiplexer mit niedriger Leistungsaufnahme

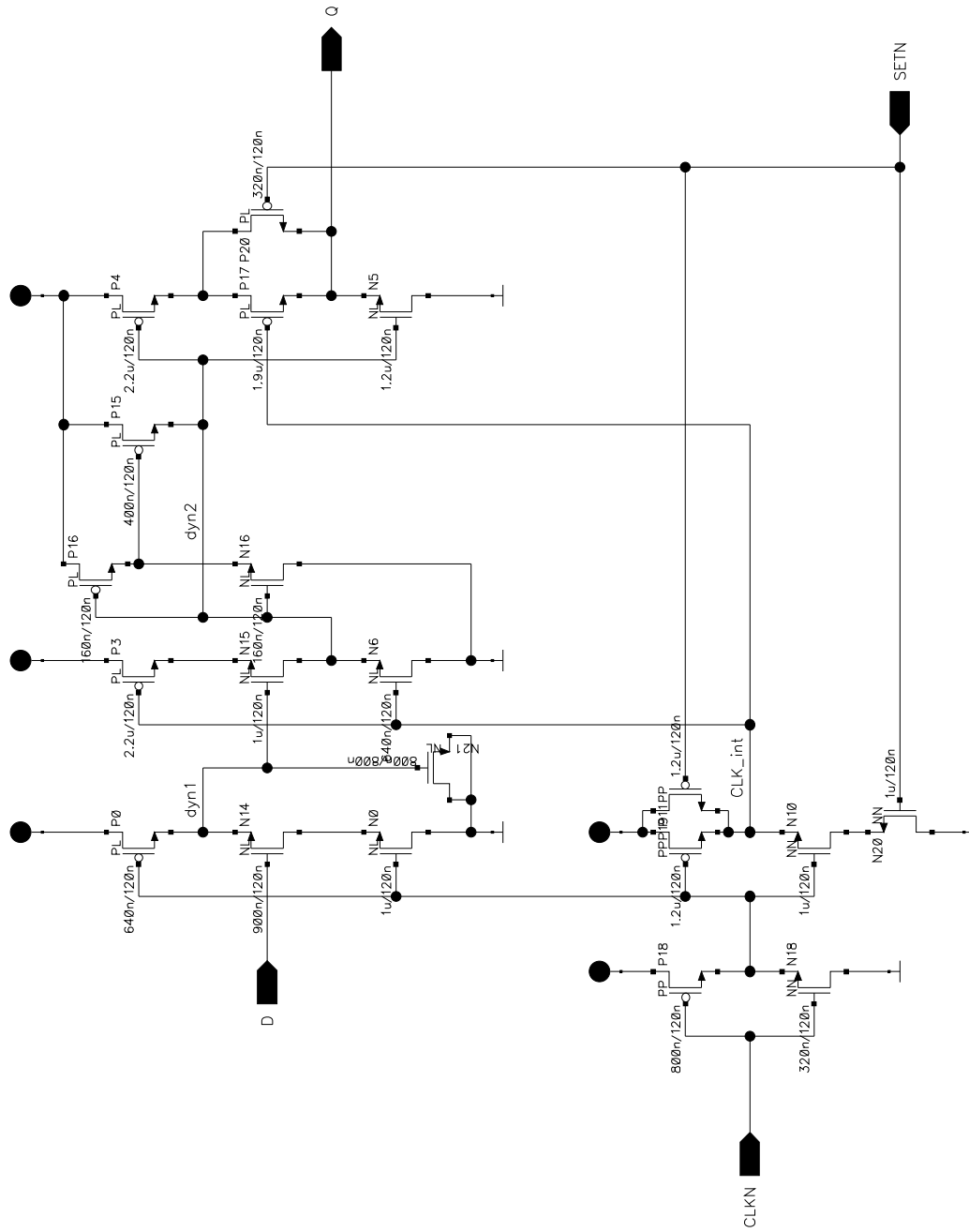


Abbildung 3.25: Das dynamische Flip-Flop

aufgeladen, der zweite hingegen wird entladen. Die erste Stufe entspricht dem gewöhnlichen Schaltkonzept dynamischer Schaltungen. Ihr dynamischer Knoten wird anhängig vom logischen Zustand des Dateneinganges D entladen. Zwei solcher dynamischer Stufen lassen sich jedoch nicht hintereinanderschalten, bestünde doch die Gefahr, daß der zweite dynamische Knoten aufgrund der vorgeladenen logischen Eins am ersten dynamischen Knoten entladen würde, bevor letzterer seinen korrekten Wert angenommen hat. Mit vorsichtigem Entwurf kann dieser Effekt höchstens abgemildert werden, so daß auf jeden Fall die Ladung auf dem zweiten Knoten und damit die Störsicherheit des Flip-Flops verringert würde.

Der pull-up-Pfad des zweiten dynamischen Knotens besitzt einen NMOS-Transistor. Dieser ist nötig, um einen Vorlade- und einen Vorentladeknoten hintereinanderschalten zu können. Denn der zweite dynamische Knoten soll nur dann seinen vorbestimmten Zustand ändern, wenn der erste Knoten aufgrund einer logischen Null am Dateneinganges D seine Ladung gehalten hat. Scheinbar entsteht hier wieder dasselbe Problem wie bei zwei hintereinandergeschalteten Vorladeknoten: der zweite Knoten kann wenigstens teilweise umgeladen werden, solange der erste noch nicht seinen entgültigen Wert angenommen hat. Dies wird in dieser Schaltung aber dadurch verhindert, daß der für die zweite Stufe benötigte Takt durch ein NAND-Gatter verzögert wird. Der zweite Knoten wird also erst aktiviert, wenn der erste schon eine gewisse Zeit hatte, seinen neuen Zustand anzunehmen. Durch diese Maßnahme wird der zweite Knoten sehr effektiv geschützt.

Würde der erste Knoten etwa anderthalb Inverterlaufzeiten lang nicht entladen, weil D logisch Null war und geblieben ist, dann wird der zweite Knoten aufgeladen und Änderungen am D-Eingang beeinflussen das Ergebnis der Operation nicht mehr. Das Flip-Flop ist folglich genaugenommen nicht flankengesteuert, sondern besitzt ein Zeitfenster von etwa anderthalb Inverterlaufzeiten, in dem sein nächster Ausgangswert bestimmt wird.

Wegen des NMOS-Transistors kann der pull-up-Pfad den zweiten Knoten jedoch nur bis auf ungefähr  $V_{DD} - V_{\theta}$  ziehen; hierzu wurden die Transistoren P15, P16 und N16 vorgesehen. P16 und N16 bilden einen Inverter, der wiederum den Transistor P15 bedingt anschaltet und auf diese Weise sowohl den zweiten dynamischen Knoten auf  $V_{DD}$  zieht, als ihn auch statisch sichert. Damit das Flip-Flop den zweiten Knoten zuverlässig aufladen kann, muß er vom pull-up-Pfad hoch genug aufgeladen worden sein, um N16 anzuschalten. Dies ist bei Einsatz von sog. low-threshold-Transistoren gewährleistet.

Der Transistor N21 fungiert lediglich als zusätzliche Sicherung des ersten dynamischen Knotens und erhöht dessen Kapazität. Diese Maßnahme war nötig, da die Leckströme bei einigen Betriebs- und Prozeßparameterkombinationen zu groß wurden und den korrekten Betrieb des Flip-Flops gefährdeten.

Das Flip-Flop besitzt ein Set-Signal, das Null-aktiv ist. Wenn SETN Null ist, dann wird der zweite dynamische Knoten aufgrund des NAND-Gatters im Taktpfad permanent entladen. Der Transistor P20 zieht dann den Ausgang des Flip-Flops auf logisch Eins.

## Biasing

Eingänge:

- `iref_320u`
- `pd0`

Ausgänge:

- `refn<>`

Als letzter Block verbleibt der Biasblock<sup>3</sup>. Er soll für alle differentiellen Flip-Flops und Treiber passende Biasströme bereitstellen. Es müssen deshalb *Biasströme* benutzt werden, weil Spannungen für die Stromquellentransistoren an den Fußpunkten der differentiellen Schaltungen nicht störungsfrei über den Chip verteilt werden können. Zuvor war ein Konzept erwogen worden, bei dem die Stromquellentransistoren in den Biasblock verlegt wurden. Das hätte jedoch zur Folge gehabt, daß der gesamte in den differentiellen Komponenten fließende Strom im Biasblock in das Substrat geleitet worden wäre. Dadurch wäre der Chip punktuell wegen der Konzentrierung der Ströme auf eine hohe Temperatur aufgeheizt worden und ein starker Temperaturgradient über den Chip entstanden. Deshalb wurden die Stromquellen in den einzelnen Schaltungen belassen und vor die Gates der Fußpunkttransistoren eine Diode geschaltet, durch die der Biasstrom aus dem Biasblock fließt und die richtige Spannung erzeugt. Einige solcher Dioden kann man auch in der Schaltung des 16:2-Multiplexers auf Seite 53 sehen.

Die Weite des Diodentransistors ist vier bis acht Mal kleiner als die des Stromquellentransistors, so daß auch nur ein Viertel bzw. ein Achtel des Stromes, der durch die differentielle Schaltung fließt, zusätzlich als Biasstrom zur Verfügung gestellt werden muß. Zwar bedeutet der Biasstrom einen erhöhten Leistungsverbrauch, doch nur auf die beschriebene Weise kann das Biaskonzept Störungen der Daten über die Biaseingänge unterdrücken. Um den zusätzlichen Leistungsverbrauch möglichst klein zu halten, wird ein Diodentransistor immer für wenigstens zwei nahe beieinander stehende Blöcke wie z. B. für Master- und Slave-Latch eines Flip-Flops benutzt.

Die Idee des Biasblockes ist einfach. In ihrem Kern befindet sich ein differentielles Latch, dessen Ausgangsspannung mit einer Referenz verglichen wird. Diese wird von einem temperaturkonstanten Biasstrom gestellt, der durch einen Widerstand fließt. Dabei ist

---

<sup>3</sup>Der Vorkodierer wurde per Synthese aus einer einfachen VHDL-Beschreibung erzeugt. Die naturgemäß sehr unübersichtliche Schaltung wird hier ausgespart, weil der ihr zugrundeliegende Algorithmus bereits ausführlich beschrieben wurde.

### 3 Ein asynchroner Multiplexer mit niedriger Leistungsaufnahme

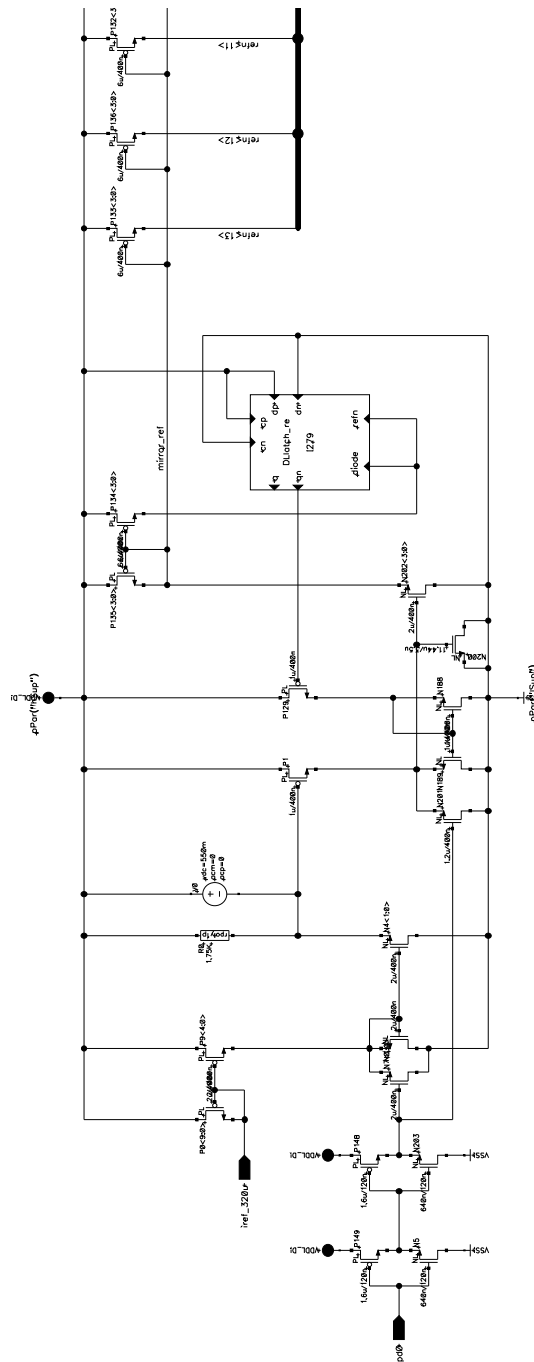


Abbildung 3.26: Die Schaltung des Biasblocks



genaugenommen der Strom nicht temperaturkonstant, sondern so geartet, daß sich seine Temperaturabhängigkeit und die des Widerstandes kompensieren.

Der Biasblock stellt einen Regelkreis dar, der automatisch die Differenzspannung am Ausgang des Latches gleich der Referenz einstellt. Der Strom durch die beiden PMOS-Transistoren P1 und P129 ist im stabilen Zustand gleich. Ist der Strom durch P1 zuerst noch höher, dann steigt die Spannung am Gate von Transistor N202. Dadurch wird ein höherer Strom durch den PMOS-Transistor P135 erzwungen, der wiederum über P134 in das differentielle Latch gespiegelt wird. Dessen Ausgangsspannungsdifferenz steigt mit dem eingepprägten Strom, was solange den Strom durch P129 erhöht, bis dieser gleich dem durch P1 ist. Der Biasstrom wird über über zahlreiche Stromspiegeltransistoren vervielfacht und an die weiteren differentiellen Blöcke verteilt. Von diesen Transistoren sind der Übersichtlichkeit wegen nur drei auf der rechten Seite der Schaltung dargestellt. Die Stabilität des Regelkreises wird durch den als MOS-Kapazität geschalteten NMOS-Transistor N200 sichergestellt.

## 3.5 Meßergebnisse

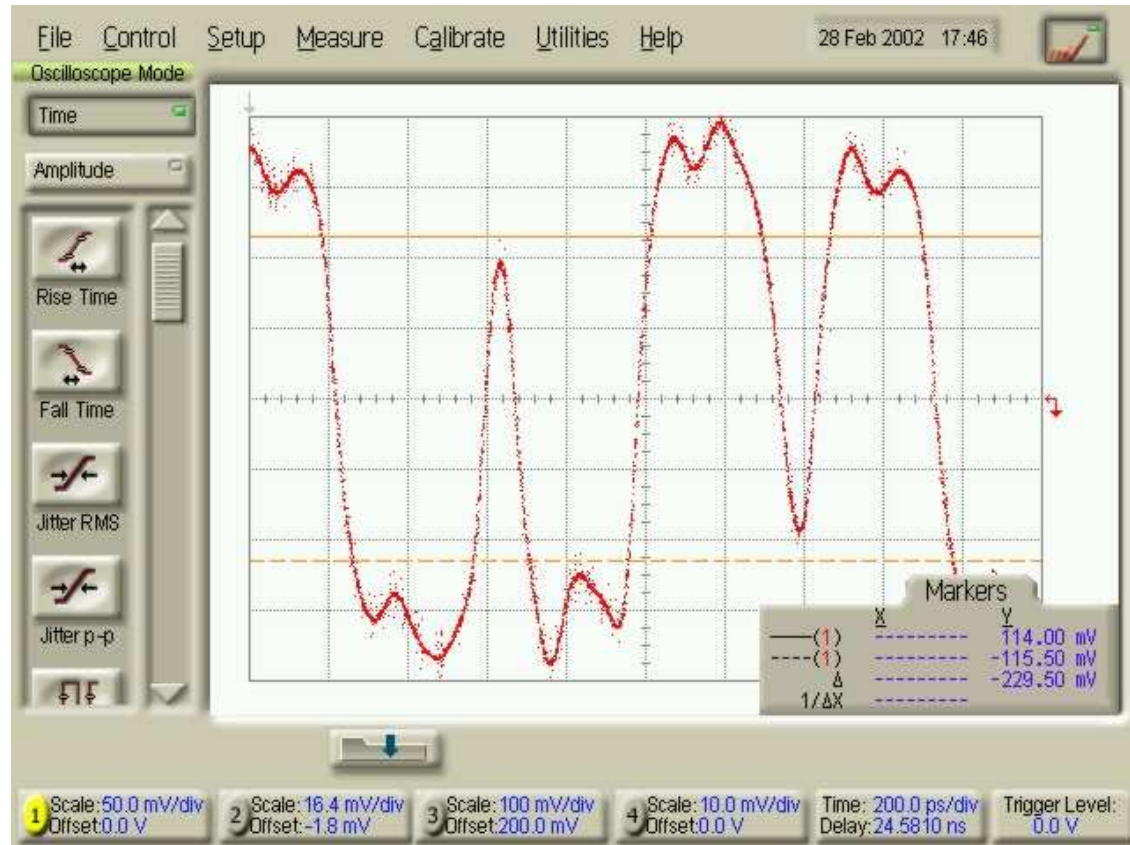


Abbildung 3.27: Das gemessene Datensignal

Die Ausbeute bei der Produktion des Testchips mit den vorgestellten Schaltungen war bedingt durch den frühen Zeitpunkt in der Entwicklung des 120 nm-Prozesses sehr gering. Es fanden sich insgesamt lediglich einzelne Chips, die mehr als nur rudimentäre Funktionalität zeigten. Die Überprüfung elektronischer und sonstiger fertigungstechnischer Parameter bewies, daß die Ausbeuteschwierigkeiten allein auf die Produktion zurückzuführen war. Tatsächlich waren die entworfenen Schaltungen sogar trotz der völlig von der Zielsetzung abweichenden Parameter noch in einigen Fällen lauffähig.

Die grundsätzliche Funktion des Multiplexers wurde überprüft, indem das Sendesignal schon auf dem Chip auf den Eingang der Takt-/Datenrückgewinnung geschaltet wurde. Die Sendedaten bestanden aus einer pseudo-zufälligen Serie, die auf der Empfängerseite mit einer lokal erzeugten Kopie des Bitstroms verglichen wurde. Auf diese Weise konnte die Bitfehlerrate festgestellt werden, die für ein Generatorpolynom siebter Ordnung bei

etwa  $10^{-12}$  lag, für ein solches der 54. Ordnung jedoch auf zu schlechte  $10^{-9}$  stieg. Dieses Ergebnis weist jedoch darauf hin, daß die Bitfehler erst auf der Empfangsseite entstanden, da sich der Sendetakt eines längeren pseudo-zufälligen Bitstroms gewöhnlich schwieriger extrahieren läßt und deshalb die Datenrückgewinnung unzuverlässiger wird.

Als weitere Kontrolle war es im Testaufbau möglich, Bitmuster von 32 Bit Länge einzustellen, die fortlaufend gesendet wurden. Die Überprüfung der Ausgangsdaten wurde in diesem Fall durch ein Oszilloskop per Augenschein vorgenommen. Dabei fielen bei Voreinstellen bestimmter Bitmuster gefolgt von einer Umprogrammierung auf bestimmte andere Bitmuster auf, daß fortan jedes zweite Bit invertiert gesendet wurde. Diese Symptomatik weist auf ein Problem des Vorkodierers in einer Multiplexerhälfte hin. Der Vorkodierer wurde nach dem in Abbildung 3.11 auf Seite 36 gezeigten Prinzip mit den in Abschnitt 3.2.5 diskutierten Nachteilen aufgebaut. Es erscheint deshalb wahrscheinlich, daß die Invertierung der Bits von einer mangelnden Synchronisation des Resets der Toggle-Flip-Flops im Vorkodierer herrührt. Das Problem wird durch die in Abbildung 3.15 auf Seite 43 dargestellte verbesserte Vorkodierschaltung gelöst.

Abbildung 3.27 zeigt einen Ausschnitt des Sendesignals. Die Horizontalablenkung des Oszilloskops wurde zu 200 ps, die Vertikalaussteuerung zu 50 mV pro Kästchen gewählt. Man erkennt deutlich, daß auch alleinstehende Bits trotz der offensichtlichen Bandweitenbegrenzung korrekt gesendet werden. Abbildung 3.28 zeigt ein Datenauge des Sendesignals. Die Horizontalablenkung wurde bei dieser Messung auf 50 ps pro Kästchen gesetzt. Das Datenauge ist für einen frühen Testchip bereits erstaunlich weit geöffnet und unterliegt einem akzeptablen Jitter, der mit abgelesenen 20 ps peak-to-peak jedoch schon außerhalb der Spezifikationen liegt. Da jedoch der Ausgangsmultiplexer, der nach klassischer Bauweise ausgeführt ist, das Timing des Ausgangssignales bestimmt, dürfte dieser Mangel im wesentlichen von dieser Stufe und nicht vom neuartigen Multiplexer herrühren.

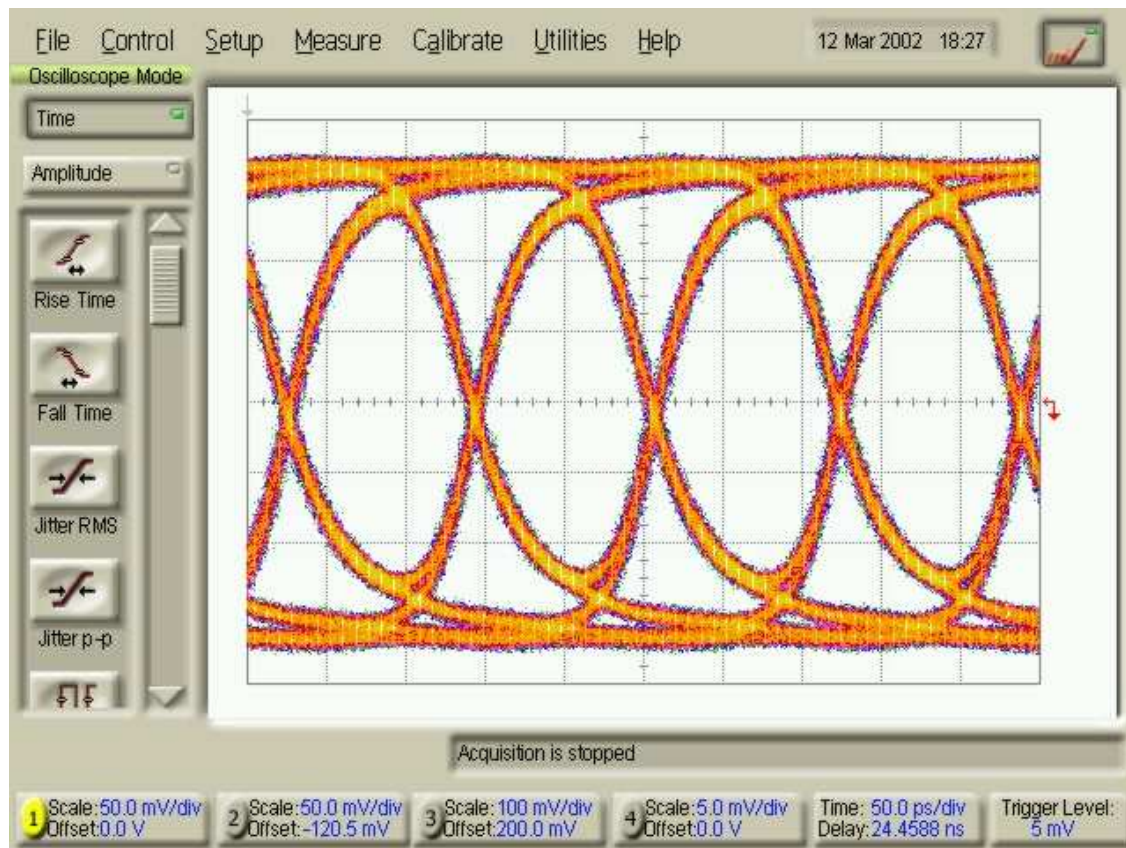


Abbildung 3.28: Das Datenauge der Sendedaten

## Die neuartige Empfängerstruktur mit Taktrückgewinnung

## 4 Ein asynchroner Empfänger

### 4.1 Klassifizierung von Empfängeranordnungen

In der Literatur finden sich mannigfaltige Ansätze zum Empfang serieller Daten. Sie lassen sich in zwei Gruppen unterteilen, die im Folgenden anhand für das Verständnis der hier neu vorgestellten Empfängeranordnung wichtiger Beispiele erläutert werden. Genaugenommen ergibt sich die Teilung in zwei Gruppen eher aus der Existenz einer großen Gruppe, den Empfängern mit Phasenregelkreis, und allen sonstigen Entwürfen, die sich von den vorgenannten Ansätzen eben dadurch unterscheiden, daß sie keinen solchen Phasenregelkreis bilden. Sie werden deshalb auch als „Open-Loop“-Architekturen bezeichnet.

Beinahe allen Ansätzen ist der Versuch gemein, den nicht übertragenen Takt der empfangenen Daten aus den Daten zurückzugewinnen, um die so erhaltene Information über die Positionierung der einzelnen Bits im Datenstrom zur Abtastung der Daten zu nutzen. Diese soll nach Möglichkeit genau in der Mitte des Bits erfolgen, so daß beidseitig ein maximaler Abstand zu den durch Jitter beeinträchtigten Flanken der Bits eingehalten wird. Denn je größer der Abstand des Abtastpunktes zu den Flanken, desto geringer ist die Bitfehlerrate des gesamten Systems.

Um die Taktrückgewinnung zu vereinfachen, werden häufig verschiedene Kodierungen der Daten angewendet. So gibt es Codes, die dafür sorgen, daß das Spektrum der übertragenen Daten eine starke Komponente bei der Datenrate aufweist. Dieser Ton kann durch geeignete Filterung aus dem Spektrum gelöst werden und dient als rückgewonnener Takt. Die Codes benötigen allerdings mehr Bandbreite zur Datenübertragung, was sie für Hochgeschwindigkeitsdatenübertragung unbrauchbar macht<sup>1</sup>. Als extremes Beispiel sei der Manchester-Code [9] angeführt, dessen spektraler Bandbreitenbedarf doppelt so hoch ist wie das für die gegebene Datenrate theoretische Minimum. Aus diesem Grund wird für die Datenübertragung gemäß den optischen Standards auf eine Kodierung verzichtet. Die Daten werden im NRZ-Code übertragen, also als einfache Folge von gleichlangen logischen Nullen und Einsen.

---

<sup>1</sup>Bei der optischen Datenübertragung wirkt sich nicht das eigentliche Übertragungsmedium, die Glasfaser, bandweitenbegrenzend aus, sondern die empfangen- und senderseitigen elektrischen Schnittstellen.

### 4.1.1 Open-Loop Architekturen

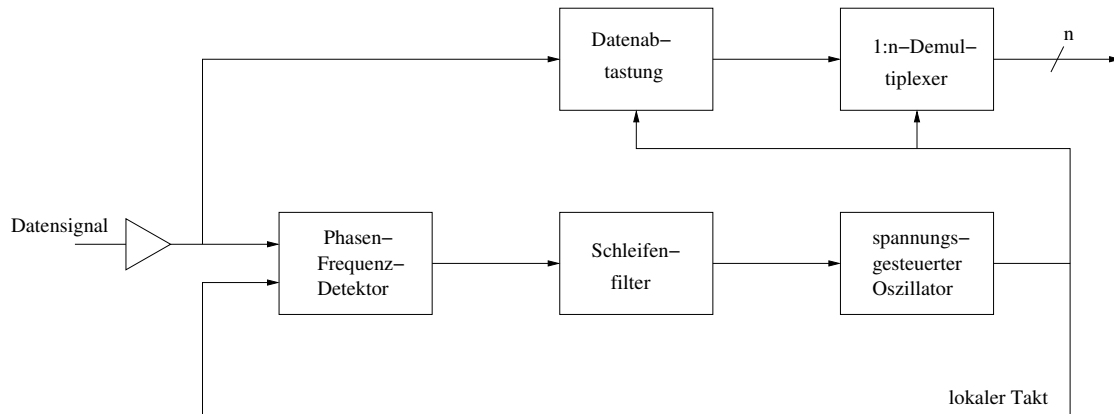
Da die NRZ-Kodierung keinen starken Ton an der Frequenz der Datenrate aufweist, versagen alle Methoden, die allein auf lineare Filterung zurückgreifen. Es müssen deshalb nichtlineare Operationen durchgeführt werden, die darauf abzielen, einen solchen Ton zu erzeugen. Der wohl älteste und bekannteste Ansatz erreicht dies durch Differenzierung und Gleichrichtung des Datensignales. Die Differenzierung erzeugt ein Signal, das einen kurzen Ausschlag beim Auftreten einer Flanke im Datensignal aufweist. Die Polarisierung dieses kurzen Ausschlages ist davon abhängig, ob es sich um eine steigende oder fallende Flanke handelt. Erst die Gleichrichtung dieses Signales bewirkt jedoch eine ausreichend starke Spektralkomponente bei der Trägerfrequenz. Dieser Ton muß durch Filter extrahiert und anschließend verstärkt werden.

Da die Filterung sehr engbandig durchgeführt werden muß, kann sie praktisch nicht durch Schaltungen bewältigt werden, die sich auf integrierten Schaltkreisen realisieren lassen, weshalb dieser Ansatz für den hier vorgesehenen Zweck ungeeignet ist. Auch allgemein kann festgestellt werden, daß die Gruppe der Open-Loop-Architekturen für die verfolgte Aufgabenstellung ohne Bedeutung ist, weshalb sich die Darstellung weiterer Beispiele erübrigt.

### 4.1.2 Phasenregelkreise

Die Idee eines Phasenregelkreises als Empfänger ist, mittels eines lokalen Oszillators das Datentaktsignal nachzubilden und zur Abtastung des empfangenen Datensignales zu nutzen. Dazu wird das lokale Taktsignal mit dem Datensignal verglichen und der Oszillator, wenn eine Abweichung der Phasen der beiden Signale festgestellt wird, derart nachgeregelt, daß der Phasenunterschied ausgeglichen wird. Die Oszillatorphase wird auf diese Weise an die des Datensignales gekoppelt, was den englischen Namen „phase-locked loop“, kurz PLL, erklärt. Da Frequenz und Phase eines Signales linear voneinander abhängen, besteht automatisch auch Frequenzgleichheit zwischen den beiden phasengekoppelten Signalen.

Auf diese Weise ergibt sich eine Regelschleife, die durch ein geeignetes Schleifenfilter stabilisiert werden muß. Gewöhnlich besitzen PLLs eine Regelcharakteristik zweiter Ordnung, so daß ihre Stabilität auch bei den typischen Parameterschwankungen im Fertigungsprozeß garantiert werden kann. Das Schleifenfilter ist ein Tiefpaßfilter erster Ordnung und wird meist als sog. „Charge-Pump“ ausgeführt. Eine Charge-Pump besteht aus einem Kondensator und jeweils einer vom Phasendetektor gesteuerten Stromquelle zum Auf- und Entladen des Kondensators, der auf diese Weise eine Mittelwertabschätzung der letzten Regelvorgänge bewirkt, was einer Tiefpaßfilterung entspricht.



**Abbildung 4.1:** Blockdiagramm eines typischen als PLL aufgebauten Empfängers

Abbildung 4.1 zeigt das Blockdiagramm einer typischen Empfängeranordnung, die auf einer PLL zur Taktrückgewinnung basiert. Im unteren Teil der Abbildung ist der Phasenregelkreis, bestehend aus einem Phasen/Frequenz-Detektor (kurz PFD), dem Schleifenfilter und dem regelbaren Oszillator (VCO<sup>2</sup>), dargestellt. Im oberen Teil sind die Blöcke für die Datenabtastung – im einfachsten Fall ein einzelnes Flip-Flop – und den Demultiplexer dargestellt. In einigen Ansätzen wird die Funktion der Datenabtastung innerhalb des PFDs automatisch mitausgeführt.

In der Anordnung verdient insbesondere der Phasen/Frequenz-Detektor Aufmerksamkeit. Im Falle eines Datenempfängers wird ein periodisches Taktsignal mit einem zufälligen Datensignal verglichen. Der PFD darf aber nur dann ein Steuersignal für den VCO erzeugen, wenn im Datensignal eine Flanke aufgetreten ist, da allein der Vergleich der Lage zweier Flanken zueinander eine Aussage über die Phasendifferenz zwischen den beiden Signalen zu liefern vermag.

Anstelle eines Phasen/Frequenz-Detektors kann auch ein einfacher Phasendetektor zum Einsatz kommen. Dieser kann im Gegensatz zum PFD keine Information über das Verhältnis der Frequenzen beider Signale zueinander liefern. Liegen die Frequenzen zu weit auseinander, so kann die Regelschleife nicht einrasten und der VCO läuft frei. Derartige Anordnungen sehen gewöhnlich einen Hilfsmechanismus vor, der zuerst die VCO-Frequenz in die Nähe der zu erwartenden Datenrate bringt, so daß anschließend die PLL einrasten kann. Hierfür kann beispielsweise eine Taktreferenz benutzt werden. Generell sind jedoch Lösungen mit Phasen/Frequenz-Detektor vorzuziehen, weil sie einen geringe-

<sup>2</sup>Allgemein werden regelbare Oszillatoren als „VCOs“ bezeichnet, was für „voltage-controlled oscillators“ steht, ohne daß Rücksicht darauf genommen würde, ob sie durch eine Regelspannung oder einen Regelstrom gesteuert werden.



ren Gesamtaufwand bedeuten. So sind insbesondere bei Einsatz einer Referenzfrequenz mindestens ein weiterer Gehäusepin sowie Verdrahtungsaufwand außerhalb des Chips notwendig, die den Einsatz allein eines Phasendetektors unattraktiv machen.

Der VCO muß zwei widersprüchliche Anforderungen erfüllen: einerseits muß er einen weiten Frequenzbereich aufweisen, damit die erforderliche Frequenz trotz aller Parameterschwankungen innerhalb seines realen Frequenzbereiches liegt, andererseits soll die Empfindlichkeit des Steuereinganges nicht zu hoch ausfallen, damit Rauschen sowie systembedingte Schwankungen<sup>3</sup> auf diesem Eingang keinen zu großen Störeinfluß auf den Oszillator haben.

Diese gegensätzlichen Ansprüche können erfüllt werden, indem man den Oszillator mit getrennten Eingängen für Grob- und Feinjustierung seiner Frequenz versieht. Die Grobjustierung wird dann nur bei Inbetriebnahme eingesetzt, um die Oszillatorfrequenz an die des Datensignales anzunähern. Sobald sie erreicht ist, wird der Wert auf dem Eingang der Grobjustierung festgehalten, so daß nur noch die weniger empfindliche Feinjustierung Einfluß auf den Oszillator ausübt.

Der Regelwert muß dann allerdings als digitaler Wert bereitgestellt werden, weil ein analoger Wert nicht gespeichert werden kann. Da der Einsatz eines AD-Umsetzers zu aufwendig wäre, werden die Teile der PLL, die zur Grobjustierung dienen, digital ausgelegt. Ist die Grobjustierung erreicht, wird die PLL zur Feinabstimmung aktiviert. Genau genommen zerfällt der Phasenregelkreis somit in zwei überlagerte PLLs, von denen jedoch immer nur eine aktiv ist. Deshalb können auch unterschiedliche Regelcharakteristiken für die Grobjustage bzw. Frequenzdetektion und den Normalbetrieb vorteilhaft eingesetzt werden.

## 4.2 Zur Kodierung der Daten

Der Phasendetektor des Regelkreises kann nur anhand diskreter Ereignisse, nämlich durch Vergleich der Flankenlage des empfangenen Datensignales mit dem lokalen Takt, eine Regelspannung erzeugen. Da es sich beim empfangenen Datensignal aber um eine zufällige Bitfolge handelt, können für lange Zeit Wechsel im Datenstrom ausbleiben.

Um die dadurch entstehende Problematik zu verstehen, betrachte man ein ideales System, das mit der exakt gleichen Frequenz schwingt wie der Takt der empfangenen Daten, aber einen geringen Phasenabstand aufweist. Der lokale Takt tastet dann die empfangenen Daten ein wenig neben deren Mitte ab. Im Datenstrom erscheint jetzt eine einzelne Flanke, der Phasendetektor detektiert die Phasendifferenz und erzeugt eine entsprechende Regelspannung, um die Phasen von Oszillator und Daten anzugleichen. Die Regelspannung

---

<sup>3</sup> Als Beispiel für eine solche systembedingte Störung siehe den Abschnitt über binäre Phasendetektoren auf Seite 78

erhöht oder senkt, je nach Phasenbezug, die Oszillatorfrequenz, was die Phasendifferenz linear mit der Zeit verringert.

Folgt aber auf diese einzelne Flanke sehr lange Zeit keine weitere, wird in Ermangelung einer weiteren Vergleichsmöglichkeit die Oszillatorfrequenz weiterhin von der der Daten abweichen, weshalb sich ein stetig wachsender, dem ursprünglichen entgegengesetzter Phasenfehler ergibt. Dieser wird nach einer endlichen Zeit größer als eine Bitlänge, so daß sich die Anzahl der gleichartigen hintereinander empfangenen Bits nicht mehr richtig bestimmt läßt. Da die Grenzen der einzelnen Bytes im Datenstrom nicht mehr mit den erwarteten übereinstimmen, ist die Übertragung in hohem Maße gestört.

Offensichtlich existiert aber selbst für ein ideales System eine maximale Anzahl aufeinanderfolgender gleichartiger Bits, die garantiert fehlerfrei empfangen werden kann. Diese Anzahl von Bits könnte theoretisch aus der Regelcharakteristik der PLL berechnet werden. Dieses Unterfangen ist aber aus verschiedenen Gründen sinnlos, kommen doch in der Realität viele Effekte hinzu, die die reale Anzahl weit unter die theoretische Grenze senken. Leckströme und kapazitive Einkopplungen verändern die Spannung auf dem Kondensator des Schleifenfilters mit der Zeit, so daß der Oszillator in jeder realen Umgebung nur eine endliche Zeit freilaufend einen festen Phasenbezug zum Datensignal behalten kann. Des weiteren ist sowohl das empfangene Datensignal als auch der lokal erzeugte Takt durch Jitter beeinträchtigt. Dieses Phasenrauschen akkumuliert sich in der PLL, so daß allein der fortlaufende Vergleich der beiden Signale die PLL stabil im eingerasteten Zustand halten kann.

Die Takt-/Datenrückgewinnung kann also nur dann funktionieren, wenn eine Mindestrate von Wechseln im Datenstrom garantiert wird. Um dies zu erreichen, werden die Daten einer Kodierung unterzogen. Der Standard zum 10 GBit/s OPTICALETHERNET greift auf eine 64B/66B-Kodierung [10] zurück, da sie einen minimalen Verlust an Übertragungskapazität bedeutet. Diese Kodierung erweitert einen Satz von 64 Bits um zwei Bits, die so polarisiert werden, daß Flanken auftreten müssen. Durch diese Kodierung bedingt können höchstens 65 Bits aufeinander folgen, ohne daß eine Flanke auftritt. Der Aufwand an Bandweite für die Kodierung des Datenstromes ist gerade einmal  $\frac{33}{32} \approx 3\%$ .

Ein weiteres Problem wird durch die 64B/66B-Kodierung gelöst. So könnten die Daten zufälligerweise aus einer Serie von Paaren logischer Einsen und Nullen bestehen. Ist der Ziehbereich des Oszillators groß genug, könnte das System zumindest kurzzeitig auf der halben Datenrate einrasten und somit die Bitpaare als einzelne Bits wahrnehmen. Die Kodierung stellt deshalb das Auftreten eines einzelnen Bits mit einer seinen unmittelbaren Nachbarn entgegengesetzten Polarisierung sicher. Dadurch kann das System nicht auf eine niedrigere Frequenz einrasten.

Im SDH/SONET-Standard ist keine Kanalkodierung berücksichtigt. Die Datenstruktur sieht dort für jedes Datenpaket einen sog. *header* vor, der Informationen zu seiner Übertragung und Verwaltung enthält[1]. Diese Daten enthalten aufgrund ihrer Natur immer

Wechsel zwischen Einsen und Nullen. Nichtsdestotrotz kann der Abstand zwischen zwei Flanken sehr groß werden.

## 4.3 Ausgewählte Beispiele aus der Literatur

### 4.3.1 Digitale Schleifenfilter

Neben dem Wirken der Bandweite des Phasenregelkreises auf die vorrangigen Rauschquellen gibt es noch andere wichtige Aspekte, die bei der Wahl der Regelcharakteristik eine Rolle spielen. So muß insbesondere dem Jitter–Peaking Beachtung geschenkt werden (siehe hierzu den Abschnitt auf Seite 17). Es gibt eine hohe Genauigkeit der Komponenten vor, die die Übertragungsfunktion des Schleifenfilters bestimmen. Bedenkt man zusätzlich, daß die Werte für den Kondensator des Tiefpasses durch die niedrige Eckfrequenz sehr groß werden, verwundert es nicht, daß trotz der damit verbundenen Nachteile meist externe Komponenten zum Einsatz kommen.

Es wäre wünschenswert, das Schleifenfilter digital auslegen zu können, weil so die externen Komponenten entfielen und die Filterfunktion parameterunabhängig exakt implementiert werden könnte. Bedingung hierfür wäre eine Überabtastung des Datensignales, die neben den eigentlichen Daten auch eine Information über die Phasenlage des Signales zur Verfügung stellen würde.

Die digitale Umsetzung der Filterung ist jedoch aus zwei Gründen in keinem bekannten Ansatz versucht worden. Es wäre unter dem Gesichtspunkt der konsumierten elektrischen Leistung und des erhöhten Flächenbedarfes sowohl zu aufwendig, die digitale Filterung bei der hohen Datengeschwindigkeit durchzuführen, als auch zusätzliche Demultiplexer für das Herabsetzen der Rate jener Samples, die die Phaseninformation liefern, vorzusehen, damit die Filterung bei einer niedrigeren Geschwindigkeit vorgenommen werden kann.

In der Literatur finden sich zwei Ansätze, die beide genannte Problempunkte zu umgehen vermögen. Dort wird mit einem Takt gearbeitet, der zwar phasengekoppelt zum Datentakt ist, aber eine um ein beliebiges ganzzahliges Verhältnis  $N$  herabgesetzte Frequenz erlaubt. Dadurch würde eine Extraktion der Phaseninformation direkt bei der herabgesetzten Taktgeschwindigkeit möglich, so daß die digitale Filterung einfach umzusetzen wäre.

Abbildung 4.2 stellt den ersten Ansatz gemäß [11] dar. Er erreicht die Parallelisierung durch direkte Abtastung des Datensignales vermittels mehrerer zeitlich gegeneinander verschobener Takte. Dafür benötigt er einen Oszillator, der  $n = N \cdot O$  Oszillatorphasen einer Frequenz von  $\frac{f_d}{N}$  besitzt;  $O$  bedeutet hierbei den Faktor der Überabtastung des Datensignales. Das Datensignal wird direkt an die Eingänge von  $N \cdot O$  Flip-Flops, die bei

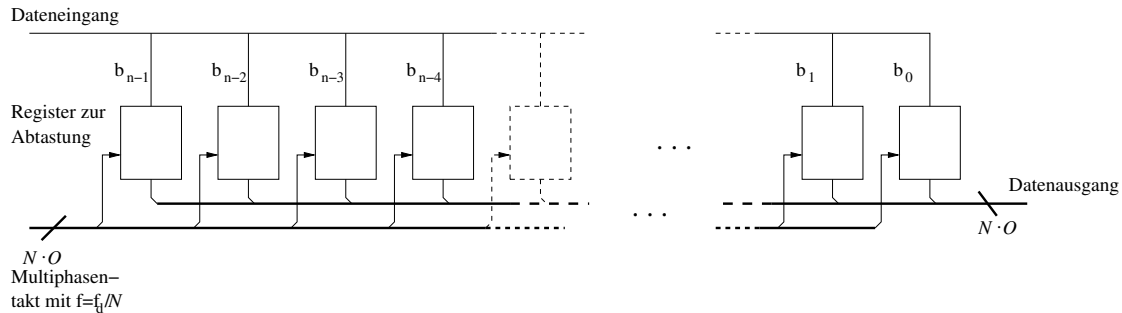


Abbildung 4.2: Parallelisierte Abtastung des Datensignal

der vollen Geschwindigkeit der Datenrate arbeiten müssen, angeschlossen. Indem jedes Flip-Flop, versetzt um eine Zeit, die um  $O$  kleiner als eine Bitlänge ist, das Datensignal abtastet, werden Abtastung und Parallelisierung in einem Schritt vorgenommen.

Der größte Nachteil dieser Schaltung ist die sehr hohe kapazitive Belastung des Eingangsverstärkers durch die vielen parallelgeschalteten Flip-Flop-Eingänge. Der Eingangsverstärker wird deshalb zusätzliche Stufen aufweisen müssen, was die Bitfehlerrate aufgrund des innerhalb des Verstärkers eingestreuten Rauschens und anderer Störeffekte verschlechtert. Zudem ist eine hohe Zahl von leistungshungrigen schnellen Flip-Flops vonnöten.

Eine niedrigere Eingangskapazität bietet die in [12] vorgestellte Schaltung. Abbildung 4.3 zeigt ein Blockdiagramm. Kern der Idee ist die Kette von Verzögerungsgliedern, durch die das Datensignal geleitet wird. Wenn jedes der  $n - 1$  Verzögerungsglieder eine Verzögerung von genau einer Bitlänge<sup>4</sup> besitzt, dann erscheinen  $n$  aufeinanderfolgende Bits wegen der unterschiedlich langen Wege durch die Kette gleichzeitig auf deren Ausgängen. Folglich können alle diese Ausgänge gleichzeitig mit einem um den Faktor  $n$  herabgesetzten Takt abgetastet werden; es wird im Gegensatz zur vorhergehenden Schaltung kein Multiphasentakt benötigt. Die Funktion des Demultiplexers ist damit auf den ersten Blick bestehend einfach verwirklicht.

In beiden Abbildungen nicht aufgeführt sind die Blöcke der PLL, die auch hier benötigt wird, um den zur Abtastung reproduzierten Takt an den Takt des Datensignales zu koppeln. Darüber hinaus ist ein für integrierte Schaltungen zuverlässiger Mechanismus unerlässlich, der die Länge der Verzögerungen der Kette dergestalt kontrolliert, daß Parameterschwankungen automatisch ausgeglichen werden.

Soll die PLL allein mit den bereits gedemultiplexten Daten funktionieren, müssen die Verzögerungen der Kette deutlich kleiner sein als eine Bitlänge, um die notwendige Über-

<sup>4</sup>oder allgemeiner  $O^{-1}$  Bitlängen im Falle von  $O$ -facher Überabtastung

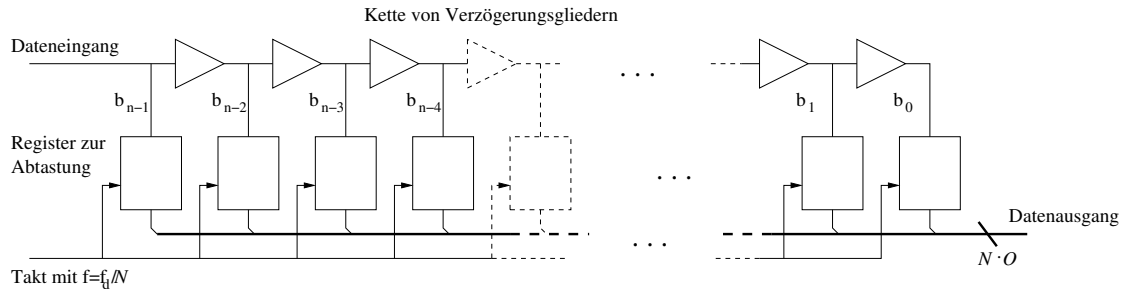


Abbildung 4.3: Verzögerungskette als Demultiplexer

abtastung der Daten zu verwirklichen. Im Falle eines 10 GBit/s-Empfängers lägen diese Verzögerungen bei maximal nur 33 ps oder gar 25 ps. Sind derart kurze Verzögerungen ohnehin schon sehr schwierig zu realisieren, erscheint es mit aktuellen CMOS-Prozessen noch unmöglich, ihre Dauer zusätzlich ausreichend genau kontrollierbar zu gestalten.

Hinzu kommt, daß sich Abweichungen von der gewünschten Verzögerung durch die Kette hindurch summieren und bei einem eventuell vorhandenen systematischen Fehler eine Abtastung des letzten Bits ( $b_0$ ) der Kette neben dem optimalen Abtastpunkt in seiner Mitte immer wahrscheinlicher wird. Außerdem wird in jedem Verzögerungsglied das Signal durch Rauschen insbesondere aus den Versorgungsquellen gestört. Dies bedingt zusätzlichen Jitter, der für das Bit  $b_0$  der Kette maximal wird. Da auch die Unsicherheit des Abtastzeitpunktes für die letzten Bits der Kette am größten ist, sinkt das Signal/Rausch-Verhältnis mit der Bitposition.

Ein weiterer Nachteil der Anordnung ist das sogenannte *Pattern noise*, das in der Kette auftreten kann: kapazitive Kopplungen zwischen den einzelnen Knoten der Verzögerungskette können musterabhängige Störungen einzelner Datenbits bedingen. Wegen all dieser Effekte wird die Bitfehlerrate eine Funktion der Position der einzelnen Bits im Datenstrom.

Die nützliche Eigenschaft der herabgesetzten Datenrate noch vor der ersten Abtastung und der deshalb entsprechend niedrigere lokale Takt sind jedoch in Hinsicht auf die in Abschnitt 3.2.4 erläuterten Vorteile der in dieser Arbeit eingeführten, ebenfalls mit herabgesetztem Takt arbeitenden Multiplexerarchitektur interessant. Die im Abschnitt 4.4 vorgestellte Empfängeranordnung übernimmt die Vorteile, wenngleich mit der praktisch eher unbedeutenden Einschränkung auf Teilverhältnisse  $N = 2^n$ . Im Gegenzug vermeidet sie jedoch die Nachteile der älteren Ansätze.

### 4.3.2 Phasendetektoren

Die Übertragungsfunktion eines Phasendetektors wird gewöhnlich durch eine lineare Übertragung des Phasenfehlers, der zuvor durch eine Differenzbildung zwischen der Oszillatorphase und der des Datensignales bestimmt wurde, dargestellt. Sie wird deshalb durch einen Proportionalitätsfaktor  $K_D$  angegeben, was jedoch eine Vereinfachung darstellt und nur für den Modellfall gilt. So weisen die sogenannten *linearen* Phasendetektoren gewöhnlich nur innerhalb eines bestimmten Bereiches um eine Phasendifferenz von 0 einen linearen Zusammenhang zwischen Phasendifferenz und Ausgangsspannung auf.

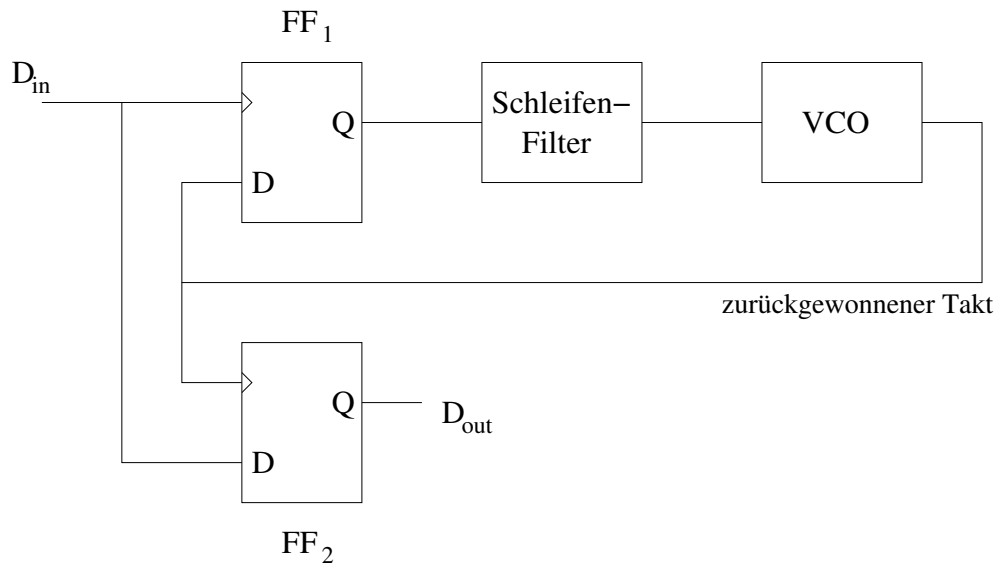
Trotzdem kann die Übertragungsfunktion ausreichend genau durch eine Linearisierung als Gerade um den Nullpunkt dargestellt werden. Diese Linearisierung ist, ebenso wie die angenommene lineare Abhängigkeit der Oszillatorfrequenz von der Eingangsspannung des VCOs, für die Bestimmung der Übertragungsfunktion der PLL unumgänglich. Andernfalls ergäbe sich für das Verhalten des Regelkreises eine mathematisch nicht mehr lösbare nichtlineare Differentialgleichung.

Aufgrund der vorgenommenen Vereinfachungen ist das dynamische Verhalten des Phasenregelkreises allerdings nur für den eingerasteten Zustand bestimmbar, in dem die Linearisierungen der Übertragungsfunktionen von Phasendetektor und Oszillator minimale Fehler aufweisen. Die Erfahrung zeigt jedoch, daß die Stabilität auch bei noch größeren Vereinfachungen gegeben ist.

Der Einsatz eines digitalen Schleifenfilters setzt die digitale Umsetzung des Phasenfehlers voraus. Selbstredend ist die Kombination eines gewöhnlichen linearen Phasendetektors mit einem AD-Umsetzer für die angestrebten Geschwindigkeiten ausgeschlossen. Der Phasendetektor selbst muß so beschaffen sein, daß er, um die erwünschte proportionale Übertragung des Phasenfehlers in eine Regelspannung möglichst gut zu erreichen, ihn in Zeitschritten auflöst. Die Übertragungsfunktion eines linearen Phasendetektors wird dann durch eine treppenförmige quantisierte Übertragungscharakteristik angenähert, weshalb die Modellierung des Phasendetektors durch einen Proportionalitätsfaktor eine noch größere Idealisierung bedeutet.

Bei einer Bitlänge von 100 ps wäre die Diskretisierung des Phasenfehlers äußerst schwierig; es dürfte kaum eine höhere Auflösung als ein Drittel einer Bitlänge möglich sein. Deshalb beschränken sich viele der bekannten Ansätze für quantisierende Phasendetektoren darauf, lediglich das Vorzeichen des Phasenfehlers zu bestimmen und eine feste Ausgangsspannung mit einer Polarisierung entsprechend der des vorhandenen Fehlers zu erzeugen. Diese Gruppe von Phasendetektoren wird *binäre* Phasendetektoren genannt.

Binäre Phasendetektoren bilden den Phasenfehler sehr ungenau ab, so daß erst die Mittelung einer Vielzahl von Entscheidungen einen brauchbaren Regelwert für den Oszillator ergibt. Da der Phasenregelkreis jedoch ein System zweiter Ordnung darstellt, kann das Schleifenfilter nur eine beschränkte Glättung der Regelspannung erreichen.



**Abbildung 4.4:** Takt-/Datenrückgewinnung nach dem „Bang-bang“-Prinzip

Ein weiterer Nachteil der binären Phasendetektoren besteht im fehlenden Nullpunkt ihrer Übertragungscharakteristik, dessenwegen das System niemals zur Ruhe kommen kann, da niemals ein Phasenfehler von Null festgestellt wird. Die Folge ist ein unter Umständen recht stark ausgeprägtes periodisches Signal auf dem Regeleingang des Oszillators, das dessen Frequenz um die Sollfrequenz herumwandern lässt. Man bezeichnet dieses Phänomen als (engl.) *wobble*. Es bewirkt einen deterministischen Jitter des rückgewonnenen Taktes und damit eine erhöhte Bitfehlerrate.

Eine einfache Realisierung eines binären Phasendetektors ist die sog. „Bang-bang<sup>5</sup>“-Schaltung, die den Phasendetektor durch ein einzelnes D-Flip-Flop verwirklicht [13]. Ein Blockdiagramm einer Takt-/Datenrückgewinnung nach dem „Bang-bang“-Prinzip wird in Abbildung 4.4 dargestellt.

Das D-Flip-Flop ist auf ungewöhnliche Art verschaltet: der Dateneingang wird mit dem lokalen Takt verbunden, der Takteingang dagegen mit dem empfangenen Datensignal. Dadurch aktualisiert der Phasendetektor seinen Ausgangswert gemäß der Bedingung für die Auswertung eines zufälligen Datenstromes immer dann, wenn im Datenstrom eine Flanke auftritt.

Der Bang-bang-Phasendetektor tastet den lokalen Takt ab und liefert so die Information, ob die Flanke im Datensignal vor oder hinter der entsprechenden Flanke des Taktsignales

<sup>5</sup>„Bang-bang“ beschreibt lautmalerisch die ruckartig um den Ruhepunkt hin- und herschwankende Phasendifferenz

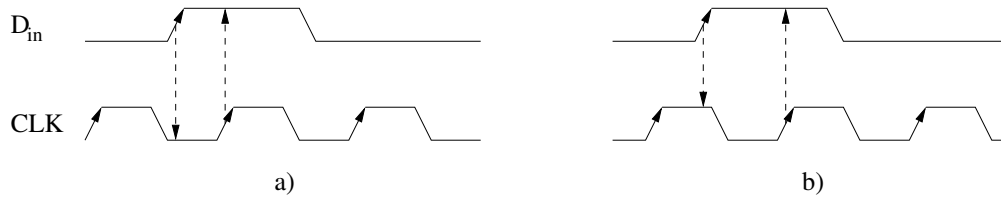


Abbildung 4.5: Beispielsignale für die Fälle a) „early“ und b) „late“

lag. Tastet das Flip-Flop den Takt in seiner negativen Halbphase ab, gibt es eine Null aus, der Takt kommt zu früh („early“). Tastet es den Takt hingegen in seiner positiven Halbphase ab, gibt es eine logische Eins aus, der Takt ist zu spät („late“). Auf diese Weise regelt der Bang-bang-Phasendetektor die VCO-Phase entsprechend dem detektierten Phasenfehler nach. Die Abbildung 4.5 zeigt jeweils ein Beispielsignal für beide Fälle.

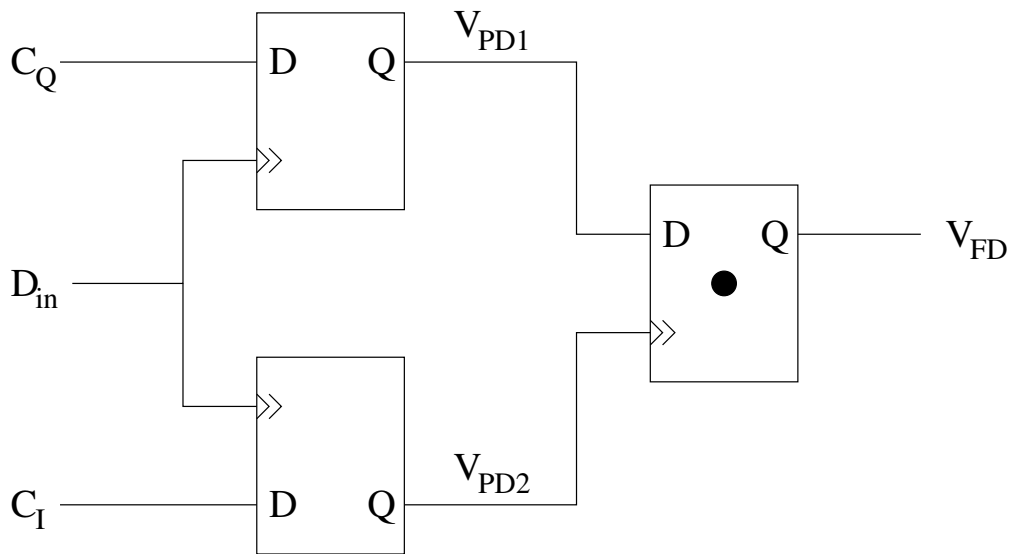
Die erläuterte Schaltung wertet nur jede zweite Flanke, nämlich alle steigenden, im Datenstrom aus. Dadurch wird die Regelspannung des VCOs nur halb so häufig korrigiert, wie eigentlich möglich. Der mittlere Phasenfehler kann halbiert werden, wenn die Schaltung derartig erweitert wird, daß sowohl steigende als auch fallende Flanken des Datensignales ausgewertet werden. Dies geschieht mit einem sog. double-edge-triggered Flip-Flop (DETFF).

Neben den bereits erwähnten Nachteilen von binären Phasendetektoren weist die Bang-bang-Schaltung noch einen weiteren auf. Jede Flip-Flop-Schaltung zeigt Verzögerungen für die Wege vom Takteingang zum Datenausgang und vom Dateneingang zum Datenausgang, die sich unter anderem wegen der Setup-Zeit der Dateneingänge voneinander unterscheiden. Die Differenz  $\Delta t_{FF}$  zwischen diesen beiden Verzögerungen bedingt einen systematischen Fehler bei der Bestimmung des Verhältnisses zwischen den Phasen von Takt und Datensignal. Dieser systematische Fehler schlägt sich in einer Abtastung der Daten neben dem optimalen Punkt in der Mitte des Datums nieder. Analysiert man die zeitlichen Gegebenheiten in der Regelschleife, dann erkennt man, daß sich das  $\Delta t_{FF}$  der beiden Flip-Flops  $FF_1$  und  $FF_2$  addiert. Der resultierende Fehler kann für sehr hohe Datenraten in die Größenordnung der Bitlänge gelangen. Durch den systematischen Offset bei der Abtastung steigt die Bitfehlerrate in unerwünschtem Maße an.

### 4.3.3 Die Frequenzdetektion

Abbildung 4.6 zeigt eine Erweiterung des „Bang-bang“-Phasendetektors um einen Frequenzdetektor nach [14]. Genau genommen handelt es sich bei der Schaltung um eine Kombination von drei Phasendetektoren nach dem durch Einsatz eines DETFF erweiter-





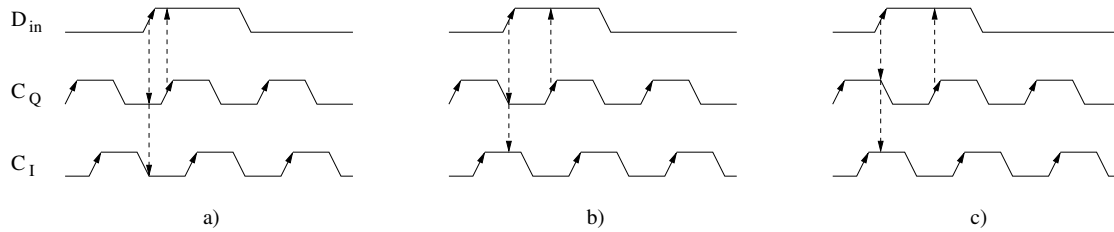
**Abbildung 4.6:** Erweiterung der „Bang-bang“-Schaltung auf einen Frequenzdetektor

ten „Bang-bang“-Prinzip mit den oben erläuterten Vorteilen. Zwei dieser Phasendetektoren vergleichen die Phasenlage des Datensignales mit dem lokalen Oszillator, der dritte vergleicht die Phasenlage der beiden Ausgangssignale der vorgeschalteten Phasendetektoren. Der letztgenannte weist eine Besonderheit auf, die durch einen schwarzen gefüllten Punkt im Blockschaltbild gekennzeichnet ist. Diese wird später erklärt.

Um die Frequenzdetektion zu ermöglichen, muß der lokale Oszillator zwei um eine feste Phasendifferenz kleiner als  $180^\circ$  versetzte Taktsignale anbieten; gewöhnlich greift man auf ein sog. Quadratur-Signal zurück, das einen Phasenversatz von  $90^\circ$  zum Haupttakt aufweist. Die beiden ersten Phasendetektoren vergleichen die Lage des Datensignales mit den zwei zeitlich versetzten Taktsignalen wie für den „Bang-bang“-Phasendetektor erläutert.

Die Ausgangssignale beider Phasendetektoren könnten dergestalt kombiniert werden, daß sich die beiden „early“- und „late“-Signale addieren<sup>6</sup>. Auf diese Weise heben sich die beiden Signale auf, wenn die Flanke des Datensignales zwischen zwei gleichartig orientierte Flanken der beiden Takte fällt, also in Phase mit dem Oszillator ist (siehe Teilabbildung 4.7 b). Die binäre Charakteristik eines einzelnen Phasendetektors nach dem „Bang-bang“-Prinzip wird dergestalt auf eine ternäre erweitert. Die Anordnung hat nun einen Ruhezustand, was die systembedingte Störung des Oszillatoreinganges in der Regelschleife senkt. Gleichwohl erfolgt bei Nutzung der beiden Phasendetektorsignale die Datenab-

<sup>6</sup>Diese einfache Verbesserung findet sich meines Wissens nicht in der Literatur.



**Abbildung 4.7:** Beispielsignale für die Fälle a) „early“ b) „in Phase“ und c) „late“

tastung mit einem systematischen Fehler von  $45^\circ$  neben dem optimalen Abtastzeitpunkt, was sich negativ auf die Bitfehlerrate auswirkt. Die Abtastung vor dem Mittelpunkt des Datenbits im Falle b) „in Phase“ ist deutlich zu erkennen. Die höhere zeitliche Auflösung der Phasendifferenz durch Einsatz mehrerer Oszillatorphasen ist jedoch eine interessante Idee, die sich auch in der in dieser Arbeit vorgestellten Empfängerarchitektur widerspiegeln wird.

Die Beispielsignale in Abbildung 4.8 dienen zur Erläuterung der Frequenzdetektion. In der Teilabbildung a) ist der Takt gegenüber den Daten zu langsam, in Teilabbildung b) zu schnell. Zur übersichtlicheren Darstellung wurden der Takt als zeitliche Referenz benutzt und die Daten entsprechend in der Länge verändert, obwohl in Wirklichkeit die Taktperiode die variable Größe ausmacht. In beiden Beispielen weicht der Takt um etwa 6% von der Datenrate ab. Bei einem Half-rate-System mit 5 GHz Takt würde das bereits einen Abstimmbereich des Oszillators von 300 MHz in jede Richtung bedeuten.

In den Teilabbildungen sind die Ausgangssignale der beiden „Bang-bang“-Phasendetektoren als Serien von logischen Werten dargestellt. Darunter wird die Bedeutung für die Phasenlage angegeben. Es sei in Erinnerung gerufen, daß in diesem Beispiel die Regelschleife noch nicht auf das Datensignal eingerastet ist und deshalb die Phaseninformation allein noch nicht ausreicht, um den Takt den Daten folgen zu lassen. Dies zeigt sich an der wechselnden Information bezüglich der Phasenlage: im ersten Beispiel wechselt sie von „early“ über „in Phase“ nach „late“, im zweiten in Gegenrichtung von „late“ über „in Phase“ nach „early“.

Der Drehsinn der Phasenlage folgt unmittelbar aus der Polarität der Frequenzdifferenz zwischen Takt und Daten. Ist der Takt langsam (Abb. 4.8 a), dann wechselt der Phasenbezug von „early“ in Richtung „late“, weil der Takt immer weiter hinter den Daten zurückbleibt. Im umgekehrten Fall verfrühen sich die Taktflanken gegenüber den Datenbits zunehmend, weshalb der Phasenbezug von „late“ nach „early“ wechselt.

Könnten allerdings nur die Fälle „late“ und „early“ unterschieden werden, wäre der Drehsinn nicht bestimmbar, denn es gäbe nur die direkten Wechsel „early“ – „late“ bzw. „late“ – „early“, die jedoch für beide Drehsinne gleichermaßen auftreten. Deshalb ist der Vergleich

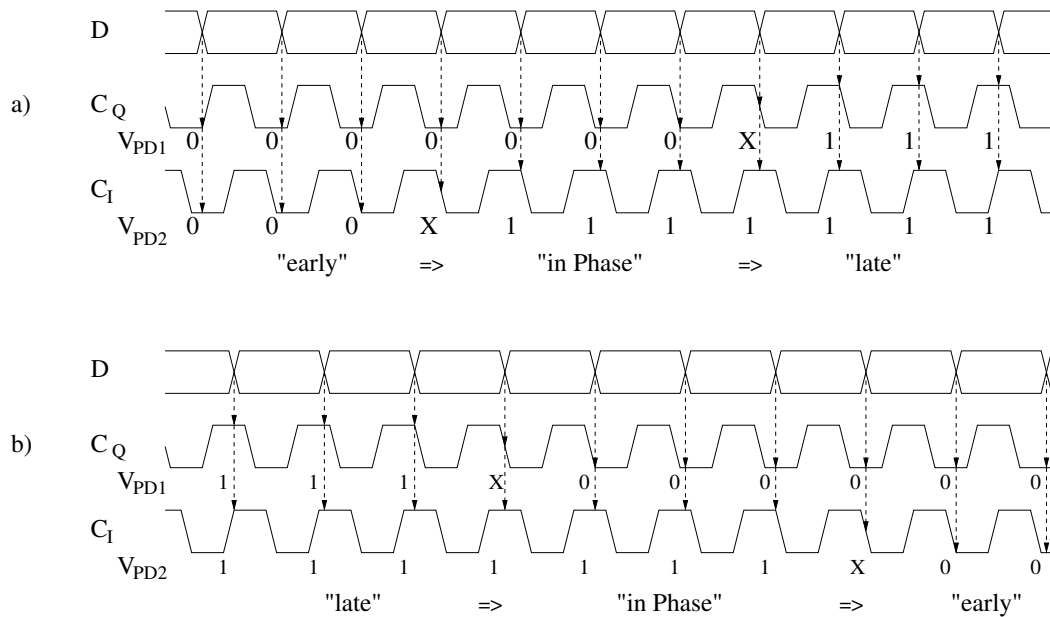


Abbildung 4.8: Beispielsignale zur Erläuterung der Frequenzdetektion

mit dem Quadraturtakt vermittelt des zweiten Phasendetektors unbedingt notwendig, soll eine Information über das Verhältnis der Frequenzen gewonnen werden.

Ein Vergleich der Ausgangssignale beider Phasendetektoren zeigt, daß sich das Verhältnis der Frequenzen aus der Phasenlage zwischen diesen beiden Signalen bestimmen läßt. Im ersten Beispiel wechselt das Signal  $V_{PD2}$  vor  $V_{PD1}$ , im zweiten Beispiel eilt das Signal  $V_{PD1}$  dem anderen voraus. Ob es sich um eine steigende oder fallende Flanke handelt, ist dabei unerheblich, denn für beide Flankenorientierungen gilt dieselbe Feststellung.

Die zufällige Natur des Datensignales außer Acht gelassen, liegt zwischen den beiden Ausgangssignalen der Phasendetektoren eine feste Phasendifferenz von  $90^\circ$  vor, die sich aus der Phasendifferenz der beiden für die Phasendetektion benutzten Taktsignale ergibt. Die Polarität dieser Phasendifferenz dagegen ist variabel und entspricht dem Vorzeichen der Frequenzdifferenz zwischen Takt und Daten.

Wird also ein dritter Phasendetektor benutzt, um die Phasenlage der Ausgangssignale der beiden ersten Phasendetektoren zu bestimmen, so zeigt dieser das Verhältnis der Frequenzen von Takt und Daten zueinander an. Bei Inbetriebnahme der Datenübertragungsstrecke kann die Frequenz des Oszillators somit schnell in die Nähe der Datenrate gebracht werden, bevor schließlich der Phasendetektor die dynamische Phasenkopplung zwischen Daten und Takt herstellt.

Als Besonderheit ist jedoch beim dritten Phasendetektor zu beachten, daß er im Gegensatz zu den ersten beiden zwei periodische Signale gleicher Momentanfrequenz vergleicht. Wäre er identisch mit den anderen, würde das an den Takteingang des DETFFs angeschlossene Signal das andere immer je einmal in dessen positiver und negativer Halbphase abtasten. Das Ausgangssignal des Frequenzdetektors würde zwischen Null und Eins wechseln und wäre unbrauchbar.

Aus den beiden Beispielsignalen ergeben sich vier mögliche Fälle:

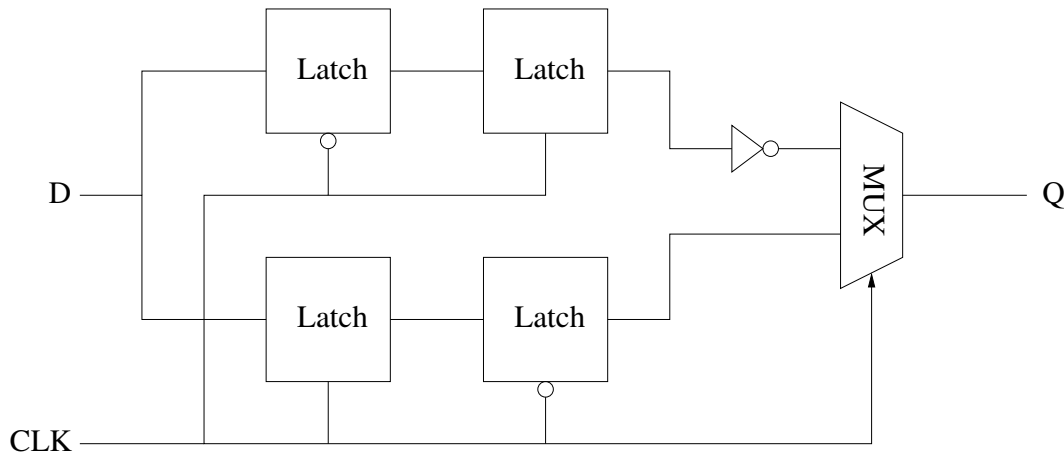
1.  $V_{PD2}$  wechselt von Null nach Eins,  $V_{PD1}$  ist Null (Beispiel a)
2.  $V_{PD2}$  wechselt von Eins nach Null,  $V_{PD1}$  ist Eins (Beispiel a, nicht in der Abbildung dargestellt)
3.  $V_{PD2}$  wechselt von Null nach Eins,  $V_{PD1}$  ist Null (Beispiel b, nicht in der Abbildung dargestellt)
4.  $V_{PD2}$  wechselt von Eins nach Null,  $V_{PD1}$  ist Eins (Beispiel b)

$V_{PD2}$  tastet jeweils  $V_{PD1}$  ab.

Die beiden ersten Fälle können nur auftreten, wenn die Frequenz des Taktes zu niedrig, die beiden letzteren, wenn sie zu hoch ist. Soll die zu niedrige Taktfrequenz durch eine logische Eins am Ausgang des Frequenzdetektors symbolisiert werden, zeigt sich, daß das Ausgangssignal für die Abtastung durch eine steigende Flanke (Fall 1) invertiert, für die Abtastung durch eine fallende Flanke (Fall 2) unverändert auf den Ausgang des Frequenzdetektors durchgeschaltet werden muß. Eine zu hohe Taktfrequenz wird durch eine logische Null am Ausgang dargestellt. Wiederum ergibt sich, daß für die Abtastung durch eine steigende Flanke (Fall 3) der Ausgangswert invertiert werden muß, für eine fallende dagegen nicht (Fall 4).

Ein DETFF wird typischerweise aus zwei gewöhnlichen flankengesteuerten Flip-Flops aufgebaut, von denen eines auf steigende Flanken, das andere auf fallende reagiert. Die Ausgangswerte beider Flip-Flops werden abwechselnd durch einen Multiplexer in ihren stabilen Taktphasen auf den Ausgang des DETFFs durchgeschaltet. Aus den zuvor erörterten Beobachtungen ergibt sich die durch Einfügen eines Inverters in einen Weg des DETFFs modifizierte Schaltung für den dritten Phasendetektor, die in Abbildung 4.9 dargestellt ist. In einer differenziellen Umsetzung der Schaltung wird die Invertierung durch Kreuzen der beiden Signalleitungen erreicht.

Auf noch einen Umstand des Einsatzes zweier Phasendetektoren nach dem „Bang-bang“-Prinzip sei hingewiesen. Betrachtet man den Fall a) „early“ in Abbildung 4.7, zeigt sich, daß der Abtastzeitpunkt bereits sehr nahe an die Flanke des Datenbits herankommt. Dies wird, wie schon mehrfach angeführt, die Bitfehlerrate des Systems deutlich erhöhen. Das



**Abbildung 4.9:** Blockdiagramm des modifizierten double-edge-triggered Flip-Flops

Problem der präzisen Positionierung des Abtastzeitpunktes in der Mitte der Bitperiode taucht auch in anderen Phasendetektorschaltungen immer wieder auf; die Suche nach Ansätzen, die diesen Punkt bereits auf prinzipieller Ebene lösen, ist geboten.

Die Schaltung eines DETFFs bildet die Information darüber, ob es sich bei der zuletzt in dem an den Takteingang angeschlossenen Datensignal aufgetretenen Flanke um eine steigende oder fallende handelte, in der örtlichen Struktur des Flip-Flops ab, da zwei unterschiedliche Pfade für beide Fälle existieren. Die Problematik der Unsicherheit des Abtastzeitpunktes muß umgehbar sein, wenn ein Weg gefunden wird, die in der Flankenorientierung enthaltene Information über das Datensignal zu nutzen. Oder klarer ausgedrückt: wenn bekannt ist, daß die letzte im Datensignal aufgetretene Flanke beispielsweise eine steigende war, ist automatisch auch sicher, daß alle Bits bis zur nächsten auftretenden Flanke logische Einsen sind. Analoges gilt für den Fall einer fallenden Flanke. Die im Abschnitt 4.4 eingeführte Empfängerarchitektur erreicht die Datenrückgewinnung ohne Abtastung unter Ausnutzung der durch die Flankenorientierung gegebenen Information.

#### 4.3.4 Zusammenfassung der Vorbetrachtungen

In den vorangegangenen Abschnitten wurden neben für das Verständnis der im folgenden vorgenommenen Untersuchungen notwendigen theoretischen Betrachtungen die grundlegenden Prinzipien verschiedener in der Literatur veröffentlichter Ansätze für Empfängerschaltungen erläutert. Diese wurden ausgewählt, weil sie Vorteile aufweisen, die eine neue Empfängeranordnung idealerweise übernehmen sollte. Dabei wurden aber auch ihre Nachteile erläutert, die ein neuer Ansatz lösen sollte.

Aus dem Wunsch nach einem digitalen Schleifenfilter in der Regelschleife leitet sich die Notwendigkeit eines quantisierenden Phasendetektors her. Die immer gültige Forderung nach niedriger Leistungsaufnahme des Transceiver-Chips bedingt eine mit herabgesetztem Takt arbeitende Empfängerarchitektur.

Aus praktischen Erwägungen muß der quantisierende Phasendetektor um ein Frequenzdetektionsschema erweitert werden. Die prinzipielle Funktionsweise einer solchen Frequenzdetektion wurde erläutert. Es wurde erkannt, daß der Einsatz mehrerer Phasen des Oszillatorsignales zur höheren Auflösung der Phasendifferenz zwischen Takt und Daten genutzt werden kann. Die Idee, die Information der Flankenpolarität zur Rückgewinnung der Daten heranzuziehen, wurde eingeführt.

### 4.4 Die neue Empfängeranordnung

#### 4.4.1 Pegel- und Flankendarstellung eines NRZ-kodierten Signales

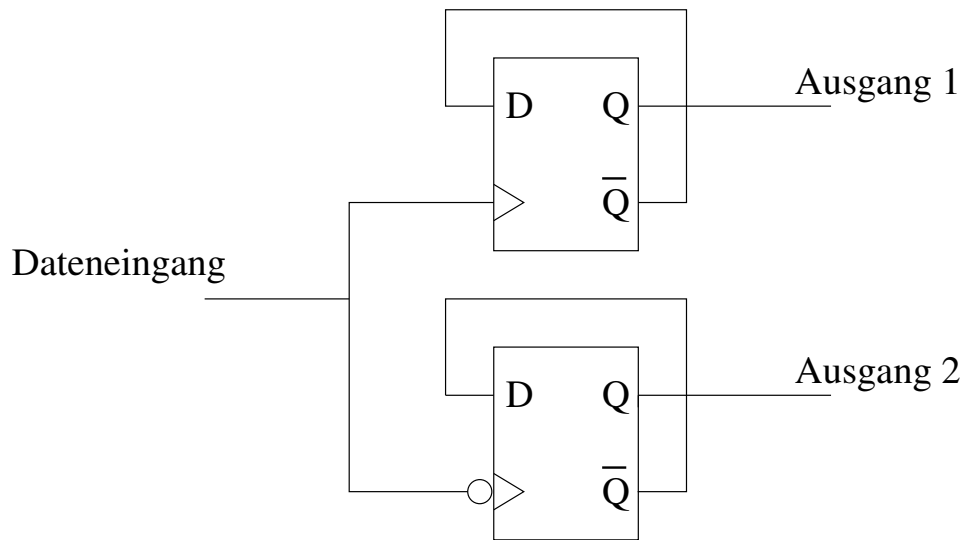
Abschnitt 3.2.2 auf Seite 33 erläuterte die alternative Darstellung eines NRZ-Signales als Folge von Flanken; zwischen den beiden Darstellungsformen kann leicht gewechselt werden. Zur Datenrückgewinnung sollen gemäß den Vorbetrachtungen Wege gefunden werden, die auf die Abtastung des Datensignales verzichten und stattdessen allein das Auftreten von Flanken im Datensignal auswerten.

Sofern sich eine Demultiplexeranordnung finden läßt, die ein flankenkodiertes Signal in zwei oder mehr niederfrequenteren, ebenfalls flankenkodierte Signale aufspaltet, kann zudem die Takt-/Datenrückgewinnung bei einer niedrigeren Geschwindigkeit als die Datenrate vorgenommen werden. Im folgenden Abschnitt wird eine solche Demultiplexeranordnung vorgestellt.

#### 4.4.2 Die neue Demultiplexergrundzelle

Rückgreifend auf die Idee des double-edge-triggered Flip-flops, das die Information darüber, ob es sich bei der zuletzt aufgetretenen Flanke um eine steigende oder fallende gehandelt hatte, in einer örtlichen Struktur abbildet, soll nun eine Struktur gefunden werden, die diese Information auf zwei verschiedene Ausgänge verteilt. Es muß folglich eine Schaltung entwickelt werden, die beim Auftreten einer Flanke am Eingang abwechselnd eine Flanke auf jeweils einem ihrer beiden Ausgänge erzeugt.

Abbildung 4.10 zeigt die resultierende Schaltung. Sie besteht aus zwei einfachen flankengesteuerten Flip-Flops, deren invertierende Ausgänge auf ihre Eingänge zurückgekoppelt sind. Von den beiden Flip-Flops reagiert eines auf steigende Flanken, das andere auf fallende. Das Datensignal wird, ähnlich wie in der „Bang-bang“-Schaltung, an die Takteingänge der Flip-Flops angeschlossen. In einer differenziellen Auslegung wird lediglich der



**Abbildung 4.10:** Schaltung der 1:2-Demultiplexer-Grundzelle

Takteingang des zweiten Flip-Flops überkreuz angeschlossen, so daß beide Flip-Flops als Schaltung und Layout identisch ausfallen.

Die Funktionsweise der Schaltung ist so einfach wie ihr Aufbau. Abbildung 4.11 zeigt ein Beispieldatensignal und die daraus resultierenden Ausgangssignale. Deutlich erkennt man, daß die mittlere Frequenz der beiden Ausgangssignale gegenüber dem Eingangssignal halbiert ist. Auf beiden Ausgängen ist die Flankenrate halb so groß wie die des Eingangssignales, was belegt, daß die Schaltung eine Demultiplexerfunktion für flankenkodierte Signale bewirkt, obwohl keinerlei Abtastung des Datensignales durchgeführt wurde. Auch bei dieser neuartigen Demultiplexerschaltung kann die 1:2-Demultiplexergrundzelle in einer Baumstruktur angeordnet werden, so daß größere Demultiplexerverhältnisse problemlos möglich sind.

Die Abbildung 4.12 zeigt, daß die beiden Ausgangssignale zusammen denselben Informationsgehalt aufweisen wie das Eingangssignal, das durch eine Exklusiv-Oder-Verknüpfung der Ausgangssignale zurückgewonnen werden kann. In der Abbildung wurden dieselben Signalverläufe wie in der vorangegangenen benutzt, um den direkten Vergleich zu ermöglichen.

Eine wichtige Eigenschaft der Ausgangssignale ist jedoch problematisch: zwar ist die Frequenz des Datensignales um die Hälfte herabgesetzt worden, was Verstärkung und Weiterleitung der Teilsignale vereinfacht, doch weist die Phase der herabgesetzten Signale immer noch eine Auflösung gleich der Bitlänge des Eingangssignales auf!

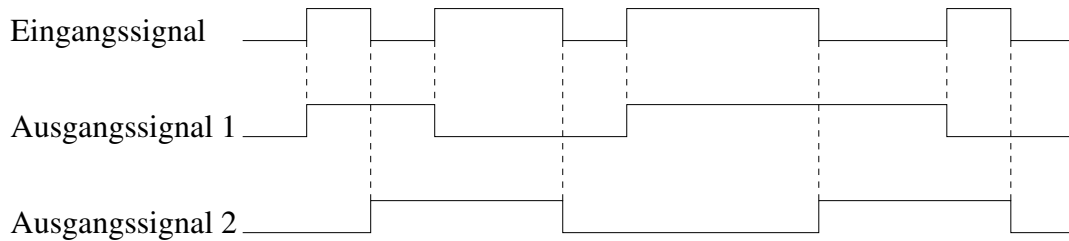


Abbildung 4.11: Beispielsignale zur Funktion der Demultiplexer-Grundzelle

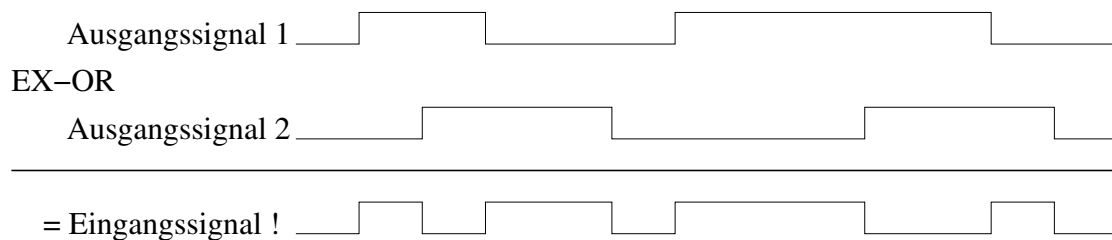


Abbildung 4.12: Rückgewinnung des Eingangssignales

Dies bedeutet, daß die Ausgangssignale nicht mit einer gegenüber der Eingangsdatenrate halbierten Frequenz abgetastet werden können, da in jedem der Ausgangssignale zu jedem beliebigen einem Vielfachen einer Bitlänge entsprechenden Zeitpunkt eine Flanke auftreten kann. Würden die Ausgangssignale mit halbiert Frequenz abgetastet, so ließe sich die Position der Flanken nicht genau bestimmen, die Datenrückgewinnung wäre unmöglich. Deshalb muß der neuartige Demultiplexer in eine spezielle Anordnung eingebunden werden, die die Datenrückgewinnung erlaubt, ohne jedoch auf die Abtastung bei voller Datenrate zurückzugreifen.

#### 4.4.3 Blockschaltbild

Abbildung 4.13 zeigt das Blockschaltbild des gesamten Empfängers, ausgelegt für ein Demultiplexverhältnis von 1:8. Er besteht aus fünf Blöcken, die zur besseren Unterscheidung mit unterschiedlichen Grautönen unterlegt wurden. Der erste Block links stellt den zweistufig aufgebauten Demultiplexerbaum dar, der somit eigentlich nur ein Demultiplexverhältnis von 1:4 aufweist. Der Mechanismus der Datenrückgewinnung bewirkt allerdings eine zusätzliche Herabsetzung der Datenrate um einen Faktor zwei, so daß sich das obengenannte Verhältnis von 1:8 ergibt.



Die vier Ausgänge des Demultiplexerbaumes sind mit den Takteingängen von jeweils zwei Registern verbunden. Sie bestehen aus einer Reihe identischer flankengesteuerter Flip-Flops und stehen im Schaltungsdiagramm links der Mitte. Je ein Register wird durch das invertierte Datensignal angesteuert, so daß es auf fallende Flanken reagiert, das andere hingegen wie gewohnt auf steigende Flanken.

Die Dateneingänge der Flip-Flops werden mit den verschiedenen Ausgängen des lokalen Oszillators verbunden. Dieser VCO ist als Mehrphasenoszillator ähnlich einem Ringoszillator aufgebaut. Um jedoch die notwendige Reinheit des Taktsignales gemäß den Standards für die optische Datenübertragung zu erreichen, muß dieser Oszillator über LC-Resonatoren gestimmt werden. Entsprechende Schaltungsanordnungen, die hinsichtlich dieses Aspektes geeignet sind, wurden bereits in der Literatur vorgestellt [15]. Die Oszillatorfrequenz wird durch einen Phasenregelkreis kontrolliert, dessen Teil er bildet. Seine Elemente sind im obersten Block der Abbildung dargestellt.

Der Oszillator schwingt, wie erwähnt, bei herabgesetzter Taktfrequenz. Da die Schaltung für ein Demultiplexverhältnis von 1:8 ausgelegt wurde, liegt seine Taktfrequenz bei gerade einmal 1,25 GHz, einer Frequenz, die in CMOS leicht mit großen Pegelhüben und deswegen robust gegenüber Rauschen verwirklicht werden kann. Diese Eigenschaft des Empfängers ist insbesondere in Hinsicht auf noch höhere Datenraten interessant, können doch in CMOS schon jetzt deutlich höhere Frequenzen stabil erzeugt werden.

Die minimale Phasendifferenz zwischen zwei Phasensignalen des Oszillators sollte nach Möglichkeit ein Drittel, höchstens aber die Hälfte, der Bitdauer eines empfangenen Bits betragen. Auf einen 10 GBit/s-Empfänger angewendet, wären das 33 ps bzw. 50 ps. Soll der Empfänger mit einem Frequenzdetektionsschema ausgestattet werden, ist wegen der notwendigen Drehsinnbestimmung des Phasoren des Datensignales<sup>7</sup> mindestens eine Phasenauflösung einer Drittelbitlänge erforderlich. Bei einer Phasendifferenz von 33 ps und einer Frequenz von 1,25 GHz muß der Oszillator dann immerhin 24 Phasensignale aufweisen, also zwölfstufig aufgebaut sein.

Die herabgesetzte Oszillatorfrequenz muß durch eine höhere Anzahl von Phasensignalen ausgeglichen werden. Dies ist einleuchtend, führt man sich vor Augen, daß der Empfang eines Hochgeschwindigkeitsdatensignales in jedem Fall ein kleineres Zeitmaß als dessen Bitlänge bedingt. In der vorgestellten Sender- und Empfängerarchitektur wird dieses Zeitmaß jedoch von der Frequenz des Oszillators auf dessen Phasenauflösung vorteilhaft verlagert.

Jede am Eingang des asynchronen Demultiplexers auftretende Flanke erscheint auf genau einem seiner Ausgänge. Die in den einzelnen Wegen des Demultiplexerbaumes anfallenden Verzögerungen weichen nur aufgrund von Parameterschwankungen über die Chipfläche und infolge von Rauscheinflüssen voneinander ab, da alle Elemente des Baumes identisch aufgebaut sind. Erscheint eine Flanke auf einem der vier Ausgänge des Baumes,

---

<sup>7</sup>Vgl. hierzu den Abschnitt 4.3.3 zur Frequenzdetektion

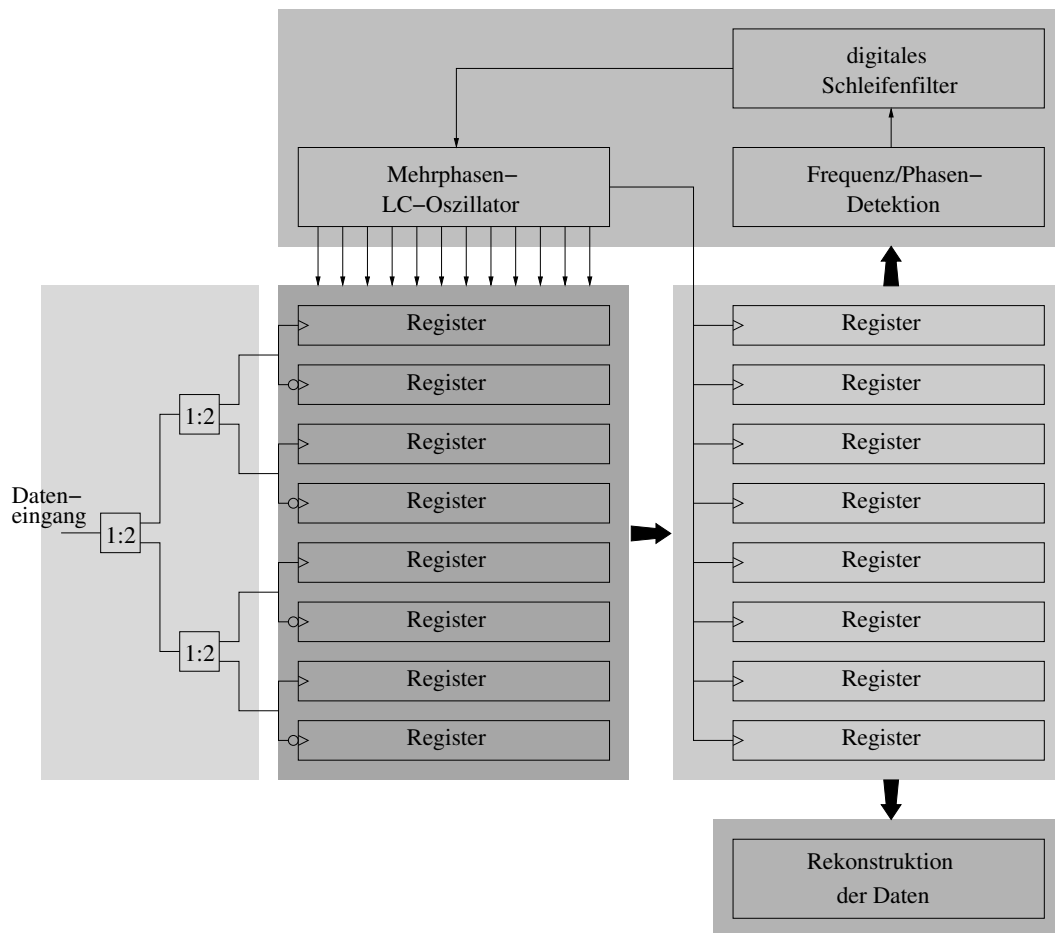


Abbildung 4.13: Blockdiagramm des seriellen Empfängers

wird entsprechend ihrer Polarisierung eines der beiden an jeden Ausgang angeschlossenen Register sämtliche Phasensignale des Oszillators abtasten.

Auf diese Weise fällt dem Oszillator die Rolle der „Stoppuhr“ des Systems zu. Indem jede Flanke die Zustände aller Oszillatorausgänge einliest, wird der exakte Zeitpunkt ihres Auftretens festgehalten. Die gesendeten Daten können aus den gespeicherten Zeitpunkten zurückgewonnen werden.

Da auf jedem Ausgang des Demultiplexerbaumes höchstens alle vier Eingangsbits eine Flanke erscheinen und abwechselnd eines der beiden angeschlossenen Register aktivieren kann, reicht es aus, die in den asynchron getakteten Registern gespeicherten Werte mit einer um den Faktor acht gesenkten Frequenz auszulesen<sup>8</sup>. Dies geschieht mit dem Takt des Multiphasen-Oszillators, der einen zweiten Satz von Registern taktet.

An diesen zweiten Registersatz (im Blockdiagramm in der Mitte rechts) schließt sich zum einen die Datenrückgewinnung (unten rechts), zum anderen die aus Frequenz-/Phasendetektion, digitalem Schleifenfilter und dem spannungsgesteuerten Oszillator bestehende PLL an (oben). Beide Blöcke arbeiten synchron und bei der leicht beherrschbaren Frequenz von 1,25 GHz. Jedem wird im Folgenden ein eigener Abschnitt gewidmet.

### 4.4.4 Die Datenrückgewinnung

Die Datenrückgewinnung gestaltet sich einfach. Der Mehrphasenoszillator der Beispielformatkonfiguration besitzt 24 Phasensignale. Aufgrund der differentiellen Auslegung besteht er aus zwölf Stufen. Es ist unnötig, alle 24 Signale abzutasten, denn tatsächlich enthalten jeweils die um 180° gegeneinander versetzten Phasen dieselbe Information über die Zeit. Der Unterschied besteht lediglich in der Polarisierung des Oszillatorsignales.

Werden also die ersten zwölf Phasenausgänge des Oszillators in ein Register gelesen, kann der Zeitpunkt des Auftretens der zugeordneten Flanke im Datensignal anhand des Wechsels zwischen Nullen und Einsen und der Polarisierung des letzten Phasenausganges bestimmt werden. Die Abbildung 4.14 stellt die Zustände der zwölf Phasenausgänge des Oszillators zu acht verschiedenen Zeitpunkten dar, wobei gefüllte Kreise eine logische Eins, leere eine logische Null bedeuten. Zu den acht verschiedenen Zeitpunkten, an denen eine Flanke in einem idealen Datensignal auftreten kann, sind die Zustände des Oszillators eindeutig voneinander unterscheidbar.

Natürlich stellt die Abbildung nur den Idealfall dar, in dem die Flanken des Datensignales exakt in Abständen von ganzzahligen Vielfachen von 100 ps auftreten und die Phasenlagen von Oszillator und Datensignal übereinstimmen. In einer realen Umgebung wird der Empfang durch Jitter und Schwankungen der Datenrate erschwert.

---

<sup>8</sup>Diese Behauptung ist nur unter der Voraussetzung idealer Register wahr. Die sich aus der endlichen Schaltgeschwindigkeit der Register ergebenden praktischen Umstände werden in Abschnitt 4.4.5 untersucht.

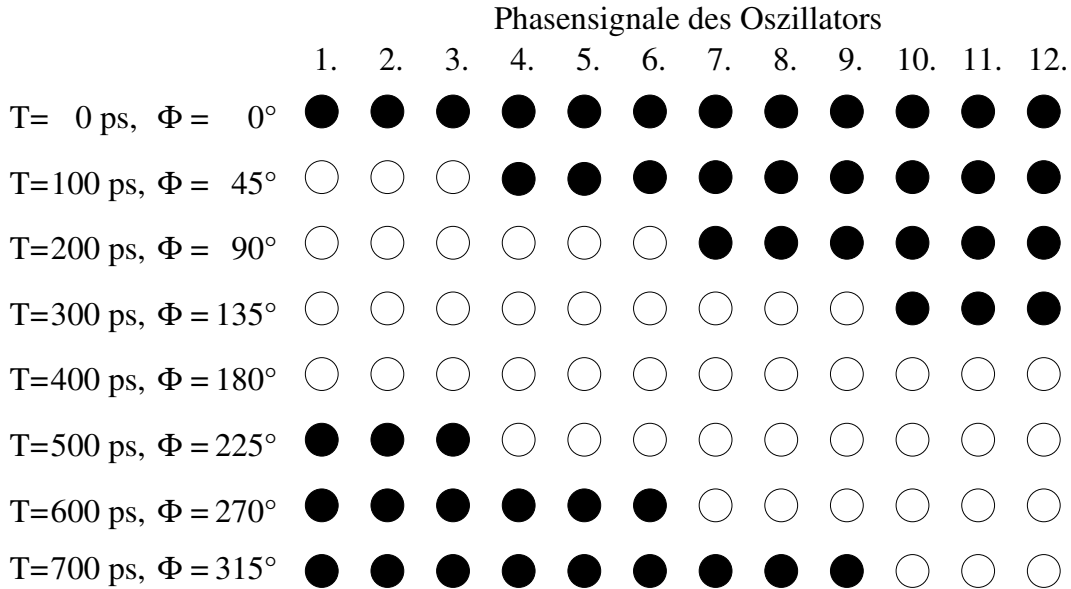


Abbildung 4.14: Schematische Darstellung der möglichen Oszillatorzustände

Die Abbildung 4.15 dient zur Veranschaulichung der Datenrückgewinnung; sie bildet in ihrem oberen Teil ein Beispielsignal mit nummerierten Flanken ab. Darunter sind die jeweiligen durch die Flanken gespeicherten logischen Zustände der Oszillatorausgänge gemäß der vorangegangenen Darstellung 4.14 zu sehen.

Die gespeicherten Zeitpunkte sind in zwei Gruppen unterteilt, die sich durch die Polarisierung des 11. Ausganges<sup>9</sup> des Oszillators unterscheiden: in der ersten Gruppe sind die Zeitpunkte zusammengefaßt, die an dieser Stelle eine logische Eins aufweisen, in der zweiten Gruppe die gegenteiligen. Die erste Gruppe enthält somit alle Ereignisse der ersten 400 ps des Datensatzes, die zweite solche der letzten 400 ps.

Die Datenrückgewinnung wird durch logische Verknüpfung der gespeicherten Zeitpunkte innerhalb beider Gruppen erreicht. Die Verknüpfungsvorschriften beider Gruppen unterscheiden sich geringfügig wegen der invertierten Oszillatorzustände in der zweiten Takt-hälfte.

Eine offensichtliche Möglichkeit läge in der Exklusiv-Oder-Bildung der einzelnen Oszillatorzustände. Ein Exklusiv-Oder-Gatter, zumal mit mehreren Eingängen, ist allerdings schaltungstechnisch relativ aufwendig. Deshalb sollte ein Ansatz verfolgt werden, der in der Implementierung einfacher ausfällt.

<sup>9</sup>bzw. der Mehrheit der Bits 10, 11 und 12. Siehe hierzu auch den Abschnitt „Senkung der Bitfehlerrate“

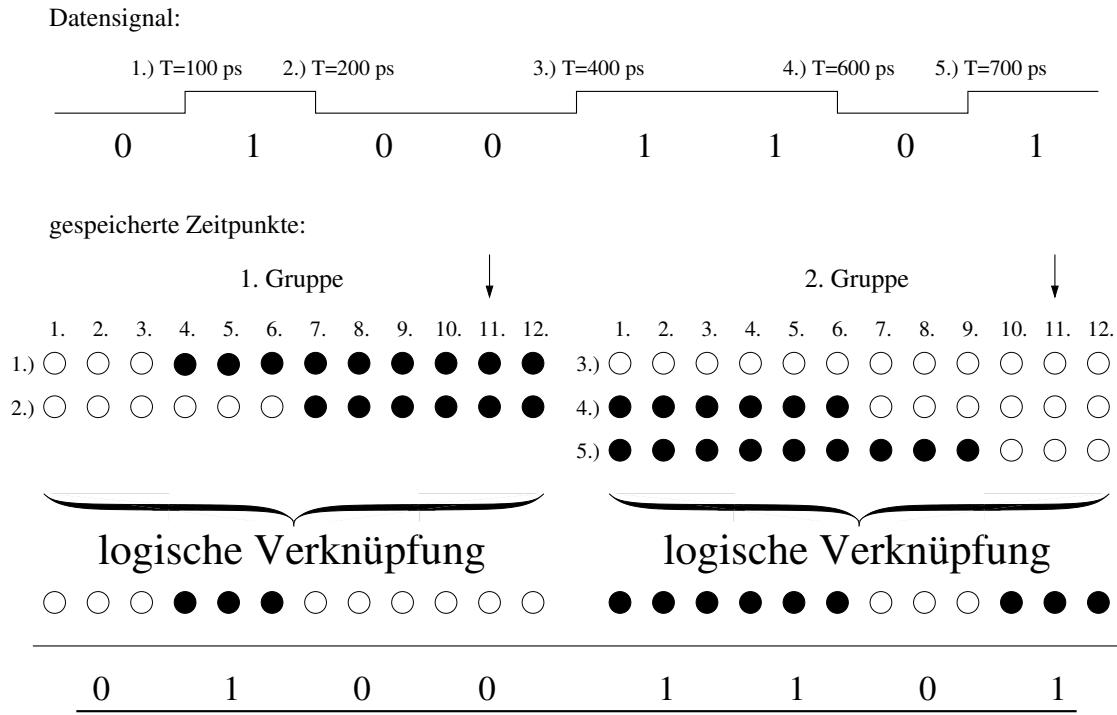


Abbildung 4.15: Beispiel zur Datenrückgewinnung

Die einzelnen Flankenzeitpunkte werden in Paaren von aufeinander folgenden steigenden und fallenden Flanken angeordnet. Orientierung und Reihenfolge ergeben sich aus der Verdrahtung des Demultiplexerbaumes. War die erste Flanke im aktuellen Datensatz eine fallende, so wird sie mit einem Satz Einsen verknüpft; dasselbe gilt für die letzte Flanke, wenn es sich um eine steigende handelte. Fehlt eine Flanke innerhalb eines Datensatzes, weil sie wegen einer Störung derartig verflacht wurde, daß sie das zugehörige Register nicht durchgeschaltet hat, dann wird sie als eine Reihe Nullen angesehen.

Die durch fallende Flanken gespeicherten Oszillatorzustände werden logisch invertiert, wohingegen die durch steigende Flanken gespeicherten unverändert bleiben. Für die zweite Gruppe von Zeitpunkten ist diese Vorschrift umzukehren. Die Resultate der Operation werden logisch UND-verknüpft, woraus sich die zwischen den Flanken liegenden Datenbits in NRZ-Darstellung ergeben. Die Zwischenergebnisse werden durch einer ODER-Verknüpfung zu einem einzigen Datum zusammengefügt. Das Ausgangsdatum setzt sich aus den hintereinandergestellten Ergebnissen beider Gruppen zusammen.

Die logische Verknüpfung kann durch zwei Stufen von NAND-Gattern realisiert werden, was den Aufwand verringert. Außerdem eliminiert die UND-Verknüpfung unvollständige,

weil fehlerhaft detektierte, Flankenpaare, so daß nur noch die zwischen ihnen liegenden Bits falsch empfangen werden, aber nicht der gesamte Datensatz verfälscht wird<sup>10</sup>.

Der Spezialfall eines Datensatzes von vier Bits ohne Pegelwechsel kann nur dann korrekt verarbeitet werden, wenn die Orientierung der vor diesem Satz zuletzt aufgetretenen Flanke bekannt ist. Deshalb wird sie zur Datenverarbeitung gespeichert. Dabei gilt folgende Gesetzmäßigkeit: das Ergebnis ist ein Satz von logischen Einsen, wenn die letzte Flanke eine steigende war, ein Satz von logischen Nullen, wenn die sie eine fallende war.

### Senkung der Bitfehlerrate

Faßt man die Ausgänge des Oszillators in Gruppen zu je drei Signalen zusammen, kann ein einfacher Abstimmungsalgorithmus genutzt werden, um die Bitfehlerrate in der Datenrückgewinnung aufgrund von Störungen des Datensignales zu senken. Dabei wird der Wert als gültig angenommen, den mindestens zwei der drei Signale in der Gruppe aufweisen. Diese Operation läßt sich durch einen Volladdierer durchführen; das Übertragsbit des Volladdierers stellt dann das Ausgangsbit dar.

Die Abstimmung könnte entweder für alle gespeicherten Zeitpunkte einzeln durchgeführt werden oder aber erst für das zurückgewonnene Signal. Bei Abstimmung nach der logischen Verknüpfung sinkt der schaltungstechnische Aufwand, weil die gegenüber der zweistufigen Logik komplexere Abstimmung nur einmal durchgeführt wird. Auch die in Abschnitt 4.4.6 erläuterte Taktrückgewinnung gestaltet sich einfacher mit dem aus der logischen Verknüpfung der gespeicherten Oszillatorzustände hervorgehenden Datensatz.

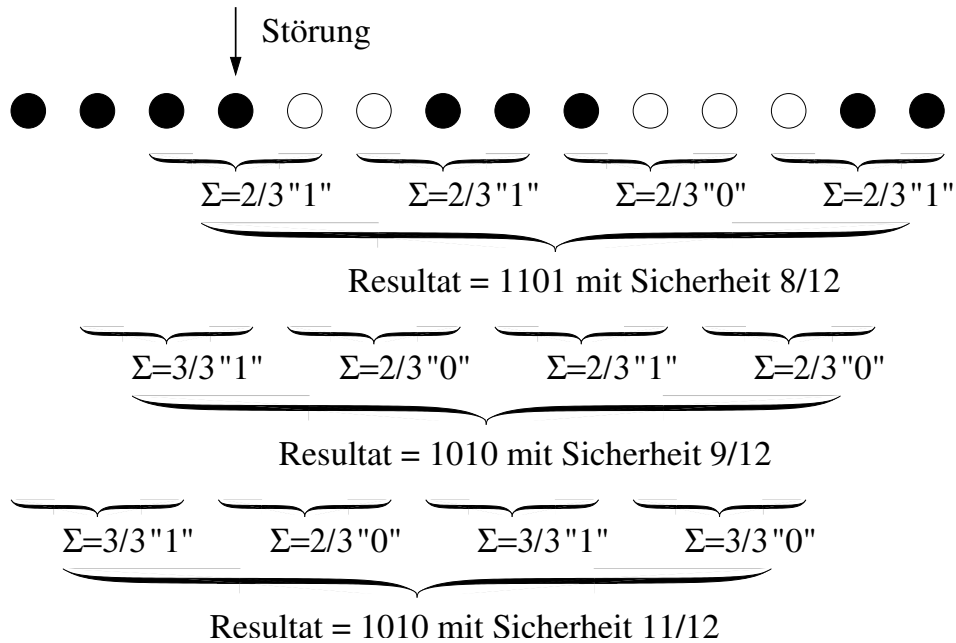
Es empfiehlt sich, die Abstimmung parallel auch für um eine Bitposition nach links bzw. rechts verschobene Kopien des Datensatzes durchzuführen. Auf diese Weise können kurzzeitige Phasendifferenzen zwischen Oszillator und Datensignal ausgeglichen werden; dazu müssen allerdings ein Bit des jeweils vorhergehenden und des folgenden Datensatzes zur Verfügung stehen. Die Ergebnisse der Abstimmungen der einzelnen Dreierblöcke werden innerhalb jeder der drei Kopien addiert. Das höchste Ergebnis bezeichnet das Datum mit der geringsten Fehlerwahrscheinlichkeit.

Abbildung 4.16 zeigt ein Beispiel, in dem das rückgewonnene Datum einen Phasenversatz von einer Drittelbitlänge zur Ideallage besitzt und eine Flanke durch Jitter derartig verschoben wurde, daß ein Bit verkürzt erscheint. Das Ergebnis der Abstimmung weist auf den Datensatz mit der geringsten Fehlerwahrscheinlichkeit.

Zur Durchführung der Datenrückgewinnung müssen aus allen Registern diejenigen ausgewählt werden, die sich tatsächlich innerhalb des letzten Lesezyklus' geändert haben.

---

<sup>10</sup>Der resultierende Fehler besteht aus einer unter Umständen langen Serie falsch polarisierter Bits. Diese sogenannten „Burst-errors“ können jedoch auf zwei Bitfehler reduziert werden, wenn der Datenstrom nach der Regel  $Y(n) = X(n) \oplus X(n-1)$  auf Sender- und Empfängerseite kodiert bzw. dekodiert wird [16] und [17].



**Abbildung 4.16:** Zur Erläuterung des Abstimmungsalgorithmusses

Dadurch ist gleichzeitig auch die Anzahl der aufgetretenen Flanken bekannt, so daß das zurückgewonnene Datum anhand dieser Information geprüft werden kann. Es sind beliebig komplizierte Algorithmen denkbar, die diese Information nutzen, um die Bitfehlerrate der Übertragungsstrecke zu senken.

Grundsätzlich sind einzelne Bits innerhalb längerer Folgen gegenteiliger Polarisierung besonders gefährdet, falsch empfangen zu werden. Werden im Ausgangsdatum zwei Flanken weniger festgestellt, als eigentlich aufgetreten sind, ist wahrscheinlich ein solches einzelnes Bit verlorengegangen. Die Ergebnisse des Abstimmungsalgorithmus' könnten dann auf diese Fehlerform überprüft werden.

In jedem Falle läßt die herabgesetzte Verarbeitungsgeschwindigkeit, die durch traditionelles Demultiplexen noch weiter verringert werden kann, die Umsetzung von komplexen Algorithmen zu, die die Bitfehlerrate des Systemes senken. Der neuartige Empfänger bietet diese Möglichkeit bei geringem Schaltungsaufwand.

Die einfachste Ausführung erfordert lediglich einige Dutzend Volladdierer und Exklusiv-Oder-Gatter, um die Datenrückgewinnung unter Ausnutzung der durch die größere Phasenaufösung des Oszillators erreichte Überabtastung des Datensignales durchzuführen.

#### 4.4.5 Das Auslesen des asynchronen Registersatzes

Der Inhalt des ersten Registersatzes muß zur Auswertung in den zweiten übertragen werden. Dabei müssen zwei Dinge sichergestellt sein:

1. Es werden aus den Registern zur Datenauswertung nur jene gespeicherten Oszillatorzustände herangezogen, die sich innerhalb des letzten Oszillatortaktes geändert haben.
2. Die ausgelesenen Werte der Register sind gültig, d.h. keines wurde ausgelesen, bevor seine Ausgänge einen gültigen Ausgangswert angenommen hatten.

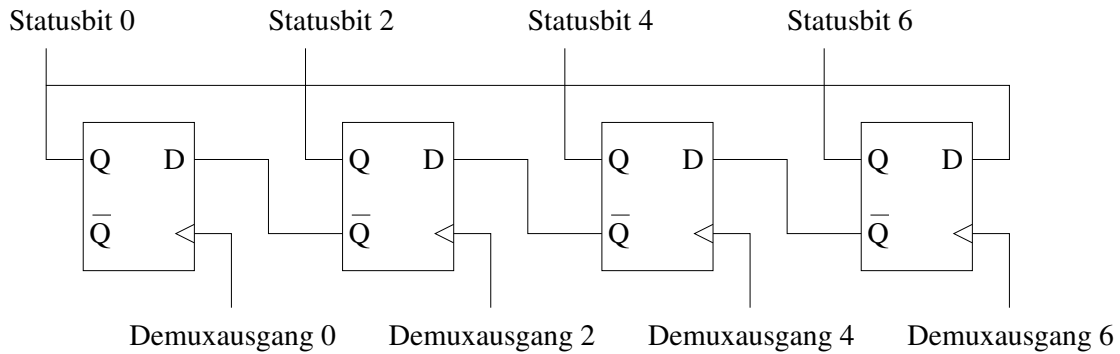
Die Reihenfolge, in der die einzelnen Register durch die Flanken aktiviert werden, ist durch die Struktur des Demultiplexers fest vorgegeben. Wird ein dreizehntes Bit in jedem Register vorgesehen, dessen logischer Zustand sich invertiert, wenn das zugehörige Register durch eine Flanke im Datensignal aktiviert worden ist, so kann diese Information genutzt werden, um die neu hinzugekommenen gespeicherten Zeitpunkte zu erkennen.

Diese Statusbits werden am einfachsten aufgebaut wie in Abbildung 4.17 dargestellt. Die in einer einem Ringoszillator ähnelnden Struktur angeordneten Flip-Flops lesen jeweils den invertierten Wert des ihnen nachfolgenden Statusbits ein, wobei das letzte Bit des Ringes eine Ausnahme bildet, damit durch den ganzen Ring hindurch sich die Invertierungen nicht gegenseitig aufheben. Aus Symmetriegründen sollte die Schaltung zweimal ausgelegt werden, und zwar einmal für die vier durch fallende Flanken aktivierten Register, ein anderes Mal für jene, die durch steigenden Flanken aktiviert werden. Die Schaltung stellt für jedes Flip-Flop eine Setup-Zeit von mindestens 600 ps minus die sog. Clock-to-Q-Zeit sowie eine Hold-Zeit von mindestens 200 ps sicher.

Da das empfangene Datensignal Störungen unterliegt, kann, auch wenn die PLL auf das Datensignal eingerastet ist, das Auftreten einer Flanke zeitlich mit dem Auslesen des asynchronen Registersatzes zusammenfallen. Die Korrektheit der einzelnen Bits des durch diese Flanke gespeicherten Oszillatorzustandes ist dann nicht mehr garantiert, weil die Ausgänge der Flip-Flops noch in undefiniertem Zustand waren, als der Registersatz ausgelesen wurde. Da jede Flanke die Datenrückgewinnung eines ganzen Satzes von Bits beeinflusst, muß ein Weg gefunden werden, die Ausleseprozedur gegen solche Fehler unanfällig zu machen.

Das korrekte Auslesen des Registersatzes wird zusätzlich durch den Umstand erschwert, daß in einem Takt des lokalen Oszillators unter bestimmten Bedingungen auch neun Flanken auftreten können. Dies kann immer dann geschehen, wenn kurzzeitig die Datenrate über die Oszillatorfrequenz steigt; außerdem liefert die Datenrückgewinnung, wie





**Abbildung 4.17:** Schaltung zur Detektion neuer im asynchronen Registersatz gespeicherter Flankenzeitpunkte

im nächsten Abschnitt beschrieben, auch die Grundlage für die Frequenz- und Phasendetektion, die auch dann funktionieren muß, wenn der Phasenregelkreis der Takt-/Datenrückgewinnung noch nicht auf das Datensignal eingerastet und somit die Oszillatorfrequenz noch nicht an die Datenrate gekoppelt ist.

Ist die Datenrate höher als der Oszillatortakt<sup>11</sup>, dann könnte es passieren, daß ein gespeicherter Zeitpunkt vor seiner Berücksichtigung bei der Datenrückgewinnung bereits überschrieben wurde. Dies wäre dann möglich, wenn er direkt nach einer Oszillatorflanke auftrat und erst einen Oszillatortakt später ausgelesen und ausgewertet werden sollte, innerhalb des unter der Datenrate liegenden Oszillatortaktes aber zufälligerweise neun Datenflanken auftraten. Für das Auslesen des asynchronen Registersatzes muß deshalb eine gegen diese Problematik unanfällige Methode gefunden werden. Die einfachste ist, den Registersatz sowohl mit der steigenden als auch der fallenden Flanke des lokalen Oszillators auszulesen. Dadurch wird der Registersatz ungefähr doppelt so häufig ausgelesen, wie jedes seiner Elemente einen neuen Wert annehmen kann. Es leitet sich die Bedingung ab, daß die Ausgänge des ersten Registersatzes sich weniger als einen halben 1,25 GHz-Taktzyklus, also weniger als 400 ps, in unbestimmtem Zustand befinden dürfen, bevor sie ihren neuen Wert annehmen, damit das Auslesen mit einem doppelt so hohen Takt ausreichend ist.

Wird die Möglichkeit berücksichtigt, daß die Datenrate etwas höher liegen kann als die Oszillatorfrequenz, so muß bei der Berechnung der Anforderungen an die Geschwindigkeit der Flip-Flops auch noch die in Wahrheit verkürzte zur Verfügung stehende Zykluszeit einbezogen werden. Die maximale sog. Clock-to-Q-Zeit des Flip-Flops darf deshalb nur  $800 \cdot \frac{N \cdot f_{osc}}{D}$  ps betragen. Um genügend Sicherheitsraum zu haben, sei eine maximale Abweichung der Datenrate über die Oszillatorfrequenz mal  $N$  um ein Ach-

<sup>11</sup>Präziser ausgedrückt: höher als der Oszillatortakt multipliziert mit dem Demultiplexfaktor  $N$

tel angenommen, was einer Schaltzeit des Flip-Flops von maximal 720 ps entspricht, einem Wert, der weit größer ist, als er in einer nach den optischen Standards arbeitenden Takt-/Datenrückgewinnung jemals auftreten darf. Als zweite Bedingung folgt, daß die Flip-Flops weniger als  $400 \cdot \frac{N \cdot f_{osc}}{D}$  ps in unbestimmten Zustand sein dürfen. Da der Ausgang eines Flip-Flops im Normalfall noch für einige Zeit stabil den älteren Wert bereitstellt, ist die Anforderung an die Flip-Flops gering genug für den Einsatz stromsparender dynamischer Flip-Flops. Aufgrund der wünschenswerten symmetrischen Empfindlichkeit der Flip-Flops gegenüber logischen Einsen und Nullen an ihren Eingängen und wegen des ggf. implementationsbedingten geringen Hubes der Oszillatorausgänge sind aber die leistungsintensiveren differentiellen CML-Flip-Flops, angepaßt an die niedrigere Schaltgeschwindigkeit, den dynamischen vorzuziehen. Der zweite Registersatz und die nachfolgenden Schaltungen sollten allerdings auf jeden Fall aus dynamischen Flip-Flops aufgebaut werden, um die Vorteile der neuen Empfängerstruktur hinsichtlich der Leistungsaufnahme auszunutzen.

Jedesmal, wenn der Registersatz ausgelesen wird, müssen alle gespeicherten Oszillatorzustände auf zwei Kriterien untersucht werden: im ersten Schritt wird geprüft, welche Oszillatorzustände neu sind, im zweiten festgestellt, welche wiederum alt genug sind, um gültige Ausgangswerte zu liefern. Die Statusbits der Register ermöglichen die Prüfung des ersten Kriteriums, während die des zweiten durch Untersuchung der Registerinhalte durchgeführt wird. Dazu genügt es, ein einziges Bit des gespeicherten Zeitpunktes zu überprüfen.

Wird beispielsweise der 12. Phasenausgang des Oszillators zum Auslesen des ersten Registersatzes benutzt, dann folgt aus der Abbildung 4.14 von Seite 92:

- wurde der Registersatz mit einer *steigenden* Flanke des 12. Phasenausganges ausgelesen, dann zeigen alle gespeicherten Zeitpunkte, die mindestens 400 ps alt sind und deshalb gültige Ausgangswerte aufweisen, eine logische Eins an der 12. Bitposition
- wurde der Registersatz mit einer *fallenden* Flanke des 12. Phasenausganges (bzw. mit einer steigenden Flanke des zugehörigen inversen 24. Phasenausganges) ausgelesen, dann zeigen alle gültigen gespeicherten Zeitpunkte eine logische Null an der 12. Bitposition

Um den ältesten gespeicherten Oszillatorzustand vor Überschreiben zu schützen, bevor er ausgewertet wurde, weil kurzzeitig die Datenrate höher liegt als die Oszillatorfrequenz, sollte anstelle des 12. Phasenausganges ein anderer, etwas früherer, zum Auslesen des Registersatzes gewählt werden. Dann wird der letzte gültige gespeicherte Oszillatorzustand gemäß der obenstehenden Ableitung etwas jünger als die 400 ps des halben Oszillatortaktes sein, der älteste aber vor Überschreiben durch die zum Zeitpunkt des Auslesen noch andauernde Halbphase mit 12. Bits gegenteiliger Orientierung geschützt.

Das Kriterium des 12. Bits allein ist allerdings nicht ausreichend, da jenes sich im Moment des Auslesens im metastabilen Zustand befinden und zufälligerweise einen richtigen Wert vortäuschen könnte und so bei der Eliminierung der ungültigen Zeitpunkte unerkannt bliebe. Um auch diesen recht unwahrscheinlichen Fall auszuschließen, sollten Status- und 12. Bit des fraglichen Flankenzeitpunktes mit denen desselben Registers einen halben Takt später verglichen werden. Hat sich der Zustand des 12. Bits geändert, dann war der Wechsel des Registerinhalts noch nicht beendet und der fragliche Flankenzeitpunkt muß im späteren Halbtakt ausgewertet werden.

Die Statusbits werden eingelesen und mit den im vorangegangenen Takt gespeicherten durch eine Exklusiv-Oder-Operation verglichen. Für die Datenrückgewinnung werden nur die in jenen Registern gespeicherten Flankenzeitpunkte berücksichtigt, deren Statusbit sich innerhalb des letzten Taktes geändert hat. Aus diesen werden wiederum alle jene Zeitpunkte eliminiert, deren 12. Bit nicht den richtigen logischen Wert aufweist.

Die in Abbildung 4.15 vorgenommene Unterteilung der gespeicherten Oszillatorzustände in zwei Gruppen spiegelt sich im Auslesealgorithmus des asynchronen Registersatzes wider. Sie führt auf zwei parallele, um einen halben Takt versetzt laufende nahezu<sup>12</sup> identische Verarbeitungseinheiten, die die Datenrückgewinnung in Fließband-Funktionsweise durchführen.

Obleich einige der vorzunehmenden Schritte jeweils in einem halben 1,25 GHz-Takt durchgeführt werden müssen, sind sie doch einfach genug, um mit schnellen CMOS-Schaltungstechniken wie der Pass-Transistor-Logik oder der Pseudo-NMOS-Logik und dynamischen Flip-Flops realisiert werden zu können.

### 4.4.6 Die Taktrückgewinnung

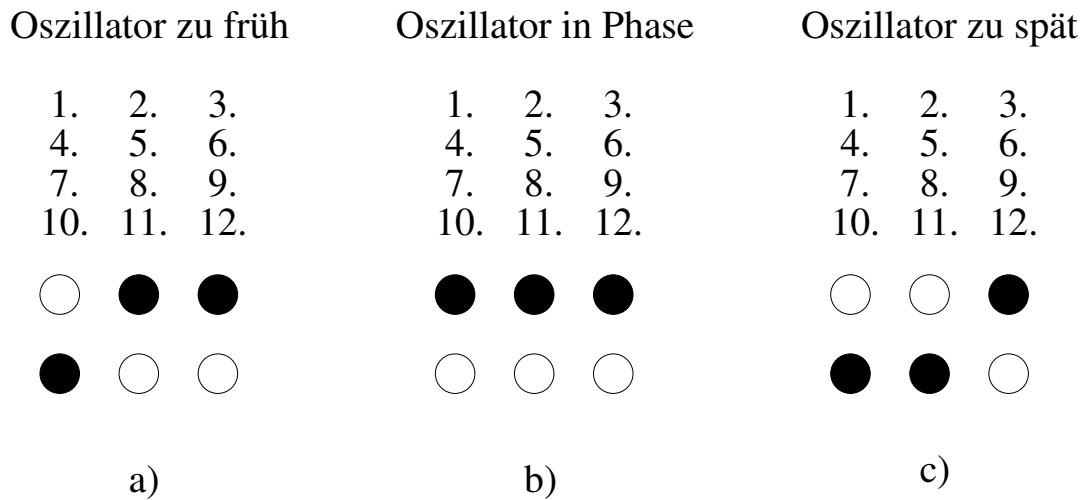
#### Die Phasendetektion

Die Phasendetektion wird ähnlich wie die Datenrückgewinnung aus den gespeicherten Zeitpunkten des Auftretens von Flanken im empfangenen Datensignal vorgenommen. Da die Phasenauflösung im Beispiel dreimal höher als die Bitlänge ist, kann das Phasenverhältnis zwischen Oszillator und Daten mit den Zuständen „zu früh“, „in Phase“ und „zu spät“ detektiert werden.

Im Abschnitt über die Datenrückgewinnung wurde dargelegt, daß jeweils drei nebeneinander liegende Phasensignale des Oszillators zur Rückgewinnung eines einzelnen Bits genutzt werden können. Für die Phasendetektion wertet man die Position des Wechsels zwischen logischen Einsen und Nullen in den gespeicherten Zeitpunkten aus. Sie wird

---

<sup>12</sup>Der Unterschied besteht lediglich in der notwendigen Invertierung der Oszillatorzustände der zweiten Gruppe.



**Abbildung 4.18:** Detektion des Phasenverhältnisses zwischen Oszillator und Daten

durch Bildung des Exklusiv-Oders der benachbarten Bits innerhalb jedes gespeicherten Zeitpunktes ermittelt.

Abbildung 4.18 zeigt die drei möglichen Phasenverhältnisse, wobei für jeden Zustand zwei Dreiergruppen beispielhaft ausgewählt wurden, die eine beliebige Gruppe der vier möglichen sein könnten. Die Detektion läßt sich offensichtlich durch Exklusiv-Oder-Bildung der jeweils benachbarten Bits in den Dreiergruppen durchführen. Liefert das erste XOR-Gatter eine Eins, ist der Oszillator zu früh (Teilabbildung a), liefert das zweite XOR-Gatter eine Eins, dann ist er zu spät (Teilabbildung c). Ist der Oszillator in Phase, weisen beide XOR-Gatter eine logische Null auf (Teilabbildung b).

Dem Zustand „zu früh“ wird der Wert -1, „in Phase“ der Wert 0 und „zu spät“ der Wert +1 zugeordnet. Um den Eingangswert für das digitale Schleifenfilter zu bilden, addiert man zuerst alle Phasenwerte der verschiedenen im letzten Satz von acht Datenbits aufgetretenen Flanken. Das Ergebnis wird sodann durch die Anzahl der Flanken geteilt, so daß ein gebrochenrationaler Wert zwischen -1 und +1 ermittelt wird.

Diese Mittelung gewichtet eine einzige innerhalb von acht Bits aufgetretene Flanke, die beispielsweise gegenüber dem Oszillator zu früh kam, gleich dem Auftreten von mehreren ebenfalls verfrühten Flanken, weil in beiden Fällen die tatsächlich vorhandene Phasendifferenz gleich ist. Andernfalls wäre das Ergebnis der Phasendetektion von der doch zufälligen Zahl von Flanken im Datensignal abhängig.

Werden innerhalb eines Satzes von acht Bits mehrere Flanken detektiert, die teilweise eine bestimmte Phasendifferenz anzeigen, teilweise eine andere, so liegt die wirklich vorhandene wahrscheinlich zwischen beiden und wird aufgrund von Jittereinflüssen etc. für die einzelnen Flanken unterschiedlich detektiert. In diesem Fall bewirkt die Mittelwertbildung wiederum eine bessere Repräsentation der Phasendifferenz als die durch die einfache Summe.

Hierzu ein Beispiel: treten vier Flanken auf, von denen jeweils zwei als „zu früh“ und zwei als „in Phase“ detektiert werden, gleiche die einfache Summe derjenigen des Falles zweier, zu früher Flanken. Offensichtlich liegt im zweiten Fall aber eine Phasendifferenz von etwa 33 ps zwischen Oszillator und Datensignal vor, während sie im ersten Fall zwischen 33 ps und Null liegt und somit durch den kleineren Wert der Mittelwertbildung besser abgebildet wird.

Die Mittelung entspricht einer Divisionsoperation, die bekanntlich schwierig digital durchzuführen ist. Bedenkt man aber, daß im Zähler lediglich die ganzzahligen Werte zwischen  $-9$  und  $+9$ <sup>13</sup> sowie im Nenner die Werte 0 bis 9 auftreten können<sup>14</sup>, dann erkennt man, daß sich die Divisionsoperation durch eine sog. „Look-up“-Tabelle ersetzen läßt. Die Tabelle 4.1 zeigt die auf zwei Nachkommastellen gerundeten Werte der „Look-up“-Tabelle.

Die kleinste Differenz zwischen zwei möglichen Tabellenwerten ist  $(\frac{1}{8} - \frac{1}{9})$  und beträgt ungefähr 0,014. Der Wert liegt zwischen  $\frac{1}{64}$  und  $\frac{1}{128}$ , was bedeutet, daß die Werte der Tabelle ausreichend genau in einer binären Achtbitrepräsentation dargestellt werden können. Die Tabelle läßt sich als verdrahtetes ROM mit einer Dimensionierung von 10 x 18 Zellen aufbauen. Fast die Hälfte der Zellen kann niemals angewählt werden, weil der Betrag des Zählers immer kleiner oder gleich dem Nenner sein muß. Dadurch reduziert sich der Aufwand des ROMs auf 100 Zellen mit jeweils maximal acht Transistoren.

Der Phasenfehler kann Werte zwischen 1 und -1 für eine Bitlänge annehmen. Weil dabei eine Phasendifferenz von  $\frac{1}{3}$  bis  $-\frac{1}{3}$  Bits beschrieben wird, ist der für die Übertragungsfunktion des Phasendetektors eingeführte Proportionalitätsfaktor  $K_D$  gleich 3. Die Phasendetektion wird durch einige Dutzend Exklusiv-Oder-Gatter, eine Addition von neun vorzeichenbehafteten Zweibitwerten sowie eine Tabelle realisiert. Auch hier ist der notwendige Aufwand gering.

### Die Frequenzdetektion

Die Frequenzdetektion ist notwendig, um die Frequenz des bei Inbetriebnahme des Empfängers noch freischwingenden lokalen Oszillators in die Nähe der Datenrate des empfan-

---

<sup>13</sup>Die Neun bezieht sich auf den Spezialfall, in dem die Datenrate höher liegt als die Oszillatorfrequenz mal  $N$

<sup>14</sup>Der Wert 0 kann im Nenner nur dann auftreten, wenn auch der Zähler 0 ist. Das Ergebnis der Mittelung ist dann gleich 0 zu setzen.

Zähler\Nenner	0	1	2	3	4	5	6	7	8	9
9	x	x	x	x	x	x	x	x	x	1
8	x	x	x	x	x	x	x	x	1	0,89
7	x	x	x	x	x	x	x	1	0,88	0,78
6	x	x	x	x	x	x	1	0,86	0,75	0,67
5	x	x	x	x	x	1	0,83	0,71	0,63	0,56
4	x	x	x	x	1	0,8	0,67	0,57	0,5	0,44
3	x	x	x	1	0,75	0,6	0,5	0,43	0,38	0,33
2	x	x	1	0,67	0,5	0,4	0,33	0,29	0,25	0,22
1	x	1	0,5	0,33	0,25	0,2	0,17	0,14	0,13	0,11
0	0	0	0	0	0	0	0	0	0	0
-1	x	-1	-0,5	-0,33	-0,25	-0,2	-0,17	-0,14	-0,13	-0,11
-2	x	x	-1	-0,67	-0,5	-0,4	-0,33	-0,29	-0,25	-0,22
-3	x	x	x	-1	-0,75	-0,6	-0,5	-0,43	-0,38	-0,33
-4	x	x	x	x	-1	-0,8	-0,67	-0,57	-0,5	-0,44
-5	x	x	x	x	x	-1	-0,83	-0,71	-0,63	-0,56
-6	x	x	x	x	x	x	-1	-0,86	-0,75	-0,67
-7	x	x	x	x	x	x	x	-1	-0,88	-0,78
-8	x	x	x	x	x	x	x	x	-1	-0,89
-9	x	x	x	x	x	x	x	x	x	-1

**Tabelle 4.1:** Tabelle mit den möglichen Ausgangswerten der Phasendetektion

genen Datensignales zu bringen, so daß der Phasenregelkreis allein mit Hilfe des Phasendetektors auf das Datensignal einrasten kann.

Wie im Abschnitt 4.3.3 ab Seite 80ff. dargelegt, läßt sich die Frequenzdetektion am einfachsten durchführen, indem das Phasenverhältnis zwischen zwei zu unterschiedlichen Zeitpunkten innerhalb einer Taktphase vorgenommenen Phasendetektionen über die Zeit betrachtet wird.

Ein Vorteil des hier konzipierten Empfängers ist, daß die Phasendifferenz durch Vergleich mit drei Phasensignalen des Oszillators automatisch in verschiedene Zustände aufgelöst wird. Dadurch entfällt die Notwendigkeit eines zweiten Phasendetektors, der das Datensignal mit einem Quadraturtakt vergleicht wie in dem in Abbildung 4.6 dargestellten Frequenzdetektor.

Die Polarität des Frequenzunterschiedes ergab sich in diesem zuvor dargestellten Beispiel aus der Phasenlage der Ausgangssignale der beiden Phasendetektoren zueinander und wurde durch einen dritten Phasendetektor festgestellt. Da auch dieser lediglich eine Phasendifferenz ermittelt, in der neuartigen Empfängerarchitektur die Phasenlage aber bereits als digitaler Wert vorliegt, braucht nur die Differenz zweier aufeinanderfolgen-

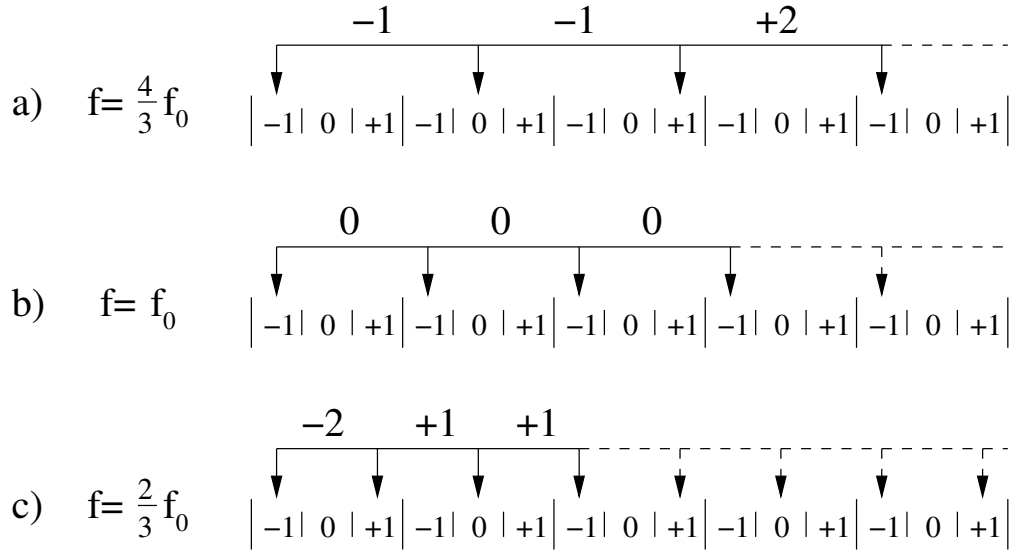


Abbildung 4.19: Zur Frequenzdetektion

der Phasendetektionsergebnisse ermittelt zu werden. Dies läßt sich auch mathematisch einfach belegen.

Weil

$$\Delta f = \frac{d\Phi(t)}{dt} \approx \frac{\Delta\Phi}{\Delta t} \quad (4.1)$$

und  $\Delta t$  zwingend  $>0$  ist, reicht die Bestimmung des Vorzeichens von  $\Delta\Phi$  für die qualitative Frequenzdetektion aus.

Im vorhergehenden Abschnitt zur Phasendetektion wurde dargelegt, daß sie für jede im Datensignal aufgetretene Flanke einen Wert von entweder  $-1$ ,  $0$  oder  $+1$  liefert. Die Differenz zweier aufeinanderfolgender Ergebnisse kann deshalb Werte zwischen  $-2$  und  $+2$  betragen. Die Abbildung 4.19 zeigt die Bestimmung des Frequenzverhältnisses anhand dreier Beispiele. Im ersten Fall liegt die Frequenz des Oszillators zu hoch, im zweiten stimmt sie mit der Datenrate des empfangenen Signales überein und im dritten liegt sie zu niedrig.

In den drei verschiedenen Fällen gibt es jeweils drei mögliche Wertepaare, die in die Subtraktion eingehen. Statistisch gesehen treten sie gleich häufig auf, unabhängig davon, wieviele Bits zwischen jeweils zwei Flanken liegen. Stimmen Oszillatorfrequenz und Datenrate überein, so betragen alle drei Differenzen Null (siehe Teilabbildung 4.19 b). Für den ersten und dritten Fall trifft dies nicht zu.

Betrachtet man die Ergebnisse der einzelnen Wertepaare für diese beiden Fälle genauer, so zeigt sich, daß jeweils zwei den gleichen Wert aufweisen, nämlich  $-1$  bei zu hoher Oszillatorfrequenz und  $+1$  bei zu niedriger. Diese beiden Werte stimmen im Vorzeichen mit der notwendigen Frequenzkorrektur überein. Der Wert des dritten Wertepaares ist jedoch jeweils doppelt so hoch und im Vorzeichen entgegengesetzt, so daß sich alle drei Werte der beiden Beispielfälle zu Null addieren. Diese Abweichung in Vorzeichen und Betrag würde die Frequenzdetektion unmöglich machen, da die drei verschiedenen Fälle im statistischen Mittel vieler aufeinander folgender Frequenzbestimmungen Null ergäben und sich somit nicht voneinander unterscheiden ließen.

Die Lösung dieses Problems ist indes denkbar einfach: berechnet man die Differenz mit einem Überlauf bei den Werten  $+1$  und  $-1$ , ergibt sich im Fall der ersten Teilabbildung statt  $+2$  der Wert  $-1$ , im Fall der dritten Teilabbildung statt  $-2$  der Wert  $+1$ . Diese Ergebnisse stimmen mit den anderen beiden Wertepaaren überein und die Differenz zwischen Oszillatorfrequenz und Datenrate kann zuverlässig bestimmt werden.

Der Tatsache, daß nicht bekannt ist, ob zwischen zwei einander folgenden Flanken nur ein oder aber mehrere Bits lagen, wird Rechnung getragen, indem automatisch eine Modulo-Operation durchgeführt wird, wenn einfach nur die zu den Flanken korrespondierenden Werte subtrahiert werden, gleich wieviele Dreiergruppen in der Phasendetektion zwischen ihnen liegen.

Die Sicherheit der Frequenzbestimmung erhöht sich, wenn die Ergebnisse von eventuell mehreren in einem Datensatz von acht Bits aufgetretenen Flanken summiert werden. Das Vorzeichen des Resultates zeigt dann die Richtung der notwendigen Frequenzkorrektur an. Auf diese Weise können Einflüsse des Datenjitters auf die Frequenzdetektion vermindert werden.

Da die Frequenzdetektion nur bei Inbetriebnahme des Empfängers notwendig ist, kann einem einfachen Algorithmus gefolgt werden. Man addiert das Ergebnis der Frequenzdetektion einer zuvor festgelegten Anzahl Flanken im Datensignal und bestimmt so das Vorzeichen der Frequenzdifferenz mit großer Sicherheit. Dann wird der Oszillator, der einen diskreten Regeleingang für die Grobeinstellung seiner Frequenz besitzt, einen Schritt in die entsprechende Richtung verstellt und der Akkumulator für die Frequenzbestimmung zurückgesetzt. An dieser Stelle wiederholt sich der Algorithmus.

Übersteigt für die festgelegte Anzahl von Flanken im Datensignal der Inhalt des Akkumulators einen bestimmten Wert nicht, dann befindet sich die Oszillatorfrequenz so nahe an der Datenrate, daß der Phasenregelkreis allein mit Hilfe der Phasendetektion auf das Datensignal einrasten kann. Die Grobeinstellung des Oszillators wird festgehalten und die Phasendetektion, die die Feineinstellung des Oszillators kontrolliert, gestartet.



#### 4.4.7 Der Phasenregelkreis

Die widersprüchlichen durch die optischen Standards vorgegebenen Anforderungen an die Regelbandweite der Takt-/Datenrückgewinnung, die im Text zur Jitter-Tolerance und zum Jitter-Transfer auf den Seiten 17f. erläutert wurden, werden in der neuartigen Empfängerstruktur versöhnt. Dies setzt die Anwendung eines zweiten Regelmechanismus voraus, der die Phasendifferenz zwischen dem Datensignal und dem Oszillator ausgleicht.

Der Phasenregelkreis der Taktrückgewinnung koppelt die Oszillatorfrequenz an das Datensignal und besitzt eine Schleifenbandweite von nur 120 kHz, um möglichst viel Eingangsruschen des Datensignales zu unterdrücken. Da das Datensignal aber starken Störungen durch Jitter unterworfen ist, könnte die PLL dem Eingangssignal wegen der beschränkten Schleifenbandweite nicht schnell genug folgen, um die Datenrückgewinnung zuverlässig durchzuführen. Deshalb wird ein zweiter Phasenregelkreis mit einer Schleifenbandweite von vier oder mehr MHz aufgebaut, der aber zum größten Teil aber mit dem ersten Regelkreis identisch ist.

Der zusätzliche Schaltungsaufwand für diese Doppel-PLL ist viel geringer als in herkömmlichen Empfänger/Senderarchitekturen, die gewöhnlich zwei oder sogar drei komplette PLLs einsetzen, um die verschiedenen Anforderungen zu erfüllen. Die wichtigsten Vorteile der neuen Regelschleife bestehen in der gesenkten Komplexität und Leistungsaufnahme sowie der Tatsache, daß auf dem Chip nur ein einziger Oszillator zum Senden und Empfangen der Daten benötigt wird (siehe hierzu Abschnitt 4.4.8 ab Seite 117). Dies verhindert störende Effekte wie die Kreuzmodulation durch Kopplungen über das Chipsubstrat, die immer dann auftreten, wenn sich mehrere Oszillatoren auf einem Chip befinden, und die Phasenreinheit des Oszillatorsignales beeinträchtigen.

Abbildung 4.20 zeigt das Blockdiagramm der Doppel-PLL. In ihr sind die zwei Regelkreise zu erkennen. Der Regelkreis zur Taktrückgewinnung wird durch die Blöcke des äußeren Umlaufes, der dem Datensignal zur Datenrückgewinnung schnell folgende, durch die kleinere Masche in der unteren Bildhälfte gebildet.

Für den Aufbau eines solchen doppelten Regelkreises müssen, wie eingangs erwähnt, zwei Regelmechanismen zum Ausgleich der Phasendifferenz zwischen Oszillortakt und Datensignal vorhanden sein. Der äußere Regelkreis kontrolliert unmittelbar den Oszillortakt über den Regeleingang des VCOs, während der andere der sich schnell ändernden Phasenlage des Datensignales über einen Mechanismus folgt, der im Blockdiagramm als „Rotierer“ bezeichnet wurde.

Der Rotierer ist ein einfacher digitaler Block, in dem die von den Registern gespeicherten Oszillatorzustände um so viele Bitpositionen nach links oder rechts rotiert werden, wie der Regeleingang des Rotierers vorgibt. Weil aber jeweils nur eine Hälfte der Phasenausgänge

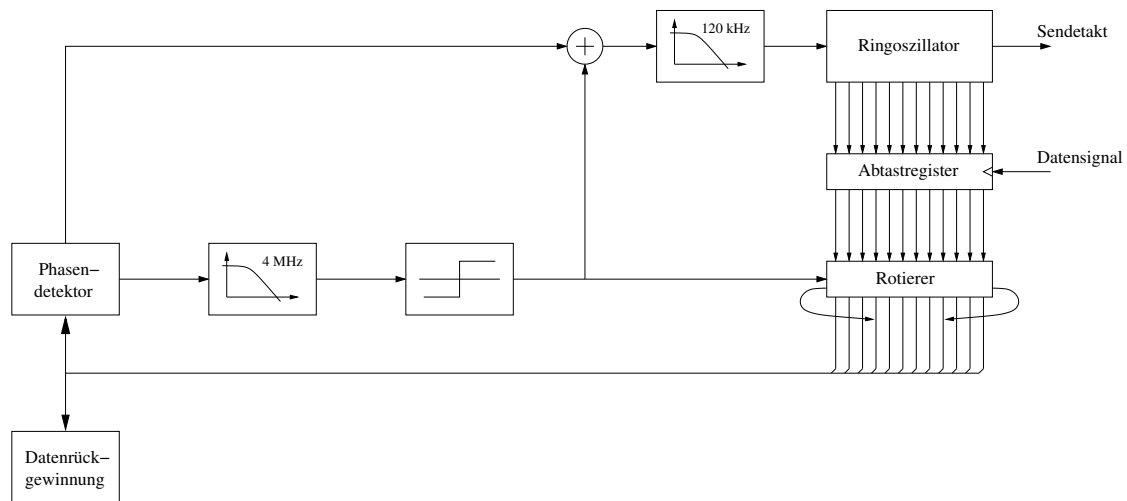


Abbildung 4.20: Blockdiagramm der Doppel-PLL

des Oszillators abgetastet wird, müssen die links oder rechts herausrotierten Bits am anderen Ende invertiert an den Oszillatorzustand angefügt werden. Da jede Bitposition der gespeicherten Oszillatorzustände ein Drittel der Länge eines Datenbits bedeutet, kann durch das Rotieren indirekt eine Phasenverschiebung des Oszillators vor der Auswertung der gespeicherten Oszillatorzustände bewirkt werden. Weil jedoch diese Phasenverschiebung nicht wirklich auf den Oszillator angewendet wird, bleibt er unbeeinflusst von den dynamischen Verhältnissen des schnelleren Regelkreises.

Den Rotierer regelt ein digitales Schleifenfilter, das die Schleifenbandweite dieses Regelkreises auf mindestens 4 MHz festlegt. Weil aber der Regelkreis dem Datensignal schneller folgt als die Taktrückgewinnung, wird in dem hinter dem Rotierer im Ring platzierten Phasendetektor eine viel geringere Phasendifferenz festgestellt, als aus Sicht des langsameren Regelkreises eigentlich vorhanden ist. Deshalb wird der vom Filter des schnellen Regelkreises ausgegebene Stellwert, der die aktuell auszugleichende Phasendifferenz zum Datensignal darstellt, zum Ausgangswert des Phasendetektors addiert. Der Eingangswert des ebenfalls digital aufgebauten Schleifenfilters des äußeren Regelkreises, das die Schleifenbandweite der Taktrückgewinnung auf 120 kHz begrenzt, erhält so den Wert der wirklich zwischen Oszillator und Datensignal vorhandenen Phasendifferenz.

Der Regelkreis zur Taktrückgewinnung wird durch das Vorhandensein des zweiten schnelleren Regelkreises in keiner Weise beeinflusst. Das liegt daran, daß jener die zwischen dem Datensignal und dem Oszillatortakt vorhandene Phasendifferenz durch den Rotierer ausgleicht, also die vorhandene Phasendifferenz vom Oszillatorsignal subtrahiert, nur um sie hinter dem Phasendetektor wieder durch Addition des Wertes in den äußeren Kreis ein-

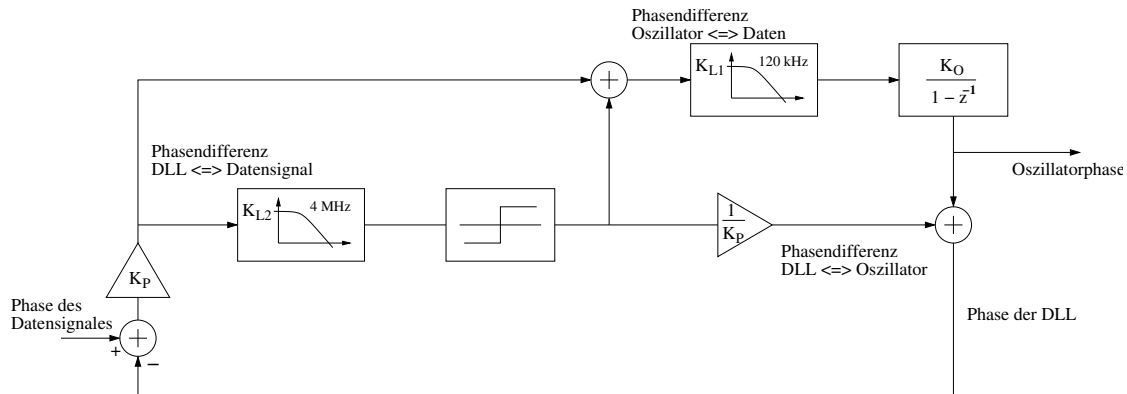


Abbildung 4.21: Die Doppel-PLL als Regelkreis

zuspeisen. Deshalb kann die Stabilität der äußeren PLL genauso leicht garantiert werden wie für jeden anderen Regelkreis zweiter Ordnung auch.

Der zweite Regelkreis ist strenggenommen keine PLL, sondern eine DLL (engl.: „Delay-locked Loop“). Der Unterschied besteht darin, daß in einer DLL kein Oszillator geregelt wird, sondern lediglich eine Verzögerung bzw. Phasenverschiebung. Ein Oszillator kann in der Phasendimension als Integrator betrachtet werden, weil ein konstanter Frequenzunterschied eine linear steigende Phasendifferenz bewirkt. Die Übertragungsfunktion eines Oszillators wird deshalb in der  $z$ -Ebene durch die Gleichung  $H_O(z) = \frac{K_O}{1-z^{-1}}$  beschrieben und stellt eine Funktion erster Ordnung mit einem Pol bei  $z^{-1} = 1$ , also Gleichspannung, dar. Der Rotierer dagegen bewirkt eine einfache Phasenverschiebung und wird durch eine Subtraktion des Regelwertes von der Phasenlage des Oszillators repräsentiert. Die Ordnung einer DLL ist deshalb gleich der des eingesetzten Schleifenfilters, während die einer PLL um Eins größer ist als die Ordnung ihres Schleifenfilters.

Wird das Schleifenfilter der DLL als Tiefpaßfilter erster Ordnung aufgebaut, dann besitzt die DLL eine Regelcharakteristik erster Ordnung und ist somit stabil. Allerdings muß bedacht werden, daß das Oszillatorsignal, dem die DLL folgt, selbst Produkt einer Regelschleife ist und somit durch seine dynamischen Gegebenheiten die Stabilität der DLL beeinträchtigen könnte. Im vorliegenden Fall ist dies aber auszuschließen, weil das Oszillatorsignal durch eine Regelschleife mit einer deutlich niedrigeren Schleifenbandweite kontrolliert wird und somit die DLL mit ihrer größeren Bandweite dem Oszillatorsignal immer folgen kann. Aus Sicht der DLL ist eine Phasenverschiebung aufgrund von Jittereinflüssen auf das Datensignal nicht von einer durch die PLL bewirkten Veränderung der Oszillatorphase unterscheidbar.

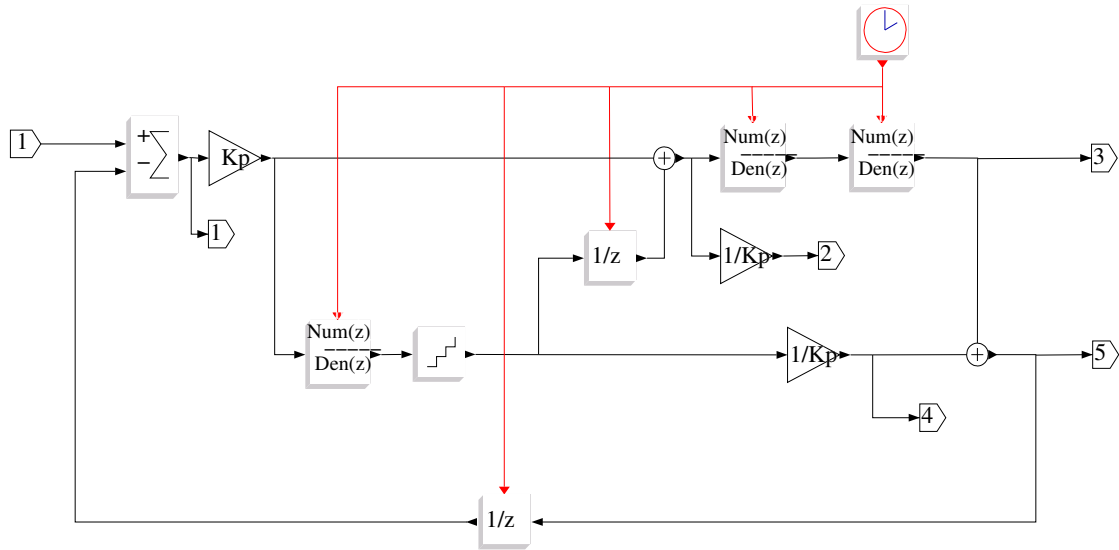


Abbildung 4.22: Die Doppel-PLL in der Darstellung des Systemsimulators `sicos`

Die Abbildung 4.21 zeigt eine Darstellung der Doppel-PLL als Regelkreis. Der Rotierer wurde durch eine Addition<sup>15</sup> des Stellwertes zum Regelwert in der Schleife ersetzt, der Phasendetektor durch eine Differenzbildung zwischen Regelwert und der Phase des Datensignales gefolgt von einer Verstärkung um den Faktor  $K_P$ . Der Ausgangswert des Schleifenfilters der DLL wird vor dem Steuereingang des Rotierers gerundet, weil der Rotierer nur Phasenverschiebungen um Vielfache einer Drittelbitlänge zuläßt. Der Quantisierer im Blockdiagramm symbolisiert die Rundung. Die Abtastregister fehlen in der Abbildung, weil sie keine regeltechnische Auswirkung haben; schließlich wird erst im Phasendetektor die Phase der gespeicherten Oszillatorzustände ausgewertet, nachdem diese durch den Rotierer entsprechend dem Stellwert der DLL verändert wurden.

Eine Implementation des Regelkreises mit dem Systemsimulator `sicos` aus dem mathematisch-naturwissenschaftlichen Programmpaket `scilab` [18] ist in Abbildung 4.22 zu sehen. Diese Implementation wird im folgenden für Simulationen zur Funktionsweise der Doppel-PLL benutzt. Man erkennt drei Blöcke, die eine Übertragungsfunktion in der  $z$ -Ebene repräsentieren. Ein Vergleich mit der vorhergehenden Abbildung zeigt, daß es sich um die beiden digitalen Filter und den Oszillator handelt. Die Übertragungsfunktion des Oszillators wurde bereits angegeben, die der digitalen Filter müssen noch hergeleitet werden.

<sup>15</sup>Der Stellwert muß addiert werden, weil sich der Rotierer vor dem Phasendetektor in der Rückkopplungsschleife befindet. Im Phasendetektor wird der summierte rückgekoppelte Wert von der Phase des Datensignales subtrahiert, so daß sich das Vorzeichen wie gewünscht umkehrt.

Da die Regelfunktion der PLL nur eine Resonanzspitze von maximal 0,1 dB besitzen darf, was einer Verstärkung um lediglich etwa ein Prozent entspricht, muß sie eine im Durchlaßbereich flache Charakteristik ohne Welligkeit aufweisen. Für die Übertragungsfunktion eines Systems zweiter Ordnung existieren allgemein bekannte Optimierungen auf verschiedene Charakteristiken. Die Butterworth-Charakteristik ist hinsichtlich eines maximal flachen Verlaufes im Durchlaßbereich optimiert. Die Übertragungsfunktion für die Phase des Datensignales auf den Oszillatorausgang und somit die Jitter-Transferfunktion soll deshalb eine Butterworth-Charakteristik besitzen.

Die Übertragungsfunktion eines digitalen Filters läßt sich in `scilab` mit der Funktion `iir()` bestimmen. Der Aufruf `iir(2,'lp','butt',[.12/1250,0],[0,0])` liefert eine Übertragungsfunktion mit den entsprechenden Filterkoeffizienten für einen Infinite-Impulse-Response Tiefpaß zweiter Ordnung gemäß der Butterworth-Charakteristik mit der relativen Grenzfrequenz  $\frac{120 \text{ kHz}}{1250 \text{ MHz}}$  und einer Welligkeit von Null:

```
--> iir(2,'lp','butt',[.12/1250,0],[0,0])
ans =
                                     2
    9.092E-08 + 1.818E-07z + 9.092E-08z
-----
                                     2
    0.9991473 - 1.999147z + z
```

Es zeigt sich, daß die angestrebte Butterworth-Charakteristik zweiter Ordnung allgemein formuliert werden kann:

$$H(z) = \frac{a + 2a \cdot z^{-1} + a \cdot z^{-2}}{1 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2}} \quad (4.2)$$

Wenn die sog. Open-Loop-Funktion, also die Übertragungsfunktion der geöffneten negativ rückgekoppelten Schleife,  $OLF(z)$  ist, dann läßt sich die Übertragungsfunktion  $H(z)$  der geschlossenen Schleife schreiben:

$$H(z) = \frac{OLF}{1 + OLF} \quad (4.3)$$

$OLF(z)$  wird allgemein angesetzt:

$$OLF(z) = \frac{c_0 + c_1 \cdot z^{-1} + c_2 \cdot z^{-2}}{1 + d_1 \cdot z^{-1} + d_2 \cdot z^{-2}} \quad (4.4)$$

Dieser Ansatz eingesetzt in Gl. 4.3 ergibt:

$$H(z) = \frac{c_0 + c_1 \cdot z^{-1} + c_2 \cdot z^{-2}}{(1 + c_0) + (c_1 + d_1) z^{-1} + (c_2 + d_2) z^{-2}} = \frac{\frac{c_0}{1+c_0} + \frac{c_1}{1+c_0} \cdot z^{-1} + \frac{c_2}{1+c_0} \cdot z^{-2}}{1 + \frac{c_1+d_1}{1+c_0} \cdot z^{-1} + \frac{c_2+d_2}{1+c_0} \cdot z^{-2}} \quad (4.5)$$

Aus dem Koeffizientenvergleich zwischen den Gleichungen 4.2 und 4.5 folgt:

$$\begin{aligned} c_0 &= \frac{c_1}{2} = c_2 = \frac{a}{1-a} = c \\ d_1 &= b_1 + \frac{b_1 a}{1-a} - \frac{2a}{1-a} = \frac{b_1 - 2a}{1-a} \\ d_2 &= b_2 + \frac{b_2 a}{1-a} - \frac{a}{1-a} = \frac{b_2 - a}{1-a} \end{aligned} \quad (4.6)$$

Die Open-Loop-Funktion setzt sich aus den Übertragungsfunktionen des Phasendetektors, des digitalen Filters  $L(z)$  und des Oszillators  $H_O(z)$  zusammen. Da diese Blöcke in Serie geschaltet sind, ergibt sich  $OLF(z)$  als Produkt der drei Teilübertragungsfunktionen:

$$OLF(z) = K_P \cdot L(z) \cdot H_O(z) \quad (4.7)$$

Für  $L(z)$  gelöst:

$$\begin{aligned} L(z) &= \frac{OLF(z)}{K_P \cdot H_O(z)} \\ &= \frac{1 - z^{-1}}{K_P \cdot K_O} \cdot \frac{\frac{a}{1-a} + \frac{2a}{1-a} \cdot z^{-1} + \frac{a}{1-a} \cdot z^{-2}}{1 + \frac{b_1-2a}{1-a} \cdot z^{-1} + \frac{b_2-a}{1-a} \cdot z^{-2}} \\ &= \frac{1}{K_P \cdot K_O} \cdot \frac{\frac{a}{1-a} + \frac{a}{1-a} \cdot z^{-1} - \frac{a}{1-a} \cdot z^{-2} - \frac{a}{1-a} \cdot z^{-3}}{1 + \frac{b_1-2a}{1-a} \cdot z^{-1} + \frac{b_2-a}{1-a} \cdot z^{-2}} \\ &= K_L \cdot \frac{c + c \cdot z^{-1} - c \cdot z^{-2} - c \cdot z^{-3}}{1 + d_1 \cdot z^{-1} + d_2 \cdot z^{-2}} \end{aligned} \quad (4.8)$$

$K_L$  ist die Verstärkung des Gleichanteils des Filters und kann frei gewählt werden. Sie ist gleich  $\frac{1}{K_P \cdot K_O}$  zu setzen, damit sich die Verstärkung des Gleichanteils der PLL zu Eins ergibt, was Bedingung dafür ist, daß die PLL auch eine konstante Phasendifferenz zwischen den Daten und dem Oszillatortakt auszugleichen vermag.

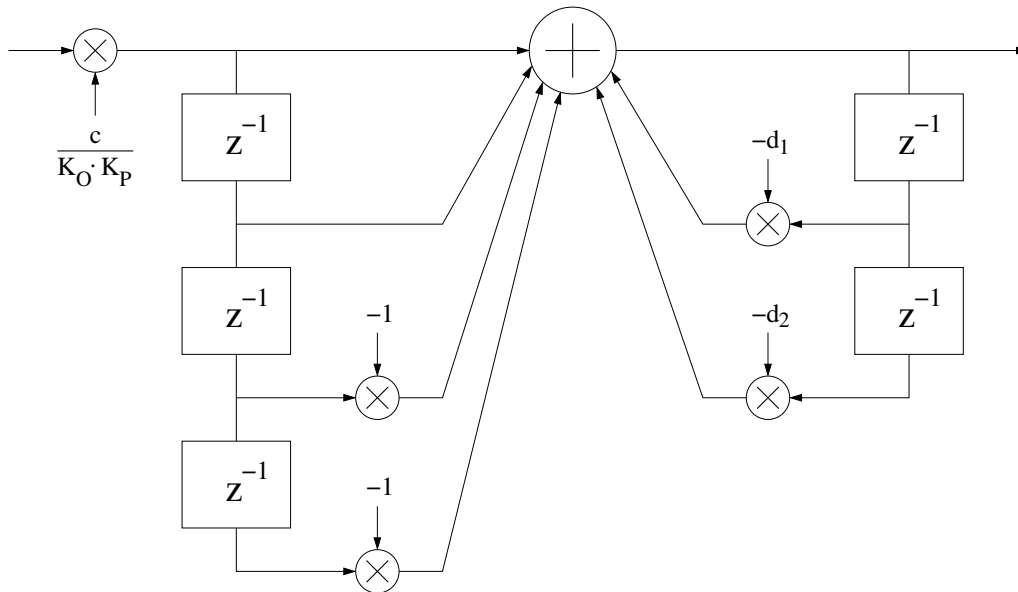


Abbildung 4.23: Schaltungsstruktur des digitalen Schleifenfilters der PLL

Die Gleichung 4.8 besitzt drei Nullstellen und zwei Pole. Sie läßt sich in die in Abbildung 4.23 dargestellte Filterstruktur überführen. Das Filter kann mit drei Multiplikationen und mehreren Additionen umgesetzt werden. Die Multiplikationen werden mit festen Faktoren ausgeführt, was zahlreiche Optimierungen zuläßt und somit die Realisierung in Hardware erheblich vereinfacht. Gleichwohl bedingen die Filterkoeffizienten wegen der im Verhältnis zur Taktfrequenz von 1,25 GHz sehr geringen Grenzfrequenz des Filters eine sehr hohe Präzision der Berechnung. Außerdem begrenzt die Rückkopplenschleife in der rechten Hälfte der Abbildung 4.23 den Durchsatz des digitalen Filters. Deshalb ist es praktisch unumgänglich, die Taktrate der Regelschleife zu senken, um auf ein digitales Filter mit einer deutlich geringeren Anforderung an die Rechengenauigkeit und einer wegen der geringeren Präzision und der gesenkten Taktgeschwindigkeit erheblich vereinfachten Realisierung zu kommen. Eine Herabsetzung der Taktrate auf ein Viertel bietet sich an; bei 312 MHz kann das Filter unter Verwendung von schnellen CMOS-Schaltungstechniken realisiert werden, ohne den Leistungsverbrauch zu groß werden zu lassen.

Analysiert man die Gleichung 4.8, dann erkennt man, daß  $L(z)$  eine Nullstelle besitzt, die den Pol des Oszillators bei  $z^{-1} = 1$  kompensiert. Dies ist notwendig, da die Butterworth-Charakteristik keinen Pol an diesem Punkt zuläßt. Durch diese Kompensation wird die Ordnung des Regelkreises effektiv um eins gesenkt, durch Einführung eines zweiten Poles in der Filterfunktion jedoch wieder auf zwei erhöht.

Stattdessen könnte auch der für die Herleitung der Filterfunktion in Gleichung 4.2 gemachte Ansatz für die Butterworth-Charakteristik auf einen Ansatz erster Ordnung reduziert werden. Diese Maßnahme würde nach ähnlicher Herleitung der Filterfunktion  $L(z)$  in ein einfacheres Filter münden, das mit nur zwei Nullstellen und einem Pol eine weniger aufwendige Umsetzung als Schaltung erlaubt. Auch die auf Seite 17 genannte Vorgabe der optischen Übertragungsstandards schreibt für die Jitter-Transfer-Charakteristik lediglich eine Tiefpaßfunktion von *mindestens* erster Ordnung vor, bedingt also keinesfalls einen Phasenregelkreis zweiter Ordnung.

Der einfacheren Realisierung eines digitalen Tiefpaßfilters erster Ordnung steht allerdings die geringere Filterwirkung auf das Oszillatorrauschen gegenüber. Da das Phasenrauschen des lokalen Oszillators durch eine Hochpaßfunktion mit gleicher Ordnung und Grenzfrequenz wie die der Schleifenfunktion gefiltert wird, überträgt ein Schleifenfilter erster Ordnung mehr Rauschen auf den Taktausgang des Phasenregelkreises als eines höherer Ordnung. Die vereinfachte Umsetzung des digitalen Filters würde deshalb den Entwurf der für die Einhaltung der Vorgaben zur Jitter-Generation entscheidenden kritischen Komponenten, des Oszillators und des Ausgangstreibers, erschweren, was kaum wünschenswert sein kann.

Die Schleifenbandweite der DLL zur Datenrückgewinnung wird durch die Vorgaben zur Jitter-Tolerance bestimmt. Ihnen zufolge muß die Datenrückgewinnung bis zu einer Jitterfrequenz von 4 MHz Störungen des Datensignales folgen können. Dies bedingt eine Schleifenbandweite von mindestens 4 MHz. Je höher die Schleifenbandweite, desto größer erscheint die Öffnung des Augendiagrammes in der Datenrückgewinnung und desto geringer ist die Bitfehlerrate.

Doch sind der Realisierung des digitalen Filters wiederum Grenzen gesetzt. So muß, damit die Abbildung von der Laplace-Ebene auf die  $z$ -Ebene mit nur geringem Fehler behaftet ist, die Taktfrequenz des Filters erfahrungsgemäß mindestens dreißigmal so hoch sein wie seine Grenzfrequenz. Auch hier begrenzt die in einem IIR-Filter unvermeidliche Rückkopplung innerhalb des Filters die Taktrate. Die Taktrate von 312 MHz erlaubt eine Grenzfrequenz der DLL von etwa 8 MHz, was ungefähr einer relativen Grenzfrequenz des Schleifenfilters von  $\frac{1}{40}$  entspricht. Die Simulation des doppelten Regelkreises wurde deshalb mit einer Grenzfrequenz von 8 MHz für die DLL und den fest vorgegebenen 120 kHz für die PLL der Taktrückgewinnung durchgeführt.

Die Übertragungsfunktion des digitalen Filters der DLL läßt sich nach dem oben verfolgten Verfahren durch Koeffizientenvergleich bestimmen. Die Abwesenheit eines weiteren Blockes mit Einfluß auf das Regelverhalten in der DLL vereinfacht die Berechnung. Wegen der Freiheit des Systementwicklers hinsichtlich des Schleifenfilters der DLL wird die einfachste Implementierung eines IIR-Filters gewählt, also die eines Filters erster Ordnung.



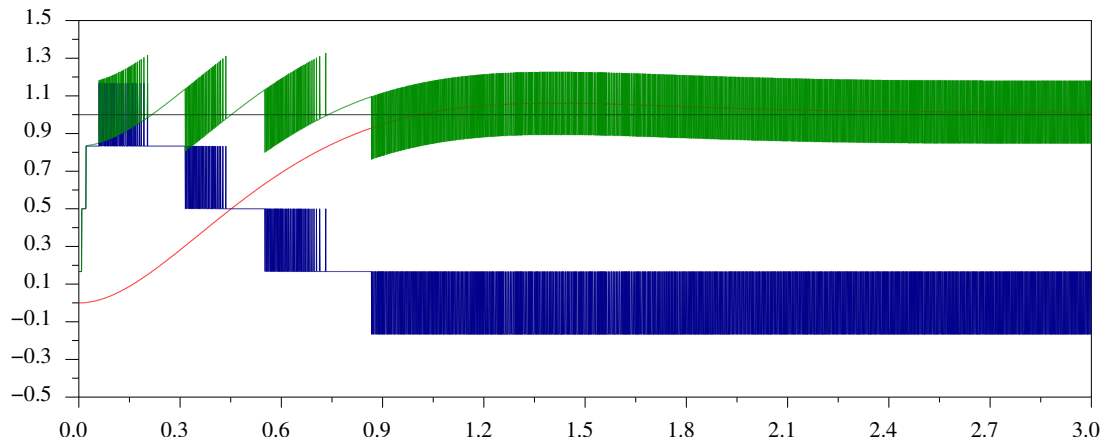


Abbildung 4.24: Antwort der Doppel-PLL auf einen Phasensprung

### Systemsimulation des doppelten Regelkreises

Die in Abbildung 4.22 auf S. 108 gezeigte Implementierung des doppelten Regelkreises wurde mit dem System Simulator `scicos` auf ihre systemtechnischen Eigenschaften untersucht. Es zeigte sich, daß der Regelkreis gemäß der Vorhersage stabil ist. Die erläuterte Dimensionierung der Schleifenfilter brachte die gewünschten Eigenschaften des Verhaltens des Regelkreises hinsichtlich der Übertragung des Eingangsjitters auf den Oszillatortakt sowie der Toleranz der Datenrückgewinnung diesem gegenüber.

Abbildung 4.24 zeigt die Antwort des doppelten Phasenregelkreises auf einen Phasensprung des Eingangsdatensignales. Die Einheit der Abszisse ist  $\mu\text{s}$ , die Ordinate ist einheitenlos. Die kantigen Verläufe der blauen und grünen Kurven entstehen durch die diskretisierte Aussteuerung der DLL. Da der Rotierer die gespeicherten Oszillatorzustände nur bitweise rotieren kann, ist die Ausgangsphase der DLL unstetig. Sie springt für kleine Phasenfehler zwischen zwei Zuständen hin- und her, was die blockartigen Kurvenverläufe erklärt.

Um die Übersichtlichkeit der Graphik zu gewährleisten, wurde die Schleifenbandweite der PLL der Taktrückgewinnung auf 500 kHz heraufgesetzt. Bei einer Schleifenbandweite von 120 kHz erschiene die rote Linie zeitlich gestreckt, wodurch sich der schnelle Phasenanstieg der DLL der Datenrückgewinnung schlecht erkennen ließe (blaue Linie). Die PLL antwortet wegen der geringeren Schleifenbandweite langsamer als die DLL. Sobald sie aber dem Phasensprung zu folgen beginnt, senkt die DLL ihre Auslenkung wieder, so daß die Summe der Phasen der beiden Teilregelkreise weiterhin mit der veränderten Eingangsphase übereinstimmt (grüne Linie).

Die Abbildungen 4.25 und 4.26 zeigen das Systemverhalten des Phasenregelkreises hin-

sichtlich der Jitter-Tolerance für vier Fälle. An den Eingang der PLL wurde eine sinusförmige Störung mit einer Amplitude von 1,5 UI gelegt. Ihre Frequenz beträgt in den vier Fällen 60 kHz, 400 kHz, 4 MHz sowie 8 MHz. Die Abszissen sind wiederum in  $\mu\text{s}$  aufgetragen, die Einheit der Ordinate ist das Unit-Intervall UI.

In der ersten Abbildung wurde die Reaktion der beiden Phasenregelkreise auf den Eingangsjitter aufgetragen. Die roten Kurven bedeuten wieder die Oszillatorphase, die blauen die Phase der DLL und die grünen die Summe der Auslenkungen der beiden Regelkreise. Die Eingangsstörung selbst wird in beiden Abbildungen schwarz dargestellt. Die zweite Abbildung zeigt die Phasendifferenzen zwischen dem gestörten Datensignal und den beiden Regelkreisen. Die Farbgebung ist analog zu der der vorhergehenden Abbildungen, wobei zu bedenken ist, daß sich die Auslenkungen von Oszillator und Rotierer summieren und die am Eingang der DLL wahrgenommene Phasendifferenz sich aus den genannten Teilen zusammensetzt. Die blauen Kurven der zweiten Abbildung stehen deshalb in Bezug zu sowohl den grünen als auch blauen Kurven der ersten.

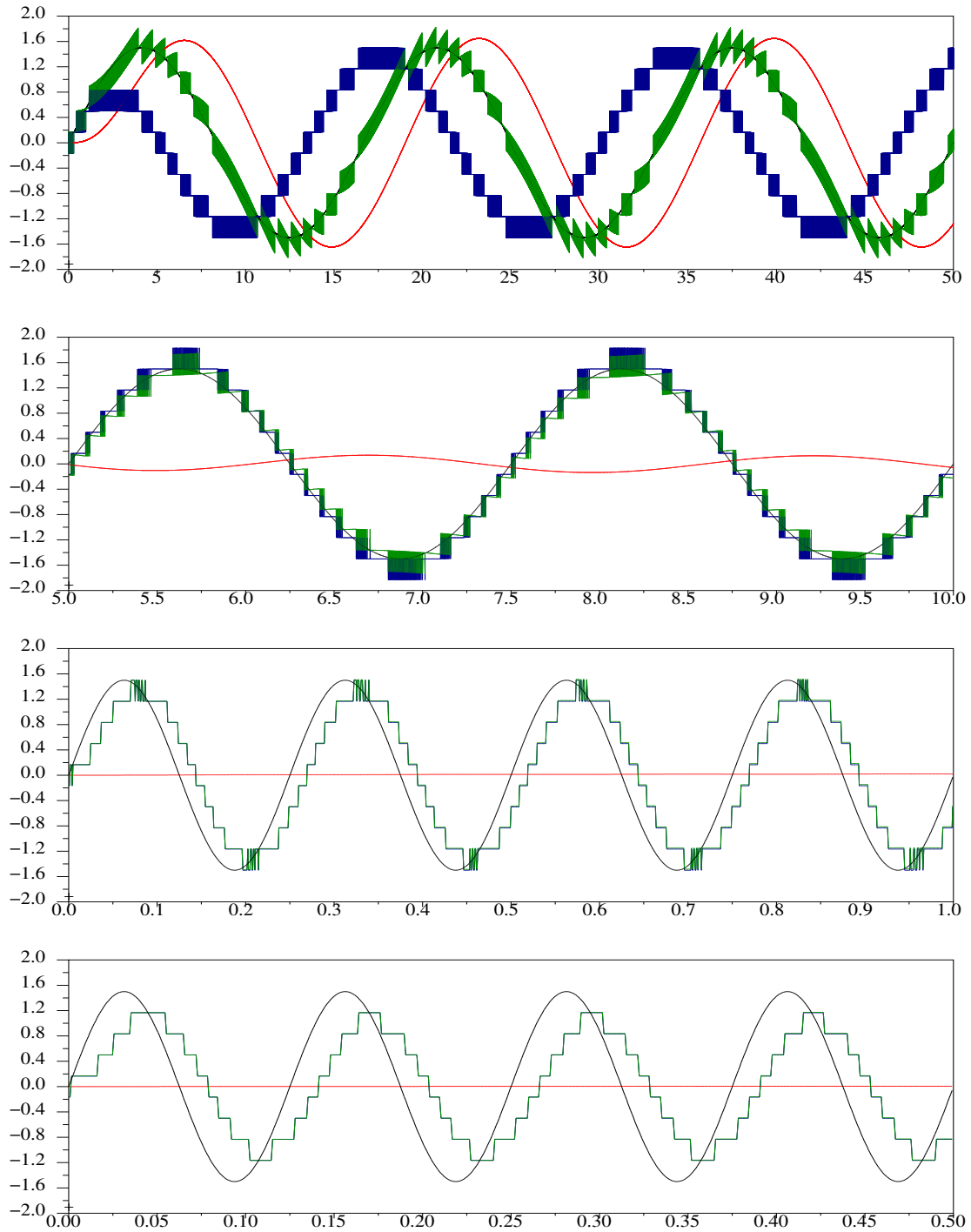
In der ersten Teilabbildung von Abb. 4.25 ist dargestellt, wie die Taktrückgewinnung den Oszillator regelt, um dessen Phase der des Datensignales anzupassen. Die Jitterstörung fällt mit ihrer Frequenz von 60 kHz in die Schleifenbandweite der Taktrückgewinnung, weshalb die rote Kurve im Unterschied zu den anderen Teilabbildungen eine deutlich sichtbare Auslenkung zeigt. Bei 60 kHz besteht jedoch bereits eine Phasenverschiebung von etwa  $45^\circ$  zwischen Daten- und Oszillatorphase. Sie wird durch die der Eingangsphase vorausseilenden DLL kompensiert, so daß die Summe der Auslenkungen von DLL und PLL der schwarz dargestellten Eingangsstörung exakt folgt. Die grüne Kurve überlagert deshalb die schwarze vollständig.

In der zweiten Teilabbildung zeigt sich bereits, daß die Taktrückgewinnung der Eingangsstörung kaum noch folgt. Die Filterwirkung der PLL ist deutlich sichtbar. Die rote Kurve zeigt nur eine geringe Auslenkung, die zudem um nahezu  $180^\circ$  gegenüber der Eingangsphase versetzt ist, was die Wirkung der Tiefpaßcharakteristik der PLL zweiter Ordnung illustriert. Die DLL folgt weiterhin dem Datensignal dicht. Da die PLL bei dieser Störfrequenz schon beinahe unbeeinflusst von der Störung ist, fallen jetzt die blaue und grüne Kurve fast zusammen.

Der wichtigste Unterschied zwischen der zweiten und dritten Teilabbildung besteht darin, daß die DLL bei einer Eingangsstörung von 4 MHz wie schon die PLL der Taktrückgewinnung bei 60 kHz einen Phasenversatz zur Phase des Datensignales aufweist, ihr also nicht mehr vollständig folgen kann. Im vierten Fall wird dieser Phasenversatz so groß, daß die DLL nicht mehr dem Datensignal folgt. Jitter höherer Frequenzen verringert deshalb die Weite des zur Datenrückgewinnung zur Verfügung stehenden Datenauges allmählich, was die notwendige Begrenzung der Jitter-Generierung des Sendesignales jedes Netzknotens begründet.

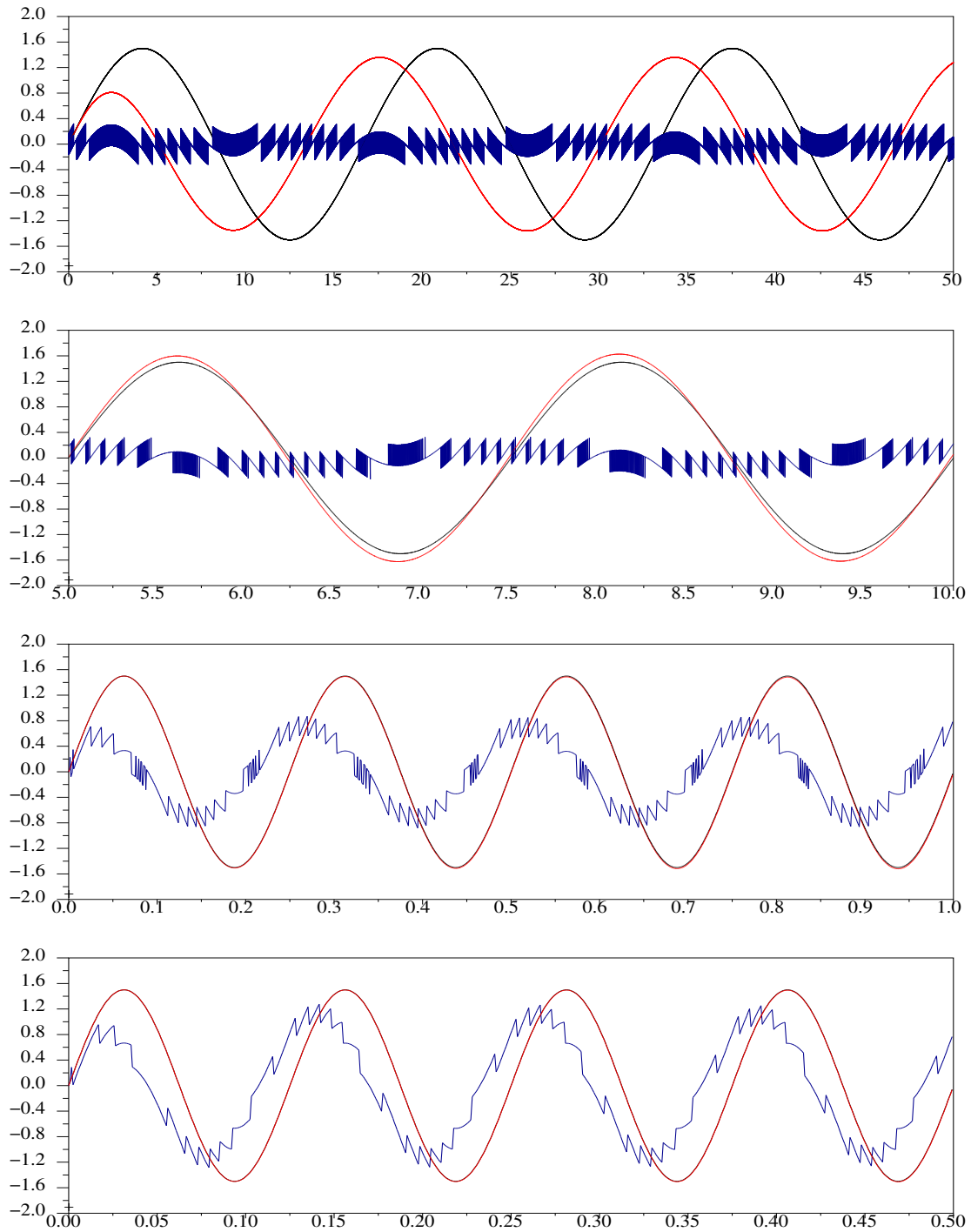
Die Teilabbildungen der Abb. 4.26 erläutern die soeben gemachten Feststellungen zu-

#### 4 Ein asynchroner Empfänger



**Abbildung 4.25:** Die Phasen von Takt- und Datenrückgewinnung bei Auftreten von Datenjitter

#### 4 Ein asynchroner Empfänger



**Abbildung 4.26:** Die Phasendifferenzen zwischen Datensignal und den Regelkreisen verursacht durch Datenjitter

sätzlich. Schon im ersten Fall ist eine große Phasendifferenz zwischen Oszillator und Datensignal feststellbar (rote Kurve). Obwohl die PLL versucht, dem Datensignal zu folgen, ergibt sich wegen der Phasenverschiebung zwischen Oszillator und Daten ein Phasenfehler, der kaum geringer ist als die am Eingang anliegende Störung. Bei höheren Jitterfrequenzen fallen die roten und schwarzen Kurven mehr oder weniger zusammen, was zeigt, daß die Taktrückgewinnung den an ihrem Eingang anliegenden Phasenfehler nicht mehr verringern kann, indem sie ihre Phase an die des Datensignales angleicht.

Die blauen Kurven beschreiben den Phasenfehler am Eingang der DLL. In den ersten beiden Teilabbildungen bleibt dieser nahe Null, wobei die kleinen Abweichungen auf die Rundungsfehler des Rotierers zurückzuführen sind. Im dritten Fall beginnt sich ein Phasenfehler am Eingang der DLL zu bilden. Die DLL kann ihn fortan nur noch teilweise ausgleichen. Der verbleibende Phasenfehler besitzt eine Amplitude von etwa 0,6 UI. Dies bedeutet, daß sich die Augenöffnung bereits um 60% schließt, was zu einer inakzeptabel hohen Bitfehlerrate führt. Allerdings ist anzumerken, daß die Amplitude der Störung mit 1,5 UI deutlich über den durch die Jitter-Tolerance für eine Jitterfrequenz von 4 MHz geforderten 0,15 UI liegt. Wird eine geringere Störung angelegt, verringert sich auch der Phasenfehler entsprechend, so daß die Jitter-Tolerance-Bedingung auch hier noch vollauf erfüllt werden kann.

Im vierten Fall zeigt sich auch für die DLL ein Phasenfehler, der in etwa dem Eingangsjitter gleicht. Jitter mit einer Frequenz von 8 MHz oder darüber kann nicht mehr ausgeglichen werden und verringert unmittelbar die Augenöffnung und erhöht somit die Bitfehlerrate.

Zusammenfassend ist festzustellen, daß der neuartige Regelkreis die nur scheinbar widersprüchlichen Anforderungen an Jitter-Transfer und Jitter-Tolerance erfolgreich vereint. Die Stabilität beider Teilregelkreise ist gewährleistet, weil — wie sich auch aus den Simulationsergebnisse ablesen läßt — die Taktrückgewinnung durch die volldigitale DLL vollkommen unbeeinflusst bleibt.

### 4.4.8 Die asynchrone Betriebsweise

Ein Transceiver muß sowohl im Master- als auch im Slave-Modus betrieben werden können. Die beiden Modi werden mittels zweier Regelschleifen auf dem Chip implementiert. Dabei wird der Oszillator entweder in die PLL der Taktrückgewinnung eingebunden (Slave-Modus) oder alternativ in eine sog. *Clock-Multiplier-Unit* (CMU; Master-Modus). Der Betriebsmodus wird über einen Multiplexer gewählt, der den Ausgang des Schleifenfilters entweder des einen oder des anderen Regelkreises auf die Kontrolleingänge des Oszillators schaltet.

Im Master-Modus folgt der Oszillator einer Referenz, die von außen in das System eingespeist wird. Der Referenztakt wird in der CMU mit dem notwendigen Faktor multipliziert,

so daß ein Sendetakt gewünschter Frequenz gewonnen wird. Eine CMU ist nichts anderes als eine PLL mit einem Taktteiler in der Rückkoppelschleife. Der Taktteiler bestimmt den Faktor, um den der Oszillatortakt höher ist als die am Eingang der PLL anliegende Referenz. Siehe hierzu z.B. [3].

Im Master-Modus können offensichtlich Unterschiede zwischen den Raten des empfangenen und zu sendenden Datensignales auftreten, die sich, je nach Topologie des Netzwerkes, nicht unbedingt mit der Zeit ausgleichen; er kann als Spezialfall des asynchronen Betriebes angesehen werden. Der neuartige Empfänger läßt sich wegen seiner doppelten Regelschleife bei Einsatz nur eines Oszillators auf dem Chip auch bei abweichenden Sende- und Empfangsdatenraten betreiben, so daß der Master-Betrieb automatisch gewährleistet ist.

Beim asynchronen Betrieb tritt verkomplizierend hinzu, daß wegen der abweichenden Datenraten der beiden Datenströme mit der Zeit überzählige oder fehlende Bits auflaufen. Wenn die Frequenz des lokalen Oszillators an eine Referenz gekoppelt ist, müssen deshalb in den empfangenen und gedemultiplexten Datenstrom in regelmäßigen Abständen entweder Bits eingefügt werden, wenn die empfangene Datenrate höher ist, oder aber Bits aus diesem herausgenommen werden, wenn sie niedriger ist als die der zu sendenden Daten. Jedes Mal, wenn die DLL den Rotierer eine Position weiterschaltet, wurde ein Phasenunterschied von einem Drittelbit ausgeglichen. Folglich summiert sich der Phasenunterschied beim dritten Schritt in dieselbe Richtung zu einer ganzen Bitlänge. An diesen Punkten wird deshalb je nach Vorzeichen des Frequenzunterschiedes ein zusätzliches Bit in den Datenstrom eingefügt oder eines herausgenommen.

Liegt die Rate der empfangenen Daten über der Senderate, dann muß das letzte Bit des aus den gespeicherten Oszillatorzuständen gewonnenen Datenwortes zwischengespeichert und dem folgenden Datenwort vorangestellt werden. Um die für den Abstimmungsalgorithmus notwendige Zahl von 24 Bits zu vervollständigen, wird ein Bit des Datenwortes verdoppelt. Es spielt dabei keine Rolle, welches. In den nächsten Takten bleiben weitere Bits des aus den gespeicherten Oszillatorzuständen gewonnenen Datenwortes übrig und werden jeweils für den folgenden Takt zwischengespeichert. Ist eine Phasendifferenz von einem ganzen Datenbit aufgelaufen, dann werden die drei aus dem Rotierer gefallenen Bits einfach in die Datenrückgewinnung gegeben, wo sie durch den Abstimmungsalgorithmus in ein zusätzliches Ausgangsdatenbit umgewandelt werden.

Ist die Datenrate des Empfangssignales niedriger als die Senderate, muß aus den Datenwörtern ein Bit herausgelöscht werden. Um wiederum den Satz der in der Abstimmung ausgewerteten Bits auf 24 zu bringen, wird nun ein Bit des nächsten Datenwortes dem aktuellen Datenwort angefügt. Dies setzt sich solange fort, bis sich die Phasendifferenz auf ein ganzes fehlendes Bit summiert hat. An dieser Stelle wird ein Satz mit nur 21 Bits zur Abstimmung gestellt und folglich ein Bit weniger ausgegeben.

Eine Schätzung der maximalen Differenz zwischen den beiden Datenraten erscheint an-

gebracht. Die Genauigkeit einer für eine SDH/SONET-Datenstrecke zulässigen Taktquelle muß 20 ppm betragen. An Knoten, die die Verbindung zwischen zwei Subnetzen mit jeweils einer eigenen Taktreferenz bilden, kann sich die Taktungenauigkeit im schlimmsten Fall zu einer Abweichung von 40 ppm oder eben  $4 \cdot 10^5$  Hz summieren. Zusätzlich zu berücksichtigen ist der Einfluß von Jitter auf die Änderung der momentanen Datenrate des empfangenen Signales. Aus der Jitter-Tolerance-Maske in der Abbildung 2.1 auf Seite 18 kann die größte Änderungsgeschwindigkeit der Phase und damit die am stärksten abweichende Momentanfrequenz bestimmt werden, die das System verkraften können muß, indem das Maximum der Ableitung der Jitterfunktion errechnet wird. Der Jitter ist laut Definition sinusförmig, wird also durch diese Formel beschrieben:

$$J(t) = A_{UI} \cdot \sin \omega t \quad (4.9)$$

Die Ableitung ergibt sich zu:

$$J'(t) = A_{UI} \cdot \omega \cdot \cos \omega t \quad (4.10)$$

Das Maximum der Ableitung beträgt  $A_{UI} \cdot \omega$ . Aus der Jitter-Tolerance-Maske ergeben sich verschiedene Kombinationen von  $A_{UI}$  und  $\omega$ , deren höchstes Produkt die Wertepaare  $1,5 UI$  bei 400 kHz bzw.  $0,15 UI$  bei 4 MHz liefern. Die maximale Änderungsgeschwindigkeit  $\hat{J}'$  der Zykluszeit beträgt  $1,2 \cdot \pi \cdot 10^6$  Unit-Intervalle pro Sekunde und ist damit dimensionslos. Die maximale Abweichung der momentanen Datenrate berechnet sich über die Veränderung der Zykluszeit innerhalb eines Datenbits:

$$\Delta D_{max} = T \cdot \hat{J}' \quad (4.11)$$

Die Abweichung der Datenrate durch Jitter erreicht maximal  $1,2 \cdot \pi \cdot 10^6$  Bits/s. Im schlechtesten Fall summieren sich alle Abweichungen zu ungefähr  $4,2 \cdot 10^6$  Bits/s, was bei der Datenrate von 10 GBit/s 420 ppm entspricht. In der Datenrückgewinnung kann folglich frühestens alle  $\frac{24}{420 \text{ ppm}} \approx 57.000$  Achtbitdatensätze ein zusätzliches oder fehlendes Bit der gespeicherten Oszillatorzustände auftreten, das vor Anwendung der Abstimmung zwischengespeichert oder ausgeglichen werden muß.

Die rechnerische Maximalabweichung zwischen den Datenraten von empfangenem und gesendetem Signal ist wegen der hohen Genauigkeit der verwendeten Taktreferenzen offenbar sehr gering und läßt sich mit dem oben beschriebenen Mechanismus in jedem Fall ausgleichen, da niemals mehr als ein zusätzliches Bit in einem Achtbitdatensatz anfällt.

Die Verarbeitung der empfangenen und parallelisierten Daten würde erschwert, wenn zusätzliche oder fehlende Bits im Datenstrom berücksichtigt werden müßten. Deshalb muß eine quasi-synchrone Betriebsweise vorgesehen werden, die eine Änderung der Taktung

der folgenden Verarbeitungseinheiten erlaubt, um die Verarbeitung jeweils gleichlanger Datensätze zu ermöglichen.

Die Datenrückgewinnung wird zuerst bei einer Frequenz von 1,25 GHz vorgenommen, was für die folgenden komplexeren digitalen Funktionseinheiten zu schnell ist, weshalb die Taktgeschwindigkeit auf ein Viertel geteilt wird. In der Datenrückgewinnung werden jeweils vier aufeinanderfolgende Datenwörter zu einem 32-Bit-Datenwort zusammengefaßt und mit dem heruntergeteilten Takt ausgegeben. Erlaubt der Taktteiler ein variables Teilerverhältnis, dann kann der heruntergeteilte Takt für einen Teilerzyklus verlängert oder verkürzt werden. Diese Maßnahme kompensiert die wegen des asynchronen Betriebes von Sender und Empfänger anfallenden überzähligen oder fehlenden Bits; neben dem Taktteiler ist dazu lediglich ein FIFO-Puffer notwendig.



## 5 Fazit

Die neuartige Sender/Empfängeranordnung besitzt viele Vorteile gegenüber dem Stand der Wissenschaft. Die bei Einsatz der CMOS-Technologie notwendige Kompensierung der niedrigeren Geschwindigkeit der Schaltungselemente wird durch Parallelisierung und durch Verschieben der Aufgaben hin zu niedrigeren Taktfrequenzen erreicht. Die einzigen Elemente, die bei der der Hälfte der Datenrate entsprechenden Taktfrequenz von 5 GHz arbeiten, sind:

- zwei identisch aufgebaute Flip-Flops in der ersten Demultiplexerzelle
- der diese Flip-Flops treibende Eingangsverstärker
- das letzte Exklusiv-Oder-Gatter, das als Multiplexer und Ausgangstreiber auf der Sendeseite fungiert

Wegen der minimalen Zahl schneller Elemente ist eine gesenkte Leistungsaufnahme des gesamten Chips zu erwarten.

Die äußerst geringe kapazitive Belastung des Eingangsverstärkers durch die Empfängerschaltung erleichtert dessen schwierigen Entwurf, der eine extrem niedrige „Noise-Figure“ mit hoher Verstärkung vereinen muß. Die Bandweite der beiden Flip-Flops der ersten Demultiplexerstufe sowie des Exklusiv-Oder-Gatters können durch „Shunt-peaking“ [19] erhöht werden. Dabei werden die Lasten der in CML-Schaltungstechnik aufgebauten Elemente um einen LC-Resonator erweitert und dessen Resonanzfrequenz derart gewählt, daß die Bandweite der Schaltung verbessert wird, ohne eine Resonanzspitze aufzuweisen.

Das kritischste Element der Empfängerarchitektur ist der Oszillator, den hohe Phasereinheit auszeichnen muß. Grundsätzlich gilt jedoch, daß die Anforderungen an den Oszillator durch die neue Empfängerarchitektur nicht höher ausfallen, als durch die maßgeblichen Standards ohnehin gefordert. Indessen vereinfacht sie wegen der Notwendigkeit nur eines Oszillators auf dem Chip die Einhaltung der Anforderungen erheblich und senkt gleichzeitig wegen der herabgesetzten Taktfrequenz die Leistungsaufnahme des Oszillators. Der Empfänger erreicht letzteres, indem er auch hier parallelisiert: anstelle eines 5 GHz-Oszillators mit wenigstens sechs Phasensignalen beruht er auf einem 1,25 GHz-Oszillator mit 24 Phasensignalen. Das bedingt jedoch als größten Nachteil der neuen Empfängerarchitektur einen erhöhten Flächenbedarf wegen der vielen notwendigen LC-Resonatoren.

Die vorgestellte Empfängerarchitektur ermöglicht die Ausführung aller notwendigen Arbeitsschritte zur Frequenz-/Phasendetektion und der Takt-/Datenrückgewinnung bei einer in CMOS leicht beherrschbaren Frequenz. Ein bedeutender Teil kann hierbei in synthetisierbaren VHDL-Beschreibungen entwickelt werden. Auf diese Weise ist die Architektur einfach auf zukünftige Fertigungsprozesse übertragbar, wobei nur eine Überarbeitung der Hochgeschwindigkeitselemente nötig wäre.

Die neuartige Empfänger-/Senderanordnung stellt einen starken Kandidaten für eine erste erfolgreiche vollintegrierte Realisierung eines 40 GBit/s-Empfängers gemäß OC-768 in BiCMOS-Fertigungstechnologie. Die exponentielle Abhängigkeit des Kollektorstromes eines Bipolartransistors von seiner Basis-Emitter-Spannung macht ihn besonders attraktiv für die erste Demultiplexerstufe, bei der niedrige Eingangsspannungshübe zu erwarten sind. Ihre hohe Stromergiebigkeit läßt Bipolartransistoren auch wegen der mit CMOS-Transistoren nur schwierig zu erreichenden notwendigen Ausgangstreiberleistung bei noch höheren Datenraten als 10 GBit/s in näherer Zukunft unverzichtbar erscheinen.

Wegen des nur punktuell notwendigen Einsatzes schneller Bauelemente ist die neuartige Architektur für einen BiCMOS-Prozeß besonders geeignet. Die wenigen bipolaren Bauelemente und die Art ihres Einsatzes machen sogar eine Senkung der Leistungsaufnahme des Chips wahrscheinlich. So müßte der Bias-Strom der andernfalls in CMOS ausgeführten Blöcke stark erhöht werden, um dieselbe Schaltgeschwindigkeit zu erreichen. Dies würde zu größeren Transistorweiten und damit zu größeren Gate-Kapazitäten und wiederum erhöhten Anforderungen an die Treiberleistung der vorhergehenden Stufen führen.

Bei einer Datenrate von 10 GBit/s ist die Umsetzung dagegen vollständig in CMOS-Technologie möglich, wie die Erfahrungen aus dem Projekt der INFINEON TECHNOLOGIES AG und die ersten wissenschaftlichen Publikationen zum Thema zeigen [20] [21].

Für die Empfängerseite des in dieser Arbeit vorgestellten Ansatzes kann festgestellt werden, daß lediglich zwei Flip-Flops bei der Frequenz der halben Datenrate funktionsfähig sein müssen. Diese Flip-Flops sind als Toggle-Flip-Flops aufgebaut, so daß sich ihr jeweils nächster Zustand aus dem aktuellen ergibt. Die schaltungstechnische Umsetzung des Flip-Flops läßt sich möglicherweise gegenüber der eines gewöhnlichen Abtast-Flip-Flops vereinfachen, besteht doch bei jenem keinerlei statistischer Zusammenhang zwischen zwei aufeinander folgenden Inhalten. Es kann deshalb ohne Beweis behauptet werden, daß die höchstmögliche Datenrate dieser Empfängeranordnung *mindestens* so hoch ist wie die der Half-rate-Empfängeranordnungen, wahrscheinlich aber sogar höher.

Ein weiterer Vorteil des neuen Sender/Empfängers liegt in der fehlenden Notwendigkeit eines 5 GHz-Taktbaumes. Da man derart hohe Taktfrequenzen nur schwer über größere Distanzen übermitteln kann, wären die Taktsignale etwa alle 100  $\mu\text{m}$  Verdrahtungsstrecke zu verstärken. Dies bedingt einen hohen Leistungsbedarf und verschlechtert die Taktreinheit wegen der Einkopplung von Rauschen über die Versorgungsanschlüsse der Takttreiber. In der neuen Architektur können die wenigen kritischen Wege im Layout kurz und

symmetrisch gehalten werden, so daß die Daten weniger Störeinflüssen ausgesetzt sind. Wegen der volldigitalen Realisierung der Taktrückgewinnung ist die hohe Genauigkeit ihrer Regelcharakteristik ohne Einsatz von zu kalibrierenden und das Gesamtsystem vertuernden externen Komponenten erreichbar. Neben all diesen Vorteilen bietet die neue Architektur einfache Mittel zur Senkung der Bitfehlerrate der empfangenen Daten.

## Literaturverzeichnis

- [1] TEKTRONIX, „SONET Telecommunications Standard Primer“, <http://www.tektronix.com/optical/>, S. 4ff.
- [2] BELL COMMUNICATIONS RESEARCH, INC., Sonet OC-192 Transport System Generic Criteria, Generic Requirements GR-1377-CORE, Issue 5, December 1998, Abschnitt 5.6
- [3] R. BEST, „Theorie und Anwendungen des Phase-locked Loops“, 5. Auflage, AT-Verlag, S. 20ff.
- [4] M. RAU et al., „Clock/Data Recovery PLL Using Half-Frequency Clock“, *IEEE Journal of Solid-State Circuits*, vol. 32, Juli 1997, S. 1156ff.
- [5] W. BÖGE (Hrsg.), „Handbuch Elektrotechnik“, Vieweg, 2. Auflage, Mai 2002, S. 570
- [6] J. SAVOJ, B. RAZAVI, „High-Speed CMOS Circuits for Optical Receivers“, Kluwer Academic Publishers, 2002, S. 9, Abbildung 1.7 unten
- [7] N. F. GONCALVES, H. DE MAN, „NORA: A Racefree Dynamic CMOS Technique for Pipelined Logic Structures“, *IEEE Journal of Solid-State Circuits*, vol. 18, Juni 1983
- [8] C. A. MEAD, L. A. CONWAY, „Introduction to VLSI Systems“, Addison-Wesley, 1980
- [9] E. HERTER, W. LÖRCHER, „Nachrichtentechnik“, Hanser Verlag, 9. Auflage, 2004, S. 570
- [10] R. WALKER, B. AMRUTUR, T. KNOTTS, „64B/66B Coding Update“, *IEEE 802.3ae Meeting*, Albuquerque, März 2000
- [11] B. KIM, D. N. HELMAN, P. R. GRAY, „A 30-MHz Hybrid Analog/Digital Clock Recovery Circuit in 2- $\mu$ m CMOS“, *IEEE Journal of Solid-State Circuits*, vol. 25, Dezember 1990, S. 1385ff.
- [12] C. T. GRAY, W. LIU et al., „A Sampling Technique and Its CMOS Implementation with 1 Gb/s Bandwidth and 25 ps Resolution“, *IEEE Journal of Solid-State Circuits*, vol. 29, März 1994, S. 340ff.

- [13] J. D. H. ALEXANDER, „Clock Recovery from Random Binary Signals”, *Electronics Letters*, vol. 11, Oktober 1975, S. 541ff.
- [14] A. POTTBACKER, U. LANGMANN, H. U. SCHREIBER, „A Si Bipolar Phase and Frequency Detector IC for Clock Extraction up to 8 GBit/s”, *IEEE Journal of Solid-State Circuits*, vol. 27, Dezember 1992, S. 1747ff.
- [15] A. ROFOUGARAN, J. RAEL et al., „A 900MHz CMOS LC-Oscillator with Quadrature Outputs”, *International Solid-State Circuits Conference 1996*, Analog Techniques, Paper SP 24.6
- [16] M. BOSSERT, „Kanalcodierung”, B. G. Teubner Stuttgart, 2. Auflage 1998, S. 361ff.
- [17] P. G. FARRELL, „Digital Transmission – Lecture Hand-outs”, University of Manchester, 1997
- [18] INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE, scilab, Scilab Home Page, <http://scilabsoft.inria.fr/>
- [19] S. S. MOHAN, M. DEL MAR HERSHENSON et al., „Bandwidth Extension in CMOS with Optimized On-Chip Inductors”, *IEEE Journal of Solid-State Circuits*, vol. 35, März 2000, S. 346ff.
- [20] J. SAVOJ, B. RAZAVI, „A 10-Gb/s CMOS Clock and Data Recovery Circuit with a Half-Rate Linear Phase Detector”, *IEEE Journal of Solid-State Circuits*, vol. 36, Mai 2001, S. 761ff.
- [21] J. SAVOJ, B. RAZAVI, „A 10-Gb/s CMOS Clock and Data Recovery Circuit With a Half-Rate Binary Phase/Frequency Detector”, *IEEE Journal of Solid-State Circuits*, vol. 38, Januar 2003, S. 13ff.

## Abbildungsverzeichnis

1.1	Übersicht über eine typische optische Datenübertragungsstrecke . . . . .	8
1.2	Ein als Ring ausgeführtes Datennetz . . . . .	9
1.3	Übersicht über die Funktionsblöcke eines typischen Transceiverchips . . . .	12
2.1	Maske für den Jitter-Tolerance Test . . . . .	18
3.1	Blockdiagramm eines als Schieberegister ausgeführten Multiplexers . . . . .	25
3.2	Blockdiagramm eines als Binärbaum ausgeführten Multiplexers . . . . .	26
3.3	Timing im Multiplexer bei Einsatz eines asynchronen Teilers . . . . .	28
3.4	Eine typische 2:1-Multiplexeranordnung . . . . .	29
3.5	Eine Multiplexerschaltung in CMOS-CML-Schaltungstechnik . . . . .	30
3.6	Blockdiagramm des Multiplexers nach [6] . . . . .	31
3.7	Schaltung der Multiplexer-Grundzelle . . . . .	32
3.8	Logische Verknüpfung der beiden Datensignale in der Multiplexerzelle . .	33
3.9	Die Multiplexerzellen zu einem binären Baum angeordnet . . . . .	34
3.10	Umkodierung eines pegelkodierten Signales in ein flankenkodiertes . . . . .	35
3.11	Prinzipschaltbild des Vorkodierers . . . . .	36
3.12	Umgeordneter Multiplexer-Baum nach Abb. 3.9 . . . . .	39
3.13	Darstellung von acht Signalen gemäß Abb. 3.8 . . . . .	40
3.14	Typische schaltungstechnische Realisierung eines Toggle-Flip-Flops . . . . .	41
3.15	Alternative Ausführung des Kodierers . . . . .	43
3.16	Blockdiagramm des Multiplexers . . . . .	44
3.17	Überblick über die logischen Blöcke des Multiplexers . . . . .	45
3.18	Augendiagramm für die verschiedenen Parametersätze . . . . .	48
3.19	Die Schaltung der Schnittstelle . . . . .	49
3.20	Die Schaltung des 32:16-Multiplexers . . . . .	51
3.21	Die Schaltung des 16:2-Multiplexers . . . . .	53
3.22	Der 8:1-Multiplexer . . . . .	56
3.23	Eine Hälfte des dynamischen Buffers in PMOS-Ausführung . . . . .	57
3.24	Der asynchrone Zähler . . . . .	59
3.25	Das dynamische Flip-Flop . . . . .	61
3.26	Die Schaltung des Biasblocks . . . . .	64
3.27	Das gemessene Datensignal . . . . .	66

3.28	Das Datenaugenauge der Sendedaten . . . . .	68
4.1	Blockdiagramm eines typischen als PLL aufgebauten Empfängers . . . . .	72
4.2	Parallelisierte Abtastung des Datensignals . . . . .	76
4.3	Verzögerungskette als Demultiplexer . . . . .	77
4.4	Takt-/Datenrückgewinnung nach dem „Bang-bang“-Prinzip . . . . .	79
4.5	Beispielsignale für die Fälle a) „early“ und b) „late“ . . . . .	80
4.6	Erweiterung der „Bang-bang“-Schaltung auf einen Frequenzdetektor . . . . .	81
4.7	Beispielsignale für die Fälle a) „early“ b) „in Phase“ und c) „late“ . . . . .	82
4.8	Beispielsignale zur Erläuterung der Frequenzdetektion . . . . .	83
4.9	Blockdiagramm des modifizierten double-edge-triggered Flip-Flops . . . . .	85
4.10	Schaltung der 1:2-Demultiplexer-Grundzelle . . . . .	87
4.11	Beispielsignale zur Funktion der Demultiplexer-Grundzelle . . . . .	88
4.12	Rückgewinnung des Eingangssignales . . . . .	88
4.13	Blockdiagramm des seriellen Empfängers . . . . .	90
4.14	Schematische Darstellung der möglichen Oszillatorzustände . . . . .	92
4.15	Beispiel zur Datenrückgewinnung . . . . .	93
4.16	Zur Erläuterung des Abstimmungsalgorithmus . . . . .	95
4.17	Schaltung zur Detektion neuer im asynchronen Registersatz gespeicherter Flankenzeitpunkte . . . . .	97
4.18	Detektion des Phasenverhältnisses zwischen Oszillator und Daten . . . . .	100
4.19	Zur Frequenzdetektion . . . . .	103
4.20	Blockdiagramm der Doppel-PLL . . . . .	106
4.21	Die Doppel-PLL als Regelkreis . . . . .	107
4.22	Die Doppel-PLL in der Darstellung des Systemsimulators <b>scicos</b> . . . . .	108
4.23	Schaltungsstruktur des digitalen Schleifenfilters der PLL . . . . .	111
4.24	Antwort der Doppel-PLL auf einen Phasensprung . . . . .	113
4.25	Die Phasen von Takt- und Datenrückgewinnung bei Auftreten von Da- tenjitter . . . . .	115
4.26	Die Phasendifferenzen zwischen Datensignal und den Regelkreisen verur- sacht durch Datenjitter . . . . .	116

## Tabellenverzeichnis

3.1	Umbenennung der Ein- und Ausgänge des Multiplexers . . . . .	32
3.2	Der Leistungsbedarf des Multiplexers für verschiedene Betriebsparameter .	46
4.1	Tabelle mit den möglichen Ausgangswerten der Phasendetektion . . . . .	102