UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

FAKULTÄT FÜR
ELEKTROTECHNIK,
INFORMATIK UND
MATHEMATIK

Analyse der Ressourceneffizienz leitungsgebundener Kommunikation in Multiprozessorsystemen

Von der Fakultät für Elektrotechnik, Informatik und Mathematik
der Universität Paderborn

zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften (Dr.-Ing.)

genehmigte Dissertation

von

Dipl.-Ing. Manuel Strugholtz

Erster Gutachter: Prof. Dr.-Ing. Ulrich Rückert
Zweiter Gutachter: Prof. Dr.-Ing. Christoph Scheytt

Tag der mündlichen Prüfung: 14.05.2013

Paderborn 2013

Diss. EIM-E/230

Abstract

Deutsch
In verteilten Rechenarchitekturen und Multiprozessorsystemen hat die Kommunikationsinfrastruktur einen großen Einfluss auf die Leistungsfähigkeit des Gesamtsystems. Während die eigentlichen Recheneinheiten zunehmend auf hohe Energieeffizienz hin optimiert werden, liegt der Hauptfokus in der Entwicklung der Kommunikationsverfahren auf hoher Leistungsfähigkeit. In dieser Arbeit wird deshalb die Ressourceneffizienz leitungsgebundener Kommunikation in Multiprozessorsystemen untersucht. Es werden Grundlagen der kupferbasierten Übertragung von Daten erläutert und Transceiver vorgestellt, mit denen die entsprechenden Untersuchungen durchgeführt werden. Ein Kanalmodell zur Bestimmung des Einflusses auf das Übertragungssignal wird vorgestellt und verifiziert. Der grundsätzliche Aufbau von seriellen Hochgeschwindigkeitstransceivern wird erklärt und wichtige Komponenten werden näher erläutert. Eine Evaluierungsmethodik wird vorgestellt, mit der die Ergebnisse der Energieuntersuchungen quantifiziert und eingeordnet werden können. Die Übertragungsverfahren werden auf Basis ihrer technologischen Implementierung eingeordnet und auf ihre Effizienz und durchschnittliche Leitungsaufnahme hin untersucht. Eine standardübergreifende Evaluation findet statt. Hier werden die betrachteten Übertragungsverfahren untereinander verglichen und eine Unterteilung in technologische Unterschiede wie Busse oder Punkt-zu-Punkt-Verbindungen und Intra- beziehungsweise Intersystemverfahren vorgenommen. Anschließend werden die gewonnenen Erkenntnisse auf verschiedene Clustersysteme angewendet und der Anteil der Inter- und Intrasystemkommunikation an der Gesamtverlustleistung bestimmt. Es werden zwei Clustersysteme beschrieben, die mit Hilfe der gewonnenen Erkenntnisse aus den vorhergehenden Kapiteln entwickelt wurden. Das erste System stellt ein FPGA-Cluster dar, welcher eng gekoppelte rekonfigurierbare Architekturen einsetzt. RECS, ein ressourceneffizienter Cluster-Server, ist auf niedrigen Energiebedarf sowie auf eine hohe Packungsdichte an physikalischen Rechenknoten optimiert. RECS beinhaltet ein effizientes System zur Überwachung und Steuerung von Multiprozessorarchitekturen. Ein effizientes Langzeitarchivsystem auf Basis von Festplatten erweitert das RECS-System.

English
In dedicated computing architectures and multi processor systems the influence of communication on the performance of the whole system is growing. While computing units like processors are increasingly optimized on high energy efficiency, the main focus in developing communication techniques still lies on high performance. In this work the resource efficiency of copper based communication in multi processor systems is evaluated. Basics of copper based transmission of data are explained and transceivers used in the evaluation as well as a model of the copper channel are introduced. The fundamental structure of serial high speed transceivers is explained and important parts are clarified. A methodology for evaluation is introduced that is used for quantifying the results of the analysis like the average power consumption of the communication method and the needed amount of energy per bit under different views. An overall evaluation is performed to compare all communication techniques and sort them based on their characteristics like bus based connections or point to point connections as well as inter system or intra system communication. Two cluster systems are introduced which have been developed based on the results of the previous theoretically evaluation. The first system is a FPGA cluster that uses tightly coupled reconfigurable architectures. RECS, a resource efficient cluster server, is optimized for low power consumption and high density of physical processing nodes. RECS also uses an efficient management system for monitoring and controlling multi processor architectures. An efficient storage system based on hard drives expands the functionality of RECS.

Inhaltsverzeichnis

1 Einleitung

Die Erfindung mikroelektronischer Schaltkreise im 20. Jahrhundert hat zu einem radikalen Umbruch in bestimmten Technologiezweigen wie der Kommunikations- und Informationstechnologie geführt. Der weltweit erste, auf Basis mikroelektronischer Schaltkreise arbeitende Rechner, war der „Electronic Discrete Variable Automatic Computer“ (EDVAC [R50]) und stammt aus den späten 1940er Jahren. Er besaß eine Speicherkapazität von 5,5 kByte und konnte circa 345 Multiplikationen pro Sekunde berechnen. Im Jahr 2011 schafft der schnellste Rechner der Welt über 10 Billiarden Fließkommaoperationen ($10\,\mathrm{PFlop/s} = 10 \cdot 10^{15}\,\mathrm{Flop/s}$) pro Sekunde [R45]. Diese Leistungssteigerung wurde nur durch eine massive Weiterentwicklung integrierter Schaltkreise möglich, welche durch das sogenannte Mooresche Gesetz beschrieben wird. Das Mooresche Gesetz [R48] beschreibt die Verdopplung der Integrationsdichte von Schaltkreisen alle 18 Monate. Die Entwicklung integrierter Schaltkreise verläuft seit dem Beginn der digitalen Revolution bis heute im Einklang mit diesem Gesetz. Lange Zeit wurde die Steigerung der Leistungsfähigkeit von integrierten Schaltkreisen ebenfalls durch das Mooresche Gesetz abgedeckt. Seit circa 2002 divergiert jedoch die zeitliche Zunahme der Leistungsfähigkeit von der Integrationsdichte (vgl. Abbildung 1.1). Die Leistungsfähigkeit flacht in ihrer zeitlichen Zunahme gegenüber der Integrationsdichte ab, weshalb die Abweichung zwischen beiden Graphen als die Mooresche Lücke bezeichnet wird.

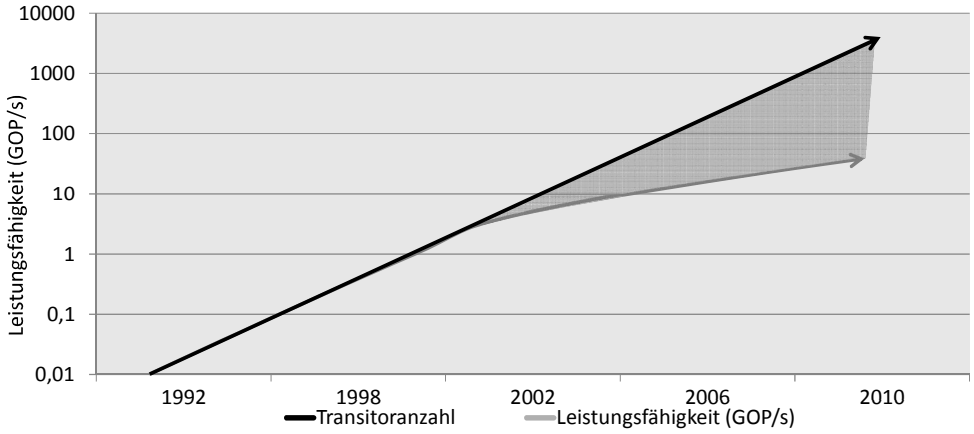

Abbildung 1.1: Die zeitliche Entwicklung der Integrationsdichte und der Leistungsfähigkeit von integrierten Schaltkreisen [R7].

Einen großen Anteil an dieser Entwicklung trägt das zeitliche Anwachsen der Verlustleistungsdichte, bei immer kleineren Strukturgrößen und wachsenden Taktfrequenzen. Diese wurde bei Prozessoren wie dem Intel Pentium 4 und dem AMD Athlon XP nicht mehr beherrschbar [R55], wodurch sich eine weitere Steigerung der Prozessortaktraten als impraktikabel erwies [R24]. Um dieses Problem zu umgehen, wurden zunehmend parallele Architekturen eingeführt, welche eine weitere Steigerung der Leistungsfähigkeit bei niedrigerer oder konstant bleibender Taktrate ermöglichen.
Der steigende Energiebedarf integrierter Schaltungen hat einen negativen Einfluss auf die Entwicklung der Leistungsfähigkeit. Zunehmend drängen sich wirtschaftliche Aspekte des wachsenden Energiebedarfs in Form von steigenden Energiepreisen in den Vordergrund. Dieses Problem tritt besonders deutlich bei Rechenzentren hervor, wenn deren Wirtschaftlichkeit untersucht wird (vgl. Abbildung 1.2). Die Anschaffungskosten für einen Server können über die Jahre als konstant angesehen werden. Die Unterhaltskosten für einen Server nehmen jedoch stetig zu. Während im Jahre 2001 die Summe aus jährlichen Infrastruktur- und Energiekosten den Anschaffungspreis eines Servers überstieg, wurde der Anschaffungspreis 2004 allein von den Infrastrukturkosten übertroffen. Als Infrastruktur zählt hierbei die zum Betrieb eines Servers notwendige Peripherie wie beispielsweise eine Klimaanlage. 2008 kostete die Energieversorgung eines Servers pro Jahr so viel wie seine Anschaffung.

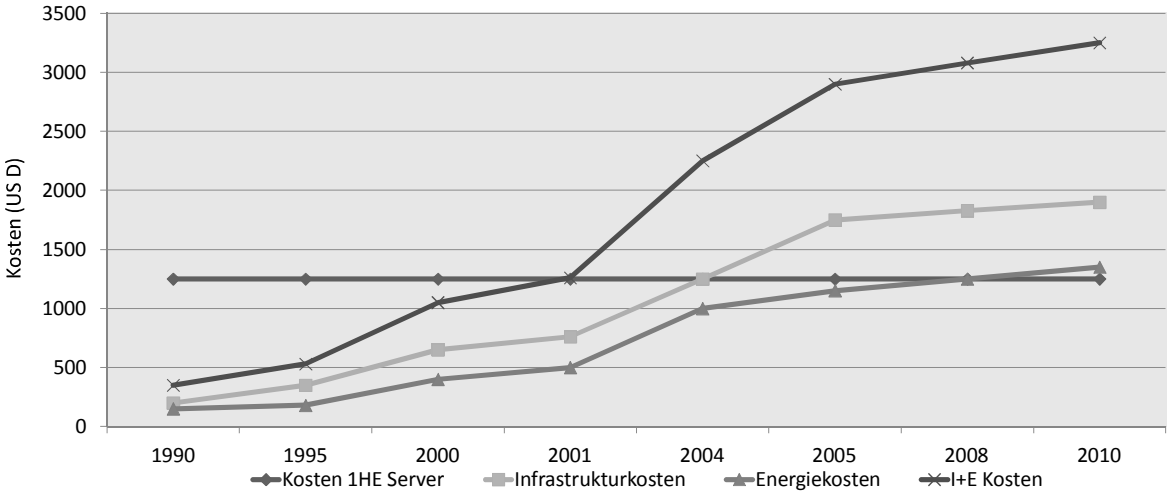

Abbildung 1.2: Die Energie- und damit die Unterhaltungskosten von Rechenzentren steigen zunehmend [R17].

Die steigenden Kosten zeigen die zunehmende Problematik bei dem Betrieb von Rechenzentren aus wirtschaftlicher Sicht, wobei sich gleichzeitig auch eine ökologische Problematik abzeichnet. Eine zunehmende Leistungsaufnahme bei Rechenarchitekturen bedeutet gleichzeitig auch steigende CO_2-Emissionen durch die Erzeugung dieser Energie. Die jährliche Summe der weltweiten CO_2-Emission von Rechenzentren ent-

spricht mittlerweile dem Ausstoß des gesamten Flugverkehrs, welcher jährlich 2 % der Gesamtemission weltweit ausmacht [R42].
Dies hat zu einem Umdenken bei der Entwicklung von Rechensystemen geführt. Neben der Erhöhung der Leistungsfähigkeit steht zunehmend die Verbesserung der Energieeffizienz im Vordergrund. Unter dem Begriff der „Green-IT“, also „grüner“ und damit umweltfreundlicher Informationstechnologie, wurden alternative Blickwinkel bei der Bewertung der Leistungsfähigkeit von Rechenarchitekturen erschlossen. Als Beispiel hierfür ist die Etablierung der Green500-Liste zu nennen [R24], welche eine Rangliste der 500 energieeffizientesten Supercomputer weltweit führt. Wie in Abbildung 1.3 zu sehen ist, kehrt sich gegen Ende 2009 der Trend wachsender Leistungsaufnahme bei Supercomputern um. Zugleich steigt jedoch die Leitungsfähigkeit weiter an, was zu einer zunehmend ansteigenden Energieeffizienz führt. Diese Erhöhung der Energieeffizienz ist auf die zunehmende Optimierung fast aller beteiligten Komponenten von Rechenarchitekturen zurückzuführen. Moderne Prozessoren verfügen über diverse Technologien, um ihre Leistungsaufnahme je nach Bedarf zu senken, beispielsweise in Phasen niedriger Aktivität, in denen die Taktfrequenz und die Spannungsversorgung auf ein niedrigeres Niveau gesetzt werden. Netzteile besitzen einen zunehmend hohen Wirkungsgrad, sie erzeugen entsprechend weniger Verluste bei der Umwandlung der Netzspannung auf die systemintern benötigten Spannungswerte. Ein weiterer großer Trend besteht in der Verwendung von hardwarebasierten Beschleunigern wie Grafikkarten oder rekonfigurierbaren Architekturen. Diese Komponenten sind für bestimmte Anwendungen optimiert und können Berechnungen schneller und vor allem energieeffizienter ausführen als die für allgemeinere Aufgaben entwickelten Hauptprozessoren.

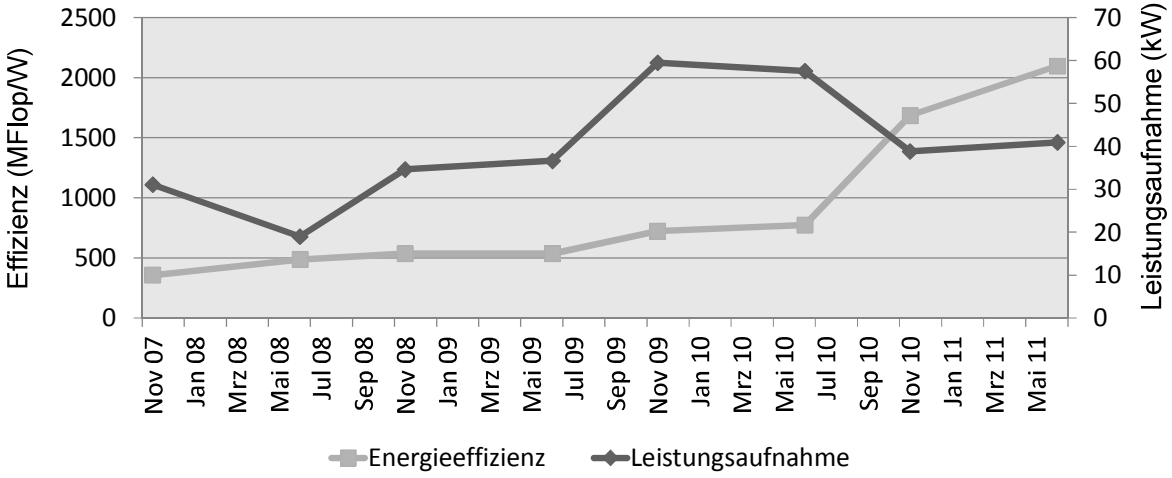

Abbildung 1.3: Die Energieeffizienz von Supercomputern nimmt stetig zu [R23].

Neben all diesen Trends zur Nutzung effizienterer Rechenarchitekturen ist eine Steigerung der Parallelität sowohl innerhalb der einzelnen Rechenknoten in Form mehrerer Rechenkerne als auch zwischen den Rechenknoten in Form einer steigenden Knotenanzahl zu beobachten. Hierbei wird sowohl eine Intrasystemparallelität durch die Nutzung mehrerer, physikalisch getrennter Prozessoren als auch eine Intersystemparallelität verwendet, bei der viele einzelne Rechensysteme zu sogenannten Clustern (Rechnerverbund) zusammengefügt werden. In Clustern müssen alle Systeme und auch alle Komponenten innerhalb der Systeme untereinander kommunizieren. Es gibt verschiedene Verfahren zur Realisierung dieser Kommunikation, wobei sich diese neben der Performanz auch in der Energieeffizienz unterscheiden. Quantitative oder qualitative Angaben über die Energieeffizienz von kupferbasierten Kommunikationsverfahren sind bislang nicht verfügbar. Das Ziel dieser Arbeit liegt in der Angabe der fehlenden Informationen und kann entsprechend zusammengefasst werden:

„Evaluierung von kupferbasierter Inter- und Intrasystemkommunikation in Bezug auf Energieeffizienz."

Hierzu gehört die kupferbasierte Datenübertragung zwischen Systemen und die Datenübertragung innerhalb eines Systems, beispielsweise zwischen Prozessor und Chipsatz oder Chipsatz und Netzwerkadapter. Alle diese Kommunikationsstrecken benötigen Energie für die Übertragung von Informationen. Der Anteil der benötigten Energie an der Gesamtleistungsaufnahme und der Unterschied zwischen den einzelnen Verfahren soll in dieser Arbeit betrachtet werden.
In **Kapitel 2** werden Grundlagen der kupferbasierten Übertragung von Daten erläutert und Transceiver vorgestellt, mit denen die entsprechenden Untersuchungen durchgeführt werden. Ein Kanalmodell zur Bestimmung des Einflusses auf das Übertragungssignal wird vorgestellt und verifiziert. Aufgrund des vorgestellten Kanalmodells und der sich ergebenden Effekte auf die Signalintegrität werden Techniken zur Signalformadaption beschrieben, welche später in der Evaluation Anwendung finden. Der grundsätzliche Aufbau von seriellen Hochgeschwindigkeitstransceivern wird erklärt und wichtige Komponenten werden näher erläutert.
Kapitel 3 stellt eine Evaluierungsmethodik vor, mit der die Ergebnisse der Energieuntersuchungen quantifiziert und eingeordnet werden können. Hierzu gehört eine Aussage über die durchschnittliche Leistungsaufnahme der betrachteten Verfahren sowie über die benötigte Energie pro übertragenem Bit unter verschiedenen Gesichtspunkten. Zu dieser Energie zählt die Betrachtung eines physikalischen Signals auf dem Kanal, eine Angabe über den Einfluss von Kanalkodierungen und Paketformaten auf die benötigte Energie für ein Nutzdatenbit, sowie eine Angabe der zu erwartenden Energieeffizienz des Standards in praktischen Anwendungsszenarien. Die Übertragungsverfahren werden in diesem Kapitel auf Basis ihrer technologischen Implementierung eingeordnet

und auf ihre Effizienz und durchschnittliche Leitungsaufnahme hin untersucht. Alle Verfahren werden auf jedem kompatiblen Transceiver in verschiedenen Varianten implementiert.
In **Kapitel 4** findet eine standardübergreifende Evaluation statt. Hier werden die betrachteten Übertragungsverfahren untereinander verglichen und eine Unterteilung in technologische Unterschiede wie Busse oder Punkt-zu-Punkt-Verbindungen und Intra- beziehungsweise Intersystemverfahren vorgenommen. Anschließend werden die gewonnenen Erkenntnisse auf verschiedene Clustersysteme angewendet und der Anteil der Inter- und Intrasystemkommunikation an der Gesamtverlustleistung bestimmt.
Kapitel 5 beschreibt zwei Clustersysteme, die mit Hilfe der gewonnenen Erkenntnisse aus den vorhergehenden Kapiteln entwickelt wurden. Das erste System stellt ein FPGA-Cluster dar, welcher eng gekoppelte rekonfigurierbare Architekturen einsetzt. RECS, ein ressourceneffizienter Cluster-Server, ist auf niedrigen Energiebedarf sowie auf eine hohe Packungsdichte an physikalischen Rechenknoten optimiert. RECS beinhaltet ein effizientes System zur Überwachung und Steuerung von Multiprozessorarchitekturen. Diese Entwicklung umgeht die Problematik der größer werdenden Kommunikationslast bei der Clusterüberwachung in Rechenzentren. Ein effizientes Langzeitarchivsystem auf Basis von Festplatten erweitert das RECS-System und bietet eine hohe Speicherdichte, bei gleichzeitig niedriger Leistungsaufnahme.
Kapitel 6 fasst die in dieser Arbeit gewonnenen Erkenntnisse zusammen und gibt einen Ausblick auf zukünftige Entwicklungen im Bereich effizienter Multiprozessorarchitekturen.

2 Kupferbasierte Datenübertragung in Multiprozessorarchitekturen

Heutige informationsverarbeitende Systeme bestehen in der Regel aus einer Vielzahl dedizierter Komponenten wie Prozessoren und Peripherie. Ein anschauliches Beispiel stellt ein Computer mit Hauptprozessor, Speicher und Schnittstellen dar. Soll eine Datei an einen anderen Computer gesendet werden, so muss ein Informationsaustausch zwischen allen beteiligten Komponenten wie Festplatte, Prozessor, Netzwerkkarte usw. stattfinden. Dies beinhaltet sowohl eine Kommunikation innerhalb des Systems als auch zwischen den einzelnen Computern. Die physikalische Realisierung einer Kommunikation zur Informationsübertragung bedingt immer drei grundlegende Komponenten in Form von Sender, Kanal und Empfänger. Um Daten über kupferbasierte Übertragungskanäle zu senden, muss der Kanal an einen Sender und Empfänger gekoppelt werden. Die Aufgabe des Senders besteht in der Ansteuerung des Kanals mit einer Spannungsform, welche das Signal darstellt. Der Empfänger detektiert diese Spannung am Ende des Kanals, um daraus das Signal zurückzugewinnen (siehe Abbildung 2.1). Ein einfacher Sender besteht im Wesentlichen aus Halbleiterschaltern, die eine Spannung auf den Kanal geben und einer Terminierung, um die Senderimpedanz an den Kanal anzupassen. Die Funktionalität kann durch beliebige Blöcke wie beispielsweise Filter oder Entzerrer erweitert werden, um die Datenübertragung zu optimieren. Dasselbe gilt für einen Empfänger, welcher den Kanal terminiert und die ankommende Spannung über einen Verstärker in ein Empfangssignal wandelt.

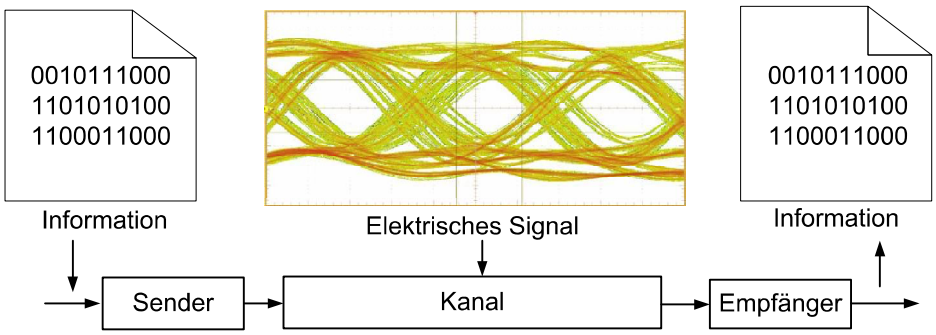

Abbildung 2.1: Der allgemeine Aufbau einer Übertragungsstruktur.

Der genaue Aufbau von Sender und Empfänger, sowie die physikalischen Größen zur Datenübertragung, Kanaltopologie und Signalkodierung sind in Übertragungsstandards festgeschrieben. Jeder dieser Standards implementiert die Datenübertragung auf eine spezielle Weise, was den jeweiligen Standard für bestimmte Szenarios geeigneter als andere macht. Deshalb werden in einem typischen Szenario, welches aus der Aneinanderreihung unterschiedlicher Kanäle (siehe Abbildung 2.2) besteht, auch unterschiedliche Übertragungsstandards und Verfahren angewandt. Ein Kanalmodell zur Beschreibung des Übertragungsverhaltens kann mathematisch hergeleitet werden und wird nachfolgend beschrieben.

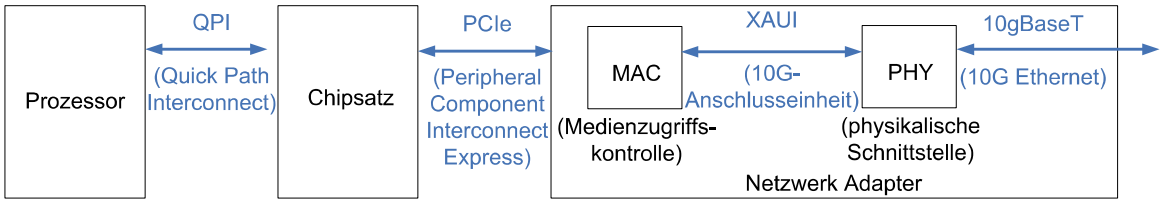

Abbildung 2.2: Bei einer Datenübertragung zwischen Systemen werden oft mehrere Verfahren hintereinander angewendet.

2.1 Modellierung von kupferbasierten Übertragungskanälen

Das zentrale Element für die Übertragung von Daten zwischen Kommunikationspartnern ist der Übertragungskanal. Er bildet neben den Transmittern, welche den Kanal mit Signalen speisen, die physikalische Schicht des Datentransfers. Ein kupferbasierter Kanal kann auf verschiedene Weise realisiert werden, z. B. als Kabel oder als Leiterbahn auf einer Platine (siehe Abbildung 2.3). Um das Verhalten eines Kanals und seinen Einfluss auf die Verlustleistung im Gesamtsystem beschreiben zu können, muss ein Modell erstellt werden, das alle relevanten Eigenschaften des Kanals korrekt beschreibt. Im Folgenden wird ein entsprechendes Modell ausgehend von der Betrachtung physikalischer und elektrotechnischer Grundlagen hergeleitet. Dieses Modell wird anschließend auf einige Implementierungen von Übertragungskanälen angewendet, um das spezifische Verhalten unterschiedlicher Kanäle herauszustellen. Die Informationen zur Herleitung der Kanalparameter stammen aus [R30] und [R13]. Simulationen und Graphen wurden mit *Maple* [R3], dem *Si9000 PCB Transmission Line Field Solver* [R1] und *HyperLynx SI* [R2] erstellt.

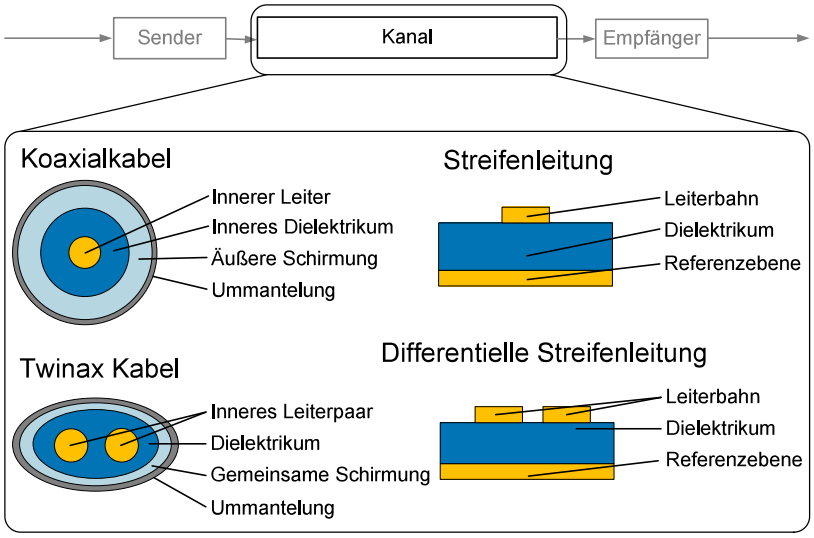

Abbildung 2.3: Gängige Implementierungsvarianten von Übertragungskanälen, links in Form von Kabeln und rechts in Form von Leiterbahnvarianten.

2.1.1 Kanalparameter und Effekte

Jeder kupferbasierte Übertragungskanal besteht aus einem Leiter, in dem ein Signal und damit ein Strom von der Quelle zum Ziel transportiert wird, und einem Leiter in Form einer Schirmung oder Referenzlage, in der ein Strom in entgegengesetzter Richtung zum ersten Leiter fließt. Diese zwei Leiter treten in einem Übertragungskanal miteinander in Wechselwirkung und rufen Effekte hervor, die zwei räumlich hinreichend getrennte Leiter nicht zeigen würden.

- Es bilden sich Kapazitäten zwischen den Leitern aus.
- Die Leiter zeigen induktive Eigenschaften.
- Es treten Verluste in Form von Strömen zwischen den Leitern auf, falls sie nicht voneinander isoliert sind.

Ein solches Leiterpaar kann als Aneinanderreihung von gleich langen Elementen modelliert werden, wobei jedes Element als eine Struktur aus einer Impedanz z in Reihe mit dem Signal und einer Admittanz y zwischen Leiter und Rückleiter betrachtet wird (siehe Abbildung 2.4). Die Impedanz z enthält den Serienwiderstand der Hin/- und Rückleitung R als auch die Induktivität L des Elements, hervorgerufen durch kombinierten Stromfluss im Hin/- und Rückleiter. Die Admittanz y enthält die parasitäre Kapazität zwischen dem Leiterpaar C und die Gleichstromverluste durch das Dielektrikum zwischen den Leitern, gegeben als Konduktanz G. Die jeweiligen Größen R, L, C und G repräsentieren den gesamten Widerstand, die Induktivität, die Kapazität und den Leitwert in einem Element des Übertragungskanals, wie in Abbildung 2.5 dargestellt. Wenn die Impedanz und Admittanz der Elemente bekannt ist, können die

Eingangsimpedanz, als charakteristische Impedanz Z_c und der Transferkoeffizient γ des gesamten Systems bestimmt werden. Diese Größen bestimmen die Antwort des Systems auf ein beliebiges digitales Signal.

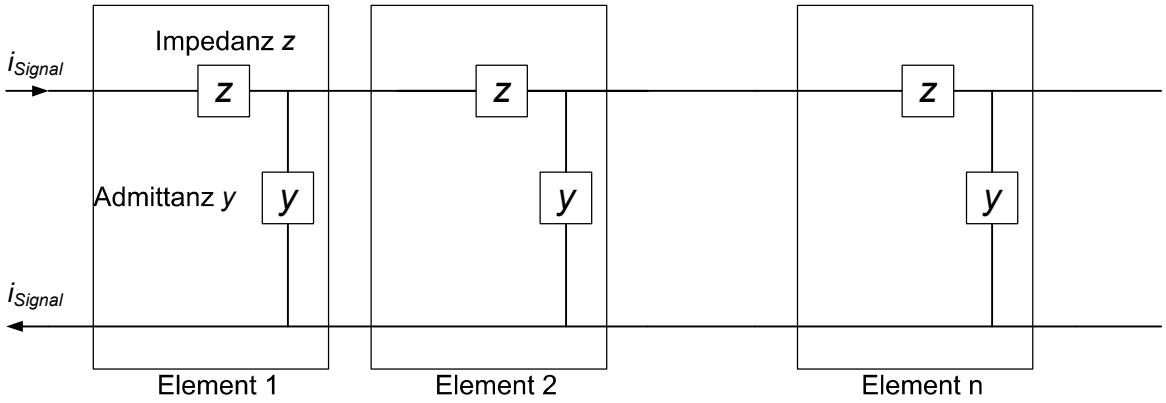

Abbildung 2.4: Ein Übertragungskanal kann als Aneinanderreihung von einzelnen Elementen modelliert werden [R13].

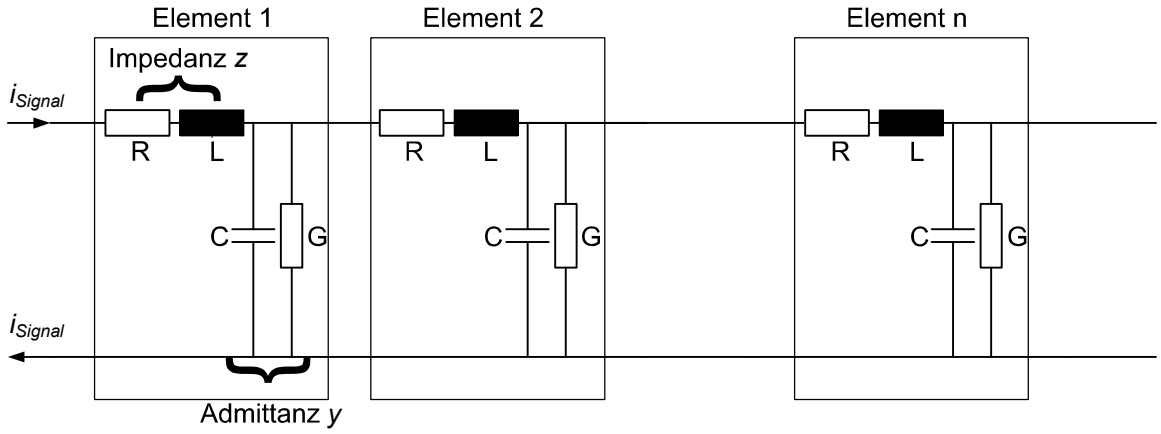

Abbildung 2.5: Die Impedanz z besteht aus dem Widerstand R und der Induktivität L. Die Admittanz y besteht aus dem Leitwert G und der Kapazität C [R13].

Die charakteristische Impedanz Z_c

Ein sich durch einen Kanal ausbreitendes Signal erzeugt an jedem Ort im Kanal eine bestimmte Spannung u und einen bestimmten Strom i. Das Verhältnis von Spannung u zu Strom i ist dabei immer gleich und wird als charakteristische Impedanz Z_c bezeichnet.

$$Z_c = \frac{u}{i} \tag{2.1}$$

Die charakteristische Impedanz Z_c ist abhängig von der Frequenz ω_0. Typischerweise wird für einen gegebenen Kanal, wie z. B. ein Koaxialkabel, eine charakteristische Impedanz Z_0 (z. B. 50Ω) bei einer bestimmten Frequenz ω_0 angegeben.

$$Z_0 = Z_c(\omega_0) \tag{2.2}$$

Um Z_c des Übertragungskanals, also einer Folge von infinitesimal kleinen Kanalelementen (siehe Abbildung 2.5), aus den Parametern R, L, G und C zu bestimmen, werden die Parameter eines Elements zu den Termen z und y zusammengefasst.

$$z = j\omega L + R \tag{2.3}$$

$$y = j\omega C + G \tag{2.4}$$

Das Hinzufügen von Elementen an eine Aneinanderreihung von infinitesimal kleinen Elementen verändert nicht die charakteristische Impedanz Z_c der Reihe. Das Anfügen eines Elements an den Anfang der Reihe, kann als Parallelschaltung der Admittanz y mit der charakteristischen Impedanz der Kette $\tilde{Z}_c$ und anschließenden Hinzufügen der Serienimpedanz z betrachtet werden. Da sich die charakteristische Impedanz der erweiterten Kette nicht ändert, ergibt sich:

$$\tilde{Z}_c = z + \frac{1}{\tilde{Z}_c{}^{-1} + y} \tag{2.5}$$

Durch weiteres Umformen ergibt sich:

$$\tilde{Z}_c = \sqrt{\frac{z}{y} + z\tilde{Z}_c} \tag{2.6}$$

Teilt man die einzelnen Elemente aus Abbildung 2.5 in n Unterelemente auf, so ändern sich die Werte der Elementparameter zu R/n, L/n, C/n und G/n. Die Parameter z und y verändern sich entsprechend zu z/n und y/n. Substituiert man z und y in der obigen Formel durch die neuen Werte und lässt n gegen unendlich laufen, erzeugt also infinitesimal kurze Elemente, so ergibt sich die charakteristische Impedanz eines Übertragungskanals zu:

$$Z_c = lim_{n\to\infty}\sqrt{\frac{z/n}{y/n} + \frac{z}{n}\tilde{Z}_c} \approx \sqrt{\frac{z}{y}} \tag{2.7}$$

$$\rightarrow Z_c(\omega) = \sqrt{\frac{j\omega L + R}{j\omega C + G}} \tag{2.8}$$

Bei digitalen Signalen, in denen sehr hohe Frequenzanteile vorhanden sind, wie z. B. steile Flankenübergänge, wird die charakteristische Impedanz durch die Parameter L und C dominiert. Es folgt für Z_0:

$$Z_0 := lim_{\omega \to \infty} Z_c(\omega) \approx \sqrt{\frac{L}{C}} \tag{2.9}$$

Die Ausbreitungsgeschwindigkeit v

Da die Ausbreitungsgeschwindigkeit von physikalischen Signalen von der Lichtgeschwindigkeit begrenzt ist, benötigt ein Signal, das sich in einem Kanal ausbreitet eine gewisse Zeit, um eine Strecke im Kanal zurückzulegen. Im Vakuum ist die Signalausbreitungsgeschwindigkeit v gleich der Lichtgeschwindigkeit c. Die Anwesenheit von magnetischen oder dielektrischen Materialien in einem Kanal verzögern die Ausbreitung des Signals zu:

$$v = \frac{c}{\sqrt{\epsilon_r \mu_r}} = \frac{1}{\sqrt{LC}} \tag{2.10}$$

- v ist die Signalausbreitungsgeschwindigkeit.
- c ist die Lichtgeschwindigkeit.
- ϵ_r gibt die relative Permittivität des Dielektrikums an.
- μ_r gibt die relative Permeabilität des Dielektrikums an.

Die meisten Übertragungskanäle verwenden nicht-magnetische Dielektrika, für die $\mu_r = 1$ gilt. Deshalb kann die Formel für die Ausbreitungsgeschwindigkeit v häufig vereinfacht werden zu:

$$v = \frac{c}{\sqrt{\epsilon_r}} \tag{2.11}$$

Reflektionen in Übertragungskanälen

Wenn ein durch den Kanal propagierendes Signal auf eine Veränderung der Kanalimpedanz trifft, so wird ein Teil des Signals reflektiert und breitet sich entgegengesetzt der ursprünglichen Richtung im Kanal aus. Ein anderer Teil des ursprünglichen Signals setzt den Weg hinter der Impedanzänderung fort. Solche Reflektionen treten beispielsweise an den Schnittstellen von Kanalsegmenten, wie z. B. Konnektoren auf. Sie sind ein Hauptgrund für Signalverzerrungen bei hochfrequenter Datenübertragung. Impedanzdiskontinuitäten müssen selbst in der einfachsten Übertragungsstruktur beachtet werden, da eine solche immer mindestens aus Signalquelle, Kanal und Signalempfänger besteht. Sender (Quelle) und Empfänger weisen genauso wie der Kanal eine Impedanz auf, was bei der Kopplung zwischen ihnen beachtet werden muss (siehe Abbildung 2.6).

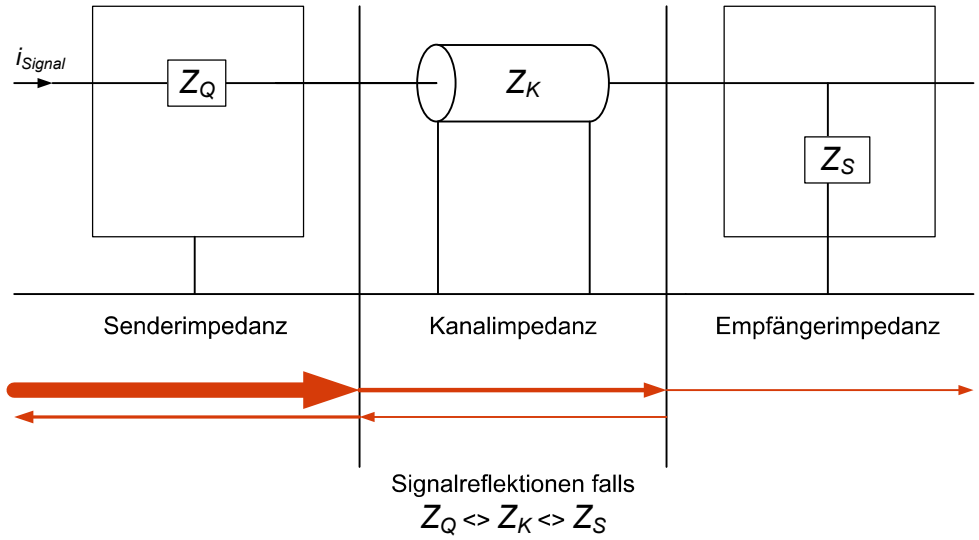

Abbildung 2.6: Wenn die einzelnen Impedanzen einer Übertragungsstruktur nicht identisch sind, so kommt es zu Signalreflektionen.

Der Anteil des Signals, der reflektiert wird, hängt von der Größe der Impedanzänderung ab. Das Verhältnis der Signalamplituden von dem reflektierten Signal zu dem ursprünglichen Signal wird als Reflektionskoeffizient ρ bezeichnet:

$$\rho = \frac{V_r}{V_0} = \frac{Z_2 - Z_1}{Z_2 + Z_1} \tag{2.12}$$

- V_r ist die Amplitude des reflektierten Signals.
- V_0 ist die Amplitude des ursprünglichen Signals.
- Z_1 bezeichnet die Impedanz des Kanals vor dem Impedanzsprung.
- Z_2 beschreibt die Impedanz des Kanals hinter dem Impedanzsprung.

Mit der Annahme das $V_0 + V_r = V_t$, kann der Transmissionskoeffizient t beschrieben werden als:

$$t = \frac{V_t}{V_0} = \frac{2Z_2}{Z_2 + Z_1} \tag{2.13}$$

- V_t ist die Amplitude des Signals hinter dem Impedanzsprung.
- V_0 ist die Amplitude des ursprünglichen Signals.
- Z_1 bezeichnet die Impedanz des Kanals vor dem Impedanzsprung.
- Z_2 beschreibt die Impedanz des Kanals hinter dem Impedanzsprung.

Es gibt drei Extremfälle für Impedanzsprünge in Übertragungskanälen. Im Folgenden wird angenommen, dass der Kanal eine Impedanz von $Z = 50\,\Omega$ aufweist.
Im ersten Fall wird ein offenes Ende des Kanals betrachtet (z.B. ein offenes Kabelende).

Die Impedanz hinter dem offenen Kanalende ist unendlich groß. Der Reflexionskoeffizient ist $\rho = \frac{V_r}{V_0} = \frac{\infty - 50\Omega}{\infty + 50\Omega} = 1$. Es wird also ein Signal mit derselben Amplitude wie die des ursprünglichen Signals am Impedanzsprung generiert und reflektiert. Direkt an der Diskontinuität findet eine Superposition der beiden Signale statt, weswegen hier eine Amplitude mit der doppelten Größe des Ursprungssignals V_0 gemessen werden kann. Abbildung 2.7 verdeutlicht das Szenario. Der zweite Extremfall ergibt sich bei einem kurzgeschlossenen Ende des Kanals. Die Impedanz hinter dem Kanalende ist Null. Der Reflexionskoeffizient ergibt sich zu $\rho = \frac{V_r}{V_0} = \frac{0 - 50\Omega}{0 + 50\Omega} = -1$. Es wird also ein Signal mit derselben Amplitude wie die des ursprünglichen Signals reflektiert, jedoch mit negativem Vorzeichen. Am Impedanzsprung löschen sich beide Signale aus (siehe Abbildung 2.8.

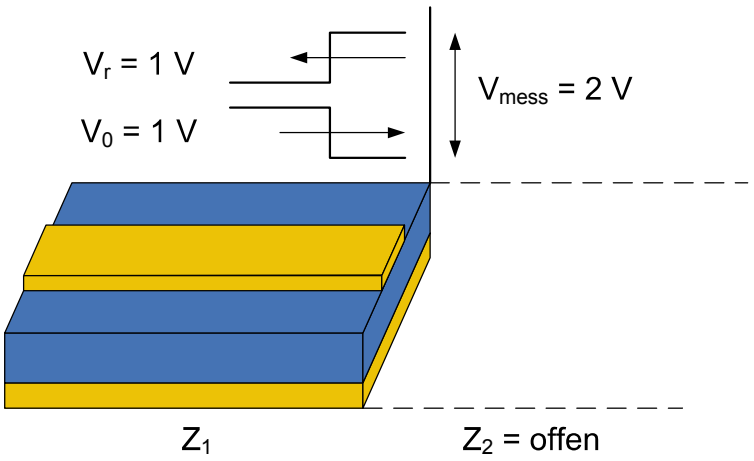

Abbildung 2.7: Wenn ein Signal auf ein offenes Kanalende trifft, so wird das komplette Signal reflektiert. Hin- und rücklaufendes Signal addieren sich auf [R13].

Der dritte Fall beschreibt einen Übergang zwischen zwei Kanalsegmenten, bei denen die Impedanzen einander gleichen. Hier ist der Reflexionskoeffizient $\rho = \frac{V_r}{V_0} = \frac{50 - 50\Omega}{50 + 50\Omega} = 0$. Das Signal passiert den Übergang ohne eine Reflektion hervorzurufen, es wird also am Übergang genau die ursprüngliche Spannung gemessen. Dieses Verhalten ist im Allgemeinen gewünscht, da am Ende eines Kanals das möglichst unverfälschte Originalsignal detektiert werden soll. Abbildung 2.9 zeigt die Reflektion eines 10 MHz Rechteck-Signals an einem offen Kanalende. Grün stellt das am Sender gemessene Signal dar und rot das Signal am offenen Ende des Kanals. Wie oben beschrieben wurde, ist die gemessene Amplitude am Kanalende mit 6 V doppelt so groß wie das Ursprungssignal mit 3 V Amplitude. Die abnehmende Oszillation des roten Signals entsteht durch multiple Reflektionen des Signals zwischen Sender und Kanalende.

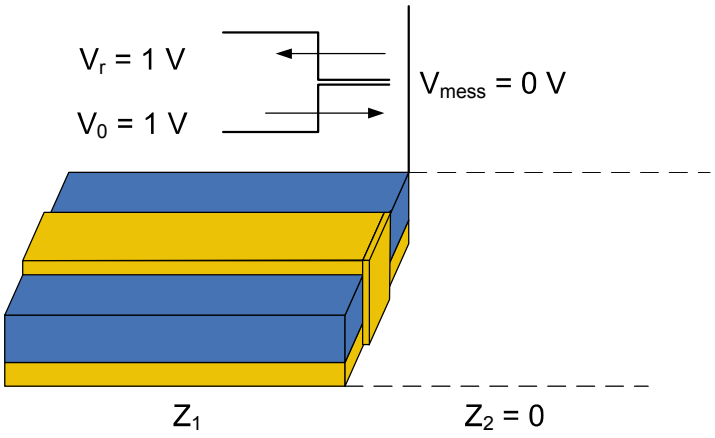

Abbildung 2.8: Wenn ein Signal auf ein kurzgeschlossenes Kanalende trifft, so wird das komplette Signal reflektiert. Hin- und rücklaufendes Signal heben sich jedoch auf [R13].

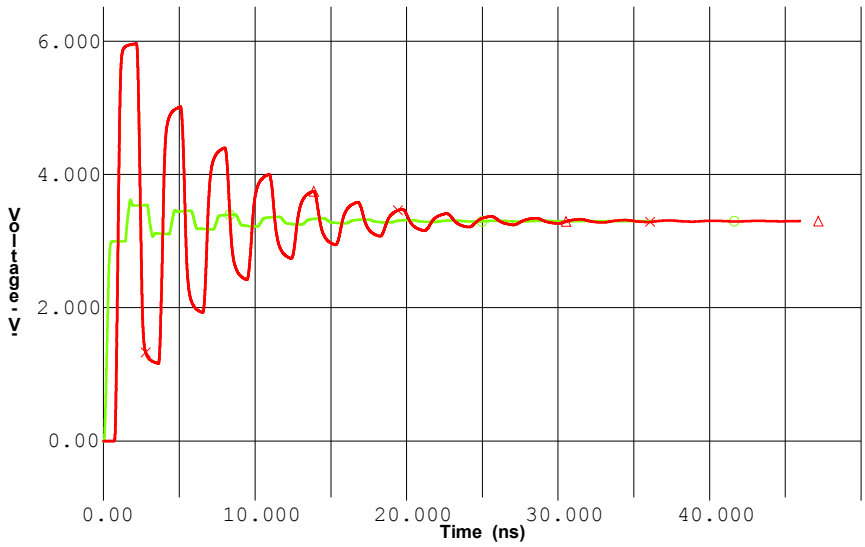

Abbildung 2.9: Eine totale Reflektion eines Signals (grün), gemessen an einem offenen Kanalende (rot) mit unendlich hoher Impedanz.

Der Gleichstromwiderstand R_{DC}

Jeder kupferbasierte Übertragungskanal weist einen Gleichstromwiderstand auf (siehe Abbildung 2.5). Dieser Widerstand wandelt als ohmsche Last Teile des Signals in Wärme um und trägt so zu den Verlusten im Kanal bei. Der Gleichstromwiderstand eines Leiters R_{DC} wird im Wesentlichen von seinem spezifischen Widerstand ρ und seiner Querfläche a bestimmt. Da in einem kupferbasierten Übertragungskanal immer auch ein Rückleiter in Form einer Schirmung oder Referenzlage zur Signalausbreitung beiträgt, muss der Widerstand dieses Rückleiters durch den Korrekturfaktor k_a mit in die Berechnung von R_{DC} eingebunden werden. Der effektive Gleichstromwiderstand ist also die Summe aus den Einzelwiderständen von Hin/- und Rückleiter. So ergibt sich für den Gleichstromwiderstand eines Übertragungskanals, gemessen in Ω/m:

$$R_{DC} = \frac{k_a \rho}{a} \tag{2.14}$$

Der Korrekturfaktor k_a bildet einen Quotienten aus der Summe der Einzelwiderstände von Hin/- und Rückleiter als Zähler und dem Widerstand des Hinleiters als Nenner. Beispielsweise ist bei einer Mikrostreifenleitung (siehe Abbildung 2.3), wo der Widerstand des Rückleiters aufgrund seiner großen Fläche kaum zum Tragen kommt, der Korrekturfaktor $k_a \approx 1$. Bei einer verdrillten Leitung, bei der ein Kabel als Hinleiter und das andere Kabel als Rückleiter dient, entspricht der Korrekturfaktor k_a folglich 2.
Hin/- und Rückleiter sind bei Übertragungskanälen durch einen Isolator bzw. ein Dielektrikum voneinander getrennt. Ein realer Isolator besitzt im Gegensatz zu einem idealen Isolator einen Gleichstromwiderstand $R < \infty$. Dies resultiert in einem Stromfluss zwischen Hin/- und Rückleiter, der zu Verlusten und Verzerrungen im Kanal beiträgt.

Der Skineffekt und der Wechselstromwiderstand R_{AC}

Fließt durch einen Leiter ein Gleichstrom, so ist in diesem Leiter die Stromdichte auf dem gesamten Querschnitt gleich. Der Stromfluss verteilt sich folglich gleichmäßig auf die gesamte Fläche des Leiters. Wechselt die Polarität, wie es bei Wechselstrom periodisch der Fall ist, verändert sich das Magnetfeld und erzeugt im Leiter Wirbelströme, die dem das Magnetfeld erzeugenden Strom entgegenwirken (siehe Abbildung 2.10). Im Leiter führt dieser Effekt dazu, dass der Strom in der Mittelachse des Leiters abgeschwächt wird. Die Elektronen in der Mitte des Leiters sind einem stärkeren Magnetfeld ausgesetzt als die Elektronen weiter außen. Dieser Effekt führt zu einer Verdrängung der freien Ladungsträger in die Randbereiche des Leiters und somit zu einer Verringerung des effektiven Leiterquerschnitts, bzw. zu einer Vergrößerung des

Widerstands der Leitung. Dieser Effekt ist proportional zur Frequenz. Bei sehr hohen Frequenzen wird der Skineffekt so stark, dass ein Stromfluss praktisch nur noch auf der Außenfläche des Leiters, also auf seiner „Haut“ stattfindet. Daher die Bezeichnung Skineffekt.

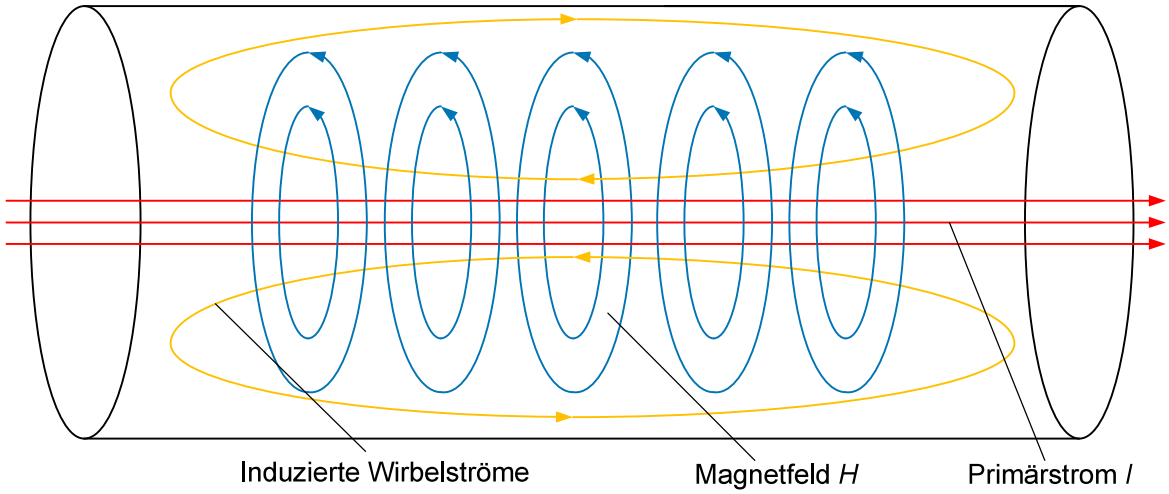

Abbildung 2.10: Ursache des Skineffekts: Ein von Wechselstrom durchflossener Leiter erzeugt ein magnetisches Wechselfeld. Dieses Feld induziert Wirbelströme, die den Stromfluss im Innern des Leiters abschwächen.

Die Strecke von der Außenkante eines Kanals bis zu der Tiefe, in der die Stromdichte um den Faktor *1/e* abgesunken ist, bezeichnet man als Skintiefe δ.

$$\delta = \sqrt{\frac{2}{\omega\mu\sigma}} = \frac{1}{\sqrt{\pi f \mu_0 \mu_r \sigma}} \tag{2.15}$$

- $\omega = 2\pi f$ beschreibt die Kreisfrequenz des Signals.
- μ ist die magnetische Permeabilität des Leiters.
- σ bezeichnet die Konduktivität des Leiters.

Die Skintiefe δ nimmt mit zunehmender Frequenz ab und erhöht dadurch den effektiven Wechselstromwiderstand R_{AC}. Es hängt also von der Signalfrequenz, dem Material und der Leiterfläche des Kanals ab, ob der Skineffekt Auswirkung auf den Wechselstromwiderstand R_{AC} zeigt oder nicht. D. h. ein runder Kanal kann ohne Betrachtung des Skineffekts bis zu der Frequenz ω_δ betrieben werden, bei der der Skineffekt zu Tage tritt.
Wenn der Übertragungskanal bei einer Frequenz $\omega > \omega_\delta$ betrieben wird, also der Skineffekt zu Tage tritt und den Wechselstromwiderstand R_{AC} beeinflusst, so ist dieser gegeben durch:

$$R_{AC} = \frac{k_p k_r}{p\delta\sigma} = \frac{k_p k_r \sqrt{\omega\mu}}{p\sqrt{2\sigma}} \tag{2.16}$$

- p beschreibt den Umfang des Leiters.
- δ bezeichnet die Skintiefe.
- σ ist die Konduktivität des Leiters.
- k_p gibt einen Korrekturfaktor an, der die Kopplung zwischen mehreren Leitern berücksichtigt.
- k_r gibt einen Korrekturfaktor an, der die Oberflächenrauheit des Leiters beschreibt.

Bei Gleichstrom und niedrigen Frequenzen kann der Serienwiderstand des Kanals durch R_{DC} beschrieben werden (siehe Kapitel 2.1.1). Bei hohen Frequenzen, welche den Skineffekt hervorrufen, wird der Serienwiderstand besser durch R_{AC} beschrieben. Die entsprechende Frequenz ω_δ ist gegeben durch:

$$\omega_\delta = \frac{2}{\mu\sigma}\left(\frac{k_a p}{k_p a}\right)^2 \tag{2.17}$$

- p beschreibt den Umfang des Leiters.
- a bezeichnet die Fläche des Leiters.
- σ ist die Konduktivität des Leiters.
- μ gibt die Permeabilität des Leiters an.
- k_a ist der Korrekturfaktor für en Gleichstromwiderstand (siehe Kapitel 2.1.1).
- k_p gibt einen Korrekturfaktor an, der die Kopplung zwischen mehreren Leitern berücksichtigt.
- R_{ser} beschreibt den Serienwiderstand des Kanals.

$$\rightarrow R_{ser} = \begin{cases} \frac{k_p k_r \sqrt{\omega\mu}}{p\sqrt{2\sigma}} = R_{AC} & ,\omega > \omega_\delta \\ \frac{k_a \rho}{a} = R_{DC} & ,\omega \leq \omega_\delta \end{cases} \tag{2.18}$$

Korrekturfaktor k_p

Ein von Strom durchflossener Leiter erzeugt ein Magnetfeld in seinem Innern und in seiner Umgebung. Bei Wechselstrom werden dadurch Wirbelströme im Leiter erzeugt, die den Skineffekt hervorrufen. Liegt im Wirkungsradius seines Magnetfeldes ein weiterer Leiter, so werden auch hier Wirbelströme zusätzlich zu den vom Leiter selbst induzierten Strömen erzeugt. Dies führt zu einer Profilveränderung des stromführenden Leiterbereichs. Der Stromfluss verlagert sich hierdurch zu den sich zugewandten Seiten der Leiter, wie in Abbildung 2.11 dargestellt. Durch die ortsabhängige Veränderung der Stromdichte im Leiter ändert sich auch der Wechselstromwiderstand R_{AC}.

Der Korrekturfaktor k_p wird typischerweise durch numerische Simulationen, wie z. B. Field-Solver, ermittelt. Das Verhalten von k_p kann wie folgt verallgemeinert werden:

- Ein einzelner Signalleiter mit ausreichendem Abstand zu seinem Rückleiter bzw. seiner Referenzlage wird durch einen Korrekturfaktor von $k_p \approx 1$ beschrieben.
- Eine differentielle Konfiguration weist einen Korrekturfaktor $k_p \approx 2$ auf.
- Werden Leiter und Referenzlage näher zusammen gebracht, so erhöht sich k_p.
- Unterhalb der Frequenz ω_δ ist $k_p = 0$, da hier kein Skineffekt auftritt.

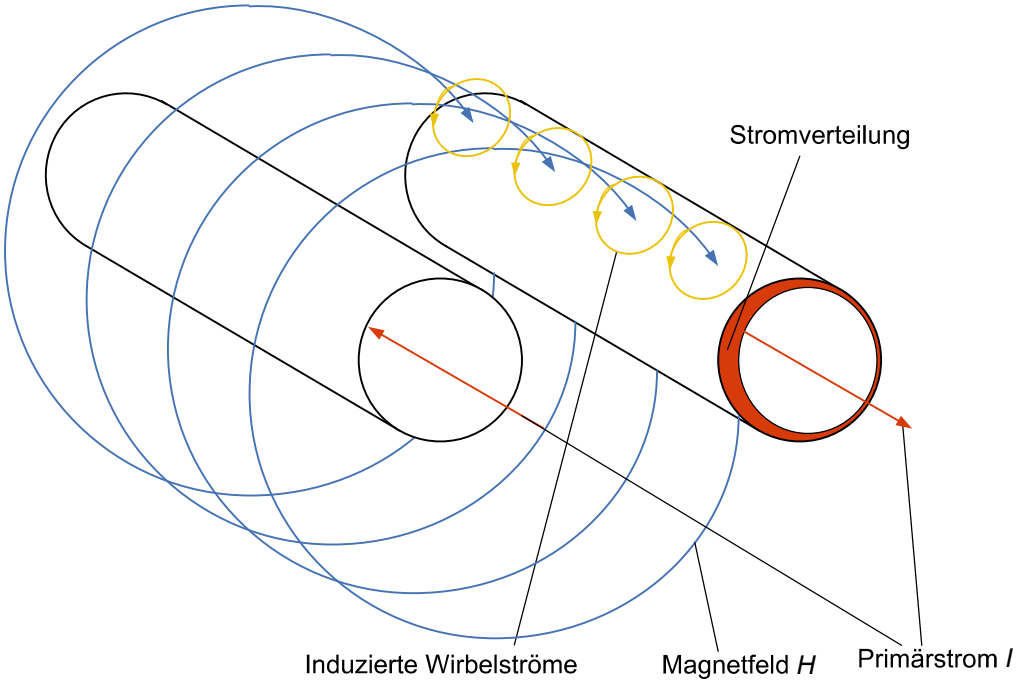

Abbildung 2.11: Ein sich veränderndes magnetisches Feld eines Leiters induziert Wirbelströme in einem anderen Leiter. Dieses bewirkt eine Veränderung der Stromverteilung in diesem Leiter.

Korrekturfaktor k_r

Auf mikroskopischer Ebene weist kein Material eine perfekt glatte Oberfläche auf. Jede Leiteroberfläche besitzt Erhebungen und Rillen, wobei die Rauheit des Leiters über den quadratischen Mittelwert der Unebenheiten h_{RMS} gegeben ist. Unterhalb der Frequenz ω_δ hat die Oberflächenrauheit keinen Einfluss auf das Übertragungsverhalten des Kanals. Wenn jedoch bei hohen Frequenzen der Skineffekt zu Tage tritt und die Skintiefe δ h_{RMS} unterschreitet, so erhöht sich der Widerstand des Leiters. Dies ist darauf zurückzuführen, dass der Strom jeder Unebenheit in der Oberfläche folgt und so eine längere Strecke im Leiter zurücklegen muss (siehe Abbildung 2.12).

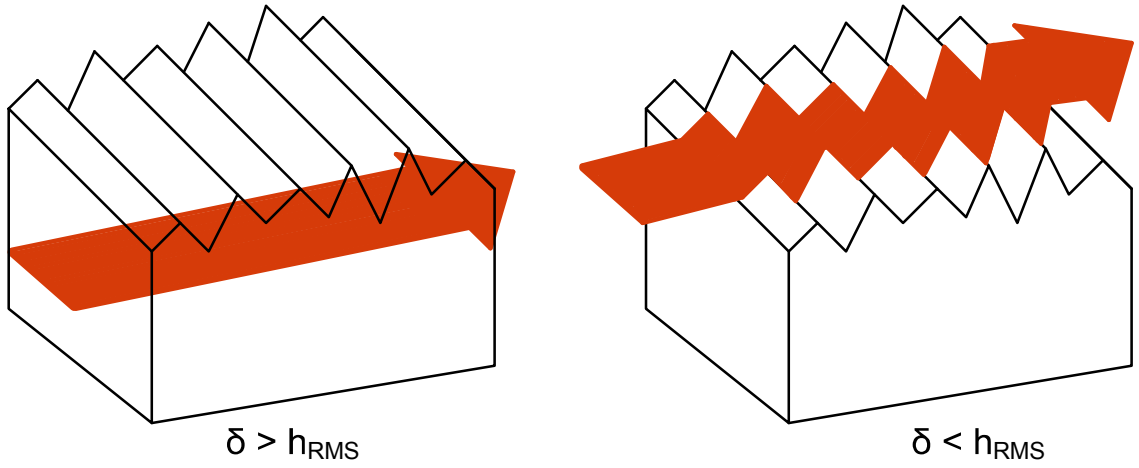

Abbildung 2.12: Links: Skintiefe ist größer als die mittlere Oberflächenrauheit des Kanals. Rechts: Skintiefe ist kleiner als die mittlere Kanalrauheit. Der Strom muss eine längere Strecke zurücklegen, weshalb sich der Wechselstromwiderstand R_{AC} erhöht.

Die Frequenz, bei der die Oberflächenrauheit zum Tragen kommt, ist gegeben durch:

$$\omega_{rau} = \frac{2}{\mu\sigma h_{rms}^2} \tag{2.19}$$

Mit Hilfe von ω_{rau} kann der Korrekturfaktor k_r angegeben werden mit:

$$k_r = 1 + \frac{2}{\pi} \arctan(1,4\frac{\omega}{\omega_{rau}}) \tag{2.20}$$

Als Beispiel sei die Rauheit eines auf FR-4 Material (mit Epoxidharz getränkte Glasfasermatten) aufgedampften Leiters gegeben. h_{RMS} ist mit ca. $5\,\mu m$ angegeben. Das heißt ω_{rau} liegt bei circa 1 GHz. Ab dieser Frequenz muss also die Rauheit des Materials mit in die Widerstandsberechnung eingebunden werden. Abbildung 2.13 zeigt den Widerstand eines Kupferkanals pro Meter Leitungslänge in Abhängigkeit der Frequenz. Bei einer Frequenz von ca. 100 MHz teilt sich der Graph in zwei Äste auf. Der grüne Ast berücksichtigt die Widerstandserhöhung aufgrund des Skineffekts. Der rote Ast berücksichtigt neben dem Skineffekt auch die Oberflächenrauheit des Leitermaterials, was sich in einer Erhöhung des Widerstands zeigt.

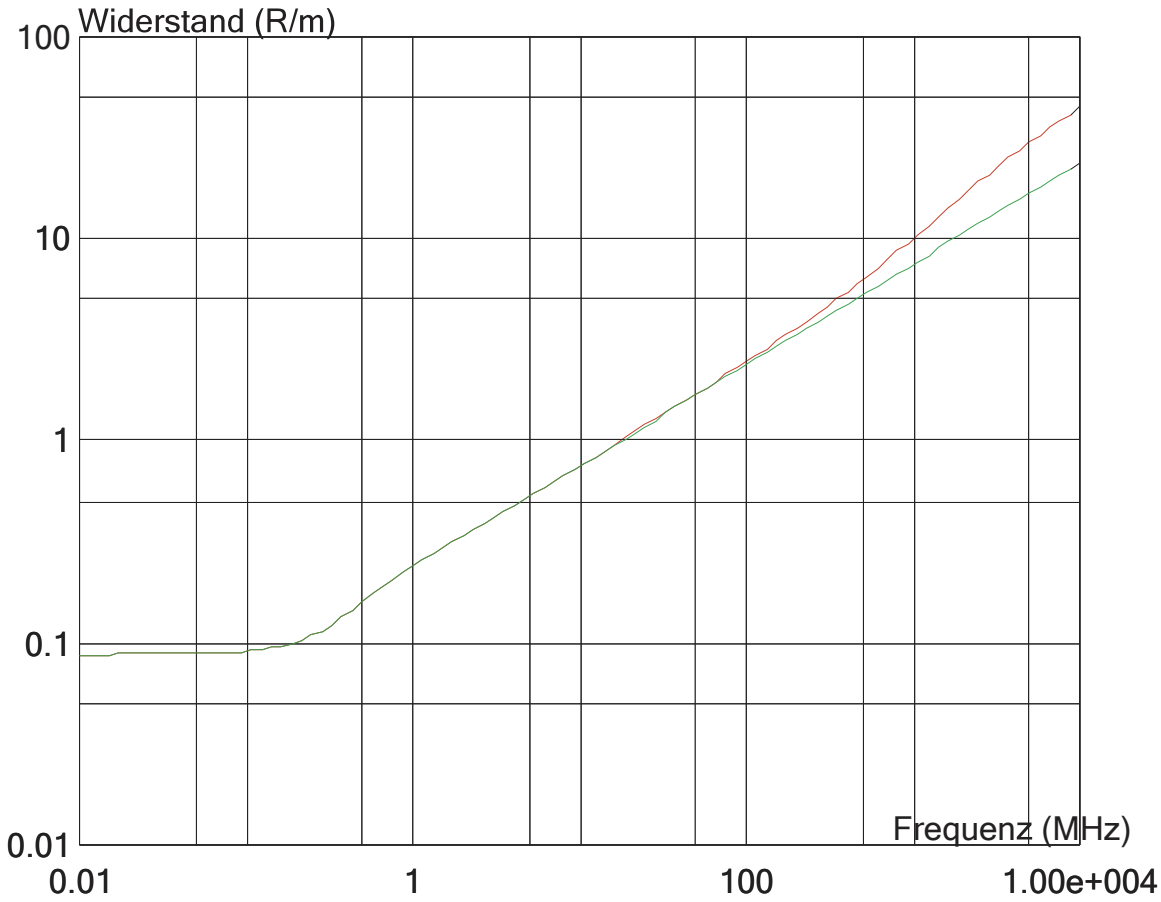

Abbildung 2.13: Der beispielhafte Widerstand eines Leiters pro Meter (grün) in Abhängigkeit der Frequenz. Der rote Graph berücksichtigt die Oberflächenrauheit des Materials.

Dielektrische Verluste

Bei einem kupferbasierten Übertragungskanal sind Hinleiter und Rückleiter, bzw. Leiter und Referenzlage, durch ein als Isolator dienendes Dielektrikum voneinander getrennt. Im Gegensatz zu einem idealen Isolator weisen Dielektrika Verluste auf, die das Signal verzerren. Die Verluste werden zum Einen durch die Leitfähigkeit des Materials bestimmt, also der Fähigkeit einen Strom zu leiten, und zum Anderen von der Materialpermittivität, also der Durchlässigkeit von elektrischen Feldern. Abbildung 2.14 zeigt die Messung eines Stromes durch ein Dielektrikum bei angelegter Wechselspannung.

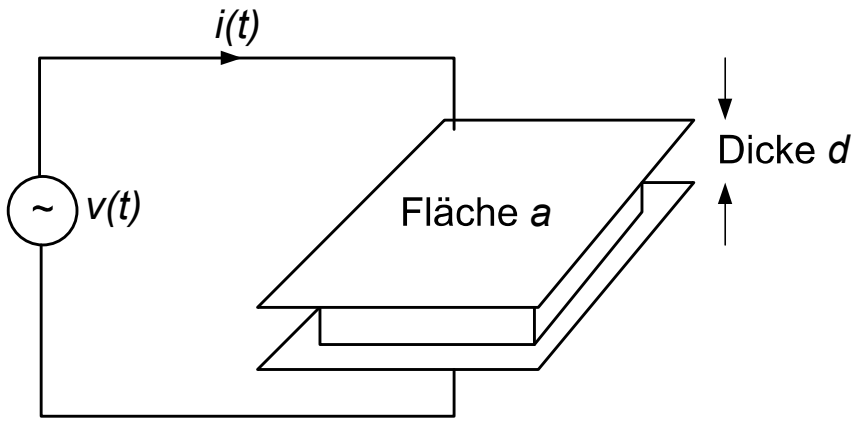

Abbildung 2.14: Eine an ein Dielektrikum angelegte Wechselspannung erzeugt einen Stromfluss in Phase und außer Phase zur Wechselspannung.

Der in Abbildung 2.14 resultierende Stromfluss ist gegeben durch:

$$I(\omega) = U(\omega)\frac{a}{d}(\delta + j\omega\epsilon) \tag{2.21}$$

- $I(\omega)$ ist die Stromstärke in Abhängigkeit der Frequenz ω.
- $U(\omega)$ ist die angelegte Spannung in Abhängigkeit der Frequenz ω.
- a ist die Fläche des Dielektrikums.
- d ist die Dicke des Dielektrikums.
- δ ist die Leitfähigkeit des Dielektrikums.
- ϵ ist die Permittivität des Dielektrikums.

Der Stromfluss in Phase zur Wechselspannung wird durch δ bestimmt und als *Leitungsstrom* bezeichnet. Er beschreibt das resistive Verhalten des Dielektrikums. Der Stromanteil mit unterschiedlicher Phase gegenüber der Wechselspannung wird *Verschiebungsstrom* genannt und beschreibt das kapazitive Verhalten des Dielektrikums.

Er ist bestimmt durch $\omega\epsilon$. Während δ unabhängig von der Frequenz ist, wächst der Verschiebungsstrom mit steigender Frequenz an. Ab einer kritischen Frequenz $\omega_c = \delta/\epsilon$ dominiert das Wachstum von $\omega\epsilon$ das Wachstum von δ. Beide Größen wachsen linear mit der Frequenz. Wenn δ durch $\omega\epsilon''$ substituiert wird, kann der resultierende Stromfluss beschrieben werden als:

$$I(\omega) = U(\omega)\frac{a}{d}(\omega\epsilon'' + j\omega\epsilon') = U(\omega)\frac{a}{d}(j\omega(\epsilon' - j\epsilon'')) \tag{2.22}$$

Der Ausdruck $\epsilon = \epsilon' - j\epsilon''$ wird als komplexe Permittivität bezeichnet, wobei ϵ' den Verschiebungsstrom bezeichnet und $-\epsilon''$ den Leitungsstrom.
Dielektrische Verluste in Übertragungskanälen werden durch den Ausdruck *dielektrische Verlusttangente* beschrieben, welche den Tangens des Winkels zwischen Leitungsstromphase und Verschiebungsstromphase definiert (siehe Abbildung 2.15).

$$tan\Theta = \frac{-Im(\epsilon)}{Re(\epsilon)} = \frac{\epsilon''}{\epsilon'} \tag{2.23}$$

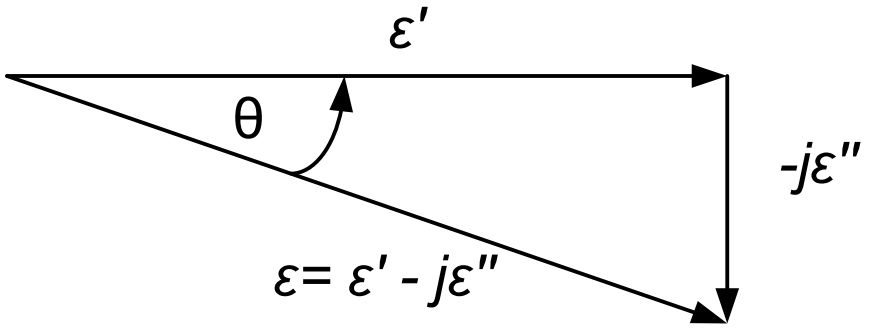

Abbildung 2.15: Die Komponenten der komplexen Permittivität sind so definiert, dass ϵ'' immer positiv ist und der Winkel Θ ebenfalls.

Der Ausbreitungskoeffizient γ

Signale, die sich in einem Übertragungskanal ausbreiten, werden um einen Faktor H pro Einheit des Kanals in ihrer Amplitude gedämpft. Da die Dämpfung in jeder Einheit dieselbe ist, steigt sie exponentiell mit der Länge des Kanals. Der frequenzabhängige Faktor H pro Längeneinheit bildet die sogenannte Übertragungsfunktion *H(ω)*. Sie repräsentiert die Dämpfung eines Signals durch ein Kanalelement bei gegebener Frequenz. Analog zu *H(ω)* gibt *H(ω,l)* die Dämpfung des Signals bei zusätzlich ange-

gebener Kanallänge an. Da die Dämpfung exponentiell mit der Kanallänge zunimmt, gilt:

$$H(\omega, l) = (H(\omega))^l \tag{2.24}$$

Der Ausbreitungskoeffizient γ ist definiert als der Logarithmus der Übertragungsfunktion *H(ω)*.

$$\gamma(\omega) := -\ln H(\omega) \tag{2.25}$$

$$H(\omega) = e^{-\gamma(\omega)} \tag{2.26}$$

$$\rightarrow H(\omega, l) = e^{-l \cdot \gamma(\omega)} \tag{2.27}$$

Um den Ausbreitungskoeffizienten γ aus den Werten *R/n*, *L/n*, *C/n* und *G/n* zu bestimmen, wird das Modell des Kanals erweitert. Abbildung 2.16 zeigt eine Einheit eines Übertragungskanals, an die ein weiteres Element mit einer Eingangsimpedanz Z_i gekoppelt ist. Dieses weitere Element kann beispielsweise einen anderen Kanal darstellen. Wird *z'* wie in Abbildung 2.16 gezeigt definiert, dann berechnet sich die Übertragungsfunktion $\tilde{H}$ wie folgt:

$$\tilde{H} = \frac{z'}{z + z'} \tag{2.28}$$

Ersetzt man *z'* durch die parallele Kombination aus der Impedanz Z_i und der Admittanz y und substituiert Z_i durch die Definition der Impedanz, so ergibt sich die Übertragungsfunktion der Struktur als:

$$\tilde{H} = \frac{\dfrac{1}{y + \sqrt{y/z}}}{z + \dfrac{1}{y + \sqrt{y/z}}} \tag{2.29}$$

$$\rightarrow \tilde{H} = \frac{1}{zy + \sqrt{zy} + 1} \tag{2.30}$$

Nun kann diese Struktur, wie schon in Kapitel 2.1.1 beschrieben, in n Blöcke gespalten werden, jeder mit der Länge *1/n*. Die Parameter z und y verändern sich dadurch zu *z/n* und *y/n*. Die kombinierte Übertragungsfunktion von n Blöcken entspricht der Übertragungsfunktion eines Blockes der Länge *1/n* mit exponentiellem Wachstum von n.

$$\tilde{H} = lim_{n\to\infty} \left[\frac{1}{(z/n)(y/n) + \sqrt{(z/n)(y/n) + 1}} \right]^n = lim_{n\to\infty} \left[\frac{(zy/n) + \sqrt{zy}}{n} + 1 \right]^{-n} \tag{2.31}$$

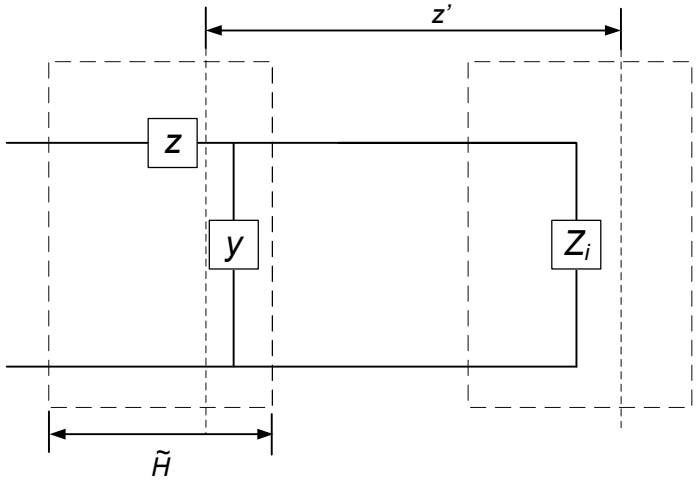

Abbildung 2.16: z' wird definiert als Eingangsimpedanz Z_i mit paralleler Admittanz y [R13].

Unter der Annahme das $lim_{n\to\infty}((a/n)+1)^{-n} = e^{-a}$, ergeben sich die Übertragungsfunktion H und der Ausbreitungskoeffizient γ zu:

$$H = e^{-\sqrt{zy}} \tag{2.32}$$

$$\gamma = \sqrt{zy} \tag{2.33}$$

Die Substitution von z und y durch ihre Definitionen ergibt den Ausbreitungskoeffizienten γ in Abhängigkeit von R, L, C und G als:

$$\gamma(\omega) = \sqrt{(j\omega L + R)(j\omega C + G)} \tag{2.34}$$

Dieser Ausdruck kann in seinen Realteil und seinen Imaginärteil aufgespalten werden, wobei der Realteil die Amplitudendämpfung des Signals und der Imaginärteil die Phasenverschiebung des Signals beschreibt. Schließlich ist die Übertragungsfunktion *H(ω,l)* eines Kanals in Abhängigkeit von R, L, C und G gegeben als:

$$H(\omega, l) = e^{-l\sqrt{(j\omega L + R)(j\omega C + G)}} \tag{2.35}$$

Die Dämpfung eines Signals in einem Kanal ist in Abbildung 2.17 gezeigt. Der rote Graph gibt hierbei die resistiven Verluste an, der grüne Graph repräsentiert dielektrische Verluste und der blaue Graph gibt die gesamten Verluste an. Die dielektrischen Verluste dominieren ab einer Frequenz von 30 MHz. Abbildung 2.18 fasst die vorgestellten Parameter und die Beteiligung an den Kanalverlusten zusammen.

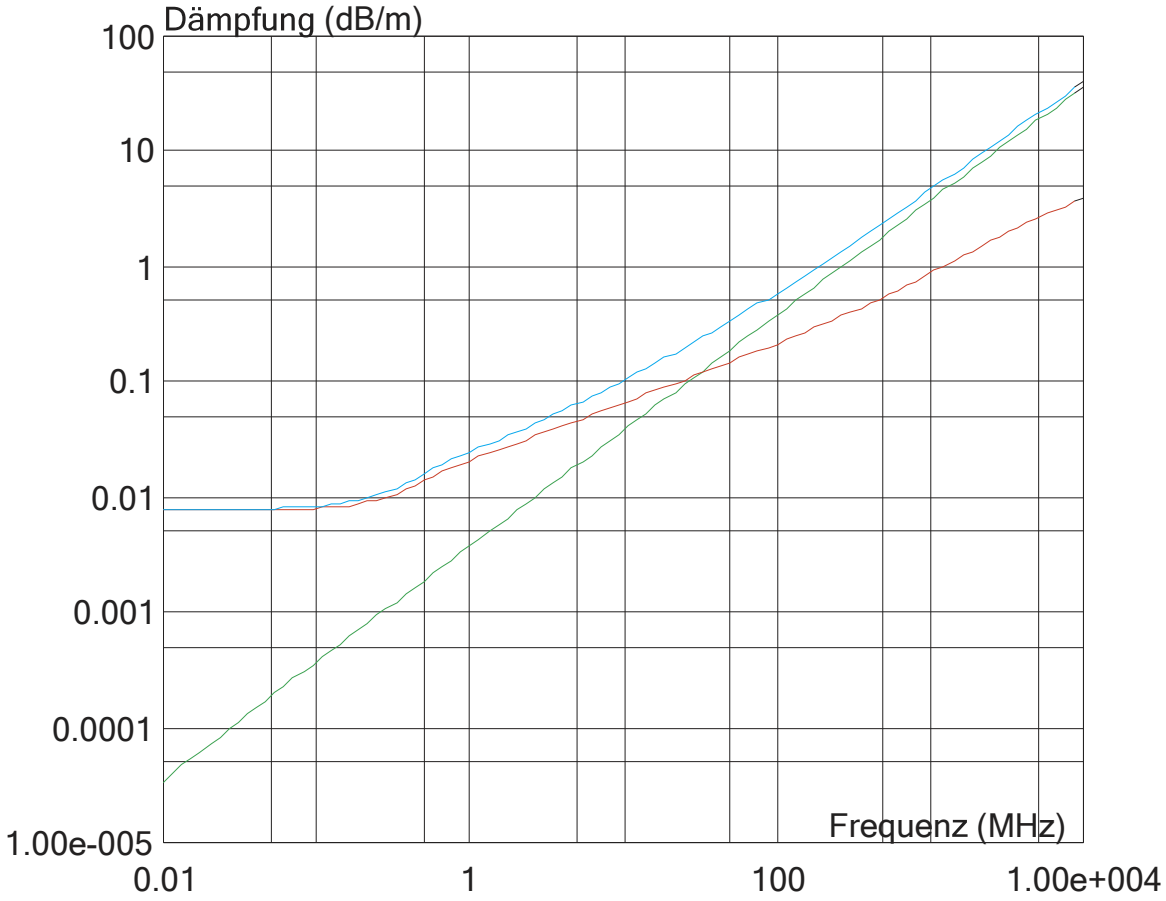

Abbildung 2.17: Signalverluste in einem Kanal aufgrund resisitiver und dielektrischer Effekte: rot: resistive Verluste, grün: dielektrische Verluste, blau: akkumulierte Verluste.

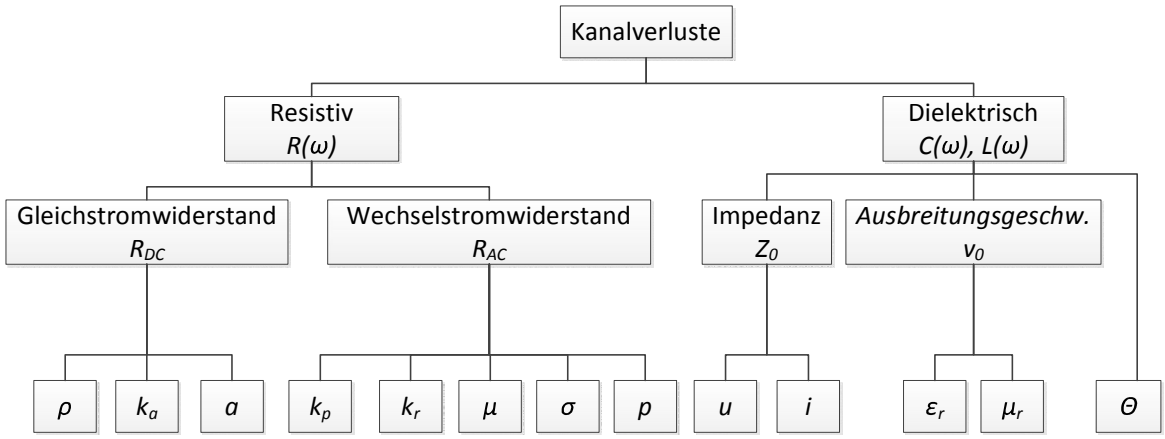

Abbildung 2.18: Übersicht der an Kanalverlusten beteiligten Größen.

2.1.2 Die allgemeine, frequenzabhängige Übertragungsfunktion eines kupferbasierten Übertragungskanals

In Kapitel 2.1.1 wurden wichtige Eigenschaften und Größen eines kupferbasierten Übertragungskanals vorgestellt, mit denen die Übertragungsfunktion eines Kanals beschrieben werden kann. Mit der Übertragungsfunktion kann die Dämpfung eines Signals, also der Kanalverlust, gut beschrieben werden. Allerdings wird dabei von frequenzunabhängigen Größen L und C ausgegangen. Ein realer Übertragungskanal weist jedoch eine frequenzabhängige Induktivität und Kapazität auf. Die Übertragungsfunktion muss also um frequenzabhängige Größen erweitert werden. Zuerst wird der Ausdruck R_{AC} (siehe 2.1.1), welcher den Skineffekt-Widerstand bei einer gegebenen Frequenz angibt, erweitert, um einen Frequenzbereich abzudecken.

$$R_{AC}(\omega) = R_0\sqrt{\frac{2j\omega}{\omega_0}} \tag{2.36}$$

- R_0 ist der Wechselstromwiderstand bei der Frequenz ω_0.
- R_{AC} ist der komplexe, frequenzabhängige Widerstand, welcher mit der Quadratwurzel der Frequenz steigt.

Der Widerstand des Kanals wird unterhalb von Frequenzen, die den Skineffekt hervorrufen, durch R_{DC} dominiert. Oberhalb dieser Frequenzen dominiert R_{AC}. Der frequenzabhängige Widerstand eines Kanals kann durch die Wurzel aus der Summe der Quadrate von R_{DC} und R_{AC} angegeben werden.

$$R(\omega) = \sqrt{(R_{DC})^2 + (R_{AC}(\omega))^2} \tag{2.37}$$

Ein Kanal wird zudem durch eine frequenzabhängige Induktivität gekennzeichnet, die sich in eine charakteristische Induktivität L_0 bei der Frequenz ω_0 und einen frequenzabhängigen Teil aufteilt. Der frequenzabhängige Teil wird durch den imaginären Teil von R_ω beschrieben. L_0 ist eine Konstante, welche durch die charakteristische Impedanz Z_0 und die Ausbreitungsgeschwindigkeit v_0 gegeben ist.

$$L_0 = \frac{Z_0}{v_0} \tag{2.38}$$

Neben der Induktivität wird der Kanal durch eine frequenzabhängige Kapazität beschrieben, welche eine charakteristische Kapazität C_0 enthält.

$$C_0 = \frac{1}{Z_0 v_0} \tag{2.39}$$

Die komplexe, frequenzabhängige Kapazität ergibt sich zu:

$$C_\omega = C_0 \left(\frac{j\omega}{\omega_0} \right)^{-\frac{2\Theta}{\pi}} \tag{2.40}$$

Der komplexe, frequenzabhängige Ausbreitungskoeffizient $\gamma(\omega)$ kann nun, ähnlich wie in Kapitel 2.1.1, angegeben werden zu:

$$\gamma(\omega) = \sqrt{(j\omega L_0 + R_\omega)(j\omega C_\omega)} \tag{2.41}$$

Der Realteil $\alpha = Re[\gamma(\omega)]$ gibt die Dämpfung des Signals an, der Imaginärteil $\beta = Im[\gamma(\omega)]$ beschreibt die Phasenverzögerung. Die Übertragungsfunktion in Abhängigkeit der Frequenz ω und der Kanallänge l ist gegeben durch:

$$H(\omega, l) = e^{-l\gamma(\omega)} \tag{2.42}$$

Die Signaldämpfung wird häufig in dB angegeben als:

$$\alpha(\omega) = -20log(|H(\omega, l)|) = \frac{20l}{ln(10)} Re[\gamma(\omega)] \tag{2.43}$$

Wenn die charakteristische Impedanz als Funktion der Frequenz ω modelliert wird, so ergibt sie sich zu:

$$Z_c(\omega) = \sqrt{\frac{j\omega L_0 + R(\omega)}{j\omega C(\omega)}} \tag{2.44}$$

Die Kanalparameter R, L, C und G sind in die einzelnen Terme des Ausbreitungskoeffizienten und der charakteristischen Impedanz eingebunden und können über die folgende Tabelle errechnet werden:

Tabelle 2.1: R, L, C und G eines Kanals, bestimmt bei der Frequenz ω_n

Parameter	Wert	Einheit
Serienwiderstand	$R_n = Re[j\omega_n L_0 + R(\omega_n)]$	Ω/m
Induktivität	$L_n = Im[j\omega_n L_0 + R(\omega_n)]/\omega_n$	H/m
Kapazität	$C_n = Im[j\omega_n C(\omega_n)]/\omega_n$	F/m
Leitwert	$G_n = Re[j\omega_n C(\omega_n)]$	S/m

Verifikation des Modells

Um das Kanalmodell zu verifizieren, wurde der Übertragungskanal mit einem sogenannten Field-Solver simuliert und die Signaldämpfung im resistiven und dielektrischen Bereich mit den Ergebnissen der Simulation verglichen. Ein Field-Solver ermittelt auf Basis der physikalischen Abmessungen der Struktur auf nummerischem Wege die Kanalparameter, dies geschieht durch Lösung der Maxwellschen Feldgleichungen. Das vorgestellte Modell abstrahiert den Kanal und kommt ohne die Lösung von Feldgleichungen aus. Ein Field-Solver ist dadurch immer genauer als das vorgestellte Modell, da keine Vereinfachungen und Abstraktionen vorgenommen werden, die Berechnung ist jedoch ungleich aufwendiger und nur durch numerische Simulationen durchführbar. Im Modell und in der Simulation wurde ein Kanal mit den Parametern aus Tabelle 2.2 betrachtet.

Tabelle 2.2: Übersicht der verwendeten Kanalparameter und physikalischen Größen.

Parameter	Bezeichnung	Wert
Leiterbahnbreite	W	$101,6\,\mu$m
Leiterbahndicke	T	$34,3\,\mu$m
Dielektrikumsdicke	H	$254\,\mu$m
Spezifischer Widerstand	ρ	$1,68\,\frac{\Omega\cdot\mathrm{mm}^2}{\mathrm{m}}$
Permeabilität	μ	$12,56\cdot10^{-7}\,\frac{\mathrm{H}}{\mathrm{m}}$
Relative Permittivität	ϵ_r	$4,3$
Kanallänge	LL	1000 cm
Leiterkonduktivität	σ	$5,8\cdot10^7\,\mathrm{S/m}$
Verlusttangente	$tan(\Theta)$	$0,02$
Frequenzbereich	f	100 MHz bis 10 GHz
Oberflächenrauheit	h_{rms}	$5,4\,\mu$m
Korrekturfaktor	k_p	1
Korrekturfaktor	k_a	2
Kapazität	C_0	58,6 pF
Induktivität	L_0	526 nH

Der Rückleiter des Kanals besitzt die gleichen Abmessungen wie der Hinleiter, deshalb wird der Korrekturfaktor für den Gleichstromwiderstand $k_a = 2$ gesetzt. Außerdem liegt keine differentielle Konfiguration vor, also gilt $k_p = 1$. Mit den Gleichungen 2.9 und 2.10 ergibt sich $v = 180050414\,\frac{\mathrm{m}}{\mathrm{s}}$ und $Z = 94.76\,\Omega$. Die Frequenz, ab welcher der Skineffekt zu Tage tritt, bestimmt sich über Gleichung 2.17 zu $\omega_\delta = 667\,\mathrm{MHz}$. Der Einfluss der Oberflächenrauheit wird in Form des Korrekturfaktors k_r berechnet und ergibt sich über die Gleichungen 2.19 und 2.20 zu $k_r(\omega_\delta) = 1,497$. Mit den ermittelten Werten stehen genug Informationen zur Verfügung um alle anderen relevanten Größen zu berechnen und die Übertragungsfunktion über Gleichung 2.42 zu bestimmen. In

Abbildung 2.19 ist die sich ergebende Signaldämpfung des betrachteten Kanals in Abhängigkeit der Signalfrequenz gezeigt. Die Signaldämpfung teilt sich in resistive Verluste, hervorgerufen durch den Serienwiderstand R_n, und dielektrische Verluste, hervorgerufen durch den Leitwert G_n auf. Die Simulationsergebnisse sind ebenfalls dargestellt.

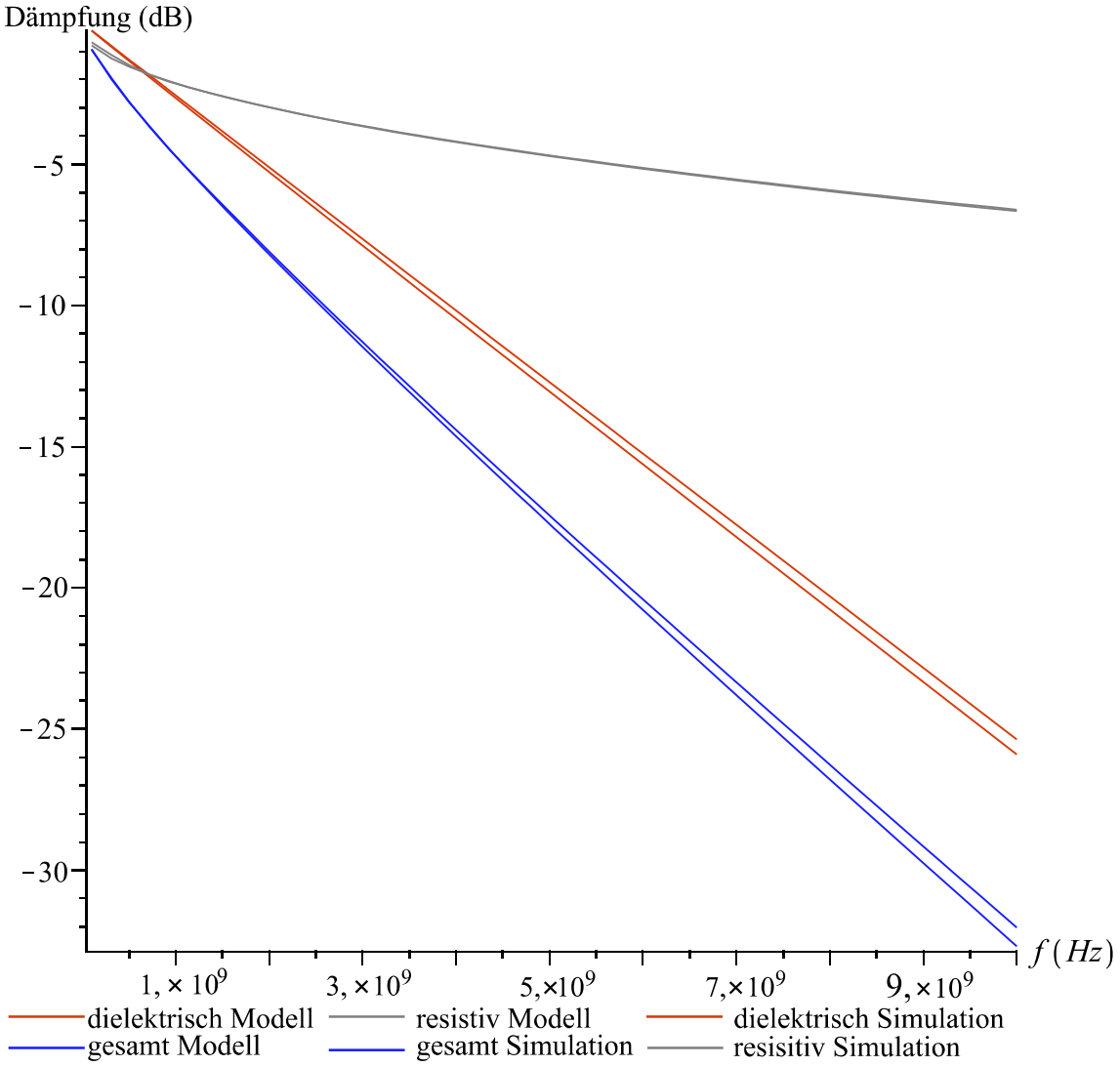

Abbildung 2.19: Berechnete und simulierte Verluste in einem kupferbasierten Übertragungskanal.

Die Ergebnisse des Modells und der Simulation stimmen gut überein, sodass ein kupferbasierter Übertragungskanal mit dem vorgestellten Modell beschrieben werden kann. Mit dem erstellten Modell kann eine erste qualitative Aussage über die Verluste in verschiedenen Kanälen getroffen werden. Drei unterschiedliche Kanalvarianten werden betrachtet (siehe Abbildung 2.20), wobei die physikalischen Abmessungen des Signal-

leiters in jeder Variante gleich sind. Dasselbe gilt bei den ersten beiden Varianten auch für den Abstand zwischen den Leitern, beziehungsweise zwischen Signalleiter und Referenzleiter. Bei Variante drei wird der Leiterabstand als hinreichend gering angenommen, so dass eine Stromverschiebung aufgrund des Skineffekts auftritt.

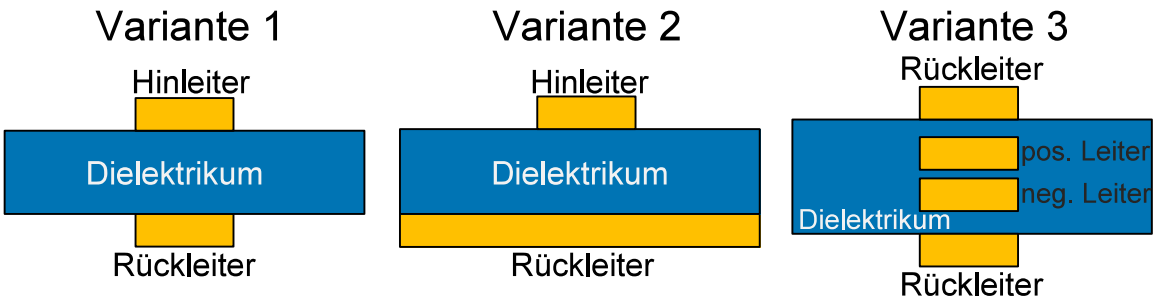

Abbildung 2.20: Die untersuchten Kanalkonfigurationen.

- Variante 1: Zwei übereinander liegende Leiter, von denen ein Leiter den Signalpfad bildet und der andere als Referenzlage dient. Die Leiter sind durch ein Dielektrikum getrennt. Die Korrekturfaktoren werden gesetzt zu: $k_a = 2$ $k_p = 1$.
- Variante 2: Eine Konfiguration aus Leiter und Kupferfläche, von denen der Leiter den Signalpfad bildet und die Kupferfläche als Referenzlage dient. Die Leiter sind durch ein Dielektrikum getrennt. Die Korrekturfaktoren werden gesetzt zu: $k_a = 1$ $k_p = 1$.
- Variante 3: Zwei übereinander liegende Leiter in differentieller Konfiguration, getrennt voneinander und den Rückleitern durch ein Dielektrikum. Es wird jeder Leiter als Signalpfad verwendet. Die Korrekturfaktoren werden gesetzt zu: $k_a = 2$ $k_p = 2$.

Abbildung 2.21 zeigt die frequenzabhängige Signaldämpfung der verschiedenen Kanalvarianten. Der geringere Widerstand einer Kupferfläche im Gegensatz zu einem Leiter mit einem kleinen Querschnitt wird besonders bei Variante 2 deutlich. Dieser Kanal weist insbesondere im unteren Frequenzbereich die geringste Dämpfung auf. Je höher die Frequenz wird, umso geringeren Einfluss hat die Kupferfläche auf die Signaldämpfung. Die Dämpfung wird bei hohen Frequenzen durch den Wechselstromwiderstand dominiert. Die differentielle Konfiguration von Variante 3 beschreibt den Kanal mit den höchsten Verlusten. Bei einer differentiellen Konfiguration bewirkt der Skineffekt eine zusätzliche Verformung des Stromflusses (siehe Kapitel 2.1.1), beschrieben durch den Korrekturfaktor k_p. Dies bewirkt eine zusätzliche Erhöhung des Wechselstromwiderstands und damit eine höhere Signaldämpfung.

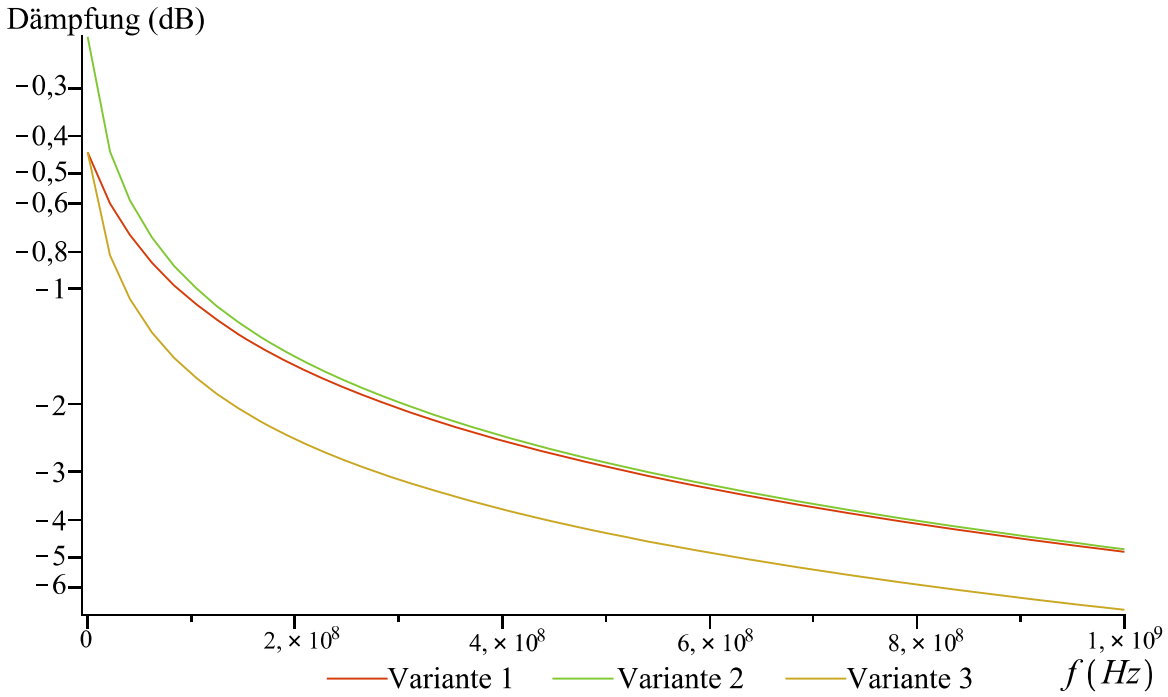

Abbildung 2.21: Frequenzabhängige Signaldämpfung in unterschiedlichen Übertragungskanälen.

Um ein Signal unverfälscht am Ende des Kanals zu detektieren, müsste der Sender die frequenzabhängige Amplitudendämpfung ausgleichen und damit bei einer Übertragung über die Kanalvariante 3 mehr Energie aufbringen als bei den anderen beiden Varianten. Den geringsten Energieaufwand erreicht man mit einem Kanal in Form von einem Signalleiter und einer Kupferfläche als Referenz, den höchsten Energieaufwand bedingt eine differentielle Übertragung aufgrund der zusätzlichen Verluste bei hohen Frequenzen. Zusätzlich werden für eine differentielle Übertragung immer zwei Leitungstreiber und Empfänger benötigt, was den Energiebedarf weiter erhöht. Um eine quantitative Aussage über den Anteil des Kanals an der Verlustleistung in einer Kommunikationsstruktur zu erhalten, müssen die Transceiver und Übertragungsstandards betrachtet werden, die den Kanal zur Signalübertragung verwenden. Verschiedene Übertragungsstandards werden in Kapitel 3 diskutiert.

2.1.3 S-Parameter-Beschreibung von Übertragungskanälen

Im vorherigen Kapitel wurden die wesentlichen Übertragungskanalparameter und Effekte beschrieben, die ein durch einen Übertragungskanal propagierendes Signal beeinflussen. Außerdem wurde über einen Vergleich des Modells mit Ergebnissen einer Simulation die Tauglichkeit des Modells zur Beschreibung des Kanals bestätigt. Die in Kapitel 3 verwendete Software zur Evaluierung von Übertragungsstandards bedingt die Beschreibung von Kanälen mittels S-Parametern. Zur Erstellung dieser Parameter wird das Programm *Si9000 PCB Transmission Line Field Solver* von Polar Instruments verwendet [R1]. Aus diesem Grund wird in diesem Kapitel das S-Parameter-Modell vorgestellt, das die Anwendung aller beschriebenen Parameter und Effekte auf einen Übertragungskanal ermöglicht und so sein Verhalten geschlossen beschreibt. Der Begriff S-Parameter stammt aus dem Bereich der Radartechnik. Hier wurden die Streueigenschaften eines Signals an einem Hindernis gemessen und durch Streuparameter beschrieben, S steht also für Streuung. Als die Datenraten und damit auch die Frequenzen in kupferbasierten Übertragungskanälen in den Bereich der Radartechnik vorstießen, konnte das Modell der S-Parameter auch auf die kupferbasierte Signalübertragung übertragen werden [R44]. Das S-Parameter Modell basiert auf der Betrachtung eines Übertragungskanals als „Black-Box“ mit ein oder mehreren Ein/Ausgängen (Toren). Ein S-Parameter beschreibt das Streuverhalten der Black-Box bezüglich eines Signals und seiner Tore. Dabei gibt der Parameter immer das Verhältnis eines Ausgangssignals zu einem Eingangssignal an.

$$S = \frac{Ausgangssignal}{Eingangssignal} \tag{2.45}$$

Der Parameter S besitzt zwei Komponenten: das Verhältnis der Signalamplituden und die Phasendifferenz der Signale. Das Amplitudenverhältnis ist gegeben durch:

$$mag(S) = \frac{Amplitude(Ausgangssignal)}{Amplitude(Eingangssignal)} \tag{2.46}$$

Die Amplitude wird normalerweise in dB angegeben, wobei:

$$S_{dB} = 20\log(mag(S)) \tag{2.47}$$

Die zweite Komponente eines S-Parameters ist die Phasendifferenz von Ausgangssignal und Eingangssignal.

$$Phase(S) = Phase(Ausgangssignal) - Phase(Eingangssignal) \tag{2.48}$$

Die S-Parameter werden mit Indizes versehen, die das entsprechende Tor des Systems kennzeichnen. Dabei gibt der erste Index das Tor an, an dem das Ausgangssignal gemessen wird, und der zweite Index entsprechend das Tor mit dem Eingangssignal. Der Parameter S_{21} bezeichnet also das Verhältnis von dem Ausgangsignal an Tor 2 zu dem Eingangssignal an Tor 1 (siehe Abbildung 2.22). Allgemein ausgedrückt ergibt sich:

$$S_{kj} = \frac{Signal_{aus,k}}{Signal_{in,j}} \tag{2.49}$$

Ein System mit n Toren kann durch eine Matrix mit n^2 Elementen beschrieben werden, also allen Kombinationsmöglichkeiten von Ein- und Ausgängen. Das Verhalten des Systems ist dann durch eine Matrixgleichung gegeben:

$$b_k = S_{kj} \times a_j \tag{2.50}$$

wobei b_k als Antwortvektor des Systems und a_j als Stimulusvektor zu verstehen ist.

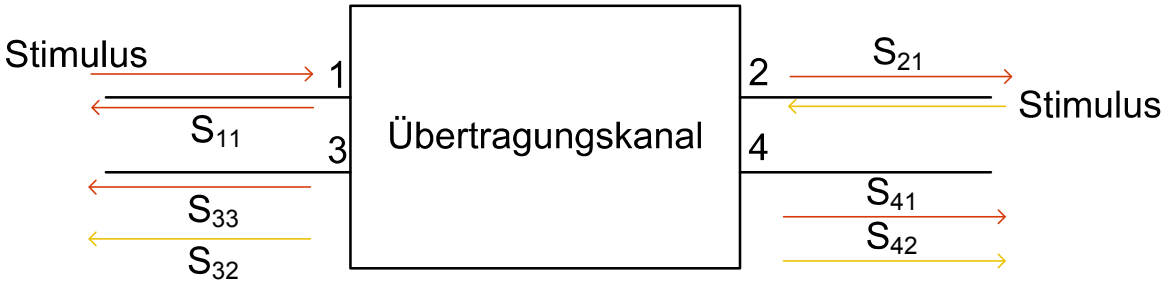

Abbildung 2.22: Ein Übertragungskanal als Black-Box mit vier Toren und zwei Stimulussignalen mit den zugehörigen S-Parametern.

2.2 Transceiver für kupferbasierte Übertragungskanäle

Damit ein Informationsaustausch zwischen zwei Kommunikationspartnern stattfinden kann, muss neben dem Kanal ein Sender und ein Empfänger zur Verfügung stehen. Für eine bidirektionale Kommunikation muss jeder Partner sowohl einen Sender als auch einen Empfänger beinhalten. Beide Funktionen werden oft zusammen in einer elektrischen Komponente implementiert und als Transceiver bezeichnet. Es handelt sich hierbei um ein Mischwort aus den Begriffen Transmitter und Receiver.
Unterschiedliche Transceiver für ähnliche Anwendungen beinhalten immer eine Auswahl an denselben Funktionseinheiten, sie unterscheiden sich je nach Technologie und Hersteller aber stark in der feingranularen technischen Realisierung. Hersteller von Transceiverbausteinen erlauben außerdem keinen Einblick in den genauen Aufbau ihrer Produkte. Nachfolgend werden deshalb die typischerweise in Transceivern implementierten Funktionseinheiten vorgestellt und erläutert.
Transceiver bestehen im einfachsten Fall aus einer halbleiterbasierten Treiberstufe auf der Senderseite und einem Detektor wie beispielsweise einem Differenzverstärker im Empfänger (siehe Abbildung 2.23). Es gibt verschiedenste technische Realisierungen dieser Einheiten, von denen die am häufigsten verwendeten Verfahren nachfolgend aufgezählt und in Kapitel 3 betrachtet werden.

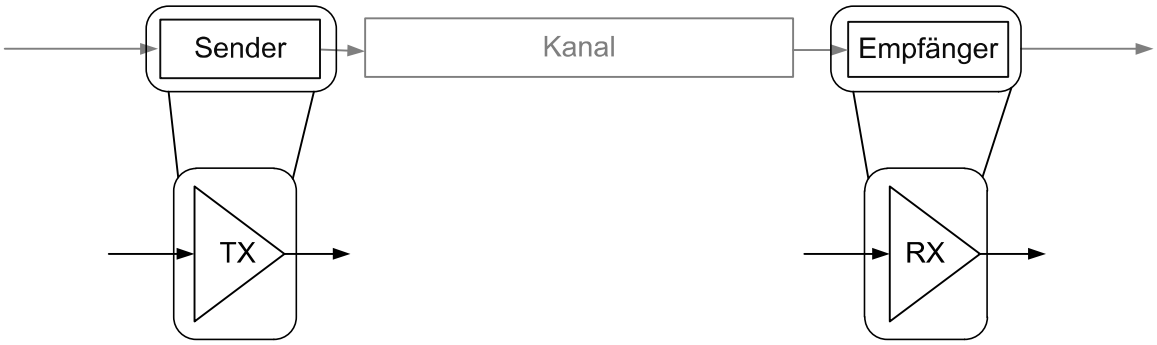

Abbildung 2.23: Sender und Empfänger bestehen im einfachsten Fall aus einer Treiber- und Detektorstufe.

- **LVTTL: L**ow **V**oltage **T**ransistor **T**ransistor **L**ogic. Dieses Verfahren wird bei älteren Übertragungsstandards, wie Peripheral Computer Interconnect (PCI), verwendet. Bei PCI handelt es sich um ein busbasiertes Verfahren, bei dem Sender und Empfänger den gleichen Kanal verwenden. Aus diesem Grund arbeiten solche Bussysteme im Halbduplexmodus.
- **GTL: G**unning **T**ransistor **L**ogic. GTL verwendet niedrigere Signalpegel als LVTTL und wird in moderneren Bussystemen, wie beispielsweise medienunabhängigen Schnittstellen oder dem Frontside Bus (FSB) angewandt. Während

der FSB aufgrund variabler Busteilnehmeranzahl unidirektional ausgelegt ist, werden medienunabhängige Schnittstellen als bidirektionale Verbindungen ausgelegt, da hier nur zwei Teilnehmer vorhanden sind.

- **CML: C**urrent **M**ode **L**ogic. Der Großteil der in dieser Arbeit betrachteten Verfahren und alle seriellen Hochgeschwindigkeitstransceiver der FPGAs verwenden CML Logik. Diese Verfahren werden oft in seriellen Hochgeschwindigkeitsanwendungen eingesetzt und sind als serielle Punkt-zu-Punkt Verbindungen realisiert.
- **Multipegel Logik:** Als Multipegel Logik werden die von manchen Ethernet-Standards (100BaseTX, 1000BaseT) verwendeten Kanalkodierungen bezeichnet. Hierbei wird ein Signal nicht binär auf einen festen Spannungspegel kodiert, vielmehr wird von mehr als zwei Referenzpegeln auf den Leitungen Gebrauch gemacht. Hierbei kann entweder ein Symbol durch mehr als einen Pegel beschrieben werden (100BaseTX) oder ein Pegel kodiert eine ganze Bitfolge (1000BaseT). Da Multipegel-Logik nicht auf den betrachteten FPGAs realisiert werden kann, kommen hierfür *SPICE*-Modelle kommerzieller Komponenten zum Einsatz.
- **LVDS: L**ow **V**oltage **D**ifferential **S**ignaling. Diese Übertragungstechnik kommt beispielsweise in den Rapid-Prototyping-Systemen der RAPTOR-Familie zum Einsatz, welche in Kapitel 5.1 betrachtet werden. Es handelt sich um eine differentielle Übertragung mit niedrigem Spannungspegel.

Die oben beschriebenen elektrischen Verfahren realisieren die Übertragung des elektrischen Signals über einen physikalischen Kanal. Moderne Übertragungsverfahren nutzen oft einen seriellen Datenstrom auf der Kanalseite, um beispielsweise die Anzahl von Kupferleitungen gering zu halten, während die Anbindung auf der anderen Seite parallel erfolgt. Das bedingt eine Erhöhung der Datenrate auf dem Kanal, welche proportional zur Eingangsbitbreite ist. Im Datenstrom muss zusätzlich eine Information über die Taktung der einzelnen Bits übertragen werden. Ein Transceiver beinhaltet zur Umsetzung der geforderten Funktionen zusätzliche Einheiten (siehe Abbildung 2.26). Nachfolgend sind diese im Falle des Transmitters angegeben.

- **8B/10B bzw. 64B/66B Kodierung:** In einem seriellen Bitstrom kann das Vorhandensein langer Bitsequenzen ohne Pegelwechsel einen negativen Einfluss auf die Taktrückgewinnung haben, oder es kann ein datenabhängiger Jitter auftreten, welcher durch differentielle Leitungskapazitäten auftritt. Aus diesem Grund werden dem Datenstrom zusätzliche Informationen hinzugefügt (8(64) Bit mit 10(66) Bit kodiert), um eine möglichst gleiche Anzahl von logischen Einsen und Nullen zu erreichen und so unerwünschte Effekte auszuschließen. Außerdem können diese Zusatzinformationen für eine Fehlererkennung genutzt werden.
- **TX Ring Puffer:** Dieser als Ringpuffer implementierte, elastische Puffer ist in der Lage unterschiedliche Lese- und Schreibraten anzugleichen. Dafür werden in

regelmäßigen Abständen IDLE-Daten vom System empfangen, welche der Puffer erkennt. Diese Daten werden dazu verwendet, den Puffer möglichst halb gefüllt zu lassen. Falls der Puffer schneller ausgelesen als beschrieben wird, so stellt der Puffer wiederholt die gleichen IDLE-Daten zum Lesen bereit (der Lesepointer bleibt auf dem IDLE-Feld stehen), um so den Puffer aufzufüllen. Bei einer höheren Schreibrate in den Puffer als Leserate aus dem Puffer können mehrere IDLE-Felder einfach übersprungen werden, um Platz im Puffer zu schaffen. Abbildung 2.24 und 2.25 illustrieren diese Vorgänge anschaulich.

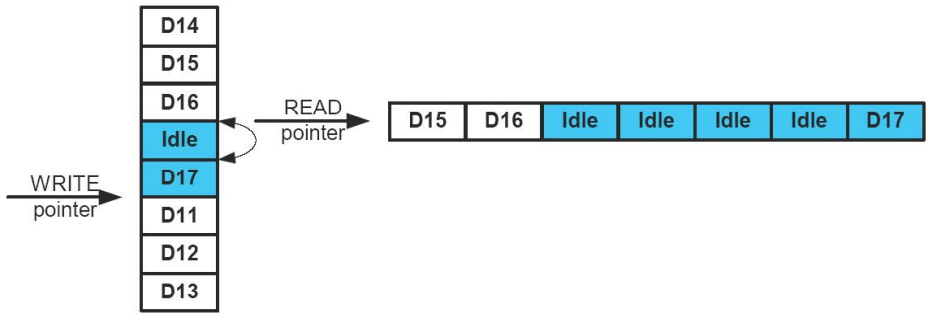

Abbildung 2.24: Ausgleichen des Pufferfüllstands bei höherer Lese- als Schreibrate [R73].

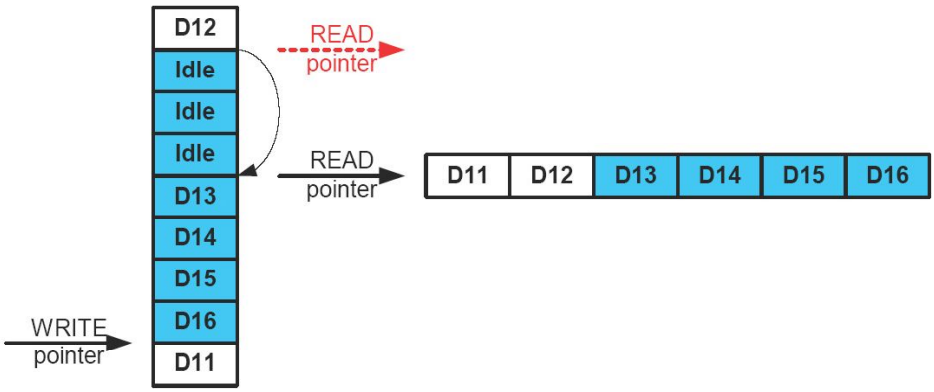

Abbildung 2.25: Ausgleichen des Pufferfüllstands bei höherer Schreib- als Leserate [R73].

- **Scrambler:** Zusätzlich oder alternativ zu einer 8B/10B-Kodierung der Daten kann ein Scrambler (Verwürfler) verwendet werden. Dieser ordnet die Daten in zufälliger Reihenfolge an, sodass ein möglichst gleichspannungsfreier Datenstrom entsteht. Erreicht wird dies durch die Nutzung von linear rückgekoppelten Schiebregistern oder Tabellen. Auf der Empfängerseite kann durch Anwendung desselben Algorithmus auf den Bitstrom der unveränderte Datenstrom zurückgewonnen werden.

- **Serialisierer:** Der Serialisierer bringt die parallel ankommenden Daten in eine serielle Reihenfolge und gibt diesen Datenstrom mit der geforderten Übertragungsrate auf die differentiellen Ausgangstreiber (vgl. Kapitel 2.2.3).

Die im Transmitter vorhandenen Funktionseinheiten müssen in entsprechender Form auch im Empfänger vorhanden sein.

- **Deserialisierer:** Dieser Block nimmt den seriellen und hochfrequenten Datenstrom der Eingänge entgegen und bringt ihn in eine parallele Form mit entsprechend geringerer Datenrate (vgl. Kapitel 2.2.3).
- **Kommata-Detektion und Anordnung:** Damit die einzelnen Bytes der parallelisierten Daten korrekt erkannt werden können, werden je nach Kodierung entsprechende Steuerinformationen in den Strom eingefügt (Kommata), welche die Bytes voneinander abgrenzen. Diese Einheit übernimmt die Detektion der Kommata und erzeugt aus dem Rohdatenstrom, der aus dem Deserialisierer kommt, ein paralleles Wort aus Bytes.
- **8B/10B bzw. 64B/66B Dekodierung:** Entsprechend der durchgeführten Kodierung im Transmitter erfolgt hier die Dekodierung der Daten.
- **RX Ring Puffer:** Dieser Puffer dient, wie der TX Ring Puffer, der Angleichung zwischen Lese- und Schreibrate.

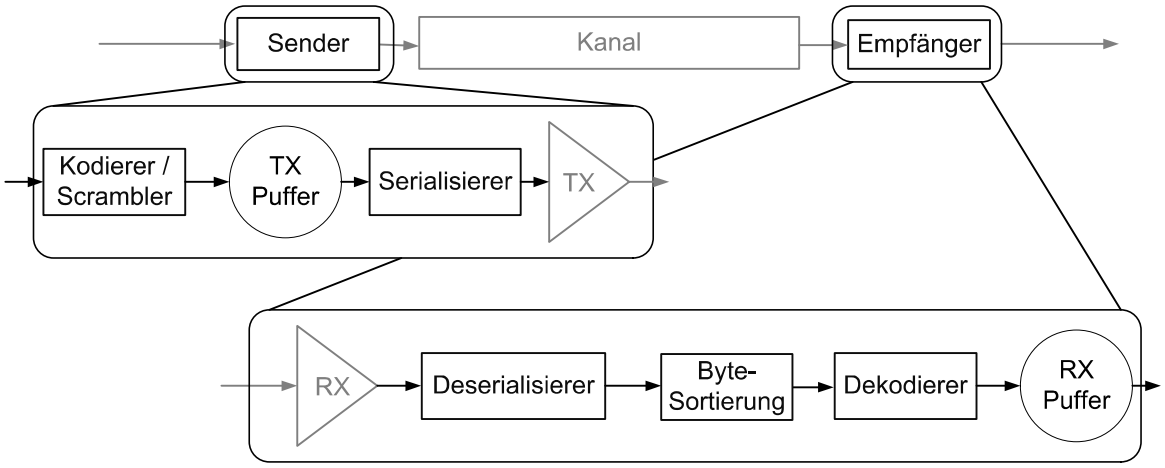

Abbildung 2.26: Funktionseinheiten eines modernen Transceivers.

2.2.1 Takterzeugung, Synchronisation und Fehlerkorrektur

Transceiver für serielle Datenübertragung müssen eine Information über die Taktrate im Bitstrom kodieren. In einem dafür vorgesehenen Modul werden die PLLs (Phase

Locked Loop) eingestellt, um entsprechend dem Takt der parallelen Daten einen seriellen Datenstrom mit einem höheren Takt erzeugen zu können. Dieser Takt ist ein ganzzahliges Vielfaches des Ursprungstaktes und muss exakt eingestellt werden. Über einstellbare Multiplikatoren und Divisoren wird aus einem Referenztakt ein zweiter erzeugt, der auf Phasengleichheit mit dem Referenztakt überprüft wird. Ist diese nicht vorhanden, so wird der Takt durch die PLL solange in der Phase verschoben bis beide phasengleich sind. Dieser erzeugte Takt wird nun für den seriellen Datenstrom verwendet. Entsprechend der Bitbreite des parallelen Datenstroms und der verwendeten Kodierung wird gleichzeitig ein weiterer Takt erzeugt, der die am Transceiver angeschlossene Logik treibt. Um im Empfänger einen Takt aus einem seriellen Datenstrom zu gewinnen, werden entweder die Flanken der einzelnen Datenbits erkannt oder spezielle Symbole im Datenstrom ausgewertet (vgl. Kapitel 2.2.3). Diese Symbole können außerdem dazu verwendet werden, parallele Datenströme an unterschiedlichen Empfängern zu synchronisieren (vgl. Abbildung 2.27 und Kapitel 3.3.3). Hierzu werden die Daten in den einzelnen Empfangspuffern so zueinander verschoben, dass die Synchronisationssymbole parallel zueinander stehen.

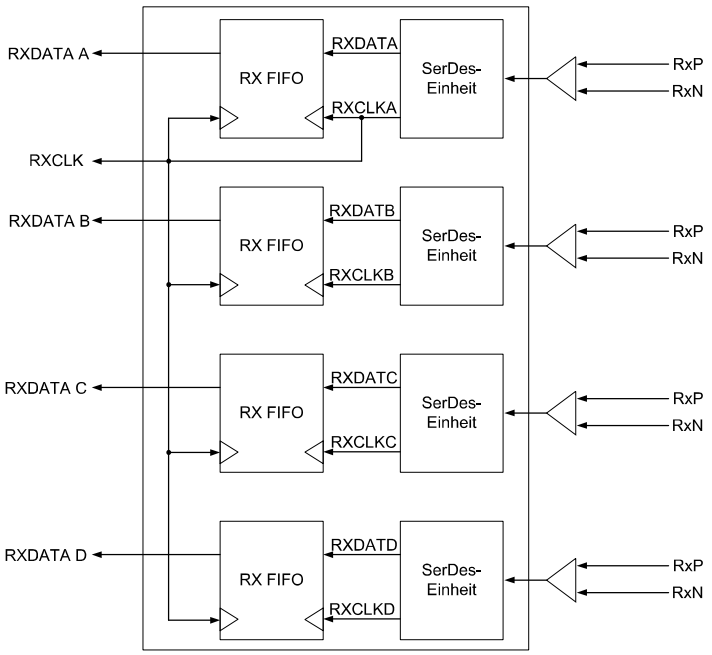

Abbildung 2.27: Synchronisation von vier Kanälen am Beispiel eines XAUI-Empfängers [R64].

Eine weitere Funktionseinheit des Transceivers ermöglicht die Erkennung und Korrektur von Fehlern im Empfangsdatenstrom. Mittels eines CRC-Verfahrens (Cyclic Redundancy Check) wird über eine mathematische Funktion aus den Datenpaketen eine Prüfsumme erstellt und an das Paket angehängt. Bei dem Empfang von Daten kann über eine erneute Anwendung der Funktion auf die Daten festgestellt werden,

ob diese korrekt übertragen wurden oder nicht. Abbildung 2.28 erweitert den Aufbau von Sender und Empfänger mit Funktionen zur Takterzeugung, Synchronisation und Fehlerkorrektur.

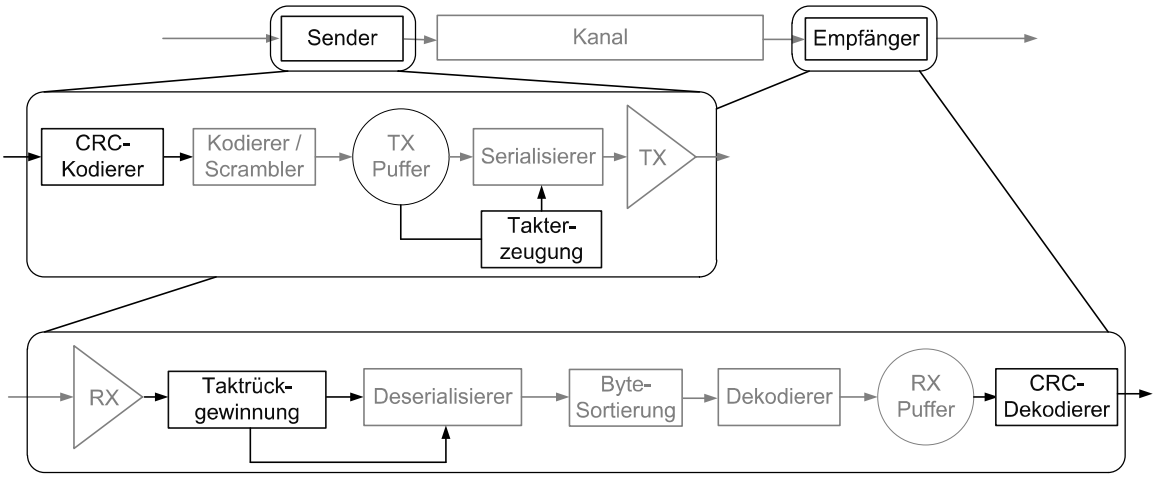

Abbildung 2.28: Funktionseinheiten eines modernen Transceivers zur Takterzeugung, Synchronisation und Fehlerkorrektur.

2.2.2 Signalformadaption

Wie in Kapitel 2.1.1 erläutert, werden Signale während der Ausbreitung durch den Übertragungskanal aufgrund dielektrischer Verluste und des Skin-Effekts verzerrt. Diese Verzerrung kann so stark sein, dass das resultierende Signal am Empfänger nicht mehr korrekt detektiert werden kann und dadurch Information verloren geht. Die frequenzabhängigen Verluste im Kanal verändern das Signal auf unterschiedliche Weise, was zu verschiedenen Kompensationsanforderungen führt.

- Skin-Effekt Verluste: Der Skin-Effekt verursacht bei hohen Frequenzen eine Verdrängung des Stromes in die Außenbereiche des Leiters, was zu einer frequenzabhängigen Erhöhung des Leiterwiderstands führt. Die Skin-Effekt-Verluste sind proportional zu der Quadratwurzel der Signalfrequenzanteile und resultieren in einem eher geringen Anstieg des Widerstands.
- Dielektrische Verluste: Wenn ein Signal durch einen Leiter propagiert, welcher durch ein Dielektrikum von einem anderen Leiter getrennt ist, so wird ein Teil des Signals vom Dielektrikum absorbiert. Diese Verluste sind direkt proportional zu den Signalfrequenzanteilen und ergeben einen steileren Anstieg des frequenzabhängigen Widerstands als bei dem Skin-Effekt.

Beide Verlustmechanismen sorgen für eine Verringerung der Flankensteilheit eines Signals. Diese geringere Flankensteilheit führt zu sogenannten Inter-Symbol-Interferenzen

(ISI), welche ein einzelnes Bit über mehrere Bitperioden dehnen. Die Skin-Verluste sind hierbei der dominierende Faktor bei Übertragungskanälen in Form von Kabeln, während dielektrische Verluste bei Kanälen auf Leiterplatten dominieren. Hieraus ergibt sich die Konsequenz, unterschiedliche Arten der Kompensation bei unterschiedlichen Kanälen zu verwenden. Um den Einfluss des Übertragungskanals auf die Signalform zu verdeutlichen, werden oft sogenannte Augendiagramme verwendet (siehe Abbildung 2.30). Hierfür wird ein zufälliger Bitstrom über die Transmitter geschickt und die entstehenden Signalverläufe dabei aufgezeichnet und überlagert. Dabei entsteht das charakteristische Augendiagramm, dessen vertikale Öffnungen die minimalen High- und Low-Pegel des Empfangssignals angeben. Je weiter das Auge geöffnet ist, desto sicherer können die Signalpegel erkannt werden. Die horizontale Augenöffnung gibt den Grad der Inter-Symbol-Interferenzen an. Es gibt zwei Strategien, um mit den genannten Arten von Verlusten umzugehen. Zum einen können bessere Übertragungsmedien mit geringen Verlusten verwendet werden, zum anderen können Maßnahmen zur Beeinflussung der Signalform angewendet werden. Während bei der Wahl der Übertragungsmedien oft enge Grenzen gesetzt sind, können durch die Verwendung von Signalformungstechniken wie Verzerrern und Entzerrern deutliche Steigerungen der Signalintegrität erreicht werden.

Vorverzerrung und Nachverzerrung

Techniken zur Vorverzerrung und Nachverzerrung werden auf das Problem des verlustbehafteten Kanals angewendet, indem sie eine frequenzabhängige Dämpfung auf die zu sendende Signalform aufprägen. Die Kanalverluste führen zur Abflachung der Signalflanken, was Inter-Symbol-Interferenzen hervorruft. Um diese Interferenzen zu kompensieren, erhöhen Vorverzerrer und Nachverzerrer die Amplitude der Signalflanken und damit der hochfrequenten Anteile im Vergleich zu den flachen Anteilen der Signalform, bzw. den niederfrequenten Anteilen. Die Verzerrer werden so eingestellt, dass sie das Signal in einer Form adaptieren, die der Inversen der Kanalimpulsantwort gleicht. So ergibt sich am Ende des Kanals ein flacher, kombinierter Frequenzverlauf aus Verzerrer und Kanalverlusten. Der Unterschied zwischen Vorverzerrer und Nachverzerrer liegt in der Art, wie die Frequenzkompensation erreicht wird. Ein Vorverzerrer erhöht die Energie der Signalflanken durch ein Überschwingen, während ein Nachverzerrer die flachen Bereiche der Signalform abschwächt, die Flanken jedoch nicht verändert. Entsprechend wird eine höhere Energie für die Vorzerrung als für die Nachverzerrung benötigt, dafür ergibt sich durch die höhere Energie eine größere Augenöffnung am Kanalende. Eine Vorverzerrung oder Nachverzerrung wird durch das Verhältnis der maximalen Spannungsamplitude des Signals zu der Signalamplitude im

flachen Bereich (siehe Abbildung 2.29) repräsentiert. Das Maß der Vorverzerrung (VV) und Nachverzerrung (NV) wird dabei in dB angegeben.

$$VV = 20 \cdot log_{10}(\frac{A}{B}), B = V_{Signal} \tag{2.51}$$

$$NV = 20 \cdot log_{10}(\frac{B}{A}), A = V_{Signal} \tag{2.52}$$

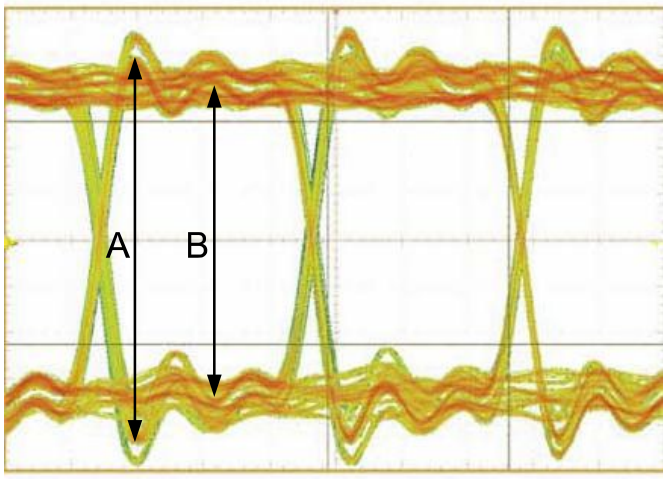

Abbildung 2.29: Verschiedene Anteile des Signals werden bei der Verzerrung adaptiert.

Entzerrung

Alternativ zur Pulsformung im Sender kann die Verzerrung durch den Übertragungskanal auch im Empfänger ausgeglichen werden. Da das Tiefpassverhalten des Kanals bekannt ist und berechnet werden kann, wird dieses Verhalten durch Anhebung der höheren Frequenzanteile des Signals im Empfänger ausgeglichen. Die Vorteile einer Entzerrung des Signals im Empfänger gegenüber einer aktiven Verzerrung im Sender liegen im geringeren Energiebedarf des Entzerrers. Es muss nur Energie für die Versorgung des frequenzabhängigen Verstärkers aufgewendet werden. Bei einer aktiven Verzerrung müssen die Kanalverluste durch erhöhte Flankenamplituden aktiv ausgeglichen werden, die Treiberstufe im Transmitter muss diese Energie zusätzlich auf den Kanal geben. Wenn ein Signal durch einen Kanal so stark verfälscht wird, dass eine Entzerrung allein nicht zur Rekonstruktion der Information genügt, so muss eine Vorverzerrung oder eine Kombination aus beiden Verfahren eingesetzt werden. Abbildung 2.30 zeigt den kompletten Aufbau eines Transmitters und Receivers, wie er in modernen Systemen realisiert wird.

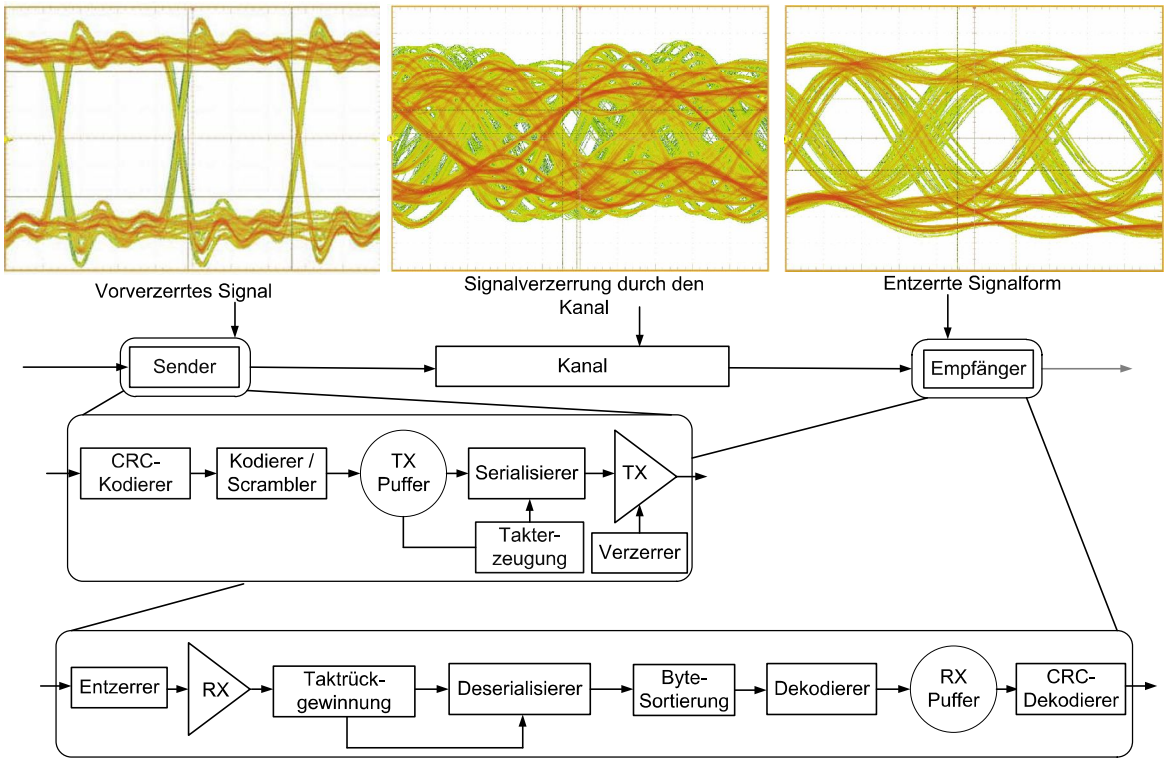

Abbildung 2.30: Der Aufbau von Transmitter und Receiver, erweitert durch Einheiten zur Signalformadaption

2.2.3 Serialisierung und Deserialisierung von Datenströmen

Viele der nachfolgend vorgestellten Übertragungsstandards und Verfahren, wie z. B. Infiniband, verwenden sogenannte SerDes-Techniken zur Übermittlung von Daten. Diese Technik soll hier zum besseren Verständnis erläutert werden. SerDes steht für Serialisierer/Deserialisierer und beschreibt zwei Funktionsblöcke, welche in Kommunikationsanwendungen zum Einsatz kommen, wenn Daten von parallelen Schnittstellen seriell übertragen werden sollen. Diese Blöcke konvertieren Daten zwischen einer parallelen Schicht und einer seriellen Schicht. Beispielsweise sei eine serielle Übertragungsstrecke mit einer Bandbreite von 10 Gbit/s gegeben, über die zwei Partner mit dieser Bandbreite kommunizieren sollen. Jeder der Partner hat dabei mehrere Übertragungskanäle zur Verfügung, um mit dem anderen Partner zu kommunizieren, jeder Kanal verfügt jedoch nur über eine Bandbreite von 1 Gbit/s. Wie in Abbildung 2.31 gezeigt, können nun jeweils zehn dieser Kanäle an den Serialisierer bzw. Deserialisierer angeschlossen werden, zwischen denen die serielle Übertragungsstrecke liegt. Der Serialisierer nimmt in diesem Fall die zehn parallelen Datenströme entgegen und kombiniert diese zu einem seriellen Datenstrom mit der zehnfachen Datenrate gegenüber der ursprünglichen. Der so übertragene Datenstrom wird vom Deserialisierer wieder in zehn parallele Datenströme mit entsprechend niedriger Datenrate aufgespalten. SerDes-Verfahren können in drei Arten aufgeteilt werden, jede Art wird für unterschiedliche Szenarien angewendet, kann jedoch auch mit einer anderen gekoppelt werden.

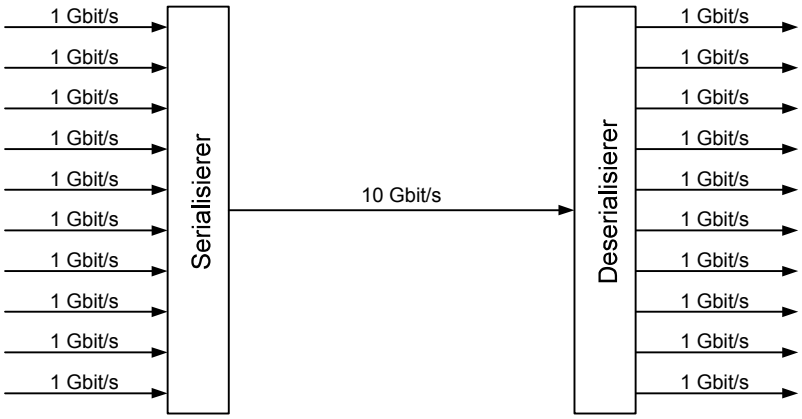

Abbildung 2.31: Grundlegendes Funktionsprinzip eines SerDes-Verfahrens zur Datenübertragung [R21].

SerDes mit parallelem Takt

Bei der Serialisierung von parallelen Bussen wie beispielsweise PCI kann es sehr aufwendig sein, alle Leitungen mit einem einzigen Multiplexer zu bündeln. Stattdessen ist es oft sinnvoller, den Bus mit Hilfe kleinerer Serialisierer in mehrere Datenströme zu konvertieren. Ein zusätzliches Taktsignal wird hierbei parallel zu den Datenströmen zum Empfänger geleitet, um die Daten korrekt wiederherzustellen (siehe Abbildung 2.32). Hierbei muss auf einen hinreichenden Längenausgleich der Leitungen geachtet werden, um Laufzeitunterschiede zu vermeiden.

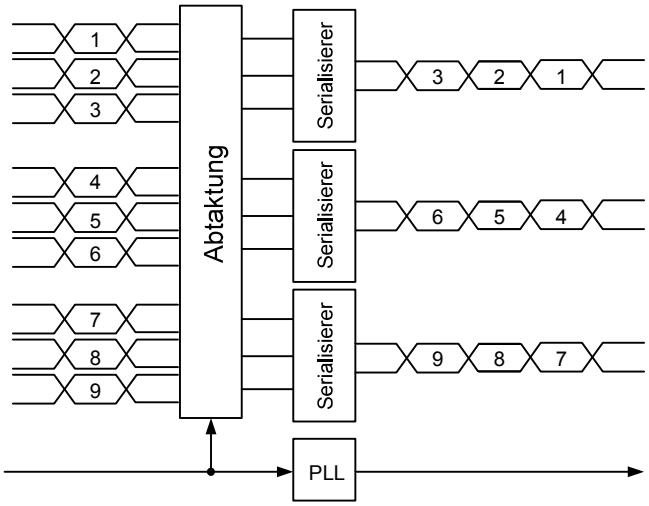

Abbildung 2.32: Beispiel einer SerDes-Übertragung mit parallelem Takt [R21].

SerDes mit eingebettetem Takt

SerDes-Transmitter mit eingebettetem Taktsignal serialisieren einen parallelen Datenstrom und das zugehörige Taktsignal in einen gemeinsamen Datenstrom. Nach jedem Zyklus wird hierbei eine steigende Taktflanke in den Datenstrom eingebettet. Dies ermöglicht dem Empfänger eine einfache Taktsynchronisation, ohne dass eine weitere Leitung zur Übermittlung dieser Synchronisation notwendig ist (siehe Abbildung 2.33).

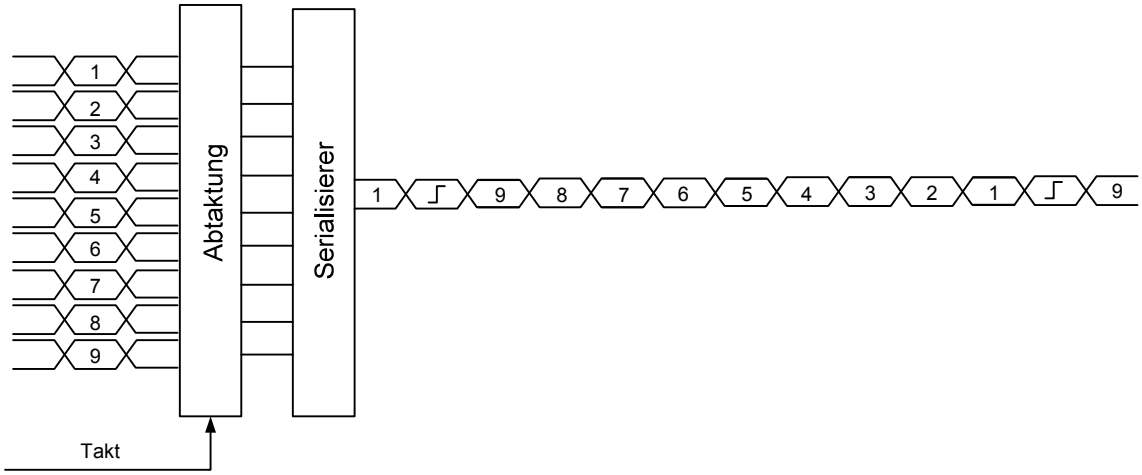

Abbildung 2.33: Beispiel einer SerDes-Übertragung mit eingebettetem Takt [R21].

8b/10b SerDes

Das 8b/10b-SerDes Verfahren bildet ein Byte auf ein zehn Bit breites Datenwort ab und serialisiert dieses anschließend (siehe Abbildung 2.34). Die Abbildung erfolgt nach einem festen Verfahren und garantiert dadurch das Vorkommen mehrerer Signalflanken pro Datenwort, um die Taktrückgewinnung zu erleichtern. Ebenso wird Gleichstromfreiheit auf der Leitung gewährleistet, also die gleiche Anzahl von logischen Nullen und Einsen. Damit der Empfänger den Beginn einer Sequenz detektieren kann, sendet der Transmitter einmal eine sogenannte Komma-Sequenz. Diese Sequenz ist einzigartig in der Bitfolge und kommt sonst nicht im Datenstrom vor. Der Empfänger kann sich dadurch auf den Sender synchronisieren.

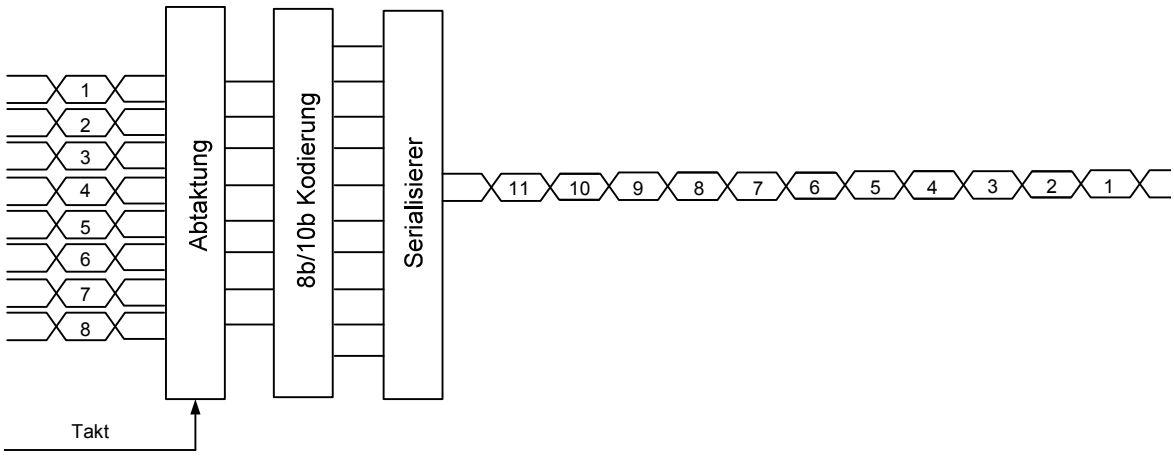

Abbildung 2.34: Beispiel einer SerDes-Übertragung mit 8b/10b Kodierung [R21].

2.3 FPGA-basierte Transceiver

Die vorhergehenden Betrachtungen zum Aufbau von Transceivern verdeutlichen die Komplexität dieser Komponenten. Offene Beschreibungen des elektrischen Aufbaus und des Verhaltens sind nicht verfügbar und unterliegen dem Herstellergeheimnis. Der FPGA-Hersteller Xilinx [R5] integriert seit einigen Produktgenerationen verschiedene Arten von Transceivern in seine FPGA-basierten Lösungen. Die Instanzen sind in weiten Bereichen konfigurierbar und können für die in dieser Arbeit betrachteten seriellen und parallelen Verfahren verwendet werden. Für diese Transceiver sind *SPICE*-Modelle verfügbar, welche das elektrische Verhalten aller Transceiverkomponenten exakt beschreiben und die Bestimmung der Leistungsaufnahme für verschiedene Szenarios ermöglichen. Hierdurch ergibt sich die Möglichkeit, das hergeleitete Modell für das Kanalverhalten mit Messungen des Transceivervehaltens zu kombinieren und so eine Evaluierung der kompletten Übertragungsstrecke durchzuführen. Die Modelle erlauben keinen genauen Einblick in die physikalische Struktur der Transceiver, sondern sind verschlüsselt als sogenannte „Black-Boxes“ implementiert. Der Transceiver kann jedoch funktional durch die Parametrisierung der einzelnen Transceiverkomponenten genau auf die jeweilige Anwendung abgestimmt werden. Größen wie die Leistungsaufnahme können wiederum nur global für den gesamten Transmitter und Receiver über die Spannungsversorgung der „Black-Box“ bestimmt werden, da kein Zugriff auf die Versorgungsleitungen einzelner Transceiverkomponenten möglich ist. Zur Effizienzevaluierung (siehe Kapitel 3) wird die Bestimmung der Leistungsaufnahme auf Transmitter- und Receiverebene durchgeführt.
In dieser Arbeit werden die Betrachtungen auf Basis von Xilinx-FPGAs durchgeführt, deren integrierte Transceiver alle erforderlichen Funktionseinheiten beinhalten und sich deshalb gut für die Evaluierungen eignen. Diese Einheiten sind als Hardware im FPGA

implementiert und können in den Benutzerentwurf eingebunden werden. Der globale Aufbau eines solchen Hardwarekerns ist in Abbildung 2.35 am Beispiel eines Virtex-4-Multi-Gigabit-Transceivers zu sehen. Busbasierte Verfahren werden mit Modellen von Standardtreibern evaluiert. Hierbei werden Modelle normaler Ein- und Ausgabeanschlüsse (Pins) von FPGAs verwendet. Diese sind in der Lage, alle geforderten elektrischen Standards wie LVTTL oder GTL umzusetzen.

In Abbildung 2.35 finden sich alle in Kapitel 2.2 beschriebenen Elemente wieder. Der für das Senden von Daten verantwortliche Teil der Transceiver ist im mittleren Drittel zu finden. Er beherrscht im Falle eines Virtex-4-MGTs die serielle Übertragung von Daten mit Raten von 622 MBit/s bis zu 6,25 GBit/s, wobei die Signalübertragung differentiell erfolgt. Alle Elemente für einen korrekten Empfang von Daten befinden sich im oberen Drittel der Abbildung.

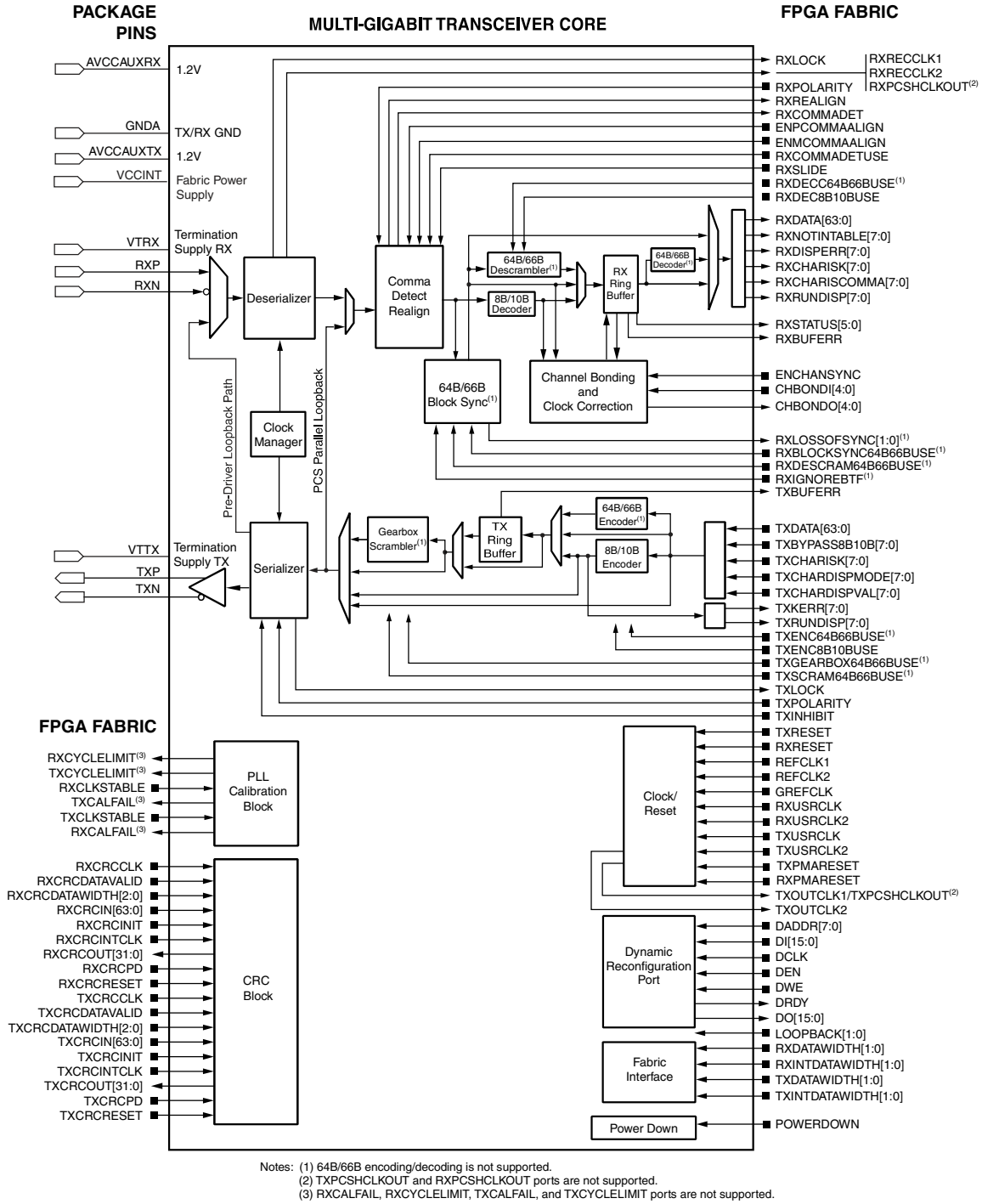

Abbildung 2.35: Der globale Aufbau eines RocketIO-Transceivers [R68].

Zum Entstehungszeitpunkt dieser Arbeit sind Modelle der Virtex-4, Virtex-5, Virtex-6 und Spartan-6 Reihen von Xilinx erhältlich. Entsprechend werden neben Aus- und Eingängen für parallele Übertragungsverfahren folgende serielle Hochgeschwindigkeitstransceiver betrachtet:

- **Virtex-4 Multi-Gigabit Transceiver:** Diese auch als RocketIO bekannten Transceiver sind in einer 90 nm Technologie implementiert. Je nach FPGA-Typ sind zwischen acht und 24 Transceiver verfügbar, welche jeweils eine Datenrate zwischen 622 Mbit/s und 6,5 Gbit/s unterstützen. Im FPGA sind immer zwei Transceiverinstanzen zu einem sogenannten Tile zusammengefasst. Hierbei teilen sich die beiden Instanzen eine gemeinsame Sende-PLL und eine gemeinsame Referenztakterzeugung.[R68]

- **Virtex-5 GTP Transceiver:** GTP-Transceiver sind auf niedrige Leistungsaufnahme optimierte serielle Transceiver mit einer Übertragungsrate zwischen 100 Mbit/s und 3,75 Gbit/s auf Basis einer 65 nm-Technologie. Je nach FPGA-Typ sind zwischen vier und 24 Transceiver verfügbar. Wie bei dem Virtex-4 sind jeweils zwei Transmitter zu einem Tile verschaltet, welches unter anderem eine gemeinsame PLL zur Verfügung stellt [R69].

- **Virtex-5 GTX Transceiver:** Zwischen acht und 48 GTX-Transceiver werden je nach FPGA-Typ bereitgestellt. Jeder Transceiver kann mit einer Übertragungsrate zwischen 150 Mbit/s und 6,5 Gbit/s arbeiten und basiert auf einer 65 nm-Technologie. Die Zusammenfassung von zwei Instanzen zu einem Tile erfolgt wie bei den GTP-Transceivern der Virtex-5 FPGAs [R70].

- **Virtex-6 GTX Transceiver:** GTX-Transceiver auf Virtex-6 Basis arbeiten mit einer Übertragungsrate zwischen 480 Mbit/s und 6,6 Gbit/s und werden in einer 40 nm-Technologie implementiert. Je nach verwendetem FPGA-Typ sind zwischen zwölf und 48 Instanzen verfügbar. Bei Virtex-6 FPGAs werden jeweils vier Transceiver zu einer sogenannten Quad-Konfiguration zusammengefasst, welche einen gemeinsamen Referenztakt zur Verfügung stellt [R72].

- **Virtex-6 GTH Transceiver:** Diese Transceivertypen sind auf höchste Übertragungsleistung hin optimiert und unterstützten Datenraten zwischen 1,24 Gbit/s und 11,182 Gbit/s. Wie bei den ebenfalls in 40 nm-Technologie implementierten Virtex-6-GTX-Transceivern werden vier GTH-Transceiver zusammengefasst. Eine gemeinsame PLL dient zur Takterzeugung. FPGAs mit diesen Transceivern stellen 24 Instanzen zur Verfügung [R71].

- **Spartan-6 GTP Transceiver:** Die auf einer 45 nm-Technologie basierenden seriellen GTP-Transceiver der Spartan-6-Reihe unterstützen Datenraten zwischen 614 Mbit/s und 3,125 Gbit/s. Je nach FPGA-Typ sind zwischen zwei und acht

Transceiver verfügbar, welche in Zweiergruppen zu einem Tile zusammengefasst sind und auf eine gemeinsame PLL zugreifen [R62].

3 Energieevaluierung von kupferbasierten Übertragungsverfahren

In diesem Kapitel werden weit verbreitete und auf rekonfigurierbaren Architekturen implementierbare, kupferbasierte Übertragungsstandards vorgestellt und miteinander verglichen. Die verschiedenen Übertragungsstandards werden in diesem Kapitel bezüglich der verwendeten Transceiver gruppiert, da sich die Evaluierung hier nur im Kanal und in bestimmten Parametern unterscheidet. Um Vergleiche zwischen den betrachteten Verfahren bezüglich ihres Energiebedarfs zu ermöglichen, müssen entsprechende Maße definiert werden. Ein solches Maß stellt die Leistungsaufnahme einer Implementierung dar. Diese trifft eine Aussage über die Leistungsaufnahme einer kompletten Kommunikationsstrecke inklusive aller Sender, Kanäle und Empfänger, welche für den Standard benötigt werden. Die allgemeine Formel zur Bestimmung der elektrischen Leistung ist gegeben durch:

$$P = \frac{1}{T} \int_{t=0}^{T} u(t) \cdot i(t) dt \tag{3.1}$$

Die Übertragungsstandards unterscheiden sich neben den elektrischen Eigenschaften vor allem in der Datenrate. Eine Aussage über den Energiebedarf eines bestimmten Übertragungsverfahrens kann deswegen nicht allein über die elektrische Leistungsaufnahme geschehen, da dieses Maß nicht die Datenrate und damit auch nicht die Bitzeit berücksichtigt. Einen qualitativen und von der Datenrate unabhängigen Vergleich kann nur die benötigte Energie pro Bitzeit liefern. Die allgemeine Formel zur Bestimmung der elektrischen Energie ist gegeben durch:

$$E = \int_{t_0}^{t_1} u(t) \cdot i(t) dt \tag{3.2}$$

Um eine exakte Aussage über die benötigte Energie treffen zu können, werden alle relevanten elektrischen Parameter benötigt. Aus diesem Grund wird die Kommunikationstrecke mit Hilfe einer *SPICE*-Simulation ausgewertet. *SPICE* (Simulation Program with Integrated Circuit Emphasis [R49]) ist ein Simulator für analoge, elektri-

sche Schaltungen. Über Modelle können alle Komponenten einer Übertragungsstrecke in ihrem elektrischen Verhalten simuliert werden. So erhält man die Möglichkeit, den Einfluss aller Komponenten und ihr Zusammenspiel zu evaluieren. Um die Übertragungsverfahren mittels *SPICE* zu simulieren, werden folgende Komponenten für die Übertragungsstrecke betrachtet:

- Transceiver: Hierbei handelt es sich um den Sender und den Empfänger in einer Übertragungsstrecke. Da sich diese in jedem Produkt unterscheiden und keine Herstellerinformationen über den exakten Aufbau verfügbar sind, wird auf verschlüsselte Modelle zurückgegriffen. Diese sind für viele Xilinx-FPGAs verfügbar und beschreiben das elektrische Verhalten.
- Übertragungskanal: Der Kanal repräsentiert die Übertragungsstrecke zwischen den Transceivern, hierzu gehören das Gehäuse des Transceivers, Stecker, Leiterbahnen und Kabel sowie Terminierungen. Der Kanal wird aus verschiedenen S-Parametermodellen gebildet, welche auf Lumped-Element-Modellen (vgl. Kapitel 2) basieren. Zur Erstellung wird der *Si9000 PCB Transmission Line Field Solver* von Polar Instruments verwendet [R1].

Um einheitliche Randbedingungen für die Simulation zu erhalten, wird für alle Komponenten eine Temperatur von 25 °C festgelegt. Parameter wie der Spannungshub der Treiberstufen oder die Vorverzerrung und Entzerrung werden so eingestellt, dass sie dem entsprechenden Standard entsprechen und die Signalintegrität am Empfänger sicherstellen. Als Datenstrom dient eine PRBS-Folge (Pseudo Random Bit Sequenze, pseudozufällige Bitfolge). Hierdurch ist sichergestellt, dass jede Kombination von logischen Einsen und Nullen in einem Symbol übertragen wird. Abbildung 3.1 zeigt den generellen Aufbau einer solchen *SPICE*-Netzliste. Während der Simulation, deren Dauer der Übertragungszeit der PRBS-Sequenz bei der gegebenen Übertragungsrate entspricht, werden alle relevanten Daten aufgezeichnet. Hierzu gehören die Spannungsversorgungen der Modelle sowie die zeitliche Stromaufnahme. Die Stromaufnahme wird bestimmt, indem sehr kleine (1 mΩ) Widerstände mit idealen Eigenschaften in Reihe zwischen die Versorgungsspannung und den entsprechenden Anschluss des Modells geschaltet werden (vgl. Abbildung 3.2). Durch den sich ergebenden Spannungsabfall an den Widerständen kann zu jedem Zeitpunkt die Stromaufnahme bestimmt werden. Über ein Integral des gemessenen Stromes und der jeweiligen Versorgungsspannung über die Zeit kann die Verlustleistung ermittelt werden. Abbildung 3.3 zeigt einen zeitlichen Ausschnitt einer *HSPICE*-Simulation für Infiniband. Die oberen drei Graphen geben die Signalform am Eingang des Transmitters sowie am Anfang und Ende des Kanals an. Die unteren vier Graphen zeichnen die Stromaufnahme der Terminierung und der Transceiver-Versorgung auf.

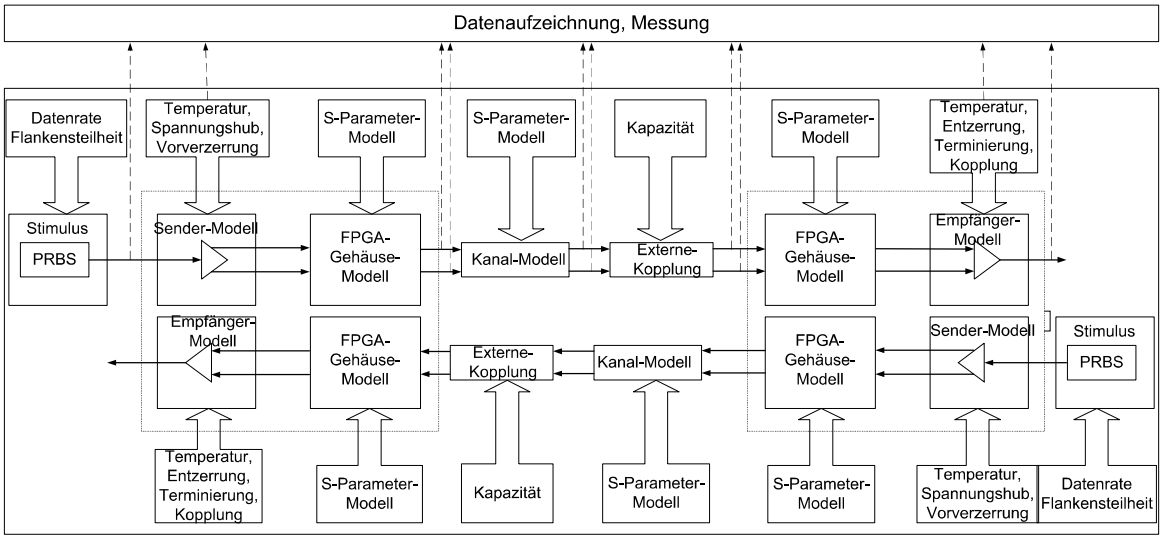

Abbildung 3.1: Blockschaltbild einer *SPICE*-Simulation für eine XAUI-Übertragungstrecke.

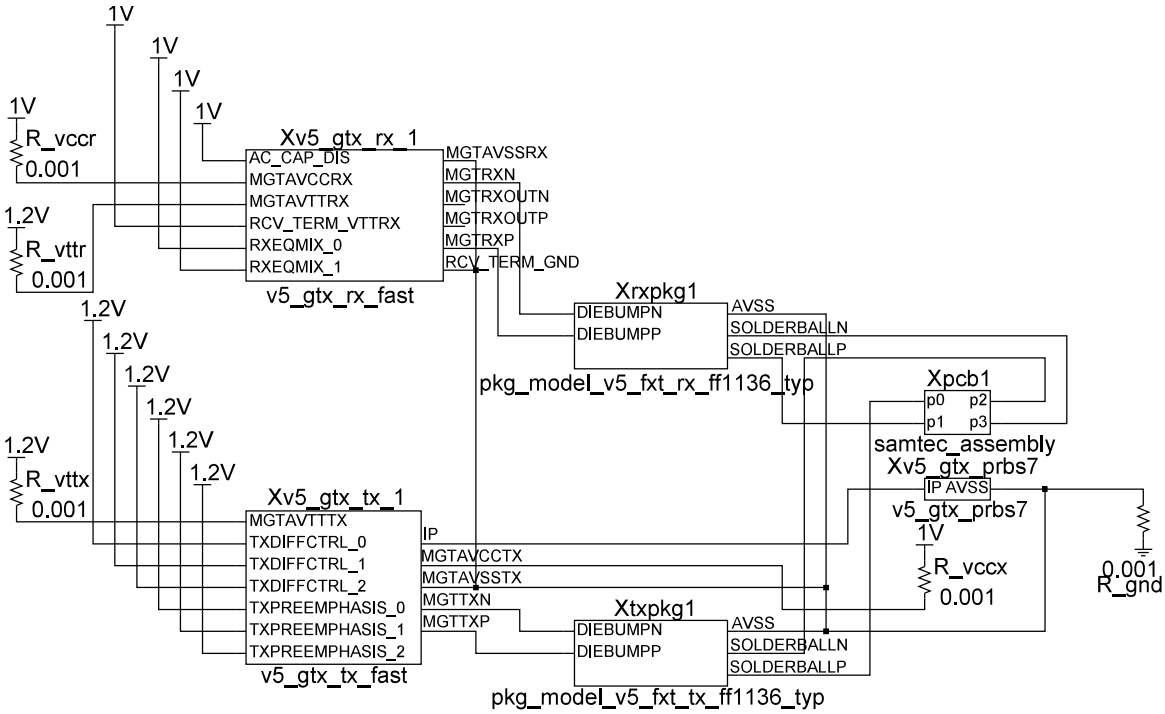

Abbildung 3.2: Graphische Darstellung der Netzliste einer Aurora-Übertragungsstrecke in *SpiceVision* [R9].

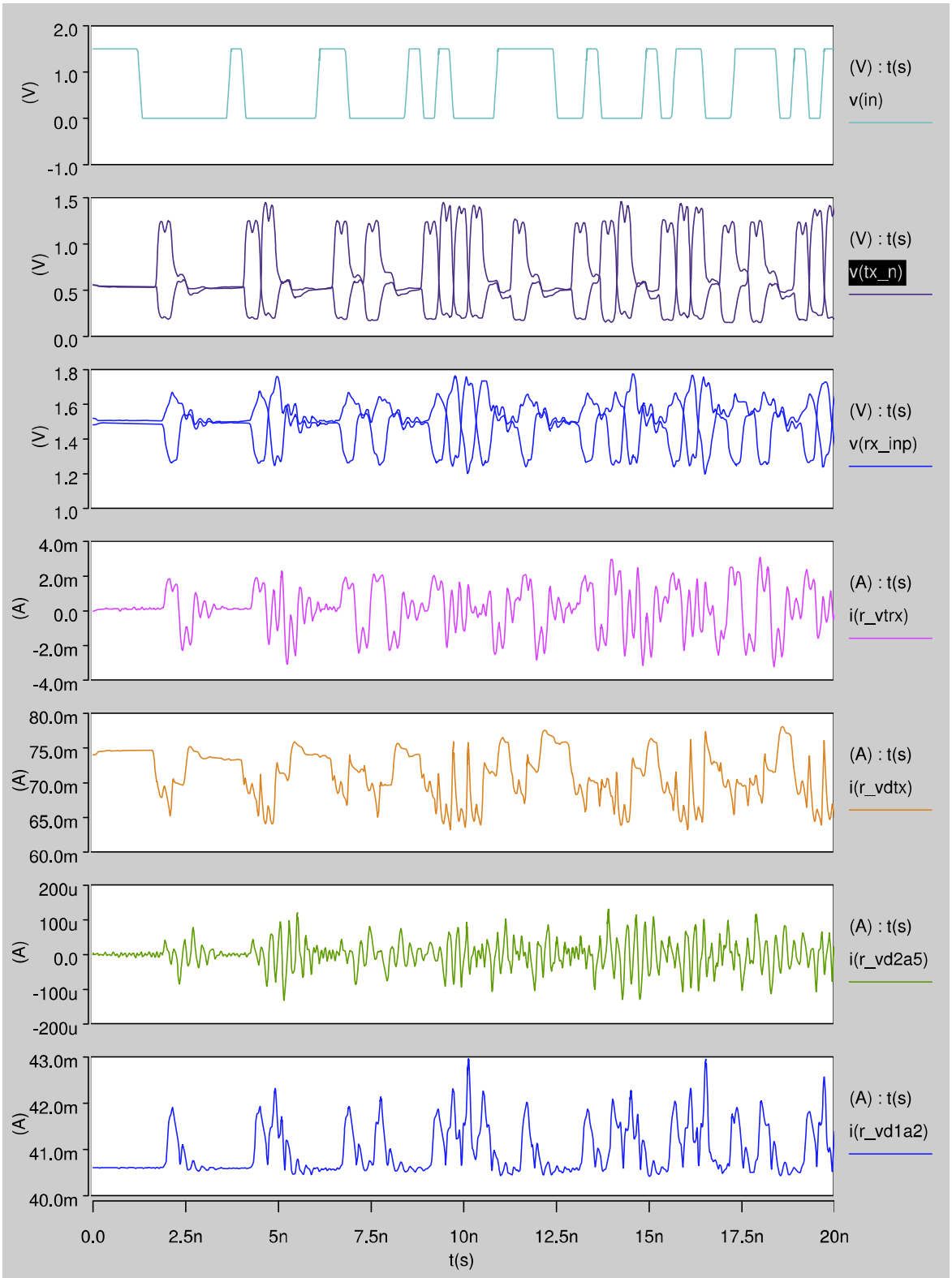

Abbildung 3.3: Aufzeichnung des Datensignals und der Stromaufnahme der FPGA-Komponenten an verschiedenen Stellen in der *HSPICE*-Netzliste.

Die verwendeten Transceivermodelle beinhalten nur die Treiber- und Empfängerstufen. Wie in Kapitel 2 gezeigt, besteht ein serieller Hochgeschwindigkeitstransceiver eines FPGAs jedoch aus weiteren Funktionseinheiten wie PLLs oder Kodierern. Die Bestimmung der Leistungsaufnahme dieser Einheiten ist über eine *SPICE*-Simulation aufgrund der eingeschränkten Modelle nicht einzeln möglich. Aus diesem Grund wird hierfür der *Xilinx Power Analyzer* [R4] verwendet. Dieser ermöglicht das genaue Einstellen aller Transceiverkomponenten nach den Vorgaben des entsprechenden Übertragungsstandards und gibt Angaben über die Leistungsaufnahme der Transceiver bezüglich der zugehörigen Versorgungsspannungen. Da die relevanten Komponenten wie die PLL oder die Kodierer über eine eigene Versorgungsspannung verfügen, die getrennt von den Treiberstufen und der Terminierungsversorgung eingespeist wird, kann so der Einfluss der Komponenten angegeben werden. Um belastbare Ergebnisse zu erzielen, müssen die Transceiver in einen VHDL-Entwurf eingebunden und durch eine Post-Place-and-Route-Simulation ein Aktivitätsmodell erstellt werden. Dieses wird in Verbindung mit dem Entwurf durch den *Xilinx Power Analyzer* in Werte für die Verlustleistung umgesetzt. Alle Transceiver werden wie bei der *SPICE*-Simulation mit einer PRBS7-Folge angesteuert. Eine Fusion der sich ergebenden Daten ermöglicht die Bestimmung der durchschnittlichen Leistungsaufnahme eines beliebigen Übertragungsverfahrens. Sie ergibt sich aus der Summe der Leistungsaufnahme aller Versorgungsleitungen.

$$P_{PRBS} = \frac{1}{T_{PRBS}} \sum_{r=0}^{N} \int_{t=t_0}^{t_0+T_{PRBS}} u_r(t) \cdot i_r(t) dt \tag{3.3}$$

Hierbei ist

- N: Gesamtanzahl von Versorgungsspannungen.
- r: Index der Versorgungspannung r.
- u_r: Spannungswert der Versorgungspannung r.
- i_r: Stromaufnahme über die Versorgungspannung r.
- t_0: Startzeit der Datenübertragung.
- T_{PRBS}: Dauer der Datenübertragung.

Die durchschnittlich benötigte Energie pro übertragenem Bit E_{Bit} kann aus P_{PRBS} abgeleitet werden.

$$E_{Bit} = P_{PRBS} \cdot \frac{T_{PRBS}}{N_{Bit} \cdot N} \tag{3.4}$$

$$= P_{PRBS} \cdot \frac{T_{Bit}}{N} \tag{3.5}$$

mit

- $N_{Bit} = 128$: Anzahl der Übertragungssymbole in der PRBS-Sequenz.
- $T_{Bit} = \frac{1}{Datenrate}$: Bitzeit.
- N: Anzahl parallel übertragener Bits.

Aufgrund der Darstellung der benötigten Energie pro Bit als Produkt aus durchschnittlicher Leistungsaufnahme und Bitzeit, wird in dieser Arbeit der Term E_{Bit} als PDP (Power Delay Product) bezeichnet. Dieser Term wird wiederum in drei Unterterme aufgeteilt, sodass mit P_{PRBS} insgesamt vier Bewertungsmaße zur Verfügung stehen.

P_{PRBS}: Die durchschnittliche Leistungsaufnahme einer Implementierung bei Übertragung einer PRBS-Folge.

$$PDP = E_{Bit} \tag{3.6}$$

PDP beschreibt die benötigte Energie pro übertragenem Bit ohne Berücksichtigung des Paketformats oder Leitungscodes.

$$PDP_{code} = PDP \cdot X_P \cdot X_K \tag{3.7}$$

PDP_{code} beschreibt die benötigte Energie pro übertragenem Bit unter Berücksichtigung des Paketformats X_P und eventuell vorhandener Leitungscodes X_K. Da die betrachteten Übertragungsverfahren paketbasiert arbeiten, fällt immer ein gewisser Mehraufwand an Steuerinformationen an. Dieser Mehraufwand erhöht die benötigte Energie pro übertragenem Nutzdatenbit, da er Bandbreite beansprucht, welche nicht für Nutzdaten verwendet werden kann. Ebenso verhält es sich mit implementierten Leitungscodes wie einer 8B/10B-Kodierung.

$$PDP_{real} = PDP_{code} \cdot X_R \tag{3.8}$$

Während PDP_{code} die theoretisch minimal benötigte Energie pro Bit eines Übertragungsverfahrens beschreibt, gibt PDP_{real} die benötigte Energie pro Bit unter realen Bedingungen an. PDP_{code} setzt beispielsweise voraus, dass jedes Paket immer maximal mit Nutzdaten besetzt ist und alle Pakete gleich ausgelastet sind. In der Realität ist dies nicht der Fall, weshalb der theoretische, maximale Durchsatz nie erreicht wird.

Dieser Einfluss, welcher die zu erwartende und benötigte Energie pro Nutzdatenbit angibt, wird durch den Faktor X_R beschrieben. Es gilt hierbei immer:

$$PDP \leq PDP_{code} \leq PDP_{real} \tag{3.9}$$

Im Folgenden werden die betrachteten Übertragungsverfahren beschrieben und bezüglich der vorgestellten Evaluierungsmethodik untersucht.

3.1 LVTTL - Transistor-Transistor-Logik mit verringerter Spannung

LVTTL basiert auf der mittlerweile veralteten Transistor-Transistor-Logik, welche seit den sechziger Jahren des zwanzigsten Jahrhunderts verwendet wird. LVTTL verwendet ursprünglich Bipolartransistoren als Treiberstufen und Empfänger. Hierbei wird der Treiber als Push-Pull ausgeführt, d. h. der Ausgang des Treibers wird jeweils aktiv auf Masse oder eine Referenzspannung gezogen. Während TTL eine Referenzspannung von 5 V aufweist, verwendet LVTTL eine Referenzspannung von 3, 3 V. Bei den betrachteten FPGA-Typen ist dieser Aufbau mit Bipolartransistoren nicht mehr vorhanden (vgl. Abbildung 3.4), vielmehr werden lediglich die geforderten Spannungspegel realisiert, um eine Kompatibilität sicher zu stellen. LVTTL-kompatible Geräte müssen bei einer logischen Null am Ausgang eine maximale Spannung von 0, 4 V aufweisen, bei einer logischen Eins müssen minimal 2, 4 V erreicht werden. Der Kanal einer LVTTL-Übertragungsstrecke kann mit einem Terminierungswiderstand ergänzt werden, um Reflexionen auf der Leitung zu vermeiden [R65].

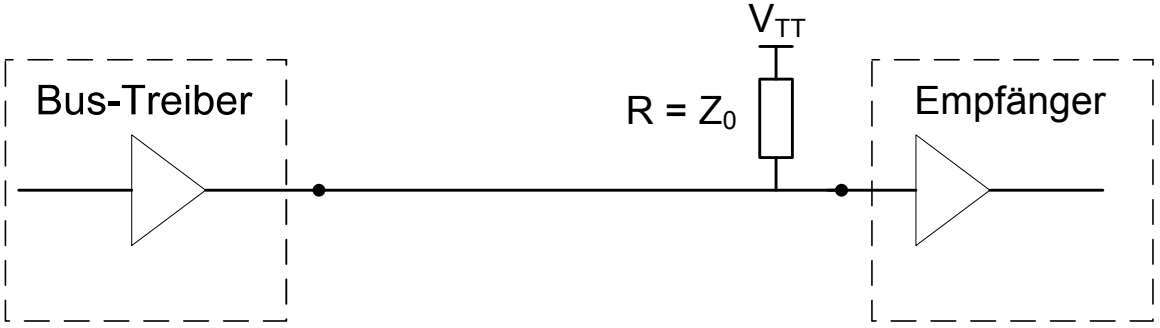

Abbildung 3.4: Vereinfachter Aufbau eines auf TTL basierenden Bussystems.

3.1.1 PCI - Peripheral Component Interconnect

PCI ist ein paralleles Bussystem innerhalb von Computern, das zunehmend von PCIe (vgl. 3.3.5) verdrängt wird. Es wird verwendet, um den Chipsatz an Peripheriegeräte wie Audio- oder Netzwerkadapter anzubinden. Es gibt PCI-Varianten für den TTL- und LVTTL-Standard. Da die betrachteten FPGAs nur LVTTL unterstützen, wird nur dieser Standard evaluiert. Die Entwicklung von PCI begann 1990, jedoch dauerte es bis 1994, um eine signifikante Marktpräsenz zu erreichen [R61]. Die erste PCI-Version verwendet einen Bus mit 32 Bit Breite und einer Taktfrequenz von 33 MHz. Spätere Varianten (PCI-X) nutzen eine Bitbreite von 64 Bit und eine Taktfrequenz von 66 MHz oder 133 MHz. Es ergeben sich so theoretische Übertragungsraten von 133 MByte/s, 533 MByte/s und 1066 MByte/s. Der Bus wird zur Übermittlung von Adressen und Daten verwendet, der Übertragungstyp wird hierbei durch zusätzliche Steuersignale festgelegt. Die Funktion dieser Signale erschließt sich aus Abbildung 3.5, welche eine Lesetransaktion zeigt.

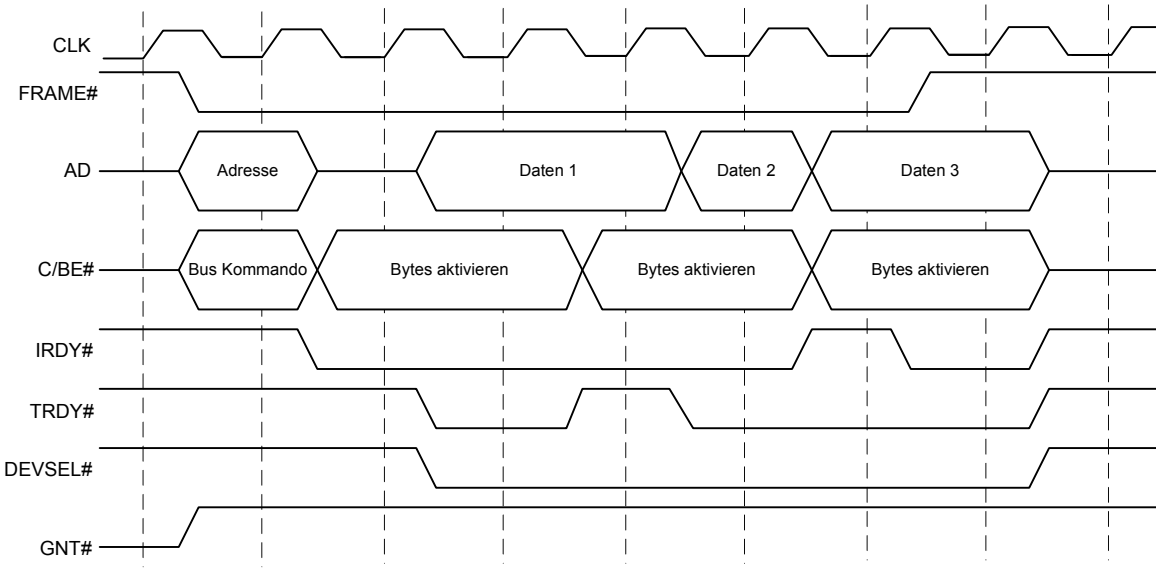

Abbildung 3.5: Eine Lesetransaktion auf einem PCI-Bus [R61].

Die Transaktion beginnt über den Initiator mit dem Setzen von FRAME#, der Zieladresse und dem Buskommando (Lesen) auf den Bus, das Ziel empfängt und dekodiert das Kommando. Anschließend maskiert der Initiator die Bytes über C/BE# und gibt über IRDY# seine Empfangsbereitschaft zu erkennen. Das Ziel registriert dies und setzt seine Sendebereitschaft (TRDY#) zusammen mit dem DVSEL#-Signal. Nun werden die angeforderten Daten vom Ziel auf den Bus geschrieben und vom Initiator gelesen. Die Transaktionen können von beiden Teilnehmern über TRDY# und IRDY# pausiert werden. Nach Abschluss der Lesetransaktion werden alle Statussignale wieder

in den Ausgangszustand zurückversetzt und der Bus freigegeben.
Für die Simulation einer PCI-Übertragungsstrecke werden die LVTTL-kompatiblen Pins der Virtex-4 und Virtex-5 FPGAs genutzt. Alle anderen behandelten FPGAs unterstützen diesen Standard nicht und können deshalb nicht evaluiert werden. Tabelle 3.1 zeigt die Leistungsaufnahme der verschiedenen PCI-Varianten.

Tabelle 3.1: Leistungsaufnahme der wichtigsten Signale der betrachteten FPGAs bei Implementierung verschiedener PCI-Varianten.

PCI 32 Bit / 33 MHz			
Signal	Bitbreite	Virtex-4	Virtex-5
AD	32	6,3 mW	7,8 mW
CLK	1	10,4 mW	12,3 mW
CBE	4	4,2 mW	5,1 mW
IRDY/TRDY	2	4,2 mW	5,1 mW
sonstige	3	3,2 mW	3,8 mW
Gesamt	**42**	**242,7 mW**	**299,0 mW**
PCI-X 64 Bit / 66 MHz			
AD	64	10,4 mW	13,0 mW
CLK	1	17,6 mW	22,0 mW
CBE	8	6,3 mW	7,8 mW
IRDY/TRDY	2	6,3 mW	7,8 mW
sonstige	3	4,2 mW	5,1 mW
Gesamt	**78**	**754,9 mW**	**967,5 mW**
PCI-X 64 Bit / 133 MHz			
AD	64	17,7 mW	21,0 mW
CLK	1	32,1 mW	43,0 mW
CBE	8	10,4 mW	15,0 mW
IRDY/TRDY	2	10,4 mW	15 mW
sonstige	3	6,3 mW	7,2 mW
Gesamt	**78**	**1279,2 mW**	**1592,3 mW**

Mit den Ergebnissen der Simulation lässt sich ausgehend von Gleichung 3.6 die benötigte Energie pro Bit bestimmen.

$$PDP_{PCI} = \frac{P \cdot T_{Bit}}{N} \mid N = \begin{cases} 32 & \text{, PCI 32 Bit} \\ 64 & \text{, PCI-X 64 Bit} \end{cases} \tag{3.10}$$

PCI unterstützt beliebig lange Block-Transfers, der Mehraufwand des Kommunikationsprotokolls ist deswegen bei langen Übertragungen zu vernachlässigen.

$$PDP_{PCI,code} = PDP_{PCI} \tag{3.11}$$

Bei einer Transaktion, wie in Abbildung 3.5 zu sehen, kommen immer wieder Wartezustände vor, bei denen effektiv keine Nutzdaten übertragen werden. Die maximal mögliche Datenrate kann also in der Realität nicht erreicht werden. Über Leistungstests mit unterschiedlichen Datenmengen ergibt sich für PCI ein Durchsatz von durchschnittlich ca. 90 % der maximalen Datenrate bei langen Block-Transfers und ca. 60 % bei kurzen Block-Transfers. Hier wird deshalb eine durchschnittliche Ausnutzung von ca. 75 % der Bandbreite angenommen [R61] [R56]. Tabelle 3.2 und Abbildung 3.6 stellen die Ergebnisse tabellarisch und grafisch dar.

$$PDP_{PCI,real} = PDP_{PCI,code} \cdot 1,25 = PDP_{PCI} \cdot 1,25 \tag{3.12}$$

Tabelle 3.2: Benötige Energie pro Bit (PDP_{real}) verschiedener PCI-Varianten, bei Implementierung auf Virtex-4 und Virtex-5 FPGAs.

FPGA	PCI 33 Bit/33 MHz	PCI 64 Bit/66 MHz	PCI 64 Bit/133 MHz
Virtex-4	354 pJ	286 pJ	234 pJ
Virtex-5	287 pJ	223 pJ	189 pJ

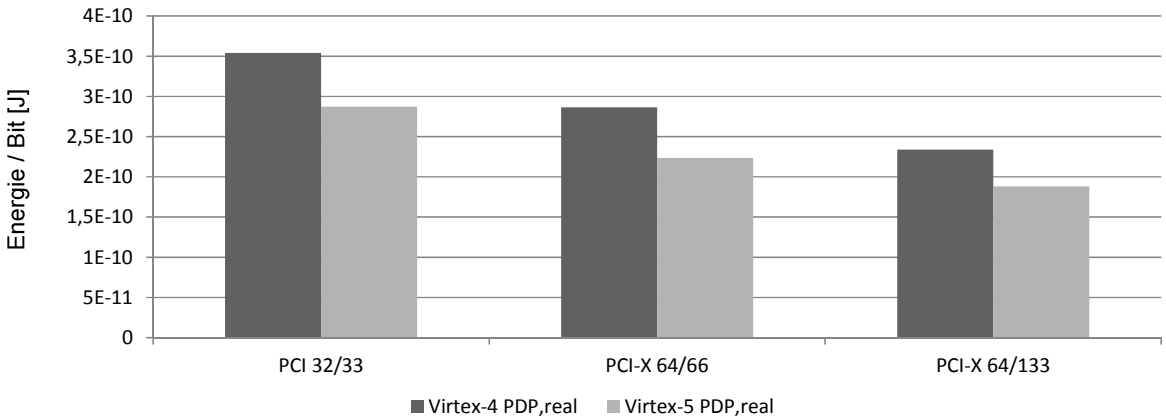

Abbildung 3.6: PDP_{real} von drei PCI-Varianten bei Implementierung auf Virtex-4 und Virtex-5 FPGAs.

Wird die durchschnittliche Leistungsaufnahme von PCI betrachtet, so weist die PCI-Variante mit 32 Bit Busbreite und 33 MHz Datenrate die geringsten Werte auf. Der Zuwachs in der Leistungsaufnahme bei Verwendung eines breiteren Busses oder einer höheren Taktfrequenz skaliert jedoch nicht linear. Ein Transceiver auf Basis eines Virtex-4 benötigt 6, 3 mW bei 33 MHz Taktfrequenz. Eine Verdopplung der Taktfrequenz ergibt eine Erhöhung der Verlustleistung um zwei Drittel auf 10, 4 mW. Eine weitere Verdopplung der Frequenz ergibt ebenfalls eine Steigerung der Verlustleistung um ca. zwei Drittel auf 17, 7 mW. Obwohl jeweils die doppelte Anzahl an Bits pro

Zeiteinheit übertragen wird, verdoppelt sich nicht die Leistungsaufnahme. Die benötigte Energie für ein zu übertragendes Bit reduziert sich also mit steigender Frequenz. Eine Erhöhung der Parallelität bei der Datenübertragung von PCI wirkt sich ebenfalls positiv auf den Parameter *PDP* aus, da die Anzahl der benötigten Zusatzsignale mit einem geringeren Faktor wächst als die Anzahl von Daten- und Adressleitungen.

3.2 GTL - Gunning Transceiver Logic

GTL (Gunning Transceiver Logic [R30]) wurde ursprünglich entwickelt, um Busse zwischen Prozessoren und Speichermodulen zu implementieren. Hierbei wird der Sender oder Leitungstreiber als offene Drainschaltung mittels Feldeffekttransistoren (ältere Variante mit offenem Kollektor und Bipolartechnik) und der Empfänger mittels Differenzverstärker (siehe Abbildung 3.7) realisiert.

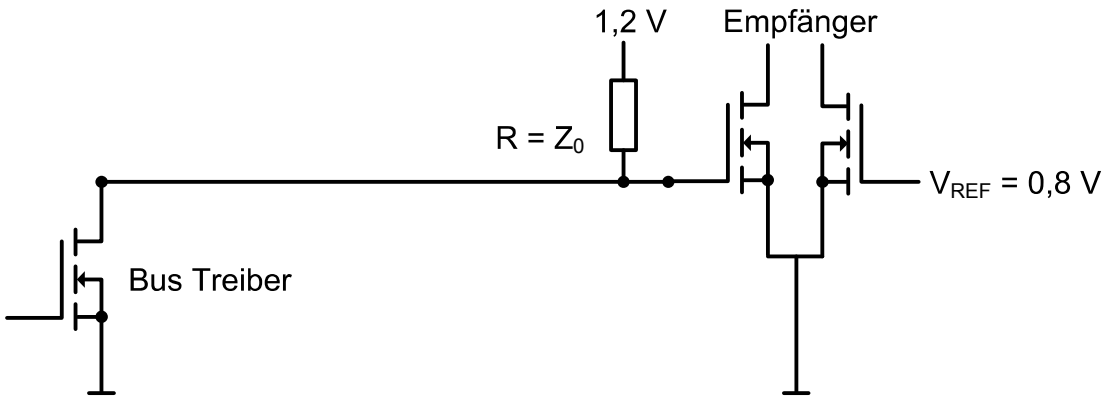

Abbildung 3.7: Eine offene Drainschaltung am Treiber und ein Differenzverstärker am Empfänger bilden einen GTL-Bus.

Eine fallende Flanke in der Datenübertragung wird durch den Leitungstreiber erzeugt, die steigende Flanke ermöglicht ein passiver Widerstand zwischen Leitung und Terminierungsspannung $U_{term} = 1,2\,\mathrm{V}$. Der Wert des Widerstands wird an die Impedanz der Busstruktur angepasst, um Reflektionen auf der Leitung zu vermeiden. Die Spannungspegel am Bus (siehe Abbildung 3.8) ergeben sich aus der verwendeten Referenzspannung $U_{OH} = U_{term}$ und der Sättigungsspannung des Transistors am Leitungstreiber $U_{OL} = U_{satt}$. U_{satt} beträgt bei GTL-Bussen $0,4\,\mathrm{V}$. So ergibt sich ein maximaler Spannungshub von $0,8\,\mathrm{V}$, wobei die Referenzspannung des Busses genau in der Mitte des Spannungshubes bei $0,8\,\mathrm{V}$ festgelegt ist. GTL-Treiber können standardmäßig einen Strom von 40 mA über den Bus treiben, was zu einer maximal zulässigen Buslast von $20\,\Omega$ führt ($\frac{800\,\mathrm{mV}}{40\,\mathrm{mA}} = 20\,\Omega$). Außerdem bedeutet der Treiberstrom von 40 mA, in Ver-

bindung mit ihrer Sättigungsspannung von 0, 4 V, eine maximale Verlustleistung pro Treiber von 16 mW. Die maximale Busfrequenz liegt bei 100 MHz.

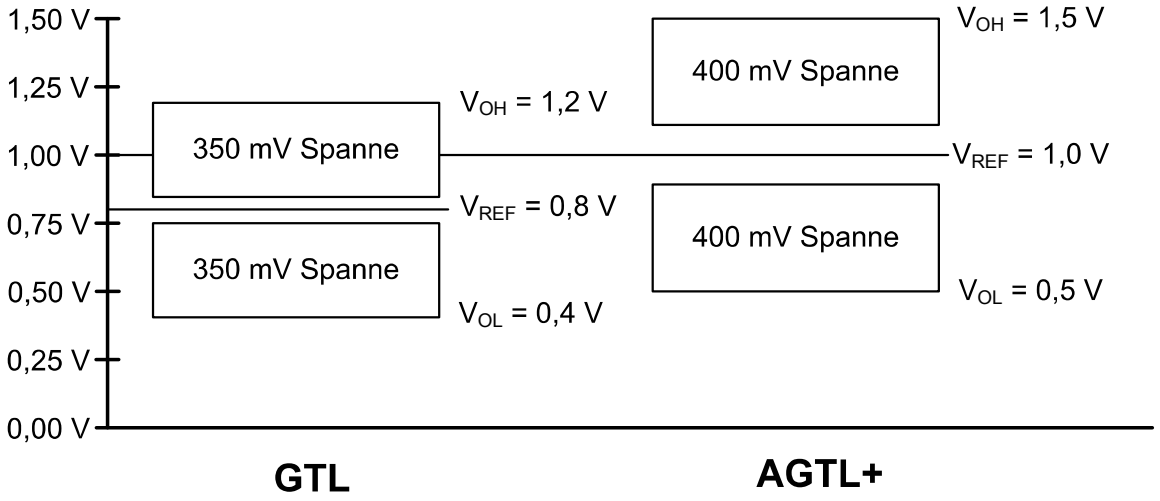

Abbildung 3.8: Signalpegel bei GTL und AGTL+ [R30].

AGTL+ (Assisted Gunning Transceiver Logic+ [R30]) ist eine Weiterentwicklung des GTL Standards und weist im Vergleich zu GTL unterschiedliche Logikpegel auf (siehe Abbildung 3.8). AGTL+ nutzt einen Spannungshub von 1 V, zwischen 0, 5 V und 1, 5 V, diese Werte können jedoch je nach Implementierung schwanken. Die Referenzspannung beträgt 2/3 der Terminierungsspannung. Die Bustreiber und Empfänger sind ähnlich aufgebaut, wie im GTL Verfahren, jedoch wird bei AGTL+ der Bus am Anfang und am Ende mit jeweils 50 Ω terminiert. Aus dem Treiberstrom von 40 mA resultiert eine maximale Buslast von 25 Ω. Ein Treiber weist eine Verlustleistung von 16 mW auf. Dieses Verfahren wird sowohl für Busse innerhalb von Schaltungen, als auch zur Kommunikation zwischen integrierten Schaltungen verwendet. Speziell in der Intel Netburst Architektur findet AGTL+ Anwendung, welche hauptsächlich bei den Prozessorfamilien Pentium 4, Athlon, Xeon sowie Core zu finden ist. Diese Technik ist besser bekannt als Front-Side-Bus. Von den betrachteten FPGAs beinhalten nur Virtex-4 FPGAs und Virtex-5 FPGAs GTL-Transceiver. Tabelle 3.3 fasst die wichtigsten Parameter des AGTL+ Standards zusammen.

Tabelle 3.3: AGTL+ Parameter [R30].

Parameter	Wert	Anmerkung
maximale Signalfrequenz	400 MHz	
maximale Datenrate	1600 Mbit/s	QDR Verfahren (siehe Kapitel 3.2.1)
Terminierungsspannung U_{TT}	$1,5$ V	
Referenzspannung U_{REF}	$2/3U_{TT}$	
Spannungshub	1 V	
Schwellwert	$0,1$ V	um U_{REF}
Terminierung	50 Ω	an beiden Bus-Enden
Treiberstrom	40 mA	

3.2.1 FSB - Frontside Bus

Der Frontside Bus (FSB [R66]) wird von einigen Prozessorgenerationen verwendet, um Daten zwischen der CPU und dem Chipsatz auszutauschen. Es handelt sich um eine auf der AGTL+ Technik basierende Busstruktur mit einem 32 Bit breiten Adressbus und einem Datenbus von 64 Bit Breite. Der FSB nutzt entweder das DDR- oder das QDR-Verfahren. Bei dem DDR-Verfahren werden bei der steigenden Taktflanke wie auch bei der fallenden Taktflanke Daten übertragen, was beispielsweise bei Athlon-Prozessoren zum Einsatz kommt. Das Quadruple Datarate Verfahren (QDR) wird hingegen bei dem sogenannten Quadpumped-FSB (QDR, vierfache Abtastrate) verwendet. Es werden hierbei zwei um 90 ° verschobene Takte verwendet, um vier Abtastzeitpunkte pro Takt zu ermöglichen. So können pro Takt doppelt so viele Daten gelesen oder geschrieben werden wie bei dem Doublepumped-FSB (DDR, doppelte Abtastrate). Prozessorfamilien, wie Pentium 4 oder Core, verwenden diese Art von FSB. Die zu übertragende Datenmenge pro Sekunde ergibt sich aus der Busbreite, der Taktfrequenz und dem verwendeten Übertragungsverfahren. So kann ein 100 MHz schneller und 64 Bit breiter FSB im QDR-Verfahren $3,2$ Gbyte/s übertragen. Hierbei muss jedoch beachtet werden, dass bei dem FSB nicht gleichzeitig mehrere Teilnehmer senden können, die Übertragung also unidirektional stattfindet. Tabelle 3.4 gibt eine Übersicht der verwendeten FSB-Varianten und ihrer Übertragungsraten. Ein FSB-Gerät weist im Wesentlichen drei Signalgruppen auf.

- Datensignale.
  - 64 Datenleitungen.
  - 4 differentielle Abtastimpulssignale.
  - 3 Signale für dynamische Inversion.
- Adresssignale.
  - 32 Adressleitungen.

  - 2 Abtastimpulssignale.
  - 5 Anforderungssignale.
- 12 verschiedene Taktsignale.

Tabelle 3.4: FSB-Varianten mit zugehörigen Bezeichnungen und Übertragungsraten.

Frequenz	Verfahren	Bezeichnung	Bitrate
100 MHz	DDR	FSB 200	1,6 Gbyte/s
133 MHz	DDR	FSB 266	2,13 Gbyte/s
166 MHz	DDR	FSB 333	2,66 Gbyte/s
200 MHz	DDR	FSB 400	3,2 Gbyte/s
100 MHz	QDR	FSB 400	3,2 Gbyte/s
133 MHz	QDR	FSB 533	4,30 Gbyte/s
166 MHz	QDR	FSB 667	5,30 Gbyte/s
200 MHz	QDR	FSB 800	6,40 Gbyte/s
266 MHz	QDR	FSB 1066	8,50 Gbyte/s
333 MHz	QDR	FSB 1333	10,6 Gbyte/s
400 MHz	QDR	FSB 1600	12,8 Gbyte/s

Aufgrund der unterschiedlichen Signalgruppen bei einer FSB-Implementierung müssen die Transceivermodelle in der *HSPICE*-Netzliste mit unterschiedlichen Stimuli angesteuert werden. So werden die Adresssignale beispielsweise mit der halben Frequenz wie die Datensignale betrieben. Tabelle 3.5 zeigt die Ergebnisse der *SPICE*-Simulation.

Da bei dem FSB immer 64 Bit gleichzeitig übertragen werden, ergibt sich Gleichung 3.6 zu:

$$PDP_{FSB} = \frac{P \cdot T_{Bit}}{64} \tag{3.13}$$

Dieses Maß gibt die benötigte Energie pro übertragenem Bit auf den Datenleitungen an, der Einfluss der zusätzlichen Signalleitungen ist in diesem Term enthalten. FSB unterstützt Block-Verfahren, die den Protokoll-Mehraufwand bei langen Transfers anteilig verringern. Hierbei wird eine volle Auslastung der Kommunikation mit Nutzdaten angenommen. Bei dem FSB werden in einer Datenübertragung Steuer- und Kontrollinformationen ausgetauscht, welche im praktischen Einsatz ca. ein Viertel der gesamten Datenmenge ausmachen [R20]. Deshalb beträgt die Nutzdatenmenge im realen Einsatz 75 % der gesamten, zu übertragenden Datenmenge. Tabelle 3.6 fasst die drei Bewertungsmaße zusammen.

$$PDP_{FSB,real} = PDP_{FSB,code} = PDP_{FSB} \cdot 1,25 \tag{3.14}$$

Tabelle 3.5: Leistungsaufnahme der wichtigsten Signale der betrachteten FPGAs, bei Implementierung verschiedener FSB-Varianten.

FSB-Typ	Taktsignale	Datensignale	Adresssignale	**Gesamt**
Virtex-5				
FSB 200	11,5 mW	10,7 mW	10,7 mW	**1365 mW**
FSB 266	10,8 mW	11,8 mW	10,9 mW	**1452 mW**
FSB 333	11,1 mW	12,3 mW	11,1 mW	**1501 mW**
FSB 400	11,6 mW	12,6 mW	10,6 mW	**1516 mW**
FSB 533	10,9 mW	13,5 mW	11,8 mW	**1616 mW**
FSB 667	11,1 mW	14,3 mW	12,3 mW	**1697 mW**
FSB 800	11,7 mW	14,9 mW	12,6 mW	**1762 mW**
Virtex-4				
FSB 200	50,3 mW	51,2 mW	47,7 mW	**2117 mW**
FSB 266	51,9 mW	53,4 mW	49,0 mW	**2197 mW**
FSB 333	54,1 mW	56,0 mW	50,2 mW	**2287 mW**
FSB 400	55,0 mW	58,0 mW	51,2 mW	**2353 mW**
FSB 533	51,9 mW	60,3 mW	53,4 mW	**2429 mW**
FSB 667	54,1 mW	62,2 mW	56,0 mW	**2520 mW**
FSB 800	55,0 mW	64,2 mW	57,9 mW	**2600 mW**

Tabelle 3.6: Die benötige Energie pro Bit unterschiedlicher FSB-Varianten.

FSB-Typ	PDP_{FSB}	$PDP_{FSB,code}$	$PDP_{FSB,real}$
Virtex-4			
FSB 200	165 pJ	165 pJ	206 pJ
FSB 266	129 pJ	129 pJ	161 pJ
FSB 333	107 pJ	107 pJ	134 pJ
FSB 400	92 pJ	92 pJ	115 pJ
FSB 533	71 pJ	71 pJ	89 pJ
FSB 667	59 pJ	59 pJ	74 pJ
FSB 800	51 pJ	51 pJ	64 pJ
Virtex-5			
FSB 200	107 pJ	107 pJ	133 pJ
FSB 266	85 pJ	85 pJ	107 pJ
FSB 333	70 pJ	70 pJ	88 pJ
FSB 400	59 pJ	59 pJ	74 pJ
FSB 533	47 pJ	47 pJ	59 pJ
FSB 667	40 pJ	40 pJ	50 pJ
FSB 800	34 pJ	34 pJ	43 pJ

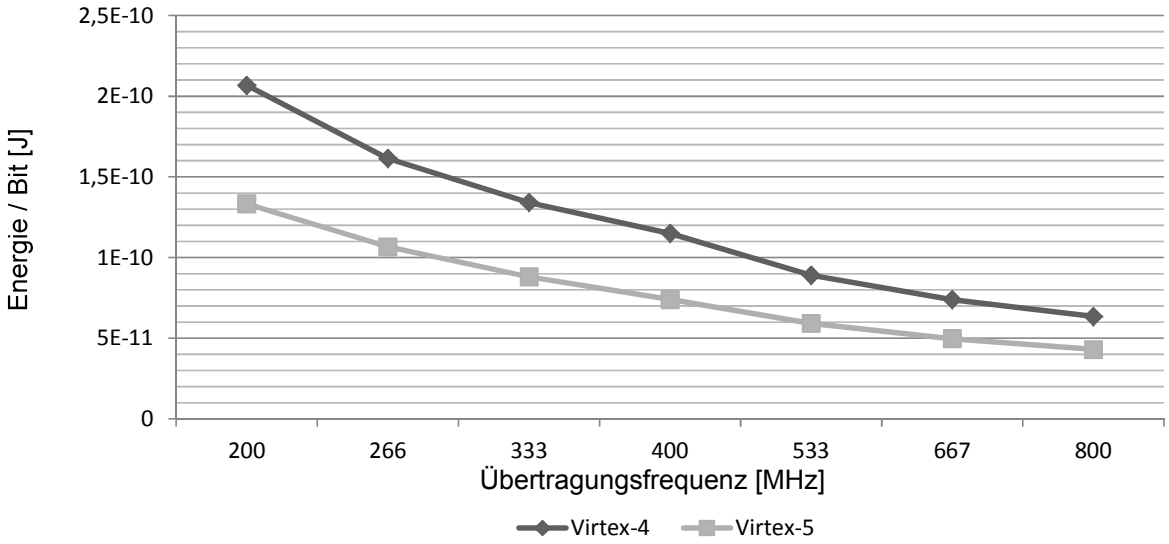

Abbildung 3.9: Die benötigte Energie pro Bit ($PDP_{FSB,real}$) verschiedener FSB-Varianten in Abhängigkeit der Übertragungsfrequenz.

Eine Erhöhung der Übertragungsfrequenz verringert im FSB-Verfahren die benötigte Energie pro Bit (vgl. Abbildung 3.9), da sich eine Verdopplung der Übertragungsfrequenz nicht in einer Verdopplung der Leistungsaufnahme sondern in einer Zunahme von ca. 55 % äußert. Im Gegensatz zu anderen in dieser Arbeit betrachteten Kommunikationsverfahren fällt die Diskrepanz zwischen einer Steigerung der Übertragungsrate und dem Wachstum der Leistungsaufnahme geringer aus. So erhöht sich beispielweise die durchschnittliche Verlustleistung bei einem Wechsel von FSB-200 auf FSB-800 nur um ca. 25 %, obwohl sich die Übertragungsrate vervierfacht. Der Grund hierfür liegt in der großen Anzahl von Signalen, deren akkumulierte frequenzunabhängige Stromaufnahme die Gesamtverlustleistung dominiert.

3.2.2 Media Independent Interface (MII, GMII, RMII, RGMII, XGMII)

Medienunabhängige Schnittstellen werden bei diversen Ethernet Standards (siehe Kapitel 3.4) benötigt, um Daten zwischen MAC und PHY zu übertragen. Bei diesen Komponenten handelt es sich um eine Komponente für die Medienzugriffskontrolle (MAC) und einen Baustein für die eigentliche Bitübertragung (PHY). Diese Schnittstellen sind für verschiedene Ethernet-Varianten verfügbar und entsprechend unterschiedlich implementiert. Allen Implementierungen gemeinsam ist die Verwendung von GTL-kompatiblen Pegeln zur Signalübertragung. Die nachfolgende Aufzählung gibt eine Übersicht der gängigen Übertragungsverfahren.

- 100BaseTX - Fast Ethernet:
  - **MII - Media Independent Interface [R47]:** Verwendet unidirektional vier parallele Datenleitungen, eine Taktleitung und diverse Kontrollsignale (insgesamt 16 Signale). Die Datenleitungen werden mit 25 MHz getaktet, was zu einer akkumulierten Datenrate von 100 Mbit/s führt.
  - **RMII - Reduced Media Independent Interface [R58]:** Reduziert die von MII benötigte Anzahl an Signalen und verwendet nur zwei anstatt vier Datenleitungen. Um eine Datenrate von 100 Mbit/s zu erreichen, muss der Takt entsprechend 50 MHz betragen.
- 1000BaseT - Gigabit Ethernet
  - **GMII - Gigabit Media Independent Interface [R25]:** Das GMII ist abwärtskompatibel mit dem MII, verwendet jedoch unidirektional acht Datenleitung. Über ein dediziertes Signal wird ein Takt mit 125 MHz gesendet. Hieraus ergibt sich eine Übertragungsrate von 1000 Mbit/s.
  - **RGMII - Reduced Gigabit Media Independent Interface [R59]:** Ähnlich wie das RMII reduziert das RGMII die benötigten Datenleitungen einer GMII von 28 auf 12 Signale. Die acht Datenleitungen werden durch vier Signale ersetzt und im DDR-Verfahren betrieben. Zusammen mit dem Taktsignal von 125 MHz wird die benötigte Datenrate von 1000 Mbit/s erreicht.
- 10G Ethernet
  - **XGMII - 10 Gigabit Media Independent Interface [R27]:** Das XGMII besteht pro Richtung aus einem 32 Bit breiten Datenbus mit einem dedizierten Taktsignal von 156, 25 MHz und vier Kontrollsignalen. Die Kontrollsignale kodieren jeweils 1 Byte der Datenleitungen und geben an, ob es sich um Nutzdaten oder Leerlaufdaten handelt. Außerdem werden Fehler im Datenstrom sowie der Start und das Ende eines Ethernetpakets angezeigt. Die benötigte Datenrate von 10 Gbit/s wird durch den Betrieb der Datenleitungen im DDR-Verfahren ermöglicht.

Tabelle 3.7 fasst die Merkmale der unterschiedlichen, medienunabhängigen Schnittstellen zusammen. Tabelle 3.8 zeigt die Ergebnisse der *SPICE*-Simulationen aller betrachteten medienunabhängigen Schnittstellen.

Tabelle 3.7: Merkmale unterschiedlicher, medienunabhängiger Schnittstellen auf GTL-Basis.

	Signale (gesamt)	Signale (Daten)	Verfahren	Taktfrequenz	Datenrate
MII	16	4 / Richtung	SDR	25 MHz	100 Mbit/s
RMII	10	2 / Richtung	SDR	50 MHz	100 Mbit/s
GMII	28	8 / Richtung	SDR	125 MHz	1000 Mbit/s
RGMII	12	4 / Richtung	DDR	125 MHz	1000 Mbit/s
XGMII	74	32 / Richtung	DDR	156, 25 MHz	10000 Mbit/s

Tabelle 3.8: Leistungsaufnahme der betrachteten FPGAs, bei Implementierung verschiedener, medienunabhängiger Schnittstellen.

Schnittstelle	Taktsignale	Datensignale	**Gesamt (parallel)**
Virtex-5			
MII	10, 3 mW	10, 3 mW	**51,6 mW**
RMII	10, 6 mW	10, 3 mW	**31,3 mW**
GMII	11, 6 mW	10, 9 mW	**98,5 mW**
RGMII	11, 6 mW	11, 6 mW	**58,2 mW**
XGMII	11, 7 mW	12, 1 mW	**400,2 mW**
Virtex-4			
MII	46, 0 mW	45, 1 mW	**82,0 mW**
RMII	47, 5 mW	45, 9 mW	**65,9 mW**
GMII	51, 6 mW	42, 7 mW	**120,0 mW**
RGMII	51, 6 mW	52, 8 mW	**93,9 mW**
XGMII	53, 4 mW	55, 1 mW	**406,1 mW**

Die verschiedenen medienunabhängigen Schnittstellen unterscheiden sich im Wesentlichen in der Anzahl paralleler Datenleitungen und in der Übertragungsrate. *PDP* ergibt sich deshalb aus Gleichung 3.6 zu:

$$PDP_{xxII} = \frac{P \cdot T_{Bit}}{N} \mid N = \begin{cases} 4 & \text{, bei MII} \\ 2 & \text{, bei RMII} \\ 8 & \text{, bei GMII} \\ 4 & \text{, bei RGMII} \\ 32 & \text{, bei XGMII} \end{cases} \tag{3.15}$$

Diese Angabe liefert die benötigte Energie pro übertragenem Datenbit. Hierbei wird jedoch keine Unterscheidung getroffen, ob es sich um ein Datenbit oder ein Kontrollbit handelt. Da alle medienunabhängigen Schnittstellen für die Übertragung von Ethernetpaketen eingesetzt werden, kann die benötigte Energie unter Berücksichtigung des Ethernet-Paketformats und Gleichung 3.7 angegeben werden.

$$PDP_{xxII,code} = PDP_{xxII} \cdot \frac{1542}{1500} \tag{3.16}$$

Dieser Term gibt die theoretisch minimal benötigte Energie für ein zu übertragendes Bit an. Im praktischen Einsatz ergibt sich eine durchschnittliche Paketauslastung von ca. 87 % [R11] [R14]. Diese Angabe bezieht sich auf den Ethernet-Standard im Allgemeinen. Gleichung 3.8 ergibt sich zu:

$$PDP_{xxII,real} = PDP_{xxII,code} \cdot 1,13 \approx PDP_{xxII} \cdot 1,162 \tag{3.17}$$

Tabelle 3.9: Die benötige Energie pro Bit unterschiedlicher MII-Varianten.

Schnittstelle	PDP	PDP_{code}	PDP_{real}
Virtex-4			
MII	820 pJ	843 pJ	953 pJ
RMII	659 pJ	677 pJ	765 pJ
GMII	120 pJ	123 pJ	139 pJ
RGMII	94 pJ	96 pJ	109 pJ
XGMII	41 pJ	42 pJ	47 pJ
Virtex-5			
MII	516 pJ	530 pJ	599 pJ
RMII	313 pJ	322 pJ	363 pJ
GMII	98 pJ	101 pJ	114 pJ
RGMII	58 pJ	60 pJ	68 pJ
XGMII	40 pJ	41 pJ	46 pJ

Tabelle 3.9 und Abbildung 3.10 fassen die Ergebnisse zusammen. Bei medienunabhängigen Schnittstellen zeigt sich die annähernd frequenzunabhängige Leistungsaufnahme von GTL-Transceivern. Eine Verdopplung der Übertragungsfrequenz von 25 MHz (MII) auf 50 MHz (RMII) resultiert in einer Steigerung der Leistungsaufnahme von nur 300 μW pro Datenleitung, während die Gesamtleistungsaufnahme einer Datenleitung 10,3 mW beträgt. Entsprechend lohnt sich die Verwendung von Schnittstellen mit reduzierter Kanalbreite wie der Vergleich der Leistungsaufnahme von MII und RMII bzw. GMII und RGMII zeigt.

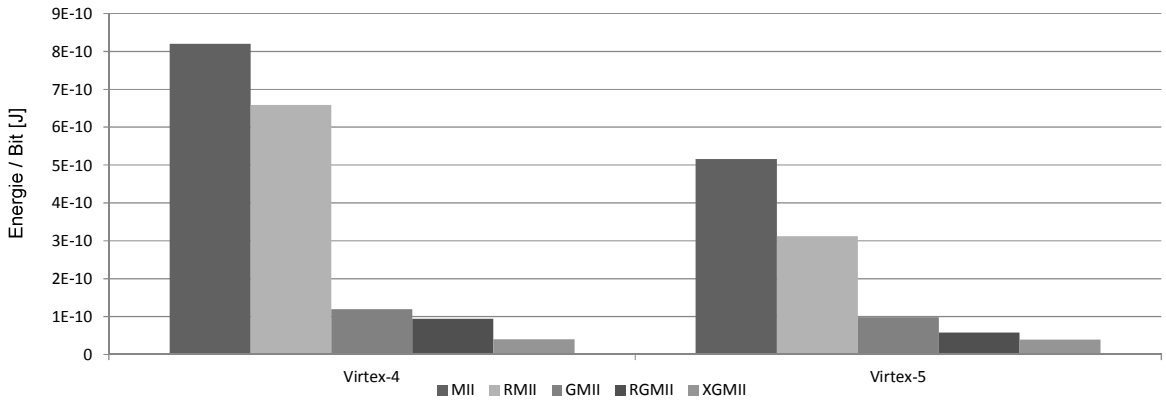

Abbildung 3.10: Die benötigte Energie pro Bit (PDP) unterschiedlicher medienunabhängiger Schnittstellen.

3.3 CML - Current Mode Logic

Current Mode Logic (CML [R38]) ist eine Übertragungstechnik, die in vielen modernen Hochgeschwindigkeitsschnittstellen implementiert wird. Sie bildet einen Quasi-Standard für die Datenübertragung in SerDes-basierten Komponenten (siehe Kapitel 2.2.3). Gängige CML-Implementierungen arbeiten mit Datenraten von 1 Gbit/s bis über 10 Gbit/s, wobei die Art der Implementierung bezüglich der Spannungs- und Strompegel nicht vorgegeben ist. CML-Transmitter und Receiver können daher über Kondensatoren miteinander gekoppelt werden, falls sie unterschiedliche Spannungspegel aufweisen, was als AC-Kopplung bezeichnet wird. Ohne diese Komponenten würde sich ein permanenter Stromfluss durch die Terminierungswiderstände einstellen. Eine Verbindung ohne Kondensatoren wird dementsprechend DC-Kopplung genannt und kann nur bei gleichen Pegeln eingesetzt werden. Obwohl CML keine festen Spannungspegel vorschreibt, verwenden viele Implementierungen die in Abbildung 3.11 gezeigten Werte.

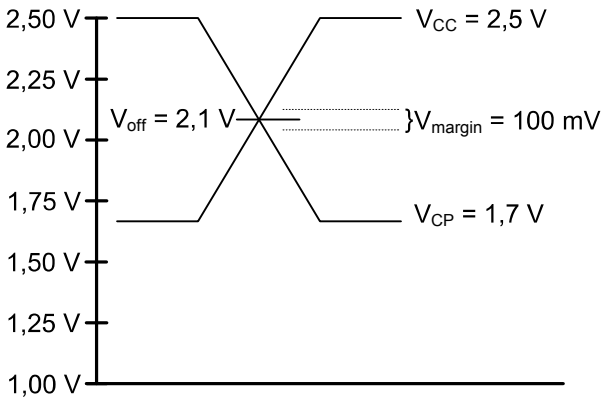

Abbildung 3.11: CML-Referenzpegel [R30].

Der Aufbau eines typischen CML-Senders und dazugehörigen Empfängers ist in Abbildung 3.12 gegeben. Der Sender besteht im Wesentlichen aus zwei Transistoren, an deren Emittern eine gemeinsame Konstantstromquelle hängt. Zwischen den jeweiligen Kollektoren und der Versorgungsspannung V_{CC} befinden sich Widerstände mit $50\,\Omega$. Die Transistoren werden so angesteuert, dass jeweils ein Transistor leitet und der andere sperrt. Dadurch fällt über dem entsprechenden Widerstand eine Spannung ab, welche durch den Widerstandswert und die Stromstärke definiert ist. In Abbildung 3.12 ergibt dies einen Spannungshub von $800\,\mathrm{mV}$ ($16\,\mathrm{mA} \cdot 50\,\Omega = 800\,\mathrm{mV}$). Die Widerstände dienen gleichzeitig als Terminierung bei der Verwendung von Übertragungskanälen mit entsprechender Impedanz.
Die Empfänger einer CML-Implementierung sind oft als Emitterfolger aufgebaut, die einen nachgeschalteten Differenzverstärker treiben. Dies sorgt für eine hohe Eingangsimpedanz der Verstärkerstufe. Die Terminierungswiderstände sind im abgebildeten Fall in den Empfänger integriert, können aber sowohl bei dem Sender als auch bei dem Empfänger als externe Komponenten vorgesehen sein, um eine möglichst große Flexibilität bei der Wahl der Übertragungskanäle zu gewährleisten. Nachfolgend werden wichtige, CML-basierte Übertragungsverfahren vorgestellt.

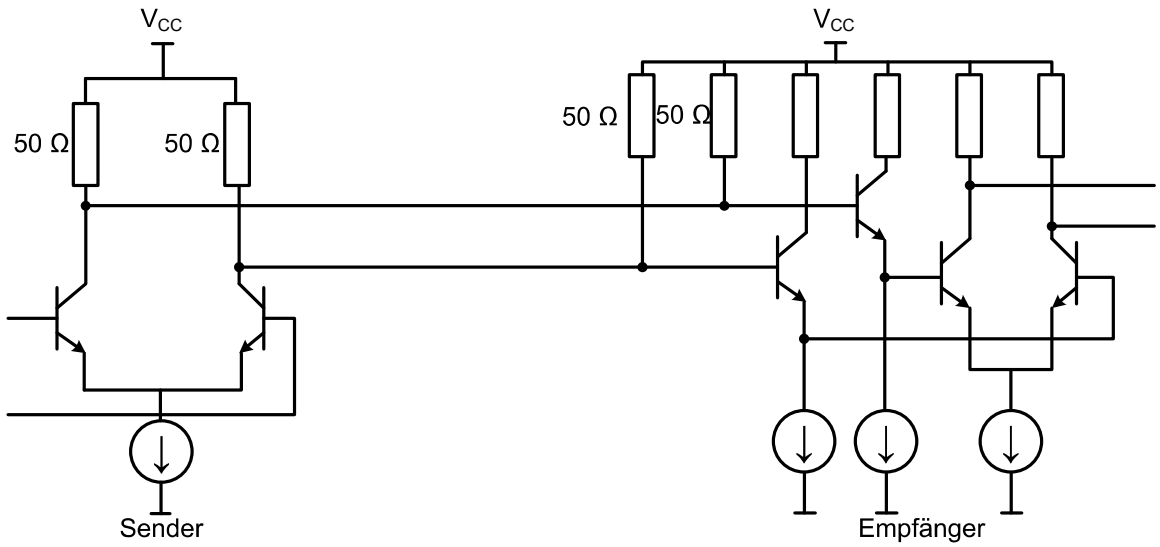

Abbildung 3.12: Aufbau eines typischen CML Senders und CML Empfängers [R30].

3.3.1 InfiniBand

InfiniBand [R31] ist eine auf dem CML-Standard basierende Übertragungstechnik mit Punkt zu Punkt Verbindungen. Sie wird vor allem in Anwendungsszenarien mit hohem Anspruch an die Leistungsfähigkeit des Netzwerks verwendet und zeichnet sich

durch hohen Durchsatz, geringe Latenzen und Skalierbarkeit aus. Eine InfiniBand-Verbindung besteht aus einem oder mehreren bidirektionalen CML-Übertragungsstrecken. Gängige InfiniBand-Implementierungen bestehen aus 1, 4 oder 12 Kanälen, wobei über einen einzelnen Kanal im SDR-Modus 2 Gbit/s an Daten (2,5 Gbit/s Bitrate) übertragen werden können. Die Daten werden hierbei über ein 8B/10B-Verfahren zur Taktwiederherstellung kodiert. Die resultierende Datenrate auf einem Kanal liegt also 20 % höher als die Nutzdatenrate. Der InfiniBand-Standard sieht bei kupferbasierten Leitungen auch Datenübertragungen im DDR- und QDR-Verfahren vor, weshalb die aus dem Verfahren und der Kanalbreite resultierenden Datenraten in Tabelle 3.10 aufgelistet sind. Als Übertragungsmedium werden Twinax-Kabel verwendet, welche die geforderten Datenraten bis zu einer Länge von 15 m umsetzen können.

Tabelle 3.10: Unidirektionale InfiniBand-Bitraten bei Verwendung unterschiedlicher Kanalbreiten und Verfahren.

	SDR	DDR	QDR
1X	0,31 GByte/s	0,625 GByte/s	1,25 GByte/s
4X	1,25 GByte/s	2,5 GByte/s	5 GByte/s
12x	3,75 GByte/s	7,5 GByte/s	15 GByte/s

InfiniBand verwendet ein Paketformat mit variabler Anzahl von Nutz- und Steuerdaten (siehe Abbildung 3.13). Je nach Größe des Paketkopfes und der Nutzdatenmenge schwankt der Anteil des Mehraufwands einer InfiniBand-Übertragung zwischen 2 % und 52 %.

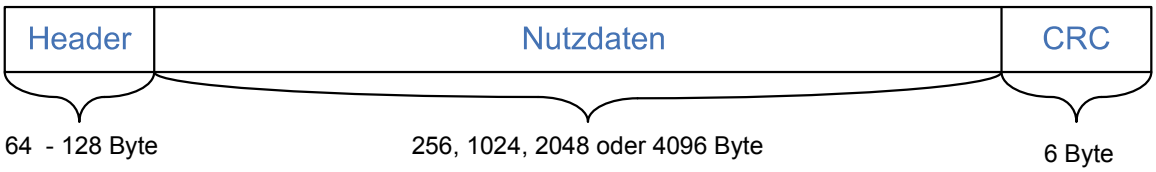

Abbildung 3.13: Allgemeiner Aufbau eines InfiniBand-Pakets [R31].

InfiniBand nutzt ein im Datenstrom eingebettetes Taktsignal, weshalb bei der *HSPICE*-Simulation keine dedizierte Implementierung des Signals erfolgen muss. Ebenso wenig werden separate Kanäle für Steuerinformationen benötigt. Die *HSPICE*-Netzliste besteht bei InfiniBand im Wesentlichen aus einer Anordnung von Sender, Empfänger, einem verbindenden Kanal und einem PRBS-Stimulus. Während die erste InfiniBand-Generation eine Datenrate von 2,5 Gbit/s nutzt, übertragen die zweite und dritte Generation Daten mit einer Rate von 5 Gbit/s bzw. 10 Gbit/s. Aufgrund dieser Voraussetzungen kann InfiniBand, je nach Generation, nicht auf allen verfügbaren Transceivertypen evaluiert werden. Tabelle 3.11 gibt die durchschnittliche Leistungsaufnahme

aller evaluierten InfiniBand-Varianten an. Die genauen Werte aller Simulationsergebnisse sowie die verwendeten Parameter sind im Anhang dieser Arbeit zu finden.

Tabelle 3.11: Die durchschnittliche Leistungsaufnahme verschiedener InfiniBand-Implementierungen.

FPGA-Typ	Anzahl Kanäle	InfiniBand Variante		
		SDR	DDR	QDR
V4-MGT	1x	315,8 mW	406,9 mW	
	4x	735,6 mW	1022 mW	
	12x	2207 mW	3066 mW	
V5-GTP	1x	121,6 mW		
	4x	414,4 mW		
	12x	1243 mW		
V5-GTX	1x	162,4 mW	180,0 mW	
	4x	530,1 mW	600,6 mW	
	12x	1590 mW	1802 mW	
S6-GTP	1x	309,5 mW		
	4x	1100 mW		
	12x	3301 mW		
V6-GTX	1x	176,1 mW	253,1 mW	
	4x	704,4 mW	1013 mW	
	12x	2113 mW	3038 mW	
V6-GTH	1x	365,2 mW	468,9 mW	647,3 mW
	4x	896,9 mW	1211 mW	1725 mW
	12x	2691 mW	3633 mW	5175 mW

PDP ergibt sich bei InfiniBand zu:

$$PDP_{IB} = \frac{P \cdot T_{Bit}}{N} \mid N = \begin{cases} 1 \\ 4 \\ 12 \end{cases} \tag{3.18}$$

Die maximal zu übertragende Bitmenge pro Paket beträgt 4096 Byte bei einem Mehraufwand von 134 Byte. Unter Berücksichtigung der 8B/10B-Kodierung erhöht sich PDP_{IB}.

$$PDP_{IB,code} = PDP_{IB} \cdot \frac{10}{8} \cdot \frac{4230}{4096} \tag{3.19}$$

$PDP_{IB,code}$ gibt die theoretische, minimal benötigte Energie pro Nutzdatenbit an. Unter praktischen Gesichtspunkten wird nicht jedes Paket mit 4096 Byte aufgefüllt. Durchschnittliche Auslastungen von 90 % werden im praktischen Einsatz erreicht [R40]

[R12]. Es ergibt sich eine realistische Aussage über die benötigte Energie pro Nutzdatenbit von InfiniBand (vgl. Tabelle 3.12).

$$PDP_{IB,real} = PDP_{IB,code} \cdot 1,1 \approx PDP_{IB} \cdot 1,42 \tag{3.20}$$

Die für ein zu übertragendes Nutzdatenbit benötigte Energie lässt sich durch die Verwendung mehrerer, paralleler InfiniBand-Kanäle senken (vgl. Abbildung 3.14). Die Verringerung der benötigten Energie bei einem Wechsel von einem Kanal auf vier Kanäle fällt bei V6-GTH-Transceivern und V4-MGT-Transceivern am höchsten aus, da diese Transceiver gemeinsame Ressourcen für jeweils zwei Instanzen verwenden. So steigt der Energiebedarf entsprechend weniger stark an als bei Transceivern, welche jeder Instanz eigene Ressourcen zuordnen. Entsprechend dieser Hierarchie im Transceiveraufbau ergibt sich keine Energieersparnis bei dem Wechsel von vier auf zwölf Transceiver. Die Einsparung von Energie bei InfiniBand lässt sich durch die Nutzung einer leistungsfähigeren Version stärker nutzen als bei der Parallelisierung der Kanäle (vgl. Abbildung 3.15). Es gibt keine Kombinationen aus unterstützten Datenraten der verschiedenen Generationen und variabler Kanalanzahl, um eine geforderte Datenrate mit unterschiedlichen Implementierungen zu realisieren, anders als z. B. PCIe (siehe Abbildung 3.23). Das bedeutet beispielsweise, eine benötigte Datenrate von genau 20 Gbit/s kann nur durch eine DDR-Variante mit vier parallelen Kanälen implementiert werden, da keine der beiden anderen Versionen diese Datenrate mit mehreren Kanälen genau erreicht.

Tabelle 3.12: Energie pro Bit ($PDP_{IB,code}$) verschiedener InfiniBand-Varianten.

Transceiver-Typ	PDP-Typ	InfiniBand-Variante		
		1x	4x	12x
InfiniBand SDR				
V4-MGT	PDP_{IB}	126 pJ	74 pJ	74 pJ
	$PDP_{IB,code}$	163 pJ	95 pJ	95 pJ
	$PDP_{IB,real}$	179 pJ	104 pJ	104 pJ
V5-GTP	PDP_{IB}	49 pJ	41 pJ	41 pJ
	$PDP_{IB,code}$	63 pJ	53 pJ	53 pJ
	$PDP_{IB,real}$	69 pJ	58 pJ	58 pJ
V5-GTX	PDP_{IB}	65 pJ	53 pJ	53 pJ
	$PDP_{IB,code}$	84 pJ	68 pJ	68 pJ
	$PDP_{IB,real}$	92 pJ	75 pJ	75 pJ
S6-GTP	PDP_{IB}	124 pJ	110 pJ	110 pJ
	$PDP_{IB,code}$	160 pJ	142 pJ	142 pJ
	$PDP_{IB,real}$	176 pJ	156 pJ	156 pJ
V6-GTX	PDP_{IB}	70 pJ	70 pJ	70 pJ
	$PDP_{IB,code}$	91 pJ	91 pJ	91 pJ
	$PDP_{IB,real}$	100 pJ	100 pJ	100 pJ
V6-GTH	PDP_{IB}	146 pJ	90 pJ	90 pJ
	$PDP_{IB,code}$	189 pJ	116 pJ	116 pJ
	$PDP_{IB,real}$	207 pJ	127 pJ	127 pJ
InfiniBand DDR				
V4-MGT	PDP_{IB}	81 pJ	51 pJ	51 pJ
	$PDP_{IB,code}$	105 pJ	66 pJ	66 pJ
	$PDP_{IB,real}$	116 pJ	73 pJ	73 pJ
V5-GTX	PDP_{IB}	36 pJ	30 pJ	30 pJ
	$PDP_{IB,code}$	46 pJ	39 pJ	39 pJ
	$PDP_{IB,real}$	51 pJ	43 pJ	43 pJ
V6-GTX	PDP_{IB}	51 pJ	51 pJ	51 pJ
	$PDP_{IB,code}$	65 pJ	65 pJ	65 pJ
	$PDP_{IB,real}$	72 pJ	72 pJ	72 pJ
V6-GTH	PDP_{IB}	94 pJ	61 pJ	61 pJ
	$PDP_{IB,code}$	121 pJ	78 pJ	78 pJ
	$PDP_{IB,real}$	133 pJ	86 pJ	86 pJ
InfiniBand QDR				
V6-GTH	PDP_{IB}	65 pJ	43 pJ	43 pJ
	$PDP_{IB,code}$	84 pJ	56 pJ	56 pJ
	$PDP_{IB,real}$	92 pJ	61 pJ	61 pJ

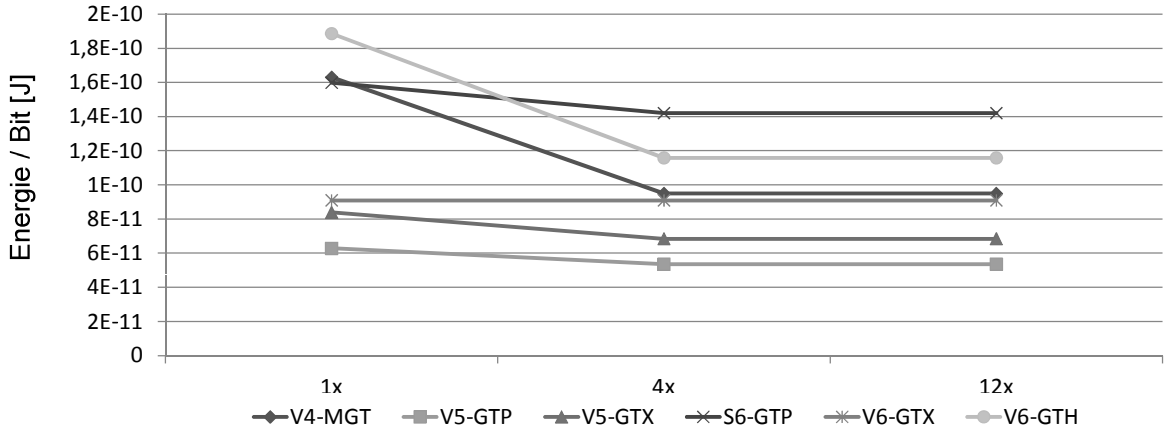

Abbildung 3.14: Die benötigte Energie pro Bit ($PDP_{IB,code}$) von InfiniBand-SDR in Abhängigkeit von der Anzahl paralleler Kanäle.

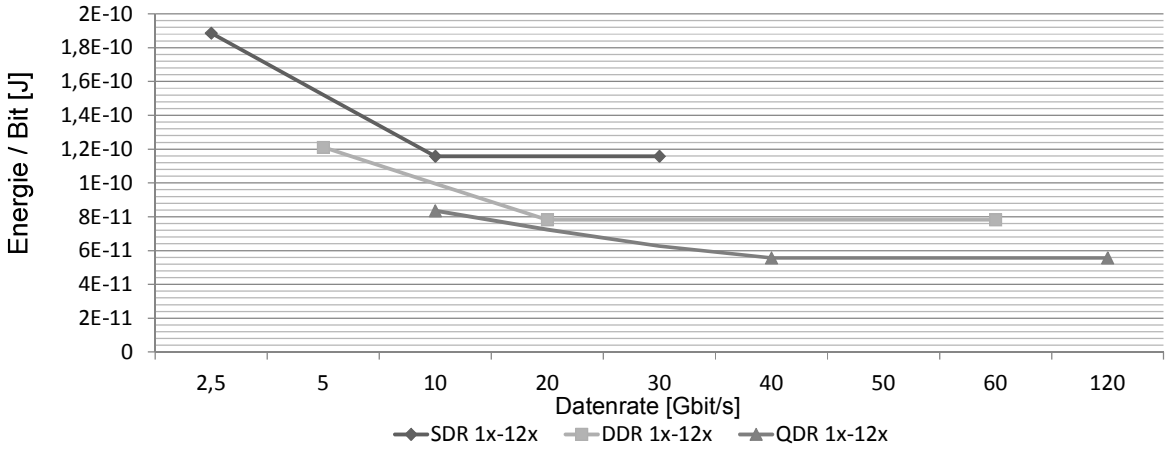

Abbildung 3.15: Die benötigte Energie pro Bit verschiedener InfiniBand-Varianten in Abhängigkeit von der zugehörigen Übertragungsrate, auf Basis eines Virtex6-GTH-Transceivers.

3.3.2 SGMII - Serial Gigabit Media Independent Interface

Die serielle, medienunabhängige Schnittstelle für Gigabit Ethernet (SGMII [R18]) dient zur Kommunikation zwischen Medienzugangskontrolle (MAC) und physikalischer Schicht (PHY) auf einem Netzwerkadapter (vgl. Kapitel 3.4 und 3.2.2). Im Speziellen wird sie für glasfaserbasiertes Ethernet nach IEEE802.3z verwendet. SGMII basiert auf der CML-Technik und verwendet zur Datenübertragung je ein differentielles Datensignal und einen differentiellen Takt mit 100 Ω Impedanz und einer differentiellen Terminierung von ebenfalls 100 Ω. Die Datenleitungen werden mit einer Bitrate von 1,25 Gbit/s betrieben. Diese kommt durch die verwendete 8B/10B- Kodierung zu Stande und ergibt eine Nutzdatenrate von 1 Gbit/s. SGMII verwendet einen differentiellen Spannungshub von 300 mV bis 800 mV, bei einer Referenzspannung von 1075 mV bis 1325 mV. Da SGMII dedizierte Kanäle für die Übertragung von Takt- und Datensignal verwendet, müssen für die Simulation zwei verschiedene Übertragungsstrecken modelliert werden. Der Datenkanal wird mit dem PRBS-Signal angesteuert, während der Taktkanal ein Rechtecksignal als Stimulus erhält. Durch die Übertragungsfrequenz von 1,25 Gbit/s kann SGMII nicht mittels der Virtex6-GTH-Transceiver implementiert werden. Die Leistungsaufnahme der verschiedenen FPGAs (vgl. Tabelle 3.13) unterscheidet sich je nach Art der Transceiver und Technologie deutlich. Die ermittelten Werte ermöglichen einen Vergleich zwischen den unterschiedlichen FPGA-Typen bezüglich der durchschnittlichen Leistungsaufnahme bei Implementierung von SGMII. PDP gibt die Energie an, welche zur Übertragung eines Bits benötigt wird (vgl. Tabelle 3.14 und Abbildung 3.16) und ergibt sich bei SGMII zu:

$$PDP_{SGMII} = P \cdot T_{Bit} \tag{3.21}$$

Dies berücksichtigt jedoch nicht die von SGMII verwendete 8B/10B-Leitungskodierung und das Ethernet-Protokoll, welches maximal 1500 Byte an Nutzdaten, und inklusive Zusatzinformationen 1542 Byte pro Paket überträgt (vgl. Kapitel 3.3.4). Unter Berücksichtigung dieser Einschränkung erhöht sich PDP zu:

$$PDP_{SGMII,code} = P \cdot T_{Bit} \cdot \frac{10}{8} \cdot \frac{1542}{1500} = PDP_{SGMII} \cdot 1,285 \tag{3.22}$$

Praktische Tests zeigen eine Reduzierung der Paketauslastung in praktisch relevanten Anwendungen [R11]. Als plausiblen Wert für die reale, durchschnittliche benötigte Energie pro Nutzdatenbit von Ethernet wird ein Durchsatz von maximal 87 % der möglichen Bandbreite angegeben [R14]. Hierdurch erhöht sich PDP schließlich auf:

$$PDP_{SGMII,real} = PDP_{SGMII} \cdot 1,285 \cdot 1,13 = PDP_{SGMII} \cdot 1,45 \tag{3.23}$$

Tabelle 3.13: Leistungsaufnahme der betrachteten FPGAs bei Implementierung einer SGMII-Einheit. Die Einzelwerte beziehen sich jeweils auf einen einzelnen Transceiver, der Gesamtwert auf die gesamte Implementierung.

FPGA-Typ	Spannungsversorgung	Leistung Daten	Leistung Takt
Virtex4 MGT	VCC_{TX}	61,6 mW	62,3 mW
	VCC_{RX}	123,2 mW	124,5 mW
	$VCC_{TX-Term}$	112,8 mW	113,8 mW
	$VCC_{RX-Term}$	3,3 mW	3,2 mW
	VCC_{Int}	26,0 mW	26,0 mW
	Gesamt	**335,0 mW**	**337,7 mW**
		672,7 mW	
Virtex5 GTP	VCC_{TX}	18,0 mW	18,4 mW
	VCC_{RX}	12,3 mW	12,6 mW
	$VCC_{TX-Term}$	19,7 mW	20,1 mW
	$VCC_{RX-Term}$	21,4 mW	21,8 mW
	VCC_{Int}	11,0 mW	11,2 mW
	VCC_{PLL}	36,0 mW	36,7 mW
	Gesamt	**118,4 mW**	**120,8 mW**
		239,2 mW	
Virtex5 GTX	$VCC_{TX,RX}$	54,9 mW	57,1 mW
	$VCC_{TX-Term}$	68,6 mW	71,4 mW
	$VCC_{RX-Term}$	41,2 mW	42,8 mW
	VCC_{Int}	7,6 mW	8,0 mW
	VCC_{PLL}	58,8 mW	61,0 mW
	Gesamt	**232,1 mW**	**243,3 mW**
		475,4 mW	
Spartan6 GTP	VCC_{TX}	35,4 mW	36,8 mW
	VCC_{RX}	24,1 mW	24,6 mW
	$VCC_{TX-Term}$	57,9 mW	60,3 mW
	$VCC_{RX-Term}$	21,0 mW	21,8 mW
	VCC_{Int}	22,9 mW	23,9 mW
	VCC_{PLL}	67,5 mW	70,2 mW
	Gesamt	**228,8 mW**	**237,6 mW**
		466,4 mW	
Virtex6 GTX	$VCC_{TX,RX}$	46,6 mW	22,6 mW
	$VCC_{TX-Term}$	48,5 mW	50,2 mW
	$VCC_{RX-Term}$	39,5 mW	39,2 mW
	VCC_{Int}	12,5 mW	12,5 mW
	Gesamt	**147,0 mW**	**124,5 mW**
		271,5 mW	

Tabelle 3.14: Auflistung der benötigten Energie pro Bit von verschiedenen FPGA-Typen in Picojoule (10^{-12} J) bei Implementierung von SGMII.

FPGA-Typ	PDP_{SGMII}	$PDP_{SGMII,code}$	$PDP_{SGMII,real}$
Virtex4 MGT	538,2 pJ/Bit	691,6 pJ/Bit	781,5 pJ/Bit
Virtex5 GTP	189,6 pJ/Bit	243,6 pJ/Bit	275,2 pJ/Bit
Virtex5 GTX	377,0 pJ/Bit	484,4 pJ/Bit	547,4 pJ/Bit
Spartan6 GTP	373,4 pJ/Bit	479,9 pJ/Bit	315,3 pJ/Bit
Virtex6 GTX	217,2 pJ/Bit	279,0 pJ/Bit	610,3 pJ/Bit

In einer realistischen Anwendung liegt der Nutzdatendurchsatz von SGMII ca. 13 % unterhalb der maximalen Auslastung des Ethernet-Protokolls. Das Energieminimum liegt mit 275,2 pJ/Bit bei der Implementierung mittels Virtex5-GTP-Transceivern. Mit 781,5 pJ/Bit ist der Energiebedarf auf Basis von Virtex4-MGTs fast dreimal so hoch. Zu bemerken ist auch der erhöhte Energiebedarf des Taktsignals gegenüber dem Datensignal. Dies liegt an der größeren Anzahl von Taktflanken gegenüber dem PRBS-Datensignal und führt zum häufigen Umladen der Kapazität.

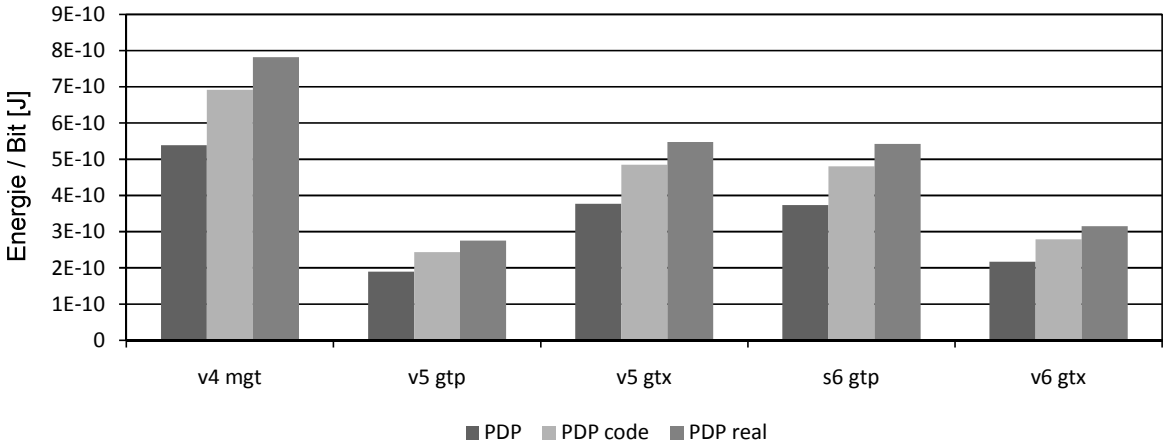

Abbildung 3.16: Grafische Übersicht der benötigten Energie pro Bit verschiedener FPGA-Typen bei Implementierung von SGMII (vgl. Tabelle 3.14).

3.3.3 XAUI - 10G Attachment Unit Interface

XAUI [R46] steht für 10-Gigabit-Schnittstellen-Anschlusseinheit und wird hauptsächlich für die Verbindung von Komponenten über impedanzkontrollierte Leiterbahnen bis zu 50 cm Entfernung verwendet. Hauptanwendungsbereich ist die Verbindung zwischen MAC (Medium Access Controller, Medienzugriffskontrolle) und PHY (Physical Interface, physikalische Schnittstelle) bei 10G Ethernet, oder die Punkt-zu-Punkt Verbindung zweier Komponenten (siehe Abbildung 3.17). Die vorgeschlagene Impedanz

der differentiellen Leiterbahnen und Konnektoren beträgt 100 Ω bei einem Frequenzbereich von 100 MHz bis 2,5 GHz. Wie der Name suggeriert, verwendet XAUI eine aggregierte Übertragungsrate von 10 Gbit/s. XAUI teilt diesen Datenstrom in vier parallele Ströme auf, die jeweils nach einer 8B/10B Kodierung über separate, differentielle Übertragungskanäle geleitet werden. Jeder dieser Kanäle weist demzufolge eine Bitrate von 3,125 Gbit/s auf und verwendet ein selbsttaktendes SerDes-Verfahren (siehe Kapitel 2.2.3). XAUI wird häufig als Ersatz für XGMII (siehe Kapitel 3.2.2) eingesetzt und nutzt die im XGMII-Standard enthaltenen Steuerinformation für die Synchronisation der einzelnen Kanäle sowie für das Markieren von Leerlaufzyklen.

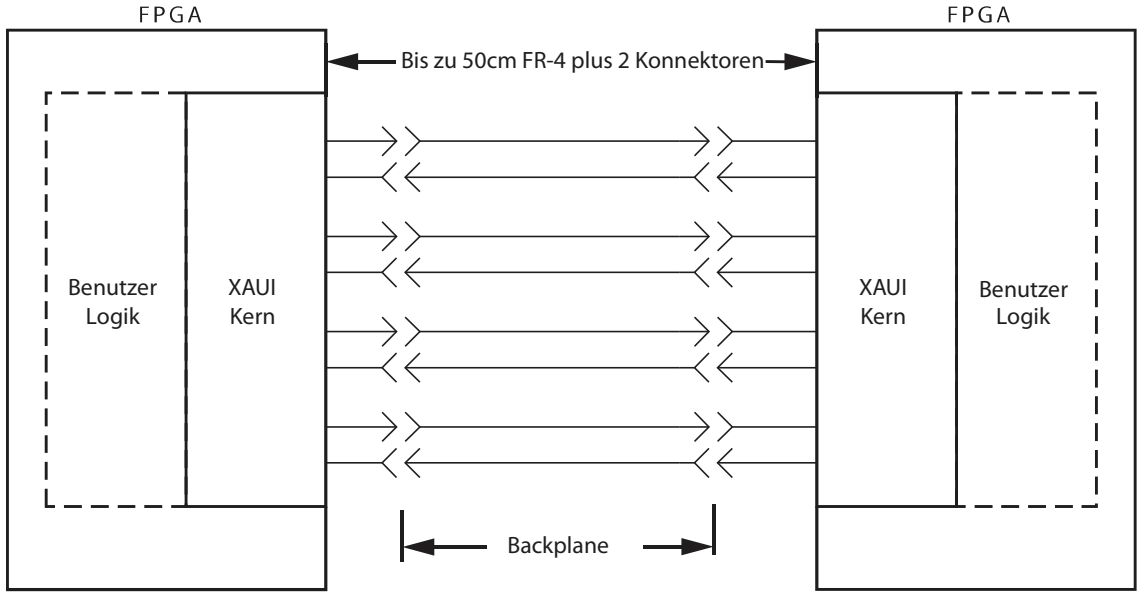

Abbildung 3.17: Anwendung von XAUI bei der Verbindung von zwei FPGAs über eine zentrale Bus-Leiterplatte (Backplane) [R46].

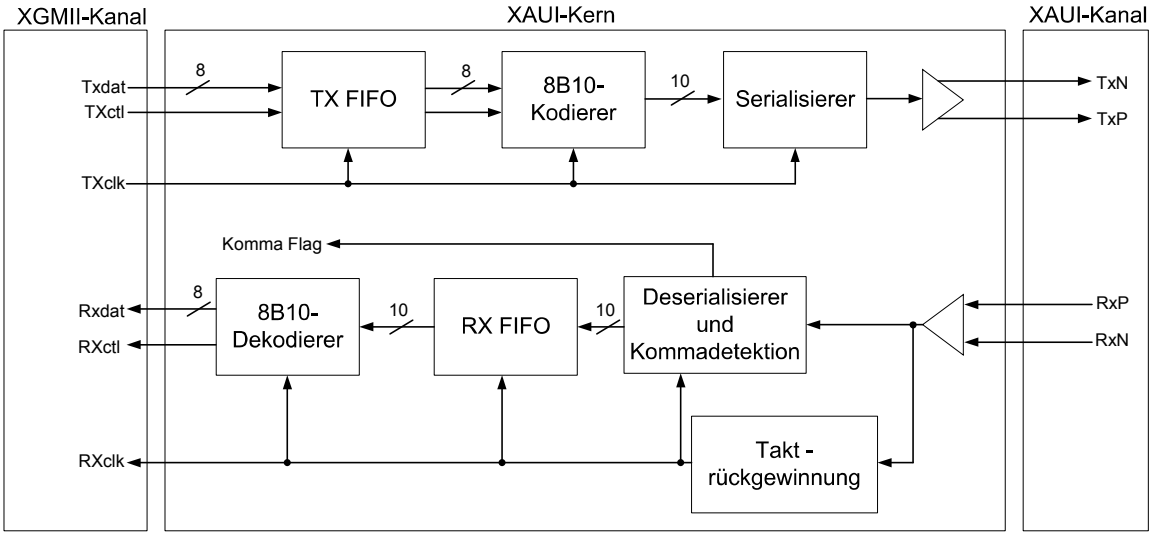

Abbildung 3.18: Schematischer Aufbau eines XAUI-Transceivers (nur ein Kanal gezeigt) [R64].

Die differentiellen Kanäle sind AC-gekoppelte Leiterbahnpaare für Punkt-zu-Punkt Verbindungen. Dies erlaubt die Verwendung unterschiedlicher Terminierungsspannungen der Kommunikationspartner. In jede Richtung sind vier differentielle Leiterbahnen vorhanden, was für eine komplette XAUI-Architektur eine Anzahl von acht Leiterpaaren bzw. 16 Leitern ergibt. Die typischen elektrischen Charakteristika einer XAUI-Übertragung gleichen den CML-Referenzangaben in Kapitel 3.3. Durch die Übertragungsfrequenz von $3,125\,\mathrm{Gbit/s}$ kann XAUI auf allen betrachteten FPGAs evaluiert werden. Die Ergebnisse der Simulation finden sich on Tabelle 3.15. Die ermittelten Werte ermöglichen einen Vergleich zwischen den unterschiedlichen FPGA-Typen bezüglich der durchschnittlichen Leistungsaufnahme bei Implementierung von XAUI. Da XAUI vier parallele Datenkanäle verwendet, ergibt sich *PDP* (vgl. Tabelle 3.16 und Abbildung 3.19) zu:

$$PDP_{XAUI} = \frac{P \cdot T_{Bit}}{4} \tag{3.24}$$

Dies berücksichtigt jedoch nicht die von XAUI verwendete 8B/10B-Leitungskodierung und das Ethernet-Protokoll. Unter Berücksichtigung dieser Einschränkung ergibt sich PDP zu:

$$PDP_{XAUI,code} = \frac{P \cdot T_{Bit}}{4} \cdot \frac{10}{8} \cdot \frac{1542}{1500} = PDP_{XAUI} \cdot 1,285 \tag{3.25}$$

XAUI nutzt in praktischen Anwendungen durchschnittlich 87 % der möglichen Bandbreite. Daraus ergibt sich:

$$PDP_{XAUI,real} = PDP_{XAUI} \cdot 1,285 \cdot 1,13 = PDP_{XAUI} \cdot 1,45205 \tag{3.26}$$

Tabelle 3.15: Leistungsaufnahme der betrachteten FPGAs bei Implementierung einer XAUI-Einheit. Die Einzelwerte beziehen sich jeweils auf einen einzelnen Transceiver, der Gesamtwert auf die gesamte Implementierung.

FPGA-Typ	Spannungsversorgung	Leistungsaufnahme
Virtex4 MGT	VCC_{TX}	86,5 mW
	VCC_{RX}	173,1 mW
	$VCC_{TX-Term}$	114,1 mW
	$VCC_{RX-Term}$	7,0 mW
	VCC_{Int}	65,3 mW
	Gesamt	**1211,9 mW**
Virtex5 GTP	VCC_{TX}	18,4 mW
	VCC_{RX}	11,5 mW
	$VCC_{TX-Term}$	19,7 mW
	$VCC_{RX-Term}$	21,4 mW
	VCC_{Int}	27,5 mW
	VCC_{PLL}	43,2 mW
	Gesamt	**480,3 mW**
Virtex5 GTX	$VCC_{TX,RX}$	41,3 mW
	$VCC_{TX-Term}$	34,5 mW
	$VCC_{RX-Term}$	21,0 mW
	VCC_{Int}	75,0 mW
	VCC_{PLL}	39,1 mW
	Gesamt	**615,1 mW**
Spartan6 GTP	VCC_{TX}	36,7 mW
	VCC_{RX}	23,1 mW
	$VCC_{TX-Term}$	59,0 mW
	$VCC_{RX-Term}$	21,4 mW
	VCC_{Int}	58,8 mW
	VCC_{PLL}	70,0 mW
	Gesamt	**936,0 mW**
Virtex6 GTX	$VCC_{TX,RX}$	92,4 mW
	$VCC_{TX-Term}$	48,7 mW
	$VCC_{RX-Term}$	39,5 mW
	VCC_{Int}	31,3 mW
	Gesamt	**847,2 mW**
Virtex6 GTH	VCC_{TX}	87,9 mW
	VCC_{RX}	44,4 mW
	VCC_{Term}	37,0 mW
	VCC_{Int}	30,0 mW
	VCC_{PLL}	192,2 mW
	Gesamt	**989,2 mW**

Tabelle 3.16: Die benötigte Energie pro Bit von verschiedenen FPGA-Typen in Picojoule (10^{-12} J) bei Implementierung von XAUI.

FPGA-Typ	PDP_{XAUI}	$PDP_{XAUI,code}$	$PDP_{XAUI,real}$
Virtex4 MGT	97,0 pJ/Bit	124,6 pJ/Bit	140,8 pJ/Bit
Virtex5 GTP	38,4 pJ/Bit	49,4 pJ/Bit	55,8 pJ/Bit
Virtex5 GTX	49,2 pJ/Bit	63,2 pJ/Bit	71,4 pJ/Bit
Spartan6 GTP	74,9 pJ/Bit	96,2 pJ/Bit	108,7 pJ/Bit
Virtex6 GTX	67,8 pJ/Bit	87,1 pJ/Bit	98,423 pJ/Bit
Virtex6 GTH	79,1 pJ/Bit	101,7 pJ/Bit	114,9 pJ/Bit

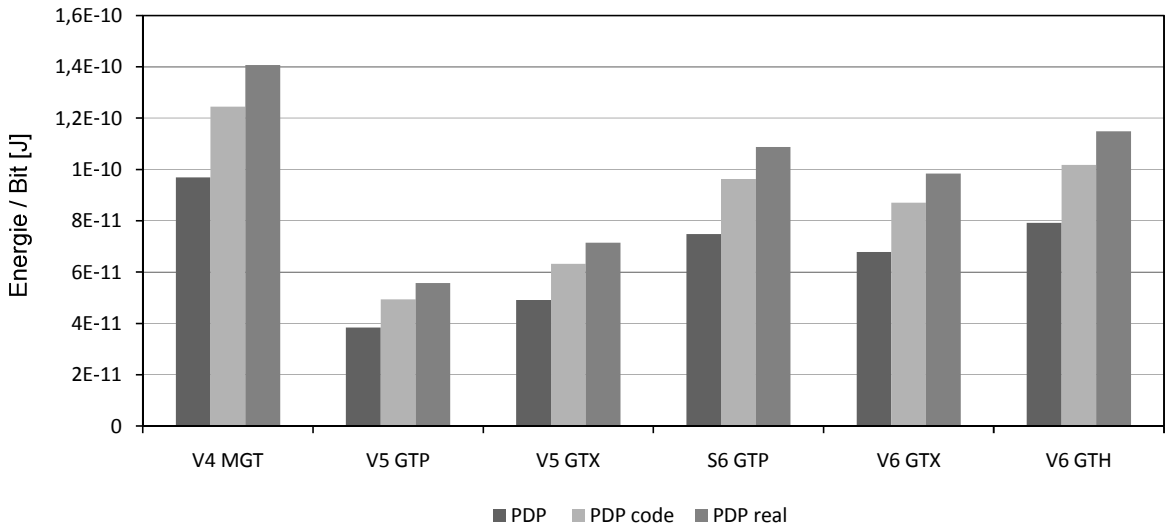

Abbildung 3.19: Grafische Übersicht der benötigten Energie von verschiedenen FPGA-Typen bei Implementierung von XAUI in Form von Energie pro Bit (vgl. Tabelle 3.16).

Die geringste benötigte Energie pro Bit liegt mit 55, 8 pJ/Bit bei der Implementierung mittels Virtex5-GTP-Transceivern, mit 140, 8 pJ/Bit liegt der Energiebedarf auf Basis von Virtex4-MGTs mehr als doppelt so hoch. Die ähnlich aufgebauten GTP-Transceiver des Spartan6 nehmen trotz modernerer Fertigungstechnologie mehr Leistung auf. Die ermittelten Werte für die einzelnen Spannungsversorgungen beziehen sich jeweils auf einen einzelnen Transceiver, je nach FPGA-Typ teilen sich mehrere Transceiver jedoch gemeinsame Funktionseinheiten. Deshalb entspricht der Gesamtwert nicht der Summe der Einzelwerte.

3.3.4 Ethernet über zentrale Bus-Leiterplatten und Twinax-Kabel

Um Daten mit einer Übertragungsrate von 10 Gbit/s bis 100 Gbit/s im Ethernet Format (siehe Kapitel 3.4) zu übertragen, gibt es je nach Anwendungsfall unterschiedliche Möglichkeiten. Im Falle von 10GBase-CX4, 10GSFP+Cu, 40GBASE-CR4 und 100GBASE-CR10 steht die kabelgebundene Datenübertragung im Vordergrund, während 10GBASE-KX4, 10GBASE-KR und 40GBASE-KR4 die Übertragung über eine Backplane (zentrale Bus-Leiterplatte) beschreiben. Alle Verfahren basieren auf XAUI (Kap. 3.3.3) zur Übertragung des Datenstroms. 10GBase-CX4, 40GBASE-CR4 und 100GBASE-CR10 nutzen als Übertragungskanal eine Twinax Anordnung in Form von einem Kabel, wie es auch bei 4x-InfiniBand verwendet wird, während 10GSFP+Cu auf sogenannte SFP+ Kabel (Small Form Factor Plugable) zurückgreift. 10GBASE-KX4 und 40GBASE-KR4 nutzen wiederum jeweils vier differentielle Leiterbahnen pro Richtung auf einer Backplane. 10GBASE-KR reduziert die Anzahl der benötigten Leiterbahnen auf eine differentielle Leitung pro Richtung. Um die Signalintegrität bei Verwendung des XAUI Verfahrens über die geforderte Kanallänge von 15 m bei 10GBASE-CX4 und 10GSFP+CU zu garantieren, müssen zwischen XAUI und Kabel entsprechende Signalaufbereiter verwendet werden, welche eine Taktrückgewinnung und 8B/10B Dekodierung vornehmen, die Lanes (logische Kanäle aus gebündelten Übertragungsstrecken) synchronisieren und die Daten nach einer 8B/10B Kodierung auf den Kanal geben. Bei 10GBASE-KR, 40GBASE-CR4, 40GBASE-KR4 und 100GBASE-CR10 wird statt einem 8B/10B Kodierer ein 64B/66B Verfahren eingesetzt. Die Verfahren weisen eine niedrige Latenz auf, da wenige Signalanpassungen zwischen MAC und PHY nötig sind. Tabelle 3.17 fasst die Eigenschaften der vorgestellten Standards zusammen. Bis auf zu einer Taktrate von 3, 125 Gbit/s pro Leitung können die Verfahren auf allen FPGA-Transceivern implementiert werden. Taktraten von 10, 3125 Gbit/s pro Leitung werden aufgrund der Taktrate nur von Virtex6-GTH-Tranceivern unterstützt. Tabelle 3.18 zeigt die Ergebnisse der Simulation.

Tabelle 3.17: Merkmale von 10GBase-CX4, 10GSFP+Cu und Backplane-Ethernet [R34] [R41] [R29].

	max. Länge	Signale gesamt	Kodierung	Taktrate	Bitrate (unidirektional)
10GBase-CX4	15 m	16	8B/10B	3,125 Gbit/s	12 Gbit/s
10GSFP+Cu	15 m	16	8B/10B	3,125 Gbit/s	12 Gbit/s
10GBASE-KX4	1 m	16	8B/10B	3,125 Gbit/s	12 Gbit/s
10GBASE-KR	1 m	16	64B/66B	10,3125 Gbit/s	10,3125 Gbit/s
40GBASE-CR4	7 m	16	64B/66B	10,3125 Gbit/s	41,25 Gbit/s
100GBASE-CR10	7 m	40	64B/66B	10,3125 Gbit/s	103,125 Gbit/s
40GBASE-KR4	1 m	16	64B/66B	10,3125 Gbit/s	41,25 Gbit/s

Tabelle 3.18: Leistungsaufnahme der betrachteten FPGAs bei Implementierung verschiedener Ethernet-Einheiten in Milliwatt (mW).

FPGA-Typ	Komponente	10G-CX4	10G-Cu	10G-KX4	10G-KR	40G-CR4	40G-KR4	100G-CR10
Virtex4 MGT	Kanal (mW)	440	478	520				
	Gesamt (mW)	**1183**	**1260**	**1326**				
Virtex5 GTP	Kanal (mW)	111	135	186				
	Gesamt (mW)	**372**	**468**	**672**				
Virtex5 GTX	Kanal (mW)	181	225	295				
	Gesamt (mW)	**576**	**749**	**1031**				
Spartan6 GTP	Kanal (mW)	220	275	372				
	Gesamt (mW)	**741**	**963**	**1351**				
Virtex6 GTX	Kanal (mW)	190	220	275				
	Gesamt (mW)	**761**	**879**	**1100**				
Virtex6 GTH	Kanal (mW)	398	412	466	564	455	477	421
	Gesamt (mW)	**1015**	**1072**	**1288**	**564**	**1820**	**1907**	**4208**

Die Verfahren nutzten eine unterschiedliche Anzahl an Übertragungskanälen, deshalb gilt:

$$PDP_{BPE} = \frac{P \cdot T_{Bit}}{N} \mid N = \begin{cases} 1 & \text{bei 10GBASE-KR} \\ 10 & \text{bei 100GBASE-CR10} \\ 4 & \text{sonst} \end{cases} \tag{3.27}$$

Wird die Leitungskodierung und das Ethernet-Paketformat berücksichtigt, so erhöht sich die benötigte Energie.

$$PDP_{BPE,code} = PDP_{BPE} \cdot X_L \cdot \frac{1542}{1500} \mid X_L = \begin{cases} \frac{10}{8} & \text{bei 10GBASE-CX4,Cu,KX4} \\ \frac{66}{64} & \text{sonst} \end{cases} \tag{3.28}$$

Dieser Term berücksichtigt die Leitungskodierung der betrachteten Verfahren, wobei die geringere benötigte Energie pro Bit der 10GBASE-KR-Kodierung ersichtlich wird. Zudem wird das Ethernet-Paketformat mit in die Berechnung einbezogen. Im praktischen Einsatz wird eine Paketauslastung von ca. 87 % erreicht [R11] [R14]. Entsprechend erhöht sich die benötigte Energie pro Bit.

$$PDP_{BPE,real} = PDP_{BPE,code} \cdot 1,13 \tag{3.29}$$

Wie zu erwarten ist, weisen die Virtex-4-basierten Implementierungen im Vergleich zu den anderen FPGAs die höchste Leistungsaufnahme auf, während GTP-Transceiver auf Virtex-5-Basis am wenigsten Energie benötigen (vgl. Tabelle 3.19 und Abbildung 3.20). Wie bei anderen Übertragungsverfahren zeigt sich der Vorteil einer erhöhten Taktrate gegenüber der Nutzung von vergrößerter Parallelität. Die Verfahren mit einer Bitrate von 10,3125 Gbit/s pro Leitung profitieren außerdem von der effektiveren Leitungskodierung im Gegensatz zu den anderen Varianten. Des Weiteren arbeiten die kabelgebundenen Verfahren effizienter als solche, die Backplanes zur Übertragung verwenden. Dies liegt unter anderem an den höheren elektrischen Verlusten auf Leiterbahnen im Vergleich zu Kabeln, bedingt durch den kleineren Leiterquerschnitt.

Tabelle 3.19: Benötigte Energie pro Bit verschiedener Ethernet-Varianten.

FPGA-Typ	PDP	10G-CX4	10G-Cu	10G-KX4	10G-KR	40G-CR4	40G-KR4	100G-CR10
V4 MGT	PDP_{BPE}	112 pJ	101 pJ	106 pJ				
	$PDP_{BPE,code}$	144 pJ	129 pJ	136 pJ				
	$PDP_{BPE,real}$	162 pJ	146 pJ	154 pJ				
V5 GTP	PDP_{BPE}	30 pJ	37 pJ	54 pJ				
	$PDP_{BPE,code}$	38 pJ	48 pJ	69 pJ				
	$PDP_{BPE,real}$	43 pJ	54 pJ	78 pJ				
V5 GTX	PDP_{BPE}	46 pJ	60 pJ	83 pJ				
	$PDP_{BPE,code}$	59 pJ	77 pJ	106 pJ				
	$PDP_{BPE,real}$	67 pJ	87 pJ	120 pJ				
S6 GTP	PDP_{BPE}	59 pJ	77 pJ	108 pJ				
	$PDP_{BPE,code}$	76 pJ	99 pJ	139 pJ				
	$PDP_{BPE,real}$	86 pJ	112 pJ	157 pJ				
V6 GTX	PDP_{BPE}	61 pJ	70 pJ	88 pJ				
	$PDP_{BPE,code}$	78 pJ	90 pJ	113 pJ				
	$PDP_{BPE,real}$	88 pJ	102 pJ	128 pJ				
V6 GTH	PDP_{BPE}	81 pJ	86 pJ	103 pJ	55 pJ	44 pJ	46 pJ	41 pJ
	$PDP_{BPE,code}$	104 pJ	110 pJ	132 pJ	57 pJ	46 pJ	48 pJ	43 pJ
	$PDP_{BPE,real}$	118 pJ	125 pJ	150 pJ	65 pJ	52 pJ	54 pJ	48 pJ

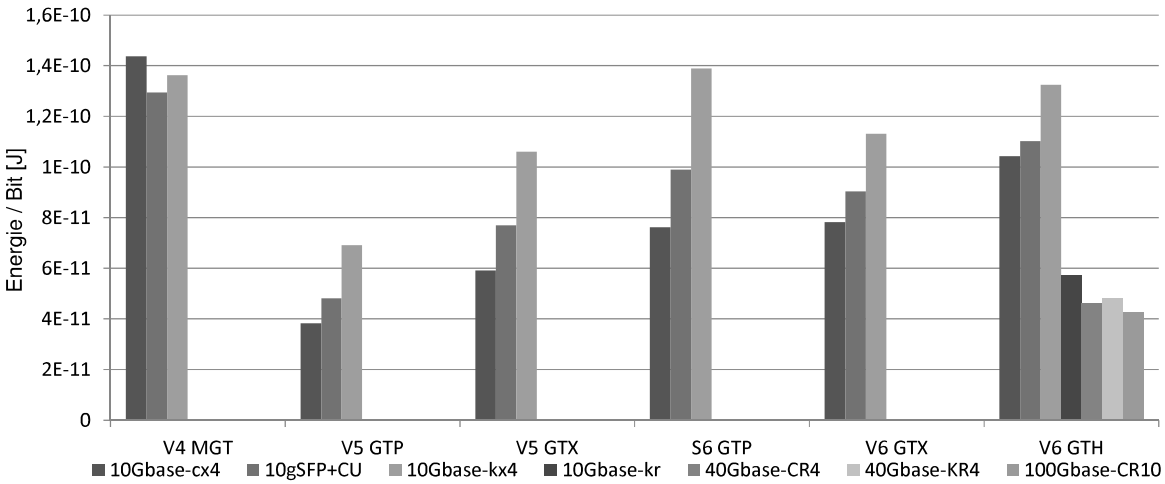

Abbildung 3.20: Grafische Übersicht der benötigten Energie (PDP_{code}) pro Bit verschiedener Ethernet-Einheiten.

3.3.5 PCIe - Peripheral Component Interconnect Express

PCIe [R15] ist ein serieller Übertragungsstandard und stellt den Nachfolger des PCI-Bus dar. Aufgrund der elektrischen Eigenschaften ist PCIe jedoch nicht mit PCI kompatibel. PCI-Express verwendet AC-gekoppelte, serielle und differentielle Punkt-zu-Punkt Übertragungskanäle mit differentieller Terminierung am Empfänger, basierend auf der CML-Spezifikation. Jeweils zwei dieser Kanäle (vier Leiter) bilden einen PCIe-Kanal, welcher je nach PCIe-Version unterschiedlich getaktet ist und unterschiedliche Kodierverfahren verwendet (siehe Tabelle 3.20). Geräte die PCIe nutzen können mehrere dieser Kanäle parallel verwenden, je nach Anforderung bis zu 32 Kanäle. Die Grundversion 1 der PCI-Express Spezifikation verwendet eine Bitrate von $2,5$ Gbit/s und nutzt eine 8B/10B-Kodierung zur Sicherstellung der Gleichstromfreiheit und Taktrückgewinnung. Dies führt zu einer Nutzdatenrate von 2 Gbit/s bzw. 250 MByte/s in jede Richtung. Version 2.0 erhöht die Bitrate auf 5 Gbit/s und erzielt unter Berücksichtigung der 8B/10B-Kodierung eine Nutzdatenrate von 500 MByte/s. Version 3 verwendet schließlich eine Bitrate von 8 Gbit/s und eine 128B/130B-Kodierung, um den Mehraufwand zu minimieren. Es ergibt sich eine Nutzdatenrate von 984 MByte/s. Zusätzlich zu den Datenleitungen wird bei PCIe ein differentielles Taktsignal mit 100 MHz für jede Richtung über den Übertragungskanal geführt. Gleichzeitig ist das Taktsignal auch im Datenstrom der anderen Leitungen kodiert.

Tabelle 3.20: Kenndaten der unterschiedlichen PCIe-Versionen.

PCIe Version	Bitrate	Kodierung	Datenrate	x16 Datenrate (unidirektional)
PCIe 1.x	$2,5$ Gbit/s	$8B/10B$	250 MByte/s	4 GByte/s
PCIe 2.0	5 Gbit/s	$8B/10B$	500 MByte/s	8 GByte/s
PCIe 3.0	8 Gbit/s	$128B/130B$	1000 MByte/s	16 GByte/s

Das Datenformat für eine PCIe-Übertragung ist in Abbildung 3.21 dargestellt. Zum Paketkopf gehören verschiedene Datenfelder wie 1 Byte Start-of-Frame, 2 Byte Sequenznummer und 15 Byte bis 20 Byte sonstige Paketkopfdaten. Die Nutzdaten können zwischen 0 Byte und 4096 Byte groß sein. Das CRC-Feld enthält 8 Byte CRC-Informationen und ein 1 Byte End-of-Frame Feld. Je kleiner die zu übertragende Menge an Daten ist, desto größer ist der Mehraufwand und umso ineffizienter wird die Übertragung. In einer PCIe-Übertragungstopologie muss immer ein Kommunikationspartner als Steuerknoten dienen. Dieser Knoten zeichnet sich dadurch aus, dass er ein differentielles Taktsignal mit einer Frequenz von 100 MHz ausgibt. Dieses Taktsignal kann zur Synchronisation verwendet werden und wird an alle anderen Kommunikationspartner weitergeleitet. Entsprechend wird dieses Taktsignal bei der Simulation beachtet und zusätzlich evaluiert. Die Stromversorgung des Taktsignals wird

getrennt ausgewertet, sodass eine Evaluierung der verschiedenen Arten von Kommunikationspartnern ermöglicht wird. PCIe ist in drei Versionen spezifiziert, die sich in der Datenrate und Leitungskodierung unterscheiden. PCIe 1.0 kann auf allen betrachteten FPGA-Typen evaluiert werden, während PCIe 2.0 aufgrund der hohen Datenrate nicht mehr auf GTP-Transceivern implementiert werden kann. Version 3.0 ist nur auf Virtex6-GTH-Transceivern zu realisieren. Alle Generationen von PCIe wurden jeweils in Versionen mit einer Anzahl von einem bis 16 parallelen Datenkanälen simuliert. Tabelle 3.21 zeigt die durchschnittliche Leistungsaufnahme der betrachteten FPGAs bei Implementierung einer PCIe-Kommunikationsstrecke für unterschiedliche Generationen und bei verschiedener Kanalanzahl.

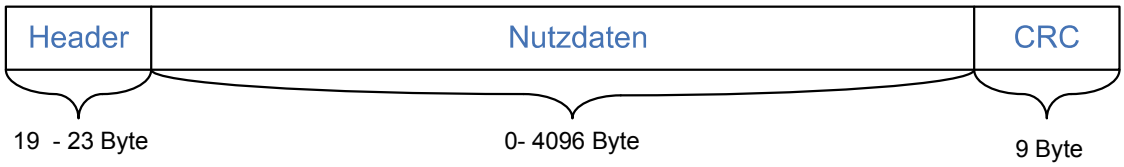

Abbildung 3.21: Genereller Paketaufbau aus Paketkopf, Nutzdaten und CRC bei PCI-Express [R15].

Tabelle 3.21: Leistungsaufnahme unterschiedlicher PCIe-Varianten bei Implementierung auf FPGA-Transceivern.

PCIe-Variante	Transceivertyp und Leistungsaufnahme in mW					
	V4-MGT	V5-GTP	V5-GTX	S6-GTP	V6-GTX	V6-GTH
1.0, 1x	724,7	244,5	326,9	590,3	371,4	710,6
1.0, 2x	851,7	337,5	430,5	816,5	577,4	903,8
1.0, 4x	1412,4	559,5	697,5	1338,0	989,4	1290,2
1.0, 8x	2537,0	1003,5	1231,7	2380,8	1813,3	2251,1
1.0, 16x	4784,0	1891,5	2299,9	4466,5	3461,1	4172,9
2.0, 1x	775,4	-	361,1	-	448,2	855,6
2.0, 2x	953,9	-	482,4	-	730,9	1099,7
2.0, 4x	1678,0	-	784,7	-	1296,4	1588,1
2.0, 8x	2946,0	-	1389,3	-	2427,3	2786,5
2.0, 16x	5601,2	-	2598,5	-	4689,0	5183,2
3.0, 1x	-	-	-	-	-	1051,1
3.0, 2x	-	-	-	-	-	1334,0
3.0, 4x	-	-	-	-	-	1899,8
3.0, 8x	-	-	-	-	-	3319,4
3.0, 16x	-	-	-	-	-	6158,9

Tabelle 3.22 und Abbildung 3.22 zeigen PDP_{PCIe}, welches sich aus Gleichung 3.30 berechnet.

$$PDP_{PCIe} = \frac{P \cdot T_{Bit}}{N} \mid N = 1, 2, 4, 8, 16 \tag{3.30}$$

PDP_{PCIe} beachtet nicht die den jeweiligen PCIe-Generationen zugrunde liegende Leitungskodierung (8B/10B oder 128B/130B). Dies gilt ebenso für den Einfluss des Paketformats, welches eine Übertragung von maximal 4096 Byte bei einem Protokoll-Mehraufwand von 32 Byte zulässt. Es gilt deshalb:

$$PDP_{PCIe,code} = PDP_{PCIe} \cdot X_L \cdot X_P \mid X_L = \begin{cases} \frac{10}{8} & \text{bei PCIe 1.0, 2.0} \\ \frac{130}{128} & \text{bei PCIe 3.0} \end{cases}, X_P = \frac{4128}{4096} \tag{3.31}$$

Im realen Einsatz gibt es Transaktionen, die nur Steuerinformationen beinhalten und keine Nutzdaten übertragen, oder es werden nicht alle Pakete voll ausgelastet. Deshalb kann PCIe eine realistische Auslastung von ca. 75 % der maximalen Kapazität erreichen [R53].

$$PDP_{PCIe,real} = PDP_{PCIe,code} \cdot 1,25 = PDP_{PCIe} \cdot X_L \cdot X_P \cdot 1,25 \tag{3.32}$$

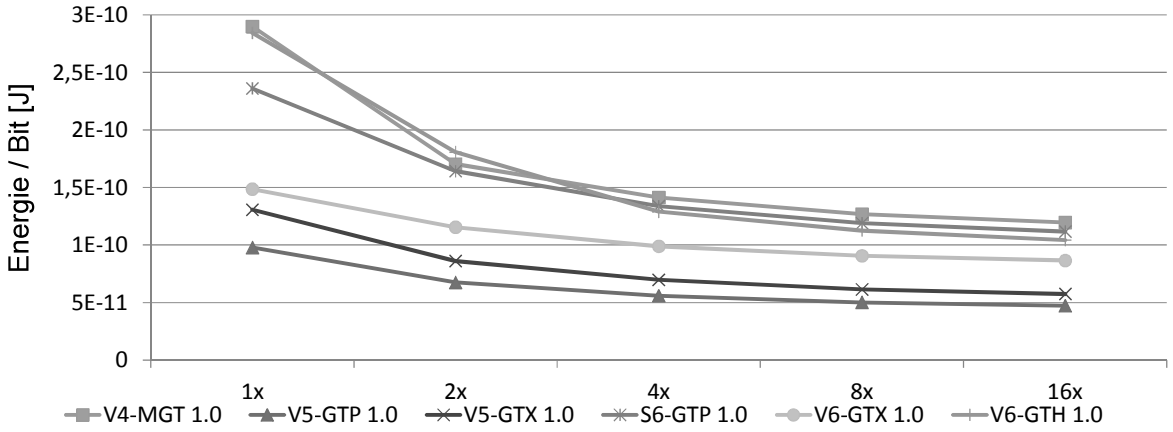

Abbildung 3.22: Die benötigte Energie pro Bit (PDP) einer PCIe 1.0-Implementierung auf verschiedenen FPGAs.

Die benötigte Energie pro Bit sinkt bei PCIe durch die Verwendung paralleler Übertragungskanäle. Dieser Effekt dominiert bei Verwendung weniger Transceiver. Dieses Verhalten lässt sich dadurch erklären, dass manche FPGA-Einheiten von mehreren Transceivern gemeinsam genutzt werden. Beispielsweise wird eine PLL im FPGA immer von zwei oder gleich vier Transceivern verwendet. Deshalb ist die Energieersparnis bei der Nutzung von zwei parallelen Transceivern anstatt einem einzelnen wesentlich

Tabelle 3.22: Energie pro Bit verschiedener PCIe-Varianten.

Transceiver-Typ	PDP-Typ	PCIe-Variante				
		1x	2x	4x	8x	16x
PCIe 1.0						
V4-MGT	PDP_{PCIe}	289 pJ	170 pJ	141 pJ	126 pJ	119 pJ
	$PDP_{PCIe,code}$	365 pJ	214 pJ	178 pJ	159 pJ	150 pJ
	$PDP_{PCIe,real}$	456 pJ	268 pJ	222 pJ	199 pJ	188 pJ
V5-GTP	PDP_{PCIe}	97 pJ	67 pJ	56 pJ	50 pJ	47 pJ
	$PDP_{PCIe,code}$	123 pJ	85 pJ	70 pJ	63 pJ	59 pJ
	$PDP_{PCIe,real}$	154 pJ	106 pJ	88 pJ	79 pJ	74 pJ
V5-GTX	PDP_{PCIe}	130 pJ	86 pJ	70 pJ	61 pJ	57 pJ
	$PDP_{PCIe,code}$	165 pJ	109 pJ	88 pJ	78 pJ	72 pJ
	$PDP_{PCIe,real}$	206 pJ	136 pJ	110 pJ	97 pJ	91 pJ
S6-GTP	PDP_{PCIe}	236 pJ	163 pJ	134 pJ	119 pJ	117 pJ
	$PDP_{PCIe,code}$	297 pJ	206 pJ	169 pJ	150 pJ	141 pJ
	$PDP_{PCIe,real}$	372 pJ	257 pJ	211 pJ	187 pJ	175 pJ
V6-GTX	PDP_{PCIe}	149 pJ	116 pJ	99 pJ	91 pJ	87 pJ
	$PDP_{PCIe,code}$	187 pJ	146 pJ	125 pJ	114 pJ	109 pJ
	$PDP_{PCIe,real}$	372 pJ	257 pJ	211 pJ	187 pJ	176 pJ
V6-GTH	PDP_{PCIe}	284 pJ	180 pJ	129 pJ	113 pJ	104 pJ
	$PDP_{PCIe,code}$	358 pJ	228 pJ	163 pJ	142 pJ	131 pJ
	$PDP_{PCIe,real}$	448 pJ	285 pJ	203 pJ	177 pJ	164 pJ
PCIe 2.0						
V4-MGT	PDP_{PCIe}	155 pJ	95 pJ	81 pJ	74 pJ	70 pJ
	$PDP_{PCIe,code}$	195 pJ	120 pJ	102 pJ	93 pJ	88 pJ
	$PDP_{PCIe,real}$	244 pJ	150 pJ	127 pJ	116 pJ	110 pJ
V5-GTX	PDP_{PCIe}	72 pJ	48 pJ	39 pJ	35 pJ	32 pJ
	$PDP_{PCIe,code}$	91 pJ	61 pJ	49 pJ	44 pJ	41 pJ
	$PDP_{PCIe,real}$	114 pJ	76 pJ	62 pJ	55 pJ	51 pJ
V6-GTX	PDP_{PCIe}	90 pJ	73 pJ	65 pJ	61 pJ	59 pJ
	$PDP_{PCIe,code}$	113 pJ	92 pJ	82 pJ	76 pJ	74 pJ
	$PDP_{PCIe,real}$	141 pJ	115 pJ	102 pJ	96 pJ	92 pJ
V6-GTH	PDP_{PCIe}	171 pJ	110 pJ	79 pJ	70 pJ	65 pJ
	$PDP_{PCIe,code}$	216 pJ	139 pJ	100 pJ	88 pJ	82 pJ
	$PDP_{PCIe,real}$	269 pJ	173 pJ	125 pJ	110 pJ	102 pJ
PCIe 3.0						
V6-GTH	PDP_{PCIe}	96 pJ	63 pJ	47 pJ	42 pJ	40 pJ
	$PDP_{PCIe,code}$	98 pJ	65 pJ	48 pJ	43 pJ	40 pJ
	$PDP_{PCIe,real}$	123 pJ	81 pJ	60 pJ	54 pJ	51 pJ

höher, als bei der Nutzung von 16 parallelen Transceivern statt acht parallelen Transceivern.

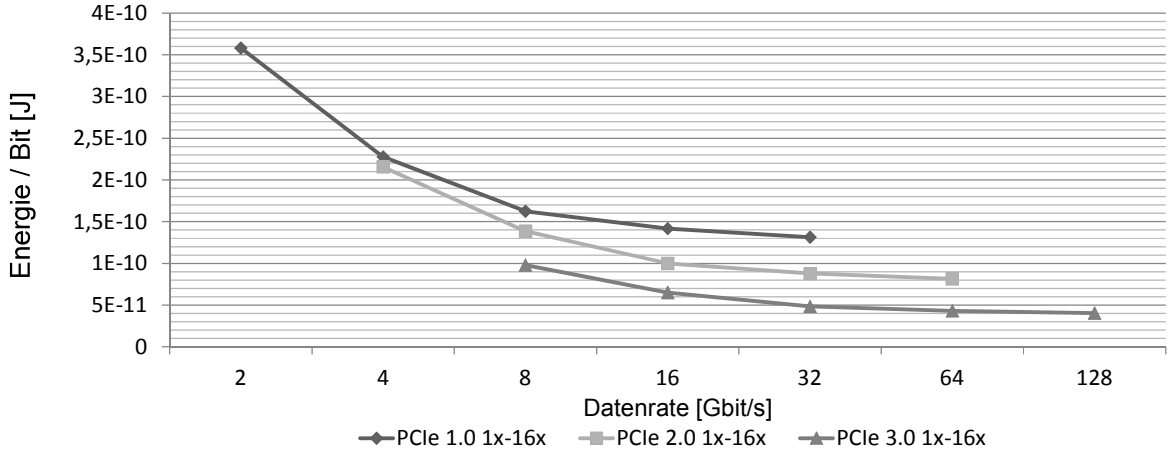

Abbildung 3.23: Die benötigte Energie pro Bit (PDP) unterschiedlicher PCIe-Versionen in Abhängigkeit der Datenrate, implementiert mit Virtex6-GTH-Transceivern.

Um einen Vergleich der benötigten Energie pro Bit bei unterschiedlich vielen Kanälen und Datenraten zu ermöglichen, wird die Realisierung aller PCIe-Varianten auf einem Transceivertyp betrachtet (vgl. Abbildung 3.23). Hierbei wird deutlich, dass eine Erhöhung der Übertragungsrate den Energiebedarf stärker senkt als eine entsprechende Parallelisierung. Dieser Effekt wird mit zunehmender Übertragungsfrequenz größer. Beispielsweise unterscheidet sich eine PCIe-Implementierung der Generation 1 mit zwei Transceivern Energiebedarf kaum von einer Implementierung der Generation 2 mit nur einem Transceiver. Beide weisen einen Durchsatz von 4 Gbit/s auf und benötigen hierfür ähnlich viel Energie. Deutlicher fällt der Unterschied bei einer Übertragungsrate von ca. 32 Gbit/s aus. Hier benötigt eine PCIe-Implementierung der Generation 3 mit vier Transceivern nur ca. 38 % der Energie einer PCIe-Implementierung der Generation 1 mit 16 Transceivern.

3.3.6 QPI - Quick Path Interconnect

Das von Intel entwickelte Quick Path Interconnect (QPI [R22]) Verfahren stellt den Nachfolger des FSB dar. Es dient primär zur Vernetzung mehrerer Prozessoren und Speichercontroller. Im Gegensatz zu FSB basiert QPI auf differentiellen Übertragungskanälen und verwendet Current Mode Logic (CML). Eine unidirektionale Verbindung wird als Link bezeichnet und kann aus bis zu 20 differentiellen Leiterpaaren bestehen. Sender und Empfänger sind bei QPI generell DC gekoppelt, müssen also über einen identischen Referenzpegel verfügen. Im Gegensatz zu den meisten anderen seriellen Übertragungsverfahren findet bei QPI keine Leitungskodierung des Datenstroms (z. B. 8B/10B) statt, die Datenrate entspricht also der Bitrate.

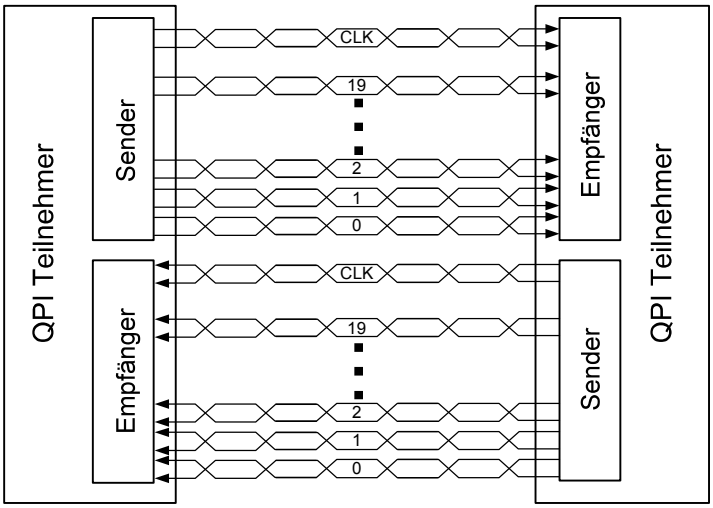

Abbildung 3.24: Aufbau einer bidirektionalen Datenverbindung im QPI-Verfahren. 20 differentielle Datenleitungen und eine differentielle Taktleitung bilden einen QPI-Link [R22].

Aus diesem Grund kann kein Taktsignal in den Datenstrom kodiert werden. Es muss daher ein dediziertes, differentielles Leiterpaar pro Richtung für die Taktübermittlung implementiert werden (siehe Abbildung 3.24). Das Taktsignal wird für die Datenrekonstruktion mit einer Frequenz entsprechend der halben Datenrate der anderen Leitungen im DDR-Verfahren betrieben. QPI verwendet zur Datenübertragung sogenannte Flits mit 80 Bit Länge. Diese Flits bestehen jeweils aus vier Phits mit 20 Bit Länge (siehe Abbildung 3.25). Ein Phit beinhaltet 16 Bit an Nutzdaten und 4 Bit Paketkopfdaten. Die Nutzdatenrate ist also um 20 % geringer als die Bitrate. Eine QPI-Verbindung mit 20 Links kann in einem DDR Transfer 20 Bit übertragen und benötigt für ein Flit entsprechend vier Transferzyklen. Die Taktrate bei QPI beträgt $2,4$ GHz bzw. $3,2$ GHz. Tabelle 3.23 veranschaulicht die sich ergebenden Datenraten bei einer QPI-Verbindung mit 20 Links. QPI verwendet verschiedene Techniken zur Signalformadaption, um die

Datenintegrität sicher zu stellen. Im Sender wird ein frequenzabhängiger Verstärker zur Vor- und Nachverzerrung eingesetzt, während im Empfänger ein frequenzabhängiger Verstärker zur Signalentzerrung verwendet wird.

Tabelle 3.23: Kenndaten der unterschiedlichen QPI-Versionen.

Taktfrequenz (Taktleitung)	Datenrate (20 Links)	Nutzdatenrate (unidirektional)	Nutzdatenrate (bidirektional)
2,4 GHz	12 GByte/s	9,6 GByte/s	19,2 GByte/s
3,2 GHz	12 GByte/s	12,8 GByte/s	25,6 GByte/s

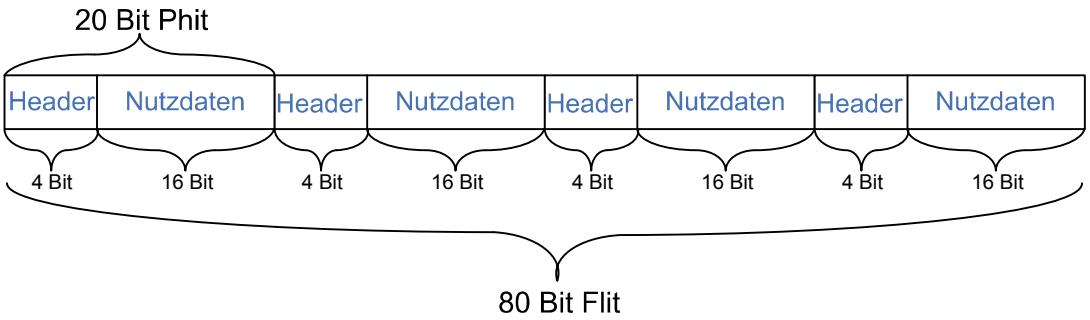

Abbildung 3.25: QPI verwendet zur Übertragung 80 Bit breite Flits die jeweils aus vier Phits bestehen [R22].

Da die QPI-Variante mit einer Übertragungsrate von 4,8 Gbit/s pro Kanal schon die Spezifikation aller Virtex-GTP-Transceiver übersteigt, kann QPI nur auf GTX- und GTH-Transceivern evaluiert werden. Tabelle 3.24 gibt die Leistungsaufnahme der untersuchten QPI-Varianten an.

QPI verwendet eine fixe Anzahl von 20 parallelen Kanälen. PDP_{QPI} ergibt sich zu:

$$PDP_{QPI} = \frac{P \cdot T_{Bit}}{20} \tag{3.33}$$

Bei QPI findet keine energieerhöhende Leitungskodierung statt, die Aufteilung von Daten in Flits und Phits wirkt sich jedoch negativ auf den Energiebedarf aus. Es können immer 16 Bit in einem Phit mit 4 Bit Steuerinformationen übertragen werden. Deshalb ergibt sich das PDP unter Beachtung des Protokolls zu:

$$PDP_{QPI,code} = PDP_{QPI} \cdot \frac{16+4}{16} = PDP_{QPI} \cdot 1,25 \tag{3.34}$$

Tabelle 3.24: Leistungsaufnahme der betrachteten FPGAs bei Implementierung einer QPI-Einheit. Die Einzelwerte beziehen sich jeweils auf einen einzelnen Transceiver, der Gesamtwert auf die Implementierung aller parallelen Kanäle.

		QPI 4,8		QPI 6,4	
FPGA-Typ	Spannungs-versorgung	Leistung Daten	Leistung Takt	Leistung Daten	Leistung Takt
Virtex4 MGT	VCC_{TX}	24,6 mW	24,8 mW	24,6 mW	24,5 mW
	VCC_{RX}	49,3 mW	49,7 mW	49,3 mW	49,1 mW
	$VCC_{TX-Term}$	114,6 mW	114,1 mW	114,9 mW	113,5 mW
	$VCC_{RX-Term}$	3,1 mW	3,1 mW	3,0 mW	3,2 mW
	VCC_{Int}	104,0 mW	104,0 mW	130,6 mW	130,4 mW
	Gesamt	**6001 mW**		**7375 mW**	
Virtex5 GTX	$VCC_{TX,RX}$	42,0 mW	42,3 mW	44,0 mW	44,5 mW
	$VCC_{TX-Term}$	34,0 mW	34,2 mW	35,0 mW	35,1 mW
	$VCC_{RX-Term}$	22,0 mW	22,6 mW	23,0 mW	23,3 mW
	VCC_{Int}	31,3 mW	31,3 mW	39,1 mW	39,1 mW
	VCC_{PLL}	59,8 mW	59,8 mW	75,0 mW	75,0 mW
	Gesamt	**3362 mW**		**3714 mW**	
Virtex6 GTX	$VCC_{TX,RX}$	99,4 mW	101,3 mW	127,0 mW	127,8 mW
	$VCC_{TX-Term}$	48,9 mW	49,9 mW	49,0 mW	49,9 mW
	$VCC_{RX-Term}$	39,4 mW	39,1 mW	39,4 mW	39,2 mW
	VCC_{Int}	50 mW	50 mW	62,5 mW	62,5 mW
	Gesamt	**5292 mW**		**6217 mW**	
Virtex6 GTH	VCC_{TX}	101,5 mW	102,0 mW	52,3 mW	52,6 mW
	VCC_{RX}	48,5 mW	48,6 mW	36,9 mW	37,0 mW
	VCC_{Term}	36,9 mW	37,0 mW	36,9 mW	37,0 mW
	VCC_{Int}	53,7 mW	53,9 mW	60,0 mW	60,2 mW
	VCC_{Int}	221,2 mW	221,6 mW	229,4 mW	230,0 mW
	Gesamt	**6382 mW**		**6876 mW**	

Diese Information gibt die theoretisch minimal benötigte Energie pro Bit an. In der Realität kann von einer benötigte Energie pro Bit von ca. 111 %, bezogen auf $PDP_{QPI,code}$ ausgegangen werden [R8].

$$PDP_{QPI,real} = PDP_{QPI,code} \cdot 1,11 = PDP_{QPI} \cdot 1,375 \tag{3.35}$$

Tabelle 3.25: Die benötigte Energie pro Bit von QPI, bei Implementierung mit unterschiedlichen FPGA-Transceivern.

	QPI 4,8		
FPGA-Typ	PDP_{QPI}	$PDP_{QPI,code}$	$PDP_{QPI,real}$
Virtex4 MGT	62,5 pJ	78,1 pJ	86,7 pJ
Virtex5 GTX	35,0 pJ	43,8 pJ	48,6 pJ
Virtex6 GTX	55,1 pJ	68,9 pJ	76,5 pJ
Virtex6 GTH	66,5 pJ	83,1 pJ	92,2 pJ
	QPI 6,4		
Virtex4 MGT	57,6 pJ	72,0 pJ	79,9 pJ
Virtex5 GTX	29,0 pJ	36,3 pJ	40,3 pJ
Virtex6 GTX	48,6 pJ	60,7 pJ	67,4 pJ
Virtex6 GTH	53,7 pJ	67,2 pJ	74,5 pJ

Wie die Evaluierung der durchschnittlichen Leistungsaufnahme gezeigt hat, weisen die Bewertungsmaße der verschiedenen Transceiver einen nicht so großen Unterschied auf wie andere Implementierungen (vgl. Tabelle 3.25 und Abbildung 3.26). Eine Ausnahme bildet hier der Virtex5-GTX-Transceiver, er benötigt fast die Hälfte der Energie verglichen mit Implementierungen auf Virtex4-MGT und Virtex6-GTH-Transceivern. Bei dem Vergleich der zwei QPI-Varianten bezüglich der benötigten Energie (vgl. Abbildung 3.27) zeigen Virtex4-MGT, Virtex5-GTX und Virtex6-GTX ein ähnliches Verhalten, die benötigte Energie pro Bit sinkt bei diesen drei Transceivern in gleichem Maße. Dies liegt an der jeweils gleichen Spezifikation bezüglich der maximalen Datenrate der Transceiver, welche nahe der Leistungsgrenze bezüglich der Datenrate betrieben werden. Hierin liegt auch der geringere Energiebedarf des Virtex6-GTH-Transceivers bei hohen Frequenzen begründet. Während QPI 4,8 den Transceiver in der unteren Hälfte der Spezifikation betreibt, liegt QPI 6,4 im oberen Drittel der Datenratenbereichs des Transceivers. Durch die Optimierung des Transceivers für diesen Leistungsbereich fällt der Energiebedarf deutlich geringer aus als beispielsweise bei einem Virtex4-MGT. Bei QPI 4,8 verhält es sich genau anders herum.

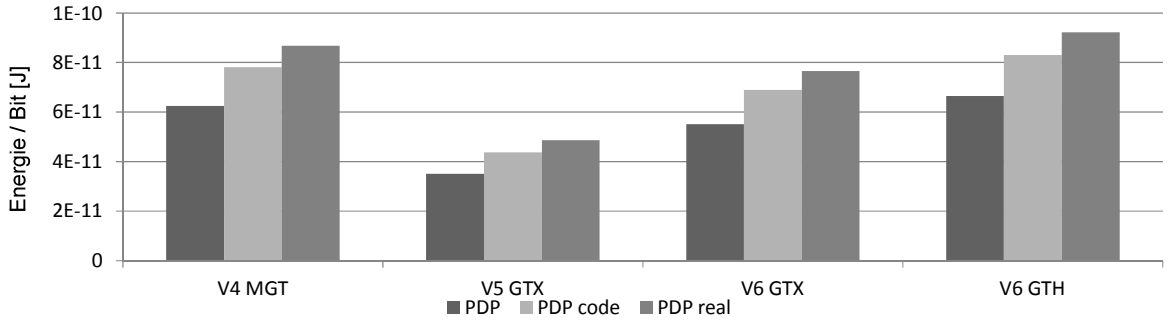

Abbildung 3.26: Die verschiedenen Energiemaße von QPI 4,8 bei Implementierung auf unterschiedlichen FPGAs.

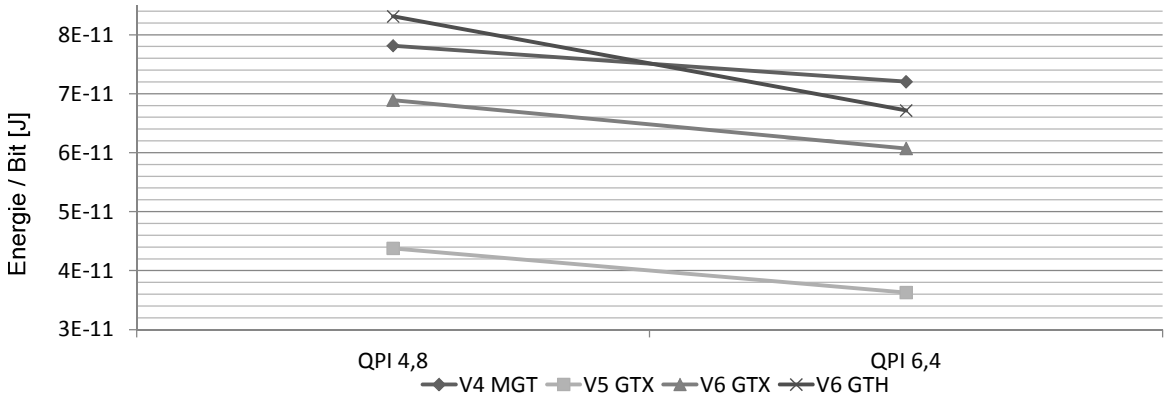

Abbildung 3.27: Die Änderung des Energiebedarfs bei einem Wechsel von QPI 4,8 auf QPI 6,4.

3.3.7 HyperTransport

HyperTransport [R33] ist eine serielle, differentielle Übertragungstechnik, um Daten zwischen mehreren Prozessoren oder Prozessoren und Peripherie auszutauschen. Hauptziel bei diesem Verfahren ist die latenzarme Übertragung von Daten, weshalb bei HyperTransport das Taktsignal dediziert über den Übertragungskanal geführt wird, anstatt es im Datenstrom zu kodieren. Hierdurch brauchen keine Kodierer und Dekodierer verwendet werden, dafür steigt die benötigte Anzahl an Leitungen. HyperTransport verwendet drei Arten von Signalen, wie in Tabelle 3.26 zu sehen ist. Steuerbefehle, Adressen und Daten werden über 2, 4, 8, 16 oder 32 differentielle Leitungen gesendet. Ein Steuersignal wird für jede Achtergruppe von Datenleitungen verwendet, um Daten und Adressen von Steuerbefehlen zu unterscheiden. Gleichermaßen wie bei den Steuersignalen wird pro Achtergruppe an Datenleitungen ein Taktsignal benötigt. Eine HyperTransport Übertragungsstrecke mit 1, 2, 4 oder 8 Leitungen besitzt jeweils ein differentielles Steuer- und Taktsignal. 16 Bit und 32 Bit breite Verbindungen benötigen entsprechend jeweils 2 bzw. 4 Steuer- und Taktsignale. Eine voll ausgebaute und bidirektionale HyperTransport Verbindung benötigt zur Datenübertragung 80 differentielle Leiterpaare bzw. 160 Leitungen (Steuersignale nicht betrachtet).

Tabelle 3.26: Wichtige Signalgruppen einer HyperTransport-Verbindung.

Signal	Bitbreite	Erklärung
CAD	2,4,8,16,32	Kommandos, Adressen und Daten.
CTL	1,2,4	Unterscheidet Kommandos von Adressen/Daten. Ein CTL Signal pro 8 CAD-Signale.
CLK	1,2,4	Taktsignale für CAD und CTL. Ein CLK Signal pro 8 CAD-Signale und ein CTL-Signal.

HyperTransport verwendet eine paketbasierte Übertragung, wobei zwischen Datenpaketen ($4 - 64$ Byte) und Kontrollpaketen (4 Byte oder 8 Byte) unterschieden wird. Beispielsweise besteht eine Leseanforderung aus einem 8 Byte großen Kontrollpaket, gefolgt von einem Datenpaket. Eine Schreibanforderung hingegen besteht aus einem 8 Byte großen Kontrollpaket, gefolgt von einem 4 Byte großen Antwort-Kontrollpaket, gefolgt von einem Datenpaket. Der Mehraufwand beträgt also maximal 12 Byte. Die Pakete werden immer zu Worten mit 32 Bit Länge auf die Achtergruppen der Datenleitungen verteilt und versendet. Entgegen der CML-Referenzpegel (Kapitel 3.11) verwendet HyperTransport eine differentielle Spannung von $1,2$ V und eine Common-Mode-Spannung von $0,6$ V. Eine differentielle On-Die-Terminierung von $100\,\Omega$ ist vorgeschrieben, die Kopplung zwischen Sender und Empfänger kann jedoch als AC oder DC Variante erfolgen. Die seriellen Leitungen werden im DDR-Verfahren getaktet (sie-

he Tabelle 3.27). Um Gleichstromfreiheit zu erreichen, kombiniert HyperTransport ein Scrambling-Verfahren mit einer 8B/10B Kodierung.

Tabelle 3.27: Taktraten bei HyperTransport und resultierende, unidirektionale Datenraten.

Taktrate	Bitrate	Datenrate	Datenrate (32 Bit Breite)
0,4 GHz	0,8 Gbit/s	0,08 GByte/s	2,56 GByte/s
0,8 GHz	1,6 Gbit/s	0,16 GByte/s	5,12 GByte/s
1,4 GHz	2,8 Gbit/s	0,28 GByte/s	8,96 GByte/s
2,6 GHz	5,2 Gbit/s	0,52 GByte/s	16,61 GByte/s
3,2 GHz	6,4 Gbit/s	0,61 GByte/s	20,48 GByte/s

Da HyperTransport neben den Nutzdaten auch den Takt und Kontrollsignale überträgt, müssen in der HPSICE-Simulation entsprechend viele parallele Transceiverinstanzen mit unterschiedlichen Stimuli implementiert werden. HyperTransport ist in fünf Varianten spezifiziert, welche sich in ihren Übertragungsraten unterscheiden. Diese reichen von 800 Mbit/s bis hin zu 6400 Mbit/s. Entsprechend können die leistungsfähigeren Versionen nur auf GTX- und GTH-Transceivern evaluiert werden. Tabelle 3.28 fasst die durchschnittliche Leistungsaufnahme der FPGAs bei unterschiedlichen Implementierungen von HyperTransport zusammen. Es ist jeweils nur die Gesamtleistungsaufnahme der Implementierung gezeigt. Eine ausführliche Auflistung aller Parameter und Simulationsergebnisse finden sich im Anhang dieser Arbeit.

Die benötigte Energie pro Bit ist durch Gleichung 3.36 gegeben, Tabelle 3.29 zeigt die ermittelten Werte.

$$PDP_{HT} = \frac{P \cdot T_{Bit}}{N} \mid N = 2, 4, 8, 16 \tag{3.36}$$

Da der Einfluss der Takt- und Steuersignale schon in der Leistungsaufnahme integriert ist, braucht er hier nicht extra betrachtet zu werden. PDP_{HT} berücksichtigt nicht die 8B/10B Leitungskodierung und das Datenprotokoll von HyperTransport. Das Datenprotokoll ermöglicht das Versenden von 64 Byte auf einem Kanal. Bei einer Schreibanforderung kommen 12 Byte Mehraufwand hinzu, bei einer Leseanforderung 8 Byte. Hier wird angenommen, dass Schreib- und Leseanforderungen gleichverteilt vorkommen, deshalb kann der durchschnittliche Mehraufwand mit 10 Byte angegeben werden. Durch diese Annahmen erhöht sich das PDP.

$$PDP_{HT,code} = PDP_{HT} \cdot \frac{10}{8} \cdot \frac{74}{64} \approx PDP_{HT} \cdot 1,445 \tag{3.37}$$

Tabelle 3.28: Leistungsaufnahme der betrachteten FPGAs bei Implementierung einer HyperTransport-Einheit mit unterschiedlicher Bitrate und Kanalanzahl.

HT-Variante	FPGA-Typ					
	V4-MGT	V5-GTP	V5-GTX	S6-GTP	V6-GTX	V6-GTH
HT 0.8 2x	621 mW	404 mW	549 mW	740 mW	526 mW	
HT 0.8 4x	869 mW	603 mW	801 mW	1111 mW	789 mW	
HT 0.8 8x	1365 mW	1001 mW	1305 mW	1853 mW	1315 mW	
HT 0.8 16x	2730 mW	2002 mW	2611 mW	3706 mW	2631 mW	
HT 1.6 2x	963 mW	399 mW	604 mW	789 mW	616 mW	1231 mW
HT 1.6 4x	1387 mW	599 mW	870 mW	1184 mW	924 mW	1590 mW
HT 1.6 8x	2233 mW	998 mW	1400 mW	1975 mW	1541 mW	2479 mW
HT 1.6 16x	4466 mW	1996 mW	2801 mW	3950 mW	3083 mW	4958 mW
HT 2.8 2x	1179 mW	444 mW	650 mW	885 mW	633 mW	1334 mW
HT 2.8 4x	1705 mW	666 mW	937 mW	1329 mW	949 mW	1720 mW
HT 2.8 8x	2757 mW	1110 mW	1512 mW	2216 mW	1582 mW	2679 mW
HT 2.8 16x	5514 mW	2220 mW	3024 mW	4432 mW	3165 mW	5358 mW
HT 5.2 2x	1543 mW		717 mW		603 mW	1636 mW
HT 5.2 4x	2251 mW		10357 mW		905 mW	2122 mW
HT 5.2 8x	3669 mW		16723 mW		1509 mW	3316 mW
HT 5.2 16x	7338 mW		33446 mW		3019 mW	6633 mW
HT 6.4 2x	1686 mW		755 mW		751 mW	1735 mW
HT 6.4 4x	2467 mW		1086 mW		1126 mW	2259 mW
HT 6.4 8x	4030 mW		1748 mW		1878 mW	3536 mW
HT 6.4 16x	8059 mW		3496 mW		3756 mW	7073 mW

$PDP_{HT,code}$ entspricht der theoretisch minimal benötigten Energie pro Bit von HyperTransport. Praktische Anwendungen zeigen eine Paketauslastung von ca. 82 % [R51], weshalb der Energiebedarf in realistischen Szenarien höher liegt.

$$PDP_{HT,real} = PDP_{HT,code} \cdot 1,18 \approx PDP_{HT} \cdot 1,705 \tag{3.38}$$

Tabelle 3.29 zeigt die benötigte Energie ($PDP_{HT,code}$) ausgewählter HyperTransport-Varianten, alle ermittelten Werte können im Anhang dieser Arbeite gefunden werden.

Tabelle 3.29: Die benötigte Energie pro Bit ($PDP_{HT,code}$) unterschiedlicher HyperTransportvarianten auf verschiedenen FPGAs.

	HT-Variante und $PDP_{HT,code}$					
FPGA-Typ	0,8 2x	0,8 16x	2,8 2x	2,8 16x	6,4 2x	6,4 16x
V4-MGT	561 pJ	308 pJ	304 pJ	178 pJ	190 pJ	114 pJ
V5-GTP	365 pJ	226 pJ	115 pJ	71,6 pJ		
V5-GTX	497 pJ	295 pJ	168 pJ	97,6 pJ	85,3 pJ	49,4 pJ
S6-GTP	669 pJ	419 pJ	229 pJ	143 pJ		
V6-GTX	475 pJ	297 pJ	163 pJ	102 pJ	84,8 pJ	53,0 pJ
V6-GTH			345 pJ	173 pJ	196 pJ	99,8 pJ

Der Energiebedarf sinkt bei PCIe durch die Verwendung paralleler Übertragungskanäle (vgl. Abbildung 3.28). Der Effekt ist besonders deutlich im Bereich der Verwendung weniger Transceiver. Dies liegt an der Nutzung von gemeinsamen Ressourcen für mehrere Transceiver. Der Energiebedarf lässt sich stärker verringern, indem eine HyperTransport-Variante mit höherer Übertragungsrate anstatt größerer Parallelität verwendet wird (vgl. Abbildung 3.29). Der Unterschied dominiert am stärksten im unteren Performanzbereich von HyperTransport und wird geringer bei Varianten mit einem höheren Datendurchsatz.

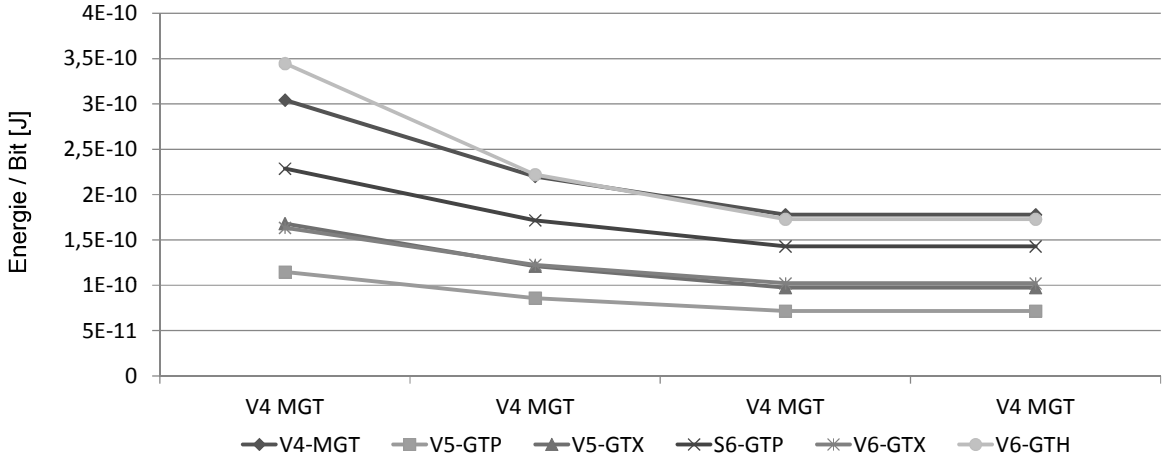

Abbildung 3.28: Der Energiebedarf von HyperTransport 2,8 in Abhängigkeit der Anzahl paralleler Kanäle.

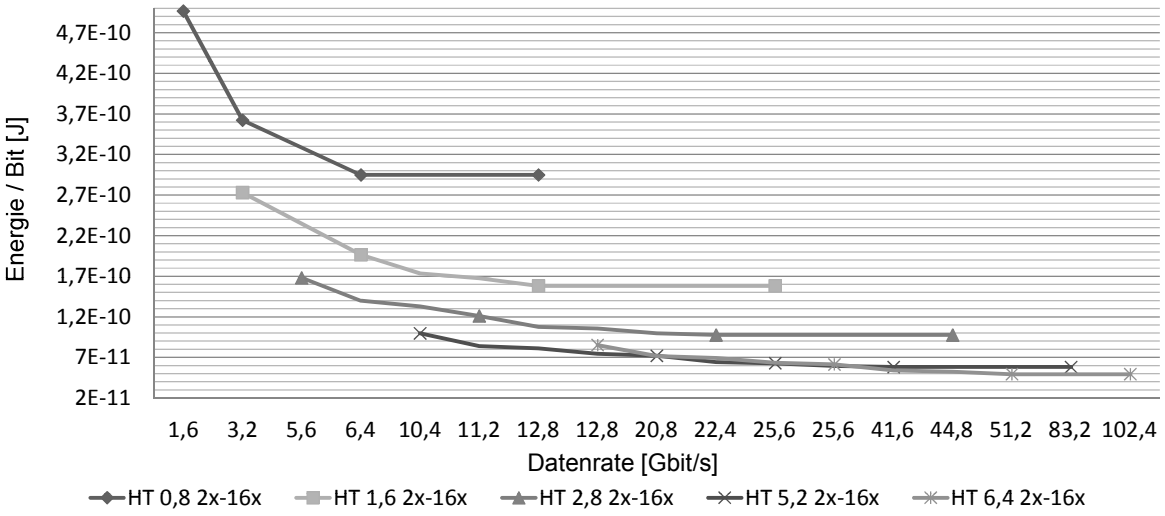

Abbildung 3.29: Der Energiebedarf verschiedener HyperTransport-Varianten in Abhängigkeit der zugehörigen Übertragungsrate auf Basis eines Virtex5-GTX-Transceivers.

3.3.8 Aurora

Aurora [R10] ist ein offener Standard zur Übertragung von Daten in Form von Punkt-zu-Punkt-Verbindungen auf Leiterplatten oder über Kabel. Ein einzelner Transceiver besitzt eine 64 Bit breite Benutzerschnittstelle, deren Daten zusammen mit dem Taktsignal über ein SerDes-Verfahren übertragen werden. Die Kopplung zwischen zwei Transceivern kann sowohl über eine AC- als auch über eine DC-Kopplung erfolgen. Ebenso kann der differentielle Spannungshub auf den Leitungen angepasst werden. Die Terminierung des Übertragungskanals erfolgt auf dem Baustein im Empfänger. Aurora ist universell konfigurierbar und implementiert Mechanismen zur Flusskontrolle und Taktsynchronisation. Es können zwischen einem und 24 Transceiver genutzt werden. Aurora nutzt die entsprechende Anzahl der Kanäle parallel und bündelt sie zu einem logischen Kanal. Der Mehraufwand für Steuer- und Synchronisationsbefehle ist sehr gering und liegt, je nach Anzahl der parallel genutzten Lanes, bei einer typischen Nutzdatenmenge von 4096 Bit zwischen 0,2 % und 1 %. Tabelle 3.30 zeigt die möglichen Datenraten verschiedener Aurora-Konfigurationen.

Tabelle 3.30: Unidirektionale Datenraten verschiedener Aurora-Konfigurationen.

Bitrate	Datenrate (ohne Kodier., 1 Lane)	Datenrate (8B/10B, 1 Lane)	Datenrate (8B/10B, 20 Lanes)
0,622 Gbit/s	0,08 GByte/s	0,06 GByte/s	1,24 GByte/s
1,25 Gbit/s	0,156 GByte/s	0,125 GByte/s	2,5 GByte/s
2,5 Gbit/s	0,3125 GByte/s	0,25 GByte/s	5 GByte/s
3,125 Gbit/s	0,39 GByte/s	0,3125 GByte/s	6,25 GByte/s
6,2 Gbit/s	0,8125 GByte/s	0,64 GByte/s	13 GByte/s

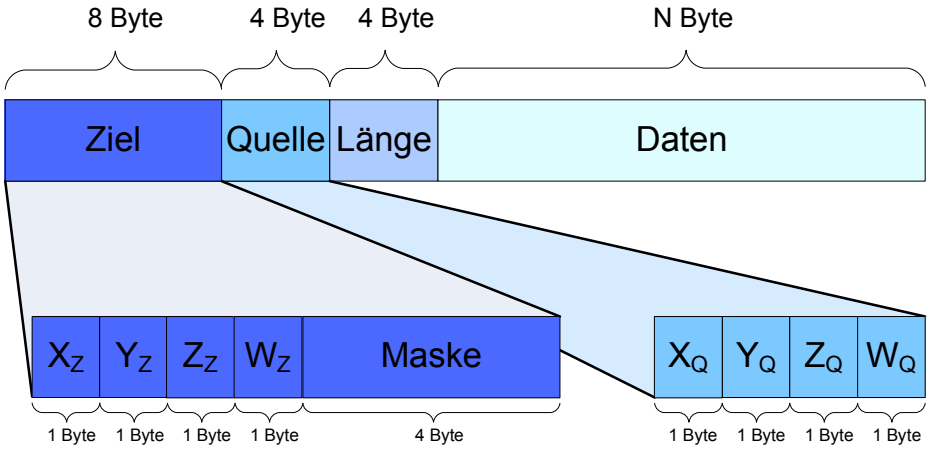

Abbildung 3.30: Schema des verwendeten Paketformats.

Um das Aurora-Protokoll effektiv nutzen zu können, wird ein eigens entwickeltes Protokoll verwendet [E3]. Für die Evaluierung des Aurora-Standards in dieser Arbeit ist hauptsächlich das Paketformat von Interesse (vgl. Abbildung 3.30). Die einzelnen Bestandteile eines Paketes belaufen sich auf:

- **Ziel:** 8 Byte bestehend aus:
  - X_Z : Gibt die X-Koordinate des Ziels in absoluter Schreibweise an (1 Byte $\Rightarrow 2^8 = 256$ Ziele in dieser Dimension adressierbar).
  - Y_Z : Gibt die Y-Koordinate des Ziels in absoluter Schreibweise an (1 Byte $\Rightarrow 2^8 = 256$ Ziele in dieser Dimension adressierbar).
  - Z_Z : Gibt die Z-Koordinate des Ziels in absoluter Schreibweise an. Wird nur bei Hyperwürfeltopologie ausgewertet (1 Byte $\Rightarrow 2^8 = 256$ Ziele in dieser Dimension adressierbar).
  - W_Z : Gibt die W-Koordinate des Ziels in absoluter Schreibweise an. Wird nur bei Hyperwürfeltopologie ausgewertet (1 Byte $\Rightarrow 2^8 = 256$ Ziele in dieser Dimension adressierbar).
  - Maske: Dient zum Maskieren mehrerer Ziele zwecks Multicast (4 Byte, 1 Byte für jede Dimension).
- **Quelle:** 4 Byte bestehend aus:
  - X_Q : Gibt die X-Koordinate der Quelle in absoluter Schreibweise an (1 Byte).
  - Y_Q : Gibt die Y-Koordinate der Quelle in absoluter Schreibweise an (1 Byte).
  - Z_Q : Gibt die Z-Koordinate der Quelle in absoluter Schreibweise an. Wird nur bei Hyperwürfeltopologie ausgewertet (1 Byte).
  - W_Q : Gibt die W-Koordinate der Quelle in absoluter Schreibweise an. Wird nur bei Hyperwürfeltopologie ausgewertet (1 Byte).
- **Länge:** Gibt die Menge der im Datenfeld folgenden Nutzdaten in Einheiten von 4 Byte Worten an. Es können in einem Paket $4 * 8 * 2^{32} = 4.294.947.296\text{Bit} \approx 4.29 * 10^9$ Worte der Länge 4 Byte, bzw. der Länge 32 Bit gesendet werden. Das entspricht einer Datenmenge von 4096 MByte oder 4 GByte pro Paket. Die minimale Menge an Nutzdaten beträgt 0 Byte. Die große mögliche Menge zu übertragender Daten wurde gewählt, um den sehr hohen Datendurchsatz der RocketIOs möglichst effizient ausnutzen zu können (siehe Kapitel 2).
- **Daten:** Nutzdaten

Als Übertragungskanal in der *HSPICE*-Simulation dient eine Twinax-Kabelanordnung von Samtec [R60] mit entsprechenden Steckern, welche als S-Parametermodell vorliegt

und eingebunden wird. Je nach verwendetem Transceivertyp können nicht alle Taktfrequenzen und damit nicht alle Übertragungsraten realisiert werden. Tabelle 3.31 zeigt die durchschnittliche Leistungsaufnahme der betrachteten FPGAs bei Implementierung einer Aurora-Kommunikationsstrecke mit unterschiedlicher Kanalanzahl und Taktrate.

Tabelle 3.31: Leistungsaufnahme selektierter Aurora-Varianten bei Implementierung auf FPGA-Transceivern.

Taktrate (Gbit/s) und Parallelität	Transceivertyp und Leistungsaufnahme in mW					
	V4MGT	V5GTP	V5GTX	S6GTP	V6GTX	V6GTH
1,25/1x	263	110	157	224	138	309
1,25/2x	418	185	255	378	277	474
1,25/16x	3348	1482	2037	3027	2215	3107
2,5 /1x	314	123	166	253	176	358
2,5 /2x	520	210	273	437	352	582
2,5 /16x	4160	1677	2184	3497	2817	3905
3,125/1x	351	136	186	269	195	369
3,125/2x	596	229	297	437	391	605
3,125/16x	4764	1833	2374	3497	3127	4073

Da Aurora mit mehreren Kanalbreiten evaluiert wurde, ergibt sich PDP_{Aurora} zu:

$$PDP_{Aurora} = \frac{P \cdot T_{Bit}}{N} \mid N = 1, 2, 4, 8, 16, 24 \tag{3.39}$$

Durch die hohe Effizienz des betrachteten Paketformats kann der Einfluss des Protokoll-Mehraufwands vernachlässigt werden. Es muss nur die Leitungskodierung beachtet werden.

$$PDP_{Aurora,code} = PDP_{Aurora,real} = PDP_{Aurora} \cdot \frac{10}{8} \tag{3.40}$$

Die sich ergebende Tabelle mit den verschiedenen Werten ist aufgrund ihres Umfanges im Anhang dieser Arbeit zu finden. Die Abbildungen 3.31 und 3.32 fassen stattdessen die Ergebnisse zusammen und lenken den Fokus auf die Abhängigkeit zwischen Energiebedarf und Takt-, bzw. Datenrate. Der Energiebedarf pro Bit sinkt mit zunehmender Datenrate, dieser Trend schwächt sich jedoch bei höheren Datenraten ab. Anders verhält es sich mit dem Einfluss paralleler Kanäle. Während ein Absinken des Energiebedarfs bei dem Wechsel von einem auf zwei Transceiver zu verzeichnen ist, können Varianten mit vier, acht, sechszehn oder vierundzwanzig Transceivern den Bedarf nicht weiter senken. Dies liegt an der Nutzung von gemeinsamen Ressourcen eines FPGAs für mehrere Transceiver. Da diese jedoch immer für zwei oder vier Transceiver gelten, bringt eine Parallelität mit entsprechenden Vielfachen dieser Werte keine Vorteile mit sich. Wenn also eine bestimmte Datenrate erreicht werden soll, so muss

zur Minimierung der benötigten Energie pro Bit, bei Datenraten über 2 Gbit/s immer die höchst mögliche Taktrate verwendet werden, gefolgt von der Nutzung paralleler Transceiver.

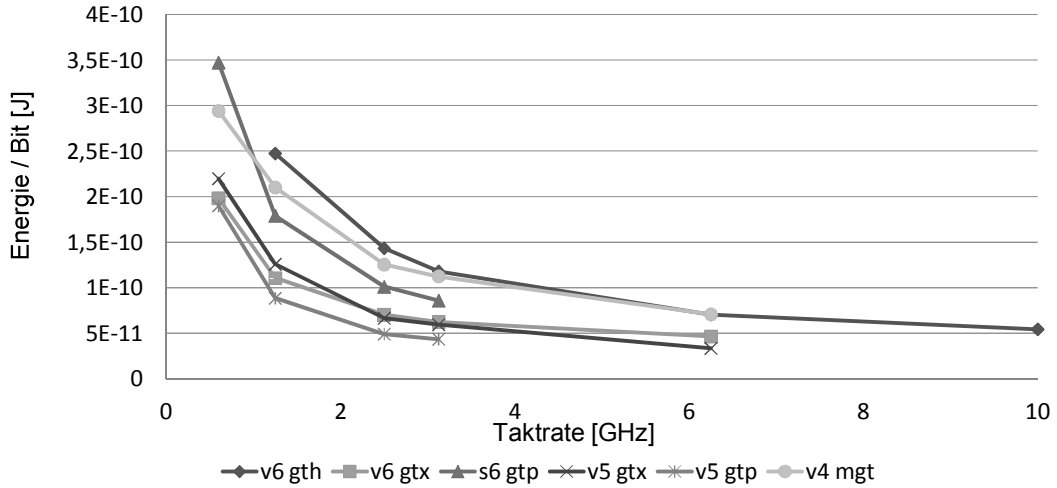

Abbildung 3.31: Die benötigte Energie pro Bit verschiedener Aurora-Varianten in Abhängigkeit der zugehörigen Taktrate.

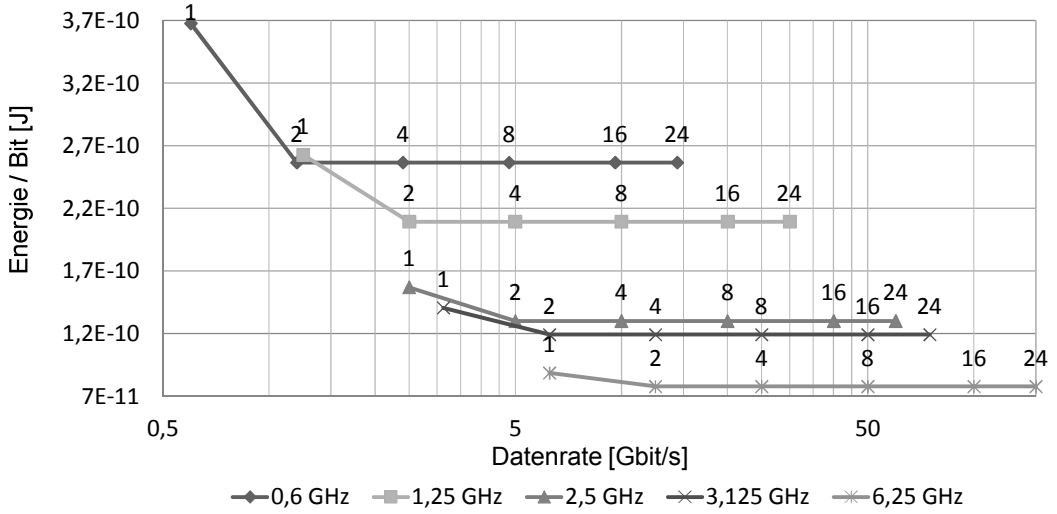

Abbildung 3.32: Die benötigte Energie pro Bit von Aurora, bei unterschiedlichen Taktraten und in Abhängigkeit der Parallelität bei Implementierung auf Virtex4-MGTs.

3.4 Multipegellogik Ethernet

Ethernet ist ein Standard für die Datenübertragung in lokalen Netzwerken. Die Kommunikation kann hierbei über Kabel, Glasfaser oder Funk erfolgen und sowohl als

Punkt-zu-Punkt-Verbindung oder busbasiert realisiert werden. Hier werden nur die geläufigsten kabelgebundenen Substandards betrachtet, welche alle Punkt-zu-Punkt-basiert sind. Die hier betrachteten Standards verwenden verdrillte Leiterpaare. Ethernet nutzt eine paketbasierte Übertragung mit einem Paketaufbau wie in Abbildung 3.33 zu sehen.

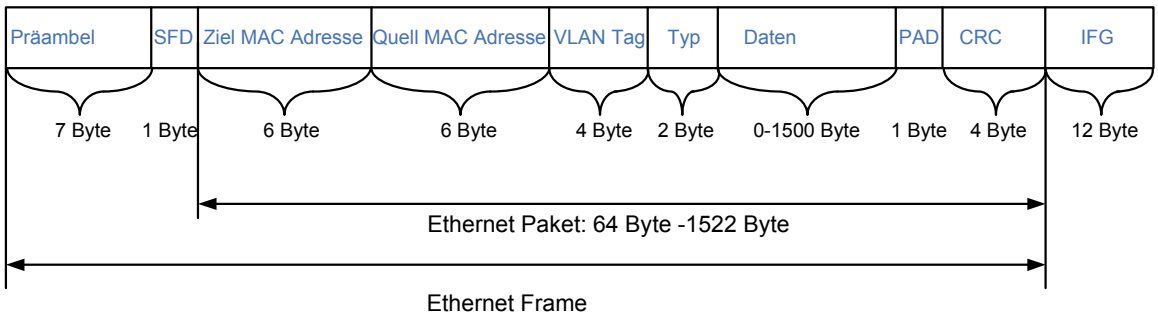

Abbildung 3.33: Der Aufbau eines Ethernet-Pakets [R63].

Ein sogenannter Ethernet-Frame stellt die über den Kanal zu transportierende Bitfolge dar und beinhaltet das eigentliche Ethernet-Datenpaket. Ein Frame beginnt mit der sogenannten Präambel und dem Start-of-Frame-Delimiter (SFD). Die Präambel stellt eine alternierende Bitfolge von 7 Byte Länge dar. Dies diente in älteren Ethernet Strukturen zur Bitsynchronisation, welche durch den 1 Byte langen SFD abgeschlossen wurde. Danach beginnt das Ethernet Paket mit den jeweils 6 Byte langen Adressen für Ziel und Quelle des Datentransfers. Ein sogenannter VLAN-Tag mit einer Länge von 4 Byte ermöglicht die Einteilung des Datenstromes in unterschiedliche virtuelle Netze, um beispielsweise sicherheitsrelevante Datenkommunikation zu kapseln. Dem VLAN-Tag folgen 2 Byte als Typ-Feld. Dieses gibt Auskunft über das verwendete Protokoll innerhalb der folgenden Nutzdaten, welche eine Länge von 64 Byte bis 1500 Byte aufweisen können. Das folgende PAD-Feld dient zur Auffüllung des Ethernet-Pakets auf eine minimale Größe von 64 Byte, was bei älteren Ethernet Varianten notwendig war. Das abschließende CRC-Feld (Cylic Redundancy Check) ist 4 Byte lang und beinhaltet einen 32 Bit CRC-Code zur Überprüfung der Framedatenintegrität. Zwischen zwei Frames müssen 12 Byte Abstand als sogenanntes Inter-Frame-Gap eingebracht werden. Der maximale Anteil von Nutzdaten des Ethernet-Formats bestimmt sich so zu:

$$Nutzdatenanteil = \frac{Nutzdaten}{Framegröße} = \frac{1500}{1542} = 97,28\,\% \tag{3.41}$$

Um die Bitfolge des Ethernet-Frames über einen Kanal zu übertragen, muss diese in einen physikalischen Datenstrom umgewandelt werden. Hierzu wird die Bitfolge vom Media Access Controller (MAC), welcher Aufgaben höherer Schichten bearbeitet, über eine medienunabhängige Schnittstelle (MII) zur physikalischen Schnittstelle

(PHY) geleitet. Dieses ist für die Umwandlung des Bitstroms in ein elektrisches Signal und dessen Übertragung zuständig. Nachfolgend, sowie in Kapitel 3.3.4, werden die meist verwendeten Ethernetverfahren vorgestellt. In diesem Kapitel liegt der Fokus auf sogenannter Multi-Pegel-Logik, also einer Datenübertragung mit Hilfe von mehr als zwei logischen Zuständen.

3.4.1 100BaseTX - Fast Ethernet over twisted Pair

Die Datenübertragung nach 100BaseTX - Fast Ethernet [R26] stellt heute das meist verwendete Verfahren zur Vernetzung von Rechnern mit einer Geschwindigkeit von 100 Mbit/s dar und wird langsam vom schnelleren 1000BaseT-Standard verdrängt. Fast Ethernet nutzt pro Richtung eine differentielle Punkt-zu-Punkt Übertragung über ein verdrilltes Leiterpaar der Kategorie 5, mit einer differentiellen Impedanz von 100 Ω. Diese Kabel ermöglichen die Kommunikation über Strecken von bis zu 100 m bei einer Signalfrequenz von maximal 100 MHz. Um die Kommunikation zwischen mehreren Knoten zu ermöglichen, werden Switche (Paketverteiler) eingesetzt, welche die Punkt-zu-Punkt Verbindungen anhand der Adressdaten im Paket umleiten. Ein Fast-Ethernet-Adapter besteht typischerweise aus mehreren funktionalen Einheiten wie in Abbildung 3.34 zu sehen. Über die Medienzugriffssteuerung (MAC) werden die Rohdaten entweder über eine medienunabhängige Schnittstelle (MII, Kapitel 3.2.2) mit 4 Bit Breite bei 25 MHz oder über eine reduzierte, medienunabhängige Schnittstelle (RMII, Kapitel 3.2.2) mit 2 Bit Breite bei 50 MHz an die physikalische Schnittstelle (PHY) gesendet. Diese Einheit beinhaltet mehrere Funktionen, um die Daten in ein elektrisches Signal zu transformieren. Die physikalische Kodierungsschicht (PCS - Physical Coding Sublayer) führt eine 4B/5B-Kodierung der Daten durch und sendet diese dann mittels eines NRZI-Leitungscodes an die physikalische Anschlusseinheit (PMA - Physical Medium Attachment). Der NRZI-Code führt einen Flankenwechsel auf dem Kanal herbei wenn eine logische Eins übertragen wird, und hält den aktuellen Pegel bei der Übertragung einer logischen Null. Durch die 4B/5B Kodierung steigt die Signalfrequenz in dieser Strecke auf 125 MHz. Die physikalische Anschlusseinheit formt anschließend den Datenstrom in einen differentiellen MLT-3 Leitungscode um und sendet ihn über den Kanal. MLT-3 durchläuft sequentiell die differentiellen Spannungspegel -1 V, 0 V, 1 V, 0 V. Der nächste Spannungspegel wird auf den Kanal gegeben, wenn eine logische Eins gesendet wird, bei einer logischen Null wird im aktuellen Pegel verblieben. Da vier Pegelwechsel für eine vollständige Sequenz benötigt werden, verringert sich die fundamentale Signalfrequenz auf ein Viertel der Bitrate. Es ergibt sich dadurch eine Signalfrequenz auf dem Kanal von 31, 25 MHz.

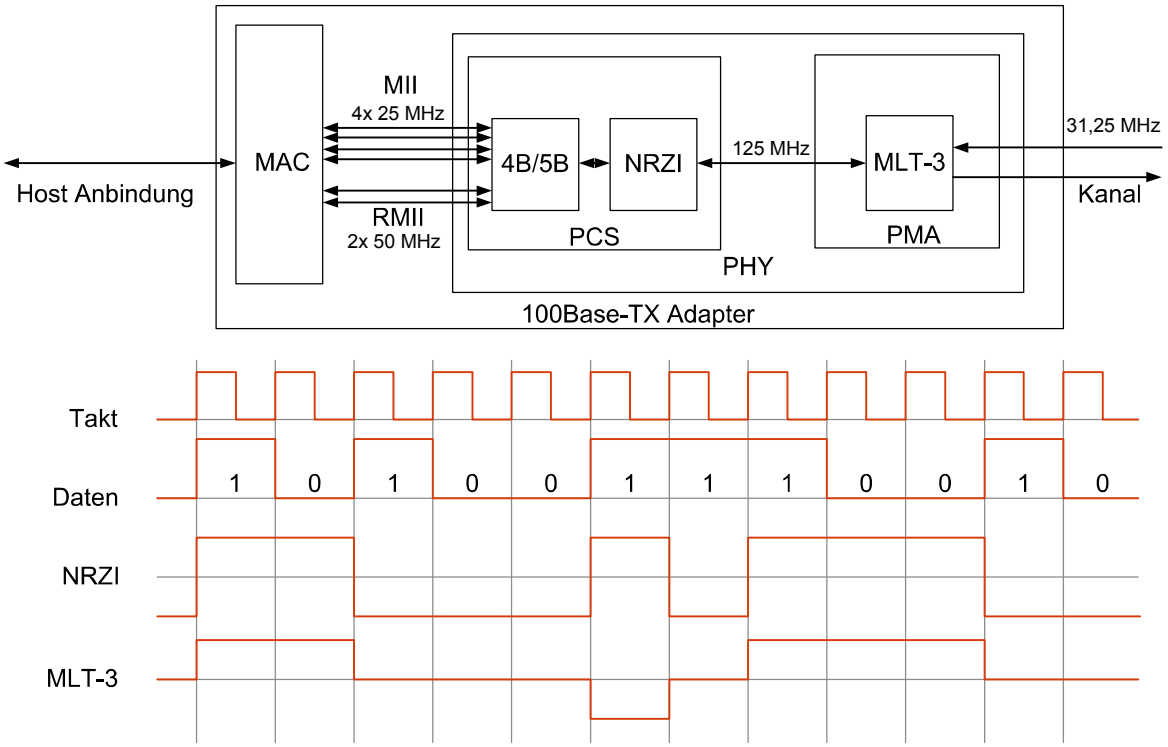

Abbildung 3.34: Der funktionale Aufbau eines Fast Ethernet Adapters mit Veranschaulichung der verwendeten Leitungscodes.

3.4.2 1000BaseT - Gigabit Ethernet over twisted Pair

Ähnlich wie 100BaseTX verwendet 1000BaseT [R26] verdrillte Leiterpaare mit einer differentiellen Impedanz von 100 Ω über Strecken von bis zu 100 m im Punkt-zu-Punkt-Verfahren. Im Gegensatz zu Fast-Ethernet, wird bei Gigabit Ethernet eine Datenrate von 1000 Mbit/s erreicht. Hierzu werden vier verdrillte Aderpaare eines Cat5-Kabels parallel im Vollduplexmodus betrieben.

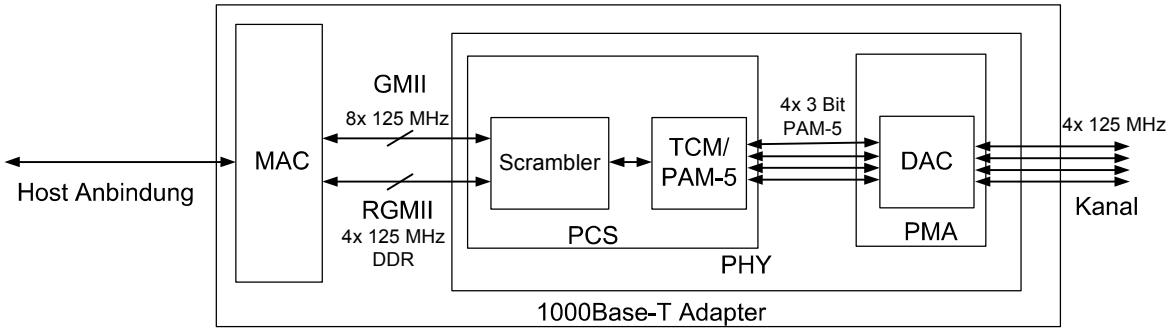

Abbildung 3.35: Der funktionale Aufbau eines Gigabit Ethernet Adapters.

Wie in Abbildung 3.35 zu sehen, kommuniziert die Medienzugriffssteuerung (MAC) bei 1000BaseT über eine medienunabhängige Schnittstelle für Gigabitanwendungen (GMII, Kapitel 3.2.2) mit 8 Bit Breite bei 125 MHz oder eine reduzierte medienunabhängige Schnittstelle für Gigabitanwendungen (RGMII, Kapitel 3.2.2) mit 4 Bit Breite bei 125 MHz im DDR-Verfahren mit der physikalischen Schnittstelle (PHY). Innerhalb der PHY werden die acht parallel empfangenen Bits innerhalb der physikalischen Kodierungsschicht (PCS - Physical Coding Sublayer) serialisiert und mittels eines Scramblers vermischt. Dies dient der tendenziell besseren Gleichstromfreiheit des entstehenden Datenstroms gegenüber den ursprünglichen Daten. Anschließend wird der Datenstrom in vier Blöcke mit jeweils 2 Bit geteilt und über eine Trellis-Codierung (TCM) mit jeweils einem Bit erweitert. Die Trellis-Kodierung erzeugt hierbei das dritte Bit durch eine Faltungskodierung, welches die zur späteren Fehlerkorrektur notwendigen Redundanzdaten enthält. Die Kodierung hängt hierbei nicht nur vom aktuellen Zustand ab, sondern auch von den vorherigen. Eine folgende Pulsamplitudenmodulation auf fünf Symbole legt die spätere Zuordnung der 3 Bit-Kombinationen auf die Leitungspegel fest. Aufgrund der Eigenschaften der TCM ändern sich diese Zuordnungen im laufenden Betrieb. Dadurch lässt sich für einen Zeitpunkt der Übertragung keine Zuordnung der Sendesymbole zu den Nutzdaten angeben. In der physikalischen Anschlusseinheit (PMA) werden die digitalen PAM5-Signale in Spannungspegel von 1 V, 0,5 V, 0 V, -0,5 V und -1 V umgesetzt und auf den Kanal gegeben. Jedes der vier Aderpaare wird mit 125 MHz betrieben, was durch die Übertragung von 3 Bit pro Signalpegel eine Bitrate von 1500 Mbit/s ergibt. Die tatsächliche Datenrate liegt

aufgrund der Trellis-Kodierung bei 1000 Mbit/s. Tabelle 3.32 zeigt eine beispielhafte PAM-5-Zuordnung von 3 Bit-Symbolen auf Signalpegel.

Tabelle 3.32: PAM-5-Kodierung von Symbolen auf Spannungspegel.

3 Bit-Symbol	Signalpegel auf Kanal
000	-1 V
001	0,5 V
010	0 V
011	1 V
100	-0,5 V
101	-1 V
110	0 V
111	0,5 V

Ethernet über verdrillte Leiterpaare setzt spezielle Transceiver voraus, welche von den betrachteten FPGAs nicht unterstützt werden. Deshalb wird für die Evaluierung von 100BaseTX und 1000BaseT ein *SPICE*-Modell eines kommerziell verfügbaren Ethernet-PHYs (Vitesse VSC8634) verwendet. Neben den Transceivern werden die benötigten Transformatoren (Magnetics) in der Simulation als Modell aus gekoppelten Spulen modelliert. Der Kanal in Form eines 10 m langen Kabels aus verdrillten Leiterpaaren wird über ein S-Parameter-Modell beschrieben. Obwohl Gigabit-Ethernet gegenüber Fast-Ethernet eine fast vierfach erhöhte Leistungsaufnahme aufweist (vgl. Tabelle 3.33), benötigt dieses Verfahren aufgrund der zehnfachen Datenrate wesentlich weniger Energie pro Bit. Bei Fast-Ethernet ergibt sich *PDP* aus dem Produkt zwischen Übertragungsfrequenz des Kanals und der Leistungsaufnahme, geteilt durch den Faktor Vier. Dieser Faktor liegt in der MLT-3-Kodierung begründet. Bei Gigabit-Ethernet wird das entsprechende Produkt durch den Faktor 3/4 dividiert. Dies liegt an der PAM-5-Kodierung und der Anzahl von vier parallelen Kanälen.

$$PDP_{100BaseTX} = \frac{P \cdot T_{Bit}}{4} \tag{3.42}$$

$$PDP_{1000BaseT} = \frac{P \cdot T_{Bit}}{3/4} \tag{3.43}$$

Ethernet ermöglicht die Übertragung von maximal 1500 Byte bei einem Mehraufwand von 42 Byte. Unter Berücksichtigung der Leitungskodierung erhöht sich die benötigte Energie pro Bit entsprechend.

$$PDP_{100BaseTX,code} = PDP_{100BaseTX} \cdot \frac{1542}{1500} \cdot \frac{5}{4} = PDP_{100BaseTX} \cdot 1,285 \tag{3.44}$$

$$PDP_{1000BaseT,code} = PDP_{1000BaseT} \cdot \frac{1542}{1500} \cdot \frac{3}{2} = PDP_{1000BaseT} \cdot 1,542 \tag{3.45}$$

Eine Paketauslastung von ca. 87 % kann im praktischen Einsatz erreicht werden [R11]. Entsprechend erhöht sich die benötigte Energie pro Bit.

$$PDP_{100BaseTX,real} = PDP_{100BaseTX,code} \cdot 1,13 \approx PDP_{100BaseTX} \cdot 1,452 \tag{3.46}$$

$$PDP_{1000BaseT,real} = PDP_{1000BaseT,code} \cdot 1,13 \approx PDP_{1000BaseT} \cdot 1,742 \tag{3.47}$$

Der Vergleich von Fast-Ethernet und Gigabit-Ethernet bescheinigt letzterem eine ca. dreifach niedrigere benötigte Energie pro Bit als Fast-Ethernet.

Tabelle 3.33: Leistungsaufnahme und benötigte Energie pro Bit bei Fast- und Gigabit-Ethernet.

Verfahren	Leistung Übertragung	Leistung Bereitschaft	PDP	PDP_{code}	PDP_{real}
100BaseTX	244,3 mW	14,2 mW	1954 pJ	2511 pJ	2838 pJ
1000BaseT	975,1 mW	20,4 mW	650 pJ	1002 pJ	1133 pJ

3.5 LVDS - Low Voltage Differential Signaling

LVDS [R37] steht für differentielle Signalübertragung mit niedriger Spannung und ist durch den EIA-644 Standard beschrieben. Im Gegensatz zu unsymmetrischen Datenübertragungsverfahren wie GTL, nutzt LVDS eine differentielle, symmetrische Technik zur Übertragung. Die differentielle Datenübertragung über symmetrische Kanäle dient der größeren Robustheit gegenüber Störungen. LVDS darf nicht mit anderen differentiellen Übertragungsverfahren verwechselt werden, welche oftmals umgangssprachlich als LVDS bezeichnet werden. Abbildung 3.36 zeigt den vereinfachten Aufbau eines LVDS Senders und entsprechenden Empfängers.

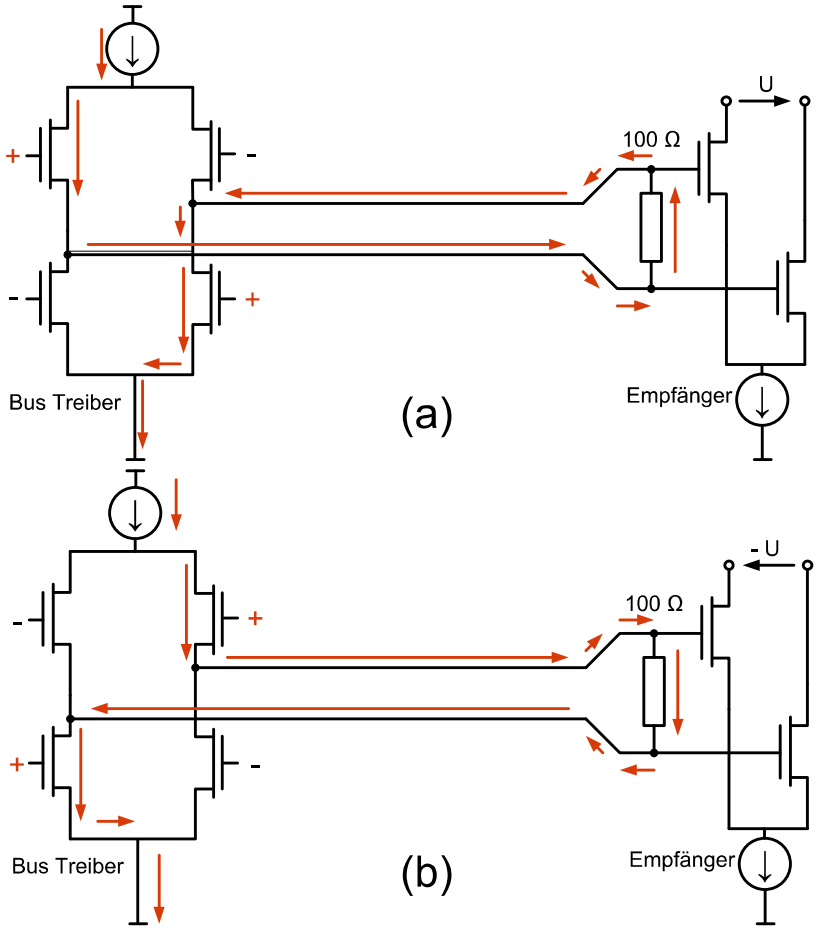

Abbildung 3.36: Differentielle Datenübertragung einer logischen Eins (a) und einer logischen Null (b) bei LVDS [R30].

Der Sender besteht aus einer Brückenschaltung von Transistoren und einer Konstantstromquelle. Wenn die Brücke wie in Abbildung 3.36 gezeigt geschaltet wird, so fließt der durch die Konstantstromquelle vorgegebene Strom durch das differentielle Leiterpaar und erzeugt am Terminierungswiderstand eine Spannung, welche vom Diffe-

renzverstärker im Empfänger detektiert wird. Nur die Virtex-5 FPGAs nutzen LVDS-Transceiver mit einer differentiellen Spannung von 350 mV bis 820 mV und einer differentiellen Terminierung von 100 Ω. Tabelle 3.34 fasst die wichtigsten Parameter der verwendeten LVDS-Transceiver zusammen.

Tabelle 3.34: LVDS-Parameter.

Parameter	Wert
maximale Signalfrequenz	612 MHz
maximale Datenrate	1,25 Gbit/s
Offsetspannung U_{OS}	1,25 V
Spannungshub	350 − 820 mV
Schwellwert	0,1 V
Terminierung	100 Ω
Treiberstrom	3,5 − 8,2 mA

LVDS wird in den betrachteten Implementierungen (siehe Kapitel 4) zur Realisierung von proprietären Übertragungsverfahren auf Basis von Virtex-5 FPGAs ohne Protokoll und Leitungskodierung verwendet. Deshalb findet hier keine Betrachtung dieser Einflüsse statt, sondern nur die Evaluierung der Leistungsaufnahme und der benötigten Energie pro Bit. Die verwendete Datenrate beträgt 1,25 Gbit/s bei maximal 820 mV differentiellem Spannungshub. Die durchschnittliche Leistungsaufnahme und benötigte Energie pro Bit einer einzelnen, unidirektionalen LVDS-Strecke ergeben sich zu:

- Leistungsaufnahme: 77 mW
- Benötigte Energie pro Bit: 61,6 pJ

Im Gegensatz zu den betrachteten CML-Transceivern teilen sich mehrere LVDS-Einheiten keine gemeinsamen Ressourcen. Die errechneten Werte skalieren also mit der Anzahl paralleler LVDS-Kanäle.

3.6 Technologiebasierter Vergleich von seriellen Transceivern und Übertragungskanälen

Durch anhaltenden Fortschritt in der technologischen Entwicklung können integrierte Schaltkreise mit zunehmend kleineren Strukturgrößen gefertigt werden. Die Verkleinerung der Strukturgrößen impliziert eine Verringerung der umzuladenden Kapazitäten. Hierdurch können im Vergleich zur Vorgängertechnologie die Betriebsparameter Frequenz, Spannung und Verlustleistung optimiert werden. So ist es beispielsweise bei gleichbleibender Spannung möglich, die maximale Schaltfrequenz zu erhöhen, oder bei gleichbleibender Frequenz, die benötigte Spannung zu senken. Bleibt die Spannung und die Taktfrequenz auf dem alten Wert, so verändert sich die Verlustleistung um den Faktor der Größenskalierung. Die betrachteten seriellen Transceiver werden in Größen von 90 nm (Virtex-4), 65 nm (Virtex-5), 45 nm (Spartan-6) und 40 nm (Virtex-6) gefertigt. Wenn alle Transceiver mit der gleichen Frequenz betrieben werden, so sollte die Leistungsaufnahme der Transceiver in ähnlicher Weise skalieren wie der Strukturgrößenfaktor. Zur Überprüfung der Annahme werden alle betrachteten Transceiver unter den gleichen Bedingungen evaluiert. Sender und Empfänger werden über einen verlustfreien Kanal verbunden und bei verschiedenen Taktraten betrieben. Das Ergebnis ist in Abbildung 3.37 zu sehen. Es zeigt sich, dass die allgemeinen Annahmen zur Skalierung von integrierten Schaltkreisen nicht auf die seriellen Transceiver der betrachteten FPGAs angewendet werden können. Wird die Leistungsaufnahme aller Transceiverkomponenten betrachtet, so weisen die GTH-Transceiver des Virtex-6 FPGAs die höchste Verlustleistung auf, obwohl sie in der kleinsten Strukturgröße gefertigt werden. Ebenso benötigt ein Spartan-6-Transceiver in 45 nm mehr Energie als ein Virtex-5-Transceiver in 65 nm. Einen großen Einfluss auf die Verlustleistung haben die integrierten Komponenten eines Transceivers, welche über dedizierte Versorgungsleitungen mit Spannung versorgt werden. Hierzu gehören beispielsweise die Kodierer und die PLL. Der Anteil dieser Komponenten (gestrichelte Linien in Abbildung 3.37) an der Verlustleistung unterscheidet sich stark zwischen den Transceivertypen. Sehr deutlich sticht dieser Anteil bei dem GTH-Transceiver eines Virtex-6 FPGAs heraus. Er erzeugt unabhängig von der Übertragungsrate die Hälfte der Gesamtverlustleistung. Eine Erklärung für dieses Verhalten kann im großen Übertragungsbereich des Transceivers bis über 10 Gbit/s liegen, da der Transceiver anstatt auf Energieeffizienz für hohe Datenraten optimiert wurde. Bei dem MGT des Virtex-4 FPGAs liegen beide Verlustleistungsanteile bei niedrigen Frequenzen nahe beieinander und steigen dann mit zunehmender Übertragungsrate an. Bei Transceivern der Virtex-5 und Spartan-6 FPGAs weisen die Treiberstufen und die Terminierung im Gegensatz zu den anderen FPGAs kaum eine Frequenzabhängigkeit auf. Sowohl bei der älteren Technologie der

MGTs als auch bei der neueren Technologie der GTH-Transceiver ist eine deutliche Frequenzabhängigkeit zu sehen.

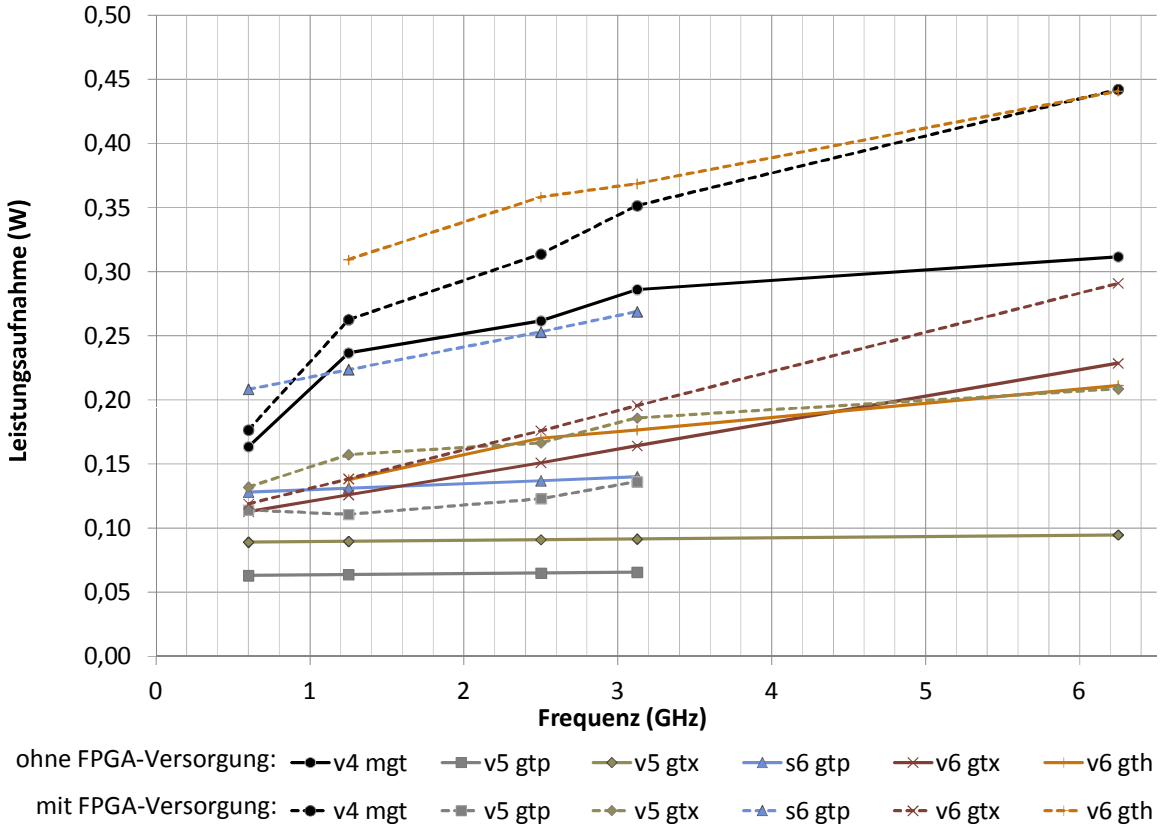

Abbildung 3.37

Neben den Transceivern bildet der Übertragungskanal einen wichtigen Teil der Datenübertragungsstrecke. Zusätzlich zu den theoretischen Betrachtung zum Einfluss des Kanals auf das Signal (siehe Kapitel 2.1) sollen die Kanalverluste für verschiedene Übertragungsverfahren empirisch ermittelt werden. Hierzu wird eine Übertragungsstrecke in der *SPICE*-Simulation einmal mit Kanal zwischen Sender und Empfänger und einmal ohne Kanal betrachtet. Es zeigt sich im Ergebnis eine Veränderung der Leistungsaufnahme der Kanalterminierung, der Energiebedarf der Sende- und Empfangsstufen bleibt hingegen unverändert. Aus der Differenz des Energiebedarfs für die Terminierung kann so auf die Kanalverluste geschlossen werden. Tabelle 3.35 zeigt die Leistungsaufnahme der Terminierung bei unterschiedlichen Übertragungsraten mit und ohne Kanal, es handelt sich dabei um eine Aurora Übertragungsstrecke mit Virtex-5 GTX-Transceiver.

Tabelle 3.35: Die Kanalverluste einer Aurora Übertragungsstrecke auf Virtex-5 GTX-Transceivern mit 50 cm Länge.

Übertragungsrate	P_{Term}, mit Kanal	P_{Term}, ohne Kanal	Kanalverluste
1,25 Gbit/s	33 mW	20,7 mW	12,3 mW
2,5 Gbit/s	33 mW	20,7 mW	12,3 mW
3,125 Gbit/s	33 mW	20,7 mW	12,3 mW
5 Gbit/s	33 mW	20,7 mW	12,3 mW

Wie Tabelle 3.35 zeigt, sind die Kanalverluste nicht abhängig von der Übertragungsrate der Signale. Dieses Ergebnis deckt sich sehr gut mit den Erkenntnissen aus Kapitel 2.1. Das zu übertragende Signal kann im Wesentlichen als Rechteckimpuls beschrieben werden, dessen Form und Frequenzspektrum unabhängig von der Übertragungsrate ist. Das Frequenzspektrum des Signals wird durch eine Si-Funktion ($\frac{sin(x)}{x}$)beschrieben, hohe Frequenzanteile fallen also schnell ab. Deswegen werden die Kanalverluste bei einer Anregung des Kanals mit einem Rechtecksignal durch den resistiven Anteil dominiert. Der resistive Anteil steigt linear mit der Länge des Kanals, was in Abbildung 3.38 zu sehen ist. In der Abbildung ist die Leistungsaufnahme der Kanalterminierung bei einer Aurora Übertragungsstrecke mit variabler Länge zu sehen. Die mit der Kanallänge zunehmende Amplitudendämpfung des Signals sorgt für einen erhöhten Stromfluss durch die Terminierungswiderstände am Empfänger. Dies liegt am größer werdenden Potentialunterschied zwischen Terminierungsspannung und Signalpegel.

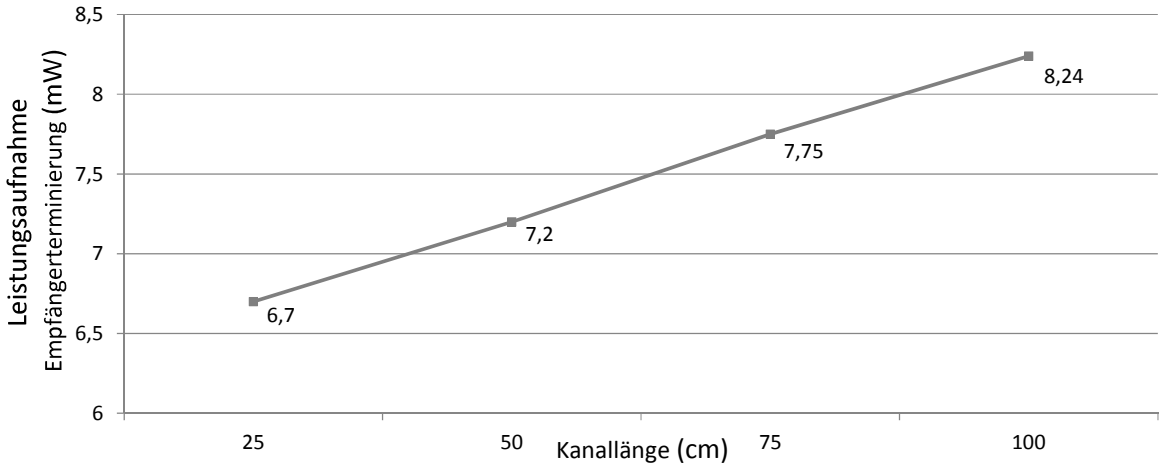

Abbildung 3.38: Verlustleistung der Kanalterminierung einer Aurora Übertragungsstrecke bei 2,5 Gbit/s und einer Länge von 25 cm, 50 cm, 75 cm und 100 cm.

Tabelle 3.36 zeigt die ermittelten Kanalverluste der betrachteten Verfahren. Zwecks besserer Vergleichbarkeit wurden die seriellen Verfahren bei ähnlichen Datenraten betrachtet. Die in mW angegebenen Verluste beziehen sich auf die akkumulierten Verlus-

te in allen parallelen Übertragungskanälen. Die vierte Spalte gibt an, wie viel Energie pro übertragenem Bit im Kanal in Wärme umgesetzt wird. In der letzten Spalte ist der Anteil der Kanalverluste an der Gesamtverlustleistung zu sehen. Bei der Vielzahl der Übertragungsverfahren liegen die Kanalverluste bei weniger als 10 % der gesamten Leistungsaufnahme. Der Anteil der Kanalverluste steigt mit zunehmender Anzahl von parallel genutzten Datenleitungen an. Auch zusätzliche Leitungen wie z. B. Taktleitungen erhöhen die Kanalverluste, bezogen auf die gesamte Implementierung. Dies ist bei einem Vergleich von Infiniband, PCI-Express und Aurora zu beobachten. PCI-Express verwendet als einiges der drei Verfahren zusätzliche Taktsignale, welche weitere Verluste hervorrufen. Zudem sind die Gleichspannungsverluste in den kabelgebundenen Verfahren wie Infiniband und Aurora aufgrund des größeren Leiterquerschnitts geringer als bei leiterplattenbasierten Übertragungsstrecken.

Tabelle 3.36: Kanalverluste der betrachteten Übertragungsverfahren.

Übertragungs-verfahren	Bitbreite Datenleit.	Kanalverluste (mW)	Kanalverluste (pJ/Bit)	Anteil an Gesamt-verlusten(%)
PCI	32	1,2 mW	0,3 pJ	0,5 %
PCI-X 66	64	2,5 mW	0,6 pJ	0,3 %
PCI-X 133	64	2,5 mW	0,3 pJ	0,2 %
FSB 200	64	109 mW	8,4 pJ	7,9 %
FSB 400	64	109 mW	4,3 pJ	7,2 %
FSB 667	64	109 mW	2,5 pJ	6,4 %
FSB 800	64	109 mW	2,1 pJ	6,2 %
MII	4	0,04 mW	0,4 pJ	0,08 %
RMII	2	0,025 mW	0,25 pJ	0,08 %
GMII	8	0,3 mW	0,3 pJ	0,3 %
RGMII	4	0,2 mW	0,3 pJ	0,5 %
XGMII	32	2,4 mW	0,2 pJ	0,6 %
Infiniband SDR	1	9,3 mW	3,7 pJ	5,7 %
	4	37,2 mW	3,7 pJ	7 %
	12	112 mW	3,7 pJ	7 %
SGMII	1	20 mW	15,8 pJ	4,2 %
XAUI	4	55,2 mW	4,4 pJ	8,9 %
10GBase-CX4	4	36,4 mW	2,9 pJ	6,3 %
10GSFP+Cu	4	64 mW	5,1 pJ	8,5 %
10GBASE-KX4	4	138 mW	11 pJ	13,4 %
10GBASE-KR	4	20,3 mW	2 pJ	3,6 %
PCIe 1.0	1	10,3 mW	4,1 pJ	3,1 %
	2	20,6 mW	4,0 pJ	4,7 %
	4	41,2 mW	4,3 pJ	6,1 %
	8	82,4 mW	4,1 pJ	6,6 %
	16	164,8 mW	4,1 pJ	7,1 %
QPI	20	321 mW	3 pJ	8,6 %
HyperTrans. 2,8	2	28,6 mW	3,0 pJ	4,4 %
	4	57,2 mW	4,9 pJ	6,1 %
	8	114 mW	3,0 pJ	7,5 %
	16	229 mW	3,0 pJ	7,6 %
Aurora 2,5	1	12,3 mW	3,9 pJ	6,6 %
	2	24,6 mW	3,9 pJ	8,3 %
	4	49,2 mW	3,9 pJ	8,3 %
	8	98,4 mW	3,9 pJ	8,3 %
	16	196 mW	3,9 pJ	8,3 %
	24	295 mW	3,9 pJ	8,3 %
100BaseTX	1	0,8 mW	6,4 pJ	0,3 %
1000BaseT	4	3,6 mW	2,4 pJ	0,4 %

4 Standardübergreifende Evaluierung

In diesem Kapitel sollen die Ergebnisse aus Kapitel 3 zusammengefasst und die betrachteten Verfahren miteinander verglichen werden. Neben einer Bewertung aller Verfahren werden topologieinterne Evaluationen mit Unterscheidung von seriellen und busbasierten Strukturen durchgeführt. Außerdem werden Verfahren zu Intra- und Intersystemkommunikation miteinander verglichen.

4.1 Allgemeine Gegenüberstellung der Übertragungsstandards

Um einen standardübergreifenden Vergleich der betrachteten Evaluationsgrößen zu ermöglichen, muss die Varianz zwischen den Transceivertypen bezüglich der Ergebnisse berücksichtigt werden. Hierfür wird für jedes Übertragungsverfahren der Durchschnitt der Bewertungsmaße der verschiedenen Transceiver verwendet.

$$X_{mean} = \frac{1}{N} \cdot \sum_{n=0}^{N-1} X_n \mid N = \text{Anzahl betrachteter Transceivertypen} \tag{4.1}$$

Wenn ein Verfahren beispielsweise mit sechs unterschiedlichen Transceivern evaluiert wird, dient der Durchschnitt der sechs Einzelergebnisse als Vergleichsgröße (siehe Tabelle 4.1). Ein Übertragungsverfahren mit nur einem nutzbaren Transceivertyp liefert eine Vergleichsgröße. Die unterschiedliche Anzahl der Angaben zur Leistungsaufnahme und zum Energiebedarf liegt in dem eingeschränkten Frequenzbereich der unterschiedlichen Transceivertypen begründet. Diese Vereinfachung ermöglicht einen qualitativen Vergleich bezüglich des Energiebedarfs pro übertragenem Bit und der Leistungsaufnahme der Verfahren untereinander (vgl. Abbildung 4.1 und Tabelle 4.1, nur jeder zweite Wert ist beschriftet).

Tabelle 4.1: Ermittlung der durchschnittlichen Leistungsaufnahme X_{mean} am Beispiel von SGMII und Gegenüberstellung von Transceiver-Merkmalen.

Transceiver	Leistung	Technologie	Übertragungsrate
Virtex4 MGT	672,7 mW	90 nm	622 Mbit/s - 6,5 Gbit/s
Virtex5 GTP	239,2 mW	65 nm	100 Mbit/s - 3,75 Gbit/s
Virtex5 GTX	475,4 mW	65 nm	150 Mbit/s - 6,5 Gbit/s
Spartan6 GTP	466,4 mW	45 nm	614 Mbit/s - 3,125 Gbit/s
Virtex6 GTX	271,5 mW	40 nm	480 Mbit/s - 6,6 Gbit/s
Virtex6 GTH		40 nm	1,24 Gbit/s - 11,182 Gbit/s
Durchschnitt	**425 mW**		

Wie zu sehen ist, steigt die Leistungsaufnahme von Transceivern mit höher werdender Datenrate des Übertragungsverfahrens. Die Zunahme der Leistungsaufnahme erfolgt dabei mit einem geringeren Wachstum als die Zunahme der Taktfrequenz. Ein Übertragungsverfahren mit einer Taktfrequenz von 5 GBit/s benötigt nicht die doppelte elektrische Leistung wie das gleiche Verfahren bei $2,5$ GBit/s Taktfrequenz. Dieses Verhalten ist bei allen verwendeten elektrischen Modellen zu beobachten, sowohl bei *SPICE*-Modellen von kommerziellen Produkten als auch bei den verschiedenen FPGA-Ressourcen. Dies hat eine Verringerung der benötigten Energie für ein zu übertragendes Bit mit steigender Datenrate zur Folge. Dabei trägt sowohl eine Verwendung von parallelen Übertragungskanälen als auch eine Erhöhung der Taktfrequenz zu der Verringerung des Energiebedarfs bei. Dies bedeutet jedoch nicht, dass der schnellste verfügbare Übertragungsstandard immer auch die optimale Lösung für eine konkrete Anwendung ist. Hierfür müssen die elektrischen Eigenschaften der Übertragungsstrecke betrachtet werden, speziell die Topologie. So kann beispielsweise eine ethernetbasierte Kommunikation über Kabel nicht über mehrere Meter mit einem Backplane-Verfahren wie 10GBase-KR realisiert werden. In dieser Arbeit werden zwei Kommunikationstopologien betrachtet, Punkt-zu-Punkt und busbasierte Varianten. Alle Punkt-zu-Punkt basierten Verfahren müssen eine Synchronisation zwischen Sender und Empfänger sicherstellen. Dies geschieht entweder durch dedizierte Taktsignale oder durch ausreichend viele Taktflanken auf den Datenleitungen, um die jeweiligen PLLs synchron zu halten. Diese Taktflanken müssen auch in Situationen übertragen werden, in denen keine Datenübertragung stattfindet. Es finden also auch in Zeiträume ohne Datenübertragung Pegelwechsel auf den Leitungen statt. Folglich ist die Leistungsaufnahme von seriellen Punkt-zu-Punkt-Verbindungen bei der Datenübertragung und in Bereitschaftszuständen gleich. Anders verhält es sich bei busbasierten Strukturen, hier muss keine andauernde Synchronisation zwischen den Übertragungspartnern stattfinden. Außerhalb der Übertragungszyklen findet keine Aktivität auf dem Bus statt.

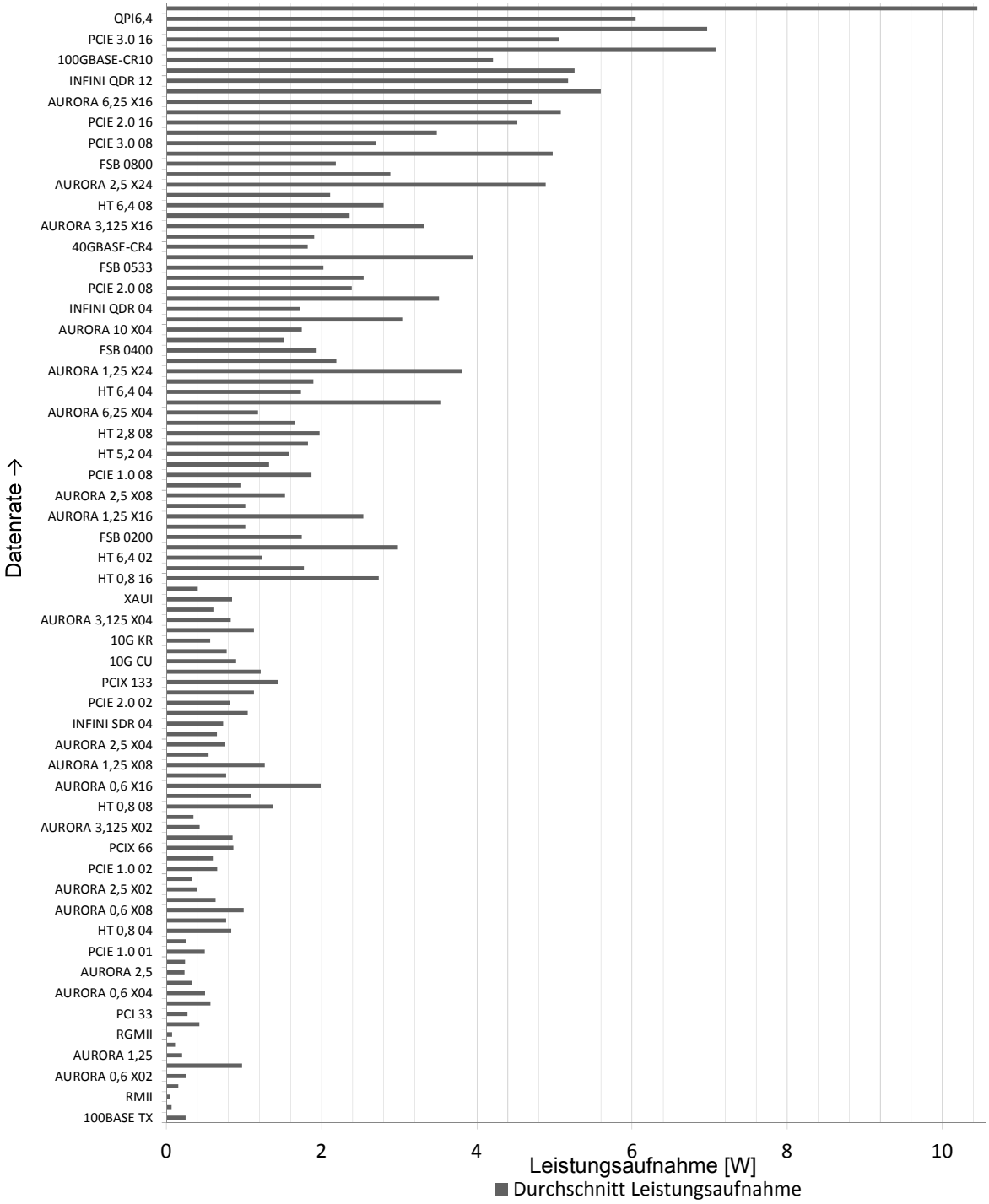

Abbildung 4.1: Durchschnittliche Leistungsaufnahme der betrachteten Übertragungsverfahren bei zunehmender Datenrate.

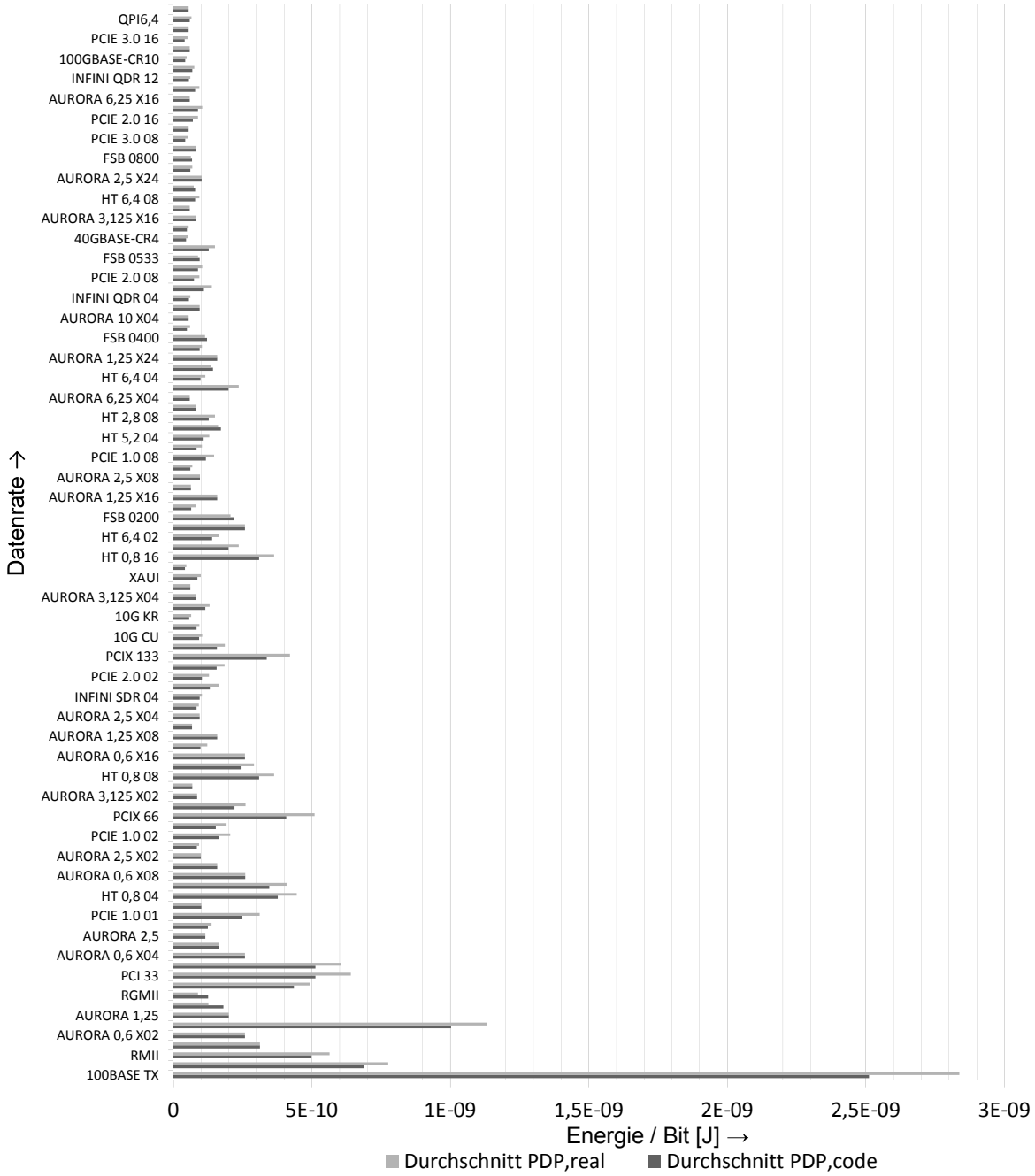

Abbildung 4.2: Durchschnittliche Energieeffizienz der betrachteten Übertragungsverfahren bei zunehmender Datenrate.

Zusammenfassend kann folgende Aussage über die Übertragungstopologie und ihre optimale Anwendung getroffen werden:

- Busbasierte Übertragung: Da zwischen den Übertragungszyklen keine Aktivität auf dem Bus stattfindet, wird in diesen Pausen auch kaum Verlustleistung erzeugt. Um eine möglichst effiziente Kommunikation zu implementieren, sollte immer die leistungsfähigste Variante des einzusetzenden Standards gewählt werden, da diese Variante auch die geringste Energie pro Bit benötigt.
- serielle Punkt-zu-Punkt Übertragung: Die PLLs von Sender und Empfänger müssen auch außerhalb der Übertragungszyklen synchron gehalten werden. Dies geschieht durch andauernde Pegelwechsel auf den Datenleitungen und die Übertragung von Steuerinformationen. Verfahren mit dedizierter Taktleitung wie PCI-Express erhalten die Synchronität ebenfalls durch Aktivität auf den Datenleitungen aufrecht, da der Takt zusätzlich im Datenstrom kodiert wird. Die dedizierte Taktleitung wird hauptsächlich bei Inbetriebnahme der Übertragungsstruktur zur initialen Kommunikation vom Master verwendet. Der Bedarf von Energie in Phasen der Datenübertragung und bei Bereitschaft, ist identisch. Bei der Auswahl einer geeigneten Variante des einzusetzenden Standards sollte die Wahl immer auf diejenige Variante fallen, welche die Anforderungen an die Übertragungsrate erfüllt und gleichzeitig die geringste Verlustleistung aufweist. Die betrachteten Übertragungsstandards unterstützen verschiedene Energiesparmodi wie die Abschaltung von einzelnen Kanälen oder die Reduzierung der Taktrate. Die große Anzahl der unterschiedlichen Ansätze zum Reduzieren der Leistungsaufnahme würde einen qualitativen Vergleich der Übertragungsstandards sehr erschweren. Deshalb wird in dieser Arbeit von einer Datenübertragung mit maximaler Leistung ausgegangen.

4.1.1 Topologiebasierte Evaluation

Nachdem eine erste Aussage über den Energiebedarf der in dieser Arbeit betrachteten Übertragungsverfahren getätigt wurde, soll in diesem Abschnitt eine Unterteilung in busbasierte Verfahren und serielle Punkt-zu-Punkt-Übertragungen stattfinden. Es wird eine Betrachtung der auf ähnlichen Transceivertypen implementierten Standards durchgeführt. Alle hier betrachteten busbasierten Standards verwenden entweder LVTTL- oder GTL-basierte Transceiver, die seriellen Standards nutzen CML- oder LVDS-Transceiver. Aufrgrund der Ähnlichkeit der Transceivertypen innerhalb einer Topologie bezüglich ihrer elektrischen Eigenschaften kann ein Vergleich gezogen werden, welcher die Protokolle und Leitungskodierungen der Verfahren gegenüber transceiverspezifischen Einflüssen hervorhebt.

Busbasierte Verfahren

Die hier verglichenen Verfahren verwenden entweder GTL- oder LVTTL-Transceiver und eine busbasierte Kommunikationsinfrastruktur. Die parallelen, medienunabhängigen Schnittstellen stellen einen Sonderfall dar, da hier immer nur zwei Partner miteinander kommunizieren. Es handelt sich also um keine Busstruktur mit mehr als zwei Teilnehmern, trotzdem werden diese Verfahren aufgrund der verwendeten Transceiver in diesem Abschnitt betrachtet. Im Vergleich zu anderen Bussystemen werden die medienunabhängigen Schnittstellen für Fast-Ethernet mit einer vergleichsweise geringen Taktrate betrieben. Unter Berücksichtigung der vorherigen Ergebnisse ergibt sich so ein größerer Energiebedarf. Höher getaktete Verfahren wie GMII, RGMII oder XGMII arbeiten wesentlich effektiver. Allgemein weisen die reduzierten, medienunabhängigen Schnittstellen eine niedrigere Leistungsaufnahme auf als die Varianten mit höherer Parallelität (vgl. 4.3). Ein ähnliches Verhalten ergibt sich bei den systeminternen Bussen wie PCI oder FSB, der Energiebedarf für ein zu übertragenes Bit fällt mit steigender Datenrate. Sehr deutlich ist auch der zunehmend flacher werdende Anstieg der Verlustleistung, beispielsweise zwischen FSB400 und FSB800. Während sich hier die Datenrate verdoppelt, erhöht sich die Leistungsaufnahme nur um wenig mehr als ein Zehntel. Für lange Zeit wurden busbasierte Verfahren wie der Frontside-Bus in vielen Systemen wie beispielsweise Computern eingesetzt. Die verwendete LVTTL- und GTL-Technik stößt jedoch bei der Realisierung von immer höheren Datenraten an ihre Grenzen. Da die Transceiver nicht beliebig hohe Schaltfrequenzen umsetzen können, muss ein entsprechendes Übertragungsverfahren zur Realisierung höherer Übertragungsraten mehr parallele Kanäle einsetzen. Die bedeutet unter anderem eine Zunahme des Platzbedarfs für die Signalwegeplanung und Signalintegritätsprobleme durch Gruppenlaufzeiten und Übersprechen. Aus diesem Grund werden zunehmend serielle Verfahren mit Hochgeschwindigkeitstransceivern eingesetzt.

Serielle Punkt-zu-Punkt Übertragung und Inter-System-Kommunikation

Bis auf die Multipegeltransceiver von Fast- und Gigabit-Ethernet verwenden alle in dieser Arbeit betrachteten Standards FPGA-basierte CML-Transceiver. Die Multipegelimplementierungen dienen zum qualitativen Vergleich untereinander und den anderen Verfahren. Ein quantitativer Vergleich kann prinzipbedingt nur zwischen 100BaseTX und 1000BaseT bzw. innerhalb der CML-Varianten erfolgen. Ähnlich den zuvor gewonnenen Erkenntnissen bei FPGA-basierten Verfahren steigt auch zwischen 100BaseTX und 1000BaseT die Datenrate schneller als die Verlustleistung. So erzeugt 1000BaseT bei einer zehnfachen Datenrate gegenüber 100Base-TX nur die fünffache Menge an elektrischer Verlustleistung. Bei den CML-Verfahren kann ein annähernd linearer Anstieg der Verlustleistung bei höher werdender Datenrate beobachtet werden.

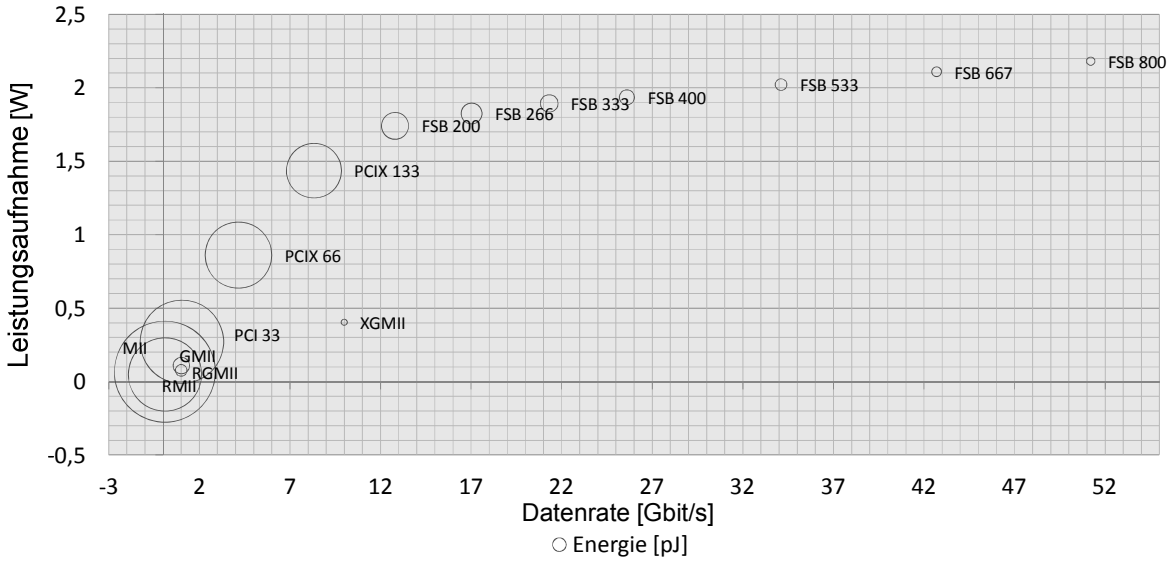

Abbildung 4.3: Vergleich von GTL- und LVTTL-basierten Kommunikationsstandards bezüglich Energiebedarf, Leistungsaufnahme und Datenrate.

Auch hier fällt die benötigte Energiemenge pro Bit mit steigender Datenrate, besonders deutlich wird dies bei dem Aurora Standard (vgl. Abbildung 4.4, Familien sind gleich eingefärbt). Hierbei wird auch die geringere Leistungsaufnahme einer Implementierung mit erhöhter Datenrate gegenüber einer Implementierung mit höherer Parallelität deutlich. So erreichen beispielsweise Aurora mit 6,25 Gbit/s und einem einzelnen Kanal und Aurora mit 3,125 Gbit/s und zwei parallelen Kanälen dieselbe Datenrate. Die erste Variante benötigt jedoch weniger Energie pro übertragenem Bit und besitzt eine geringere Leistungsaufnahme. Dies zeigt sich auch im Vergleich zwischen dem HyperTransport-Verfahren mit einer großen Anzahl an parallelen Kanälen und dem höher getakteten PCI-Express mit weniger Parallelität. Auffällig ist die durchschnittlich niedrige Leistungsaufnahme von Infiniband, welche trotz eines höheren Protokollaufwands in den meisten Fällen mit Aurora und PCIe konkurrieren kann und sich so als idealer Standard für Inter-System-Kommunikation erweist, also für die Datenübertragung über längere Strecken zwischen physikalisch getrennten Systemen (beispielsweise PCs). Welcher Standard für die Kommunikation zwischen Systemen optimal ist, hängt jedoch von der benötigten Datenrate ab. So dominiert bei einer Rate von 4 Gbit/s Infiniband vor Aurora, bei 8 Gbit/s ist es jedoch umgekehrt. Bei 10 Gbit/s ist wiederum das Ethernet-Verfahren 10G-KR die optimale Wahl bezüglich der Leistungsaufnahme. Allgemein ist in Abbildung 4.4 ein linearer Anstieg der Leistungsaufnahme mit zunehmender Datenrate zu sehen. Bei niedrigen Datenraten herrscht eine größere Streuung in der Leistungsaufnahme und der benötigten Energie zwischen den einzelnen Verfahren. Dies liegt am relativ hohen Anteil der frequenzunabhängigen Verluste an der gesamten Verlustleistung eines Transceivers, entsprechend

stark wirkt sich dieser Anteil auf die Verwendung von mehreren parallelen Transceivern aus. Dieses Verhalten setzt sich in einem abnehmenden Maße bis hin zu hohen Datenraten fort, da der frequenzabhängige Verlustleistungsanteil nicht so stark anwächst wie der frequenzunabhängige Anteil, begründet durch die wachsende Anzahl paralleler Transceiver. Werden beispielsweise 16 oder 24 Transceiver bei maximaler Datenrate verwendet, so dominiert die Anzahl der Transceiver die gesamte Leistungsaufnahme. Diese Divergenz fällt bei verschiedenen Übertragungsstandards unterschiedlich stark aus. PCI-Express unterstützt beispielsweise nur drei serielle Bitraten während Aurora eine Vielzahl von Bitraten realisieren kann. Zusammen mit der jeweils möglichen Anzahl von Kanälen ergibt sich so bei Aurora eine größere Zahl von Varianten für eine bestimmte Datenrate. Aurora kann deswegen besser auf verschiedene Optimierungsziele angepasst werden als andere Verfahren. Dies kann eine bessere EMV-Verträglichkeit durch die Nutzung einer hohen Parallelität und einer niedrigen seriellen Bitrate sein oder eine geringere Verlustleistung bei Verwendung von wenigen parallelen Kanälen mit hoher Bitrate. Abbildung 4.4 stellt in gewisser Weise auch die zeitliche Entwicklung von seriellen Übertragungsverfahren dar. Ältere Verfahren wie 100Base-TX oder 1000Base-T finden sich weiter links im Diagramm und benötigen viel Energie pro Bit, neuere Verfahren wie 40GBase-CR4 oder 100GBase-CR10 sind am rechten Rand angeordnet und benötigen weitaus weniger Energie pro Bit. Setzt sich der Trend in Abbildung 4.4 zukünftig fort, so ist sowohl von einer Steigerung der Leistungsfähigkeit als auch von einer Senkung des Energiebedarfs auszugehen. Dies gilt beispielsweise bei Betrachtung neuester und zukünftiger Infiniband-Standards wie beispielsweise Infiniband FDR (Fourteen Datarate, 14,0625 Gbit/s) und Infiniband EDR (Extended Datarate, 25 Gbit/s). Diese Standards werden aggregierte Datenraten von ca. 170 Gbit/s bzw. 300 Gbit/s erreichen und voraussichtlich über 20 Watt an Verlustleistung erzeugen. Aufgrund der Vielzahl der PCI-Express-Varianten und der damit verbundenden Abdeckung von Anwendungszenarien stellt dieses Verfahren die optimale Wahl für die Realisierung von Kommunikationsstrecken über wenige Meter dar. Für längere Strecken von bis zu 15 m können 40GBase-CR4 und 100GBase-CR10 eingesetzt werden. PCI-Express und Ethernet sind weit verbreitete Standards, für die eine Vielzahl von Produkten erhältlich ist. Diese Produkte lassen sich durch vorhandene Treiberunterstützung gut in eine Anwendung integrieren. Aurora deckt zwar den größten Bereich an Übertragungsraten ab, benötigt jedoch mehr Aufwand bei der Integration in eine Anwendung. Eine einfache Treiberanbindung von Aurora-Übertragungsstrecken in ein Betriebssystem ist nicht vorhanden, weshalb Aurora eher für Anwendungen geeignet ist, die eine angepasste und benutzerdefinierte Kommunikationsschnittstelle benötigen.

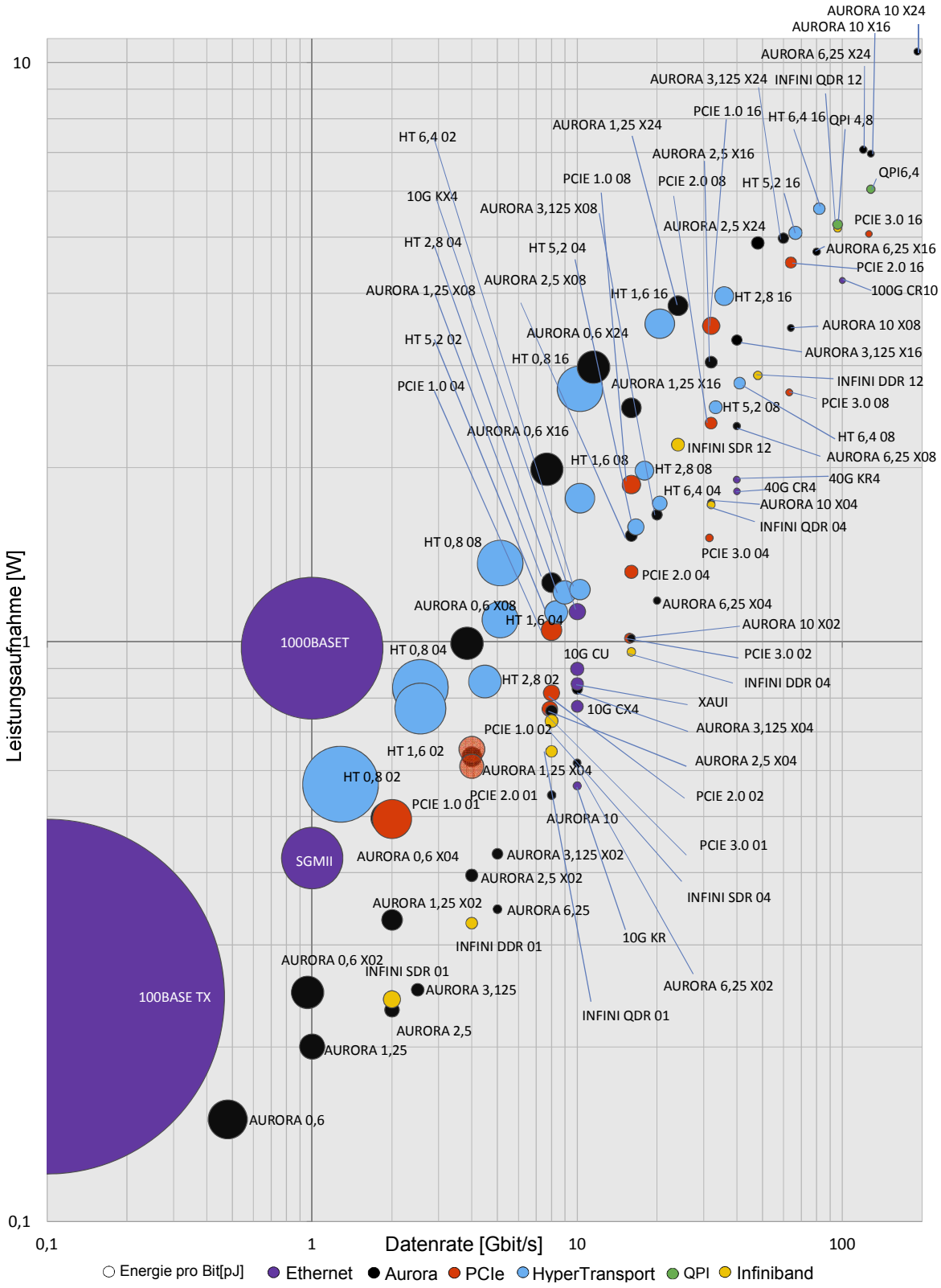

Abbildung 4.4: Vergleich serieller Kommunikationsstandards bezüglich Energiebedarf, Leistungsaufnahme und Datenrate.

Evaluation von Standards zur Intra-System-Kommunikation

Bei dem Vergleich systeminterner Kommunikationsstandards, also Verfahren zur Datenübertragung innerhalb eines Systems wie z. B. eines PCs, weisen veraltete Standards wie PCI einen hohen Energiebedarf auf (vgl. 4.5). Neuere Verfahren wie QPI besitzen wie erwartet eine hohe Leistungsfähigkeit bei einem gleichzeitig geringen Energiebedarf. Betrachtet man die Verteilung der betrachteten Verfahren, so erweist sich der ältere Frontside-Bus als ein Verfahren mit vergleichsweise niedriger Leistungsaufnahme. Er liegt in weiten Bereichen bezüglich benötigter Energie pro Bit und Leistungsaufnahme vor dem wesentlich moderneren HyperTransport-Verfahren. Die bei HyperTransport verwendeten seriellen Transceiver besitzen im Vergleich zu den GTL-Transceivern eine hohe Verlustleistung. HyperTransport kann lediglich bei Verwendung hoher Frequenzen und einer kleinen Anzahl von parallelen Transceivern (z.B. 6,4 Gbit/s und 4 Kanäle) eine geringere Leistungsaufnahme als der Frontside-Bus erzielen. Durch die höhere Parallelität des Frontside-Bus Verfahrens benötigt dieses jedoch mehr physikalische Leitungen und damit mehr Platz als HyperTransport. Beide Verfahren werden zur Kopplung von CPU und Chipsatz, aber auch zur Kopplung mehrerer CPUs verwendet. Für diese Anwendung wurde auch QPI entwickelt, es stellt sich gegenüber HyperTransport als leistungsfähiger heraus. Würde HyperTransport mit ähnlichen Datenraten wie QPI implementiert, so würden beide Verfahren ähnliche Werte in der Leistungsaufnahme, Datenrate und der benötigten Energie pro Bit zeigen. PCI-Express deckt den größten Performanzbereich ab, weist im jeweiligen Bereich die geringste Leistungsaufnahme auf und ist zugleich vielseitiger einsetzbar als die anderen Verfahren. Durch die zukünftige Entwicklung von PCI-Express 4.0 mit 16 GT/s wird die verfügbare Übertragungsrate von PCI-Express voraussichtlich 512 GT/s bei Verwendung von 32 Kanälen erreichen [R16]. Aus diesen Gründen ist PCI-Express das zurzeit optimale Übertragungsverfahren zur Anbindung von Peripherie in Rechenarchitekturen, wenn ein hoher Datendurchsatz erreicht werden soll. Bei der Kopplung von Prozessoren und Chipsatz zeigt das Frontside-Bus-Verfahren die beste Kombination aus Leistungsfähgkeit und Leistungsaufnahme. Allerdings benötigt der FSB über 100 Signale in Form von Kupferleitungen und damit weit mehr als beispielweise HyperTransport. Dadurch wird die Entflechtung und die Wegeplanung der Signale erschwert und mehr Platz benötigt.

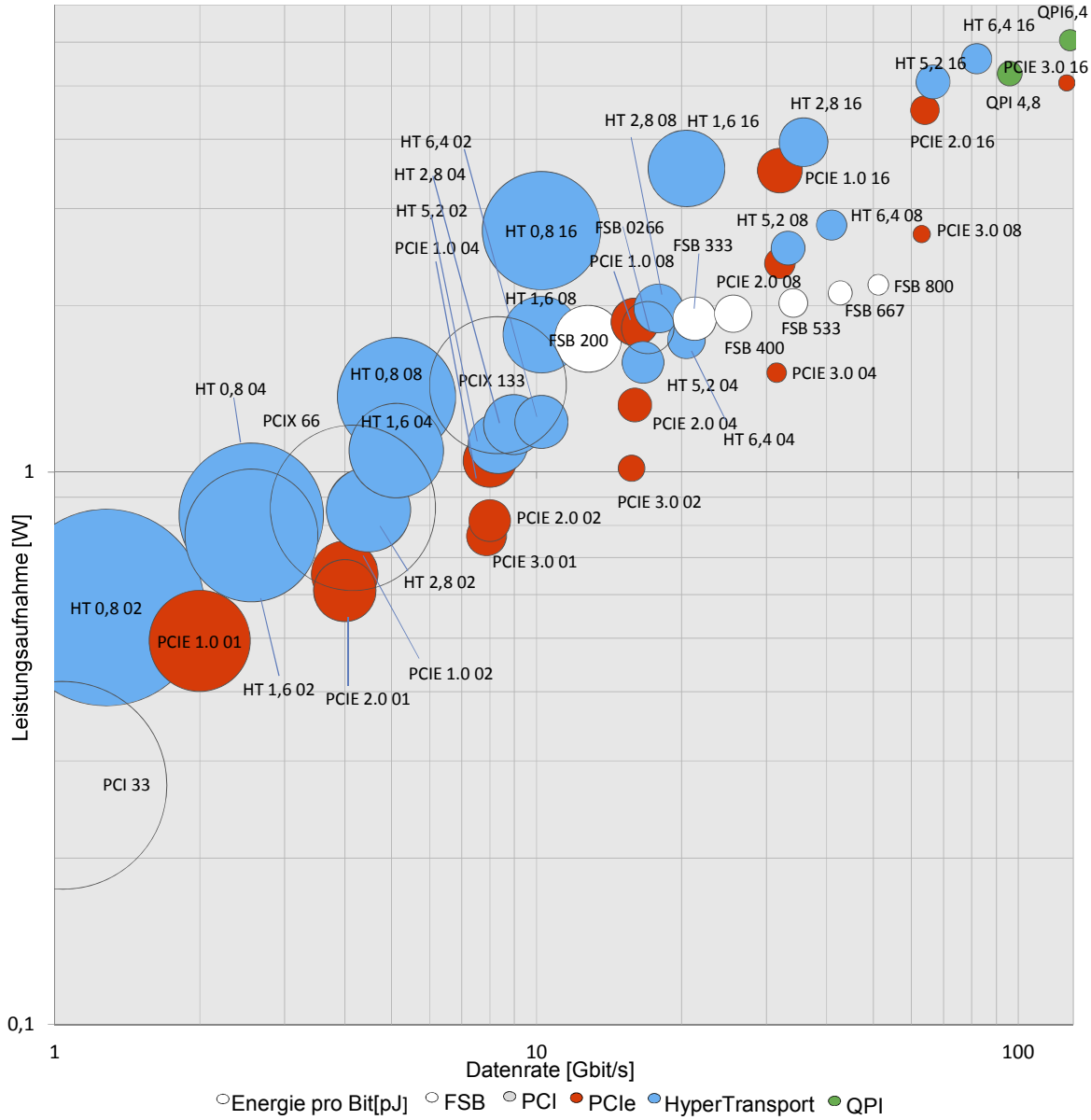

Abbildung 4.5: Vergleich von Standards zur Intra-System-Kommunikation bezüglich Effizienz, Leistungsaufnahme und Datenrate.

4.2 Evaluation von Kommunikation in Multiprozessorarchitekturen

In den vorherigen Kapiteln wurden verschiedene Übertragungsstandards bezüglich Leistungsaufnahme und Energieeffizienz evaluiert und untereinander verglichen. Anhand der Erkenntnisse aus den Kapiteln 4.1 und 3 soll der Einfluss der Kommunikation in einer Multiprozessorarchitektur auf die Gesamtverlustleistung ermittelt werden. In diesem Kapitel sollen komplette Architekturen evaluiert werden, in denen Kommunikation eine große Rolle spielt. Hierbei handelt es sich um Clustersysteme aus Recheneinheiten, welche untereinander stark vernetzt sind. Alle heutigen HPC-Systeme (Hochleistungsrechner) werden in Form solcher Cluster realisiert, von denen exemplarisch die Systeme des Paderborn Center of Parallel Computing (PC^2) betrachtet werden. Diese Architekturen verwenden Standardprozessoren und Komponenten, welche auch in normalen Computern oder Servern eingesetzt werden. Für alle angesprochenen Rechnerarchitekturen und Systeme soll der Anteil der Intra- und Intersystemkommunikation an der Gesamtverlustleistung bestimmt werden. Für Komponenten wie CPUs, Netzwerkkomponenten oder Chipsatz sind keine exakten Angaben zur Leistungsaufnahme der implementierten Kommunikationsverfahren erhältlich, jedoch können diese durch die FPGA-basierten Verfahren angenähert und auf die Gesamtleistung aufgeschlagen werden. Hierzu wird die durchschnittliche Leistungsaufnahme der jeweils evaluierten FPGA-Transceiver genutzt (siehe Definition in Kapitel 4.1). Tabelle 4.2 zeigt einen Vergleich zwischen Herstellerangaben zur Verlustleistung ihrer Produkte und den ermittelten Werten aus Kapitel 3.

Tabelle 4.2: Leistungsaufnahme von kommerziellen Kommunikationskomponenten im Vergleich zu FPGA-basierten Verfahren.

Produkt	Leistung Produkt	Schnittstellen	Leistung FPGA-basiert
QLogic QLE 7342 [R57]	**6,2 W**	2x InfiniBand QDRx4	2x 1,73 W
		PCIe 2.0 x8	2,39 W
			5,84 W
LSI VC1052 [R67]	**0,9 W**	4x SerDes	0,23 W
		PCIX-66	0,64 W
			1,09 W
Intel 82599 [R35]	**4,5 W**	2x 10GBase-CX4	2x 0,77 W
		PCIe 2.0 x8	2,39 W
			3,93 W
Mellanox ConnectX [R19]	**3,0 W**	InfiniBand SDRx4	0,73 W
		PCIe 1.0 x8	1,87 W
			2,61 W

Betrachtet wird jeweils die Leistungsaufnahme eines Systems inklusive Kommunikationsinfrastrukturen wie Switches. Die Angaben beziehen sich auf den Betrieb des Systems bei Ausführung des *Linpack-Benchmarks* [R6]. Dieser Benchmark ist ein typisches Lastszenario für prozessorbasierte Rechencluster und lastet neben den Prozessoren und dem Speicher insbesondere die Kommunikationswege aus. Wie in Kapitel 5.2 zu sehen ist, kann durch die Ausführung eines *Linpack-Benchmarks* auf einem System die maximal angegebene Verlustleistung des Systems erzeugt werden. Hierdurch können die Angaben zur maximalen Leistungsaufnahme als realistische Angaben in der Evaluation verwendet werden. Nachfolgend sind die Spezifikationen der betrachteten HPC-Systeme [R54] aufgelistet.

- Arminius (2005)
  - 7700 GFlop/s
  - 60 Rechenknoten mit jeweils zwei X5650 Prozessoren, untereinander und mit dem Chipsatz über QPI mit 6,4 GT/s gekoppelt.
  - Anbindung des Netzwerkadapters an den Chipsatz über PCIe der ersten Generation mit acht Lanes.
  - Kopplung der Rechenknoten über Infiniband mit vier Kanälen im SDR-Modus.
  - Infiniband-Switch
- PLING2
  - 4700 GFlop/s
  - 49 Rechenknoten mit jeweils zwei Xeon E5540 Prozessoren, untereinander und mit dem Chipsatz über QPI mit 6,4 GT/s gekoppelt.
  - 8 Rechenknoten mit jeweils zwei Xeon X5570 Prozessoren, untereinander und mit dem Chipsatz über QPI mit 6,4 GT/s gekoppelt.
  - Anbindung des Infinibandadapters an den Chipsatz über PCIe der ersten Generation mit acht Lanes.
  - Kopplung der Rechenknoten über Infiniband mit vier Kanälen im SDR-Modus.
  - Zusätzliche Kommunikation über Gigabit-Ethernet.
  - Anbindung des Gigabit-Ethernet-Adapters an den Chipsatz über PCIe der ersten Generation mit einer Lane.
  - Infiniband- und Gigabit-Ethernet-Switch
- HPC-Cloud
  - 2600 GFlop/s

  - 38 Rechenknoten mit jeweils zwei Xeon E5506 Prozessoren, untereinander und mit dem Chipsatz über QPI mit 4,8 GT/s gekoppelt.
  - Kommunikation zwischen den Knoten über Gigabit-Ethernet.
  - Anbindung des Gigabit-Ethernet-Adapters an den Chipsatz über PCIe der ersten Generation mit einer Lane.
  - Gigabit-Ethernet-Switch
- WinHPC
  - 224 GFlop/s
  - 4 Rechenknoten mit jeweils zwei Xeon 5160 Prozessoren, untereinander und mit dem Chipsatz über FSB 1066 gekoppelt.
  - Anbindung des Infinibandadapters an den Chipsatz über PCIe der zweiten Generation mit acht Lanes.
  - Anbindung des Gigabit-Ethernet-Adapters an den Chipsatz über PCIe der ersten Generation mit einer Lane.
  - Kopplung der Rechenknoten über Infiniband mit vier Kanälen im DDR-Modus.
  - Zusätzliche Kommunikation über Gigabit-Ethernet.
  - 2 Rechenknoten mit jeweils zwei Opteron 270 Prozessoren, untereinander und mit dem Chipsatz über QPI mit 6,4 GT/s gekoppelt.
  - Anbindung des Infinibandadapters an den Chipsatz über PCIe der ersten Generation mit acht Lanes.
  - Anbindung des Gigabit-Ethernet-Adapters an den Chipsatz über PCIe der ersten Generation mit einer Lane.
  - Kopplung der Rechenknoten über Infiniband mit vier Kanälen im SDR-Modus.
  - Zusätzliche Kommunikation über Gigabit-Ethernet.
  - Infiniband- und Gigabit-Ethernet-Switch

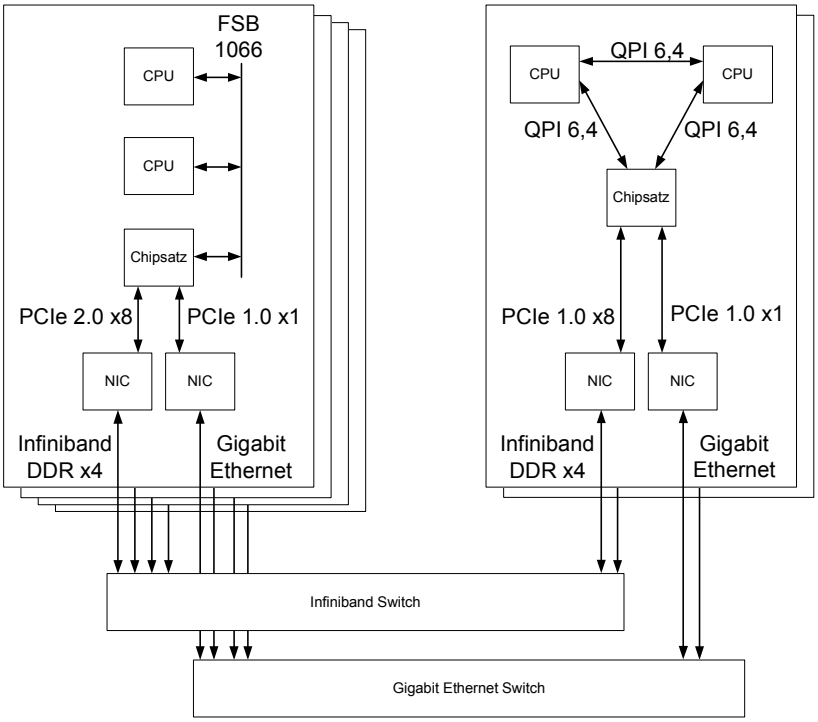

Abbildung 4.6: Übersicht der an der Kommunikation beteiligten Komponenten am Beispiel des WinHPC-Systems.

Tabelle 4.3: Der Anteil von Intra- und Intersystemkommunikation an der Verlustleistung von HPC-Systemen.

System	Verlustleistung Gesamt	Verlustleistung Kommunikation	Intern	Extern
Arminius	19500 W	2488 W **(12,8 %)**	2400 W QPI + PCIe	87,6 W IB
PLING2	18901 W	2530 W **(13,4 %)**	2336 W QPI + PCIe	81,4 W IB + Eth.
HPC Cloud	11470 W	1310 W **(11,4 %)**	1237 W QPI + PCIe	74,1 W Eth.
WinHPC	2151 W	114 W **(5,3 %)**	91,4 W QPI + FSB + PCIe	22,3 W IB + Eth.

Die Leistungsaufnahme der einzelnen Systeme wird durch die Betrachtung der Einzelkomponenten abgeschätzt und entspricht der Leistungsaufnahme in einem realistischen Szenario wie der Ausführung eines *Linpack-Benchmarks*. Der Kommunikationsanteil der Verlustleistung wird durch die Addition der Leistungsaufnahme aller beteiligten Übertragungsstandards ermittelt. Darstellung 4.6 verdeutlicht die einzelnen Kommunikationsanteile am Beispiel des WinHPC-Systems. Tabelle 4.3 zeigt eine Übersicht der betrachteten Systeme, der gesamten Verlustleistung und den Anteil der Kommunikation an der gesamten Verlustleistung. Es stellt sich heraus, dass bei größeren Clustersystemen der Anteil der Systemkommunikation an der gesamten Verlustleistung zwischen ca. 11 % und 14 % liegt. Ein System wie WinHPC nutzt bei zwei Drittel seiner Knoten den Frontside-Bus-Standard zur Kommunikation zwischen CPUs und Chipsatz. Dieser weist eine deutlich niedrigere Leistungsaufnahme auf, als der in den anderen Systemen verwendete QPI-Standard. Hiervon profitiert das gesamte System, was sich in einem deutlich geringeren Kommunikationsanteil an der Verlustleistung zeigt. Dies liegt neben der höheren Leistungsaufnahme der seriellen Transceiver auch an ihrer Anzahl. Aufgrund der Punkt-zu-Punkt-Verbindung bei QPI werden sechs Transceiver für die Kopplung von zwei CPUs und dem Chipsatz benötigt. Bei Verwendung des Frontside-Busses werden aufgrund der Busstruktur nur drei Transceiver benötigt. Bei allen Rechenclustern liegt der Anteil der Intrasystemkommunikation an der Gesamtverlustleistung wesentlich höher als der Anteil der Intersystemkommunikation. Der Grund liegt dabei hauptsächlich in der großen Anzahl an seriellen Transceivern des QPI-Verfahrens. Diese Erkenntnisse ermöglichen eine Auswahl an Komponenten und Übertragungsverfahren, um ein Multiprozessorsystem auf verschiedene Ziele hin zu optimieren. Soll ein System beispielsweise auf niedrige Verlustleistung hin optimiert werden, so muss zusätzlich zur Auswahl von Komponenten wie sparsamen Prozessoren auch die Kommunikationsstruktur betrachtet werden und Verfahren wie der Frontside-Bus bevorzugt eingesetzt werden. Sollen mehere Prozessoren möglichst latenzarm miteinander kommunizieren, so sollte ein Punkt-zu-Punkt-Verfahren eingesetzt werden, da hierbei jeder Kommunikationspartner gleichzeitig senden und empfangen kann. Bei Busstrukturen kann zu jeder Zeit immer nur eine Einheit schreibend auf den Bus zugreifen.

5 Energieeffiziente Multiprozessorarchitekturen mit optimierter Kommunikationsinfrastruktur

Bei der Entwicklung von Multiprozessorarchitekturen wird seit einiger Zeit auf einen geringen Energiebedarf der Systeme geachtet. Der Hauptaufwand liegt dabei in der Optimierung der Systemkomponenten wie Prozessoren oder auch Netzteilen. Die Energieeffizienz wird unter anderen durch Mechanismen zur Takt- und Spannungsreduzierung erhöht. Statt immer höher getaktete Prozessoren einzusetzen, wird die Last auf parallel arbeitende Prozessoren mit relativ niedriger Taktrate verteilt, um Energie zu sparen. Die zunehmende Parallelisierung von Rechenarchitekturen bedingt jedoch auch eine schnelle Kommunikation zwischen den einzelnen Knoten. Wie in Kapitel 4.1 gezeigt, unterscheiden sich die Übertragungsstandards in der Leistungsaufnahme und der benötigten Energie pro Bit. Außerdem gibt es Szenarien, in denen ein Übertragungsstandard effizienter eingesetzt werden kann als ein anderer. Die Erkenntnisse aus den Kapiteln 4.1 und 4.2 ermöglichen die Einbeziehung der Inter- und Intrasystemkommunikation in den Entwurf von Multiprozessorarchitekturen und deren Optimierung für verschiedene Szenarien. Im Fachgebiet Schaltungstechnik der Universität Paderborn und dem Fachgebiet Kognitronik und Sensorik der Universität Bielefeld wurden zwei energieeffiziente Multiprozessorarchitekturen für verschiedene Szenarien entwickelt. Das erste System ist ein Rechnerverbund mit einer Vielzahl an eng gekoppelten Hardwarebeschleunigern auf Basis von FPGAs mit einer speziellen Kommunikationsinfrastruktur [E1]. Das zweite System nutzt Universalprozessoren mit X86-Architektur und eine Kommunikationsinfrastruktur mit Standard-Übertragungsverfahren. Nachfolgend werden beide Systeme vorgestellt.

5.1 SCT-Cluster - Ein dynamisch rekonfigurierbarer Rechencluster

Große Entwürfe von SOCs (System on Chip) oder Prozessoren müssen vor der Fertigung als ASIC umfassend simuliert werden, um das korrekte Verhalten des Entwurfs zu verifizieren. Dies kann auf handelsüblichen Computern mit Standardprozessoren geschehen. Durch die serielle Arbeitsweise der CPUs kann eine Simulation großer Systementwürfe jedoch sehr lange dauern. Aus diesem Grund werden für diese Aufgaben oft rekonfigurierbare Architekturen wie FPGAs eingesetzt. Diese Architekturen können so konfiguriert werden, dass ihr Verhalten dem des Systementwurfs entspricht. Sehr große Entwürfe können jedoch nicht auf einem einzelnen FPGA umgesetzt werden, weshalb ein Verbund aus eng gekoppelten FPGAs benötigt wird. Im Fachgebiet Schaltungstechnik der Universität Paderborn und dem Fachgebiet Kognitronik und Sensorik der Universität Bielefeld wurde hierfür ein auf FPGAs basierender, dynamisch rekonfigurierbarer Rechencluster entwickelt [E2]. Dieser dient zur Simulation großer Schaltungsentwürfe oder neuronaler Netze. Der wesentliche Unterschied zu prozessorbasierten Rechnerverbünden besteht in der Konzeptionierung als Verbund aus FPGAs mit Unterstützung von Prozessoren, nicht als Prozessorverbund mit FPGAs als Hardwarebeschleuniger. Abbildung 5.1 zeigt eine Übersicht des Clusteraufbaus.

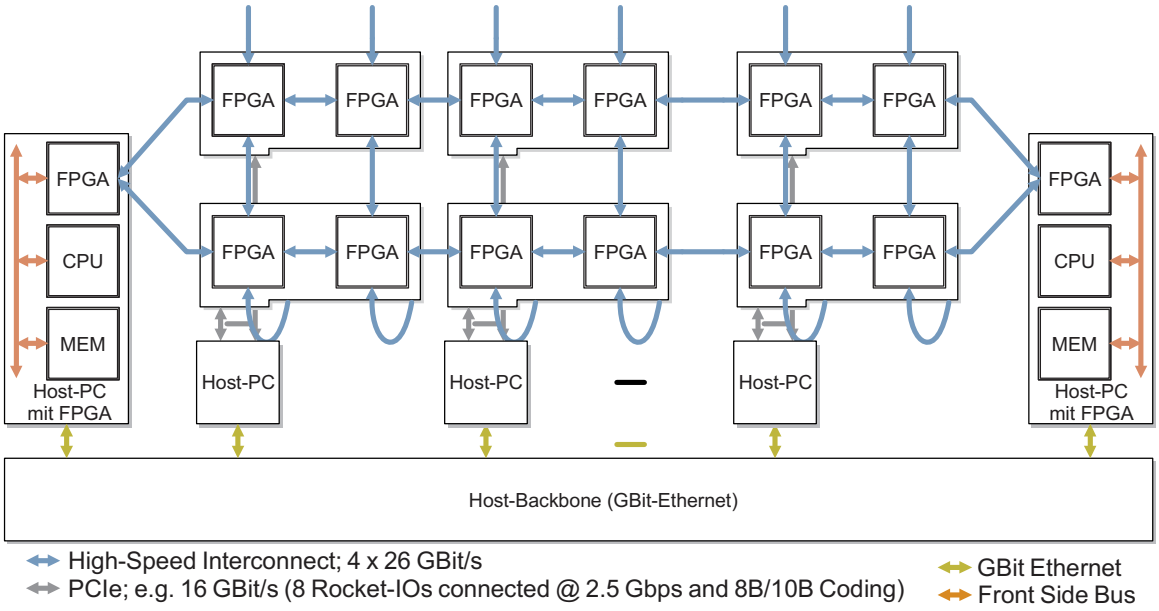

Abbildung 5.1: Der schematische Aufbau des SCT-FPGA-Clusters.

Als Basis dient eine flexible Anzahl von PCs, die untereinander kommunizieren. Da hier keine zeitkritische Kommunikation stattfindet, wurde auf dieser Ebene Gigabit-Ethernet in Form von 1000Base-TX (siehe Kapitel 3.4.2) verwendet. Dieses Verfahren besitzt eine niedrige Leistungsaufnahme und ist günstig zu realisieren. Jeder der PCs kann ein oder mehrere RAPTOR-XPress Rapid-Prototyping-Systeme beinhalten. Diese Rapid-Prototyping-Systeme können mit jeweils vier FPGA-basierten Modulen bestückt werden. Neben den Plattformen mit integrierten RAPTOR-XPress Systemen werden spezielle PCs mit eng an den Prozessor gekoppelten, FPGA-basierten Hardwarebeschleunigern eingesetzt. Diese enge Kopplung muss latenzarm erfolgen und trotzdem einen hohen Datendurchsatz ermöglichen. Das Frontside-Bus-Verfahren (Kapitel 3.2.1) weist im Vergleich zu anderen geeigneten Verfahren wie QPI (Kapitel 3.3.6) oder HyperTransport (Kapitel 3.3.7) einen geringen Mehraufwand für das Protokoll auf und besitzt bei hohen Übertragungsraten die geringste Leistungsaufnahme. Deshalb wurde das Frontside-Bus-Verfahren für diese Anwendung gewählt. Eine weitere Kommunikationsebene findet zwischen den einzelnen FPGA-Platinen des Verbunds auf Basis von Kupferkabeln statt. Hierbei liegt das Optimierungsziel auf einem möglichst hohen Durchsatz, einem geringen Protokoll-Mehraufwand und einer geringen Latenz. Aus diesen Vorgaben wurde das Aurora-Übertragungsverfahren (Kapitel 3.3.8) als optimale Lösung gewählt. Zusammenfassend ergeben sich verschiedene Arten der Systemkommunikation zwischen den Komponenten, welche im Wesentlichen in drei Typen eingeordnet werden können.

- Inter-FPGA-Kommunikation
  - Direkte, feste Verbindungen zwischen benachbarten FPGAs auf einem RAPTOR-XPress-Modul auf Basis von LVDS (Kapitel 3.5).
  - Geschaltete LVDS-Broadcastverbindungen zwischen den FPGAs auf einem RAPTOR-XPress-Modul.
  - Serielle, dynamisch rekonfigurierbare Verbindungen zwischen allen FPGAs im Verbund (Aurora, siehe Kapitel 3.3.8).
- FPGA-CPU-Kommunikation
  - PCI-Express-Anbindung der RAPTOR-XPress-Systeme an die CPU des jeweiligen Rechners.
  - Direkte Verbindung der FPGAs in speziellen Knoten an den Frontside-Bus des Systems mit minimaler Latenz.
- Inter-PC-Kommunikation
  - Gigabit-Ethernet Verbindung aller PCs im Verbund.

Die einzelnen Komponenten des FPGA-Clusters und ihre Kommunikationstechniken werden im Folgenden detaillierter vorgestellt.

Das DB-V5 FPGA-Modul

Das im Fachgebiet Schaltungstechnik entwickelte FPGA-Modul DB-V5 ist auf die Verwendung in Kombination mit dem RAPTOR-XPress zugeschnitten worden. Wie der Name suggeriert, kommen hier Virtex-5-FPGAs von Xilinx zum Einsatz. Ein LX50T, LX85T, LX110T, LX155T, SX50T, SX95T, FX70T oder FX100T kann auf diesem Modul als Haupt-FPGA eingesetzt werden (vgl. Abbildung 5.2) und bietet Platz zur Implementierung des Benutzerentwurfs. Es stehen außerdem bis zu vier Gigabyte DDR3-Speicher zur Verfügung. Neben dem Haupt-FPGA ist ein Virtex-5-LX30T auf dem Modul untergebracht. Auf diesem FPGA ist ein PCI-Express-Endpunkt implementiert, der das Modul mit dem RAPTOR-XPress und dem PC koppelt. Da in dieser Arbeit die Evaluation der Kommunikation im Vordergrund steht, werden insbesondere folgende Schnittstellen des DB-V5 evaluiert.

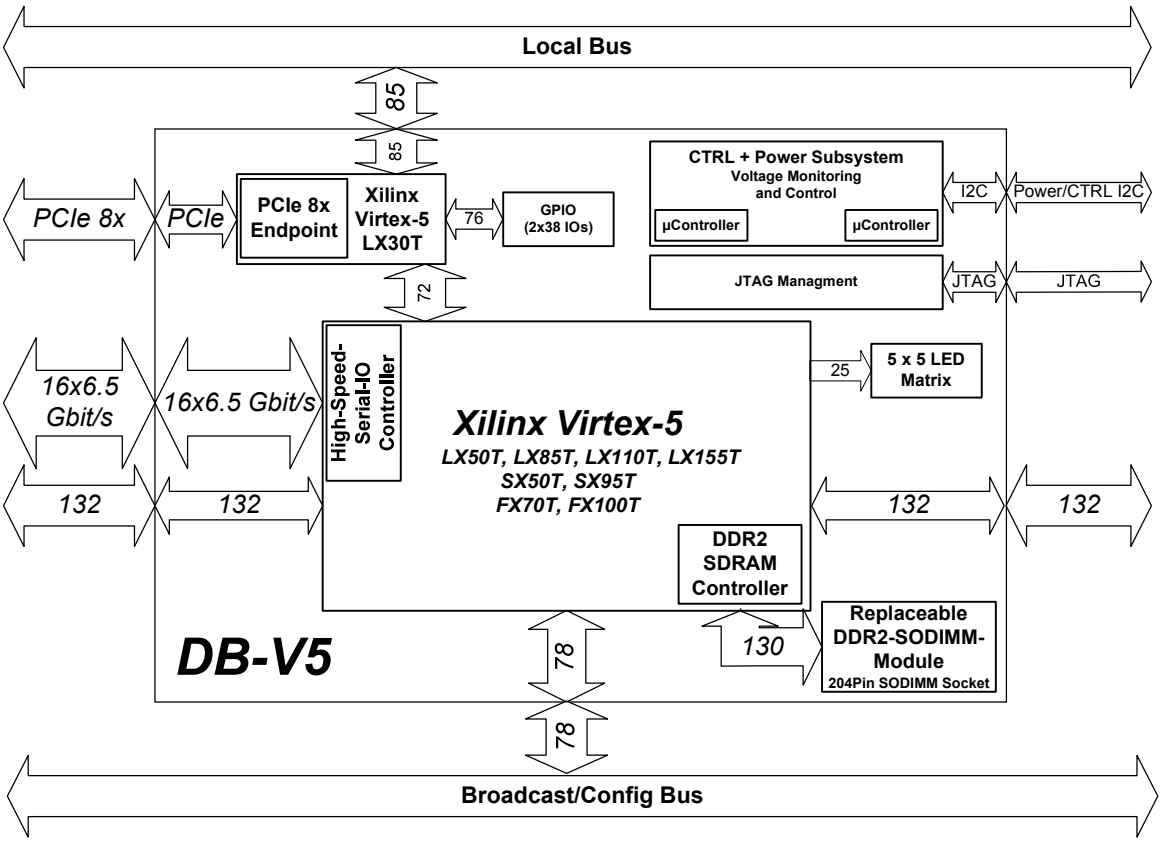

Abbildung 5.2: Der schematische Aufbau des DB-V5 FPGA-Moduls.

- 36 differentielle LVDS-Verbindungen zwischen dem Haupt-FPGA und dem PCI-Express Endpunkt mit einer Datenrate von 1,25 Gbit/s.
- Eine 8x PCI-Express Verbindung der zweiten Generation zwischen RAPTOR-XPress und DB-V5 mit einer aggregierten Datenrate von 40 Gbit/s.
- Eine 85 Bit breite Anbindung an den Lokalbus des RAPTOR-XPRESS, bestehend aus 32 Datensignalen, 32 Adresssignalen und 11 Kontrollsignalen.
- 16 GTX-Transceiver für serielle Punkt-zu-Punkt-Verbindungen mit bis zu 6,5 Gbit/s.
- Je 66 differentielle LVDS-Verbindungen zum linken und rechten Nachbarmodul des DB-V5 mit einer Datenrate von 1,25 Gbit/s.
- 39 unidirektionale und differentielle LVDS-Verbindungen zum Broadcastbus des RAPTOR-XPress mit einer Datenrate von 1,25 Gbit/s pro Signal.

Das RAPTOR-XPress Rapid-Prototyping-System und die zugehörige Plattform

Das modulare, FPGA-basierte Rapid-Prototyping-System RAPTOR-XPress dient zur Aufnahme, Kopplung und Versorgung verschiedener Tochtermodule (vgl. Abbildung 5.3). Im Falle des FPGA-Clusters werden die bereits erwähnten DB-V5-Module verwendet. Das RAPTOR-XPress-Basismodul kann modular mit bis zu vier FPGA-Modulen bestückt werden und stellt umfangreiche Funktionen für Systemmanagement und Kommunikation zur Verfügung. Die Verbindung zum Rechner ist über acht PCI-Express (PCIe) 2.0 Kanäle realisiert. Über einen PCIe-Switch auf dem RAPTOR-XPress-Basismodul ist es möglich, die volle Bandbreite von 32 GBit/s direkt an jedem Modul zu nutzen. Eine PCIe-zu-Lokalbus-Schnittstelle stellt sicher, dass alle bisherigen RAPTOR-Module, die keine PCIe-Schnittstellen besitzen, weiter genutzt werden können. Zusätzlich stehen Schnittstellen für USB 2.0 High-Speed und Gigabit Ethernet zur Verfügung. Zur Kommunikation zwischen Modulen stellt das RAPTOR-XPress-Basismodul Verbindungen zwischen benachbarten Modulen bereit, die in einer Ringtopologie eine Bandbreite von 80 GBit/s bei einer Latenz von unter 10 ns erlauben. Über ein zentrales Switch-FPGA auf dem Basismodul können beliebige Module zudem mit einer Bandbreite von 10 GBit/s kommunizieren. Für die Kommunikation zwischen mehreren RAPTOR-XPress-Basismodulen werden serielle High-Speed-Verbindungen verwendet. RAPTOR-XPress stellt zu jedem Modul eine Schnittstelle mit 16 seriellen High-Speed-Kanälen (Vollduplex, 11 GBit/s) bereit, für die Kommunikation zwischen verschiedenen Basismodulen stehen weitere 64 Vollduplex-Verbindungen (11 GBit/s) zur Verfügung. Die Topologie dieser Kommunikationsinfrastruktur kann zur Laufzeit über einen integrierten 128x128 Crosspoint-Switch (Schaltmatrix) (1408 GBit/s akkumulierte Bandbreite) verändert werden.

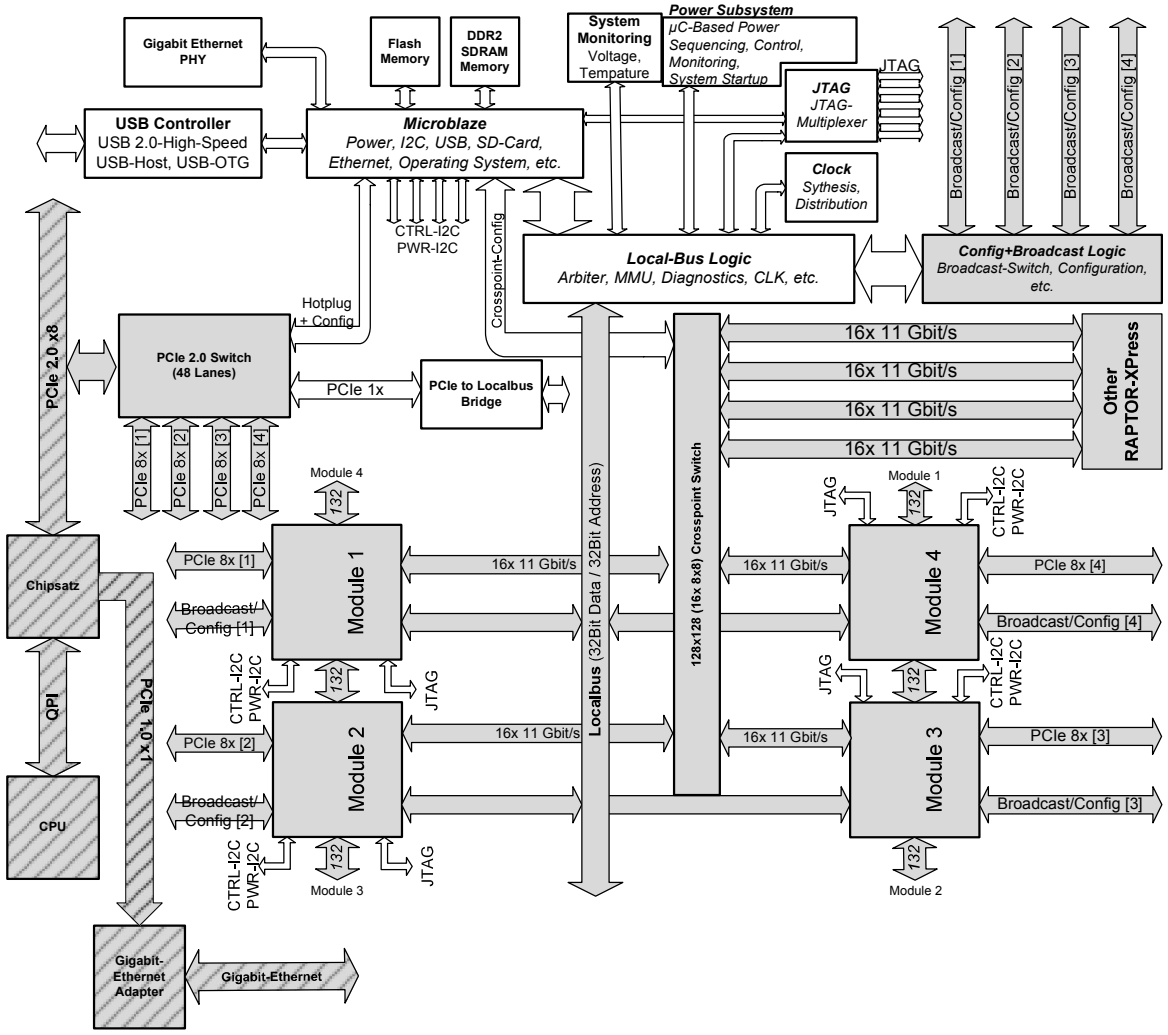

Abbildung 5.3: Der schematische Aufbau des RAPTOR-XPress Rapid-Prototyping-Systems und der Anbindung an den Prozessor.

Als Basis dient ein System auf Basis eines Intel Core-i7-960 Prozessors. Dieser ist über eine QPI-Verbindung mit dem Chipsatz verbunden, welcher wiederum über eine 8x PCI-Express Verbindung der zweiten Generation mit dem RAPTOR-XPress kommuniziert. In Verbindung mit den DB-V5-Modulen und der Anwendung des RAPTOR-XPress im FPGA-Cluster sind mehrere Kommunikationsverfahren implementiert worden.

- QPI-Verbindung zwischen Prozessor und Chipsatz mit einer Datenrate von 96 Gbit/s.
- 8x PCI-Express Verbindung der zweiten Generation zwischen Chipsatz und RAPTOR-XPress.
- 1x PCI-Express Verbindung der ersten Generation zwischen Chipsatz und Gigabit-Ethernet Adapter.

- Gigabit-Ethernet Verbindung des PCs mit einem Switch.
- Crosspoint Switch mit acht seriellen Verbindungen mit einer Breite von jeweils acht Bit und einer Datenrate von 6,5 Gbit/s zur Inter-FPGA-Kommunikation innerhalb des RAPTOR-XPress und zwischen mehreren RAPTOR-XPress-Modulen.
- Fünf PCI-Express-Verbindungen der zweiten Generation mit jeweils acht logischen Kanälen zur Kommunikation zwischen den FPGA-Modulen und dem Heimatsystem mit einer aggregierten Datenrate von 40 Gbit/s.
- 39 unidirektionale, differentielle LVDS-Verbindungen des geschalteten Broadcastbusses des RAPTOR-XPress mit einer Datenrate von 1,25 Gbit/s pro Signal.

Das Nallatech FSB-Modul und die zugehörige PC-Plattform

Um eine enge und latenzarme Kopplung ausgewählter FPGAs im Verbund mit dem jeweiligen Prozessor zu ermöglichen, werden spezielle Plattformen der Firma Nallatech eingesetzt. Diese Plattformen integrieren ein FPGA-System in ein Heimatsystem. Das FPGA-System ist dabei direkt auf einem freien CPU-Sockel des Heimatsystems angebracht und über den Frontside-Bus mit der CPU verbunden. Diese Kopplung übernimmt ein Virtex-5-LX110T FPGA (vgl. Abbildung 5.4).

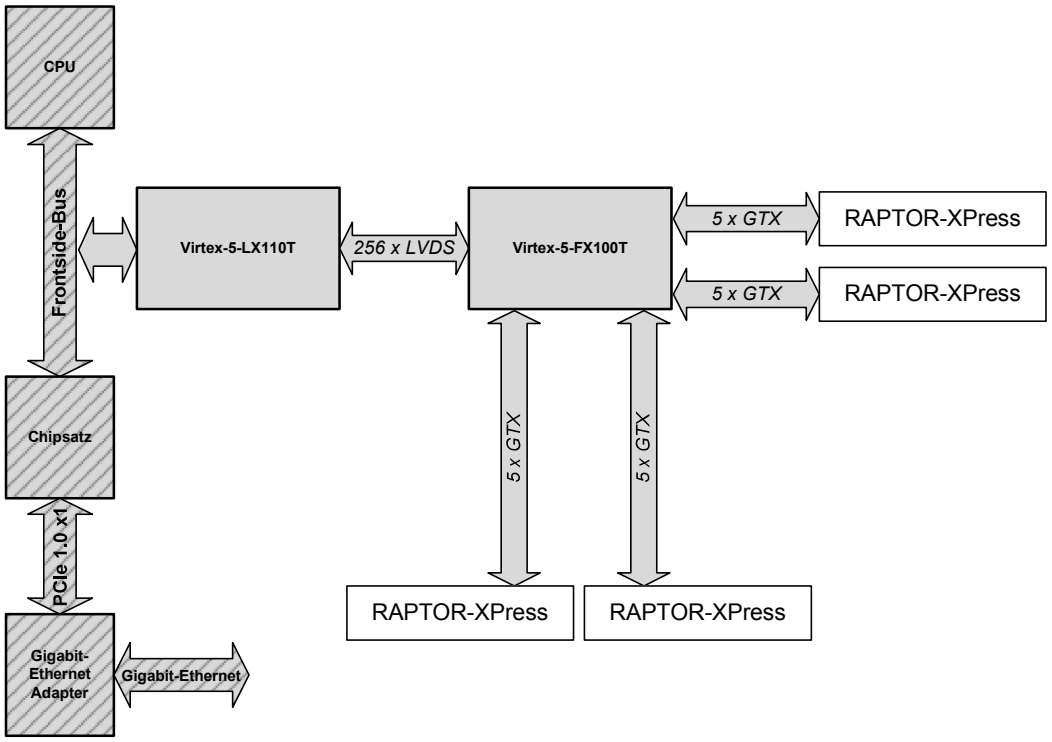

Abbildung 5.4: Der schematische Aufbau eines Nallatech-FSB-Systems.

Zur Implementierung von Nutzerentwürfen ist ein Virtex-5-FX100T FPGA mit dem anderen FPGA über LVDS-Verbindungen gekoppelt. Außerdem stellt dieser FPGA 20 differentielle Verbindungen auf Basis von GTX-Transceivern zur Verfügung, welche das Nallatech-System mit dem Verbund koppeln. Die Verbindungen zwischen dem FPGA im Nallatech System und den FPGAs auf den RAPTOR-XPress-Modulen können über eine dedizierte Platine mit einem integrierten Crosspoint-Switch dynamisch geschaltet werden. Hierbei besteht die Möglichkeit, die Hochgeschwindigkeitsverbindungen des Nallatech-Systems auf bis zu vier RAPTOR-XPress-Modulen zu verteilen. Das HeimatSystem verwendet Intel Xeon 7140 Prozessoren. Zur Evaluation der Kommunikation sind insbesondere folgende Verbindungen interessant.

- Frontside-Bus-Verbindung zwischen FPGA-Modul, Prozessor und Chipsatz mit einer Datenrate von 68,5 Gbit/s.
- 1x PCI-Express-Verbindung der ersten Generation zwischen Chipsatz und Gigabit-Ethernet Adapter.
- Gigabit-Ethernet-Verbindung des PCs mit einem Switch.
- 128 LVDS-Verbindungen in jede Richtung zwischen dem Virtex-5-LX110T und dem Virtex-5-FX100T FPGA mit einer Datenrate von jeweils 1,25 Gbit/s.
- 20 GTX-Verbindungen zwischen dem Nutzer-FPGA und dem Crosspoint-Switch mit jeweils 6,5 Gbit/s Datenrate.
- 20 GTX-Verbindungen zwischen dem Crosspoint-Switch und dem FPGA-Cluster mit jeweils 6,5 Gbit/s Datenrate.

Evaluation der Kommunikation

Bei dem FPGA-basierten System des Fachgebietes Schaltungstechnik und des Fachgebiets Kognitronik und Sensorik können exakte Aussagen über weite Teile der zur Kommunikation verwendeten Komponenten getroffen werden, da diese auf den in dieser Arbeit betrachteten Transceivern basieren. Um den Anteil der Kommunikation an der Gesamtverlustleistung des FPGA-Clusters zu berechnen, wird die Leistungsaufnahme aller Komponenten bestimmt und aufsummiert. Anschließend werden die in Kapitel 3 evaluierten Werte für die verschiedenen Kommunikationsverfahren betrachtet. So ergeben sich die Gesamtleistungsaufnahme und die Leistungsaufnahme aller Kommunikationsverfahren. Bei der Evaluierung der Komponenten wie CPU, Speicher usw. werden jeweils die in der Literatur verfügbaren Werte zum Maximalverbrauch verwendet. Die Leistungsaufnahme der verwendeten FPGAs wird durch den *Xilinx Power Analyzer* bestimmt. Der Anteil der Kommunikation an der gesamten Verlustleistung wird anhand eines Szenarios ermittelt, welches die Simulation eines großen Schaltungsentwurfs darstellt. Die Partitionierung des Entwurfs impliziert eine maximale Auslastung der FPGAs. Durch die enge Kopplung zwischen den Entwurfspartitionen sind die entsprechenden Kommunikationswege ebenfalls ausgelastet. Für den Großteil der beteiligten Komponenten kann für die Evaluation der Wert für die maximale Verlustleistung genutzt werden. Lediglich auf den FSB-Plattformen muss die Leistungsaufnahme im Leerlauf für die Festplatten und die Grafikkarte gewählt werden. Die Kopplung zwischen den einzelnen PCs mit Gigabit-Ethernet dient nur zur Überwachung und Steuerung des Gesamtsystems, deshalb findet auf dieser Kommunikationsstrecke keine andauernde Aktivität statt und auch wird nicht die maximale Leistungsaufnahme erreicht. Zur Überwachung und Steuerung des Systems wird eine durchschnittliche Auslastung von 10 % der verfügbaren Bandbreite angenommen. Bei einer maximalen Leistungsaufnahme von 975 mW während der Übertragung und 20,4 mW in Übertragungspausen (siehe Kapitel 3.4) kann die durchschnittliche Leistungsaufnahme dieser Kommunikationsstrecke mit 115,9 mW angegeben werden. Tabelle 5.1 gibt die Leistungsaufnahme aller Komponenten im Verbund an und stellt sie dem Kommunikationsanteil gegenüber. Die Ergebnisse verdeutlichen die bisherigen Erkenntnisse der Energieeffizienz von Rechenarchitekturen. Hochoptimierte Architekturen wie FPGAs benötigen im Vergleich zu Standardprozessoren relativ wenig Energie, jedoch nimmt die Kommunikation einen großen Teil der Gesamtverlustleistung ein. Dieser liegt bei dem DB-V5 FPGA-Modul bei 60 % und bei dem RAPTOR-XPress Rapid-Prototyping-System mit vier FPGA-Platinen sogar bei zwei Dritteln (68 %) der Gesamtverlustleistung. Wenn leistungsfähige Komponenten mit hoher Verlustleistung verwendet werden, sinkt der Verlustleistungsanteil der Kommunikation drastisch. Wenn jedoch ein Standard-System mit FPGA-Hardwarebeschleunigern ausgestattet wird, um eine höhere Recheneffizienz zu erreichen, stellt die Kommunikation

einen erheblichen Kostenfaktor in der Leistungsaufnahme dar. Dies gilt sowohl bei einzelnen Systemen wie dem RAPTOR-XPress-PC mit 32,7 % Kommunikationsanteil als auch für Clustersysteme mit gekoppelten FPGA-Hardwarebeschleunigern wie dem SCT-FPGA-Cluster mit 29,8 % Verlustleistungsanteil der Kommunikation. Als vergleichendes Beispiel kann das Win-HPC-System aus Kapitel 4.2 dienen, bei dem der Verlustleistungsanteil der Kommunikation bei ca. 5,3 % liegt.
Während der SCT-Cluster durch die konsequente Nutzung von Hardwarebeschleunigern ein sehr energieeffizientes System darstellt, verwendet der Großteil der weltweit betriebenen Multiprozessorsysteme ausschließlich Standardprozessoren. Das liegt unter anderem in der einfacheren Art einen Standardprozessor zu programmieren als einen Hardwarebeschleuniger. In den letzten Jahren wurden jedoch auch zunehmend mehr Grafikkarten zur Beschleunigung bestimmter Berechnungen eingesetzt. Dies führt auch zur Entwicklung neuer Softwarewerkzeuge zur Einbindung von Hardwarebeschleunigern. Zukünftig kann davon ausgegangen werden, dass konventionelle Systeme mit Standardprozessoren als alleinige Rechenkomponenten, durch mehr und mehr Hybridsysteme mit integrierten Hardwarebeschleunigern ersetzt werden.

Tabelle 5.1: Der Kommunikationsanteil and der Verlustleistung des SCT-FPGA-Clusters.

	Komponente	Leistung	Komponente	Leistung
	Gesamt		**Kommunikationsanteil**	
DB-V5			36x LVDS PCIe-FPGA	2,8 W
			85x LVTTL Lokalbus	0,2 W
			8x PCIe 2.0	2,4 W
	V5-FX100T	15,3 W	66x LVDS Links-Rechts	5,1 W
	V5-LX30T	6,1 W	16x GTX	2,7 W
	Speicher	3 W	19x LVDS Broadcast	1,5 W
		24,4 W		**14,7 W (60,1 %)**
RAPTOR				
	CP-Switch	6,9 W	CP-Switch	6,9 W
	Kontroll-FPGAs	9 W	4x 19x LVDS Broadcast	5,9 W
	Management	10 W	PCIe-Switch	11,9 W
	4x DB-V5	97,6 W	4x DB-V5	58,7 W
		123,5 W		**83,4 W (67,5 %)**
FSB-Plattform	Xeon 7140 CPU	75 W		
	4x Hauptspeicher	30 W		
	2x HDD	10 W		
	Grafikkarte	15 W	2x PCIe 1.0	1,0 W
	V5-LX110T	11,7 W	20x GTX	4,0 W
	V5-LX200T	23,3 W	256x LVDS	19,7 W
	FPGA-RAM	3 W	3x Frontside-Bus	6,5 W
	FSB-MGT-Bridge	6,9 W	FSB-MGT-Bridge	6,9 W
	Wirkungsgrad NT	85 %	2x Gigabit-Ethernet	0,232 W
		205,7 W		**38,3 W (18,6 %)**
RAPTOR-PC	Intel Core-i7 960	85 W		
	6x Hauptspeicher	30 W	2x QPI	10,5 W
	2xHDD	10 W	8x PCIe 2.0	2,4 W
	Grafikkarte	15 W	2x PCIe 1.0	1,0 W
	1x RAPTOR-XPress	123,5 W	1x RAPTOR-XPress	83,4 W
	Wirkungsgrad NT	85 %	2x Gigabit-Ethernet	0,232 W
		298,2 W		**97,5 W (32,7 %)**
Verbund	Gigabit Switch	70 W		
	8x FSB-Plattform	1645,9 W	8x FSB-Plattform	306,5 W
	16x XPress-PC	4771,8 W	16x XPress-PC	1560 W
		6487,7 W		**1936,4 W (29,8 %)**

5.2 RECS - Ein Ressourceneffizienter Cluster-Server

Die Erkenntnisse der vorhergehenden Kapitel machen deutlich, dass bei der Entwicklung von energieeffizienten Multiprozessorarchitekturen die Kommunikation zwischen den Systemkomponenten beachtet werden muss. Die Verbesserung der Energieeffizienz wurde bisher hauptsächlich durch die Optimierung von Prozessoren und der Nutzung von Hardwarebeschleunigern erreicht. Wie die Kapitel 4.2 und 5.1 zeigen, kann der Anteil der Kommunikation an der gesamten Verlustleistung nicht vernachlässigt werden, sondern trägt signifikant zur Leistungsaufnahme eines Systems bei.
RECS, ein ressourceneffizienter Rechnerverbund-Server, wurde im Rahmen dieser Arbeit in Zusammenarbeit mit der Christmann Informationstechnik + Medien GmbH & Co KG vom Fachgebiet Schaltungstechnik der Universität Paderborn und dem Fachgebiet Kognitronik und Sensorik der Universität Bielefeld entwickelt. Hierbei wurden die Erkenntnisse aus den vorhergehenden Kapiteln konsequent umgesetzt und eine Multiprozessorarchitektur realisiert, die mit einer angestrebten Energieeffizienz von 480 Megaflop/s (siehe Tabelle 5.5) pro Watt im internationalen Vergleich in der Spitzengruppe energieeffizienter Supercomputer liegt. Im Gegensatz zu aktuellen Supercomputern ist RECS auf niedrige Energieaufnahme sowie geringe Abmessungen und Kosten ausgelegt. Mit 18 Rechenknoten (zwei bis vier Rechenkerne pro Knoten) in einem 19 Zoll Gehäuse mit nur einer Höheneinheit (vgl. Abbildung 5.5) ist RECS deutlich kompakter als die bisher führenden Blade-Systeme und bietet eine Leistungsfähigkeit von über 400 Gigaflop/s pro 19 Zoll Einschub. Um in den Leistungsbereich der aktuellen Liste der TOP 500 Supercomputer vorstoßen zu können, käme der neue Cluster-Server mit der Baugröße eines üblichen 19-Zoll-Schrankes aus.
Ein RECS-System ist mittlerweile im Höchstleistungsrechenzentrum Stuttgart im Einsatz, wo das später vorgestellte Überwachungssystem zur Effizienzanalyse von HPC-Systemen eingesetzt wird [R43]. Im Rahmen des Programmes Zentrale Innovationsförderung Mittelstand (ZIM) des Bundeswirtschaftsministeriums wurde RECS als Erfolgsbeispiel ausgezeichnet.

5.2.1 Beschreibung der RECS-Systemarchitektur

RECS wurde als ressourceneffiziente Multiprozessorarchitektur entwickelt. Der Begriff Ressourceneffizienz beschreibt dabei einen geringen Platzbedarf, eine niedrige Leistungsaufnahme und niedrige Kosten. Die Basis des RECS-Clusters bilden Einplatinencomputer auf Basis von COM-Express-Modulen [R39]. Diese Module beinhalten außer einem Massenspeichermedium eine komplette X86-Plattform. Über eine standardisierte Schnittstelle werden alle Signale zum Betrieb der Plattform zur Verfügung gestellt. RECS bringt in einem 19-Zoll-Gehäuse 18 Leiterplatten für die Kopplung von COM-Express-Platie unter.

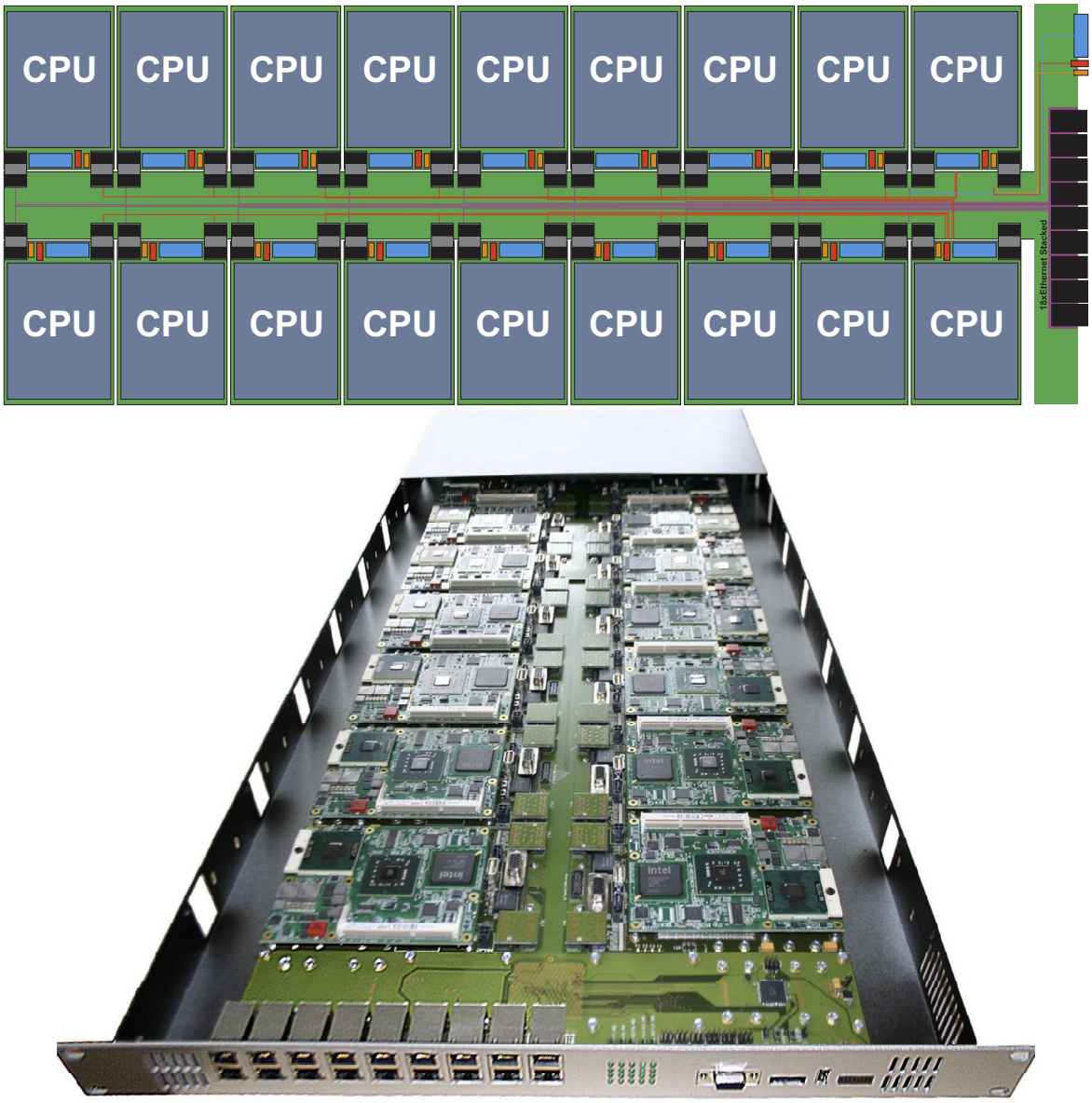

Abbildung 5.5: Schematische Darstellung des RECS-Systems und Abblidung eines RECS-Einschubs mit 18 Rechenknoten.

Über diese Leiterplatten werden die Einzelplatinenrechner mit Strom versorgt, gesteuert und in das Kommunikationsnetzwerk eingebunden. Auf den Trägerplatinen sind zusätzlich Sensoren zur Messung der Leistungsaufnahme untergebracht, wodurch die Evaluation der Leistungsaufnahme der Einplatinencomputer in verschiedenen Szenarien ermöglicht wird. Durch die standardisierten Schnittstellen können unterschiedliche Prozessorfamilien ins RECS-System integriert werden. Hierzu gehören beispielsweise X68-Plattformen mit Core-i7 Prozessoren, Core2Duo-Prozessoren, Pentium-M-Prozessoren oder Atom-Prozessoren. Auf diese Weise kann ein RECS-System je nach geforderter Leistungsfähigkeit mit geeigneten Modulen bestückt werden. Eine gemischte Bestückung mit unterschiedlichen COM-Express-Modulen ist ebenfalls möglich. So kann immer eine möglichst optimale Lösung bezüglich der Leistungsfähigkeit oder der Energieeffizienz realisiert werden. Die Anbindung an das Netzwerk erfolgt über eine zentrale Backplane, die alle benötigten Kommunikationssignale gesammelt an der Front des Gehäuses zur Verfügung stellt. Aufgrund der Zielsetzungen bezüglich niedriger Leistungsaufnahme und Kosten wurde Gigabit-Ethernet in Form von 1000Base-T als Kommunikationsmedium für die erste Realisierung des RECS-Clusters gewählt, da es zudem von allen COM-Express-Modulen direkt zur Verfügung gestellt wird. Eine RECS-Variante für den Bereich des Hochleistungsrechnens setzt Infiniband ein und befindet sich derzeit in der Entwicklung [B3].
Cluster-Systeme, wie in Kapitel 4.2 vorgestellt bestehen aus vielen einzelnen Komponenten wie Computern, rekonfigurierbaren Hardwarebeschleunigern usw.. All diese Komponenten müssen überwacht und gesteuert werden, um das ordnungsgemäße Verhalten des Gesamtsystems sicherzustellen. Eine solche Überwachung wird üblicherweise durch das periodische Abfragen jedes Knotens bezüglich seiner Performanzmetriken wie CPU- oder Speicherauslastung umgesetzt. Bei sehr großen Systemen mit Tausenden von Rechenknoten kann diese Abfrage einen Großteil [R43] der Kommunikationsbandbreite beanspruchen und dadurch die Effizienz des gesamten Systems negativ beeinflussen. Dies impliziert oft eine Erhöhung der durchschnittlichen Leistungsaufnahme des Systems. Um eine feingranulare Überwachung aller Systemkomponenten ohne Beeinflussung der Systemkommunikation zu ermöglichen, wurde ein in das RECS-System eingebettetes Überwachungssystem entwickelt. Die integrierte Monitoring-Funktionalität von RECS ermöglicht die dezentrale Erfassung aller wichtigen Kenngrößen der Recheneinheiten. Diese Lösung ist so weit optimiert worden, dass sie minimal-invasive Messungen im laufenden System erlaubt, die bei kleinstmöglichem Energiebedarf weder die Rechenleistung des Systems noch die für den Anwender zur Verfügung stehende Kommunikationsbandbreite reduzieren. Ein Patentantrag [R1] für dieses Verfahren wurde eingereicht. Realisiert wird dieses System über ein dediziertes Monitoring-Netzwerk mit minimaler Verlustleistung, das in die Gesamtarchitektur eingebettet ist. Das Netzwerk besteht aus einer Vielzahl von Mikrocontrollern, welche unabhängig von der eingesetzten Computerhardware und Software alle wichtigen

Parameter und Größen des Gesamtsystems überwachen und über eine gemeinsame, dedizierte Schnittstelle zur Verfügung stellen. Die einzelnen Mikrocontroller sind über einen gemeinsamen I2C-Bus mit einem zentralen Controller (Atmel ATMEGA) verbunden (vgl. Abbildung 5.6). Dieser Mikrocontroller arbeitet als Master und hat Zugriff auf alle Funktionen und Parameter, welche die einzelnen Mikrocontroller auf den Basismodulen zur Verfügung stellen. Der zentrale Mikrocontroller fragt regelmäßig alle auf der vorherigen Seite genannten Parameter der Rechenknoten ab und stellt sie gesammelt zur Verfügung. Zusätzlich überwacht er den Status des Gesamtsystems, wie beispielsweise den Pegel der Versorgungsspannung oder die Gesamtleistungsaufnahme, und stellt diese Informationen ebenfalls zur Verfügung. Dieses Verfahren setzt die in Kapitel 4 gewonnenen Erkenntnisse des Energiebedarfs von Busstrukturen konsequent um. Auf den Mikrocontrollern (Atmel ATMEGA) läuft eine vom eigentlichen Computersystem unabhängige Firmware, welche folgende Parameter der einzelnen Rechenknoten überwacht und steuert:

- Betriebszustand des Com-Express-Moduls (Power On/Off, Reset, Suspendmodi 3 bis 5, diverse Signale zum Aufwecken/ Herunterfahren/ Alarmieren des Rechenknotens).
- Temperatur des Trägermoduls über lokalen Sensor.
- Stromaufnahme des Rechenknotens und damit auch Leistungsaufnahme über lokale Sensoren.
- Lüfterdrehzahl.
- Status der Netzwerkschnittstelle wie Verbindungsstatus, Geschwindigkeit, und Aktivität.
- Prozessorauslastung, Spannungen auf dem Com-Express-Modul usw. über den System-Management Bus des Com-Express-Modul, welcher an den Mikrocontroller angebunden ist.
- Weitere Möglichkeiten zur Kommunikation zwischen Com-Express-Modul und Mikrocontroller sind gegeben durch einen I2C-Bus und mehrere frei verwendbare Ein- bzw. Ausgänge.

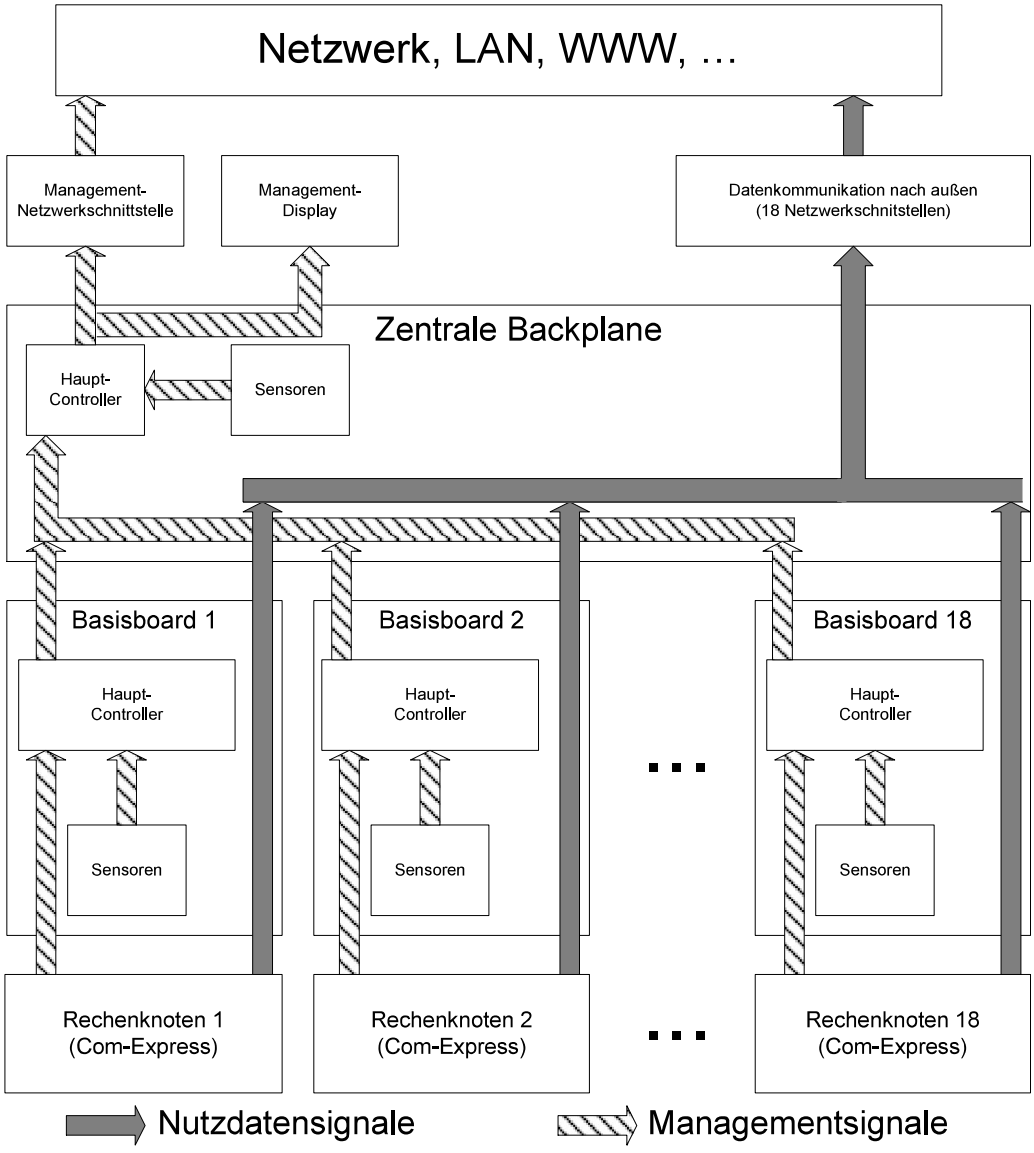

Abbildung 5.6: Der hierarchische Aufbau des Management-Systems.

Die Benutzerschnittstelle nach Außen ist auf zwei Arten realisiert worden:

- Ein LC-Display informiert direkt am System über alle wichtigen Parameter und bietet einem lokalen Anwender die Möglichkeit zur Überwachung und Steuerung des Systems und seiner Einzelkomponenten.
- Ein an den zentralen Mikrocontroller angeschlossener Netzwerkanschluss bietet die Möglichkeit, das System von einem beliebigen Ort aus zu überwachen und zu steuern. Ein integrierter und frei konfigurierbarer Webserver ermöglicht es durch simples Aufrufen einer Webseite, alle Parameter abzufragen und beispielsweise das System bzw. einzelne Rechenknoten herunter zu fahren oder zu starten. Dieser Netzwerkanschluss ist dediziert ausgeführt, d.h. er wird nur für Managementzwecke verwendet und belegt deshalb keine Datenressourcen der anderen Netzwerkschnittstellen, deren gesamter Datendurchsatz somit der eigentlichen Rechenanwendung zur Verfügung steht.

Evaluation des eingebetteten Überwachungssystems

In Rechenclustern werden die einzelnen Knoten regelmäßig auf bestimmte Statusinformation hin abgefragt. Die geschieht in konventionellen Systemen über die gleichen Übertagungskanäle, welche auch für die Datenübertagung zwischen den Knoten verwendet wird. Moderne Überwachungslösungen wie NAGIOS fragen jeden Knoten bis zu 500-mal in der Sekunde ab [R28]. Die benötigte Bandbreite steht dabei nicht für die Hauptanwendung zur Verfügung und verringert die Effizienz des Systems. Für die Evaluation des RECS-Überwachungssystems werden verschiedene Daten in Abständen von zwei Millisekunden von jedem Knoten abgefragt. Überwacht wird beispielsweise die Modultemperatur, die Leistungsaufnahme oder der Betriebszustand des Moduls über Hardware-Sensoren. Das in RECS integrierte Überwachungssystem nimmt eine Leistung von ca. 720 mW (19x Atmel Atmega169p, 1x XPORT) auf. Die Leistungsaufnahme des Überwachungssystems ist dabei konstant und unabhängig von der Aktivität. Im Falle einer Knotenüberwachung über das Gigabit-Ethernet-Netzwerk des RECS-Systems muss für jede Statusinformation ein Ethernet-Frame an den Zielknoten gesendet werden und anschließend ein Frame empfangen werden. Für jeden Knoten werden also alle zwei Millisekunden zwei Ethernet-Frames benötigt. Bei einer Paketlänge von maximal 1542 Byte ergibt sich eine Datenmenge von 1505,9 KByte pro Knoten und Sekunde. Wie in Kapitel 3.4.2 ermittelt wurde, besitzt 1000Base-T eine Leistungsaufnahme von 975,7 mW pro Netzwerkadapter. Bei einer Datenübertragung mit einer Rate von 1 Gbit/s wird für die Übertragung eines Pakets ein Zeitraum von 12,3 ms benötigt.

$$T_{mon} = \frac{N_{poll} \cdot L_{frame} \cdot 8 \cdot N_{status} \cdot 2 \cdot N_{nodes}}{10^9} \tag{5.1}$$

Die für die Übertragung der Statutsinformationen zu erbringende Arbeit ist gegeben durch:

$$W_{mon} = \frac{N_{poll} \cdot L_{frame} \cdot 8 \cdot N_{status} \cdot 2 \cdot N_{nodes}}{10^9} \cdot 0,9757 \tag{5.2}$$

mit

- T_{mon}: Die benötigte Zeit zum Abfragen der Statuswerte.
- W_{mon}: Die zusätzlich zu erbringende elektrische Arbeit zum Abfragen der Statuswerte.
- N_{poll}: Die Anzahl der Statusabfragen pro Sekunde.
- L_{frame}: Die Länge des Frames.
- N_{Status}: Die Anzahl der abzufragenden Statusinformationen.
- N_{nodes}: Die Anzahl der abzufragenden Knoten.

Die zusätzlich zu erbringende elektrische Arbeit beträgt im Fall eines vollbesetzten Pakets 12 mWs pro Knoten. Im gesamten RECS-System mit 18 Knoten ergibt sich eine elektrische Arbeit von 216,7 mWs oder eine durchschnittliche Leistungsaufnahme von 216,7 mW für eine abzufragende Statusinformation. In realen Anwendungen kann von einer Paketauslastung von 87 % ausgegangen werden [R11], wodurch sich eine durchschnittliche Leistungsaufnahme von 188,5 mW ergibt. Durch die elektrische Leistungsaufnahme des integrierten Überwachungssystems von 720 mW kann die benötigte Anzahl von abzufragenden Informationen berechnet werden, bei denen sich der zusätzliche Energieaufwand des integrierten Überwachungssystems lohnt.

$$N_{poll} \geq \lceil \frac{P_1}{P_2} \rceil = \lceil \frac{720\,\text{mW}}{188,5\,\text{mW}} \rceil = 4 \tag{5.3}$$

mit

- P_1: Die Leistungsaufnahme des Überwachungssystem.
- P_2: Die zusätzliche Leistungsaufnahme bei Abfrage über das Netzwerk.

Ab einer Menge von 4 abgefragten Werten pro Zyklus lohnt sich das integrierte Überwachungsnetzwerk aus Sicht der Leistungsaufnahme gegenüber einer Abfrage der einzelnen Knoten über das Gigabit-Netzwerk. RECS liest die Statusinformation von Hardwaresensoren aus, ohne die CPU im Programmablauf zu beeinflussen. In konventionellen Clustersystemen wird eine parallel zum Hauptprogramm laufende Software benötigt, welche Systemzeit und Systemressourcen belegt und damit Energie benötigt. Je nach verwendeter Soft- und Hardwarekombination wird der Wert der benötigten Anzahl abzufragender Statuswerte bei RECS also noch geringer ausfallen.

5.2.2 LoneStar

RECS bietet in der aktuellen Version nur eingeschränkte Möglichkeiten zur Nutzung von Massenspeichern in Form von Festplatten. Für viele Anwendungsszenarien wie das Rendern von Grafikdaten wird kein Massenspeicher für jeden Knoten benötigt. RECS kann jedoch beispielsweise auch sehr gut als zentraler Terminalserver für viele Benutzer dienen. Dabei werden oft festplattenbasierte Speichersysteme verwendet, die zentral untergebracht sind und über eine breitbandige Kommunikationsstrecke mit dem restlichen Netzwerk verbunden sind. Solche Speichersysteme stammen aus dem Bereich des Hochleistungsrechnens und müssen sehr große Datendurchsätze verarbeiten. Die Festplatten laufen wegen der Anforderung an geringe Zugriffszeiten zudem dauerhaft. Aus diesen Gründen weisen diese Systeme keine hohe Energieeffizienz auf. Viele Anwendungsszenarien benötigen jedoch keinen so hohen aggregierten Datendurchsatz wie z.B. die Nutzer eines Terminalservers, die zentralisiert auf ihre Dokumente zugreifen. Als weiteres Beispiel gelten Archivsysteme, bei denen selten ein Zugriff erfolgt und nur einzelne Dateien übertragen werden. Zwischen den Zugriffen kann viel Energie durch die Abschaltung der Festplatten gespart werden, dadurch müssen keine kostenintensiven Hochleistungsfestplatten eingesetzt werden. Energieeffiziente Speichersysteme mit entsprechenden Energiesparmaßnahmen sind derzeit nicht erhältlich. LoneStar (Long Term Storage Archive) wurde als energieeffizientes Speichersystems entwickelt. Es kann sowohl als eigenständiges System betrieben werden als auch in Kombination mit RECS.

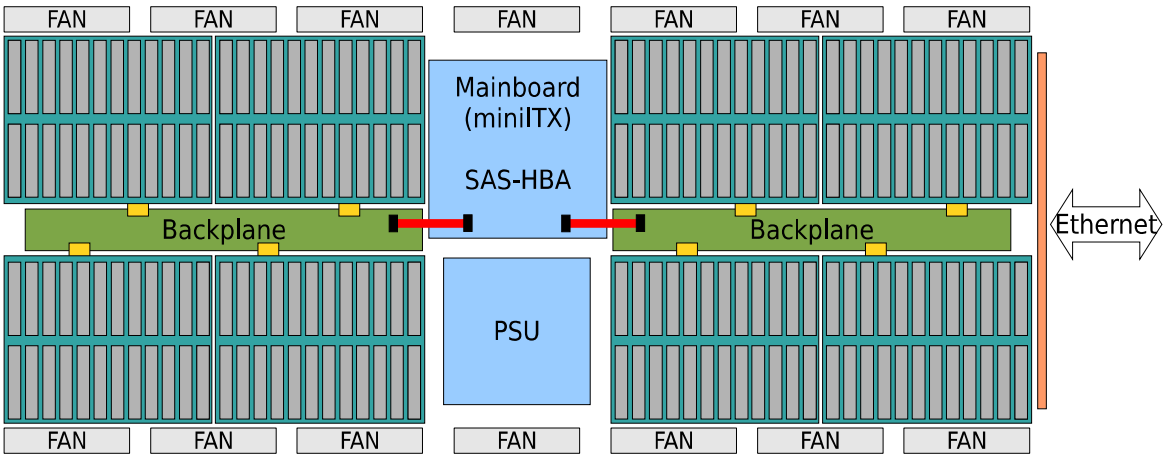

Abbildung 5.7: Eine Übersicht des mechanischen Aufbaus von LoneStar.

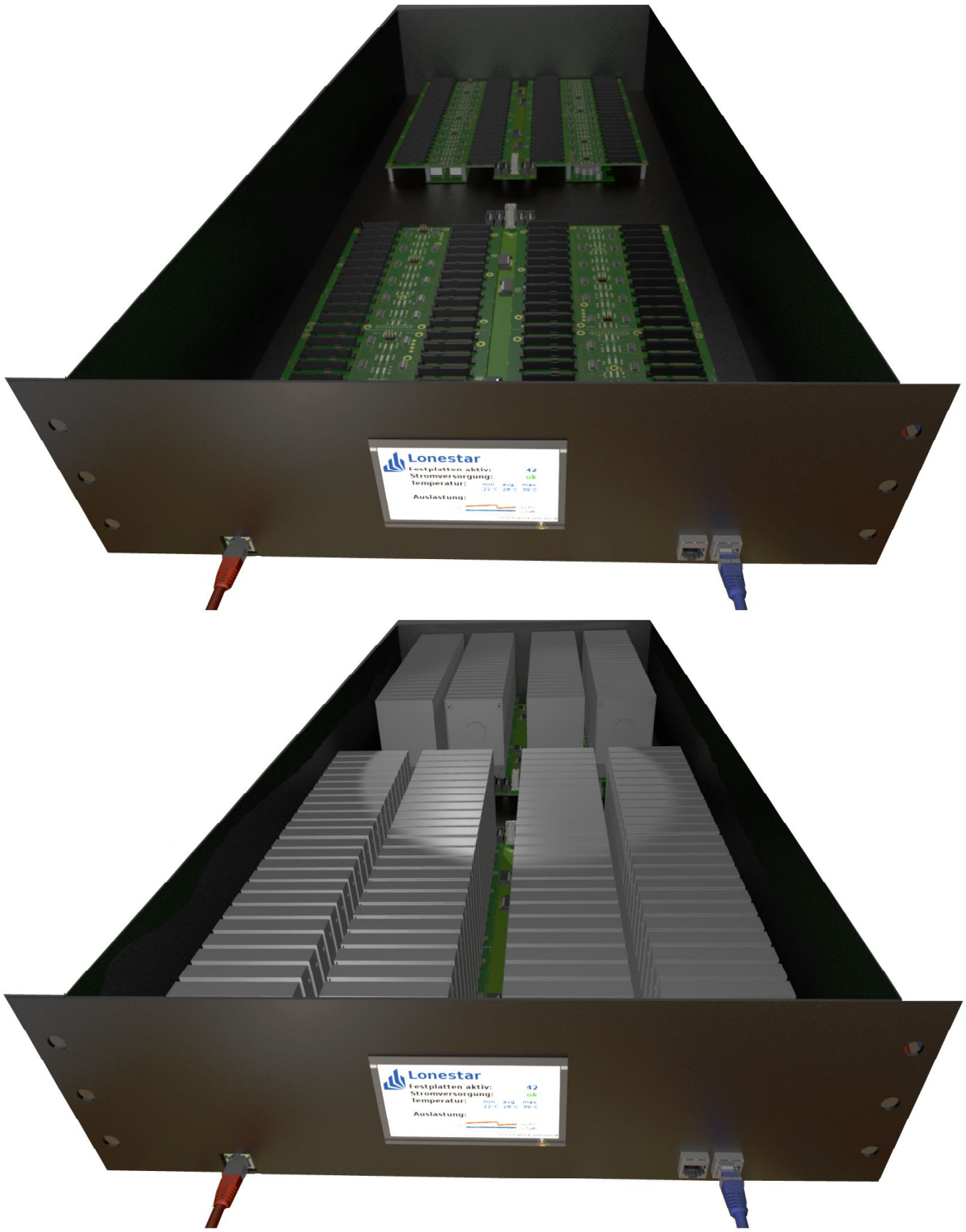

Abbildung 5.8: Zwei Abbildungen des Lonestar-Sytems. Oben: Backplanes ohne Festplatten. Unten: System bestückt mit 192 Festplatten.

Während der Entwicklung wurden Anforderungen des Speichersystems in Bezug auf Bandbreite, Latenz, Stromaufnahme und Kühlung evaluiert. Hierbei wurde eine Anbindung des Systems an die Umwelt mittels zwei dedizierter Gigabit-Ethernet-Verbindungen ausgewählt. Diese Lösung weist eine ausreichende Bandbreite und eine niedrige Leistungsaufnahme auf. Durch die Verwendung energiesparender Komponenten wie 2,5"-Festplatten wird eine sehr hohe Datendichte bei gleichzeitig niedriger Leistungsaufnahme erreicht. Dies ermöglicht außerdem die Verwendung einer einfachen Luftkühlung zum Wärmemanagement des Systems. Um die Ergebnisse der Anforderungsanalyse umzusetzen, wurde das Speichersystem als modularer, skalierbarer Aufbau von Einzelkomponenten entwickelt (vgl. Abbildungen 5.7 und 5.8). Hierbei dient ein 1 Meter tiefes Gehäuse im 19 Zoll Format mit 3 Höheneinheiten als Basis. Im Zentrum dieses Gehäuses ist eine ITX-Hauptplatine als Zentralrechner mit Standard-Komponenten wie Core-I5-Prozessoren untergebracht. Dieser Zentralrechner ermöglicht die Anbindung des Gesamtsystems über zwei Gigabit-Ethernet-Schnittstellen. Die Platzierung des Zentralrechners in der Mitte des Gehäuses ermöglicht die Verwendung von relativ kurzen und weniger komplexen Backplanes zur Anbindung der Festplatten. Vor und hinter dem Zentralrechner befinden sich jeweils 96 SATA-Festplatten mit 1 Terabyte Kapazität im 2,5 Zoll Format, welche über die Backplanes mit dem Rechner verbunden sind. Dies ergibt eine Speicherdichte von 192 Terabyte in drei Höheneinheiten. Jede Festplatte wird schraubenlos mit dem Anschlussfeld nach unten in das System gesteckt, was zusammen mit der Fähigkeit zum Auswechseln im laufenden Betrieb des SAS- und SATA-Standards einen einfachen Austausch der Platten im laufenden Betrieb ermöglicht. Je 24 Festplatten werden über einen Chenbro CK12804 SAS-Expander mit dem Zentralrechner verbunden. Dieser verfügt über einen LSI MegaRAID SAS 9211-8i RAID-Controller mit PCI-Express-Anbindung und 8 SAS-Ports. Mit jedem dieser Ports wird ein SAS-Expander verbunden, so dass sich die Gesamtanzahl von 8*24=192 Festplatten ergibt. Die Signal- und Stromverteilung erfolgt komplett kabellos über die Backplanes und Tochterplatinen, welche jeweils 24 Festplatten und einen SAS-Expander ausschließlich über Steckverbindungen versorgen. Die einzigen Kabel im System verbinden den RAID-Controller und das im Zentrum des Gehäuses sitzende Netzteil mit den Backplanes. Um die Signalintegrität und eine ausreichende Stromversorgung sicher zu stellen, wurden während der Entwurfsphase umfangreiche Simulationen durchgeführt. Mit Hilfe dieser Ergebnisse konnte eine exakte elektrische Dimensionierung der Leiterplatten bestimmt werden, welche mit 8 benötigten Signallagen auskommen. Um die Leistungsaufnahme des Systems zu optimieren, wurden verschiedene Ansätze realisiert. Zum einen wird das gesamte System (mit Ausnahme des Zentralrechners) mit 12 Volt versorgt. Diese Spannung wird erst direkt auf den Tochterplatinen in die von den SAS-Expandern und Festplatten benötigten Werte umgewandelt. Jeweils ein Spannungswandler versorgt acht Festplatten, so dass auf jeder Tochterplatine vier Wandler zum Einsatz kommen (ein Wandler für den SAS-

Expander). Dieses Point-of-Load-Prinzip erhöht die Energieeffizienz des Systems, da die einzelnen Spannungswandler so dimensioniert werden können, dass sie immer im effektivsten Lastbereich mit minimalen Verlusten arbeiten.

Zum anderen verlangen die Anforderungen des Systems als Archivsystem nicht den dauerhaften Betrieb aller Komponenten wie Festplatten oder SAS-Expander. Aus diesem Grund wurde ein Überwachungs- und Steuerungssystem auf Basis eines Mikrocontrollernetzwerks entwickelt (siehe Abbildung 5.9). Dieses System verfügt über diverse Sensoren und Aktoren zur Beeinflussung des Speichersystems. So kann die Temperatur an mehreren kritischen Punkten auf jeder Tochterplatine, die Leistungsaufnahme jeder Festplatte und die Spannungsversorgung überwacht werden. Wenn eine Festplatte über längere Zeit nicht vom Zentralrechner benötigt wird, kann diese explizit vom Steuerungssystem abgeschaltet und von der Spannungsversorgung getrennt werden. Das gleiche gilt auch für jeden der acht SAS-Expander im System. Die Steuerung der Energiesparoptionen des Überwachungssystems kann lokal, per Zentralrechner oder über eine dedizierte Netzwerkschnittstelle ortsunabhängig erfolgen. Die Kopplung zwischen dem Überwachungssystem und dem Zentralrechner erfolgt über eine serielle USB-Verbindung, sodass die Festplatte über Befehle im Betriebssystem heruntergefahren und anschließend von der Spannungsversorgung getrennt werden kann. In der Front des Gehäuses wurde ein berührungssensitives LCD-Modul angebracht, welches direkt mit dem Mikrocontrollernetzwerk gekoppelt ist und die einfache Überwachung und Steuerung aller Systemfunktionen ermöglicht. Eine dedizierte, ebenfalls gekoppelte Netzwerkschnittstelle stellt den Zugriff auf dieselben Funktionen ortsunabhängig sicher. Die so erreichte hohe Energieeffizienz des Archivsystems ermöglicht den Verzicht auf eine aufwändige Flüssigkühlung der Komponenten. Vielmehr können günstige Lüfter eingesetzt werden, um das Wärmemanagement des kompletten Systems sicherzustellen. Die optimale Realisierung des Kühlungskonzeptes wurde mit Hilfe aufwändiger Strömungssimulationen evaluiert. Hierzu wurde jeder der 192 Festplatten ein thermisches Modell zugeordnet und die Wärmeentwicklung in Abhängigkeit der Festplatten- und Lüfteranordnung simuliert. Die optimale Wärmeabfuhr wird in der senkrechten Anordnung von jeweils 4 Festplatten in der Breite eines 19-Zoll-Einschubs erreicht. Als Lüfter kommen hierbei Varianten mit 12 cm Durchmesser zum Einsatz, die genau der Höhe von 3 HE entsprechen und von links nach rechts das System durchlüften. Im Gegensatz zu klassischen Luftkühlungskonzepten von der Stirnseite eines Systems zur Rückseite verkürzt sich so die zu durchströmende Strecke bei gleichzeitiger Verringerung der Wärmequellendichte. Hierdurch können langsam drehende, energiesparende Lüfter eingesetzt werden, welche zudem durch das Überwachungs- und Steuerungssystem je nach Betriebsstatus der Komponenten im Strömungsbereich dynamisch an- oder abgeschaltet werden können.

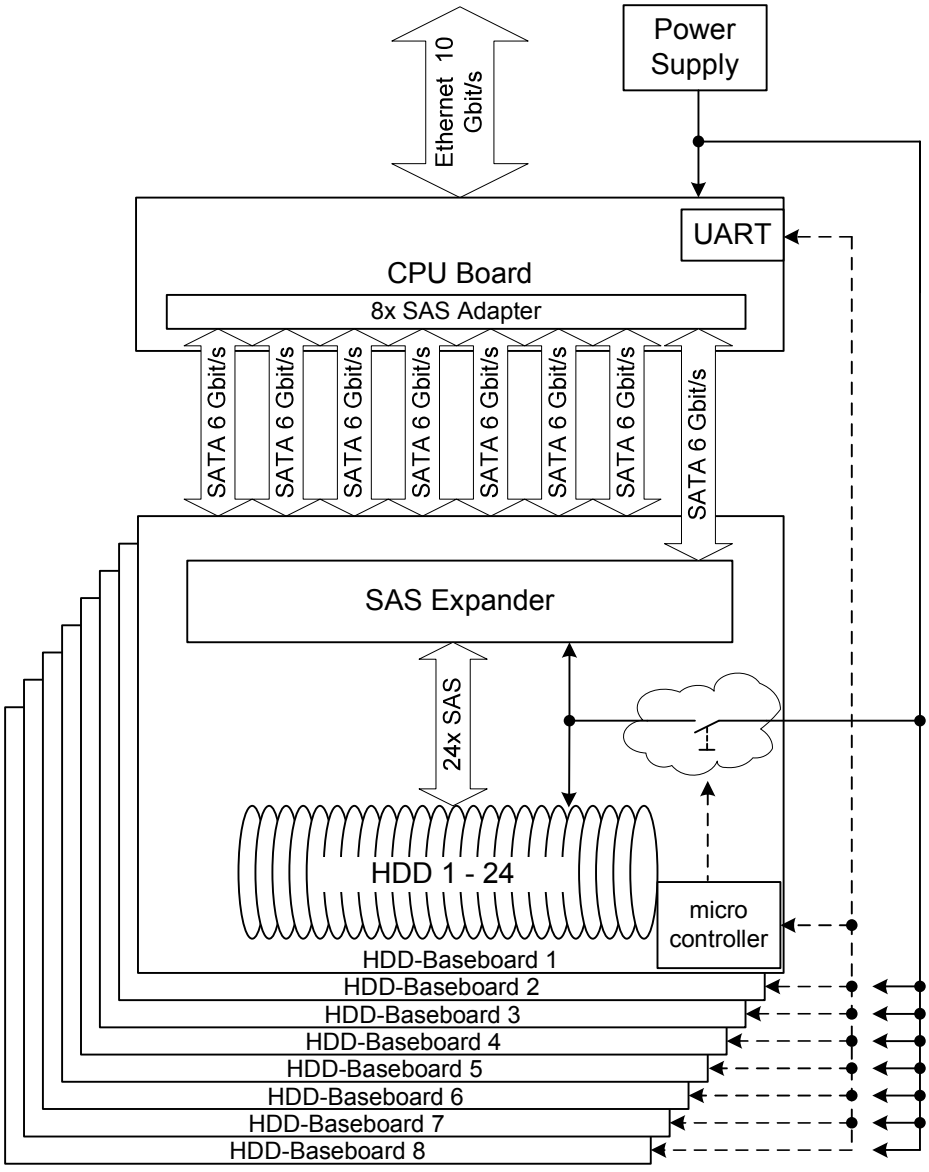

Abbildung 5.9: Der schematische Aufbau des LoneStar-Systems.

Aufgrund der speziellen Hardwarearchitektur und unterschiedlichen Energiespartechniken ist LoneStar eine sehr ressourceneffiziente Lösung. Der Begriff Ressourceneffizienz beinhaltet hier Speicherdichte, Systemkosten und Energieeffizienz. Diese Begriffe werden unter Berücksichtigung des Brutto-Speicherplatzes von 192 TByte und des Netto-Speicherplatzes zwischen 119 TByte und 151 TByte (je nach verwendetem RAID-Verfahren) evaluiert. Zum Vergleich mit existenten Lösungen wird Backblaze [R52], eines der zurzeit effizientesten Speichersysteme, betrachtet. Backblaze nutzt eine Kapazität von 67,5 TByte in einem Gehäuse mit vier Höheneinheiten, also 16,9 TByte pro Höheneinheit. LoneStar ermöglicht in Zusammenhang mit dem verwendeten Raid-Verfahren eine Nutzung von 50,2 TByte pro Höheneinheit. Die Leistungsaufnahme von LoneStar wurde im Bereitschaftsmodus und im Zugriffsmodus des Systems evaluiert. Ein weiteres Szenario bildet die Nutzung von LoneStar als Archivsystem. In diesem Szenario wird angenommen, dass jede Stunde ein Datenstrom von 100 GByte auf das System geschrieben und anschließend gelesen wird. Diese Situation könnte beispielsweise ein stündliches Backup mit anschließender Verifikation der Daten darstellen. Die Stromaufnahme der einzelnen LoneStar-Komponenten ist in Tabelle 5.2 zu sehen.

Komponente	Name	Leistungsaufnahme
Hauptplatine	Jetway NF9E-Q77	42 W (Load) 24 W (Idle)
SAS-Controller	LSI MegaRAID SAS 9211-8i	13,5 W
SAS-Expander	Chenbro CK12804	10 W
Festplatten	WD10TPVT 1 TB	3,6 W (R/W) 1,3 W (Idle) 0,85 W (Standby)

Tabelle 5.2: Die verwendete LoneStar-Hardware.

Das LoneStar-System benötigt im Bereitschaftsmodus 41,7 Watt an elektrischer Leistung. Ein Schreibzugriff erfolgt aufgrund des RAID-Verfahrens immer auf vier Festplatten gleichzeitig und benötigt 82 Watt. Beim Lesen der Daten benötigt LoneStar 76,8 Watt. Das Lesen und Schreiben von 100 GByte dauert 6,66 Minuten bzw. 0,11 Stunden bei einer Anbindung des Systems mit 2 Gbit/s. Da sich das System für 0,78 Stunden im Leerlauf befindet, ergibt sich eine durchschnittliche elektrische Arbeit von 50 Wh. Dieser niedrige Wert wird durch das automatische Abschalten der Festplatten nach den Zugriffen erreicht. Die Ergebnisse der Evaluation im Vergleich mit dem Backblaze-System sind in Tabelle 5.3 zusammengefasst. Die Kombination von stromsparenden 2,5"-Festplatten mit einer höheren Speicherdichte pro System und der Abschaltung von nicht benötigten Platten führt zu den niedrigen Betriebskosten von LoneStar. Im Falle des Archivszenarios liegen diese Kosten um den Faktor 14 unter den Kosten von Backblaze. Wenn ein Szenario gewählt wird, in dem alle Festplatten

dauerhaft laufen, ergibt sich eine Energieersparnis um den Faktor 2 bei Verwendung eines einzelnen Einschubs. Dabei liegt die Speicherdichte von LoneStar um den Faktor 2,8 -mal höher als die Speicherdichte von Backblaze.

Größe	LoneStar	Backblaze
Terabyte / Höheneinheit (Brutto)	64	16,875
Leistung (Leerlauf) / System	42 W	477 W
Leistung (Last) / System	490 W	790 W
Verlustleistung (Kommunikation) / System	8,5 W(1,7 %)	14,3 W(1,8 %)
Arbeit / Jahr (Last)	4287 kWh	6926 kWh
Arbeit / Jahr (Leerlauf)	365 kWh	5936 kWh
Arbeit / Jahr (Archivszenario)	438 kWh	6153,5 kWh
Benötigte Systeme für 1 PByte	7 - 9	18
Arbeit / Jahr (Last, 1 PByte)	38602 kWh	120762 kWh
Arbeit / Jahr (Archivszenario, 1 PByte)	3942 kWh	110764 kWh

Tabelle 5.3: Verschiedene Effizienzterme von LoneStar, verglichen mit Backblaze.

5.2.3 Evaluation der Kommunikation und Vergleich von RECS mit anderen Architekturen

Durch das integrierte Überwachungssystem ist eine Bestimmung der Leistungsaufnahme des RECS-Systems möglich. RECS wird dabei als Rechnerverbund konfiguriert und mit einem *Linpack-Benchmark* geladen. Der *Linpack-Benchmark* ist ein Leistungstest für Supercomputer und lastet alle Prozessoren und Kommunikationswege zwischen den Knoten aus. Hierdurch ist sichergestellt, dass die maximalen Werte für die Leistungsaufnahme der Kommunikationsschnittstellen verwendet werden können. Das RECS-System wurde jeweils in zwei verschiedenen Varianten untersucht. Diese unterscheiden sich nur in den verwendeten Knoten, die Kommunikationsinfrastruktur zwischen den Knoten ist bei beiden Systemen identisch.

- RECS Energiesparvariante
  - 306 GFlop/s (Linpack)
  - 18 Rechenknoten mit jeweils einem Intel Core-2-Duo SP9300 Prozessor, mit dem GM45-Chipsatz über FSB 800 gekoppelt. 8Gbyte DDR3-SODIMM (2x Kingston KVR1066D3S7/4G).
  - Anbindung des Gigabit-Ethernet-Adapters an den Chipsatz über PCIe der ersten Generation mit einer Lane.
  - Kommunikation zwischen den Knoten über Gigabit-Ethernet.

  - Gigabit-Ethernet-Switch
- RECS Hochleistungsvariante
  - 432 GFlop/s (Linpack)
  - 18 Rechenknoten mit jeweils einem Intel Intel Core i7-620LE Prozessor, mit dem Chipsatz über vier DMI Lanes mit jeweils 2,5 GT/s gekoppelt (Evaluiert mit 4x Aurora 2,5 Gbit/s und 8B/10B-Kodierung). 8Gbyte DDR3-SODIMM (2x Kingston KVR1066D3S7/4G).
  - Anbindung des Gigabit-Ethernet-Adapters an den Chipsatz über PCIe der ersten Generation mit einer Lane.
  - Kommunikation zwischen den Knoten über Gigabit-Ethernet.
  - Gigabit-Ethernet-Switch

Zur Überprüfung des verwendeten Szenarios wurde die Stromaufnahme der einzelnen RECS-Knoten mit Intel Core-2-Duo SP9300 Prozessor bei verschiedenen Lastszenarien bestimmt (siehe Abbildung 5.10). Hierbei weist der *Linpack-Benchmark* wie erwartet mit 48,3 Watt die höchste Leistungsaufnahme auf. Anschließend wurden die Angaben aus den Datenblättern zur maximalen Verlustleistung (TDP) der einzelnen Komponenten aus dem COM-Express-Modul summiert und mit dem Ergebnis des Benchmarks verglichen. Die Messung der Leistungsaufnahme erfolgt bei RECS über die zentrale 12 V-Spannungsversorgung. Zu dem berechneten Wert von 40,54 Watt als akkumulierte Leistungsaufnahme müssen noch die Verluste in den Spannungswandlern auf den Hauptplatinen und dem COM-Express-Modul hinzu addiert werden. Bei einer durchschnittlichen Netzteileffizienz von 85 % ergibt sich die berechnete maximale Leistungsaufnahme zu 47,7 Watt. Dieser Wert passt sehr gut zu den Messergebnissen des *Linpack-Benchmark*. Dieser Benchmark beschreibt ein realistisches Lastszenario für RECS und ermöglicht außerdem die Erzeugung von einem Maximum an Verlustleistung. Die in den vorhergehenden Kapiteln ermittelten Werte zur maximalen Verlustleistung können also zur Evaluation verwendet werden. Wie in Tabelle 5.4 zu sehen ist, weist RECS mit einer Verlustleistung von unter 50 W pro Knoten eine sehr geringe Verlustleistung auf. Dies liegt auch an der Nutzung von energiesparenden Kommunikationsverfahren wie dem Frontside-Bus. Genauso wichtig wie die Leistungsaufnahme eines Systems ist jedoch auch seine Leistungsfähigkeit in Form von möglichen Rechenoperationen pro Sekunde. RECS benötigt für eine Performanz von 306 GFlop/s bzw. 432 GFlop/s eine elektrische Leistung von maximal 900 W (siehe Tabelle 5.5). Um eine Rechengeschwindigkeit wie der Arminius-Rechnerverbund (siehe Kapitel 4.2) zu erreichen, reichen 18 RECS-Einschübe aus. Ein einzelner Standard-Serverschrank mit 40 Einschüben würde ausreichen, um den gesamten Rechnerverbund unterzubringen. Zusätzlich würde noch Platz für Speichersysteme wie LoneStar bleiben. Ein komplettes Rechenzentrum könnte so in einem einzelnen Schrank untergebracht werden. Ein

weiteres Merkmal von RECS ist die hohe Recheneffizienz in Form von Rechenoperation pro Watt an Verlustleistung. Mit einer Recheneffizienz von 432 MFlop/W würde RECS in der Liste der 500 energieeffizientesten Supercomputer der Welt den 58. Platz (Juni 2011). In der aktuellsten Liste vom Juni 2012 würde Platz 114 erreicht.

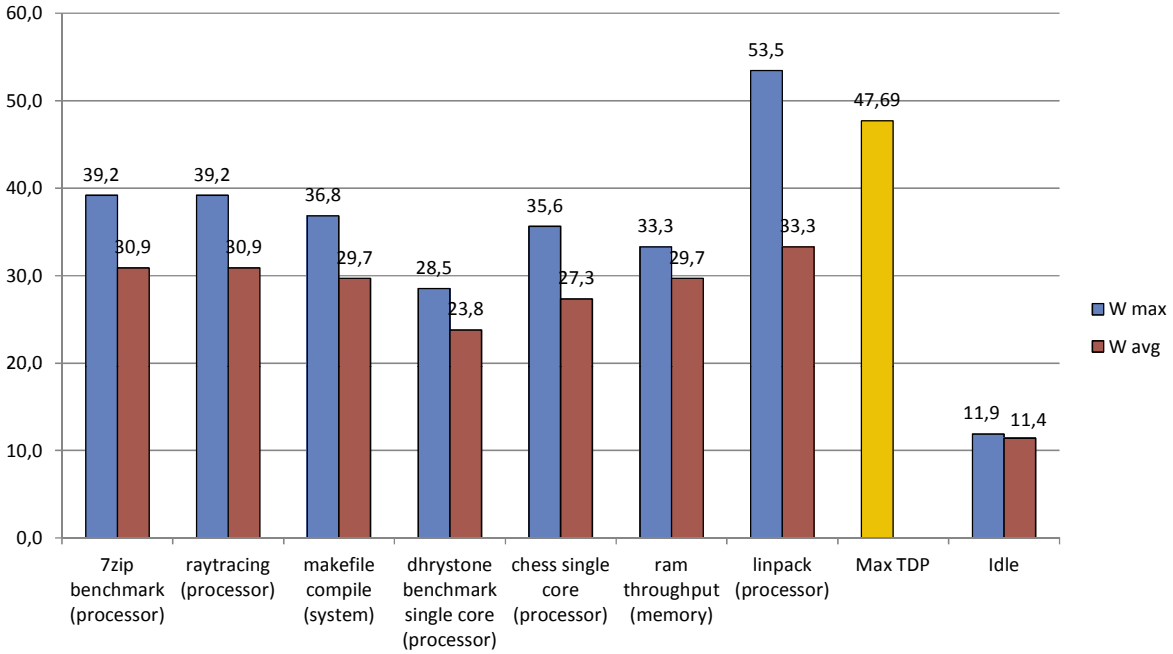

Abbildung 5.10: Die gemessene Leistungsaufnahme eines RECS-Knoten (Energiesparvariante) bei verschiedenen Szenarien und die summierte theoretische Leistungsaufnahme der Komponenten.

Tabelle 5.4: Die Leistungsaufnahme von RECS und der Anteil der Systemkommunikation in der Anwendung als Hochleistungsrechnerverbund.

RECS-Knoten	Leistungsaufnahme Gesamt	Leistungsaufnahme Kommunikation			Anteil Kommunikation
Core2Duo	47,7 W	FSB	PCIe	GbE	15,3 %
		4,36 W	0,98 W	1,96 W	
Core-i7	49,5 W	DMI	PCIe	GbE	9 %
		1,52 W	0,98 W	1,96 W	

Tabelle 5.5: Der Anteil von Intra- und Intersystemkommunikation an der Verlustleistung von HPC-Systemen, dem FPGA-Verbund und RECS.

System	Verlustl. Gesamt	Verlustl. Kommunikation	Anteil Komm.	Leistung (GFlop)	Effizienz (MFlop/W)
Arminius	19500 W	2488 W	12,8 %	7700	395
PLING2	18901 W	2530 W	13,4 %	4700	249
HPC Cloud	11470 W	1310 W	11,4 %	2600	227
WinHPC	2151 W	114 W	5,3 %	224	104
FPGA-Verbund	6487 W	1936 W	29,8 %		
RECS Core2Duo	**870 W**	**133 W**	**15,3 %**	**306**	**352**
RECS Core-i7	**900 W**	**81 W**	**9 %**	**432**	**480**

6 Zusammenfassung und Ausblick

Die rasante Entwicklung mikroelektronischer Schaltkreise stellt Wissenschaft und Forschung gleichermaßen vor neue Herausforderungen. Im Beispielszenario von Rechenzentren manifestieren sich diese Herausforderungen im Umgang mit steigenden Energiekosten, während die wachsende Leistungsdichte bei Standardprozessoren ein Umdenken auf alternative Konzepte erfordert. Hieraus ergeben sich neue Sichtweisen in der Beurteilung der Leistungsfähigkeit von Rechenarchitekturen. Neben der Leistungssteigerung werden diese Architekturen zunehmend auf Energieeffizienz hin optimiert und entwickelt. Erreicht wird dies durch die Optimierung der Hauptkomponenten wie Prozessoren oder auch Netzteile. Zudem werden vermehrt Hardwarebeschleuniger eingesetzt, welche die Energieeffizienz eines Systems erhöhen können. Die kupferbasierte Kommunikation zwischen einzelnen Komponenten wie Prozessoren oder auch kompletten Systemen wurde bisher nicht auf Energieeffizienzmaße hin untersucht. Dabei stellt eine auf Datenaustausch basierende enge Kopplung innerhalb und zwischen Systemen einen wichtigen Teil der Systemaktivität dar. Diese Aktivitäten wurden in dieser Arbeit mit Hilfe von Energieeffizienzmaßen evaluiert. Ein Kanalmodell zur Bestimmung des Einflusses auf das Übertragungssignal wurde vorgestellt und verifiziert. Dieses Kanalmodell wurde mathematisch, anhand des Lumped-Element-Verfahrens hergeleitet. Eine Beschreibung des Kanalverhaltens als S-Parameter-Modell ermöglichte die spätere Integration des Kanals in das gesamte Systemmodell. Aufgrund des vorgestellten Kanalmodells und der sich ergebenden Effekte auf die Signalintegrität wurden Techniken zur Signalformadaption beschrieben, welche später auch in der Evaluation Anwendung fanden. Hierzu gehören Verzerrungen, die das Übertragungssignal vor der Übertragung mit der Inversen der Kanalimpulsantwort überlagern und so die Kanaleinflüsse minimieren. Zudem können nicht kompensierte Einflüsse, aufgrund des Tiefpassverhaltens des Kanals durch frequenzabhängige Verstärker ausgeglichen werden. Der grundsätzliche Aufbau von seriellen Hochgeschwindigkeitstransceiver wurde erklärt und wichtige Komponenten näher erläutert. Auf Basis dieser Erkenntnisse wurden Modelle von FPGA-basierten Transceivern zur Durchführung der Simulationen ausgewählt.
Anschließend wurde eine Evaluierungsmethodik vorgestellt, mit der die Ergebnisse der Effizienzuntersuchungen quantisiert und eingeordnet werden können. Hierzu gehört eine Aussage über die durchschschnittliche Leistungsaufnahme der betrachteten

Verfahren sowie über die benötigte Energie pro übertragenem Bit unter verschiedenen Gesichtspunkten. Hierzu gehört die Betrachtung eines physikalischen Signals auf dem Kanal, eine Angabe über den Einfluss von Kanalkodierungen und Paketformaten auf die benötigte Energie für ein Nutzdatenbit sowie eine Angabe der zu erwartenden Energieeffizienz des Standards in praktischen Anwendungsszenarien. Die drei neben der durchschnittlichen Leistungsaufnahme verwendeten Größen werden auch als „power-delay-product“, also als Produkt aus Leistungsaufnahme und Bitzeit, bezeichnet. Die Übertragungsverfahren wurden auf Basis ihrer technologischen Implementierung eingeordnet und auf ihre Effizienz und durchschnittliche Leitungsaufnahme hin untersucht. Alle Verfahren wurden auf jedem kompatiblen Transceiver in verschiedenen Varianten implementiert. Insgesamt wurden 105 unterschiedliche Varianten von Übertragungsverfahren untersucht. Die Evaluationen wurden mittels *SPICE* und dem *Xilinx Power Analyzer* durchgeführt. Ein Systemmodell wurde dabei zur Übertragung einer pseudozufälligen Bitfolge verwendet und dabei alle zur Bestimmung der Leistungsaufnahme benötigten Parameter aufgezeichnet.
Es stellte sich unabhängig von den verschiedenen Implementierungen heraus, dass eine Erhöhung der Taktfrequenz eine bessere Effizienzsteigerung zur Folge hat als eine Erhöhung der Parallelität. So weist beispielsweise ein Verfahren mit einem einzelnen, bidirektionalen Kanal eine niedrigen Energie pro übertragnenes Bit auf als dasselbe Verfahren mit zwei Kanälen aber nur der halben Taktfrequenz. Die Datenrate ist dabei in beiden Fällen gleich. Dieses Ergebnis spiegelt ein Verhalten wieder, das sich entgegengesetzt zum Verhalten bei Prozessoren darstellt. Prozessoren nutzen Parallelität aus, um weniger Leistungsaufnahme bei gleicher Leistungsfähigkeit zu erreichen. Anschließend fand eine standardübergreifende Evaluation statt. Hier wurden die betrachteten Übertragungsverfahren untereinander verglichen und eine Unterteilung in technologische Unterschiede wie Busse oder Punkt-zu-Punkt-Verbindungen und Intra- beziehungsweise Intersystemverfahren vorgenommen. Für den standardübergreifenden Vergleich wurde der Durchschnitt der Leistungsaufnahme aller Implementierungen eines Verfahrens auf verschiedenen Transceivern gebildet. Der Vergleich aller untersuchten Standards ergab eine globale Steigerung der Energieeffizienz mit zunehmender Datenrate. Eine Betrachtung aller Verfahren zur Intrasystemkommunikation stellte die Überlegenheit von QPI gegenüber anderen Verfahren bezüglich Datenrate und Energieeffizienz heraus. Ältere Standards wie das busbasierte Frontside-Bus-Verfahren können in vielen Bereichen bezüglich der benötigten Energie pro übertragenem Bit mit neueren Verfahren konkurrieren. So weist der Frontside-Bus ab 25 Gbit/s eine geringere Energie pro übertragenem Bit auf als das jeweils gleich schnelle HyperTransport. Bei den Standards zur Intersystemkommunikation dominiert Aurora bei hohen Datenraten die anderen Übertragungsverfahren aufgrund des schlanken Protokollformats. Das weit verbreitete PCIe-Verfahren stellt aufgrund der großen Anzahl möglicher Implementierungsvarianten für verschiedene Übertragungsraten eine optimale Wahl bei der

Inter- und Intrasystemkommunikation dar. Ausgehend von diesen Ergebnissen wurden die gewonnenen Erkenntnisse auf verschiedene Clustersysteme angewendet und der Anteil der Inter- und Intrasystemkommunikation an der Gesamtverlustleistung bestimmt. Es wurden sowohl konventionelle Systeme mit Standardprozessoren als auch hochspezialisierte Systeme wie FPGA-Cluster untersucht. Bei Systemen mit sehr leistungsfähigen Prozessoren mit hohem Energiebedarf aber wenigen hardware-basierten Beschleunigern ist die Kommunikation für 5 % bis 13 % der Gesamtleistungsaufnahme verantwortlich. Werden sparsamere Prozessoren eingesetzt, so erhöht sich der Anteil auf bis zu 15 %. Im Extremfall kann der Kommunikationsanteil bis zu 30 % der Gesamtleistungsaufnahme eines Systems betragen, wenn viele spezialisierte Komponenten wie Hardware-Beschleuniger zum Einsatz kommen. Dies zeigt den Trend hin zu einem immer größer werden Anteil der benötigten Energie für Intra- und Intersystemkommunikation, da zunehmend sparsamere Komponenten eingesetzt werden. Die gewonnenenen Erkenntnisse der vorhergehenden Kapitel wurden genutzt, um zwei energieeffiziente Multiprozessorarchitekturen zu entwickeln. Der SCT-Cluster, ein auf FPGAs basierender dynamisch rekonfigurierbarer Rechencluster, dient zur Simulation großer Schaltungsentwürfe oder neuronaler Netze. RECS, ein ressourceneffizienter Cluster-Server, ist auf niedrigen Energiebedarf sowie auf eine hohe Packungsdichte von physikalischen Rechenknoten optimiert. Mit 18 physikalischen Rechenknoten auf einer Höheneinheit stellt RECS ein sehr effizientes Clustersystem bezüglich Packungsdichte und Energieeffizienz dar. RECS nutzt ein effizientes System zur Überwachung und Steuerung von Multiprozessorarchitekturen. Die Nutzung von dedizierten Kommunikationswegen zur Überwachung und Steuerung des RECS-Systems umgeht die Problematik der größer werdenden Kommunikationslast bei der Clusterüberwachung in Rechenzentren. Die Energieeffizienz von RECS ist so hoch, dass ein Rechencluster auf Basis von RECS im ersten Drittel der weltweit energieeffizientesten 500 Supercomputer plaziert werden kann. LoneStar ist ein energiesparendes Langzeitarchivsystem auf Basis von Festplatten und bietet eine hohe Speicherdichte bei gleichzeitig niedriger Leistungsaufnahme. LoneStar bietet mit 64 Terabyte pro Höheneinheit eine extrem hohe Speicherdichte an. Die Kombination von RECS und LoneStar ermöglicht die Realisierung eines kompletten Rechenzentrums innerhalb eines Serverschranks.

Bei den sich ständig im Bereich der Energieeffizienz verbessernden Rechenarchitekturen werden zunehmend effizientere Komponenten wie Prozessoren und Hardwarebeschleuniger eingesetzt. Jedoch berücksichtigt diese Entwicklung bisher nicht den Kommunikationseinfluss innerhalb und zwischen Architekturen. So nimmt die Leistungsaufnahme der Kommunikation ständig zu, während das Gesamtsystem insgesamt weniger Energie benötigt. Bei den betrachteten Systemen liegt der Kommunikationsanteil bei bis zu 30 % der gesamten Verlustleistung. Es ist anzunehmen, dass sich diese Entwicklung weiter fortsetzen wird und die kupferbasierte Kommunikation zunehmend die Leistungsaufnahme dominiert. Aus diesem Grund ist es für die Wissenschaft und

die Wirtschaft wichtig, das Augenmerk bei Entwicklung zukünftiger Rechnerarchitekturen zunehmend auf die Intra- und Intersystemkommunikation zu legen. Nur so wird auch zukünftig sowohl eine ökologisch als auch ökonomisch sinnvolle Informations- und Kommunikationstechnologie zu realisieren sein.

Verzeichnis verwendeter Formelzeichen, Einheiten und Abkürzungen

Abkürzungen

AC	Wechselstrom
AD	Adressen und Daten
AGTL+	Advanced Gunning Transistor Logic
AMD	American Micro Devices
BE	Byte Enable
Bit	Binärziffer, kleinste Speichereinheit
Byte	Acht Bit
CAD	Kommando-, Adress-, und Datensignal
CLK	Taktleitung
CML	Current Mode Logic
CO2	Kohlenstoffdioxid
CPU	Zentrale Verarbeitungseinheit
CRC	Zyklische Redundanzprüfung
CTL	Kontrollsignal
DAC	Digital zu Analog Wandler
DC	Gleichstrom
DDR	Double Data Rate
DVSEL	Device Select
EDVAC	Electronic Discrete Variable Automatic Computer
FEC	Vorwärtsfehlerkorrektur
Flop	Fließkomma-Operation
FPGA	Feldprogrammierbare Gatteranordnung
FR-4	Epoxidharz-Glasfasermatten
FSB	Front Side Bus
GMII	Gigabit Media Independent Interface
GNT	Global No Transmission
GTL	Gunning Transistor Logic

HE	Höheneinheit
HPC	High Performance Computing
HT	HyperTransport
IFG	Inter Frame Gap
IO	Ein-Ausgabe
IP-Adresse	Internetprotokoll Adresse
IP-Core	Wiederbenutzbarer Teil eines Chipdesigns
IRDY	Interrupt Request Data Ready
ISI	Intersymbolinterferenzen
LAN	Lokales Netzwerk
LVDS	Low Voltage Differential Signaling
LVTTL	Low Voltage Transistor Transistor Logic
MAC	Medienzugriffskontrolle
MGT, GTX, GTP, GTH	Multi-Gigabit Transceiver
MII	Media Independent Interface
MLT	Multi Level Transmission
NIC	Netzwerkschnittstelle
NV	Nachverzerrung
OP	Operation
PAM	Pulsamplituden Modulation
PCB	Leiterplatte
PCI	Peripheral Component Interconnect
PCIe	Peripheral Component Interconnect Express
PCI-X	Peripheral Component Interconnect Extended
PDP	Produkt aus Leistung und Bitzeit
PHY	Physikalische Schnittstelle
PLL	Phasenregelschleife
PMA	Physikalische Anschlusseinheit
PRBS	Pseudozufallszahlenfolge
QDR	Quadruple Data Rate
QPI	Quick Path Interconnect
RAID	Redundant Array of Independent Disks
RECS	Ressourceneffizienter Cluster Server
RGMII	Reduced Gigabit Media Independent Interface
RMII	Reduced Media Independent Interface
RX	Empfang
SDR	Single Data Rate
SerDes	Serialisierung / Deserialisierung
SFD	Start Frame Delimiter
TP	Verdrilltes Leiterpaar
SGMII	Serial Gigabit Media Independent Interface

SI	Signalintegrität
SPICE	Simulation Program with Integrated Circuits Emphasis
TCM	Trelliskodierung
TRDY	Transmit Data Ready
TTL	Transistor Transistor Logic
Twinax	Differentielles Leiterpaar mit gemeinsamer Masseschirmung
TX	Transmit
USB	Universeller, serieller Bus
USD	Amerikanische Dollar
VHDL	Hardwarebeschreibungssprache
VLAN	Virtuelles, lokales Netzwerk
VV	Vorverzerrung
XAUI	10 Gigabit Attachment Unit Interface
XGMII	10 Gigabit Media Independent Interface
Größen	
f	Femto, 10^{-3}
G	Giga, 10^{9}
k	Kilo, 10^{3}
M	Mega, 10^{6}
m	Milli, 10^{-3}
μ	Mikro, 10^{-6}
n	Nano, 10^{-9}
P	Peta, 10^{15}
p	Pico, 10^{-12}
T	Tera, 10^{10}
Einheiten	
A	Ampere
dB	Dezibel
F	Farad
Hz	Hertz
J	Joule
m	Meter
Ω	Ohm
s	Sekunden
V	Volt
W	Watt
Formelzeichen	
δ	Skintiefe
δ_r	Leitfähigkeit des Dielektrikums
ϵ_r	Permittivität des Dielektrikums
γ	Transferkoeeffizient

γ_ω	komplexer, frequenzabhängiger Ausbreitungskoeffizient
μ	magnetische Permeabilität
ω	Frequenz
ω_δ	Frequenz bei der der Skineffekt betrachtet werden muss
ω_{rau}	Frequenz bei der die Oberflächenrauheit betrachtet werden muss
ρ	Reflektionskoeffizient
σ	Konduktivität
$\tilde{Z}_c$	Kettenimpedanz
a	Fläche eines Leiters
c	Lichtgeschwindigkeit
C	Kapazität
C_ω	Komplexe, charakteristische Kapazität
C_0	Charakteristische Kapazität
E	Energie
G	Leitwert
$H(\omega, l)$	Übertragungsfunktion für bestimmte Kanallänge
$H(\omega)$	Übertragungsfunktion
h_{RMS}	quadratischer Mittelwert der Oberflächenunebenheiten
i	Momentaner Strom
j	Imaginärer Faktor
k_a	Korrekturfaktor für Einzelwiderstände von Hin- und Rückleiter
k_p	Korrekturfaktor für Kopplung zwischen mehreren Leitern
k_r	Korrekturfaktor für Oberflächenrauheit
L	Induktivität
L_0	Charakteristische Induktivität
n	Index
P	Leistung
p	Umfang eines Leiters
PDP	Energie pro Bit
PDP_{code}	Energie pro Bit mit Kodierungseinfluss
PDP_{real}	Energie pro Bit in praktischer Anwendung
R	Widerstand
t	Zeit
u	Momentane Spannung
v	Ausbreitungsgeschwindigkeit
y	Admittanz
z'	Eingangsimpedanz mit paralleler Admittanz
z	Impedanz
Z_0	Charakteristische Impedanz bei einer bestimmten Frequenz
Z_c	Charakteristische Impedanz
Z_i	Eingangsimpedanz

Literaturverzeichnis

Betreute Arbeiten

[B1] Patrick Jebramcik. *Entwurf eines energieeffizienten, festplattenbasierten Langzeitspeichersystems, Bachelorarbeit.* Dec. 2011.

[B2] Michael Karzellek. *Ressourceneffiziente Überwachung und Steuerung von Multiprozessorarchitekturen., Bachelorarbeit.* Apr. 2012.

[B3] Lennart Tigges. *Einbettung von Hardwarebeschleunigern in ein ressourceneffizientes Clustersystem, Bachelorarbeit.* Apr. 2012.

Schutzrechte, Erfindungen

[P1] Manuel Strugholtz, Mario Porrmann, Jens Hagemeyer, and Christmann Informationstechnik + Medien GmbH & Co. KG. *Mehrprozessor-Computersystem.* Schutzrecht. laufendes Patentverfahren. Christmann Informationstechnik + Medien GmbH & Co. KG.

[P2] Manuel Strugholtz, Mario Porrmann, Jens Hagemeyer, and Christmann Informationstechnik + Medien GmbH & Co. KG. *Mehrprozessor-Computersystem.* Erfindungsmeldung. Erfindungsmeldung Universität Paderborn. Universität Paderborn, PROvendis.

Eigene Veröffentlichungen

[E1] Manuel Strugholtz, Mario Porrmann, Jens Hagemeyer, Christopher Pohl, and Johannes Romoth. "RAPTOR - A Scalable Platform for Rapid Prototyping and FPGA-based Cluster Computing". In: *Proceedings of the International Conference on Parallel Computing, ParCo2009, Symposium on Parallel Computing with FPGAs.* Lyon, France, Sept. 2009.

[E2] Manuel Strugholtz, Mario Porrmann, Jens Hagemeyer, Johannes Romoth, and Mohammed Abdel-Wahab. “Rapid Prototyping of Next-Generation Multiprocessor SoCs”. In: *Proceedings of Semiconductor Conference Dresden, SCD 2009*. Dresden, Germany, July 2009.

[E3] Manuel Strugholtz, PSalim Ullah, Mario Porrmann, Jens Hagemeyer, K. M. Yahya, and Asif Manzoor. “An Efficient Communication Architecture for FPGA Based Many Core System”. In: *The 2010 International Congress on Computer Applications and Computational Science*. Singapore, 2010.

Referenzen

[R1] Polar Instruments SI9000. URL: `http://www.polarinstruments.com`.

[R2] Mentor Graphics HyperLynx. URL: `http://www.mentor.com`.

[R3] Maplesoft Maple. URL: `http://www.maplesoft.com`.

[R4] Xilinx Inc. Power Analyzer. URL: `http://www.xilinx.com`.

[R5] Xilinx Inc. URL: `http://www.xilinx.com`.

[R6] Linpack Benchmark Library. URL: `http://www.netlib.org/linpack/`.

[R7] Anant Agarwal and Markus Levy. “The KILL Rule for Multicore”. In: *DAC*. IEEE, 2007, pp. 750–753. URL: `http://doi.ieeecomputersociety.org/10.1109/DAC.2007.375264`.

[R8] *An Introduction to the Intel QuickPath Interconnect*. Tech. rep. Intel Corporation, 2009.

[R9] Gerhard Angst. *Visualization of SPICE files helps to optimize SoCs*. Tech. rep. Concept Engineering, 2004.

[R10] *Aurora 8B/10B Protocol Specification*. Tech. rep. Xilinx Inc., 2010.

[R11] Fachhochschule Basel. Netzwerkmessungen - Vergleiche DS2 PLC, Homeplug PLC und 10/100 MBit Ethernet. June 2004.

[R12] Troy R. Benjegerdes and Brett M. Bode. “InfiniBand performance review: It’s the software stupid”. In: *Proceedings of the annual conference on USENIX Annual Technical Conference*. ATEC ’04. USENIX Association, 2004, pp. 42–42.

[R13] Eric Bogatin. *Signal and power integrity - simplified*. eng. Upper Saddle River, NJ [u.a.]: Prentice Hall, 2010, XXVI, 757 S. : Ill., graph. Darst. ISBN: 978-0-13-234979-6, 0-13-234979-5.

[R14] Fachhochschule Bonn-Rhein-Sieg. Performance-Untersuchungen an einem Gigabit-Ethernet-Netzwerk. Aug. 2004.

[R15] Ravi Budruk, Don Anderson, and Tom Shanley. *PCI express system architecture*. eng. Boston [u.a.]: Addison-Wesley, 2010, LIV, 1049 S. : graph. Darst. ISBN: 0-321-15630-7, 978-0-321-15630-3.

[R16] Claire Castellanos. *PCI-SIG ANNOUNCES PCI EXPRESS 4.0 EVOLUTION TO 16GT/S, TWICE THE THROUGHPUT OF PCI EXPRESS 3.0 TECHNOLOGY*. Tech. rep. PCI SIG, 2011.

[R17] Ganesh Balakrishnan Christian Belady. *Incenting the Right Behaviors in the Data Center*. Tech. rep. Microsoft Corporation, 2008.

[R18] Yi-Chin Chu. *Serial-GMII Specification*. Tech. rep. Cisco Systems, 2005.

[R19] *ConnectX - Single/Dual-Port InfiniBand Adapter Cards*. Tech. rep. Mellanox, 2011.

[R20] Intel Corporation. FSB Presentation of Intel higher education class. Apr. 2002.

[R21] National Semiconductor Dave Lewis. DesignCon 2004 SerDes Architectures and Applications.

[R22] Michael Mirmak David Coleman. *The Essentials of the Intel Quickpath Interconnect Electrical Architecture*. Tech. rep. Intel Corporation, 2009.

[R23] Kirk W. Cameron Wu-chun Feng. *The Green500 List*. Tech. rep. July 2011.

[R24] Wu chun Feng and Thomas Scogland. "The Green500 List: Year one". In: *IPDPS*. IEEE, 2009, pp. 1–7. URL: http://dx.doi.org/10.1109/IPDPS.2009.5160978.

[R25] Dave Fifield. *GMII Electrical Specification*. Tech. rep. National Semiconductor, 1997.

[R26] Flanagan, David and Shafer, Dan. *JavaScript: The Definitive Guide*. O'Reilly & Associates, 1998, p. 776. ISBN: 1-56592-392-8.

[R27] Howard Frazier. *IEEE P802.3ae 10 Gigabit Ethernet Task Force - XGMII Update*. Tech. rep. Cisco Systems, 2000.

[R28] Ethan Galstad. *Nagios Core Documentation*. Tech. rep. Nagios Core Development Team and Community Contributors, 2009.

[R29] Ilango Ganga. *Considerations for 40G Backplane Ethernet PHY*. Tech. rep. Intel Corporation, 2007.

[R30] Tom Granberg. *Handbook of digital techniques for high-speed design*. eng. Upper Saddle River, NJ: Prentice Hall PTR, 2004, XLIV, 928 S. : Ill., graph. ISBN: 0-13-142291-X.

[R31] Tsuyoshi Hamada and Naohito Nakasato. "InfiniBand Trade Association, InfiniBand Architecture Specification, Volume 1, Release 1.0". In: *in International Conference on Field Programmable Logic and Applications*. 2005, pp. 366–373.

[R32] Arne Hilgenstein. *Multi-Gigabit-Switch für Inter-FPGA-Kommunikation*. Diplomarbeit, FG-Schaltungstechnik, Universität Paderborn, Prof. Dr.-Ing. Ulrich Rückert.

[R33] *HyperTransport I/O Link Specification*. Tech. rep. HyperTransport Technology Consortium, 2010.

[R34] *IEEE 802.3 Standard*. Tech. rep. Institute of Electrical and Electronics Engineers, IEEE, 2001.

[R35] *Intel 82599 10 Gigabit Ethernet Controller*. Tech. rep. Intel Corporation, 2011.

[R36] *Intel Pentium 4 Processor with 512-KB L2 Cache on 0.13 Micron Process and Intel Pentium 4 Processor Extreme Edition Supporting Hyper-Threading Technology1*. Tech. rep. Intel Corporation, 2004.

[R37] *Interface Circuits for TIA/EIA-644 (LVDS*. Tech. rep. Texas Instruments, 2002.

[R38] *Introduction to LVDS, PECL, and CML*. Tech. rep. Maxim Integrated Products, 2008.

[R39] Pebly Chhabra Jainandunsing Huber. *COM-Express The Next Big Trend in Embedded Computing Small Form Factors*. Tech. rep. PFU, Kontron, RadiSys Corporation, Intel Corporation, 2004.

[R40] Thomas Dippon Jennie Grosslight. *InfiniBand Compliance and System Testing, Switch Performance Measurements*. Tech. rep. Agilent Technologies, 2002.

[R41] Mark Nowell John Ambrosia David Law. *40 Gigabit Ethernet and 100 Gigabit Ethernet Technology Overview*. Tech. rep. Ethernet Alliance, 2010.

[R42] Liu J et al. Kipp A Schubert L. "Energy Consumption Optimisation in HPC Service Centres". In: *Second International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering, Ajaccio, Corsica, France*. 2011.

[R43] Alexander Kipp, Jia Liu, Tao Jiang, Dmitry Khabi, Yevgeniya Kovalenko, Lutz Schubert, Micha Vor Dem Berge, and Wolfgang Christmann. "Approach towards an Energy-Aware and Energy-
Efficient High Performance Computing Environment". In: *Proceedings of the 7th IEEE International Conference on Intelligent Computer Communication and Processing*. 2011, p. 8.

[R44] K. Kurokawa. "Power Waves and the Scattering Matrix". In: *Microwave Theory and Techniques, IEEE Transactions on* 13.2 (Mar. 1965), pp. 194 –202. ISSN: 0018-9480. DOI: 10.1109/TMTT.1965.1125964.

[R45] Fujitsu Limited. *Supercomputer „K computer“ Takes First Place in World*. Tech. rep. RIKEN Advanced Institute for Computational Science, 2011.

[R46] *LogiCORE IP XAUI v9.2*. Tech. rep. Xilinx Inc., 2010.

[R47] Dejan Minic. *MSC8122/26 Ethernet MII Quick Start*. Tech. rep. Freescale Semiconductor, 2006.

[R48] Gordon E. Moore. “Cramming more components onto integrated circuits”. In: *Electronics* 38.8 (1965), pp. 114–117. URL: http://www.intel.com/research/silicon/moorespaper.pdf.

[R49] Laurence W. Nagel and D.O. Pederson. *SPICE (Simulation Program with Integrated Circuit Emphasis)*. Tech. rep. UCB/ERL M382. EECS Department, University of California, Berkeley, Apr. 1973. URL: http://www.eecs.berkeley.edu/Pubs/TechRpts/1973/22871.html.

[R50] John von Neumann. *First Draft of a Report on the EDVAC*. Tech. rep. Moore School of Electrical Engineering, University of Pennsylvania, 1945.

[R51] Mondrian Nüssle. *Leveraging HyperTransport for a custom high-performance cluster network*. Tech. rep. Universität Heidelberg, 2009.

[R52] Tim Nufire. *Petabytes On a Budget - How to build cheap Cloud Storage*. Tech. rep. Backblaze, 2009.

[R53] *PCI Express Performance Measurements*. Tech. rep. Texas Instruments, 2006.

[R54] Paderborn Center of Parallel Computing. Available Systems and Software. URL: http://www.pc2.uni-paderborn.de/.

[R55] Fred J. Pollack. “New Microarchitecture Challenges in the Coming Generations of CMOS Process Technologies”. In: *MICRO*. 1999, p. 2. URL: http://computer.org/proceedings/micro/0437/04370002abs.htm.

[R56] Universität Mannheim Prof. Dr. U. Brüning. Vorlesung Rechnerarchitektur. 2011.

[R57] *QLogic 7300 Series Infiniband HCAs*. Tech. rep. QLogic, 2011.

[R58] *RMII Specification*. Tech. rep. RMII Consortium, 1998.

[R59] *Reduced Media Independent Interface (RGMII)*. Tech. rep. Broadcom, HP, Marvell, 2002.

[R60] Samtec. High Speed Board to Board solutions. URL: http://www.samtec.com.

[R61] Tom Shanley and Don Anderson. *PCI system architecture (3. ed.)* PC system architecture series. Addison-Wesley, 1995, pp. I–XXXI, 1–557. ISBN: 978-0-201-40993-2.

[R62] *Spartan-6 FPGA GTP Transceivers User Guide*. Tech. rep. Xilinx Inc., 2010.

[R1] Manuel Strugholtz, Mario Porrmann, Jens Hagemeyer, and Christmann Informationstechnik + Medien GmbH & Co. KG. *Mehrprozessor-Computersystem.* Schutzrecht. laufendes Patentverfahren. Christmann Informationstechnik + Medien GmbH & Co. KG.

[R2] Manuel Strugholtz, Mario Porrmann, Jens Hagemeyer, and Christmann Informationstechnik + Medien GmbH & Co. KG. *Mehrprozessor-Computersystem.* Erfindungsmeldung. Erfindungsmeldung Universität Paderborn. Universität Paderborn, PROvendis.

[R63] Jürgen Suppan-Borowka. *Ethernet-Handbuch.* ger. Pulheim: DATACOM-Buchverlag, 1987, 321 S. : Ill., graph. Darst. ISBN: 3-89238-010-4.

[R64] *TLK3114SC 10-Gbps XAUI Transceiver.* Tech. rep. Texas Instruments, 2006.

[R65] *The TTL data book for design engineers.* eng. [Dallas, Tex.]: Texas Instruments.

[R66] Rob Kowalczyk Todd Langley. *Introduction to Intel Architecture, The Basics.* Tech. rep. Intel Corporation, 2009.

[R67] *VC1053, VC1052 - Multi-Protocol, Multi-Gigabit QUAD SerDes.* Tech. rep. LSI Logic, 2011.

[R68] *Virtex-4 RocketIO Multi-Gigabit Transceiver User Guide.* Tech. rep. Xilinx Inc., 2006.

[R69] *Virtex-5 FPGA RocketIO GTP Transceiver User Guide.* Tech. rep. Xilinx Inc., 2009.

[R70] *Virtex-5 FPGA RocketIO GTX Transceiver User Guide.* Tech. rep. Xilinx Inc., 2009.

[R71] *Virtex-6 FPGA GTH Transceivers User Guide.* Tech. rep. Xilinx Inc., 2011.

[R72] *Virtex-6 FPGA GTX Transceivers User Guide.* Tech. rep. Xilinx Inc., 2011.

[R73] *X-fest 2007 Course Presentations.* Tech. rep. Xilinx Inc., 2007.

Anhang

In diesem Anhang finden sich alle Ergebnisse der Simulationen und die verwendeten Parameter.

XAUI

Parameter	Wert
Zeitschritt	5 ps
Temperatur	$25°C$
Spannungen	Typische Werte laut jeweiligem FPGA-Datenblatt.
Flankensteilheit	$^{0,125}/_{Datenrate}$
Spannungshub	„110 “
Vorverzerrung	„010 “
Entzerrung	„11 “
Terminierung	„VTTX “
Kopplung	„AC “

Tabelle 6.1: Verwendete Parameter in der XAUI-Simulation.

Transceiver	PDP	PDP_{code}	PDP_{real}
V4 MGT	9,69501E-11	1,24581E-10	1,40776E-10
V5 GTP	3,84264E-11	4,93779E-11	5,5797E-11
V5 GTX	4,92046E-11	6,32279E-11	7,14475E-11
S6 GTP	7,48799E-11	9,62207E-11	1,08729E-10
V6 GTX	6,77798E-11	8,70971E-11	9,84197E-11
V6 GTH	7,91416E-11	1,01697E-10	1,14918E-10

Tabelle 6.2: Ergebnisse der Effizienzbetrachtung bei XAUI in Joule.

S6 GTP					
VCCTX	VCCRX	VTTX	VTTR	PLL	VCCINT
0,0367208	0,02305	0,058998	0,021431	0,07	0,0588
0,9359992					
V5 GTP					
VCCTX	VCCRX	VTTX	VTTR	PLL	VCCINT
0,0183604	0,011525	0,019666	0,021431	0,0432	0,0275
0,4803296					
V5 GTX					
VTTX	VTTR	VCCX	VCCR	PLL	VCCINT
0,138	0,084	0,165	0,000017	0,14992	0,07812
0,615057					
V4 MGT					
VCCX	VCCR	VTTR	VCCAUX	VTTX	VCCINT
0,08654975	0,1730995	0,0070393	0,008	0,1141	0,0653
1,2118766					
V6 GTX					
VTTX	VTTR	VCCX	VCCR	VCCINT	
0,048731	0,03945	0,092376	0,000005	0,03125	
0,847248					
V6 GTH					
VTTX	VCCR	VCCX	PLL	VCCINT	
0,036974	0,044424	0,087867	0,19221	0,03	
0,98927					

Tabelle 6.3: Leistungsaufnahme der Versorgungsleitungen bei XAUI in Watt.

SGMII

Parameter	Wert
Zeitschritt	5 ps
Temperatur	$25°C$
Spannungen	Typische Werte laut jeweiligem FPGA-Datenblatt.
Flankensteilheit	100 ps
Spannungshub	„111 “
Vorverzerrung	„000 “
Entzerrung	„01 “
Terminierung	„VTTX “
Kopplung	„DC “

Tabelle 6.4: Verwendete Parameter in der SGMII-Simulation.

Transceiver	PDP	PDP_{code}	PDP_{real}
V4 MGT	5,38179E-10	6,91559E-10	7,81462E-10
V5 GTP	1,89546E-10	2,43567E-10	2,7523E-10
V5 GTX	3,76998E-10	4,84442E-10	5,47419E-10
S6 GTP	3,73425E-10	4,79851E-10	5,42231E-10
V6 GTX	2,17154E-10	2,79043E-10	3,15318E-10
V6 GTH	4,2032E-10	5,40112E-10	6,10326E-10

Tabelle 6.5: Ergebnisse der Effizienzbetrachtung bei SGMII in Joule.

S6 GTP					
VCCTX	VCCRX	VTTX	VTTR	PLL	VCCINT
0,0360876	0,0246128	0,059121	0,021369	0,06885	0,02335
0,4667808					
V5 GTP					
VCCTX	VCCRX	VTTX	VTTR	PLL	VCCINT
0,0180438	0,0123064	0,019707	0,021369	0,03604	0,011
0,2369324					
V5 GTX					
VTTX	VTTR	VCCX	VCCR	PLL	VCCINT
0,07	0,042	0,056	0,0000085	0,0598	0,007815
0,471247					
V4 MGT					
VCCX	VCCR	VTTR	VCCAUX	VTTX	VCCINT
0,12388	0,24776	0,0065032	0,016	0,22658	0,052
0,6727232					
V6 GTX					
VTTX	VTTR	VCCX	VCCR	VCCINT	
0,09861	0,078622	0,0692004	0,00001	0,025	
0,2714424					
V6 GTH					
VTTX	VCCR	VCCX	PLL	VCCINT	
0,07623	0,07931	0,0000004	0,343	0,02686	
0,5254004					

Tabelle 6.6: Leistungsaufnahme der Versorgungsleitungen bei SGMII in Watt.

Parameter	Wert
Zeitschritt	1 ps
Temperatur	$25°C$
Spannungen	Typische Werte laut jeweiligem FPGA-Datenblatt.
Flankensteilheit	10 ps
Spannungshub	„1111 “
Vorverzerrung	„0000 “
Entzerrung	„000 “
Terminierung	„VTTX “
Kopplung	„AC “

Tabelle 6.7: Verwendete Parameter in der PCIe-Simulation.

S6 GTP							
Breite	VCCTX	VCCRX	VTTX	VTTR	PLL	VCCINT	Gesamt
1x	0,0367	0,0212	0,1188	0,0023	0,0688	0,0471	**0,5902**
2x	0,0734	0,0424	0,2377	0,0047	0,0688	0,0942	**0,5214**
4x	0,1468	0,0848	0,4754	0,0095	0,1377	0,1884	**1,3379**
8x	0,2936	0,1697	0,9508	0,0191	0,2754	0,3768	**2,3807**
16x	0,5873	0,3395	1,9016	0,0383	0,5508	0,7536	**4,4664**
V5 GTP							
Breite	VCCTX	VCCRX	VTTX	VTTR	PLL	VCCINT	Gesamt
1x	0,018	0,011	0,040	0,002	0,036	0,022	**0,245**
2x	0,037	0,021	0,079	0,005	0,036	0,044	**0,337**
4x	0,073	0,042	0,158	0,010	0,072	0,088	**0,560**
8x	0,147	0,085	0,317	0,019	0,144	0,176	**1,004**
16x	0,294	0,170	0,634	0,038	0,288	0,352	**1,892**
V5 GTX							
Breite	VTTX	VTTR	VCCX	PLL	VCCINT		Gesamt
1x	0,031	0,016	0,041	0,060	0,016		**0,327**
2x	0,062	0,032	0,082	0,060	0,031		**0,430**
4x	0,124	0,064	0,164	0,120	0,063		**0,698**
8x	0,248	0,128	0,328	0,239	0,125		**1,232**
16x	0,496	0,256	0,656	0,478	0,250		**2,300**
V4 MGT							
Breite	VCCX	VCCR	VTTR	VCCAUX	VTTX	VCCINT	Gesamt
1x	0,075	0,150	0,041	0,008	0,109	0,052	**0,725**
2x	0,150	0,150	0,041	0,008	0,109	0,104	**0,852**
4x	0,300	0,300	0,082	0,016	0,217	0,208	**1,413**
8x	0,600	0,600	0,165	0,032	0,435	0,416	**2,537**
16x	1,200	1,200	0,330	0,064	0,869	0,832	**4,784**
V6 GTX							
Breite	VTTX	VTTR	VCCX	VCCINT			Gesamt
1x	0,046	0,064	0,072	0,025			**0,371**
2x	0,091	0,127	0,143	0,050			**0,577**
4x	0,183	0,254	0,286	0,100			**0,989**
8x	0,366	0,509	0,573	0,200			**1,813**
16x	0,732	1,018	1,146	0,400			**3,461**

Tabelle 6.8: Leistungsaufnahme der Versorgungsleitungen bei PCI-Express 1.0 in Watt (Teil 1).

V6 GTH							
Breite	VTTX	VCCR	VCCX	PLL	VCCINT		Gesamt
1x	0,040	0,043	0,083	0,188	0,027		**0,711**
2x	0,080	0,086	0,167	0,188	0,054		**0,904**
4x	0,160	0,171	0,334	0,188	0,107		**1,290**
8x	0,321	0,343	0,667	0,376	0,215		**2,251**
16x	0,642	0,686	1,334	0,752	0,430		**4,173**

Tabelle 6.9: Leistungsaufnahme der Versorgungsleitungen bei PCI-Express 1.0 in Watt (Teil 2).

V5 GTX							
Breite	VTTX	VTTR	VCCX	PLL	VCCINT		Gesamt
1x	0,031	0,016	0,043	0,0598	0,031		**0,361**
2x	0,062	0,032	0,086	0,0598	0,063		**0,482**
4x	0,124	0,064	0,172	0,1196	0,130		**0,784**
8x	0,248	0,128	0,344	0,2392	0,250		**1,389**
16x	0,496	0,256	0,688	0,4784	0,500		**2,598**
V4 MGT							
Breite	VCCX	VCCR	VTTR	VCCAUX	VTTX	VCCINT	Gesamt
1x	0,074	0,148	0,041	0,008	0,108	0,104	**0,775**
2x	0,148	0,148	0,041	0,008	0,108	0,208	**0,953**
4x	0,297	0,297	0,082	0,016	0,217	0,416	**1,617**
8x	0,595	0,595	0,164	0,032	0,435	0,832	**2,945**
16x	1,191	1,191	0,328	0,064	0,871	1,664	**5,601**
V6 GTX							
Breite	VTTX	VTTR	VCCX	VCCINT			Gesamt
1x	0,0458	0,063	0,123	0,05			**0,448**
2x	0,0917	0,127	0,246	0,1			**0,730**
4x	0,1834	0,254	0,492	0,2			**1,296**
8x	0,3669	0,508	0,985	0,4			**2,427**
16x	0,7339	1,017	1,971	0,8			**4,689**
V6 GTH							
Breite	VTTX	VCCR	VCCX	PLL	VCCINT		Gesamt
1x	0,0402	0,0489	0,101	0,221	0,053		**0,855**
2x	0,0805	0,0978	0,202	0,221	0,107		**1,099**
4x	0,1610	0,1957	0,405	0,221	0,214		**1,588**
8x	0,3221	0,3915	0,810	0,443	0,429		**2,786**
16x	0,6442	0,7831	1,620	0,886	0,859		**5,183**

Tabelle 6.10: Leistungsaufnahme der Versorgungsleitungen bei PCI-Express 2.0 in Watt.

V6 GTH							
Breite	VTTX	VCCR	VCCX	PLL	VCCINT		Gesamt
1x	0,040356	0,056107	0,1255553	0,18801	0,02685		**0,7662353**
2x	0,080712	0,112214	0,2511106	0,18801	0,0537		**1,0151036**
4x	0,161424	0,224428	0,5022212	0,18801	0,1074		**1,5128402**
8x	0,322848	0,448856	1,0044424	0,37602	0,2148		**2,6963234**
16x	0,645696	0,897712	2,0088848	0,75204	0,4296		**5,0632898**

Tabelle 6.11: Leistungsaufnahme der Versorgungsleitungen bei PCI-Express 3.0 in Watt.

Transceiver	PDP	PDP_{code}	PDP_{real}
v4 mgt 1x	2,899E-10	3,65206E-10	4,56507E-10
v4 mgt 2x	1,70345E-10	2,14595E-10	2,68243E-10
v4 mgt 4x	1,41348E-10	1,78065E-10	2,22582E-10
v4 mgt 8x	1,26849E-10	1,59801E-10	1,99751E-10
v4 mgt 16x	1,196E-10	1,50668E-10	1,88335E-10
v4 mgt 32x	1,15976E-10	1,46102E-10	1,82628E-10
v5 gtp 1x	9,78075E-11	1,23215E-10	1,54018E-10
v5 gtp 2x	6,74999E-11	8,50341E-11	1,06293E-10
v5 gtp 4x	5,59501E-11	7,0484E-11	8,8105E-11
v5 gtp 8x	5,01752E-11	6,3209E-11	7,90113E-11
v5 gtp 16x	4,72878E-11	5,95715E-11	7,44644E-11
v5 gtp 32x	4,5844E-11	5,77528E-11	7,21909E-11
v5 gtx 1x	1,30743E-10	1,64706E-10	2,05883E-10
v5 gtx 2x	8,60976E-11	1,08463E-10	1,35578E-10
v5 gtx 4x	6,97547E-11	8,78745E-11	1,09843E-10
v5 gtx 8x	6,15832E-11	7,75804E-11	9,69755E-11
v5 gtx 16x	5,74975E-11	7,24333E-11	9,05416E-11
v5 gtx 32x	5,54546E-11	6,98598E-11	8,73247E-11
s6 gtp 1x	2,36106E-10	2,97439E-10	3,71798E-10
s6 gtp 2x	1,6331E-10	2,05732E-10	2,57165E-10
s6 gtp 4x	1,33797E-10	1,68552E-10	2,1069E-10
s6 gtp 8x	1,1904E-10	1,49962E-10	1,87453E-10
s6 gtp 16x	1,11662E-10	1,40667E-10	1,75834E-10
s6 gtp 32x	1,07972E-10	1,3602E-10	1,70025E-10
v6 gtx 1x	1,48579E-10	1,87175E-10	2,33968E-10
v6 gtx 2x	1,15485E-10	1,45484E-10	1,81855E-10
v6 gtx 4x	9,89376E-11	1,24638E-10	1,55798E-10
v6 gtx 8x	9,0664E-11	1,14215E-10	1,42769E-10
v6 gtx 16x	8,65273E-11	1,09004E-10	1,36255E-10
v6 gtx 32x	8,44589E-11	1,06398E-10	1,32998E-10
v6 gth 1x	2,84235E-10	3,58069E-10	4,47586E-10
v6 gth 2x	1,80761E-10	2,27717E-10	2,84646E-10
v6 gth 4x	1,29025E-10	1,62541E-10	2,03176E-10
v6 gth 8x	1,12557E-10	1,41795E-10	1,77244E-10
v6 gth 16x	1,04323E-10	1,31422E-10	1,64278E-10
v6 gth 32x	1,00206E-10	1,26236E-10	1,57795E-10

Tabelle 6.12: Ergebnisse der Effizienzbetrachtung bei PCI-Express 1.0 in Joule.

Transceiver	PDP	PDP_{code}	PDP_{real}
v4 mgt 1x	1,55078E-10	1,95362E-10	2,44202E-10
v4 mgt 2x	9,53867E-11	1,20165E-10	1,50206E-10
v4 mgt 4x	8,08883E-11	1,019E-10	1,27375E-10
v4 mgt 8x	7,3639E-11	9,27679E-11	1,1596E-10
v4 mgt 16x	7,00144E-11	8,82018E-11	1,10252E-10
v4 mgt 32x	6,82021E-11	8,59187E-11	1,07398E-10
v5 gtx 1x	7,22217E-11	9,09824E-11	1,13728E-10
v5 gtx 2x	4,82363E-11	6,07664E-11	7,5958E-11
v5 gtx 4x	3,92336E-11	4,94251E-11	6,17814E-11
v5 gtx 8x	3,47322E-11	4,37545E-11	5,46931E-11
v5 gtx 16x	3,24815E-11	4,09191E-11	5,11489E-11
v5 gtx 32x	3,13562E-11	3,95015E-11	4,93768E-11
v6 gtx 1x	8,96392E-11	1,12924E-10	1,41156E-10
v6 gtx 2x	7,30921E-11	9,20789E-11	1,15099E-10
v6 gtx 4x	6,48186E-11	8,16562E-11	1,0207E-10
v6 gtx 8x	6,06818E-11	7,64448E-11	9,5556E-11
v6 gtx 16x	5,86134E-11	7,38391E-11	9,22989E-11
v6 gtx 32x	5,75792E-11	7,25363E-11	9,06704E-11
v6 gth 1x	1,71111E-10	2,15559E-10	2,69449E-10
v6 gth 2x	1,09975E-10	1,38543E-10	1,73178E-10
v6 gth 4x	7,9407E-11	1,00034E-10	1,25043E-10
v6 gth 8x	6,96626E-11	8,77586E-11	1,09698E-10
v6 gth 16x	6,47904E-11	8,16207E-11	1,02026E-10
v6 gth 32x	6,23543E-11	7,85518E-11	9,81897E-11

Tabelle 6.13: Ergebnisse der Effizienzbetrachtung bei PCI-Express 2.0 in Joule.

Transceiver	PDP	PDP_{code}	PDP_{real}
v6 gth 1x	9,57794E-11	9,80359E-11	1,22545E-10
v6 gth 2x	6,3444E-11	6,49387E-11	8,11734E-11
v6 gth 4x	4,72763E-11	4,83901E-11	6,04876E-11
v6 gth 8x	4,21301E-11	4,31226E-11	5,39033E-11
v6 gth 16x	3,9557E-11	4,04889E-11	5,06111E-11
v6 gth 32x	3,82704E-11	3,9172E-11	4,8965E-11

Tabelle 6.14: Ergebnisse der Effizienzbetrachtung bei PCI-Express 3.0 in Joule.

QPI

Parameter	Wert
Zeitschritt	5 ps
Temperatur	$25°C$
Spannungen	Typische Werte laut jeweiligem FPGA-Datenblatt.
Flankensteilheit	$^{0,125}/_{Datenrate}$
Spannungshub	„110 “
Vorverzerrung	„001 “
Entzerrung	„01 “
Terminierung	„VTTX “
Kopplung	„DC “

Tabelle 6.15: Verwendete Parameter in der QPI-Simulation.

V5 GTX QPI 4,8					
VTTX	VTTR	VCCX	VCCR	PLL	VCCINT
0,714	0,451	0,883	0,00009	0,6578	0,65625
3,36214					
V5 GTX QPI 6,4					
VTTX	VTTR	VCCX	VCCR	PLL	VCCINT
0,735	0,41	0,924	0,0000903	0,82456	0,820365
3,7140153					
V4 MGT QPI 4,8					
VCCX	VCCR	VTTR	VCCAUX	VTTX	VCCINT
0,814647	1,629294	0,0335855	0,08	1,25972	2,184
6,0012465					
V4 MGT QPI 6,4					
VCCX	VCCR	VTTR	VCCAUX	VTTX	VCCINT
1,083278	2,166556	0,0330923	0,088	1,26199	2,7426
7,3755163					
V6 GTX QPI 4,8					
VTTX	VTTR	VCCX	VCCR	VCCINT	
1,027142	0,826906	2,3884028	0,000105	1,05	
5,2925558					
V6 GTX QPI 6,4					
VTTX	VTTR	VCCX	VCCR	VCCINT	
1,028869	0,826385	3,0485007	0,000105	1,31313	
6,2169897					
V6 GTH QPI 4,8					
VTTX	VCCR	VCCX	PLL	VCCINT	
0,775698	1,018395	2,1309834	1,32948	1,1277	
6,3822564					
V6 GTH QPI 6,4					
VTTX	VCCR	VCCX	PLL	VCCINT	
0,775782	1,099056	2,3640708	1,3767	1,26084	
6,8764488					

Tabelle 6.16: Leistungsaufnahme der Versorgungsleitungen bei QPI in Watt.

Transceiver	PDP	PDP_{code}	PDP_{real}
QPI 4,8			
V4 MGT	6,2513E-11	7,81412E-11	8,67368E-11
V5 GTX	3,50223E-11	4,37779E-11	4,85934E-11
V6 GTX	5,51308E-11	6,89135E-11	7,6494E-11
V6 GTH	6,64818E-11	8,31023E-11	9,22435E-11
QPI 6,4			
v4 mgt	5,76212E-11	7,20265E-11	7,99494E-11
v5 gtx	2,90157E-11	3,62697E-11	4,02593E-11
v6 gtx	4,85702E-11	6,07128E-11	6,73912E-11
v6 gth	5,37223E-11	6,71528E-11	7,45396E-11

Tabelle 6.17: Ergebnisse der Effizienzbetrachtung bei QPI in Joule.

HyperTransport

Parameter	Wert
Zeitschritt	5 ps
Temperatur	$25°C$
Spannungen	Typische Werte laut jeweiligem FPGA-Datenblatt.
Flankensteilheit	$^{0,125}/_{Datenrate}$
Spannungshub	„110 “
Vorverzerrung	„010 “
Entzerrung	„01 “
Terminierung	„VTTX “
Kopplung	„DC “

Tabelle 6.18: Verwendete Parameter in der HyperTransport-Simulation.

S6 GTP							
	VCCTX	VCCRX	TTX	TTRX	PLL	VCCINT	Gesamt
2x	0,0718	0,046	0,118	0,042	0,068	0,023	**0,740**
4x	0,1436	0,092	0,236	0,085	0,137	0,046	**1,111**
8x	0,2873	0,184	0,473	0,170	0,274	0,093	**1,853**
16x	0,5746	0,369	0,946	0,341	0,548	0,187	**3,706**
V5 GTP							
	VCCTX	VCCRX	TTX	TTRX	PLL	VCCINT	Gesamt
2x	0,035	0,0230	0,039	0,042	0,044	0,014	**0,404**
4x	0,071	0,0461	0,078	0,085	0,088	0,028	**0,603**
8x	0,143	0,0922	0,157	0,170	0,176	0,056	**1,001**
16x	0,287	0,1845	0,315	0,341	0,352	0,112	**2,002**
V5 GTX							
	VTTX	VTTR	VCCX	PLL	VCCINT		Gesamt
2x	0,08	0,044	0,08	0,03793	0,009		**0,5497**
4x	0,16	0,088	0,16	0,07586	0,019		**0,8016**
8x	0,32	0,176	0,32	0,15172	0,039		**1,3055**
16x	0,64	0,352	0,64	0,30344	0,079		**2,6110**
V4 MGT							
	VCCX	VCCR	VTTR	VCCAUX	VTTX	VCCINT	Gesamt
2x	0,0491	0,0491	0,003	0,008	0,112	0,026	**0,620**
4x	0,0983	0,0983	0,006	0,016	0,224	0,052	**0,868**
8x	0,1967	0,1967	0,013	0,032	0,449	0,104	**1,364**
16x	0,3934	0,3934	0,027	0,064	0,898	0,208	**2,729**
V6 GTX							
	VTTX	VTTR	VCCX	VCCINT			Gesamt
2x	0,096	0,078	0,075	0,0125			**0,526**
4x	0,193	0,157	0,150	0,025			**0,789**
8x	0,386	0,315	0,300	0,05			**1,315**
16x	0,773	0,630	0,600	0,1			**2,631**

Tabelle 6.19: Leistungsaufnahme der Versorgungsleitungen bei HyperTransport 0,8 in Watt.

S6 GTP							
	VCCTX	VCCRX	TTX	TTRX	PLL	VCCINT	Gesamt
2x	0,072	0,046	0,118	0,043	0,069	0,047	**0,790**
4x	0,145	0,092	0,236	0,086	0,138	0,093	**1,185**
8x	0,290	0,185	0,473	0,171	0,275	0,187	**1,975**
16x	0,579	0,369	0,946	0,342	0,551	0,374	**3,950**
V5 GTP							
	VCCTX	VCCRX	TTX	TTRX	PLL	VCCINT	Gesamt
2x	0,036	0,023	0,039	0,043	0,036	0,022	**0,400**
4x	0,072	0,046	0,079	0,086	0,072	0,044	**0,599**
8x	0,145	0,092	0,158	0,171	0,144	0,088	**0,998**
16x	0,290	0,185	0,315	0,342	0,288	0,176	**1,996**
V5 GTX							
	VTTX	VTTR	VCCX	PLL	VCCINT		Gesamt
2x	0,070	0,040	0,080	0,060	0,016		**0,605**
4x	0,140	0,080	0,160	0,120	0,031		**0,870**
8x	0,280	0,160	0,320	0,239	0,063		**1,401**
16x	0,560	0,320	0,640	0,478	0,125		**2,802**
V4 MGT							
	VCCX	VCCR	VTTR	VCCAUX	VTTX	VCCINT	Gesamt
2x	0,123	0,123	0,003	0,008	0,113	0,052	**0,963**
4x	0,247	0,247	0,007	0,016	0,226	0,104	**1,387**
8x	0,494	0,494	0,013	0,032	0,452	0,208	**2,233**
16x	0,987	0,987	0,027	0,064	0,905	0,416	**4,466**
V6 GTX							
	VTTX	VTTR	VCCX	VCCINT			Gesamt
2x	0,097	0,079	0,107	0,025			**0,617**
4x	0,194	0,158	0,214	0,050			**0,925**
8x	0,389	0,316	0,429	0,100			**1,542**
16x	0,777	0,631	0,858	0,200			**3,083**
V6 GTH							
	VTTX	VCCR	PLL	VCCX	VCCINT		Gesamt
2x	0,074	0,087	0,170	0,172	0,027		**1,232**
4x	0,148	0,174	0,341	0,172	0,054		**1,590**
8x	0,296	0,349	0,682	0,343	0,108		**2,479**
16x	0,592	0,698	1,364	0,686	0,215		**4,958**

Tabelle 6.20: Leistungsaufnahme der Versorgungsleitungen bei HyperTransport 1,6 in Watt.

S6 GTP							
	VCCTX	VCCRX	TTX	TTRX	PLL	VCCINT	Gesamt
2x	0,073	0,046	0,118	0,043	0,069	0,094	**0,886**
4x	0,147	0,092	0,236	0,086	0,138	0,188	**1,329**
8x	0,293	0,184	0,472	0,171	0,275	0,377	**2,216**
16x	0,586	0,369	0,945	0,343	0,551	0,754	**4,432**
V5 GTP							
	VCCTX	VCCRX	TTX	TTRX	PLL	VCCINT	Gesamt
2x	0,037	0,023	0,039	0,043	0,036	0,044	**0,444**
4x	0,073	0,046	0,079	0,086	0,072	0,088	**0,666**
8x	0,147	0,092	0,157	0,171	0,144	0,176	**1,110**
16x	0,293	0,184	0,315	0,343	0,288	0,352	**2,220**
V5 GTX							
	VTTX	VTTR	VCCX	PLL	VCCINT		Gesamt
2x	0,07	0,044	0,082	0,060	0,031		**0,651**
4x	0,14	0,088	0,164	0,120	0,063		**0,938**
8x	0,28	0,176	0,328	0,239	0,125		**1,512**
16x	0,56	0,352	0,656	0,478	0,25		**3,024**
V4 MGT							
	VCCX	VCCR	VTTR	VCCAUX	VTTX	VCCINT	Gesamt
2x	0,148	0,148	0,003	0,008	0,114	0,104	**1,179**
4x	0,297	0,297	0,006	0,016	0,228	0,208	**1,705**
8x	0,594	0,594	0,013	0,032	0,456	0,416	**2,757**
16x	1,187	1,187	0,025	0,064	0,912	0,832	**5,514**
V6 GTX							
	VTTX	VTTR	VCCX	VCCINT			Gesamt
2x	0,097	0,079	0,090	0,05			**0,633**
4x	0,195	0,158	0,180	0,1			**0,950**
8x	0,390	0,316	0,361	0,2			**1,583**
16x	0,779	0,631	0,722	0,4			**3,166**
V6 GTH							
	VTTX	VCCR	PLL	VCCX	VCCINT		Gesamt
2x	0,074	0,087	0,170	0,188	0,054		**1,335**
4x	0,148	0,174	0,341	0,188	0,107		**1,720**
8x	0,296	0,349	0,682	0,376	0,215		**2,679**
16x	0,592	0,698	1,364	0,752	0,430		**5,358**

Tabelle 6.21: Leistungsaufnahme der Versorgungsleitungen bei HyperTransport 2,8 in Watt.

V5 GTX							
	VTTX	VTTR	VCCX	PLL	VCCINT		Gesamt
2x	0,070	0,040	0,086	0,060	0,063		**0,717**
4x	0,140	0,080	0,172	0,120	0,125		**1,036**
8x	0,280	0,160	0,344	0,239	0,250		**1,672**
16x	0,560	0,320	0,688	0,478	0,500		**3,345**
V4 MGT							
	VCCX	VCCR	VTTR	VCCAUX	VTTX	VCCINT	Gesamt
2x	0,188	0,188	0,003	0,008	0,115	0,208	**1,543**
4x	0,375	0,375	0,006	0,016	0,229	0,416	**2,251**
8x	0,750	0,750	0,012	0,032	0,459	0,832	**3,669**
16x	1,501	1,501	0,024	0,064	0,917	1,664	**7,338**
V6 GTX							
	VTTX	VTTR	VCCX	VCCINT			Gesamt
2x	0,098	0,079	0,026	0,100			**0,604**
4x	0,195	0,157	0,051	0,200			**0,906**
8x	0,391	0,315	0,102	0,400			**1,510**
16x	0,781	0,630	0,204	0,800			**3,020**
V6 GTH							
	VTTX	VCCR	PLL	VCCX	VCCINT		Gesamt
2x	0,074	0,099	0,206	0,222	0,107		**1,637**
4x	0,148	0,198	0,412	0,222	0,215		**2,123**
8x	0,296	0,396	0,823	0,443	0,430		**3,317**
16x	0,591	0,791	1,647	0,886	0,859		**6,633**

Tabelle 6.22: Leistungsaufnahme der Versorgungsleitungen bei HyperTransport 5,2 in Watt.

V5 GTX							
	VTTX	VTTR	VCCX	PLL	VCCINT		Gesamt
2x	0,070	0,020	0,088	0,075	0,078		**0,755**
4x	0,140	0,040	0,176	0,150	0,156		**1,086**
8x	0,280	0,080	0,352	0,300	0,313		**1,748**
16x	0,560	0,160	0,704	0,600	0,625		**3,497**
V4 MGT							
	VCCX	VCCR	VTTR	VCCAUX	VTTX	VCCINT	Gesamt
2x	0,197	0,197	0,003	0,008	0,115	0,261	**1,686**
4x	0,394	0,394	0,006	0,016	0,230	0,522	**2,467**
8x	0,788	0,788	0,012	0,032	0,460	1,045	**4,030**
16x	1,577	1,577	0,024	0,064	0,919	2,090	**8,059**
V6 GTX							
	VTTX	VTTR	VCCX	VCCINT			Gesamt
2x	0,097	0,079	0,075	0,125			**0,751**
4x	0,193	0,158	0,150	0,250			**1,127**
8x	0,387	0,315	0,300	0,500			**1,878**
16x	0,774	0,631	0,600	1,000			**3,756**
V6 GTH							
	VTTX	VCCR	PLL	VCCX	VCCINT		Gesamt
2x	0,074	0,105	0,225	0,229	0,120		**1,736**
4x	0,148	0,209	0,450	0,229	0,240		**2,260**
8x	0,296	0,419	0,901	0,459	0,480		**3,537**
16x	0,591	0,837	1,801	0,918	0,961		**7,073**

Tabelle 6.23: Leistungsaufnahme der Versorgungsleitungen bei HyperTransport 6,4 in Watt.

Transceiver	PDP	PDP_{code}	PDP_{real}
V4 MGT 2x	3,87882E-10	5,60611E-10	6,61521E-10
V4 MGT 4x	2,71467E-10	3,92354E-10	4,62978E-10
V4 MGT 8x	2,13259E-10	3,08226E-10	3,63707E-10
V4 MGT 16x	2,13259E-10	3,08226E-10	3,63707E-10
V5 GTP 2x	2,5253E-10	3,64984E-10	4,30681E-10
V5 GTP 4x	1,88487E-10	2,72423E-10	3,21459E-10
V5 GTP 8x	1,56466E-10	2,26142E-10	2,66847E-10
V5 GTP 16x	1,56466E-10	2,26142E-10	2,66847E-10
V5 GTX 2x	3,43605E-10	4,96616E-10	5,86007E-10
V5 GTX 4x	2,50527E-10	3,6209E-10	4,27266E-10
V5 GTX 8x	2,03988E-10	2,94826E-10	3,47895E-10
V5 GTX 16x	2,03988E-10	2,94826E-10	3,47895E-10
S6 GTP 2x	4,63041E-10	6,69239E-10	7,89702E-10
S6 GTP 4x	3,47406E-10	5,0211E-10	5,9249E-10
S6 GTP 8x	2,89588E-10	4,18545E-10	4,93883E-10
S6 GTP 16x	2,89588E-10	4,18545E-10	4,93883E-10
V6 GTX 2x	3,2888E-10	4,75334E-10	5,60894E-10
V6 GTX 4x	2,4666E-10	3,565E-10	4,2067E-10
V6 GTX 8x	2,0555E-10	2,97084E-10	3,50559E-10
V6 GTX 16x	2,0555E-10	2,97084E-10	3,50559E-10

Tabelle 6.24: Ergebnisse der Effizienzbetrachtung bei HyperTransport 0,8 in Joule.

Transceiver	PDP	PDP_{code}	PDP_{real}
V4 MGT 2x	3,01074E-10	4,35146E-10	5,13472E-10
V4 MGT 4x	2,16656E-10	3,13136E-10	3,695E-10
V4 MGT 8x	1,74447E-10	2,52131E-10	2,97514E-10
V4 MGT 16x	1,74447E-10	2,52131E-10	2,97514E-10
V5 GTP 2x	1,24847E-10	1,80443E-10	2,12923E-10
V5 GTP 4x	9,36008E-11	1,35282E-10	1,59633E-10
V5 GTP 8x	7,79775E-11	1,12702E-10	1,32988E-10
V5 GTP 16x	7,79775E-11	1,12702E-10	1,32988E-10
V5 GTX 2x	1,88962E-10	2,73108E-10	3,22268E-10
V5 GTX 4x	1,35956E-10	1,96498E-10	2,31868E-10
V5 GTX 8x	1,09453E-10	1,58193E-10	1,86668E-10
V5 GTX 16x	1,09453E-10	1,58193E-10	1,86668E-10
S6 GTP 2x	2,46755E-10	3,56638E-10	4,20833E-10
S6 GTP 4x	1,85121E-10	2,67558E-10	3,15719E-10
S6 GTP 8x	1,54304E-10	2,23018E-10	2,63161E-10
S6 GTP 16x	1,54304E-10	2,23018E-10	2,63161E-10
V6 GTX 2x	1,92693E-10	2,78501E-10	3,28631E-10
V6 GTX 4x	1,4452E-10	2,08876E-10	2,46474E-10
V6 GTX 8x	1,20433E-10	1,74063E-10	2,05395E-10
V6 GTX 16x	1,20433E-10	1,74063E-10	2,05395E-10
V6 GTH 2x	3,8493E-10	5,56344E-10	6,56486E-10
V6 GTH 4x	2,48502E-10	3,59163E-10	4,23813E-10
V6 GTH 8x	1,93687E-10	2,79938E-10	3,30327E-10
V6 GTH 16x	,93687E-10	2,79938E-10	3,30327E-10

Tabelle 6.25: Ergebnisse der Effizienzbetrachtung bei HyperTransport 1,6 in Joule.

Transceiver	PDP	PDP_{code}	PDP_{real}
V4 MGT 2x	2,10504E-10	3,04244E-10	3,59008E-10
V4 MGT 4x	1,52218E-10	2,20002E-10	2,59603E-10
V4 MGT 8x	1,23075E-10	1,77882E-10	2,099E-10
V4 MGT 16x	1,23075E-10	1,77882E-10	2,099E-10
V5 GTP 2x	7,93309E-11	1,14658E-10	1,35296E-10
V5 GTP 4x	5,94813E-11	8,59691E-11	1,01444E-10
V5 GTP 8x	4,95566E-11	7,16247E-11	8,45172E-11
V5 GTP 16x	4,95566E-11	7,16247E-11	8,45172E-11
V5 GTX 2x	1,16235E-10	1,67996E-10	1,98235E-10
V5 GTX 4x	8,37478E-11	1,21042E-10	1,42829E-10
V5 GTX 8x	6,75041E-11	9,75645E-11	1,15126E-10
V5 GTX 16x	6,75041E-11	9,75645E-11	1,15126E-10
S6 GTP 2x	1,58198E-10	2,28646E-10	2,69802E-10
S6 GTP 4x	1,18685E-10	1,71537E-10	2,02414E-10
S6 GTP 8x	9,89286E-11	1,42983E-10	1,6872E-10
S6 GTP 16x	9,89286E-11	1,42983E-10	1,6872E-10
V6 GTX 2x	1,13066E-10	1,63416E-10	1,92831E-10
V6 GTX 4x	8,47998E-11	1,22562E-10	1,44623E-10
V6 GTX 8x	7,06665E-11	1,02135E-10	1,2052E-10
V6 GTX 16x	7,06665E-11	1,02135E-10	1,2052E-10
V6 GTH 2x	2,38376E-10	3,44528E-10	4,06543E-10
V6 GTH 4x	1,53602E-10	2,22003E-10	2,61964E-10
V6 GTH 8x	1,19609E-10	1,72872E-10	2,03989E-10
V6 GTH 16x	1,19609E-10	1,72872E-10	2,03989E-10

Tabelle 6.26: Ergebnisse der Effizienzbetrachtung bei HyperTransport 2,8 in Joule.

Transceiver	PDP	PDP_{code}	PDP_{real}
V4 MGT 2x	1,48326E-10	2,14377E-10	2,52965E-10
V4 MGT 4x	1,08243E-10	1,56445E-10	1,84605E-10
V4 MGT 8x	8,82014E-11	1,27479E-10	1,50425E-10
V4 MGT 16x	8,82014E-11	1,27479E-10	1,50425E-10
V5 GTX 2x	6,89824E-11	9,97011E-11	1,17647E-10
V5 GTX 4x	4,97945E-11	7,19686E-11	8,49229E-11
V5 GTX 8x	4,02005E-11	5,81023E-11	6,85608E-11
V5 GTX 16x	4,02005E-11	5,81023E-11	6,85608E-11
V6 GTX 2x	5,80682E-11	8,39266E-11	9,90334E-11
V6 GTX 4x	4,35511E-11	6,2945E-11	7,42751E-11
V6 GTX 8x	3,62926E-11	5,24541E-11	6,18959E-11
V6 GTX 16x	3,62926E-11	5,24541E-11	6,18959E-11
V6 GTH 2x	1,57393E-10	2,27481E-10	2,68428E-10
V6 GTH 4x	1,02065E-10	1,47516E-10	1,74069E-10
V6 GTH 8x	7,97278E-11	1,15232E-10	1,35973E-10
V6 GTH 16x	7,97278E-11	1,15232E-10	1,35973E-10

Tabelle 6.27: Ergebnisse der Effizienzbetrachtung bei HyperTransport 5,2 in Joule.

Transceiver	PDP	PDP_{code}	PDP_{real}
V4 MGT 2x	1,31724E-10	1,90382E-10	2,24651E-10 3,2469E-10
V4 MGT 4x	9,63778E-11	1,39296E-10	1,64369E-10 2,37565E-10
V4 MGT 8x	7,87048E-11	1,13753E-10	1,34229E-10 1,94002E-10
V4 MGT 16x	7,87048E-11	1,13753E-10	1,34229E-10 1,94002E-10
V5 GTX 2x	5,89967E-11	8,52686E-11	1,00617E-10 1,45423E-10
V5 GTX 4x	4,24319E-11	6,13273E-11	7,23662E-11 1,04592E-10
V5 GTX 8x	3,41495E-11	4,93567E-11	5,82409E-11 8,41762E-11
V5 GTX 16x	3,41495E-11	4,93567E-11	5,82409E-11 8,41762E-11
V6 GTX 2x	5,86881E-11	8,48226E-11	1,00091E-10 1,44662E-10
V6 GTX 4x	4,4016E-11	6,36169E-11	7,5068E-11 1,08497E-10
V6 GTX 8x	3,668E-11	5,30141E-11	6,25567E-11 9,04139E-11
V6 GTX 16x	3,668E-11	5,30141E-11	6,25567E-11 9,04139E-11
V6 GTH 2x	1,35619E-10	1,96012E-10	2,31294E-10 3,34292E-10
V6 GTH 4x	8,82698E-11	1,27577E-10	1,50541E-10 2,17579E-10
V6 GTH 8x	6,90767E-11	9,98374E-11	1,17808E-10 1,7027E-10
V6 GTH 16x	6,90767E-11	9,98374E-11	1,17808E-10 1,7027E-10

Tabelle 6.28: Ergebnisse der Effizienzbetrachtung bei HyperTransport 6,4 in Joule.

Infiniband

Parameter	Wert
Zeitschritt	5 ps
Temperatur	$25°C$
Spannungen	Typische Werte laut jeweiligem FPGA-Datenblatt.
Flankensteilheit	$^{0,125}/_{Datenrate}$
Spannungshub	„110 “
Vorverzerrung	„001 “
Entzerrung	„01 “
Terminierung	„VTTX “
Kopplung	„DC “

Tabelle 6.29: Verwendete Parameter in der Infiniband-Simulation.

S6 GTP							
	VCCTX	VCCRX	TTX	TTRX	PLL	VCCINT	Gesamt
1x	0,037	0,024	0,067	0,066	0,069	0,047	**0,310**
4x	0,146	0,097	0,269	0,263	0,138	0,188	**1,100**
12x	0,438	0,291	0,806	0,788	0,413	0,565	**3,301**
V5 GTP							
	VCCTX	VCCRX	TTX	TTRX	PLL	VCCINT	Gesamt
1x	0,018	0,012	0,017	0,016	0,036	0,022	**0,122**
4x	0,073	0,049	0,067	0,066	0,072	0,088	**0,414**
12x	0,219	0,146	0,201	0,197	0,216	0,264	**1,243**
V5 GTX							
	VTTX	VTTR	VCCX	PLL	VCCINT		Gesamt
1x	0,030	0,016	0,041	0,060	0,016		**0,162**
4x	0,120	0,064	0,164	0,120	0,063		**0,530**
12x	0,360	0,192	0,492	0,359	0,188		**1,590**
V4 MGT							
	VCCX	VCCR	VTTR	VCCAUX	VTTX	VCCINT	Gesamt
1x	0,074	0,148	0,002	0,008	0,106	0,052	**0,316**
4x	0,148	0,296	0,003	0,016	0,213	0,208	**0,736**
12x	0,296	0,887	0,010	0,048	0,638	0,624	**2,207**
V6 GTX							
	VTTX	VTTR	VCCX	VCCINT			Gesamt
1x	0,043	0,037	0,072	0,025			**0,176**
4x	0,171	0,146	0,287	0,100			**0,704**
12x	0,513	0,439	0,861	0,300			**2,113**

Tabelle 6.30: Leistungsaufnahme der Versorgungsleitungen bei Infiniband SDR in Watt.

V5 GTX							
	VTTX	VTTR	VCCX	PLL	VCCINT		Gesamt
1x	0,030	0,016	0,043	0,060	0,031		**0,180**
4x	0,120	0,064	0,172	0,120	0,125		**0,601**
12x	0,360	0,192	0,516	0,359	0,375		**1,802**
V4 MGT							
	VCCX	VCCR	VTTR	VCCAUX	VTTX	VCCINT	Gesamt
1x	0,094	0,187	0,001	0,008	0,107	0,104	**0,407**
4x	0,187	0,374	0,002	0,016	0,213	0,416	**1,022**
12x	0,374	1,123	0,007	0,048	0,640	1,248	**3,066**
V6 GTX							
	VTTX	VTTR	VCCX	VCCINT			Gesamt
1x	0,043	0,037	0,124	0,050			**0,253**
4x	0,172	0,146	0,494	0,200			**1,013**
12x	0,516	0,439	1,483	0,600			**3,038**
V6 GTH							
	VTTX	VCCR	PLL	VCCX	VCCINT		Gesamt
1x	0,041	0,051	0,101	0,222	0,054		**0,469**
4x	0,165	0,204	0,405	0,222	0,215		**1,211**
12x	0,496	0,613	1,215	0,665	0,644		**3,633**

Tabelle 6.31: Leistungsaufnahme der Versorgungsleitungen bei Infiniband DDR in Watt.

V6 GTH						
	VTTX	VCCR	PLL	VCCX	VCCINT	Gesamt
1x	0,041	0,063	0,147	0,288	0,107	**0,647**
4x	0,166	0,252	0,590	0,288	0,430	**1,725**
12x	0,497	0,755	1,770	0,865	1,289	**5,175**

Tabelle 6.32: Leistungsaufnahme der Versorgungsleitungen bei Infiniband QDR in Watt.

Transceiver	PDP	PDP_{code}	PDP_{real}
V4 MGT 1x	1,26322E-10	1,63068E-10	1,79374E-10
V4 MGT 4x	7,35608E-11	9,49591E-11	1,04455E-10
V4 MGT 12x	7,35608E-11	9,49591E-11	1,04455E-10
V5 GTP 1x	4,86512E-11	6,28035E-11	6,90839E-11
V5 GTP 4x	4,14432E-11	5,34988E-11	5,88486E-11
V5 GTP 12x	4,14432E-11	5,34988E-11	5,88486E-11
V5 GTX 1x	6,49717E-11	8,38716E-11	9,22587E-11
V6 GTX 4x	5,30117E-11	6,84325E-11	7,52757E-11
V6 GTX 12x	5,30117E-11	6,84325E-11	7,52757E-11
S6 GTP 1x	1,23816E-10	1,59833E-10	1,75817E-10
S6 GTP 4x	1,10046E-10	1,42058E-10	1,56263E-10
S6 GTP 12x	1,10046E-10	1,42058E-10	1,56263E-10
V6 GTX 1x	7,04394E-11	9,09298E-11	1,00023E-10
V6 GTX 4x	7,04394E-11	9,09298E-11	1,00023E-10
V6 GTX 12x	7,04394E-11	9,09298E-11	1,00023E-10
V6 GTH 1x	1,46097E-10	1,88595E-10	2,07455E-10
V6 GTH 4x	8,96938E-11	1,15785E-10	1,27364E-10
V6 GTH 12x	8,96938E-11	1,15785E-10	1,27364E-10

Tabelle 6.33: Ergebnisse der Effizienzbetrachtung bei Infiniband SDR in Joule.

Transceiver	PDP	PDP_{code}	PDP_{real}
V4 MGT 1x	8,13981E-11	1,05076E-10	1,15584E-10
V4 MGT 4x	5,10991E-11	6,59635E-11	7,25598E-11
V4 MGT 12x	5,10991E-11	6,59635E-11	7,25598E-11
V5 GTX 1x	3,60108E-11	4,64862E-11	5,11348E-11
V5 GTX 4x	3,00308E-11	3,87666E-11	4,26433E-11
V5 GTX 12x	3,00308E-11	3,87666E-11	4,26433E-11
V6 GTX 1x	5,06286E-11	6,53561E-11	7,18917E-11
V6 GTX 4x	5,06286E-11	6,53561E-11	7,18917E-11
V6 GTX 12x	5,06286E-11	6,53561E-11	7,18917E-11
V6 GTH 1x	9,37871E-11	1,21069E-10	1,33176E-10
V6 GTH 4x	6,05501E-11	7,81638E-11	8,59801E-11
V6 GTH 12x	6,05501E-11	7,81638E-11	8,59801E-11

Tabelle 6.34: Ergebnisse der Effizienzbetrachtung bei Infiniband DDR in Joule.

Transceiver	PDP	PDP_{code}	PDP_{real}
V6 GTH 1x	6,47371E-11	8,35687E-11	9,19256E-11
V6 GTH 4x	4,31236E-11	5,5668E-11	6,12348E-11
V6 GTH 12x	4,31236E-11	5,5668E-11	6,12348E-11

Tabelle 6.35: Ergebnisse der Effizienzbetrachtung bei Infiniband QDR in Joule.

PCI

Parameter	Wert
Zeitschritt	2 ns
Temperatur	$25°C$
Spannungen	Typische Werte laut jeweiligem FPGA-Datenblatt.
Flankensteilheit	100 ps

Tabelle 6.36: Verwendete Parameter in der PCI-Simulation.

PCI 33			
Transceiver	PDP	PDP_{code}	PDP_{real}
Virtex-5	2,29812E-10	4,59623E-10	5,74529E-10
Virtex-4	2,83145E-10	5,66289E-10	7,07861E-10
PCI-X 66			
Transceiver	PDP	PDP_{code}	PDP_{real}
Virtex-5	1,78705E-10	3,57411E-10	4,46764E-10
Virtex-4	2,29054E-10	4,58109E-10	5,72636E-10
PCI-X 133			
Transceiver	PDP	PDP_{code}	PDP_{real}
Virtex-5	1,50287E-10	3,00574E-10	3,75718E-10
Virtex-4	1,87069E-10	3,74138E-10	4,67673E-10

Tabelle 6.37: Ergebnisse der Effizienzbetrachtung bei PCI in Joule.

PCI 33						
	Virtex-5			**Virtex-4**		
Signal	Sender	Empfänger	Subtotal	Sender	Empfänger	Subtotal
AD	0,0044	0,0019	0,2007	0,0054	0,0023	0,2483
CLK	0,0073	0,0031	0,0104	0,0086	0,0037	0,0123
CBE	0,0032	0,0011	0,0168	0,0038	0,0013	0,0205
IRDY	0,0032	0,0011	0,0042	0,0038	0,0013	0,0051
TRDY	0,0032	0,0011	0,0042	0,0038	0,0013	0,0051
DEVSEL	0,0025	0,0006	0,0032	0,0031	0,0008	0,0038
FRAME	0,0025	0,0006	0,0032	0,0031	0,0008	0,0038
GNT	0,0025	0,0006	0,0032	0,0031	0,0008	0,0038
	0,2427			**0,2990**		
PCI-X 66						
	Virtex-5			**Virtex-4**		
Signal	Sender	Empfänger	Subtotal	Sender	Empfänger	Subtotal
AD	0,0073	0,0031	0,6661	0,0091	0,0039	0,8576
CLK	0,0123	0,0053	0,0176	0,0154	0,0066	0,0220
CBE	0,0047	0,0016	0,0502	0,0058	0,0019	0,0621
IRDY	0,0047	0,0016	0,0063	0,0058	0,0019	0,0078
TRDY	0,0047	0,0016	0,0063	0,0058	0,0019	0,0078
DEVSEL	0,0034	0,0008	0,0042	0,0041	0,0010	0,0051
FRAME	0,0034	0,0008	0,0042	0,0041	0,0010	0,0051
GNT	0,0034	0,0008	0,0042	0,0041	0,0010	0,0051
	0,7549			**0,9675**		
PCI-X 133						
	Virtex-5			**Virtex-4**		
Signal	Sender	Empfänger	Subtotal	Sender	Empfänger	Subtotal
AD	0,0124	0,0053	1,1306	0,0147	0,0063	1,3848
CLK	0,0225	0,0096	0,0321	0,0301	0,0129	0,0430
CBE	0,0078	0,0026	0,0833	0,0113	0,0038	0,1202
IRDY	0,0078	0,0026	0,0104	0,0113	0,0038	0,0150
TRDY	0,0078	0,0026	0,0104	0,0113	0,0038	0,0150
DEVSEL	0,0050	0,0013	0,0063	0,0057	0,0014	0,0072
FRAME	0,0050	0,0013	0,0063	0,0057	0,0014	0,0072
GNT	0,0050	0,0013	0,0063	0,0057	0,0014	0,0072
	1,2792			**1,5923**		

Tabelle 6.38: Leistungsaufnahme der Versorgungsleitungen bei PCI in Watt.

FSB

Parameter	Wert
Zeitschritt	2 ns
Temperatur	$25°C$
Spannungen	Typische Werte laut jeweiligem FPGA-Datenblatt.
Flankensteilheit	100 ps

Tabelle 6.39: Verwendete Parameter in der FSB-Simulation.

Virtex-5				
Verfahren	Takt	Daten	Adressen	Total
FSB200	0,011519	0,01067	0,010668	**1,3652**
FSB266	0,010871	0,011798	0,010907	**1,4524**
FSB333	0,011121	0,01228	0,011138	**1,5011**
FSB400	0,011568	0,012647	0,01067	**1,5161**
FSB400	0,011519	0,012647	0,01067	**1,5155**
FSB533	0,010871	0,013488	0,011798	**1,6156**
FSB667	0,011121	0,01427	0,01228	**1,6968**
FSB800	0,011568	0,0148697	0,012647	**1,7621**
Virtex-4				
Verfahren	Takt	Daten	Adressen	Total
FSB200	0,0503015	0,0511696	0,047675	**2,1172775**
FSB266	0,0519203	0,0534218	0,0489786	**2,1977552**
FSB333	0,0541062	0,0559722	0,0502067	**2,2870743**
FSB400	0,0550112	0,0579499	0,0511696	**2,3533137**
FSB400	0,0503015	0,0579499	0,0511696	**2,3344749**
FSB533	0,0519203	0,0603	0,0534218	**2,4297646**
FSB667	0,0541062	0,0622466	0,0559722	**2,5209772**
FSB800	0,0550112	0,0642243	0,0579499	**2,6004091**

Tabelle 6.40: Leistungsaufnahme der Versorgungsleitungen bei FSB in Watt.

Virtex-4			
Verfahren	PDP	PDP_{code}	PDP_{real}
FSB200	1,65412E-10	3,30825E-10	4,13531E-10
FSB266	1,29097E-10	2,58195E-10	3,22744E-10
FSB333	1,07314E-10	2,14628E-10	2,68285E-10
FSB400	9,19263E-11	1,83853E-10	2,29816E-10
FSB400	7,1229E-11	1,42458E-10	1,78073E-10
FSB533	5,91445E-11	1,18289E-10	1,47861E-10
FSB667	5,07892E-11	1,01578E-10	1,26973E-10
FSB800	5,07892E-11	1,01578E-10	1,26973E-10
Virtex-5			
Verfahren	PDP	PDP_{code}	PDP_{real}
FSB200	1,06656E-10	2,13313E-10	2,66641E-10
FSB266	8,53191E-11	1,70638E-10	2,13298E-10
FSB333	7,04352E-11	1,4087E-10	1,76088E-10
FSB400	5,92234E-11	1,18447E-10	1,48058E-10
FSB400	4,73634E-11	9,47269E-11	1,18409E-10
FSB533	3,98107E-11	7,96214E-11	9,95268E-11
FSB667	3,44169E-11	6,88338E-11	8,60423E-11
FSB800	3,44169E-11	6,88338E-11	8,60423E-11

Tabelle 6.41: Ergebnisse der Effizienzbetrachtung bei FSB in Joule.

Ethernet über verdrillte Leiterpaare

100BaseTX			
Leistung	PDP	PDP_{code}	PDP_{real}
0,244	1,95421E-09	2,51116E-09	2,83761E-09
1000BaseT			
Leistung	PDP	PDP_{code}	PDP_{real}
0,975	6,50099E-10	1,00245E-09	1,13277E-09

Tabelle 6.42: Leistungsaufnahme der Versorgungsleitungen und Energie bei TP-Ethernet in Joule und Watt.

Medienunabhängige Schnittstellen

Parameter	Wert
Zeitschritt	2 ns
Temperatur	$25°C$
Spannungen	Typische Werte laut jeweiligem FPGA-Datenblatt.
Flankensteilheit	100 ps

Tabelle 6.43: Verwendete Parameter in der MII-Simulation.

Virtex-5				
Verfahren	Leistung	PDP	PDP_{code}	PDP_{real}
MII	0,052	8,20426E-10	8,43398E-10	9,5304E-10
RMII	0,033	3,1278E-10	3,21538E-10	3,6334E-10
GMII	0,098	9,8454E-11	1,01211E-10	1,1437E-10
RGMII	0,058	5,8174E-11	5,9803E-11	6,7577-10
XGMII	0,400	4,00234E-11	4,11443E-11	4,6493E-11
Virtex-4				
Verfahren	Leistung	PDP	PDP_{code}	PDP_{real}
MII	0,082	8,20426E-10	8,43398E-10	9,5304E-10
RMII	0,066	6,58518E-10	6,76956E-10	7,6496E-10
GMII	0,120	1,19989E-10	1,23349E-10	1,3938E-10
RGMII	0,094	9,38701E-11	9,6499E-11	1,0904E-10
XGMII	0,406	4,06113E-11	4,17484E-11	4,7176E-11

Tabelle 6.44: Leistungsaufnahme der Versorgungsleitungen und Energie bei MIIs in Watt und Joule.

Ethernet über Backplane und Twinax-Kabel

Parameter	Wert
Zeitschritt	5 ps
Temperatur	$25°C$
Spannungen	Typische Werte laut jeweiligem FPGA-Datenblatt.
Flankensteilheit	$^{0,125}/_{Datenrate}$
Spannungshub	„110 “
Vorverzerrung	„010 “
Entzerrung	„11 “
Terminierung	„VTTX “
Kopplung	„DC “

Tabelle 6.45: Verwendete Parameter in der 10G-Ethernet-Simulation.

10GBase-CX4				
Transceiver	Leistung	PDP	PDP_{code}	PDP_{real}
V4-MGT	1,183	1,11831E-10	1,43703E-10	1,62384E-10
V5-GTP	0,372	2,97241E-11	3,81955E-11	4,31609E-11
V5-GTX	0,576	4,60644E-11	5,91927E-11	6,68878E-11
S6-GTP	0,741	5,92773E-11	7,61713E-11	8,60736E-11
V6-GTX	0,761	6,08839E-11	7,82358E-11	8,84065E-11
V6-GTH	1,015	8,11825E-11	1,0432E-10	1,17881E-10
10GBase-SFP+CU				
Transceiver	Leistung	PDP	PDP_{code}	PDP_{real}
V4-MGT	1,259	1,00737E-10	1,29447E-10	1,46275E-10
V5-GTP	0,468	3,74567E-11	4,81319E-11	5,4389E-11
V5-GTX	0,749	5,99082E-11	7,69821E-11	8,69897E-11
S6-GTP	0,963	7,70107E-11	9,89587E-11	1,11823E-10
V6-GTX	0,879	7,03107E-11	9,03492E-11	1,02095E-10
V6-GTH	1,072	8,57958E-11	1,10248E-10	1,2458E-10
10GBase-KX4				
Transceiver	Leistung	PDP	PDP_{code}	PDP_{real}
V4-MGT	1,326	1,06067E-10	1,36296E-10	1,54015E-10
V5-GTP	0,672	5,37616E-11	6,90837E-11	7,80645E-11
V5-GTX	1,031	8,25149E-11	1,06032E-10	1,19816E-10
S6-GTP	0,672	1,081E-10	1,38908E-10	1,56966E-10
V6-GTX	1,100	8,80584E-11	1,13155E-10	1,27865E-10
V6-GTH	1,288	1,03057E-10	1,32428E-10	1,49644E-10
10GBase-KR				
Transceiver	Leistung	PDP	PDP_{code}	PDP_{real}
V6-GTH	0,564	5,47301E-11	5,71416E-11	6,457E-11
40GBase-CR4				
Transceiver	Leistung	PDP	PDP_{code}	PDP_{real}
V6-GTH	1,82	44E-12	46E-12	52E-12
40GBase-KR4				
Transceiver	Leistung	PDP	PDP_{code}	PDP_{real}
V6-GTH	1,907	46E-12	48E-12	54E-12
100GBase-CR10				
Transceiver	Leistung	PDP	PDP_{code}	PDP_{real}
V6-GTH	4,208	41E-12	43E-12	48E-12

Tabelle 6.46: Leistungsaufnahme der Versorgungsleitungen und Energie bei 10 Gigabit Ethernet in Watt und Joule.

Aurora

Parameter	Wert
Zeitschritt	5 ps
Temperatur	$25°C$
Spannungen	Typische Werte laut jeweiligem FPGA-Datenblatt.
Flankensteilheit	120 ps
Spannungshub	„111 “
Vorverzerrung	„111 “
Entzerrung	„11 “
Terminierung	„VTTX “
Kopplung	„DC “

Tabelle 6.47: Verwendete Parameter in der Aurora-Simulation.

S6 GTP							
Gbit/s	VCCTX	VCCRX	TTX	TTRX	PLL	VCCINT	Gesamt
0,6	0,036	0,024	0,053	0,015	0,069	0,012	**0,208**
1,25	0,036	0,024	0,056	0,015	0,069	0,024	**0,224**
2,5	0,036	0,024	0,063	0,013	0,069	0,047	**0,253**
3,125	0,037	0,025	0,066	0,013	0,070	0,059	**0,269**
V5 GTP							
Gbit/s	VCCTX	VCCRX	VTTX	TTRX	PLL	VCCINT	Gesamt
0,6	0,018	0,012	0,018	0,015	0,044	0,007	**0,114**
1,25	0,018	0,012	0,019	0,015	0,036	0,011	**0,110**
2,5	0,018	0,012	0,021	0,013	0,036	0,022	**0,123**
3,125	0,018	0,012	0,022	0,013	0,043	0,028	**0,134**
V5 GTX							
Gbit/s	VTTX	VTTR	VCCX	PLL	VCCINT		Gesamt
0,6	0,033	0,015	0,041	0,038	0,005		**0,132**
1,25	0,033	0,015	0,042	0,060	0,008		**0,157**
2,5	0,033	0,015	0,043	0,060	0,016		**0,166**
3,125	0,033	0,015	0,043	0,075	0,020		**0,186**
6,25	0,034	0,015	0,045	0,075	0,039		**0,207**
V4 MGT							
Gbit/s	VCCX	VCCR	VAUX	VTTX	VCCINT		Gesamt
0,6	0,024	0,049	0,008	0,106	0,013		**0,176**
1,25	0,061	0,122	0,008	0,106	0,026		**0,263**
2,5	0,073	0,146	0,008	0,107	0,052		**0,314**
3,125	0,085	0,171	0,008	0,107	0,065		**0,351**
6,25	0,098	0,196	0,008	0,107	0,131		**0,442**
V6 GTX							
Gbit/s	VTTX	VTTR	VCCX	VINT			Gesamt
0,6	0,043	0,036	0,033	0,006			**0,119**
1,25	0,043	0,036	0,047	0,013			**0,138**
2,5	0,043	0,036	0,072	0,025			**0,176**
3,125	0,043	0,036	0,085	0,031			**0,195**
6,25	0,043	0,036	0,149	0,063			**0,291**
V6 GTH							
Gbit/s	VTTX	VCCR	PLL	VCCX	VCCINT		Gesamt
1,25	0,041	0,042	0,055	0,172	0,013		**0,309**
2,5	0,041	0,045	0,084	0,188	0,027		**0,358**
3,125	0,041	0,046	0,089	0,192	0,030		**0,369**
6,25	0,041	0,054	0,116	0,229	0,060		**0,440**
10	0,041	0,064	0,151	0,288	0,107		**0,544**

Tabelle 6.48: Leistungsaufnahme der Versorgungsleitungen bei Aurora x1 in Watt.

S6 GTP						
Verfahren	x1	x2	x4	x8	x16	x24
Aurora 0,6	0,208	0,348	0,696	1,393	2,786	4,179
Aurora 1,25	0,224	0,378	0,757	1,513	3,027	4,540
Aurora 2,5	0,253	0,437	0,874	1,748	3,497	5,245
Aurora 3,125	0,269	0,468	0,935	1,870	3,741	5,611
V5 GTP						
Verfahren	x1	x2	x4	x8	x16	x24
Aurora 0,6	0,114	0,184	0,368	0,741	1,472	2,208
Aurora 1,25	0,111	0,185	0,371	0,732	1,482	2,224
Aurora 2,5	0,123	0,210	0,419	0,880	1,677	2,516
Aurora 3,125	0,136	0,229	0,458	0,910	1,833	2,749
V5 GTX						
Verfahren	x1	x2	x4	x8	x16	x24
Aurora 0,6	0,132	0,226	0,452	0,904	1,807	2,711
Aurora 1,25	0,157	0,255	0,509	1,019	2,037	3,056
Aurora 2,5	0,166	0,273	0,546	1,092	2,184	3,275
Aurora 3,125	0,186	0,297	0,593	1,187	2,374	3,561
Aurora 6,25	0,209	0,342	0,685	1,369	2,738	4,107
V4 MGT						
Verfahren	x1	x2	x4	x8	x16	x24
Aurora 0,6	0,176	0,246	0,492	0,985	1,970	2,954
Aurora 1,25	0,263	0,418	0,837	1,674	3,348	5,022
Aurora 2,5	0,314	0,520	1,040	2,080	4,161	6,241
Aurora 3,125	0,351	0,596	1,191	2,382	4,764	7,146
Aurora 6,25	0,442	0,776	1,553	3,106	6,211	9,317
V6 GTX						
Verfahren	x1	x2	x4	x8	x16	x24
Aurora 0,6	0,119	0,238	0,476	0,951	1,902	2,854
Aurora 1,25	0,138	0,277	0,554	1,107	2,215	3,322
Aurora 2,5	0,176	0,352	0,704	1,408	2,817	4,225
Aurora 3,125	0,195	0,391	0,782	1,563	3,127	4,690
Aurora 6,25	0,291	0,582	1,164	2,328	4,657	6,985
V6 GTH						
Verfahren	x1	x2	x4	x8	x16	x24
Aurora 1,25	0,309	0,474	0,777	1,553	3,107	4,660
Aurora 2,5	0,358	0,582	0,976	1,953	3,905	7,810
Aurora 3,125	0,369	0,605	1,018	2,036	4,073	6,109
Aurora 6,25	0,441	0,772	1,315	2,630	5,260	7,890
Aurora 10	0,544	1,015	1,742	3,483	6,966	10,449

Tabelle 6.49: Leistungsaufnahme der Versorgungsleitungen bei Aurora in Watt, bei unterschiedlichen Bitbreiten.

Aurora 0,6						
Metrik	V4MGT	V5GTP	V5GTX	S6GTP	V6GTX	V6GTH
PDP	2,94E-10	1,90E-10	2,19E-10	3,47E-10	1,98E-10	
PDP_{code}	3,67E-10	2,37E-10	2,74E-10	4,34E-10	2,47E-10	
Aurora 1,25						
Metrik	V4MGT	V5GTP	V5GTX	S6GTP	V6GTX	V6GTH
PDP	2,10E-10	8,85E-11	1,25E-10	1,78E-10	1,1E-10	2,474E-10
PDP_{code}	2,62E-10	1,10E-10	1,57E-10	2,2E-10	1,384E-10	3,093E-10
Aurora 2,5						
Metrik	V4MGT	V5GTP	V5GTX	S6GTP	V6GTX	V6GTH
PDP	1,25E-10	4,91E-11	6,65E-11	1,01E-10	7,04E-11	1,43E-10
PDP_{code}	1,56E-10	6,14E-11	8,31E-11	1,26E-10	8,80E-11	1,79E-10
Aurora 3,125						
Metrik	V4MGT	V5GTP	V5GTX	S6GTP	V6GTX	V6GTH
PDP	1,12E-10	4,35E-11	5,94E-11	8,60E-11	6,25E-11	1,17E-10
PDP_{code}	1,40E-10	5,44E-11	7,43E-11	1,07E-10	7,81E-11	1,47E-10
Aurora 6,5						
Metrik	V4MGT	V5GTP	V5GTX	S6GTP	V6GTX	V6GTH
PDP	7,07E-11		3,33E-11		4,65E-11	7,05E-11
PDP_{code}	8,84E-11		4,17E-11		5,82E-11	8,81E-11
Aurora 10						
Metrik	V4MGT	V5GTP	V5GTX	S6GTP	V6GTX	V6GTH
PDP						5,44E-11
PDP_{code}						6,80E-11

Tabelle 6.50: Ergebnisse der Effizienzbetrachtung bei Aurora x1 in Joule.