

# Ethernet-basierte Ultra-Hochgeschwindigkeitskommunikation für eine Regelung dezentraler Einspeisumrichter von Windenergieanlagen

Holger Flatt<sup>1</sup>, Sebastian Schriegel<sup>1</sup>, Jan F. Westerkamp<sup>2</sup>,  
Ingo Mackensen<sup>2</sup>, Albrecht Gensior<sup>3</sup> und Jürgen Jasperneite<sup>1</sup>

<sup>1</sup>Fraunhofer IOSB, Institutsteil für industrielle Automation (IOSB-INA), Lemgo  
{Holger.Flatt, Sebastian.Schriegel, Juergen.Jasperneite}@iosb-ina.fraunhofer.de

<sup>2</sup>WRD Wobben Research and Development GmbH, Aurich  
{Jan.Westerkamp, Ingo.Mackensen}@enercon.de

<sup>3</sup> Fachgebiet Leistungselektronik und Steuerungen in der Elektroenergie-technik, Technische Universität  
Ilmenau  
Albrecht.Gensior@tu-ilmenau.de

**Abstract:** In diesem Beitrag wird eine Ethernet-Ultra-Hochgeschwindigkeitskommunikation vorgestellt, welche auf der Nutzung von GBit-Ethernet-PHYs in Verbindung mit einer FPGA-Protokollverarbeitung basiert. Hierzu wurde ein mit IEEE 802-Standards kompatibles Protokoll entwickelt, welches es dem Empfänger ermöglicht, Daten bereits vor dem vollständigen Empfang eines Paketes zu nutzen. Die Protokollverarbeitung wurde vollständig in Hardware umgesetzt, welche alle Verarbeitungsschritte zwischen der GBit-Schnittstelle des PHYs und der systeminternen Datenschnittstelle vornimmt. Im Rahmen einer Fallstudie wurde ein FPGA-basierter Systemaufbau zur quantitativen Ermittlung der Kommunikationseigenschaften umgesetzt. In jedem Zyklus sendet dabei eine Zentralsteuerung Echtzeitdaten zur Ansteuerung von IGBT-Umrichter-Brücken und empfängt jeweils Strommesswerte. Für die in der Anwendung relevanten Mengen an zu übertragenden Echtzeitdaten wurden an dem Aufbau Ende-zu-Ende-Latenzen in der Größenordnung  $< 1 \mu\text{s}$  sowie ein Jitter  $< 50 \text{ ns}$  gemessen. Aufgrund der Protokollverarbeitung repräsentieren diese Werte die Kommunikationseigenschaften zwischen den späteren Applikationen, welche in der Steuerung und den verteilten IGBT-Umrichterbrücken ausgeführt werden.

## 1 Einleitung

Die Erhöhung des energetischen Wirkungsgrades ist heute für viele technische Systeme ein primäres Entwurfsziel. Technische Einrichtungen wie Einspeisumrichter von Windenergieanlagen profitieren von höheren Wirkungsgraden häufig davon, dass durch eine reduzierte Verlustleistung eine Reduktion der Baugrößen und damit auch eine Reduktion von Anschaffungskosten möglich wird.

Da Einspeisumrichter für Windenergieanlagen von Enercon auf Grund von Skalierbarkeit, Transportierbarkeit und Wartbarkeit heute vielfach modular aufgebaut sind, sind Strommesspunkte und IGBT-Ansteuerungen (IGBT) sowie die zentrale Steuerung der Umrichtereinheiten räumlich verteilt. Insbesondere bei einer zentralen Regelung der Ströme ist eine leistungsfähige Vernetzungslösung mit einer sehr geringen Latenz sowie Jitter erforderlich, da Kommunikationszeiten die Regelungsleistung negativ beeinflussen [CA18]. Werden darüber hinaus Regler für Schaltungsaufgaben eingesetzt, muss die Regelungsaufgabe innerhalb kürzester Zykluszeiten in Echtzeit erfolgen. Dabei werden die Strommesswerte aller Umrichtereinheiten (bis zu acht Stück parallel) ausgewertet und auf dieser Grundlage den IGBT-Halbbrücken aktualisierte Ansteuerungssignale zur Durchführung der Schaltungen übermittelt. Für aktuelle Anwendungen werden in der Literatur Regelungsfrequenzen von bis zu 640 kHz gefordert [Ge20].

Im Rahmen dieses Beitrages wird ein neuer Ansatz einer Ultra-Hochgeschwindigkeitskommunikation vorgestellt. Dieser basiert auf der Nutzung der GBit-Ethernet-Technologie, welche sich durch Kosteneffizienz auszeichnet. Die in einem FPGA umgesetzte Protokollverarbeitung, welche auf IEEE 802-Standards basiert, ermöglicht eine beschleunigte Auswertung von Echtzeitdaten und latenzarme Kopplung von Regler, IGBT-Ansteuerung und Strommesselektronik. Die Lösung ist im Sinne eines Plug-and-Play entworfen, so dass keine manuellen Konfigurationsschritte bei der Inbetriebnahme durchgeführt werden müssen.

Der Beitrag ist wie folgt gegliedert: Abschnitt 2 stellt zunächst die applikativen Anforderungen an die Regelung dezentraler Einspeisumrichter von Windenergieanlagen dar. Abschnitt 3 thematisiert den neuen Ansatz zur Hochgeschwindigkeitskommunikation und fokussiert insbesondere die beiden Bereiche Protokollentwicklung und die daraus resultierende Hardware-Architektur. In den Abschnitten 4 und 5 werden eine Fallstudie und die

daraus resultierenden Ergebnisse vorgestellt. Abschnitt 6 fasst den Beitrag zusammen und gibt einen Ausblick auf zukünftige Arbeiten.

## **2 Anforderungen an die Regelung dezentraler Einspeisenumrichter von Windenergieanlagen**

Für eine neue Generation von Windenergieanlagen des Unternehmens Enercon soll eine Regelung von acht IGBT-Halbbrücken pro Phase über eine Zentralsteuerung erfolgen, welche wiederum an ein überlagertes System angeschlossen ist. Eine Anforderungsanalyse des Unternehmens Enercon hat ergeben, dass bei einer Regelung eine Kommunikation von Echtzeitdaten zwischen Umrichtereinheiten mit den IGBT-Halbbrücken und der Zentralsteuerung in weniger als  $1 \mu\text{s}$  und mit einem Jitter  $< 100 \text{ ns}$  erfolgen muss. Echtzeitdaten sind beispielsweise die Stromdaten der IGBTs sowie die Schaltsignale für die IGBTs. Bei dem geplanten Aufbau, welcher jeweils einzelne Schaltschränke für die Zentralsteuerung und die Umrichter vorsieht, kann mit einer Leitungslänge von 10 m zwischen der Zentralsteuerung und den jeweiligen Umrichtereinheiten gerechnet werden.

Als weitere Anforderungen neben den extrem kleinen Latenzzeiten sind (1) eine zukunftsrobuste Technologielösung durch Nutzung von Standards, (2) Kosten und (3) einfache Inbetriebnahme in den Entwurf der Lösungsarchitektur eingeflossen.

Um Anforderungen an kürzeste Kommunikationszeiten in der Größenordnung von  $1 \mu\text{s}$ , niedrige Hardware-Kosten für Kommunikationskomponenten (z.B. PHYs) sowie Leitungslängen von bis zu 10 m zu ermöglichen, wurde GBit-Ethernet (1000BASE-T) als Kommunikationstechnologie ausgewählt. Eine unidirektionale Übertragung eines minimal großen Paketes (7 Byte Präambel, 1 Byte SFD sowie 64 Datenbytes) benötigt bei einer GBit-Übertragungsrates 576 ns ( $72 \text{ Byte} * 8 \text{ ns / Byte}$ ). Hinzu kommen noch Verzögerungszeiten der PHYs ( $0,4 \mu\text{s}$  am Beispiel von [Ma18]), interne Verarbeitungszeiten in der Zentralsteuerung sowie den Umrichtereinheiten zur Paketerzeugung bzw. Analyse. Um eine Ende-zu-Ende-Latenz von  $1 \mu\text{s}$  signifikant zu unterschreiten, sind daher weitere Optimierungsmaßnahmen an der Ethernet-Kommunikation notwendig.

Echtzeit-Kommunikationssysteme ermöglichen häufig Linientopologien, welche eine lineare Abhängigkeit der Verzögerungszeiten von der Anzahl der Geräte aufweisen [Ma19]. Um kürzeste Kommunikationszeiten ermöglichen zu können, ist es erforderlich, die Zentralsteuerung über parallele Punkt-zu-Punkt-Verbindungen direkt mit den Umrichtereinheiten zu koppeln und somit eine Sterntopologie aufzubauen.

Als weitere Anforderungen gilt es, neben den Echtzeitdaten, weitere Daten wie z.B. Temperaturen zu übertragen, Übertragungsfehler zuverlässig zu erkennen sowie Synchronisationsmechanismen für die zyklische Kommunikation zur Verfügung zu stellen.

Ein derartiges Kommunikationssystem, welches eine Zentralsteuerung mit acht Umrichtereinheiten unter den gegebenen Randbedingungen und Anforderungen vernetzen kann, ist bisher nicht bekannt.

## **3 Neuer Ansatz zur Hochgeschwindigkeitskommunikation**

Marktverfügbare und zukünftige Echtzeit-Kommunikationssysteme wie Profinet oder Ethernet TSN-basierte Systeme können den verschiedenen Anforderungen und Eigenschaften spezifischer Maschinen, Anlagen und anderer Applikationen z.B. durch Auswahl der Geräte, Konfigurationsoberflächen oder Plug-and-Play-Mechanismen angepasst werden [Re18]. Anpassbar sind dabei z.B. die Topologie, Adressierung, Anzahl der Geräte oder Datenraten. Eine reine Sterntopologie reicht für die Anwendungen nicht aus [Uc18]. Die Systeme müssen dadurch Bridging, d.h. die Weiterleitung von Frames nutzen, um verschiedene Topologiemöglichkeiten realisieren zu können. Als Anforderung für die Bridging-Latenzen bei Gigabit-Datenrate wird häufig der Wert  $< 1 \mu\text{s}$  gefordert [Es20]. Da die Anforderungen der Umrichterapplikation, die in diesem Beitrag beschrieben ist, eine Ende-zu-Ende-Latenz von  $< 1 \mu\text{s}$  fordert, können Systeme, die TSN-Bridging verwenden, nicht integriert werden.

Um kürzeste Ende-zu-Ende-Latenzen und einen minimierten Jitter zu ermöglichen, wird ein Kommunikationsansatz auf Basis von IEEE-Ethernet-Technologie und einer reinen Sterntopologie vorgeschlagen, welcher zudem auf einem für den gegebenen Anwendungszweck optimierten Protokoll basiert sowie eine Protokollverarbeitung in Hardware (hier FPGAs) ermöglicht. Abbildung 1 stellt die Gesamtarchitektur bestehend aus Zentralsteuerung und Umrichtereinheiten dar. In den nachfolgenden Abschnitten erfolgt eine Beschreibung der einzelnen Teilthemen.

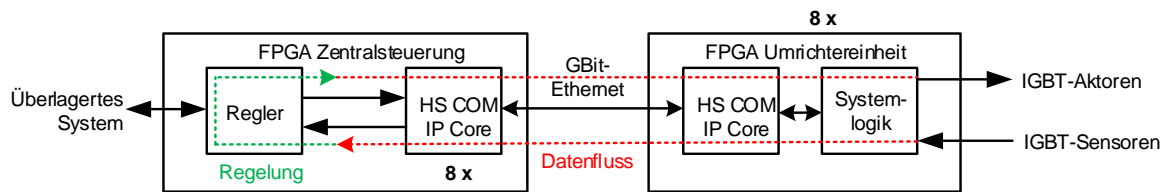


Abbildung 1: Gesamtarchitektur bestehend aus Zentralsteuerung und Umrichtereinheiten

### 3.1 Regelung und Zeitverhalten

Im Rahmen der Regelung der Ströme eines Umrichtersystems mit Hilfe von IGBT-Halbbrücken, wovon jeweils drei einen Umrichter bilden, ist es erforderlich, Sensordaten der IGBT-Halbbrücken (z.B. Stromwerte) an die Zentralsteuerung zu übertragen. Der Regler berechnet anschließend Ansteuerungssignale für die IGBT-Halbbrücken, welche dann beginnend mit einem neuen Kommunikationszyklus zurück an die Umrichtereinheiten gesendet werden.

Da die Übertragung bei Ethernet in Paketen erfolgt, werden alle Echtzeitdaten jeweils pro Zyklus in einem einzigen Paket versendet. Üblicherweise muss dabei der Empfänger warten, bis das gesamte Paket empfangen ist und kann dann erst die Datenintegrität mit Hilfe des CRCs am Ende des Paketes prüfen. Um die Übertragungslatenz zu reduzieren, werden die Echtzeitdaten am Anfang des Paketes übertragen und mit Hilfe eines weiteren CRCs abgesichert, so dass diese bereits vor Übertragungsende des vollständigen Paketes genutzt werden können. Die Sendezeitpunkte der IGBT-Sensordaten leiten sich aus den Empfangszeitpunkten der IGBT-Aktordaten ab und können für eine Optimierung der Regelungsapplikation zeitlich mit einem Offset verschoben werden. Abbildung 2 visualisiert das beschriebene Zeitverhalten des entwickelten Kommunikationsansatzes zwischen Zentralsteuerung und Umrichtereinheiten.

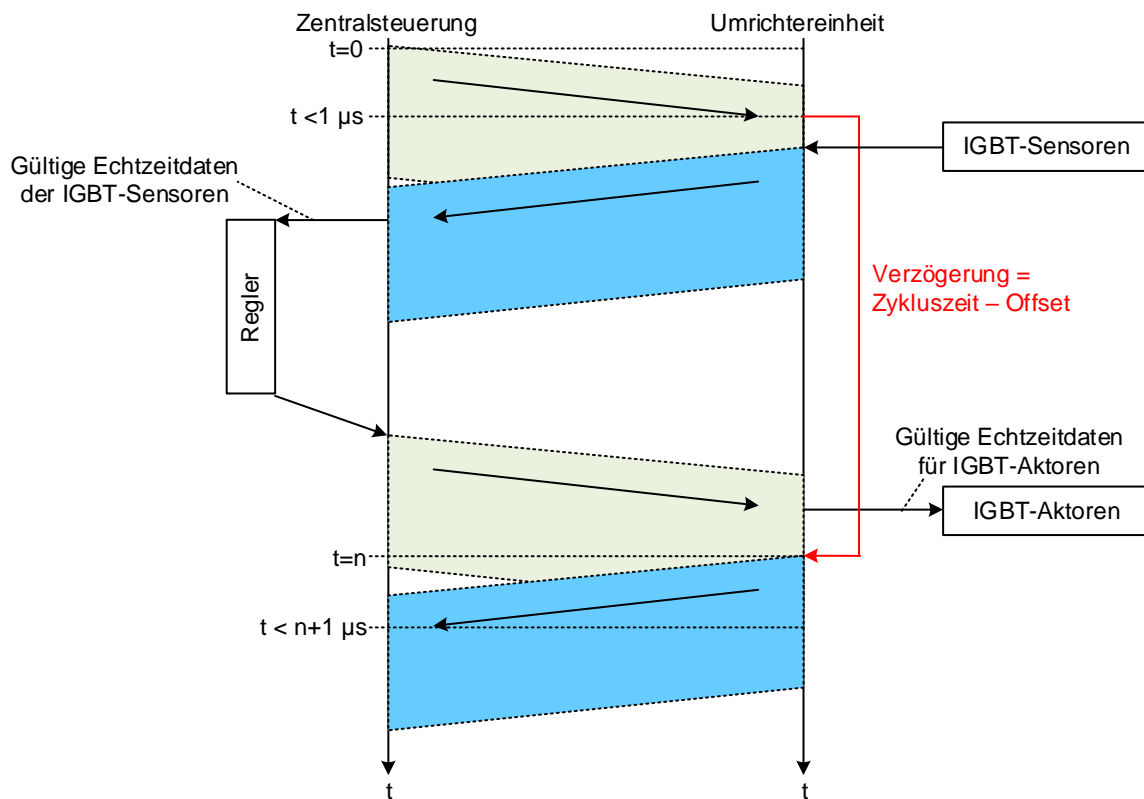


Abbildung 2: Kommunikationsansatz zwischen Zentralsteuerung und Umrichtereinheiten

### 3.2 Schnittstelle zum überlagerten System

Die von den Umrichtereinheiten gesendeten Messdaten liegen im FPGA der Zentralsteuerung vor und werden von dem dort implementierten Teil des Regelungsalgorithmus verarbeitet. Ein zweiter Teil dieses Algorithmus

ist in einer Echtzeitanwendung implementiert, die von einem der Prozessoren des verwendeten SoC abgearbeitet wird. Über eine Schnittstelle zwischen FPGA und Prozessor (implementiert als AXI-Bus) tauschen beide Teile des Regelungsalgorithmus Daten miteinander aus. Eine detaillierte Beschreibung findet sich in [Ge20].

### 3.3 Optimiertes Kommunikationsprotokoll

Zur Umsetzung des in Abschnitt 2 vorgestellten Regelungsansatzes wurde ein Ethernet-Layer-2-Protokoll konzipiert (vgl. Tabelle 1), welches die abgeleiteten Anforderungen wie in den folgenden Abschnitten beschrieben ist, umsetzt:

#### **Standardkonformität**

Um im Betrieb mit Standard-Werkzeugen Diagnosen der Kommunikation vornehmen zu können, besteht das Protokoll auf den grundlegenden Eigenschaften des IEEE 802.3-Standards. Da insbesondere die Präambel und der Start-of-Frame-Delimiter am Anfang des Paketes, der Standard-Ethernet-CRC am Paketende sowie die zulässigen Paketgrößen von 64-1514 Byte berücksichtigt sind, kann mit üblichen Werkzeugen (z.B. Ethernet-Tap in Verbindung mit Wireshark) eine Aufzeichnung und Diagnose einzelner Pakete ermöglicht werden.

#### **Optimierung auf Effizienz und einfache Möglichkeit zur Umsetzung in Hardware**

Mit dem Ziel, eine einfache Umsetzung in Hardware (z.B. FPGA oder ASIC) zu ermöglichen, wird ein einheitliches Paketformat für beide Übertragungsrichtungen genutzt.

Da die Zentralsteuerung über dedizierte Verbindungen mit den Umrichtereinheiten kommuniziert, ist im Normalfall keine Adressierung mit Angabe von Quelle und Ziel erforderlich. Wird aber berücksichtigt, dass in realen Anlagen eine Vertauschung der Leitungen möglich ist, muss hier eine systemseitige Erkennungsmöglichkeit gegeben sein. Gemäß IEEE 802.3 könnten hier per Standard die jeweils 6 Bit großen Felder Source und Destination genutzt werden. Da für die beabsichtigte Umrichtersteuerung max. 9 Geräte im Netzwerk vorhanden sind, ermöglicht ein 1-Bit-Adressfeld mit einer Aufteilung in zwei 4-Bit-Felder eine deutlich ressourceneffizientere Adressierungsmöglichkeit. Im Anschluss an das Adressfeld erfolgt eine Hardware-Versionsnummer, welche eine Erkennung unterschiedlicher Firmware-Stände auf den Geräten ermöglicht.

#### **Latenzoptimierte Übertragung von Echtzeitdaten**

Im Anschluss an die Hardware-Versionsnummer können in der Anzahl  $n$  parametrisierbare Bytes mit Echtzeitdaten übertragen werden. Bei üblicher Ethernet-Kommunikation wären auf Empfängerseite die empfangenen Echtzeitdaten erst gültig, wenn am Paketende die CRC als gültig erkannt wurde. Um die Echtzeitdaten frühzeitig auf Empfängerseite nutzen zu können, sichert ein 4-Byte-Intermediate CRC die zuvor empfangenen Daten ab. Als weitere Optimierungsmaßnahme wird die Ethernet-Präambel von 7 auf 2 Byte verkürzt.

#### **Zusätzliche Übertragung von Nicht-Echtzeitdaten**

Weitere mit der Anzahl  $m$  parametrisierbare Bytes an Nichtechtzeitdaten folgen im Anschluss an den Intermediate-CRC. Sofern erforderlich, müssen diese um  $p$  Padding Bytes ergänzt werden, um eine minimale Paketgröße gemäß IEEE 802.3 von 64 Byte nicht zu unterschreiten.

#### **Erkennung von Übertragungsfehlern**

Ein wesentliches Merkmal, welches vom Protokoll ermöglicht wird, ist die Erkennung von Fehlern. Neben der Erkennung vertauschter Leitungen zwischen den Geräten, unterschiedlichen Firmware-Ständen im Netzwerk sowie CRC-Fehlern muss es auch möglich sein, weitere folgende Fehler zu erkennen:

- Falsche Paketgröße
- Verlorene Pakete
- Link-Unterbrechungen

Falsche Paketgrößen können vom Empfänger einfach erkannt werden, da jedes Paket dieselbe Größe aufweist. Verlorene Pakete werden über einen 2-Byte-Frame Counter erkannt, der nach jedem Frame inkrementiert wird. Link-Unterbrechungen können entweder ebenfalls über den Frame Counter erkannt werden oder durch eine Überwachung der Linkverbindung in Hardware. Weitere mögliche Fehlertypen, wie z.B. falsche Paket-Reihenfolgen und doppeltes Empfangen von Paketen können in der Praxis ausgeschlossen werden, da diese

aufgrund der Punkt-zu-Punktverbindungen nicht möglich sind. Eine Erkennung ist dennoch über den Frame-Counter möglich.

Tabelle 1: Ethernet basiertes Kommunikationsprotokoll für die Regelung dezentraler Einspeisumrichter von Windenergieanlagen

Feldname	Größe in Bytes	Erläuterung
Preamble and SFD	3	Verkürzte Ethernet Präambel und SFD
Addressing	1	Nutzung von 2 Nibbles zur Adressierung von Datenquelle und -ziel
Hardware Version	1	Hardware-Versions-Index zur Absicherung unterschiedlicher Firmware-Stände
High Speed Com	n	Nutzdaten für schnelle Echtzeitkommunikation
Intermediate CRC	4	CRC-Absicherung der zuvor übertragenden Inhalte
Low Speed Com	m	Nutzdaten für Nicht-Echtzeitkommunikation
Padding	p	Einfügen von Nullen zur Vergrößerung des Frames auf 64 Byte
Frame Counter	2	Erkennung nicht sequentiell übertragener oder verlorener Frames
Sync Offset	2	Offset zur Anpassung des Zeitpunktes für Datenrücksendung durch Umrichter
Error field	1	Rückübertragung von Fehlercodes an Kommunikationspartner
CRC	4	Standard Ethernet CRC

### 3.4 IP Core für Ethernet-basierte Ultra-Hochgeschwindigkeitskommunikation

Zur Umsetzung des in Tabelle 1 vorgestellten Protokolls wurde ein FPGA-basierter IP Core entwickelt, welcher den Datenaustausch zwischen der PHY-Schnittstelle (RGMII) und einer systemseitigen internen Schnittstelle (RX Daten, TX Daten) vornimmt. Voraussetzung zur Nutzung ist, dass die komplette PHY-Schnittstelle über den Logikteil des FPGA erreichbar ist (nicht über den CPU Teil im Fall von SoC-FPGAs). Ergänzend wird ein interner Systemtakt (SYSCLK) benötigt, welcher synchron zur systemseitigen Datenschnittstelle ist. Der IP Core ist modular aufgebaut und besteht aus den folgenden Komponenten:

Ein Konverter, welcher mit der RGMII-Schnittstelle verbunden ist, stellt systemseitig eine GMII-Schnittstelle zur Verfügung, da die RGMII-Schnittstelle Daten im Double Data Rate-Betrieb (DDR) verarbeitet und systemseitig ein Single Data Rate-Betrieb erforderlich ist. Angeschlossen an die GMII-Schnittstelle sind jeweils ein RX- und ein TX-MAC Core, welche die MAC-Schicht des Ethernet Layer 2 umsetzen (Präambel und SFD hinzufügen und entfernen sowie CRC am Frameende einfügen bzw. kontrollieren). Die RX- und TX-MAC Cores kommunizieren mit zwei Modulen (Frame Decoder und Frame Encoder), welche das in Tabelle 1 dargestellte Protokoll implementieren. Zur Kommunikation mit den nachgelagerten Einheiten wird hier eine einfache parallele Datenschnittstelle (RX und TX Daten) zur Verfügung gestellt. Die hierfür notwendige Serialisierung bzw. Deserialisierung erfolgen im Encoder bzw. Decoder. Zudem sind alle im Protokoll spezifizierten Funktionen wie z.B. Intermediate CRC, Frame Counter und Fehlerüberprüfung ebenfalls im Encoder und Decoder implementiert. Abbildung 3 stellt die Hardware-Architektur des High Speed Com IP Cores mit den beschriebenen Modulen dar. Grundlegende Eigenschaften, wie z.B. die Menge an Echtzeit- und Nichtechtzeitdaten können separat nach Kommunikationsrichtung parametrisiert werden, so dass ein flexibler an den jeweiligen Anwendungsfall angepasster Betrieb möglich ist.

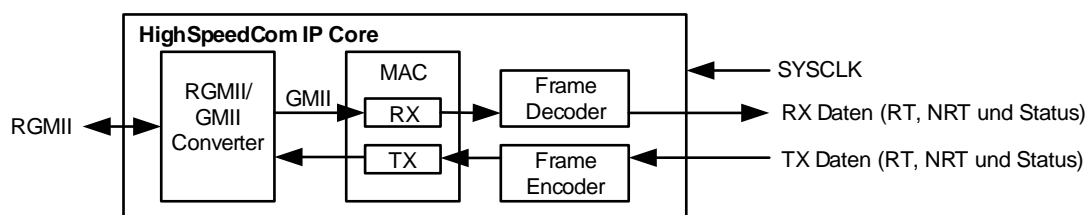


Abbildung 3: Hardware-Architektur des High Speed Com IP Cores

## 4 Fallstudie für entwickelten Prototyp

Im Rahmen einer Fallstudie wurden zur Evaluation der Kommunikation zwei FPGA-Architekturen entwickelt, welche die Zentralsteuerung und eine Umrichtereinheit repräsentieren. In die FPGA-Architektur der Zentralsteuerung wurde neben dem IP-Core ein zyklischer Datengenerator integriert, der das zeitliche Verhalten des Reglers nachbilden soll. Im FPGA der Umrichtereinheit nimmt ein Test Core die empfangenen Daten entgegen und löst Rückantworten aus. Da die Kommunikation in einem späteren Zielsystem parallel über acht identische IP-Cores auf der Zentralsteuerung stattfindet, ist für die Evaluation der Kommunikationseigenschaften eine FPGA-Architektur mit jeweils einem IP Core in der Zentralsteuerung ausreichend.

Um Latenz und Jitter zu messen, wurde ein Systemaufbau (vgl. Abbildung 4) bestehend aus zwei FPGA-Plattformen der Firma Trencz eingesetzt (TE0720-03 FPGA-Boards in Kombination mit TE0706-03 Basis-Boards) vorgenommen, welche jeweils eine Zentralsteuerung (ZS) und eine Umrichtereinheit (UR) nachstellen. Die FPGA-Boards beinhalten jeweils einen Xilinx Zynq XC7Z020CLG484-1 – FPGA.

Die Boards sind über eine 10 m lange Patch-Leitung verbunden, um eine möglichst reale Einbausituation nachzubilden. Auf beiden Boards werden Trigger-Signale erzeugt und über externe GPIO-Pins ausgegeben. Diese zeigen an, sobald das Senden eines neuen Paktes auf Senderseite veranlasst wird (TX Trigger) bzw. die gültigen Echtzeitdaten auf Empfängerseite dem Test Core zur Verfügung gestellt wurden (RX Trigger). Die Trigger-Signale werden mit einem Oszilloskop zur Messung von Latenz und Jitter erfasst. Abbildung 5 visualisiert das zeitliche Verhalten der Trigger-Signale am Beispiel der Übertragung eines Paketes n.

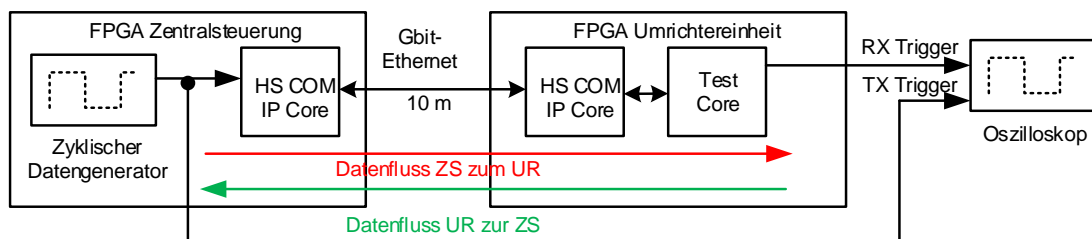


Abbildung 4: Systemaufbau für unidirektionale Messungen von Latenz und Jitter (hier am Beispiel Zentralsteuerung zur Umrichtereinheit)

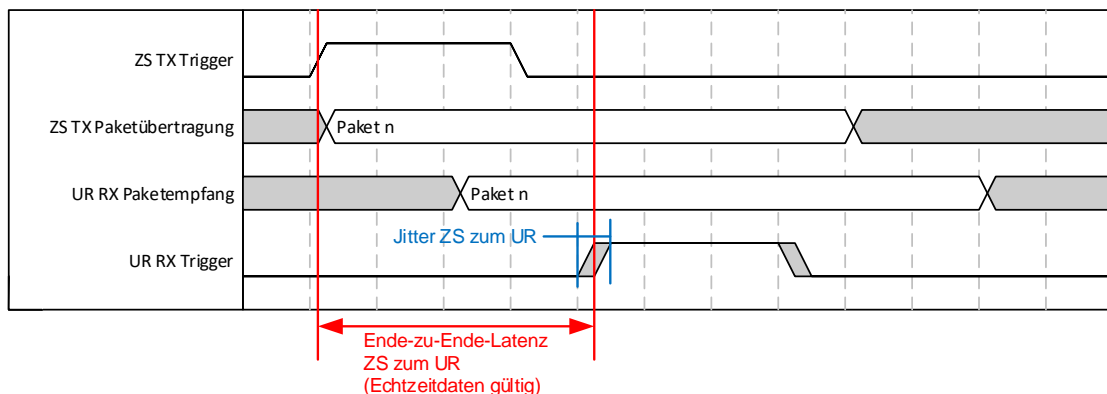


Abbildung 5: Systemaufbau für unidirektionale Messungen von Latenz und Jitter (hier am Beispiel Zentralsteuerung zur Umrichtereinheit)

## 5 Ergebnisse

### 5.1 Verkürzung der Ende-zu-Ende-Übertragungslatenz von Echtzeitdaten über einen Gigabit-Ethernet-Link

Um eine schnelle Übertragung von Echtzeitdaten zu ermöglichen, nutzt das in Tabelle 1 vorgestellte Protokoll eine verkürzte Ethernet-Präambel sowie einen Intermediate CRC zur Absicherung der Daten am Anfang des Paketes. Abbildung 6 zeigt einen Vergleich zwischen einer Standard-Übertragung, bei der alle Daten erst nach Empfang des CRC am Paketende gültig sind und der optimierten Übertragung von Echtzeitdaten, die nach

Empfang des Intermediate CRC gültig sind. Für die Berechnung wurden exemplarisch 17 Byte Nicht-Echtzeitdaten ausgewählt, welche eine Übertragung von Nicht-Echtzeitdaten im Multiplex ermöglichen (16 Byte Nutzdaten und ein Adressbyte).

Das Ergebnis zeigt, dass bis zur Gültigkeit der Echtzeitdaten je nach Größe bei optimierter Übertragung zwischen 31 und 63 Datenbytes weniger übertragen werden müssen. Bei GBit-Ethernet mit 8 ns pro Byte entspricht das einer Reduktion der Übertragungszeit von 248 – 504 ns.

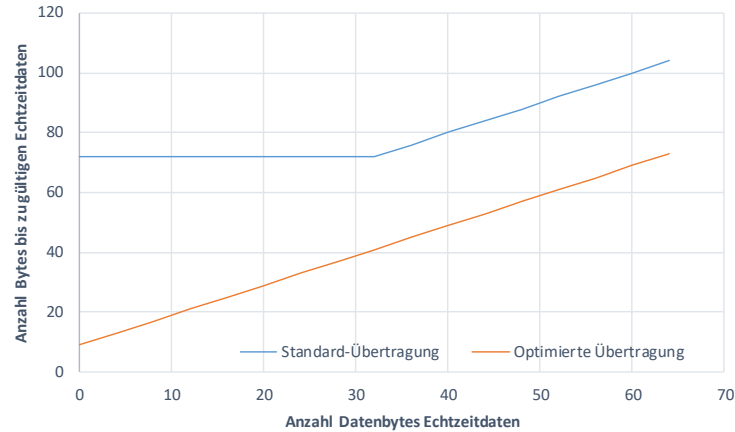


Abbildung 6: Vergleich Standard-Übertragung mit optimierter Übertragung am Beispiel einer Übertragung mit 17 Byte Nicht-Echtzeitdaten für einen Gigabit-Ethernet-Link

## 5.2 Gemessene Ende-zu-Ende-Latenz und Jitter von Echtzeitdaten

Für Messungen von Latenz und Jitter von Echtzeitdaten wurde der in Abbildung 4 dargestellte Aufbau genutzt. Hierzu wurden eine Konfiguration, welche 13 Byte Echtzeitdaten überträgt, untersucht. Diese Datenmenge ist ausreichend, um beispielsweise 3-phasige Stromdaten mit 16 Bit Auflösung (insg. 6 Byte) sowie zugehörige Statusdaten (7 Byte) zu übertragen.

Das Ergebnis der Messung ist in Abbildung 7 dargestellt. Der linke Teil der Abbildung zeigt den mit einem Oszilloskop erfassten Zeitverlauf der Trigger-Signale für eine unidirektionale Kommunikation. Da die gesamte Übertragungskette Bestandteil der Messung ist, sind die Verzögerungen der beiden beteiligten Ethernet-PHYs und die Verzögerung der 10 m Ethernet-Leitung mit enthalten. Im rechten Teil von Abbildung 7 ist eine Vergrößerung dargestellt, welche eine Bestimmung des Jitters ermöglicht. Es entsteht ein kontinuierlicher Jitter, da die Systemtakte, mit denen die beiden Boards operieren, unabhängig voneinander sind. Folgende Ergebnisse wurden ermittelt:

- Ende-zu-Ende-Latenz: ca. 758 ns
- Jitter: ca. 20 ns

Eine Änderung der Datenmenge der Echtzeitdaten hätte eine Latenzänderung um 8 ns/Byte zur Folge sowie eine Änderung der Leitungslänge um ca. 5 ns / m.

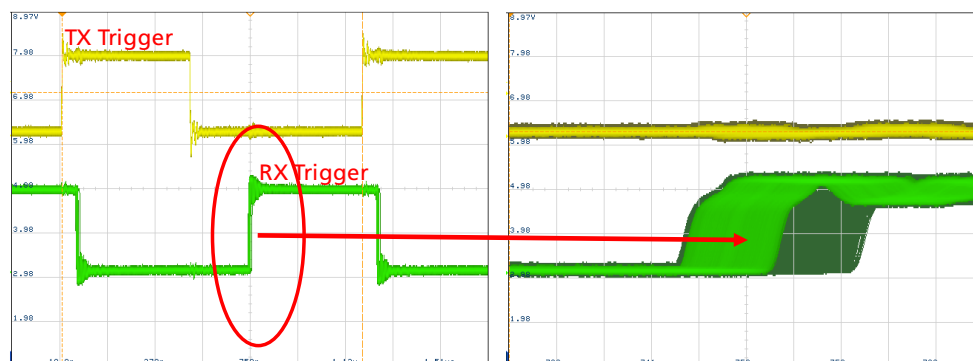


Abbildung 7: Oszilloskop-Screenshot der Messung (hier am Beispiel von 13 Byte Echtzeitdaten)

### 5.3 FPGA-Ressourcen des IP Cores

Für die Belegung der FPGA-Ressourcen sind primär die Anzahl der belegten CLB-LUTs, CLB-Register und MMCM-Module (Digital Clock Manager) sowie BlockRAMs relevant. Die in Abbildung 4 dargestellte Fallstudie (1 Ethernet-Port, 13 Bytes Echtzeitdaten) belegt zwei MMCM-Module zur Generierung von internen Takten. BlockRAMs werden nur von Xilinx-integrierten Debug-Kernen belegt, die für Entwicklungszwecke verwendet werden. Tabelle 2 listet die Anzahl der CLB-LUTs und CLB-Register auf. Das Ergebnis zeigt, dass der entwickelte IP-Core weniger als 2% der benötigten LUTs und Register belegt. Die Ergebnisse sind für eine spätere Umsetzung einer Umrichtereinheit repräsentativ, da diese ebenfalls über nur einen Ethernet-Port verfügt. Für die Zentralsteuerung muss der IP-Core acht Mal instanziiert werden, so dass in diesem Fall dann ca. 15% der LUTs und 11% der Register belegt werden würden.

Tabelle 2: Belegte CLB-LUTs und –Register der Fallstudie

Modul	CLB-LUTs, insg. 53200	CLB-Register, insg. 106400
HS Com IP Core	1000	1455
Test Core	109	364
Debug	1523	2996

## 6 Zusammenfassung und Ausblick

Im Rahmen dieses Beitrages wurde ein neuer Ansatz für eine Ethernet-basierte Ultra-Hochgeschwindigkeitskommunikation für eine Regelung dezentraler Einspeiseumrichter von Windenergieanlagen vorgestellt und evaluiert. Der Ansatz basiert auf einem für die schnelle Übertragung von Echtzeitdaten optimierten Layer-2-Ethernet-Protokoll und nutzt GBit-Ethernet als Übertragungsmedium. Die Protokollverarbeitung wird dabei von einem IP Core umgesetzt. Für die in der Anwendung relevanten Mengen an zu übertragenden Echtzeitdaten wurden an einem realen Aufbau Ende-zu-Ende-Kommunikationslatenzen in der Größenordnung  $< 1 \mu\text{s}$  sowie ein Jitter  $< 50 \text{ ns}$  gemessen. Der vorgestellte Ansatz ermöglicht im Gegensatz zu bestehenden industriellen Kommunikationssystemen diese kurzen Zykluszeiten, da sowohl Kommunikation als auch Applikation innerhalb eines FPGAs vereint sind und somit zusätzliche Verzögerungszeiten und Jitter entfallen. Eine Nutzung für Regelungsaufgaben insbesondere im Bereich der immer wichtiger werdenden steuerbaren Strom-/ Spannungswandlung in anderen Branchen ist möglich, sofern diese ebenfalls auf einer ähnlichen dezentralen Topologie basieren und ähnlich hohe Anforderungen an die Zykluszeit und den Jitter stellen.

In zukünftigen Arbeiten gilt es, den IP-Core in die FPGA-basierten Zielsysteme der Steuerung und des Umrichters zu integrieren und applikationsseitig mit dem Regler (Zentralsteuerung) bzw. der Applikationslogik als Bindeglied zwischen den Sensoren und Aktoren (Umrichtereinheiten) zu verbinden und die Gesamtsysteme in einem realitätsnahen Umfeld zu evaluieren.

## 7 Literaturverzeichnis

- [CA18] T. P. Corrêa, L. Almeida and E. B. Peña, "Hardware/Software Implementation Factors Influencing Ethernet Latency," 2018 IEEE 16th International Conference on Industrial Informatics (INDIN), 2018, pp. 323-328
- [Ge20] Gensior, Albrecht: Approximated sliding-mode control of parallel-connected grid inverters. In: 22nd European Conference on Power Electronics and Applications (EPE'20 ECCE Europe), Lyon, France, 2020
- [Ma18] Marvell, "Alaska® 88E1510/88E1518/88E1512/88E1514 Integrated 10/100/1000 Mbps Energy Efficient Ethernet Transceiver", Datasheet – Public, 2018
- [Ma19] X. Ma and C. Zhang, "A Gigabit Real-time Ethernet for Manufacture Automation Control," 2019 6th International Conference on Systems and Informatics (ICSAI), 2019, pp. 1040-1045
- [Re18] IEC/IEEE 60802 Project Group: Industrial Requirements v12, Online: <http://www.ieee802.org/1/files/public/docs2018/60802-industrial-requirements-1218-v12.pdf>, 2018.
- [Uc18] IEC/IEEE 60802 Project Group: Industrial Use Cases v13, Online: <http://www.ieee802.org/1/files/public/docs2018/60802-industrial-use-cases-0918-v13.pdf>, 2018.
- [Es20] IEC/IEEE 60802 Project Group: IEC/IEEE 60802 Example Selection. Online: <http://www.ieee802.org/1/files/public/docs2020/60802-Steindl-et-al-ExampleSelection-0520-v24.pdf>, 2020.
- [St20] IEC/IEEE 60802 Draft 1.2: Online: <http://www.ieee802.org/1/files/private/60802-drafts/d1/60802-d1.pdf>, 2020.