

Die Österreichische Tagung  
zum Themenbereich des Entwurfs  
integrierter Schaltungen und Systeme

11. Oktober 2006  
Messezentrum Wien

# Austrochip

# OVE

 **IEEE**  
AUSTRIA SECTION



Herausgegeben von  
Peter Balog und Martin Horauer

Fachhochschule Technikum Wien  
Fachbereich für Embedded Systems

# Design eines Switches für deterministische und fehlertolerante Uhrensynchronisation

Georg Gaderer, Patrick Loschmidt  
Research Unit for Integrated Sensor Systems  
Austrian Academy of Sciences  
Viktor Kaplan Strasse 2  
A-2700 Wiener Neustadt, Austria  
{Georg.Gaderer, Patrick.Loschmidt}@OEAW.ac.at

Nikolaus Kerö  
Oregano Systems  
Phorusgasse 8  
1040 Wien, Austria  
keroe@oregano.at

## Kurzfassung

*Ein von der Automatisierungstechnik besonders geförderter Trend ist real-time-fähiges switched Ethernet. Gründe dafür sind nicht nur Interoperabilitätskriterien, sondern auch der durch den Massenmarkt günstig beeinflusste Preis von Ethernet für embedded Systems. Dabei kommt, neben den Traffic-Shaping Verfahren, der altbekannte Mechanismus des Time-Division Multiple Access (TDMA) als Arbitrierungsverfahren für das physikalische Medium zum Einsatz, was wiederum synchronisierte Uhren voraussetzt. Im speziellen Fall von Ethernet kommt einem Netzwerkelement besondere Bedeutung zu: der Switch, ist nicht nur eine wesentliche Stelle im Netzwerk an der Determinismus bezüglich des Timings eingeführt werden muss, er ist auch ein Single Point of Failure, was besondere Anforderungen an Redundanzkonzepte stellt.*

## 1 Einleitung

Der wesentliche Unterschied zwischen verteilten und eng gekoppelten Systemen ist das Fehlen einer gemeinsamen Zeitbasis aufgrund derer Aktionen gesetzt werden. Üblicherweise können dadurch Funktionen, wie das gleichzeitige Schreiben bzw. Lesen von verteilten Daten nicht koordiniert werden. Wesentlich beeinflusst wird auch der Zugriff auf die Netzwerksysteme. Da bei synchronisierten Systemen ein Scheduling des Netzwerkzugriffs implementiert werden kann, wird auch Echtzeitverhalten in modernen Automatisierungsnetzwerken, wie Industrial Ethernet ermöglicht. Eben das ist die Motivation in vernetzten embedded Systems die Uhren zu synchronisieren. Prinzipiell ist das Konzept, synchronisierte Uhren für eine Echtzeitsteuerung zu verwenden, nicht an ein spezielles Medium gebunden, aber der Trend zu Ethernet als Medium für Industriekommunikationsanlagen, legt eine Verwendung dieser Technologie nahe.

In weiterer Folge werden nun nach einer kurzen Darstellung des Stands der Technik, der die grundsätzli-

chen Mechanismen von modernen hochpräzisen Uhrensynchronisationssystemen zeigen soll, ein solches Chipdesign vorgestellt, das eben diese Anforderungen in einem Switch für solche Systeme implementiert.

## 2 Stand der Technik

Moderne Uhrensynchronisations-Protokolle, wie NTP [1] oder IEC/IEEE 1588 [2] fußen auf dem Konzept des Ausmessens des sogenannten Round-Trip Delays. Dabei wird üblicherweise eine Kommunikationsverbindung zwischen zwei Knoten mit symmetrischen Delays vorausgesetzt. Durch das Beobachten der Dauer, die ein Paket benötigt, um von einem Ausgangspunkt wieder zurück zu gelangen, kann nun unter der bereits erwähnten Voraussetzung der Symmetrie der Delays die Verzögerung von Nachrichten in eine Richtung errechnet werden. Dieses sogenannte Round-Trip Delay kann verwendet werden, die üblicherweise als Broad- oder Multicast versendeten Synchronisationspakete (Pakete in dem die aktuelle Zeit verschickt wird) zu korrigieren. Letzteres ist möglich, da man den Zeitpunkt des Verschickens kennt und das Delay bereits ausgemessen hat. Damit kann die Uhrzeit übertragen werden und entfernte Knoten können ihre lokale Zeitbasis absolut gesehen nachkorrigieren.

Obwohl die Uhrensynchronisation unter Laborbedingungen bereits weitestgehend wissenschaftlich untersucht und erforscht ist [3], werfen neuerdings Entwicklungen wie die Verwendung von Standard-Ethernet auf der Feldebene Fragen für eine neuerliche Untersuchung des Themas auf. Im Wesentlichen gibt es zwei vom Prinzip her konträre Ansätze zur Uhrensynchronisation: demokratische und hierarchische (Master/Slave basierende) Verfahren.

### 2.1 IEEE 1588

Seit der Publikation des Precision Time Protocol (PTP, IEEE 1588) im Jahr 2000 gibt es eine stetig wachsende Produktpalette, die dieses Protokoll (wie z.B. Profinet und Ethernet/IP) unterstützt und verwendet [4, 5]. Das Kon-

zept hinter IEEE 1588 ist so einfach wie effizient: Nach dem Einschalten wird die Masterselektion durchgeführt. Dabei wird jener Knoten gewählt, der als Zeitreferenz am günstigsten erscheint. Die Uhrensynchronisation wird dann in zwei — nach IEEE 1588 ineinander verschränkten — Schritten durchgeführt. Zunächst wird das Delay zwischen Master und Slave mittels sogenannten *delay-request* und *delay-response* Nachrichten gemessen. Dieser Schritt wird benötigt um die Slaves das Delay bei den darauf folgenden Synchronisations-Nachrichten berechnen zu lassen. Bei diesen auch *sync-message* genannten Nachrichten wird vom Master eine Nachricht gesendet, die von einer sogenannten *follow-up message* gefolgt wird. In dieser *follow-up* Nachricht findet sich dann der eigentliche Zeitpunkt des Sendens der ersten Nachricht. Dadurch kann mit der Delay-Information die aktuelle Uhrzeit berechnet werden.

## 2.2 Demokratische Verfahren

Abgesehen von den streng Master/Slave orientierten Uhrensynchronisationsmechanismen wurden auch die demokratischen Algorithmen bereits eingehend untersucht [6, 7, 8]. Diese Klasse von Algorithmen verwendet die Zeit aller Knoten als Referenzzeit und kombiniert diese Menge an Uhrzeiten zu einer sogenannten Ensemble-Zeit. Ein besonders einfacher Ansatz wäre es demnach, alle Samples mit Mittelwertbildung zu kombinieren. Diese Lösung hat den offensichtlichen Nachteil, dass man byzantinische Knoten (das sind Knoten, die unsinnige, falsche Zeiten senden), nicht erkennen kann und diese byzantinische Zeit somit in die gefundene Uhrzeit eingeht. Dasselbe gilt, falls Uhren irrtümlich falsch eingestellt wurden. Weitaus verbesserte Algorithmenklassen, die auf der sogenannten Marzullo-Funktion basieren, können sicherstellen, dass aus  $2F + 1$  maximal  $F$  Knoten falsche Uhrenwerte liefern können, ohne die Ensemble-Zeit zu stören [9]. Ähnliche Ansätze findet man unter [10]. Als herausstechende Änderung der Algorithmen der Marzullo-Klasse ist zu finden, dass diese nicht nur die aktuelle Uhrzeit mit der lokalen Uhr mitführen, sondern auch ein sogenanntes Konvergenzintervall, bei dem obere und untere Schranke der lokalen Uhrzeit angegeben werden. Dadurch kann die Genauigkeit exakt auf die Knoten abgestimmt werden. Beispielsweise bekommen Knoten, die direkt an einem GPS hängen ein geringeres Konvergenzintervall und gehen somit implizit mit mehr Gewicht in die Berechnung der Konvergenzzeit ein.

## 3 Problemdefinition für Ethernet

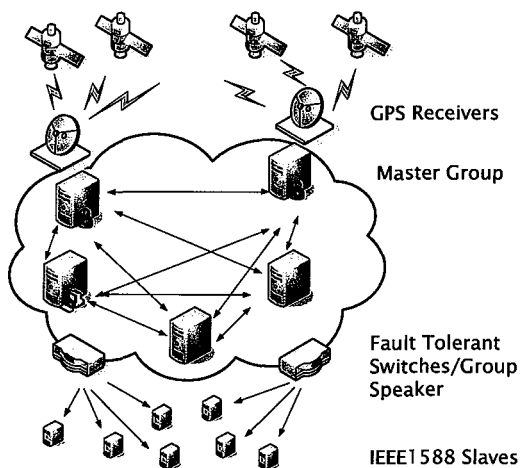
Der wohl offensichtlichste *Single Point of Failure* ist bei IEEE 1588 das Master/Slave Prinzip. Falls ein Master ausfällt (was sich beispielsweise dadurch äußert, dass auf eine *delay-request*-Nachricht keine Antwort kommt oder das Senden von Synchronisations-Paketen ausbleibt), wird ein neuer Master gewählt. Diese sogenannte Best-Master-Clock (BMC) Selektion kann nach dem PTP Stan-

dard erst nach 10 Synchronisationsrunden ohne *sync-message* durchgeführt werden. Während dieser Zeit laufen alle Knoten im System frei, was gleichbedeutend damit ist, dass die Genauigkeit des Systems durch die Präzision der lokalen Oszillatoren bestimmt wird. Letztere wird durch externe Einflüsse wie die Umgebungstemperatur, die Alterung und die elektronische Last am Oszillator bestimmt. Da in IEEE 1588 ein Synchronisationsintervall im Bereich von  $2^0$  bis  $2^6$  Sekunden liegt, kann dadurch die Genauigkeit im Fehlerfall im Bereich von  $10 \times 100 \text{ ppm} \times 2^0 \dots 2^6 \text{ s} = 1..64 \text{ ms}$  liegen. Betrachtet man die technisch mögliche Genauigkeit von *state of the art* Systemen die im Bereich von 10-100 ns liegt, so ist dieser systematische Fehler relativ groß.

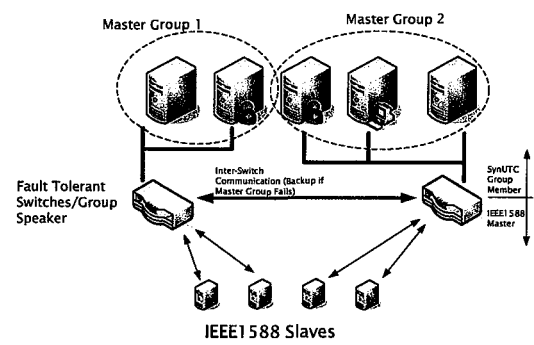
Ein zweites wesentliches Problem von Ethernet ist die Tatsache, dass der Switch einen weiteren *Single Point of Failure* darstellt. Aus diesem Grund müssen Lösungen, die in irgendeiner Form Fehlertoleranz bieten, jedenfalls auch Redundanzkonzepte für den Switch mit sich bringen. Dieser Ansatz und die Tatsache, dass demokratische Verfahren zur Uhrensynchronisation wohl bekannt sind, verleitet einerseits zum Anwenden des Prinzips der demokratischen Uhrensynchronisation auf allen Knoten, andererseits muss jedoch beachtet werden, dass damit auch der Kommunikationsaufwand steigt. Die Konsequenz für die Applikation Uhrensynchronisation ist demnach, dass demokratische Verfahren nur auf eine kleine Anzahl an Knoten angewendet werden kann, bei dem man sich den erhöhten Rechen- und Kommunikationsaufwand leisten kann. Andererseits müssen demnach Knoten, die hierarchisch weiter unten angeordnet sind, mit einer anderen Methode synchronisiert werden, was in diesem Fall dann wieder mit Master/Slave basierenden Systemen passieren kann.

## 4 Lösungskonzept

Die Erwägungen des letzten Abschnitts führen zu dem Ansatz, hybride Strukturen mit gemischt demokratisch und Master/Slave Algorithmen einzusetzen. Diese hierarchisch organisierten *Synchronization-Subnets* (SSNs) wurden bereits von [1] vorgeschlagen. Die Referenzzeit kann in diesem Fall beispielsweise von GPS Satelliten bezogen werden, die ihre Zeit auf lokale Knoten innerhalb des SSNs, die sogenannte Mastergroup, weitergeben. In diesem Zusammenhang muss jedoch auch bemerkt werden, dass es natürlich wenig sinnvoll ist, alle Knoten in der Mastergroup mit GPS-Empfängern auszustatten. Die Gründe dafür sind von finanzieller und praktischer Natur: Einerseits liegen die Kosten für hochpräzise GPS-Empfänger zur Uhrensynchronisation bei durchschnittlich 2000 EUR pro Stück, andererseits ist im Allgemeinen nicht davon auszugehen, dass jeder Knoten über eine GPS-Antenne freien Blick zum Himmel hat. Wesentlich ist allerdings in jedem Fall, dass die Knoten, die an die Mastergroup angebunden sind, mit Oszillatoren mit gesteigerter Genauigkeit verfügen. Solche Oszillatoren sind



**Abbildung 1. Konzept der Master Group mit angebundenen IEEE 1588 Slaves**



**Abbildung 2. Redundanzprinzip bei den Switches um den Ausfall eines einzelnen Switches abzudecken**

je nach Anwendungsfall beispielsweise mit Temperaturkompensation (MCXO) oder Temperatursteuerung (TCXO) ausgestattet.

Abbildung 1 zeigt das prinzipielle Konzept. Der Vorteil dieser Variante ist, dass einzelne Mitglieder der Mastergroup ausfallen können und diese dabei die Gesamtgenauigkeit des Systems nur minimal verschlechtern.

Nichtsdestotrotz stellt das Weitergeben der Ensemblezeit (die Zeit auf die sich die Mastergroup geeinigt hat) an die unterlagerten Slaves ein neuerliches Problem dar. Naheliegender ist, dass ein Netzwerkelement quasi als Sprecher der Mastergroup fungiert. Dabei wurde in der vorliegenden Implementierung der Switch als Sprecher der Mastergroup gewählt, da dieser den geringsten (auf das gesamte Delay bezogenen) Jitter zur Referenzzeit hat. Das setzt natürlich ein geeignetes Design des Switches voraus.

#### 4.1 Switch Design

Beim Design des Switches wurde besonders darauf geachtet, dass prinzipielle Mechanismen der Fehlertoleranz eingebaut werden können. Ein besonders häufig auftretender Fehlerfall ist das Ausfallen der Stromversorgung. Daher werden in der Praxis gerne redundante Stromversorgungen implementiert. Aus Gründen der Einfachheit wurde dies beim Prototypen nicht implementiert, jedoch wurde im vorliegenden embedded System prinzipiell darauf geachtet, dass der Switch mit zwei von einander unabhängigen Netzteilen gespeist werden kann. Eine andere Fehlerquelle sind die elektronischen Komponenten des Switches selbst. Diese kann durch einfache Verdopplung der Switch-Elemente vorgenommen werden, wobei diesem Fall entsprechend auch redundante Leitungen vorgesehen werden müssen. Die Auswertung welches Element gerade als aktiv geschaltet wird, findet dann in einem Software Layer statt, der nicht Gegenstand dieses Papers ist und daher nicht weiter behandelt wird. Abbildung 2 zeigt das entsprechende Prinzip.

## 5 Chipdesign

Für das Erreichen hoher Synchronisationsgenauigkeit über Ethernet ist es notwendig, den Einfluss möglichst vieler Netzwerkschichten, die Jitter produzieren können, auszuschließen. Der gewählte Ansatz basiert auf dem Detektieren der Paketübermittlung von Synchronisations- und Delay-Measurement-Paketen auf dem MII (Media Independent Interface). Dadurch ist der Einfluss von höheren (meist in Software) implementierten Protokollen auf die Genauigkeit der Zeitstempel ausgeschlossen. Nach wie vor bleibt der nichtdeterministische Fehler des Netzwerkes selbst und die Eigenschaften der Bitübertragungsschicht bestehen. Diese Einflüsse müssen im Rahmen von statistischen Algorithmen ausgeglichen werden, da alternativ nur mit einem Eingriff in den internen Aufbau des PHYs (physical layer Chip) höhere Genauigkeiten ermöglicht werden.

Da ein handelsüblicher Ethernet-Switch durch unterschiedliche Paketweiterleitungszeiten Jitter in der Größenordnung von einigen  $\mu\text{s}$  erzeugt, ist dieser ein zentrales Element, wenn Genauigkeiten im Bereich von ns erreicht werden sollen. Das in Abbildung 3 gezeigte Konzept sieht ein zentrales FPGA im MII Pfad, aller an den Switch-Prozessor geführten Ethernet Schnittstellen, vor. Die Herausforderung besteht nun darin, die übertragenen Pakete beim Eintreffen zeitlich zu markieren und beim Versand über einen anderen Port die zeitliche Differenz als Korrekturwert in das Paket einzutragen. Bei IEEE 1588 wird dies durch Korrektur (zeitliche Verschiebung nach hinten) des Sendezeitstempels erreicht. In der zurzeit in Standardisierung befindlichen Version 2 von IEEE 1588 ist geplant, mittels eigener Datenstrukturen die Summe der durch die Switches und physikalischen Links entstandenen Delays zu übertragen. Letzteres hat in den vergangenen Monaten zur Diskussion geführt, in allen weiteren Versionen von IEEE 1588 einen neuen Pakettyp, das sogenannte *A-Delay* Paket einzuführen. Dieses wird allerdings aufgrund der sehr instabilen Version des Standardentwurfs derzeit noch nicht unterstützt.

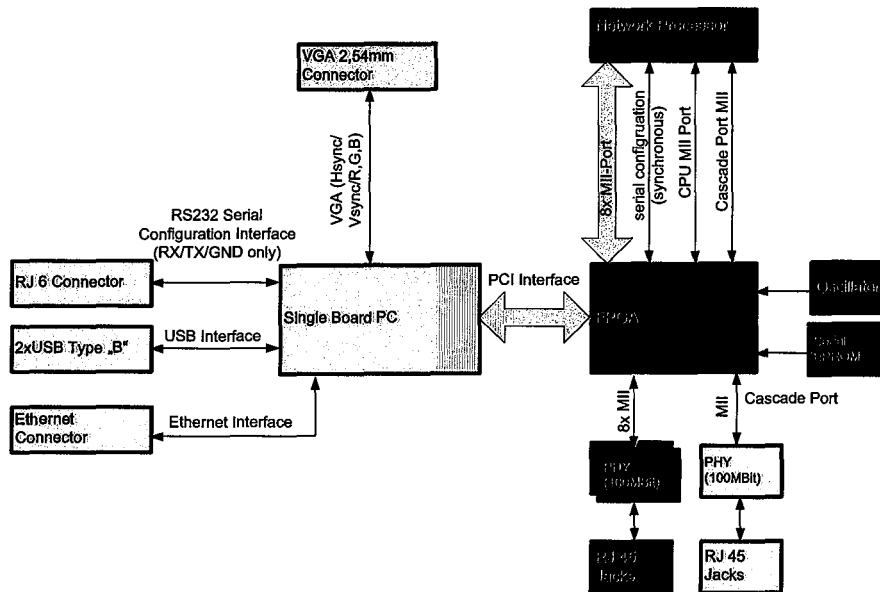


Abbildung 3. Konzept des IEEE 1588 kompatiblen Switches

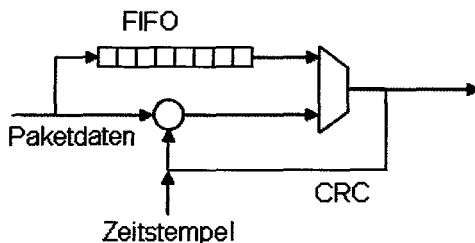


Abbildung 4. Prinzip zum Einfügen von Zeitstempeln in den MII Datenstrom

Das FPGA enthält neben einer auf einem 104-bit Ad-dierer basierenden Uhrenzelle [11, 12], die die Zeitbasis für alle Berechnungen darstellt, eine Zeitstemp-einheit für alle Ethernet Ports. Diese Einheiten speichern einen 8 Byte Zeitstempel beim Eintreffen eines Paketes auf dem zugeordneten Port, der beim Erkennen einer *PTP-sync* oder *delay-request* Nachricht mit Hilfe eines parallel geschalteten FIFOs (siehe Abbildung 4) nach dem eigent-lichen PTP Kopf eingefügt wird. Da PTP Nachrichten laut Standard [13] nie die maximale Ethernet Paketlänge erreichen, stellt das Einfügen eine erlaubte Modifikation im Rahmen des Ethernet Standards dar, sofern auch der CRC des Paketes neu berechnet wird. Dadurch ist das neu erstellte Paket für den Switch Prozessor transparent. Aller-dings wird der minimale Abstand zwischen zwei Pake-ten, der bei Ethernet 96 Bitzeiten, beträgt durch die Mo-difikation um 64 Bit verkürzt. Diese Änderung ist nur für den Switchprozessor sichtbar, und daher im Bereich des Einflusses des Designs. Außerdem tritt dieser Fall nur bei sogenannten *sync*-Bursts gehäuft auf, bei denen mehrere *sync* Nachrichten direkt hintereinander gesendet werden.

Beim Senden eines Paketes an den Zielpport wird er-

neut ein Zeitstempel gespeichert. Mit der oben erwähnten FIFO Struktur wird der im Paket eingefügte Zeitstempel extrahiert und zur Berechnung der Paketverzögerung herangezogen. Das Ergebnis wird dann je nach Protokollver-sion direkt als Verzögerung oder als korrigierter Zeitstempel im Paket gespeichert. Für die angeschlossenen PTP Endknoten bzw. weiteren Switches ist damit der betrach-tete Switch bezüglich Verzögerungen der PTP *sync* bzw. *delay-request* Nachrichten transparent. Übrig bleiben da-mit nur mehr jene Verzögerungen, die durch die Laufzeit auf den Netzwerkverbindungen selbst auftreten. Die-se können jedoch durch den im Protokoll vorgesehenen *delay-request / -response* Mechanismus exakt (unter der Annahme von symmetrischen Verzögerungen) ausgemessen werden.

Da der Switch selber nur mit Zeitdifferenzen arbei-tet ist eine absolute Synchronisation der internen Zeit-basis (außer für Analysezwecke) nicht notwendig. Trotz-dem ist es für Genauigkeiten unterhalb von 10 ns bei han-delsüblichen Quarzen mit einer Genauigkeit von 100 ppm erforderlich, die Schrittweite der Uhrenzellen an die un-terschiedlichen Quarzgeschwindigkeiten anzupassen, um eine genaue Zeitreferenz zu erhalten. Dies kann durch einen, auf dem optional implementierbaren Prozessor Mo-dul ausgeführten, PTP Stack erfolgen, der dann auch zu-gleich als *Boundary Clock*, Mitglied einer *Master Group* oder als eigener Master fungieren kann.

Ist eine genauere Referenz, in Form eines Kurzzeit (cir-ca 1 ms) stabilen Quarzes bereits vorhanden, dann ist kei-ne Anpassung der Laufgeschwindigkeit notwendig, da die üblichen Paketverzögerungen in einem Switch im Bereich von 100  $\mu$ s liegen.

Durch das Einfügen des zentralen FPGAs in sämtliche MII Datenströme ist es auch möglich, die Funktion der einzelnen Schnittstellen zu überwachen. Das FPGA kann nun so programmiert werden, dass immer zwei MII Pfa-

de zu einem Paar für eine Verbindung zusammengefasst werden. Im Falle des Ausfalles einer Verbindung kann dann der redundante Pfad zum Switchprozessor durchgeschaltet werden. Diese Variante hat gegenüber der im Switch Prozessor verwendeten *link failover* Methode, die im Falle des Ausfalles einer Verbindung zu einem anderen Switch oder Knoten innerhalb von 50 ms auf eine andere Leitung umschalten kann, den Vorteil, dass auch das PTP Protokoll über die Änderung der Leitungsverzögerung informiert werden kann. Damit kann das Redundanzkonzept auch auf Anwendungsebene ohne die sonst notwendigen Detektionsverzögerungen (PTP benötigt 10 Intervalle um einen neue Masterselektion einzuleiten) implementiert werden.

## 6 Conclusio

Der Einzug von Ethernet in die Welt der real-time Netzwerke wird durch grundlegende Technologien wie die Uhrensynchronisation ermöglicht. Obwohl gleichwertige Methoden zur Einführung von real-time in Ethernet wie das Traffic-Shaping ebenso gut erforscht sind scheint sich, wie in der Einleitung dargelegt, das Verfahren von TDMA mit Uhrensynchronisation für industrielle Netzwerke durchzusetzen. Besonderes Augenmerk ist bei der Uhrensynchronisation in Ethernet dem Switch zu widmen. Er ist einerseits *Single Point of Failure*, und kann andererseits den Gruppensprecher in einem Hybrid demokratisch/Master/Slave konzipierten Uhrensynchronisiertem Netzwerk übernehmen. Besonderes Augenmerk ist in diesem Fall auch dem Design des der Switch Fabric vorgeschalteten FPGAs zu widmen, da dieses einerseits exakte Zeitstempel zur Ermittlung der Durchlaufzeit ziehen muss und andererseits als Sprecher einer angeschlossenen Master-Group fungieren soll. Dieses Paper zeigt ein solches Chip-Design, das diese Funktion durch Einfügen von Daten in den MII-Datenstrom ermöglicht.

## 7 Danksagung

Die Autoren möchten sich an dieser Stelle bei Hrn. Felix Ring bedanken, der durch viele wertvolle Hinweise bei der Erstellung dieses Papers geholfen hat.

## Literatur

- [1] D. L. Mills. Internet time synchronization: the network time protocol. *IEEE Transactions on Communications*, pages 1482–1493, 1991.
- [2] IEC. *Precision clock synchronization protocol for networked measurement and control systems*. IEC, 2004-09.
- [3] Emmanuelle Anceaume and Isabelle Puaut. A Taxonomy of Clock Synchronization Algorithms. Publication Interne 1103, Campus Universitaire de Beaulieu, July 1997.
- [4] Victor Schiffer. *The CIP Family of Fieldbus Protocols*, chapter 14, pages 14.1 – 14.65. The Industrial Communication Technology Handbook. Taylor&Francis, 2005.
- [5] Manfred Popp, Joachim Feld, and Ralph Bsgen. *Principles and Features of PROFInet*, chapter 11, pages 11–1. The Industrial Communication Technology Handbook. Taylor&Francis, 2005.
- [6] Hermann Kopetz. *Design principles for Distributed Embedded Applications*. Kluwer Academic Publishers, 1997.
- [7] Ulrich Schmid. Synchronized Universal Time Coordinated for Distributed Real-Time Systems. *Control Engineering Practice*, 3(6):877–884, 1995. (Reprint from Proceedings 19th IFAC/IFIP Workshop on Real-Time Programming (WRTP'94), Lake Reichenau/Germany, 1994, p. 101–107.).
- [8] Christof Fetzter and Flaviu Cristian. An Optimal Internal Clock Synchronization Algorithm. In *Proceedings 10th Annual IEEE Conference on Computer Assurance*, Gaithersburg, MD, June 1995.
- [9] Ulrich Schmid and Klaus Schossmaier. Interval-based Clock Synchronization. *J. Real-Time Systems*, 12(2):173–228, March 1997.
- [10] Ulrich Schmid, Martin Horauer, and Nikolaus Kerö. How to Distribute GPS-Time over COTS-based LANs. In *Proceedings of the 31th IEEE Precise Time and Time Interval Systems and Application Meeting (PTTI'99)*, Dana Point, California, December 1999.
- [11] Thomas Bigler, Franz Winkler, and Georg Gaderer. Entwurf eines Embedded Systems und IP-Cores zur Uhrensynchronisation. In *Proceedings of the 2005 Austrochip Conference*, pages 119–126, September 2005.
- [12] Georg Gaderer, Patrick Loschmidt, Thilo Sauter, and Gerd Bumiller. Investigations on Fault Tolerant Clock Synchronization within a Powerline Communication Structure. In *Proceedings of the 2006 IEEE International Symposium on Powerline Communication ISPLC*, March 2006.
- [13] IEEE Instrumentation and Measurement Society TC9. *1588 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*. IEEE, 2000.