Markus Winter

# Unterstützung und Organisation von Quality-of-Service Techniken in Kommunikationsnetzwerken auf einem Chip (Network-on-Chip)

Beiträge aus der Informationstechnik

Mobile Nachrichtenübertragung
Nr. 55

**Markus Winter**

# Unterstützung und Organisation von Quality-of-Service Techniken in Kommunikationsnetzwerken auf einem Chip (Network-on-Chip)

VOGT

Dresden 2012

Bibliografische Information der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.ddb.de abrufbar.

Bibliographic Information published by the Deutsche Bibliothek
The Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibiograpic data is available in the internet at http://dnb.ddb.de.

Zugl.: Dresden, Techn. Univ., Diss., 2011

Die vorliegende Arbeit stimmt mit dem Original der Dissertation „Unterstützung und Organisation von Quality-of-Service Techniken in Kommunikationsnetzwerken auf einem Chip (Network-on-Chip) “ von Markus Winter überein.



Gesetzt vom Autor

ISBN 978-3-938860-46-5

Jörg Vogt Verlag
Niederwaldstr. 36
01277 Dresden
Germany

Phone: +49-(0)351-31403921
Telefax: +49-(0)351-31403918
e-mail: info@vogtverlag.de
Internet : www.vogtverlag.de

Technische Universität Dresden

# Unterstützung und Organisation von Quality-of-Service Techniken in Kommunikationsnetzwerken auf einem Chip (Network-on-Chip)

**Markus Winter**

von der
Fakultät Elektrotechnik und Informationstechnik
der Technischen Universität Dresden

zur Erlangung des akademischen Grades eines

**Doktoringenieurs**
(Dr.-Ing.)

genehmigte Dissertation

| | |
|---|---|
| Vorsitzender: | Prof. Dr.-Ing. habil. Rüdiger Hoffmann |
| Gutachter: | Prof. Dr.-Ing. Gerhard P. Fettweis |
| | Prof. Dr. Tech. Axel Jantsch |
| Tag der Einreichung: | 02.09.2011 |
| Tag der Verteidigung: | 22.12.2011 |

# Abstract

The performance of integrated circuits must be constantly increased in order to support advanced algorithms of information processing, e.g. latest multimedia and mobile communications standards, running simultaneously. Multi-Processor Systems-on-Chip (MPSoC) can fulfill these requirements by distributing the jobs onto parallel processing units where they can be executed concurrently. The efficient communication of such processing units as well as memories and I/O-Interfaces is a challenge which shall be solved by packet-switched Networks-on-Chip (NoC). They break up long connecting wires by inserting routers and transmitting the data as packets across the NoC. They reduce the wiring overhead, increase clock frequency, bandwidth and scalability, enable parallel data transfers and utilize the communications resources much more efficiently by avoiding circuit-switching.

But packet-switching only cannot provide any guarantees concerning throughput, latency and jitter required by numerous applications. These applications have realtime requirements which must be supported by the scheduling of jobs onto processors and by the data transfers via Quality-of-Service (QoS), too. The combination of packet- and circuit-switching is able to realize QoS by reserving special routes on top of the packet-switched NoC. These routes carry the guaranteed traffic including bandwidth, latency and jitter guarantees.

Today, the main challenge and, thus, the purpose of this thesis is the search and allocation of QoS resources in the NoC suitable for an application executed on the MPSoC. Based on a dynamic run-time task scheduling mechanism, we present techniques how to find and allocate QoS resources within a few dozen clock cycles and less. We analyze centralized methods which introduce a special unit in the MPSoC owning complete control about the QoS resources of the NoC. We compare them to distributed variants which utilize the NoC itself for the allocation process. Our study reveals a cost-benefit advantage of the centralized methods. They require less area and demonstrate better performance in most traffic scenarios. Our results enable the development of MPSoCs which support soft realtime constraints of applications not only at scheduling but at data transfers across the NoC, too.

# Zusammenfassung

Fortgeschrittene Algorithmen der Informationsverarbeitung, wie z.B. neueste Multimedia- und Mobilfunkstandards, sowie die Notwendigkeit, mehrere Anwendungen gleichzeitig zu unterstützen, erfordern ständige Steigerungen der Leistungsfähigkeit von Chips. Multiprozessor Systeme auf einem Chip (MPSoC) verteilen die Aufgaben auf parallele Verarbeitungseinheiten, auf denen sie gleichzeitig nebeneinander abgearbeitet werden und somit die Applikation beschleunigen. Die effiziente Vernetzung dieser Verarbeitungseinheiten sowie von Speichern und I/O-Schnittstellen ist dabei ein große Herausforderung, die von paketvermittelten Netzwerken auf einem Chip (NoC) bewältigt werden sollen. Sie zerteilen lange Verbindungsdrähte, indem sie Router einfügen und die Daten als Pakete in das NoC speisen. Dies verringert den Verdrahtungsaufwand, erhöht die Taktfrequenz, Bandbreite sowie Skalierbarkeit und erlaubt parallele Datentransfers.

Reine paketvermittelte Netze können aber keine Garantien bezüglich Latenz, Durchsatz und Jitter der Datentransfers geben, die aber für zahlreiche Anwendungen notwendig sind. Diese Applikationen haben Echtzeitanforderungen, die beim Scheduling der Aufgaben auf die Prozessoren wie auch bei den Datentransfers in Form von Quality-of-Service (QoS) unterstützt werden müssen. Daher ist eine Kombination aus paket- und leitungsvermittelten Netzen notwendig. Auf der Basis von Paketvermittlung müssen im NoC spezielle Routen reserviert werden können, über die der Verkehr mit Bandbreite-, Latenz- und Jitter-Garantien transportiert werden kann.

Die wesentliche Herausforderung ist heute, diese QoS Ressourcen des NoCs in Abhängigkeit von den auf dem Chip ausgeführten Applikationen zu finden und zu reservieren. Dieser Aufgabe widmet sich diese Arbeit. Ausgehend von einem dynamischen, zur Laufzeit ausgeführten und auf Tasks basierendem Scheduling Verfahren, stellen wir Methoden vor, wie auch die QoS Ressourcen zur Laufzeit in nur ein paar Dutzend Takten und weniger gefunden und allokiert werden können. Wir analysieren verschiedene zentralisierte Allokationsmechanismen, bei denen eine Einheit im MPSoC die Kontrolle über die QoS-Ressourcen des NoCs hat, und stellen sie verteilten Varianten gegenüber, bei denen das NoC selbst für diesen Allokationsprozess genutzt wird. Dabei zeigte sich, dass die zentralisierten Verfahren in den meisten Verkehrsszenarios einen Kosten-Nutzen-Vorteil gegenüber den verteilten Verfahren aufweisen. Sie haben einen niedrigeren Flächenbedarf und offenbarten in den meisten Fällen auch eine bessere Leistungsfähigkeit. Mit diesen Ergebnissen können MPSoCs entwickelt werden, die weiche Echtzeitfähigkeit für Applikationen nicht nur beim Scheduling sondern auch bei den Datentransfers im MPSoC unterstützen.

# Inhaltsverzeichnis

# Abbildungsverzeichnis

# Abkürzungsverzeichnis

| | |
|---|---|
| Abb. | Abbildung |
| Ack | Acknowledge |
| AMBA | Advanced Microcontroller Bus Architecture |
| ALAP | As Late As Possible |
| ASIP | Application Specific Integrated Processor |
| BE | Best Effort |
| bzgl. | bezüglich |
| ca. | circa |
| d.h. | das heißt |
| DMA | Direct Memory Access (Controller) |
| DSP | Digitaler Signalprozessor |
| FIFO | First In First Out (Speicher) |
| Flit | Flow Control Digit (kleinste Übertragungseinheit im NoC) |
| FEC | Forward Error Correction (Modul) |
| GALS | Globally Asynchronous, Locally Synchronous |
| GPS | Global Positioning System |
| GS | Garantierter (Guaranteed) Service |
| GSM | Global System for Mobile Communication |
| GT | Garantierter (Guaranteed) Traffic |
| HAGAR | Hardware Graph Array |
| IC | Integrierter Schaltkreis (Integrated Circuit) |
| I/O | Input/Output |
| kNAND | kilo NAND Gatter |

| | |
|---|---|
| LAN | Local Area Network |
| LTE | Long Term Evolution |
| LVDS | Low Voltage Differential Signaling |
| MPSoC | Multi-Processor System-on-Chip |
| NAck | Not Acknowledge |
| NoC | Network-on-Chip |
| NoCIF | Network-on-Chip Interface |
| OSI | Open Systems Interconnection (Modell) |
| PLL | Phase Locked Loop |
| QoS | Quality-of-Service |
| RTL | Register Transfer Level |
| SIMD | Single Instruction Multiple Data |
| SoC | System-on-Chip |
| TDM | Time Division Multiplex |
| TSMC | Taiwan Semiconductor Manufacturing Company |
| TUD, TU Dresden | Technische Universität Dresden |
| UMC | United Microelectronics Corporation |
| vgl. | vergleiche |
| VLIW | Very Long Instruction Word |
| WLAN | Wireless Local Area Network |
| XML | eXtensible Markup Language |
| z.B. | zum Beispiel |

# Symbolverzeichnis

| | |
|---|---|
| $\lambda$ | Verkehrsangebot/-wert |
| $K$ | Knotenanzahl im Graph |
| $N_M, N_R$ | Anzahl der Module, Anzahl der Router im MPSoC |
| $N_I$ | Anzahl der Zwischenankunftszeiten/Intervalle zwischen zwei zu sendenden Flits |
| B | Bisektionsbreite |
| D, $D_{ges}$ | Gesamtbandbreite/Durchsatz/Datenrate im NoC |
| $D_{Link}$, $D_M$, $D_R$ | Datenrate je Link, Modul, Router |
| $T_a$, $t_a$ | Zwischenankunftszeit des Forderungsstroms |
| $f_{E,T_a}$,$F_{E,T_a}$ | Dichtefunktion und Verteilungsfunktion der exponentialverteilten Zwischenankunftszeit |
| $f_{G,T_a}$,$F_{G,T_a}$ | Dichtefunktion und Verteilungsfunktion der geometrisch verteilten Zwischenankunftszeit |
| $E_G[T_a]$, $V_G[T_a]$ | Erwartungswert und Varianz der geometrisch verteilten Zwischenankunftszeit |
| $K_{G,L}$,$K_{G,U}$ | Unter und obere Schranke des 95%-Konfidenzintervalls des Erwartungswerts der geometrisch verteilten Zwischenankunftszeit |
| $S$ | Anzahl der Slots bei zeitgeteilten GS Kanälen |

# 1. Einleitung

## 1.1. Multi-Prozessor Systeme auf einem Chip

**Multi-Prozessor System-on-Chip (MPSoC)** - das sind die Worte, die in den letzten 10 Jahren die Forschergemeinde im Bereich der integrierten Schaltungen (IC) und Chipentwicklung bewegten und sie auch in den nächsten Jahren oder gar Jahrzehnten bewegen werden. Zwar wird zu Multi-Prozessor Systemen bereits seit den 70er Jahren des 20. Jahrhunderts im Rahmen der Supercomputer geforscht. Doch die Integration dieser Systeme auf einem einzigen Chip bringt spezielle Anforderungen und Notwendigkeiten mit sich. Ein komplexes Spannungsfeld aus Konsumentenwünschen, technologischen Möglichkeiten, Anforderungen an den Energieverbrauch und Komplexität der Aufgaben und Algorithmen, die ein Chip heute zu verarbeiten hat, führen zu dieser neuen Richtung der Forschung und Entwicklung.

In erster Linie zieht das stete Voranschreiten der Fertigungstechnologie alle anderen Entwicklungen nach sich. Die Strukturbreiten werden immer kleiner, wodurch immer mehr Transistoren auf einem Quadratmillimeter Chipfläche Platz finden. Dadurch kann - zumindest theoretisch - immer mehr Funktionalität durch einen einzelnen Chip realisiert werden. Diese Funktionalitätssteigerung wird wiederum vom Konsumenten nicht nur abgefragt sondern geradezu verlangt. So muss ein modernes Mobiltelefon nicht mehr nur einen verhältnismäßig einfachen Standard wie GSM in den 90er Jahren beherrschen. Vielmehr muss es Quad-Band fähig sein, Mobilfunkstandards der zweiten, dritten und bald auch vierten Generation beinhalten, aber auch Zusatzfunktionen wie GPS zur Navigation und Multimedia-Kodierung und -Dekodierung für die integrierte Kamera und zum Abspielen von aus dem Internet geladenen Filmen unterstützen. Der Wunsch von Industrie und Verbrauchern nach einer immer stärkeren Verbindung von Mobilität, Internetzugriff immer und überall, multimedialen Inhalten und der Leistungsfähigkeit von PCs, wie beispielsweise in SmartPhones, ist eine der treibenden Kräfte hinter der fortschreitenden Entwicklung und Leistungssteigerung der Chips.

Bis zum Ende der 90er Jahre konnte die Erhöhung der Taktfrequenz im Zusammenspiel mit kleineren Strukturbreiten die Integration komplexerer Algorithmen auf dem Chip realisieren. Doch im ersten Jahrzehnt des neuen Jahrhunderts wurde deutlich, dass ein 'weiter so' unmöglich ist. Ging man um die Jahrtausendwende noch davon aus, dass Prozessoren 2010 eine Taktfrequenz von etwa 10 GHz haben würden [BM02], so zeigt sich heute, dass die Frequenzen von Allzweck-Prozessoren bei 3-4 GHz stagnieren und bei digitalen Signalprozessoren (DSP) im Allgemeinen weit unter einem GHz bleiben. Der enorme Energieverbrauch hoch getakteter Prozessoren

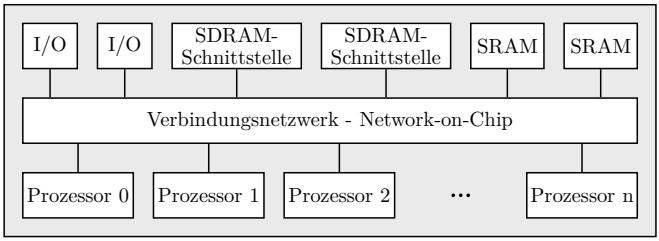


Abbildung 1.1.: Grundstruktur eines typischen MPSoCs: mehrere Prozessoren, I/O-Interfaces, Speicher und Netzwerk.

## 1.2. Networks-on-Chip als Lösung zum MPSoC Verbindungsproblem

Die Herausforderung, Einheiten auf einem Chip effizient zu vernetzen, stellt sich nicht erst seit dem Siegeszug der MPSoCs. Single-Prozessor System-on-Chips, die einen Prozessor, Speicher und Peripherie auf einem Chip integrieren, nutzten dazu Busse wie beispielsweise IBMs CoreConnect [IBM99] und ARMs Advanced Microcontroller Bus Architecture (AMBA) [ARM99]. Auf Bussen kann zu einem Taktzeitpunkt aber nur ein einziges Datum ausgetauscht werden. Voneinander unabhängige Datentransfers können nicht parallel durchgeführt werden. Hinzu kommen Probleme, wenn immer mehr Module an einen Bus angeschlossen werden sollen. Jedes Modul verlängert nicht nur die Drähte und damit deren Kapazität, sondern addiert durch die Anschlüsse eine zusätzlich Kapazität hinzu. Diese erhöhen die Umladezeiten und auch den Stromverbrauch bzw. erfordern den Einsatz stärkerer Treiber, was ebenfalls zur signifikanten Steigerung des Energieverbrauchs führt. Eine Weiterentwicklung stellen daher Layered-Busse [ARM04] und Crossbar-Switches wie [WF06] oder ARMs AXI-Protocol dar. Mit ihnen können mehrere Datenströme gleichzeitig zwischen Modulen ausgetauscht werden, solange es keine Konflikte gibt, d.h. solange nicht das gleiche Modul oder Subnetz für zwei verschiedene Datenströme benötigt wird. Außerdem verringert sich auf diese Weise die Kapazität der einzelnen Drähte der Busse.

Aber auch Layered-Busse und Crossbar Switches werden aufgrund des globalen Verdrahtungsaufwandes nicht dazu in der Lage sein, das Skalierbarkeitsproblem von MPSoCs zu lösen, d.h. MPSoCs mit 100 Kernen oder mehr effizient intern zu vernetzen. Um die Jahrtausendwende stellten sich Forscher die Frage, wie lange die Busse und die damit verbundenen, über den ganzen Chip reichenden Drähte aus rein physikalischer Betrachtung noch sinnvoll sind. [HMH01] und [SK00] kamen zu dem Schluss, dass die kurzen Drähte für die Verdrahtung von Transistoren innerhalb eines Moduls mit der Fertigungstechnologie und dem Gate-Delay skalieren würden. Für die globale Verdrahtung zeichneten sie jedoch ein pessimistischeres Bild. Signale,

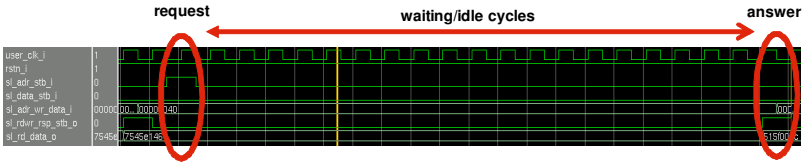


Abbildung 1.2.: Bei Leitungsvermittlung blockierter Kommunikationskanal. Aufgrund der Antwortzeit des angesprochenen Moduls (hier ein SDRAM-Controller auf dem *Tomahawk* MPSoC) und der verwendeten Leitungsvermittlung sind die ca. 20 Takte zwischen Anfrage und Antwort nicht für andere Datentransfers nutzbar.

die auf langen, globalen Drähten über den gesamten Chip transportiert werden müssen, würden dafür mehrere Takte benötigen. Sie schlagen daher eine Aufsplittung langer Drähte bzw. die Verwendung von breiten Drähten auf hoher Metall-Ebene vor, um das RC-Delay zu verringern. [The00] geht noch einen Schritt weiter und favorisiert einen Paradigmenwechsel bei der MPSoC Vernetzung. Demnach soll das Kommunikationssystem unterteilt werden in Router, die ein paketvermitteltes Netzwerk auf einem Chip (Network-on-Chip - NoC[2]) realisieren.

Der Wechsel von Leitungsvermittlung, wie sie bei Bussen oder auch Crossbar-Switches zum Einsatz kommt, hin zur Paketvermittlung verbessert auch die Kanalauslastung. Ist bei Leitungsvermittlung ein Kommunikationskanal zwischen zwei Modulen aufgebaut, so ist dieser belegt, selbst wenn die beiden Module keine Daten austauschen. Für andere Module steht das Kommunikationsmedium dann nicht zur Verfügung, was zu einer Verschwendung von Kommunikationsressourcen führt. Diese Erfahrung wurde bereits in den Telefonnetzen gemacht, die inzwischen (insbesondere getrieben durch das Internet) auf Paketvermittlung umgestellt wurden. Gleiche Erfahrungen konnten wir bei der Entwicklung des *Tomahawk*-MPSoCs am Lehrstuhl für Mobile Nachrichtensysteme der TU Dresden [LWB+08, LWB+09] mit dem verwendeten Netzwerk [WF06] machen. In Abb. 1.2 ist ein Beispiel für das Blockieren eines Kommunikationskanals gezeigt. Deutlich zu erkennen ist die Anfrage (der Aufbau des Kanals - *request*) und die Antwort (der Abbau des Kanals - *answer*) aber auch die etwa 20 Takte dazwischen, in denen der Kanal für jegliche andere Kommunikation blockiert ist. In einem paketvermittelten System könnten in diesen Takten weitere Anfragen bzw. Datenströme transferiert werden, insbesondere wenn der Empfänger Pipelining unterstützt. Einen Datentransfer in Anfrage- und Antwortteil aufzuspalten ('Split-Transfer' genannt) und den Kanal zwischen diesen beiden Teilen für andere Transfers freizugeben, verringert zwar das Blockierpro-

[2]Der Begriff 'Network-on-Chip' umfasst prinzipiell alle Arten der Vernetzung von Einheiten auf einem Chip - Busse, Punkt-zu-Punkt-Verbindungen, leitungs- oder paketvermittelte Netze. Wir werden den Begriff im folgenden allerdings nur für paketvermittelte, Router-basierte Netzwerke verwenden.

blem. Es stellt aber nur eine komplexe Hilfskonstruktion dar, die das Problem nicht grundsätzlich löst, da das prinzipielle Busprotokoll wie auch seine Architektur beibehalten wird. Konsequenter ist es daher, den Systemwechsel zur Paketvermittlung zu vollziehen. [BHG+06] vergleichen ein paketvermitteltes NoC mit einer klassischen AMBA Busarchitektur und schlussfolgern, dass ein gut an die Applikation angepasstes NoC mehr nutzbare Bandbreite zur Verfügung stellt und einen nur moderaten Flächenmehrbedarf von 16% aufweist.

[TD01] präsentieren als eine der Ersten ein paketvermitteltes NoC, indem sie den Chip in 'Kacheln' untergliedern und diese in Form eines 2D-Gitternetzes mittels Router verbinden. Obwohl zu der Zeit ähnliche Vorschläge von anderen Wissenschaftlern kamen, gilt [TD01] als eine der initialen Veröffentlichungen zu NoCs. [KJS+02] und [BM02] treiben diese Überlegungen weiter voran, indem sie sich grundlegenden Fragen der Topologie, des Routings, der Vermittlung, Flusskontrolle oder der Zwischenpufferung widmen (Kapitel 2.1). [BM02] schlägt zudem eine Organisation der NoC Funktionalitäten ähnlich dem OSI-7-Schichtenmodell vor. Eine Software-Ebene, auf der Applikationsingenieure arbeiten, eine Architektur- und Kontrollebene, die von NoC-Entwicklern entworfen wird und eine physikalische Schicht, mit der sich besonders Ingenieure im Place&Route und Backend der Chip-Entwicklung befassen. Dieses Prinzip und insbesondere die dafür notwendige Schnittstellenspezifikation zwischen den Ebenen hat sich bislang in der NoC-Gemeinschaft aber kaum durchgesetzt. Ein NoC bleibt immer an einen bestimmten Chip gebunden und Interoperabilität mit anderen Tools oder gar Entwicklergruppen ist von wesentlich geringerer Bedeutung als beim Internet oder Fernmeldenetz.

Aus diesen frühen NoC Forschungen hat sich inzwischen das Start-Up 'Arteris' gegründet, welches die Entwicklung system-spezifischer, auf die Kundenwünsche angepasster NoC-Lösungen anbietet [Art].

## 1.3. Herausforderungen bei der NoC Entwicklung

Trotz der bereits geleisteten fundamentalen Arbeit auf diesem Gebiet und den Erkenntnissen der Multi-Prozessor Systeme im Bereich der Supercomputer stehen noch viele Herausforderungen bei der NoC Entwicklung an. Im Gegensatz zu Telekommunikationsnetzen oder auch den Netzen der Supercomputer wird ein NoC für einen bestimmten Chip entwickelt. Daher kann und sollte das NoC auch auf die spezifischen Notwendigkeiten dieses Chips abgestimmt sein, um ein Maximum an Leistungsfähigkeit und ein Minimum an Flächen- und Energieverbrauch zu realisieren. Idealerweise wird ein Chip für eine bestimmte Applikation entwickelt, so dass die zu erwartenden Kommunikationsmuster beim Entwurf berücksichtigt werden können. Der Trend hin zum Software Defined Radio erschwert diese Vorgehensweise allerdings. Dabei soll ein MPSoC mehrere Standards und Applikationen, die teilweise noch nicht im letzten Detail feststehen, unterstützen. Zur Entwurfszeit des Chips ist es daher schwer, ein optimal abgestimmtes MPSoC und NoC zu entwickeln, da es zugleich auch eine gewisse Programmierbarkeit und Flexibilität mit sich bringen muss. Diese grund-

legende Problematik erkennt auch Marculescu, der in [MOP+09] einen Überblick über unbeantwortete Fragen bei der NoC Entwicklung gibt und 14 aus seiner Sicht notwendige Forschungsschwerpunkte vorstellt. So stellt er beispielsweise die Frage nach sinnvollen Topologien und Routing-Techniken unter Berücksichtigung der zu erwartenden Applikation. Doch um dies zu beurteilen sind geeignete Benchmarks und Bewertungssysteme notwendig, die in der NoC Forschergemeinde noch nicht existieren - ein weiterer offener Punkt, genauso wie Fragen des Energieverbrauchs und der lokalen Wärmeentwicklung auf dem Chip.

Die zur Zeit aber wesentlichste Frage im MPSoC-Bereich ist die des 'Application-Mapping', die [MOP+09] auch als erstes nennt. D.h., die Frage, wie die Aufgaben (Tasks), die für eine Applikation bearbeitet werden müssen, auf die parallelen Bearbeitungseinheiten am effizientesten verteilt werden können. Dabei ist unter Effizienz nicht nur geringe Fläche und niedriger Energieverbrauch zu verstehen, sondern auch der Aufwand, den die Programmierung des Systems für einen Software-Entwickler mit sich bringt. Dieses Application-Mapping ist ein Ressourcenallokationsproblem, das auch Kommunikationsressourcen über das NoC umfasst. Viele Anwendungen, wie Multimedia-Ströme oder Ack-NAck-Protokolle in Mobilfunknetzen, stellen für die Bearbeitung und den Datenaustausch über das NoC weiche Echtzeitanforderungen (soft- oder semi-hard realtime-constraints). Die Applikation benötigt Zusicherungen über die Qualität der notwendigen Ressourcen: steht genügend Rechenkapazität zur Verfügung? Kann das NoC garantieren, dass bestimmte Latenzen und ein maximaler Jitter nicht überschritten, benötigte Bandbreiten nicht unterschritten werden? Dies fasst man als Quality-of-Service (QoS) unter Echtzeitbedingungen zusammen.

Ein rein paketvermitteltes System kann allerdings diesen Quality-of-Service nicht bieten. Bei hoher Auslastung des NoCs sind die Latenzen der Datenübertragung unvorhersehbar hoch, verfügbare Bandbreiten für den Datenaustausch variieren stark, das System ist kaum vorhersagbar. Diese Art des Verkehrs nennt man Best Effort (BE - nach bestem Bemühen). Das Netzwerk versucht zwar, allen Kommunikationswünschen so gut es geht, nachzukommen, kann aber nicht garantieren, dass allen Wünschen auch entsprochen werden kann. Dies führt zu Unsicherheiten und Unvorhersagbarkeiten, die für Echtzeitanforderungen inakzeptabel sind. Um dennoch QoS in NoCs zu ermöglichen, sind besondere Techniken notwendig, die mittels Priorisierung oder vorher reservierten Kanälen Garantien über den Transport von Datenströmen im NoC geben und auch einhalten können. Eine große Herausforderung besteht darin, diese QoS Ressourcen in einem komplexen und hochdynamischen MPSoC für die Applikation zu reservieren, ohne jedoch andere Applikationen unnötig zu behindern. Diese Dissertation widmet sich dieser Problematik.

## 1.4. Beiträge dieser Arbeit

Diese Arbeit befasst sich mit dem Entwurf neuartiger NoC-Konzepte und der Analyse des Verkehrs in diesen NoCs. Zunächst wird die Leistungsfähigkeit von Best Effort

Verkehr für verschiedene Netzwerk- und MPSoC-Größen untersucht. Diese Untersuchungen bestätigen die Ergebnisse anderer Arbeiten auf dem Gebiet und liefern Ergebnisse zu neuartigen, hierarchischen NoC Topologien mit dem Ziel, einen Ausweg aus der mangelnden Skalierbarkeit bestehender, flacher 2D-Gitter-Topologien hinsichtlich der Leistungsfähigkeit zu finden. Ausgehend von den Beobachtungen bei der BE Analyse wird dann ein QoS-Mechanismus in das NoC eingeführt, um Echtzeitdatenströme mit garantierter Qualität über das NoC transportieren zu können.

Der Hauptbeitrag dieser Arbeit liegt in der Entwicklung und Analyse von Verfahren, um die QoS-Ressourcen des NoCs zur Laufzeit aus der laufenden Anwendung heraus zu reservieren. Dazu wird von einem Programmiermodell auf Task-Ebene ausgegangen, welches eine explizite Scheduling-Einheit - den CoreManager - nutzt [Sei06]. Er verteilt die zu einer Anwendung gehörenden kleinen Tasks zur Laufzeit auf die verfügbaren Verarbeitungseinheiten (Kapitel 4.3). Die Arbeit dehnt das CoreManager Modell auf die Allokation von QoS-Kommunikationskanälen zwischen Verarbeitungseinheiten und Speichern oder I/O aus. Dazu werden verteilte und zentralisierte Verfahren untersucht, welche Kanäle durch das NoC auf Anfrage suchen und allokieren. Im zentralisierten Fall übernimmt dies eine neu eingeführte Einheit, NoCManager genannt. Im verteilten Fall erledigt das NoC mittels seiner Router diese Aufgabe selbst. Unsere Untersuchungen zeigten, dass der NoCManager den verteilten Verfahren hinsichtlich des Flächenverbrauchs und für die meisten Verkehrsszenarien auch hinsichtlich des Erfolgs der NoC-Ressourcenallokation überlegen ist. Desweiteren stellten wir fest, dass der Vergleich der Zwischenankunftszeit von QoS-Ressourcenforderungen mit der Bedienzeit der Allokationsverfahren als ein grobes Kriterium zur Auswahl des geeigneten Allokationsverfahrens dienen kann. Mit dieser Erweiterung ist es möglich, zur Laufzeit nicht nur Tasks mit Echtzeitanforderungen auf die parallelen Einheiten zu verteilen, sondern auch die zugehörigen Kommunikationsressourcen mit QoS-Garantien für Echtzeitkommunikation zu allokieren. Auf diese Weise wird ein komplettes, weiches Echtzeit-MPSoC mit dynamischer Ressourcenallokation auf fein-granularer Task-Ebene möglich.

## 1.5. Übersicht über die Arbeit

Nach der Einleitung wird in Kapitel 2 ein Überblick gegeben über die Grundlagen des NoCs sowie über grundlegende Begriffe und Architekturfragen. Im darauf folgenden Kapitel wird ein rein paketvermitteltes NoC für Best Effort Verkehr entwickelt und auf seine Leistungsfähigkeit in verschiedenen Szenarien und Netzwerkgrößen analysiert. Dabei werden auch mögliche Auswege aus der limitierten Skalierbarkeit heutiger 2D-Gitter Topologien diskutiert. Ausgehend von den Erkenntnissen aus Kapitel 3 bezüglich der Nichtvorhersagbarkeit von Bandbreite, Latenz und Jitter im BE NoC, wird im darauf folgenden Kapitel 4 auf QoS und Echtzeitunterstützung im gesamten MPSoC eingegangen. Dazu werden Techniken zur Realisierung von QoS im NoC dargestellt und verschiedene Programmiermodelle mit Echtzeitunterstützung für MPSoCs erläutert, insbesondere das CoreManager Modell als Basis

für die weitere Arbeit. Das Kapitel 5 behandelt schließlich die in dieser Arbeit entworfenen Techniken zur Allokation von QoS-Kommunikationsressourcen. Zum einen beschreibt es ihre prinzipielle Funktionsweise und zum anderen analysiert es die Leistungsfähigkeit und den Flächenverbrauch der Verfahren. Dies umfasst zentralisierte Allokationsprinzipien des NoCManagers wie auch verteilte Mechanismen, die über das NoC selbst Ressourcen allokieren. Eine Erweiterung dieser QoS-Modelle sowie ihrer Allokation auf zeitgeteilte Kanäle wird in Kapitel 6 angesprochen. Das letzte Kapitel fasst die Arbeit zusammen und zeigt offene Punkte für weitere Forschungsmöglichkeiten auf.