

Network on Chip Routing & Switching

Heiko Bordasch

Anwendungen 2 - Ausarbeitung

Network on Chip - Routing & Switching

Heiko Bordasch

Ausarbeitung eingereicht im Rahmen von Anwendungen 2
im Studiengang Master Informatik
Department Informatik
in der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Professor : Prof. Dr.-Ing. Bernd Schwarz
Gutachter : Prof. Dr. rer. nat. Kai von Luck
Gutachter : Prof. Dr. rer. nat. Gunter Klemnke

Abgegeben am 18. August 2010

Inhaltsverzeichnis

1	Einleitung	4
1.1	Routing & Switching in Network-on-Chips	6
1.2	IEEE Xplore Analyse	7
2	Vergleichbare Netzwerkarchitekturen	8
2.1	KAIST - Basic On-Chip Network	8
2.2	Philips Aethereal	10
2.3	Arteris NoC Solution	12
3	Zusammenfassung & Ausblick	14
	Literaturverzeichnis	16
	Abbildungsverzeichnis	18

1 Einleitung

Durch den stetig steigenden Bedarf an Rechenleistung für spezifische Anwendungen gewinnt der Entwurf der Kommunikationsinfrastruktur bei der Entwicklung von System-on-Chips (SoC) zunehmend an Bedeutung. Der Einsatz von Multiprozessoren zur Datendurchsatzserhöhung und zur gleichzeitigen Verminderung des Stromverbrauchs ist in vielen Gebieten unabdingbar, wie z.B. in multimedialen Anwendungen [15]. Multiprozessor System-on-Chips (MPSoC) bestehen aus einer großen Anzahl an interagierenden Beschleunigermodulen (Intellectual Property Cores, IPs), die als Peripheriekomponenten mit den Prozessoren gekoppelt sind. Herkömmliche Singlebussysteme sind für derartige Systeme mit großen Datenmengen und vielen Kommunikationspartner nicht geeignet [19]. Aus diesem Grund sind reale paketorientierte Verbindungsnetzwerke, die das MPSoC als Komplettsystem sehen, erforderlich. Durch das Übertragen von bekannten Konzepten aus der heutigen Netzwerktechnik auf das Chip Design kann dies erreicht werden. Diese Art von Interkommunikationsarchitektur wird Network-on-Chip (NoC) genannt. Jede Ressource in einem NoC besitzt eine eindeutige Adresse und kann mit jeder anderen Ressource kommunizieren. Ähnlich zu dem 7-schichtigen OSI-Referenzmodell kann ein NoC durch vier Protokollschichten definiert werden [18]:

1. **Physical Layer:** Stellt die elektrische Infrastruktur bereit. Beschreibt die Anzahl und die maximale Länge der Signalleitungen / Links.
2. **Data-Link Layer:** Definiert das Protokoll für die Kommunikation zwischen Netzwerk-Interface und Router. Verantwortlich für sichere Übertragung - Flow Control Mechanismus. Beispiel: STMicroelectronics STBus [19]
3. **Transaction Layer:** Implementiert das Routing & Switching des Netzwerkes. Verpackt und entpackt ankommende Pakete vom Router oder vom Netzwerk-Interface. In Hardware implementiert.
4. **Application Layer:** Beschreibt, wie eine Ressource Daten an das zugehörige Netzwerk-Interface senden kann. Entweder per Software oder Hardware. Ist vollkommen unabhängig von der NoC Technologie [14]

Der Einsatz von Network-on-Chip auf MPSoC Plattformen bietet gegenüber Singlebussysteme, durch die parallele Transaktionsabarbeitung (Transaction Layer) und den modularen Aufbau, eine durchsatzstärkere Kommunikation. Die Modularisierung der zahlreichen Komponenten eines SoCs ermöglicht die Verwendung des GALS¹ Prinzips [14]. Gerade bei Echtzeitsystemen, die harte Zeitanforderungen erfüllen müssen, ist die schnelle und parallele Bearbeitung von Instruktionen wichtig. Nicht nur bei FPGA-basierten Systemen hat man die Vorteile und die Bedeutung der NoC Interkommunikation erkannt. Auch bei der Sony Playstation 3 kommt mit dem IBM Cell Prozessor ein aus acht CPU

¹Globally Asynchronous Locally Synchronous

bestehender Prozessor mit NoC Architektur zum Einsatz [9]. Die Firma Intel hat mit ihrem 80-kernigen „Teraflop Research Chip“ (Polaris) ebenfalls einen NoC-basierten Chip auf den Markt gebracht [10].

Die Wahl einer geeigneten NoC-Topologie ist an den Anforderungen der jeweiligen Anwendung auszurichten. Die wohl am meisten verwendete Topologie ist das 2D-Mesh-Netzwerk [21]. Hierbei ist jede Ressource mit einem oder mehreren Routern verbunden. Beim Ausfall eines Routers ist durch eine Umleitung der Pakete mit einem adaptiven Routing die Datenkommunikation weiterhin möglich. Dies ist beispielsweise bei einer Ring-Topologie durch die Pipeline Verarbeitung der Pakete nicht möglich. Fällt eine im Ring angeordnete Komponente aus, dann wird die gesamte NoC Kommunikation stillgelegt. Im Kern besteht jede NoC Architektur aus folgenden Komponenten:

- **Netzwerk-Interface:** Jede Ressource besitzt ein eigenes Netzwerk-Interface (NI). Dieses implementiert alle vier NoC Protokollschichten und entkoppelt die Kommunikation vom System. Das NI verpackt die zu sendende Nachricht (Application Layer Package) und leitet sie an den angeschlossenen Router weiter. Analog entpackt es die ankommenden Pakete (Transaction Layer Package). Als Beispiel dient das „Virtual Component Interface“ (VCI) [25].
- **Router:** Kernelement eines NoCs. Implementiert den Routing Algorithmus bzw. die Switching Technik. Ein Router besteht meist aus einer Switch mit Routing-tabelle, mehreren Buffers, einer Arbitrationslogik und mehreren In-/Output Ports. Einkommende Pakete vom Netzwerk-Interface oder einem anderen Router werden an den entsprechenden Output Port weitergeleitet. Je nach Anwendung wird zusätzlich ein Flow-Control Mechanismus implementiert. Der eingesetzte Routingalgorithmus ist stark abhängig von der gewünschten Funktion und den bereitstehenden Ressourcen (vgl. Abschnitt 1.1)
- **Links:** Stellen sowohl die direkte Verbindung zwischen NI und Router als auch die Verbindung zwischen zwei Routern bereit. Ein für die Kommunikation oft eingesetztes und weit verbreitetes Protokoll ist das „Open Core Protocol“ (OCP) [20] oder der zu AMBA² gehörende „Advanced High-performance Bus“ (AHB). Letzterer wurde von der Firma ARM entwickelt und ist seit der ISE Design Suite 12.0 voll kompatibel mit den Xilinx FPGAs Spartan 6 und Virtex 6 [27]. In Hinblick auf die Masterarbeit und die damit bevorstehende Entwicklung einer FPGA basierten NoC-Architektur ist dies ein interessanter Aspekt.

Für die Entwicklung von MPSoC's ist ein hoher Grad an Wiederverwendbarkeit und Skalierbarkeit des Netzwerkes und der einzelnen Komponenten von großer Bedeutung. Einzelne IP-Blöcke sollen schnell und ohne großen Implementationsaufwand ausgetauscht werden können. Außerdem muss das NoC die Fähigkeit besitzen, neue Komponenten und deren Kommunikation schnell zu integrieren. Dieser Aspekt wird durch den Einsatz von Netzwerk-Interfaces und die damit verbundene Trennung von System- und Netzwerkdesign erreicht. Die Hauptaufgabe bei der Entwicklung einer NoC Architektur liegt in der Implementierung des Routingalgorithmus. Hierfür haben sich in den letzten Jahren je nach Anwendung verschiedene Verfahren etabliert.

²Advanced Microcontroller Bus Architecture

1.1 Routing & Switching in Network-on-Chips

Die Switching Techniken beschreiben wie die Daten in den Routern transportiert und mit FIFOs gespeichert werden. Das Routing bestimmt wie das Paket von Sender zu Empfänger gelangt. Damit die Buffer in den Routern nicht überlaufen, werden sie mit Hilfe von zusätzlichen Kontrollbits synchronisiert. Eine Nachricht besteht immer aus einem oder mehreren Flits (Flow Control Unit). Diese haben eine minimale Bitgröße entsprechend der Dateneinheit, die innerhalb eines Zyklus über einen Link transportiert werden kann, den sogenannten Phit (Physical Unit) [19]. Es gibt zwei grundlegende Switching Techniken:

- **Circuit Switching:** Hierbei wird durch eine Set-Up-Phase (3-Wege Handshake) ein kompletter Pfad von Sender zu Empfänger reserviert. In vielen Fällen erfolgt die Reservierung mit einem TDMA Verfahren. Die Nachrichten werden im Kompletten gesendet und enthalten keine Routing Informationen. Explizite Link Reservierung hat den Nachteil, dass bei den weiteren Komponenten Verzögerungen (Delays) auftreten können. Es wird hauptsächlich bei zeitkritischen und garantierten Services verwendet [17]. Durch den Einsatz von Virtual Channels kann ein einziger physikalischer Link geteilt und somit die Delays verkürzt werden [19]. Ein Beispiel ist das in [7] vorgestellte NoC.
- **Packet Switching:** Nachrichten werden in Paketen gesendet und enthalten Routing Informationen im Header. Die im Netz enthaltenen Routern sind für die korrekte Weiterleitung des Paketes verantwortlich, z.B. mit Routingtabellen. Da keine Link Reservierung stattfindet wird der sogenannte Best-Effort Service geboten [17]. Dieser wird für zeitunkritische oder niedrig priorisierte Pakete verwendet. Ein garantierter Service ist durch die Mehrfachbenutzung der Links durch unterschiedliche Nachrichten nicht möglich (Contention). Das Packet Switching erfordert ein Buffering in den Routern und benötigt somit einen höheren Platzbedarf auf dem Chip. Man unterscheidet zwischen drei Packet Switching Techniken:
 - Store and Forward: Der Router trifft erst eine Routing Entscheidung wenn das gesamte Paket empfangen wurde und der nachfolgende Router Platz für das gesamte Paket hat. Die Buffergröße in den Routern ist abhängig von der maximalen Nachrichtengröße. Wird aufgrund des Platzbedarfs und der auftretenden Verzögerungen nur selten eingesetzt.
 - Virtual Cut Through: Reduziert die Latenz indem es den ersten Flit sofort weiterleitet wenn der nachfolgende Router Platz für das gesamte Paket hat. Ist dies nicht der Fall, wird ebenfalls die gesamte Nachricht gespeichert. Wird aufgrund der Buffer Anforderung ebenfalls selten eingesetzt [?]
 - Wormhole: Ist die meist eingesetzte Switching Technik [17]. Das Paket wird in Flits aufgeteilt und diese werden sobald der nachfolgende Router Platz für einen Flit hat geroutet. Somit entsteht eine minimale Buffer Anforderung von nur noch einem Flit.

Zur effizienten Kommunikation und zur Bündelung der Vorteile wird oft eine Kombination aus Circuit und Packet Switching gewählt. Dies ist beispielsweise in dem in Kapitel 2.2 vorgestellten NoC der Fall.

1.2 IEEE Xplore Analyse

Die immer größer werdende Bedeutung der „Network on Chip“ Technologie bei FPGA basierten Systemen zeigt sich auch an der stetig steigenden Anzahl von IEEE³ Veröffentlichungen in den letzten Jahren (vgl. Abb. 1.1). Um dies zu belegen, wurden mit der „IEEE Xplore Digital Library“ sämtliche Veröffentlichungen unter den Schlagwörtern „Network on Chip“ und „FPGA“ analysiert. Das erste Paper wurde im Jahre 1991 auf der IEEE Custom Integrated Circuits Conference vorgestellt. Hierbei ist anzumerken, dass erst 1985 mit dem Xilinx XC2064 der erste kommerziell frei zugängliche FPGA auf den Markt kam [28]. Es wurde also früh bemerkt, dass die herkömmliche Kommunikation über Bussysteme für den Einsatz in Multiprozessorsystemen nicht geeignet ist. Bis heute sind insgesamt 2.078 IEEE Aufsätze erschienen, die zum Teil nur im weitesten Sinne Bezüge zum heutigen NoC aufweisen (Stand: 29.05.2010).

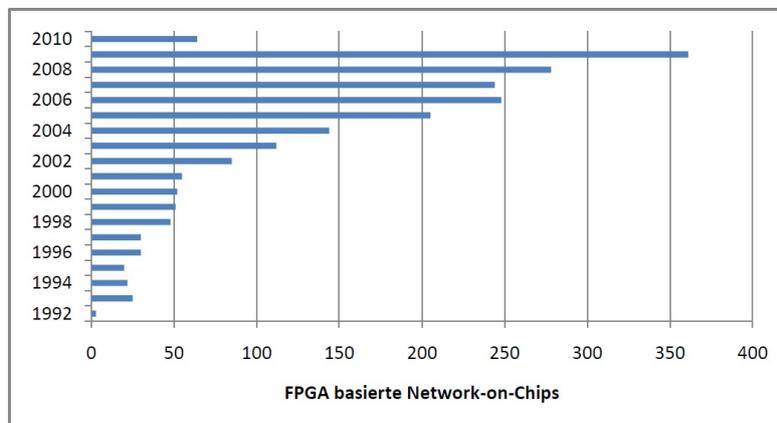


Abbildung 1.1: IEEE Xplore Veröffentlichungen mit den Schlagwörtern „Network on Chip“ und „FPGA“ (Stand: 29.05.2010)

Von allen gefundenen Veröffentlichungen erläutern lediglich 189 Arbeiten, seit dem Jahre 2002, das heute bekannte „Network on Chip“ Paradigma. Erste Ansätze sind jedoch schon in früheren Jahren, wie beispielsweise in [4] oder [5], zu erkennen. Dort wurden in ersten Forschungsprojekten heutige Grundlagen zur Kommunikationsinfrastruktur und Routingverfahren, wie das Message Passing Prinzip oder die Crossbar Switch, entwickelt.

Obwohl die Entwicklung von NoC's noch meist in akademischen Händen liegt, steigt die Anzahl der kommerziellen Anbieter von synthetisierbaren Network-on-Chip Produkten an. Hierzu zählt beispielsweise die Firma Arteris, deren NoC IP-Suite bei den Entertainment Systemen von NEC oder bei Texas Instrument eingesetzt wird [8]. Ein weiterer marktführender Anbieter für NoC's ist die Firma Sonics. Diese hat sich mit ihren SonicMX, SonicLX und SonicSX Netzen vor allem im Bereich Multimedia und mobiler Telekommunikation etabliert. Viele namhaften und führenden SoC-Hersteller haben die Sonics Produkte lizenziert und setzen diese zur Interkommunikation ein (Nokia, Motorola, Toshiba oder Samsung) [22].

³IEEE - Institute of Electrical and Electronics Engineers

ohne globaler Synchronisation. Wenn eine Nachricht an einen anderen Chip weitergeleitet werden muss, dann wird ein Strobe Signal zusammen mit dem Paket, an das an der Switch angeschlossene „Off-Chip Gateway“ gesendet. Dieses ist durch einen externen Takt mit allen weiteren OGW's synchronisiert. Die in BONE eingesetzte Crossbar Switch besteht aus einer konfigurierbaren Anzahl an „Output Port Indexes (OPI)“ und „Input Port Indexes (IPI)“. Jedes Paket verfügt über einen 16 Bit großen mit Routingin-

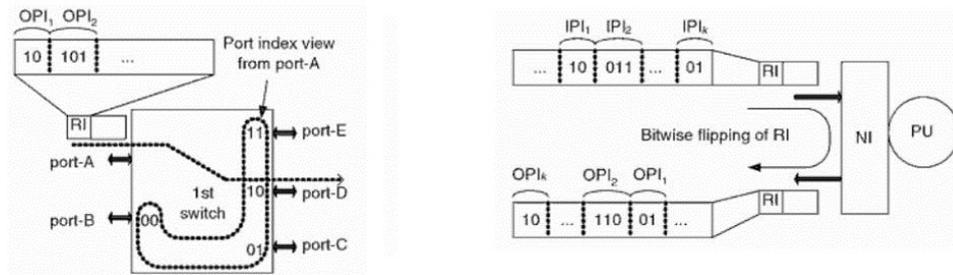


Abbildung 2.2: Switch OPI Identifikation und RI Feld Modifikation [19]

formationen bestückten Header (RI Feld) und einer 64 Bit großen Payload. Das RI Feld enthält nicht wie in den meisten NoC's die Quell- und Zieladresse, sondern eine Serie von Switch OPI's. Hierfür ist erforderlich, dass jede Komponente ein Wissen über alle Switches und deren Ports hat. Das Routing in den Switches erfolgt entsprechend dem RI Feld. Jedes Paket hat die gesamte Route von Sender zu Empfänger anhand den OPI's in seinem Header identifiziert. Kommt ein Paket an einer Switch an, dann prüft diese den ersten Eintrag im RI Feld (vgl. Abb. 2.2 links). Dieser kennzeichnet den zugehörigen OPI, an den die Switch das Paket weiterleiten muss. Nachdem sie das Paket geroutet hat, wird das RI Feld um eins nach links verschoben. So wird der nächste OPI für die nächste Switch lokalisiert und der IPI, über den das Paket eintrifft, hinten angehängt. Das Netzwerk-Interface des Ziel IP's erhält durch die Links Verschiebung des RI Feldes einen Paketheader mit dem gesamten Rückweg (vgl. Abb. 2.2 rechts).

Durch dieses statische Routingverfahren benötigen die Switches keine Routingtabellen und können die Pakete entsprechend einem „Cut Through Switching“ routen (vgl. Kap. 1.1). Jede Switch muss stets nur den ersten Eintrag im RI Feld prüfen und dementsprechend das Paket weiterleiten.

- **Vorteile:** Der Netzwerk Entwurf wird vom System Entwurf entkoppelt und bietet damit eine einfache Portierung des NoC's auf ein SoC [19]. In vielen anderen NoC's, deren Router statische Routingtabellen haben, müssen diese bei Erweiterung des Systems angepasst werden. Eine Entkopplung ist damit nicht gegeben. Ein weiterer Vorteil ist die feste Headergröße. Diese reduziert die Latenz und es ist kein „Header Parsing“ in den Switches notwendig.
- **Nachteile:** Das BONE Routing unterstützt kein adaptives Routing. Fällt eine Switch oder ein Link aus, so kann durch das statische Routing das Paket nicht zu seinem Ziel IP transportiert werden. Außerdem wird bei BONE keine Flow Control implementiert. Der Sender bekommt keine Information ob das Paket beim Empfänger angekommen ist. Dies macht das Netz ungeeignet für den Einsatz in Echtzeit oder Multimedia Anwendungen. Außerdem kann bei gleichzeitiger Anforderung des gleichen Links durch unterschiedliche Pakete „Contention“ auftreten.

2.2 Philips Aethereal

Das Aethereal NoC wurde im Jahre 2001 von den „Philips Research Laboratories“ entwickelt. Es wurde für den Einsatz auf SoC's in der Unterhaltungselektronik konzipiert (Beispiel: Digital TV). Im Vergleich zu BONE ist es weithaus komplexer und für den Einsatz bei Audio- und Videoanwendungen mit Echtzeit-Performance und hohen Datenmengen geeignet [12]. Bei der Entwicklung von Aethereal wurden die Ziele, garantierte Paket Services und garantierte Echtzeit Performance mit Energieeffizienz, verfolgt [13]. Eine hohe Anzahl an IP Blöcke werden mit einer Shared Memory Architektur verbunden und die Pakete sollten verlustlos, geordnet und mit begrenzten Latenzen von Sender zu Empfänger transportiert werden. Um diese Ziele zu realisieren wurde eine Kombination aus „Circuit Switching“ (Guaranteed Service) und „Packet Switching“ (Best-Effort Service) implementiert (vgl. Kap. 1.1). Das sogenannte „Contention Free Routing“ oder auch „Pipelined TDMA Circuit Switching“ bietet eine garantierte Datenübertragung ohne einen Paketverlust [11].

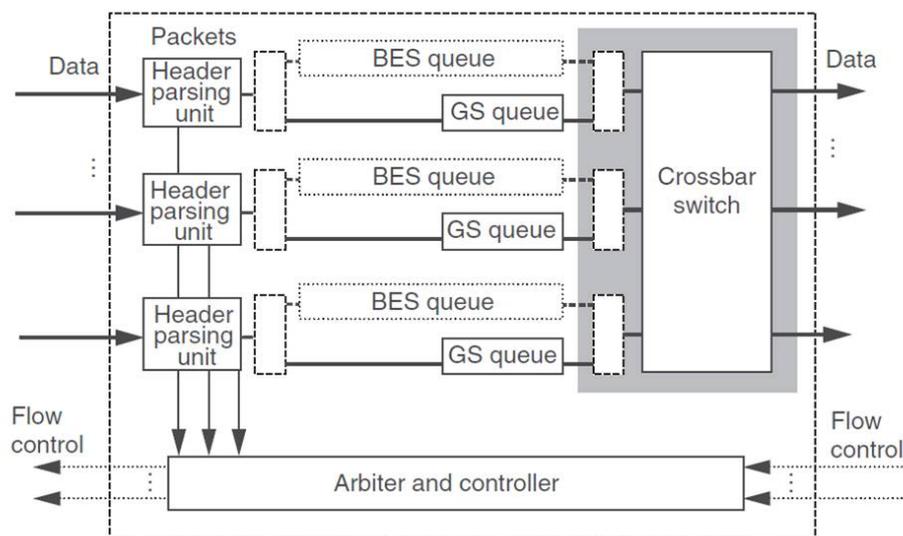


Abbildung 2.3: Aethereal Architektur mit zentralisiertem Programmiermodell [12]

- Guaranteed Service (GS) Routing:** Durch das „Contention Free Routing“ werden die Pakete vom Netzwerk-Interface (NI) in einer zeitlichen Reihenfolge gesendet. Dadurch wird garantiert, dass Pakete nicht kollidieren und verlustfrei und geordnet beim Empfänger ankommen. Durch ein TDMA¹ Verfahren wird jedem Sender statisch ein bestimmter Zeitschlitz zugeordnet. Jedes NI verfügt über eine vorkonfigurierte „Slot Table“, in der alle Zeitschlitze gespeichert werden. Für die Weiterleitung in einem GT Router (vgl. Abb. 2.3) gibt es zwei unterschiedliche Möglichkeiten. Hat der Router eine eigene „Slot Table“, dann kann das Paket ohne Header gesendet werden. Bei der Ankunft eines Paketes zum Zeitpunkt t schaut der Router in seine Tabelle und leitet es an den dort gespeicherten „Output Port“ weiter. Enthält ein Paket einen Header, dann wird mit Hilfe der „Header Parsing Unit“ (vgl. Abb. 2.3)

¹Time Division Multiple Access

der dort gespeicherte Ziel IP identifiziert. Das GS Routing erfordert eine globale Synchronisation aller Sender. Außerdem resultiert die Link Reservierung in eine schlechte Auslastung der Ressourcen [12].

- **Best-Effort Service (BES) Routing:** Nicht relevante oder zeitunkritische Daten werden mit Hilfe des BES Routings gesendet. Die vom GS Routing nicht genutzten Ressourcen werden durch ein Round Robin Verfahren mit niedrig priorisierten BES-Pakete belegt. Jedes BES-Paket besteht aus einem Header mit der kompletten Route von Sender zu Empfänger. Dies entspricht einem „Source Routing“ Algorithmus. Beim Empfang eines BES-Paketes benutzt der Router ein „Wormhole Routing“ um die Daten weiterzuleiten (vgl. Kap. 1.1). Hierfür ist ein FIFO Buffer, der durch eine Link-Level Flow Control von einem Überlauf geschützt werden, erforderlich [12]. Durch BES-Pakete können die für das GS-Routing benötigten statischen „Slot Tables“ in einem Router programmiert werden. BES-Routing hat den Vorteil, das die Ressourcen in einem Netzwerk besser ausgelastet werden. Jedoch besteht keine Garantie für eine Übertragung der Daten oder über eine bestimmte Bandbreite. Bei einem hohen Aufkommen an GS-Paketen erhöht sich die Latenz für die BES-Pakete.

Für die Kommunikation über die Netzwerk-Interfaces bietet das Aethereal NoC eine hohe Kompatibilität mit vielen bekannten Protokollen (OCP[20], AMBA[27] oder VCI[25]). Durch eine Ende-zu-Ende Flow Control mit Piggyback Antwort werden sowohl Buffer Overflows als auch Deadlocks vermieden. Eine Kernaufgabe bei der Implementierung von Aethereal ist das Programmiermodell. Dieses beschreibt wie die „Slot Tables“ in einem NI oder einem Router gefüllt werden. Hierfür gibt es zwei Ansätze: Das verteilte und das zentralisierte Modell [12]. Beim verteilten Modell sind die Tabellen zu Beginn leer und werden mit einem 3-Wege Handshake durch BES-Pakete gefüllt. Beim zentralisierten Modell werden die Tabellen von einer zentralen Instanz, beispielsweise einem Prozessor, programmiert. Die Kombination aus GS-Routing und BES-Routing in einer Router Architektur hat sowohl Vorteile als auch Nachteile [24][13]:

- **Vorteile:** Das NoC ist durch seine modulare Struktur skalierbar und bietet eine garantierte und vorhersehbare Performance. Durch ein implementierter Flow-Control Algorithmus ist eine verlustfreie und geordnete Kommunikation möglich. Weiterhin ist eine automatisierte NoC Generierung durch das „NXP Tool Chain“ und den „NoC Design Flow“ möglich [11]. Dies ermöglicht dem Entwickler bzw. dem Hersteller eine schnelle und kostengünstige Markteinführung des SoC's.
- **Nachteile:** Durch die Kombination der Routingverfahren entsteht ein erhöhter Ressourcen- und Flächenverbrauch. Außerdem ist im Gegensatz zu BONE (vgl. Kap.2.1) durch die Aktualisierung der „Slot Tables“ keine Entkopplung von System- und Netzwerkdesign möglich. Bei einer Erweiterung des Systems muss eine Neuvergabe der „Slot Tables“ in jedem Router und jedem Netzwerk-Interface stattfinden. Durch das GS Routing muss für eine sichere Kommunikation die Synchronisation aller Sender sichergestellt sein.

Zum Einsatz kommt das Aethereal NoC auf dem PNX8550 TV Chip von Philips [23]. Er besteht aus insgesamt 62 IP Blöcken, 212 Speicherblöcken und 94 unterschiedlichen Clockdomains. Diese sind in einer heterogenen Pipeline Architektur. Mehrere 32-bit MIPS CPUs kontrollieren die Verarbeitung der Daten. Zusätzlich gibt es eine Reihe von

32-bit Tri-Media VLIW Prozessoren die für die Bild- und Videoverarbeitung zuständig sind [26]. Basierend auf der „Nexperia Digital Video Platform“ wird der PNX8550 in hochauflösenden digitalen TV-Systemen eingesetzt. Diese bestehen aus mehreren unterschiedlichen Chips, die über einen HSEL² miteinander kommunizieren (vgl. Abb. 2.4 links). Verschiedene Funktionalitäten sind über mehrere Partnerchips mit jeweils eigenen SoC's verteilt. Durch den Einsatz des Aethereal NoCs werden die einzelnen Systeme

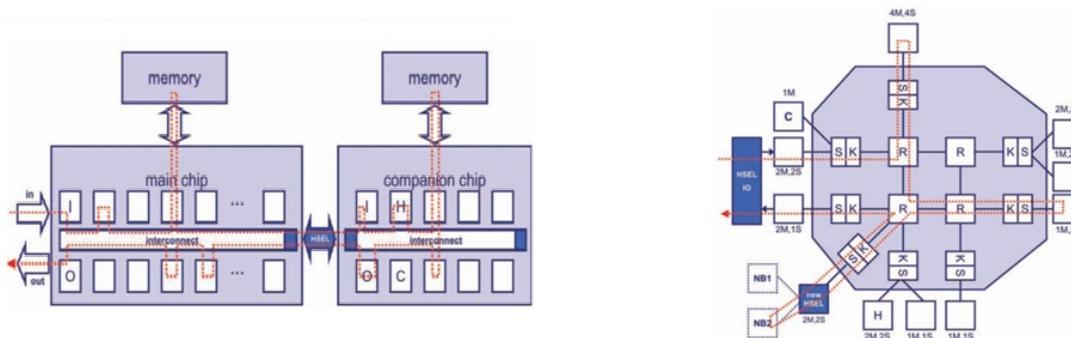


Abbildung 2.4: Nexperia PNX8550 TV SoC mit und ohne Aethereal NoC [23]

me durch eine 2D Mesh Topologie zu einem MPSoC zusammengefasst (vgl. Abb. 2.4 rechts) [23]. Zusätzlich werden die chipinternen Verbindungen eines aus neun IPs bestehenden Partnerchips zur Videoverbesserung durch ein NoC ersetzt. Dadurch wird sowohl eine bessere Skalierbarkeit des Systems als auch einer höhere Wiederverwendbarkeit der einzelnen Komponenten erreicht. Durch den Wegfall der herkömmlichen Bussysteme und dem Einsatz von GS-Routern mit TDMA (keine Header erforderlich) ist eine Echtzeitkommunikation möglich.

Die Vorteile mit Aethereal ergeben sich aus der hinzukommenden Rekonfigurierbarkeit des Systems. Einzelne IP Blöcke, beispielsweise mit neuen MPEG Codecs, können einfach und schnell ausgetauscht werden. Jedoch ist mit dem Einsatz eines NoCs auch ein höherer Platzbedarf und eventuell ein höherer Energiebedarf zu erwarten. Durch die automatische RTL VHDL Generierung mit dem „Philips Aethereal Design Flow“ ist eine kostengünstige und schnelle Markteinführung möglich.

2.3 Arteris NoC Solution

Obwohl die NoC-Technologie meist noch im Mittelpunkt des akademischen und wissenschaftlichen Interesses steht, wurde im Jahre 2005, die erste kommerziell verfügbare NoC Lösung von der Firma Arteris präsentiert [8]. Die „Arteris NoC Solution“ verbindet die einzelnen Komponenten des MPSoC Systems miteinander und steuert deren Kommunikation. Zur Verbesserung der Echtzeitanforderungen und der Wiederverwendbarkeit von IP Blöcken setzt Arteris geeignete Netzwerktechniken ein. Dadurch wird eine Maximierung der Dienstgüte (Quality of Service) und eine Verbesserung des Datendurchsatzes erreicht. Die „NoC Solution“ besteht im wesentlichen aus drei Komponenten:

²High Speed External Link

1. **Danube NoC IP Library:** Besteht aus einer Sammlung von synthetisierbaren und konfigurierbaren Bausteinen [1]. Diese können wahlweise mit den zur Verfügung stehenden Design Tools zu einem SoC zusammengefasst werden. Es gibt drei Einheiten, die das Mapping der einzelnen Schichten auf ein NoC ermöglichen (Netzwerk-Interfaces, Links, Switch Fabrics). Basierend auf dem paketorientierten „NoC Transaction and Transport Protocol (NTTP)“ ist die Bibliothek mit vielen weiteren Socket Standards kompatibel (OCP, AMBA, AXI). So ist es möglich jede Topologie zur Verbindung der einzelnen IP's zu nutzen (vgl. Abb. 2.5 rechts).
2. **NoCcompiler:** Dient als Designtool für die „Danube NoC IP Libraray“ [2]. Durch das graphische Interface kann der Entwickler die einzelnen Bibliothekseinheiten zu einem NoC zusammenfassen. Der „NoCcompiler“ generiert anschließend aus den einzelnen Instanzen synthetisierbaren HDL Code (wahlweise VHDL, Verilog oder SystemC). Das Tool ist vergleichbar mit dem „Embedded Development Kit (EDK)“ von Xilinx.
3. **NoCexplorer:** Zur Erzielung einer maximalen Leistung ist das System abhängig von der Topologie. Mit Hilfe des Simulationstools „NoCexplorer“ kann eine optimale Verbindungs- und Kommunikationsstrategie für die einzelnen Komponenten gefunden werden [3]. Das Tool basiert auf SystemC und dient zur Analyse und Modellierung von NoC's (vgl. Abb. 2.5 links).

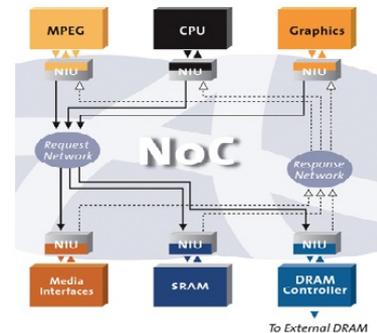


Abbildung 2.5: Arteris Simulationstool „NoCExplorer“ und „Danube IP Library“ [8]

Die Danube NoC IP Library ist als lizenzierbares IP-Produkt verfügbar und kommt beispielsweise bei den Firmen: NEC, Texas Instrument, Thales, NTT Electronics oder STMicroelectronics zum Einsatz [8]. Die Tools „NoCexplorer“ und „NoCcompiler“ sind separat erhältlich und können bei der Entwicklung von NoCs behilflich sein. Aus firmeninternen Gründen war es leider nicht möglich genauere Details über den Einsatz und die Implementierung der „NoCSolution“ zu bekommen. Der Vorteil der „NoC Solution“ liegt bei der individuellen Entwicklung eines Netzwerkes. Entwickler sind nicht festgelegt auf eine bestimmte Topologie oder ein Routingverfahren, sondern können das NoC explizit für eine spezifische Anwendung generieren. So lassen sich NoC's für kostengünstige, platzsparende und energieeffiziente MPSoC's entwickeln.

3 Zusammenfassung & Ausblick

Network on Chip ist eine neue Technologie, um Prozessoren über ein chipinternes Netzwerk miteinander zu verbinden. Eine solche Architektur besteht aus Netzwerk-Interfaces und Routern, die einen Routingalgorithmus zum Weiterleiten der Pakete an die entsprechenden Empfänger IP's implementiert haben. In dieser Arbeit wurden verschiedene Routingverfahren sowohl im Allgemeinen als auch an konkreten Netzwerken vorgestellt. NoC basierte FPGA Systeme bieten gegenüber Singlebussystemen durch die parallele Transaktionsabarbeitung eine effizientere Kommunikation. Gerade bei Multimedia- und Echtzeitanwendungen ist die schnelle und parallele Bearbeitung von Instruktionen wichtig. Derartige Anwendungen benötigen garantierte und vorhersehbare Services für die Übertragung der Daten (vgl. Kap. 2.2). NoC basierte Systeme sind in vielen Gebieten des „System on Chip Designs“ einsatzfähig. Ein Beispiel hierfür ist die Automobiltechnik. Die Anzahl der Fahrerassistenzsysteme und der Steuergeräte in zukünftigen Kraftfahrzeugen nimmt stetig zu. Hierfür ist aufgrund der echtzeit- und sicherheitsrelevanten Systemen eine effiziente Kommunikation notwendig.

In Hinblick auf die Masterarbeit wird im ersten Schritt eine FPGA basierte MPSoC Plattform entwickelt. Diese besteht aus mehreren Softcores (Xilinx MicroBlazes) und einer Shared Memory Architektur. Das „IBM PLB CoreConnect Bussystem“ wird durch ein NoC abgelöst. Jede Komponente wird zur Kommunikation mit einem zentralisiertem Router (Stern-Topologie) verbunden (vgl. Kap. 2.1). Das in BONE verwendete „Cut Through Switching“ wird durch ein durchsatzstärkeres „Wormhole Switching“ ersetzt. Durch die 1-Flit kleinen FIFO Buffer in den Routern minimiert sich der Ressourcenbedarf auf dem FPGA. Als Socket Schnittstelle für die Kommunikation hin zu den Routern ist einerseits der „Xilinx Fast Simplex Link (FSL)“ und das freiverfügbare „Open Core Protocol (OCP)“ denkbar. Eine genaue Analyse der verfügbaren Protokolle und Hardwarestrukturen steht ebenfalls an. Im zweiten Projektschritt soll das Netzwerk-Interface aus den Ergebnissen der Protokollanalyse prototypisch entwickelt und implementiert werden. Ein entscheidender Aspekt bei dieser Entwicklung ist die Entkopplung des Netzwerkentwurfs vom Systementwurf. Das MPSoC System soll, ohne eine Rekonfiguration der Router bei IP-Erweiterung, leicht erweiterbar sein.

Das NoC soll im Projekt „High Performance Embedded Computing“ in den Hardwarestrukturen mit dynamischer partiellen Rekonfiguration zum ersten Einsatz kommen. Dies hat zum Ziel, unter Verwendung von NoC und der partiellen Rekonfigurationstechnologie eine SoC abhängige Infrastruktur für die Verteilung von komplexen Berechnungsmodulen zu schaffen. Das NoC soll die verschiedenen Komponenten verbinden und eine gute Infrastruktur, mit hohem Datendurchsatz und geringen Latenzen, für das SoC bieten. Durch ein kompatibles und einheitliches Netzwerk-Interface ist mit dem Einsatz von „Partial Reconfiguration“ ein kompletter IP Austausch zur Laufzeit möglich. Zu Vergleichen ist dies mit dem Seti@Home Astronomie Projekt der Berkeley University [6].

Literaturverzeichnis

- [1] ARTERIS: Danube NoC IP Library. In: *Data Sheet* (2009), Nr. 7625v3
- [2] ARTERIS: NoCcompiler and NoCverifier. In: *Data Sheet* (2009), Nr. 7733v3
- [3] ARTERIS: NoCexplorer. In: *Data Sheet* (2009), Nr. 7232v3
- [4] A.TSUTSUI u. a.: Special Purpose FPGA for High-speed Digital Telecommunication Systems. In: *Design Automation Conference 1995* (1995), Nr. 10.1109/ASPdac.1995.486404
- [5] BOS, Michael L.: Design of a Chip Set for a Parallel Computer based on the Crossbar Interconnection Principle. In: *Circuits and Systems 1995* (1995), Nr. 10.1109/MWSCAS.1995.510198
- [6] CALIFORNIA, University of: *Seti@Home*. 2010. – URL <http://setiathome.berkeley.edu/>. – abgerufen 09.08.2010
- [7] C.HILTON ; B.NELSON: A flexible circuit switched NOC for FPGA based systems. In: *Field Programmable Logic and Applications* (2005), Nr. 0-7803-9362-7
- [8] CHIP COMPANY, Arteris The N. on: *Network-on-Chip IP for SoC Interconnect*. 2010. – URL <http://www.arteris.com/index.php>. – abgerufen 03.08.2010
- [9] CONGER, Nicola: Design and Performance Evaluation of Network-on-Chip Communication Protocols and Architectures. In: *Dottorato - Universita di Bologna* (2008), Nr. INF/01 - Ciclo XXI
- [10] CORPORATION, Intel: *Teraflops Research Chip*. 2010. – URL <http://techresearch.intel.com/articles/Tera-Scale/1449.htm>. – abgerufen 29.07.2010
- [11] GOOSSENS, Kees: *Networks on Chip at Philips Research and NXP Research*. 2007. – URL <http://ce.et.tudelft.nl/cecoll/slides/07/0117kees.pdf>. – abgerufen 06.08.2010
- [12] GOOSSENS, Kees u. a.: Aethereal Network on Chip: Concepts, Architectures, and Implementations. In: *IEEE Design & Test of Computers* (2005), Nr. 0740-7475/05
- [13] GOOSSENS, Kees ; HANSSON, Andreas: The Aethereal Network on Chip after Ten Years: Goals, Evolution, Lessons, and Future. In: *ACM Invited Paper* (2010), Nr. 978-1-4503-0002-5
- [14] JANTSCH, Axel u. a.: A Network on Chip Architecture and Design Methodology. In: *IEEE Computer Society Annual Symposium on VLSI* (2002), Nr. ISVLSI'02/0-7695-1486-3

- [15] JERRAYA, Ahmed A. ; WOLF, Wayne: *Multiprocessor Systems-on-Chips*. Elsevier, 2005. – ISBN 978-0-123-85251-9
- [16] LEE, Se-Joong u. a.: An 800 MHz Star-Connected On-Chip Network for Application to SoC. In: *2003 IEEE International Solid-State Circuits Conference* (2003), Nr. 0-7803-7707-9/03
- [17] MARCONETT, Dan: A Survey of Architectural Design and Implementation Trade-offs in Network on Chip Systems. In: *University of California* (2006), Nr. Technical Paper
- [18] M.DEHYADGARI u. a.: A new Protocol Stack model for Network on Chip. In: *IEEE VLSI Technologies and Architectures* (2006), Nr. 10.1109/ISVLSI.2006.7
- [19] MICHELI, Giovanni de ; BENINI, Luca: *Networks on Chips: Technology and Tools*. Morgan Kaufmann, 2006. – ISBN 978-0-12-370521-1
- [20] PARTNERSHIP, OCP I.: *OCP-IP*. 2010. – URL <http://www.ocpip.org/>. – abgerufen 29.07.2010
- [21] SALMINEN, Erno u. a.: Survey of Network-on-chip Proposals. In: *White Paper* (2008), Nr. OCP-IP 2008
- [22] SONICS: *On-Chip Communications Network IP*. 2010. – URL <http://www.sonicsinc.com/>. – abgerufen 03.08.2010
- [23] STEENHOF, Frits u. a.: Networks on Chips for High-End Consumer-Electronics TV System Architectures. In: *Design, Automation, and Test in Europe* (2006), Nr. 3-9810801-0-6/DATE06
- [24] TENHUNEN, Hannu ; JANTSCH, Axel: *Networks on Chip*. Springer Verlag, 2003. – ISBN 978-1402073922
- [25] VSI-ALLIANCE: Virtual Component Interface Standard. In: *On-Chip Bus Development Working Group* (2001), Nr. Version 2 (OCB2 2.0)
- [26] WANG, Laung-Terng ; STROUD, Charles E. ; TOUBA, Nur A.: *System-on-chip Test Architectures: Nanometer Design for Testability*. Morgan Kaufmann, 2007. – ISBN 978-0-12-373973-5
- [27] XILINX: *AMBA AXI 4 Interconnect Protocol*. 2010. – URL <http://www.xilinx.com/tools/axi-4.htm>. – abgerufen 27.07.2010
- [28] XILINX: *Our History - How Xilinx began*. 2010. – URL <http://www.xilinx.com/company/history.htm>. – abgerufen 29.05.2010

Abbildungsverzeichnis

1.1	IEEE Xplore Veröffentlichungen mit den Schlagwörtern „Network on Chip“ und „FPGA“ (Stand: 29.05.2010)	7
2.1	Architektur des BONE Netzwerkes [16]	8
2.2	Switch OPI Identifikation und RI Feld Modifikation [19]	9
2.3	Aethereal Architektur mit zentralisiertem Programmiermodell [12]	10
2.4	Nexperia PNX8550 TV SoC mit und ohne Aethereal NoC [23]	12
2.5	Arteris Simulationstool „NoCEXplorer“ und „Danube IP Library“ [8]	13