

4.15 **Schneller Kommunikationsprozessor mit adaptiver Schaltungsstruktur für 2D, 3D und Hypercube-Netze (CADAS)¹⁾**

(Projektlaufzeit: 01.11.93 bis 31.03.95)

Gerd Karl Heinz, Stephan Dengel, Carsten Busch

Zielstellung

Steigende Prozessorleistungen verlangen schnellere Kommunikation paralleler Rechner. Im Projekt wird ein Kommunikationsprozessor (CP) für serielle Schnittstellen entwickelt, der extrem schnell auf Basis von parallelen Automatengraphen arbeitet. Es ist möglich, deterministisch und adaptiv zu routen. Kernproblem der Entwicklung ist es, trotz verschiedener Nummerierungsschemata (2D, 3D: Signed Integer; Hypercube (HC): Binary) und verschiedener, physischer Netzkonnektivität einen gemeinsamen Grundalgorithmus in Hardware zu realisieren.

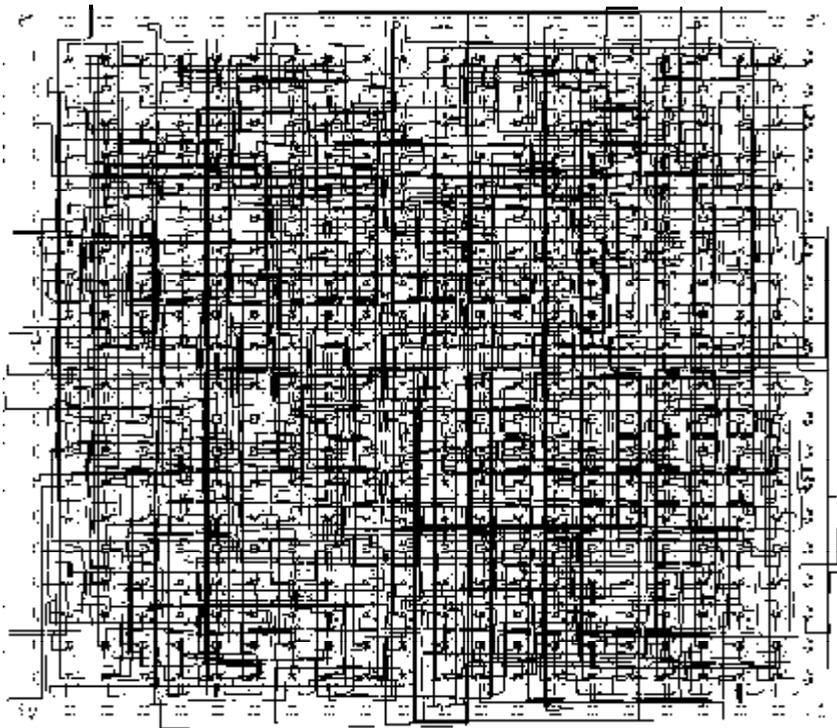


Bild 1: Realisierung eines CP für 4+1 Links in Single- Stage- Architektur auf einem PLD vom Typ Xilinx XC4010 (ca. 8500 Gates)

Resultate

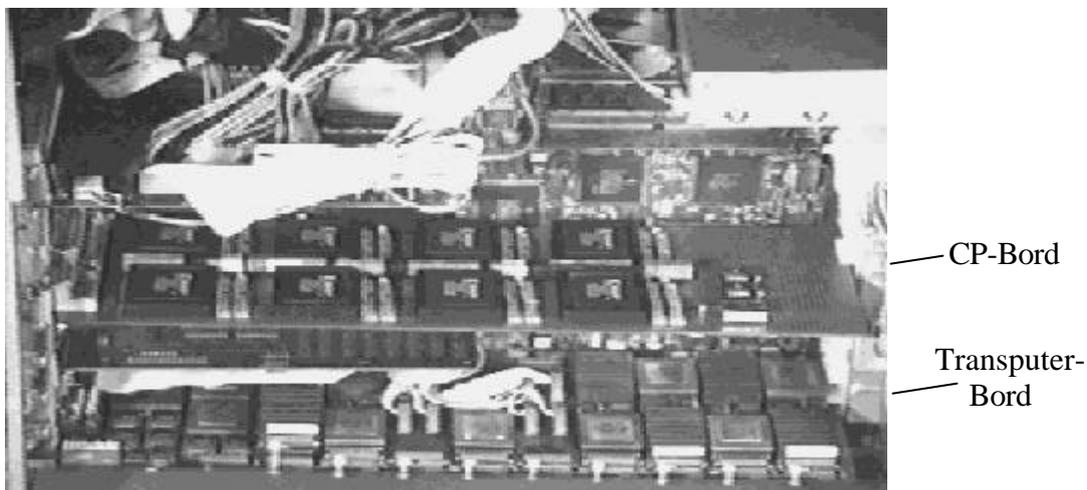
Die benutzten Algorithmen gehen auf Arbeiten von Krapp/Jossifov und fernerhin auf Wormhole-Routing und Mad-Postman-Strategien zurück. Mittels VHDL-Schaltungssynthese (Mentor Autologic) wurden verschiedene Varianten untersucht. Zur Erprobung dienen ein Inmos-T805 Transputercluster, sowie ein Parsytec-MPC. Je ein zu einem Transputer-TRAM gehörender Kommunikationsprozessor ist in einem PLD Xilinx XC4010 ²⁾ realisiert.

1) Das Projekt wird mit Mitteln des Bundesministeriums für Wirtschaft unter dem Förderkennzeichen 37/94 (1.11.93 bis 31.3.95) gefördert.

2) Die auf dem Xilinx-PLD verfügbare Gatterzahl begrenzt die Anwendung noch auf 4D-HC bzw. 2D-Netze

Krapp und Jossifov (siehe u.a. [1] bis [4]) entwarfen - aufbauend auf der Architektur paralleler Automatenmodelle nach [5] - verschiedene Algorithmen für parallel arbeitende Kommunikationsprozessoren in parallelen Netzwerken (2 dimensionaler Torus, 3 dimensionaler Torus, Hypercube). Auf der Basis des deterministischen 'e-cube-routing' Algorithmus [5] (Intel iPSC/2 und iPSC/860) und der adaptiven Strategien vom Typ der 'mad postman' [4] in den Varianten vom Typ 'store and forward' und 'wormhole packet routing' entstanden effiziente packet-routing Algorithmen vom Typ deterministisch und adaptiv ('dynamic-load-balancing') für 2D-, 3D- und Hypercube- Netzwerke, die eine vergleichbare Struktur aufweisen [1, 2].

Eine Analyse dieser verschiedenen Algorithmenklassen zeigt, daß effiziente Kommunikationsalgorithmen für verschiedene Netztypen auf eine effiziente Grundkonstruktion hinführbar sind. Dazu wurden verschiedene Algorithmen in unterschiedlicher Weise auf Basis von Gate-Logic ebenso wie mit VHDL und Synthese mit Mentor-Autologic erprobt [NP-COMPAS]. Ausgehend von strukturellen Hardwarebeschreibungen in VHDL werden Schaltungsvarianten mit Mentor-Autologic synthetisiert und simuliert. Anhand eines VHDL-Verhaltensmodells werden komplexe Netzsituationen simuliert.



**Bild 2: Versuchsaufbau von 8 CP im Zusammenwirken mit einem Transputer-Cluster
in einem Pentium (Transputer: 8 TRAM InmosT805 4MB)**

Aufbauend darauf wurde ein mit VHDL-1076 verifizierten Entwurf auf Basis komplexer PLD-Schaltkreise (XILINX XC4010, 10.000 Gates) entwickelt. Dieser kann quasi-parallel zueinander 2D-, 3D- und HC-Netze in deterministischem und adaptivem Routing bedienen. Damit wird ermöglicht, ein Parallelrechnernetz ohne Task-Switching parallel zueinander mit verschiedenen Netztypen zu betreiben. Die Entwicklung ist an Links vom Transputertyp INMOS T805 gebunden. Der zu entwickelnde Kommunikationsprozessor ist im Gegensatz zu gebräuchlichen Koppelfeldern vom 'Single-Stage' Typ, d.h. jeder CP ist an einen Verarbeitungsprozessor gekoppelt, ein Koppelfeld entfällt.

Die Entwicklung eines Knotenprozessors für Transputermodule mit 4+1 Links (physisches 2D-Netz) stößt an die Grenzen der PLD-Technik hinsichtlich erforderlicher Komplexität und erforderlicher Taktrate vor. Als besonderes Problem galt es, die Aufsynchronisation eines Ports auf den asynchronen Datenstrom mit PLD-Mitteln zu lösen. Die Routingadresse wird dem einlaufenden, seriellen Datenstrom entnommen, die Kanalprüfung und -arbitrierung erfolgt innerhalb weniger Takte während des

Einlaufens des Datenpaketes in den Empfangspuffer. Wenn der Datenstrom das Ende des Eingangspuffers erreicht hat, ist der Vermittlungsvorgang bereits abgeschlossen, das Datenpaket geht ohne Stop auf eine freie, in Richtung des Ziels liegende Leitung. Das gewählte Verfahren zeigt höchste Einfachkeit und Transparenz bei Dead-Lock Freiheit sowie maximalem Datendurchsatz. Es toleriert auf physischer Ebene unterbrochene und belegte Leitungen, und gestattet Verbindungsstrukturen, deren real vorhandene, physische Struktur vollständig von der logischen Netzstruktur abweicht. Alle physischen Datenkanäle können prinzipiell entsprechend ihrer momentanen Belegtheit von mehreren, logischen Kanälen parallel benutzt werden.

Damit kann erstmals in der Vernetzung paralleler Rechner die Realisierung *virtueller Kanäle* in *virtuellen Maschinen* im *Multitasking* hardwaremäßig unterstützt werden.

Der Hardwareaufwand zur Realisierung eines CP ist relativ gering. In einem PLD vom Typ Xilinx XC 4010 finden fünf Links (4+1) eines CP einer Adressierungstiefe für acht Knoten (2D-Netz bzw. 3D-HC) Platz. Der Übergang auf neuerdings erhältliche, größere PLD vom Typ XC 4025 würde eine Ausführung mit 8+1 Links (8 Netz, 1 CPU) mit einem Adreßraum von 64 k Knoten ermöglichen. Da sämtliche Arbeiten VHDL-basierend angelegt sind, wären entsprechende Modifikationen auf Wunsch kurzfristig verfügbar. Im Gegensatz zu Referenzlösungen kann der komplette Kommunikationsprozessor für sechs Links (physisch unterlegter 3D-Torus) auf einem PLD bzw. Gatearray statt auf einer Leiterkarte Platz finden. Mit diesem neuartigen Konzept ist es möglich, virtuell verschiedene Netztypen (2D-Torus, 3D-Torus, Hypercube) auf ein- und demselben Netz parallel zueinander und unabhängig voneinander zu routen und zu betreiben, wenngleich das real vorhandene Netz stets von ein und demselben Typ ist und bleibt (zB. 3D). Aufgrund der Beschreibbarkeit des Verfahrens in Automatengraphen ist die Geschwindigkeit des Routing skalierbar, und an Hochgeschwindigkeitsanwendungen anpassbar.

Literatur

- [1] Krapp, M., Jossifov, V.: Formal Specification of a Distributed Packet- Routing Algorithm with Automaton- Nets. Techn. Rep. No.5, Bulgarian Academy of Sciences, CICT, Sofia 1990 (nD Hypercubes, deterministisch)
- [2] Krapp, M., Jossifov, V.: Formal Specification of a Dynamic Load-Balancing Packet Routing Algorithm for nD Hypercubes with Automaton-Nets. Zentralinstitut für Kybernetik und Informationsprozesse, IBL, Berlin, Juni 1991
- [3] Jossifov, V., Krapp, M.: Adaptive Wormhole Routing Algorithm for 3D Meshes Formalized with Automaton- Nets. Zentralinstitut für Kybernetik und Informationsprozesse, IBL, Berlin, Juni 1991
- [4] Getov, V., Jesshope, C.R.: Simulation facility of distributed memory system with "mad postmen" communication work, Proc. Sec. Europ. Distrib. Memory Comp. Conference, Munich, April 1991
- [5] Tuazon, J., Peterson, K., Puiel, K., Liebermann, D.: Caltech/ IPC Marc II Hypercube Concurrent Processor, Proc of ICPP, 1985, pp.666-673



*Der Chef der Thüringer Staatskanzlei
Staatssekretär Dr. Michael Krapp
bittet anlässlich der Konferenz Parcella 94
am 21. September 1994 um 19.30 Uhr zum*

Thüringer Abend

*in die historische Gaststätte Klosterkeller
Potsdam, Friedrich-Ebert-Straße 94*