TECHNISCHE UNIVERSITÄT DRESDEN

Center for Information Services and High Performance Computing (ZIH)

# Theorie und Einsatz von Verbindungseinrichtungen in parallelen Rechnersystemen

*Statische Verbindungsnetzwerke*

06. Mai 2011

Andy Georgi

INF 1046
Nöthnitzer Straße 46
01187 Dresden

0351 - 463 38783

Center for Information Services & High Performance Computing

## Foliensatz

**Verfügbarkeit der Folien**

Vorlesungswebseite:
`http://tu-dresden.de/die_tu_dresden/zentrale_einrichtungen/zih/lehre/ss2011/tevpr`



## Agenda

1. Einführung
2. Vollständige Vermaschung
3. Stern
4. Ring
5. Gitter
6. Baum
7. Cube
8. Literaturverzeichnis



## Agenda

1. Einführung

## Einführung I

- Kommunikation zwischen Knoten über feste Verbindungsleitungen wie in Kapitel 2 beschrieben
- Kosten beschränken sich auf Hardware-Unterstützung in den Knoten und Leitungen
- Beschreibung der topologischen Verbindungsstrukturen mittels *Verbindungsfunktionen*



## Einführung II

**Verbindungsfunktion**

Verfügt ein Knoten A über eine *Verbindungsfunktion* $f(A)$, so werden bei Ausführung dieser Funktion Daten vom Knoten $A$ zu Knoten $f(A) = B$ transferiert.



## Einführung III

- Verbindungsfunktionen in SIMD-Systemen:
  - Alle aktiven PEs führen die gleiche Verbindungsfunktion aus
  - Passive PEs senden keine Daten, können allerdings Daten empfangen
  - Datenverlust möglich
- Verbindungsfunktionen in MIMD-Systemen:
  - Befehlsverarbeitung erfolgt unabhängig voneinander
  - I.d.R. Zwischenpufferung der Daten



## Agenda

2 Vollständige Vermaschung

## Vollständige Vermaschung

- Dedizierte Verbindungen von jedem Knoten zu jedem anderen Knoten
- Interne Konflikte ausgeschlossen
- Beschränkung der maximal erzielbaren Kommunikationsleistung erfolgt ausschließlich durch die technische Realisierung



## Agenda

3 Stern



## Stern

- Master-Slave-Architektur
- Verbindungsaufbau erfolgt über zentralen Knoten (*Master*)
- Effizient bei Multicast-Operationen oder Prozesssynchronisation



## Agenda

4 Ring
- Klassische Ring-Topologie
- Chordaler Ring
- Token-Ring
- Register-Insertion-Ring
- Scalable Coherent Interface

## Klassische Ring-Topologie

- Verbindungsfunktionen in einem *bidirektionalen* Ring:

$$ring_{+} = (P+1) \ mod \ N$$
$$ring_{-} = (P-1) \ mod \ N$$

- Ein *unidirektionaler* Ring implementiert nur eine der beiden Verbindungsfunktionen
- Keine Zugriffsbeschränkungen



## Chordaler Ring

- Ziel: Reduzierung des Durchmessers und der mittleren Weglänge
- Umsetzung: Einfügen zusätzlicher Verbindungen, sog. *Chords*
- Die *Chord-Länge C* bezeichnet den Abstand der verbundenen Knoten
- Ein Chordaler Ring nach Arden und Lee [ArL81] enthält stets eine gerade Anzahl von Knoten und vier Verbindungsfunktionen:

$$ring_{+1} = (P+1) \ mod \ N$$
$$ring_{-1} = (P-1) \ mod \ N$$
$$ring_{+C} = (P+C) \ mod \ N, \text{ wenn P ungerade}$$
$$ring_{-C} = (P-C) \ mod \ N, \text{ wenn P gerade}$$



## Token-Ring

- Zugangsregelung mit Hilfe von *Token*
- Ablauf im *IEEE 802.5 Token-Ring-Protokoll* standardisiert:
  1. Sender wartet auf *Free-Token*
  2. Umwandlung in *Busy-Token*, welches zusammen mit den zu sendenden Daten an den Ring übergeben wird
  3. Weiterleitung der Nachricht entsprechend der Verbindungsfunktion
  4. Entfernung der Nachricht vom Netz durch den Quellknoten
  5. Freigabe des *Tokens* nach der vollständigen Übertragung der Nachricht



## Register-Insertion-Ring

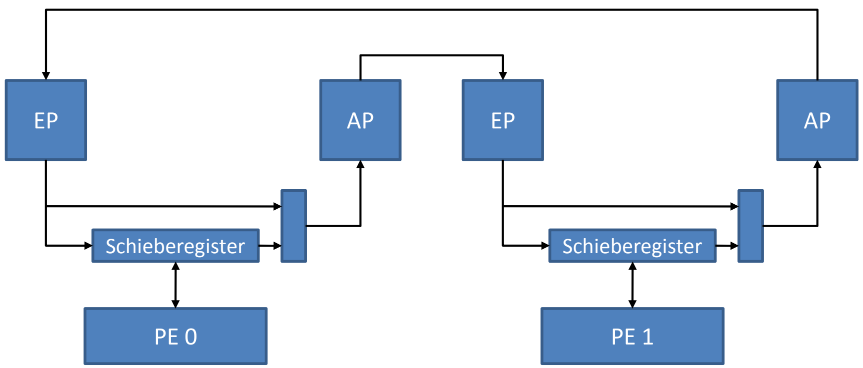


Abbildung: Register-Insertion-Ring mit zwei Knoten

## Scalable Coherent Interface

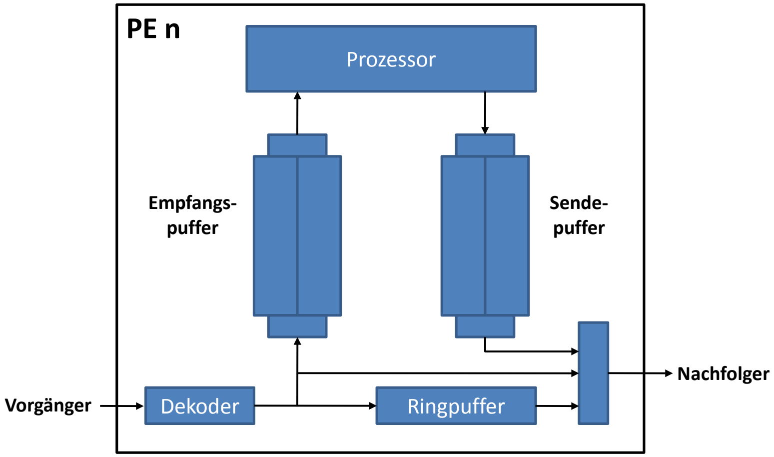


Abbildung: Blockschaltbild eines SCI-Knotens



## Agenda

5 Gitter
- Mesh-Netz
- Torus
- Illiac-Netz



## Zweidimensionale Gitter

- Verteilung der Knoten auf $M_R$ Reihen und $M_S$ Spalten
- Unterscheidung zwischen *offenem* und *geschlossenem* Gitter
- Befindet sich der betrachtete Knoten in Reihe $k$ auf Spalte $j$, so ergibt sich sein Index $I$ aus

$$I_H = j + M_S * k \quad \text{bei horizontaler, bzw.}$$
$$I_V = k + M_R * j \quad \text{bei vertikaler Zählrichtung}$$



## Mesh-Netz

- Offenes Gitter mit horizontalen und vertikalen Verbindungen der direkten Nachbarn
- Implementiert vier Verbindungsfunktionen:

$$\begin{aligned}
mesh_{right}\ (I_H) &= (j+1) + M_S * k \\
mesh_{left}\ (I_H) &= (j-1) + M_S * k \\
mesh_{down}\ (I_H) &= j + M_S * (k+1) \\
mesh_{up}\ (I_H) &= j + M_S * (k-1)
\end{aligned}$$

## Torus

- Geschlossenes Gitter mit horizontalen und vertikalen Verbindungen zwischen Randknoten einer Reihe bzw. Spalte
- Verbindungsfunktionen:

$$
\begin{aligned}
torus_{right}\ (I_H) &= ((j+1)\ mod\ M_S) + M_S * k \\
torus_{left}\ (I_H) &= ((j-1)\ mod\ M_S) + M_S * k \\
torus_{down}\ (I_H) &= j + M_S * ((k+1)\ mod\ M_R) \\
torus_{up}\ (I_H) &= j + M_S * ((k-1)\ mod\ M_R)
\end{aligned}
$$



## Illiac-Netz I

- Benannt nach dem Rechner Illiac-IV [BoD72], welcher diese Topologie implementierte
- Geschlossenes Gitter mit folgenden Eigenschaften:
  - Vertikale und horizontale Verbindungen zu den direkten Nachbarn
  - Verbindung der Randknoten einer Spalte
  - Verbindung des Endknotens einer Reihe mit dem Anfangsknoten der nachfolgenden Reihe



## Illiac-Netz II

- Verbindungsfunktionen:

$$
\begin{aligned}
illiac_{right}\ (I_H) &= ((j+1) + M_S * k)\ mod\ N \\
illiac_{left}\ (I_H) &= ((j-1) + M_S * k)\ mod\ N \\
illiac_{down}\ (I_H) &= (j + M_S * (k+1))\ mod\ N \\
illiac_{up}\ (I_H) &= (j + M_S * (k-1))\ mod\ N
\end{aligned}
$$



## Agenda

6. Baum
   - Binärbäume
   - k-fache Bäume
   - Ring-erweiterte Bäume
   - Hypertree
   - Fat-Tree

## Baumstrukturen

**Definition**

Aus graphentheoretischer Sicht handelt es sich bei einem Baum um einen ungerichteten zusammenhängenden azyklischen Graphen. Dabei ist dieser durch eine *Wurzel* gekennzeichnet, von der keine, eine oder mehrere *Kanten* ausgehen. Die *Kanten* verbinden die *Wurzel* mit ihren *Kindsknoten*, wobei es sich entweder um *Blätter* - d.h. Knoten ohne weiterführende Kanten - oder rekursiv um *Wurzeln* weiterer Bäume handelt. Die *Tiefe T* eines Baumes entspricht der maximalen Anzahl von Kanten, welche durchlaufen werden müssen, um von der Wurzel zu einem Blatt zu gelangen.



## Binärbäume I

**Definition**

Die Wurzel eines *Binärbaumes* besitzt stets einen rechten und einen linken Kindsknoten. Dabei handelt es sich entweder um ein Blatt oder die Wurzel eines weiteren Teilbaumes, welcher wiederum einem *Binärbaum* entspricht. Ist zudem die Entfernung von der Wurzel zu allen Blättern identisch, so spricht man von einem *vollständigen Binärbaum*.



## Binärbäume II

- Ein Knoten mit dem *Index I* kann in einem vollständigen Binärbaum folgende Verbindungsfunktionen ausführen:

$$\begin{array}{lll} child_{right} & (I) = 2I & ;\text{ wenn } I \text{ kein Blatt} \\ child_{left} & (I) = 2I+1 & ;\text{ wenn } I \text{ kein Blatt} \\ parent & (I) = \lfloor I/2 \rfloor & ;\text{ wenn } I \text{ nicht der Wurzel entspricht} \end{array}$$



## k-fache Bäume

- Ziel: Reduzierung des Durchmessers bei gleichbleibender Knotenzahl
- Lösung: Reduzierung der Tiefe
- Umsetzung: Erhöhung der erlaubten Anzahl von Kindern pro Knoten

## Ring-erweiterte Bäume

- Erweiterung des Baumes um horizontale Verbindungsleitungen
- Verbindung der Blattebene mit Hilfe einer Ring-Topologie ergibt *vollständig verknüpften Binärbaum* [HoZ81]
- Durch Anwendung der Ring-Erweiterung auf alle Ebenen entsteht ein *Binärbaum mit vollständigen Ringverbindungen* [HoZ81]



## Hypertree I

**Hamming-Distanz**

Als *Hamming-Distanz H* wird die minimale paarweise *Stellendistanz* eines Codes definiert [Ham86]. Die *Stellendistanz d(x,y)* bezeichnet dabei die Anzahl der Stellen, in denen sich zwei gleich lange Wörter $x$ und $y$ unterscheiden. Für Wörter unterschiedlicher Länge ist die *Stellendistanz* hingegen nicht definiert.



## Hypertree II

- Grundlage bildet ein vollständiger Binärbaum
- Verbindung der Knoten $A$ und $B$ einer Ebene, wenn $H(A, B) = 1$
- Unterscheidung zwischen *Hypertree I* und *Hypertree II* [Goo81]



## Beispiel

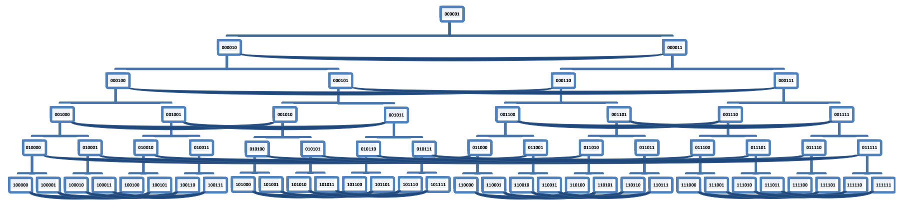

Abbildung: Hypertree mit 63 Knoten veteilt auf sechs Ebenen

## Fat-Tree

- Erhöhung der verfügbaren Datenrate in Richtung der Wurzel [Lei85]
- Beibehaltung des Routingalgorithmus und wesentlicher Eigenschaften der ursprünglichen Baumstruktur



## Agenda

7 Cube
- Klassisches Cube-Netz
- Twisted Cube-Netz
- Crossed Cube-Netz
- Enhanced Hypercube
- Folded Hypercube
- Incomplete Hypercubes



## Klassisches Cube-Netz I

**Hamming-Distanz**

Das klassische Cube-Netz besteht aus $N = 2^n$ Knoten, wobei $n$ auch als *Dimension* bezeichnet wird. Zwei Knoten $A$ und $B$ sind jeweils dann miteinander verbunden, wenn $H(A, B) = 1$.



## Klassisches Cube-Netz II

- Per Definition verfügt in einem klassischen Cube-Netz jeder Knoten über $n$ Ein- und Ausgänge
- Entsprechend ist auch die Implementierung von $n$ Verbindungsfunktionen notwendig:

$$cube_k(b_{n-1}b_{n-2}...b_k...b_1b_0) = b_{n-1}b_{n-2}...\bar{b}_k...b_1b_0, \text{ mit } 0 \leq k < n$$

## Twisted Cube-Netz

- Umstrukturierung vorhandener Verbindungsleitungen zur Reduzierung des Durchmessers
- Vorgehensweise [EsN91]:
  1. Auswahl eines Zyklus über vier Knoten $a, b, c, d \in V$ in einem klassischen Cube-Netz
  2. Bestimmung zweier Kanten $\{a, b\}, \{c, d\} \in E$ aus diesem Zyklus, welche nicht über einen Knoten miteinander verbunden sind
  3. Ersetzung der gewählten Kanten: $\varphi(\{a, b\}, \{c, d\}) = \{a, c\}, \{b, d\}$



## Crossed Cube-Netz I

**Definition**

Als $CQ_n$ wird ein *n-dimensionales Crossed Cube-Netz* [Efe92] bezeichnet. $CQ_1$ besteht aus zwei miteinander verbundenen Knoten. Ist $n > 1$ so wird $CQ_n$ aus den beiden Subnetzen $CQ^0_{n-1}$ und $CQ^1_{n-1}$ gebildet, indem die Knoten $u = 0u_{n-2}...u_0$ und $v = 1v_{n-2}...v_0$ mit $u \in CQ^0_{n-1}$ und $v \in CQ^1_{n-1}$ genau dann miteinander verbunden werden wenn gilt:

1. $u_{n-2} = v_{n-2}$, falls $n$ gerade ist, und
2. $u_{2i+1}u_{2i} \sim v_{2i+1}v_{2i}$, für alle $i$ mit $0 \leq i < \lfloor (n-1)/2 \rfloor$



## Crossed Cube-Netz II

**Paarweise Verwandtschaft**

Zwei binäre Sequenzen $x = x_1x_0$ und $y = y_1y_0$ werden als *paarweise verwandt* bezeichnet, wenn $(x, y) \in \{(00, 00), (10, 10), (01, 11), (11, 01)\}$. Sind $x$ und $y$ paarweise miteinander verwandt so schreibt man $x \sim y$, andernfalls $x \nsim y$.



## Enhanced Hypercube

**Definition**

In einem *Enhanced Hypercube* [TzW91] der *Ordnung* $\omega$, mit $0 \leq \omega < n - 1$, werden gegenüber der klassischen Cube-Topologie Verbindungsleitungen zwischen allen Knotenpaaren

$$\{(x_{n-1} \dots x_{n-\omega}\ x_{n-\omega-1} \dots x_i \dots x_0), (x_{n-1} \dots x_{n-\omega}\ \bar{x}_{n-\omega-1} \dots \bar{x}_i \dots \bar{x}_0)\}$$

hinzugefügt.

## Folded Hypercube

- Spezialfall des Enhanced Hypercubes der Ordnung $\omega = 0$ [EIL91]
- Erweiterung um $N/2$ Verbindungsleitungen zwischen Knoten mit der maximalen Hamming-Distanz
- Reduzierung des Durchmessers auf $\lceil n/2 \rceil$



## Incomplete Hypercubes

- Erweiterung der bisher vorgestellten Cube-Topologien um eine Dimension erfordert eine Verdopplung der Knotenzahl
- *Incomplete Hypercubes* lassen eine beliebige Knotenanzahl zu
- Eingrenzung auf Netze, welche aus zwei vollständigen Cube-Netzen unterschiedlicher Größe $2^n$ und $2^k$ bestehen [TzC92]



## Agenda

8 Literaturverzeichnis



## Literaturverzeichnis I

[IE1596] David B. Gustavson, Qiang Li
The Scalable Coherent Interface (SCI), Santa Clara University 1996.
http://userweb.cs.utexas.edu/users/dburger/teaching/cs395t-s08/papers/9_sci.pdf

[ArL81] B. W. Arden, H. Lee
Analysis of chordal ring networks, Apr 1981.
IEEE Transactions on Computers, Band C-30, Nr. 4, S. 291-295

[Dot84] K. W. Doty
New designs for dense processor interconnection networks, May 1984.
IEEE Transactions on Computers, Band C-33, Nr. 5, S. 447-450

[BoD72] W. J. Bouknight, S. A. Denenberg, D.E. McIntyre, J. M. Rundall, A. H. Sameh, D.L. Slotnick
The Illiac IV system, Apr 1972.
Proceedings of the IEEE, Band 60, Nr. 4, S. 369-388

## Literaturverzeichnis II


- [HoZ81] E. Horowitz, A. Zorat
  The binary tree as an interconnection network: Applications to multiprocessor system and VLSI, Apr 1981
  IEEE Transactions on Computers, Band C-30, Nr. 4, S. 247-253
- [Ham86] R. W. Hamming
  Coding and Information Theory, Jan 1986
  Prentice Hall, ISBN-13: 978-0131390720
- [Goo81] J. R. Goodman, C. H. Sequin
  Hypertree – A multiprocessor interconnection topology, Apr 1981
  Computer Sciences Technical Report #427
- [Lei85] C. E. Leiserson
  Fat-Trees: Universal networks for hardware-efficient supercomputing, Oct 1985
  IEEE Transactions on Computers, Band C-34, Nr. 10, S. 892-901




## Literaturverzeichnis III


- [EsN91] A.-H. Esfahanian, L. M. Ni, B. E. Sagan
  The twisted N-cube with application to multiprocessing, Jan 1991
  IEEE Transactions on Computers, Band C-40, Nr. 1, S. 88-93
- [Efe92] K. Efe
  The crossed cube architecture for parallel computation, Sep 1992
  IEEE Transactions on Parallel and Distributed Systems, Band 3, Nr. 5, S. 513-524
- [TzW91] N.-F. Tzeng, S. Wei
  Enhanced Hypercubes, Mar 1991
  IEEE Transactions on Computers, Band C-40, Nr. 3, S. 284-294
- [ElL91] A. El-Amawy, S. Latifi
  Properties and performance of folded hypercubes, Jan 1991
  IEEE Transactions on Parallel and Distributed Systems, Band 2, Nr. 1, S. 31-42




## Literaturverzeichnis IV


- [Kat88] H. P. Katseff
  Incomplete hypercubes, May 1988
  IEEE Transactions on Computers, Band C-37, Nr. 5, S. 604-608
- [TzC92] N.-F. Tzeng, H.-L. Chen
  An effective approach to the enhancement of incomplete hypercube computers, Feb 1992
  Journal of Parallel and Distributed Computing, Band 14, Nr. 2, S. 163-174