

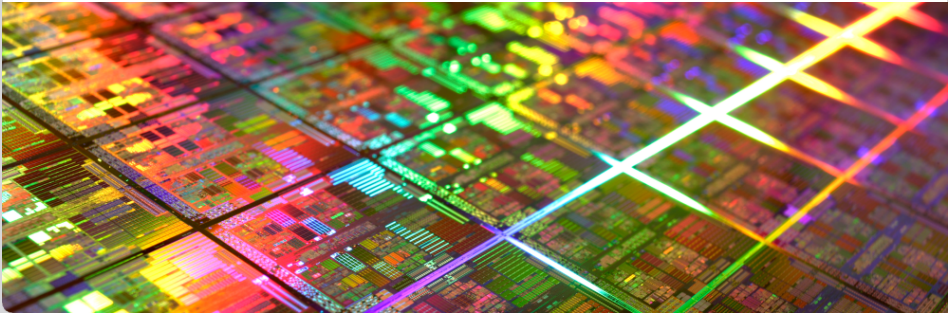
Zentralübung Rechnerstrukturen im SS 2016

Verbindungsstrukturen

Thomas Becker, Prof. Dr. Wolfgang Karl

Lehrstuhl für Rechnerarchitektur und Parallelverarbeitung

30. Juni 2016



- Ermöglichen Kommunikation und Kooperation zwischen Verarbeitungselementen (Knoten)

- Bei verteiltem Speicher:
 - Verbinden physikalisch jeden Knoten für das Versenden von Nachrichten
 - Direkte Send/Receive-Kommunikation zwischen den Knoten

- Bei gemeinsamem Speicher:
 - Ermöglicht Zugriff aller Knoten auf gemeinsamen Speicher
 - Kommunikation durch Lesen/Schreiben auf gemeinsamen Daten

Verbindungsgrad eines Knoten P

Anzahl der Kanten von einem Knoten zu anderen Knoten

Durchmesser (Diameter)

- Maximale Distanz zwischen zwei Knoten
- Maximale Pfadlänge
- Keine Aussage über die realen Leitungslängen

Blockierung

blockierungsfrei, falls jede gewünschte Verbindung unabhängig von schon bestehenden Verbindungen

Erweiterbarkeit

- begrenzt
- stufenweise, z.B. durch Verdoppelung der Knoten
- beliebig

Skalierbarkeit des Verbindungsnetzes

- Fähigkeit, die wesentlichen Eigenschaften des Verbindungsnetzes auch bei beliebiger Erhöhung der Knotenzahl beizubehalten.
- ⇒ Vergrößerung möglich ohne die wesentlichen Eigenschaften des Netzwerks zu verlieren
- **Achtung:** Nicht verwechseln mit der Skalierbarkeit eines Parallelrechners! (vgl. Übung #5, Folie 14)

Minimale Bisektionsbreite:

Schneidet man einen Graphen in zwei gleich große in sich zusammenhängende Teile und betrachtet die Menge der Kanten, die diesen Schnitt kreuzen, so bezeichnet man die Kardinalität der kleinsten Kantenmenge – über alle möglichen Schnitte – als minimale Bisektionsbreite.

Bisektionsbandbreite

Maximale Datenmenge, die das Netzwerk über die Bisektionslinie, die das Netzwerk in zwei Hälften teilt, pro Sekunde transportieren kann.

Übertragungsbandbreite / Durchsatz (bandwidth):

- Die maximale Übertragungsleistung des Verbindungsnetzes oder einzelner Verbindungen
- Meist theoretisch errechnet

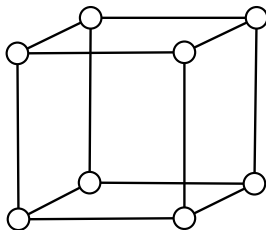
Latenz (Übertragungszeit einer Nachricht)

- SW-Overhead
- Kanalverzögerung
- Schalt-/Routing-Verzögerung (switching/routing delay)
- Blockierungszeit (contention time)

Ausfalltoleranz durch Redundanz

Ein fehlertolerantes Netz muss also zwischen jedem Paar von Knoten mindestens einen zweiten, redundanten Weg bereitstellen.

Die Eigenschaft eines Systems, bei Ausfall einzelner Komponenten unter deren Umgehung funktionstüchtig zu bleiben, wenn auch mit verminderter Leistung, wird als Graceful degradation bezeichnet.



Durchschalte- oder Leitungsvermittlung (circuit switching)

- direkte Verbindung zwischen zwei (oder mehreren) Knoten (ähnlich: analoges Telefonnetz)
- Blockierungsfreie Kommunikation
- Paket benötigt keine Routing-Information
- Kein zusätzlicher Routing-Overhead in jedem Schaltelement
- Kurze Latenzen
- teurer Verbindungsaufbau

⇒ Besonders geeignet für längere Kommunikation

Paketvermittlung (packet switching)

- Datenpakete fester Länge und Nachrichten variabler Länge (ähnlich: „Internet“)
 - Versand über mehrere Knoten hinweg
 - Wegefindungsalgorithmus (Routing) notwendig
 - Nachrichten mit Adresse und Daten
 - Adresse wird in jedem Knoten gelesen und entsprechend weitergeleitet
 - Nur kurze Blockierung einer Leitung
- ⇒ Günstig für kurze Nachrichten und viele Verbindungen

Aufbau eines Pakets

- Header
 - Enthält Routing Information
 - Anforderungs- und Antworttyp (mgwl. auch im Trailer)
- Errorcode
 - Pakete können aufgrund von Fehlern verloren gehen
 - Fehlerbehandlung auf Protokollebene des Verbindungsnetzwerks
- Payload
 - Zu übertragende Daten, irrelevant für Verbindung

- Bewertung der Entwurfsalternativen für Verbindungsnetzwerke bzgl. der Kommunikationsleistung

End-to-end packet latency model

- Betrachtet die Übertragung eines Pakets vom Sender zu einem Empfänger
- Annahme: Paket ist bereit zur Übertragung in einem Puffer an der Quelle
- End-to-end latency bedeutet:
 - Zeit, die benötigt wird von diesem Zeitpunkt an bis das gesamte Paket über das Verbindungsnetzwerk übertragen worden ist und in einem Puffer am Empfängerknoten abgelegt ist
 - N_P : Anzahl der Bits im Payload des Pakets
 - N_E : Anzahl der Bits in Header, Errorcode und Trailer
 - ⇒ $N_P + N_E$: Anzahl der zu übertragenden Bits

- Sender overhead:
 - Zusammenstellen des Pakets und Ablegen in Sendepuffer der Netzwerkschnittstelle
- Time of flight:
 - Zeit, um ein Bit von der Quelle zum Ziel zu senden, wenn Weg festgelegt und konfliktfrei
- Transmission time
 - Zusätzliche Zeit, die benötigt wird, alle Bits eines Pakets zu übertragen, nachdem erstes Bit beim Empfänger angekommen ist
 - Hängt von Linkbandbreite ab
 - Phit (physical transfer unit): Informationseinheit, die in einem Zyklus auf einem Link übertragen wird
 - Für ein Paket gilt : $\frac{N_P + N_E}{N_{Ph} \cdot f}$

- Routing time:
 - Zeit, um den Weg aufzusetzen, bevor ein Teil des Pakets übertragen werden kann

- Switching time:
 - Hängt von der Switching-Strategie ab

- Receiver overhead:
 - Ablegen der Verwaltungsinformation und Weiterleiten des Paktes aus dem Empfangspuffer

- Bestimmt, wie ein Weg in einem Verbindungsnetzwerk aufgebaut und ein Paket von der Quelle zum Ziel übertragen wird
- Modellannahmen:
 - Pfad als Übertragungspipeline
 - Paket umfasst N Phits und überquert L Schaltelemente

■ Modellierung circuit switching

- Aufbau Weg zwischen Quelle und Ziel, danach Übertragung (Pipelining)
- Routing time
 - Zeit, die notwendig ist, ein Phit von der Quelle zum Ziel und zurück zu senden, um die Quelle zu informieren, dass der Weg aufgebaut ist
 - Routing-Entscheidung in einem Schaltelement benötigt R Netzwerkzyklen
 - $routing\ time = L \times R + time\ of\ flight = L \times R + L = L(R + 1)$
- End-to-end packet latency
 - $= sender\ OH + time\ of\ flight + transmission\ time + routing\ time + receiver\ OH =$
 $sender\ OH + L + N + L(R + 1) + receiver\ OH =$
 $sender\ OH + L(R + 2) + N + receiver\ OH$

■ Packet switching store-and-forward Modus

- Jeder Knoten enthält einen Puffer zum Aufnehmen der vollständigen Nachricht
- Pfad wird bestimmt bei Erreichen eines Schaltelements
- Alle Teile müssen angekommen sein, bevor ein Teil weitergeleitet wird
- Time of flight:
 - umfasst Zeit für Übertragung eines Bits von Quelle zu Ziel (ohne Routing OH)
 - Hängt von Paketgröße ab: Paket N Zyklen in Schaltelement
- Übertragungszeit: N Zyklen
- Routing Time: $L \times R$ Zyklen (R Routing OH in jedem Schaltelement)
- End-to-end latency:
 - $= \text{sender OH} + L \times N + N + L \times R + \text{receiver OH} =$
 $\text{sender OH} + N(L + 1) + L \times R + \text{receiver OH}$

■ Packet switching cut-through switching Modus

- Kopfteil der Nachricht enthält Empfängeradresse und wird in jedem Schaltelement dekodiert, um Weg zu bestimmen
- ⇒ Routing OH von R Zyklen
- Solange keine Blockierung, wird Paket durch Schaltelemente gemäß Pipeline-Verarbeitung übertragen
- Kopf-Information wird bis zum letzten Phit festgehalten
- Hat gesamtes Paket ein Schaltelement überquert, wird dieses freigegeben
- Bei Blockierung wird gesamtes Paket aufgehalten
- Flow control unit (Flit): Teil des Pakets, der bei Blockierung aufgehalten wird
 - Bei cut-through gesamtes Paket
- end-to-end latency:
 - $= \text{sender OH} + L + N + L \times R + \text{receiver OH}$
 - $= \text{sender OH} + L(R + 1) + \text{receiver OH}$

■ Packet switching wormhole-routing Modus

- Solange keine Übertragungskanäle blockiert, mit cut-through Modus identisch
 - Falls Kopfteil auf einen belegten Kanal trifft, wird er geblockt
 - Alle nachfolgenden Übertragungseinheiten verharren ebenfalls an ihrer Position, bis Blockierung aufgehoben
- ⇒ Puffer nachfolgender Kanäle für weitere Nachrichten blockiert
- Es werden nur die Phits festgehalten, die die Wegeinformation enthalten, weshalb der Flit kleiner als ein Paket ist

Aufgabe 1 – Latenzmodell

Gegeben sei ein 4×4 Mesh Verbindungsnetzwerk. Nehmen Sie an ein Paket mit der Größe 100 Bytes soll vom linken oberen Knoten des Netzwerks zum rechten unteren Knoten übertragen werden, wobei die Größe eines Phits 10 Bits betrage. Das Verbindungsnetzwerk habe eine Frequenz von 100 MHz und eine Routing-Entscheidung benötige einen Taktzyklus. Gehen Sie außerdem von einem Sender und Receiver Overhead von jeweils 10 ns aus.

- a) **Berechnen Sie die end-to-end Latenz, falls als Switching-Strategie circuit switching verwendet wird.**

Aufgabe 1 – Latenzmodell

- a) Berechnen Sie die end-to-end Latenz, falls als Switching-Strategie circuit switching verwendet wird.

Anzahl Schaltelemente:

3 Schaltelemente nach unten 3 Schaltelemente nach rechts,
insgesamt 6

$$\text{Zykluszeit: } \frac{1}{100 \cdot 10^6 \text{ s}^{-1}} = 10 \text{ ns}$$

Time of flight: (kein switching OH)

$$\text{Anzahl Schaltelemente} \cdot \text{Zykluszeit} = 6 \cdot 10 \text{ ns} = 60 \text{ ns}$$

Transmission time:

$$\text{Anzahl Phits} \cdot \text{Zykluszeit} = \frac{800 \text{ bit}}{10 \text{ bit}} \cdot 10 \text{ ns} = 80 \cdot 10 \text{ ns} = 800 \text{ ns}$$

Routing time:

$$\begin{aligned} &\text{Anzahl Schaltelemente} \cdot \text{Routing OH} + \text{time of flight} \\ &= 6 \cdot 10 \text{ ns} + 60 \text{ ns} = 120 \text{ ns} \end{aligned}$$

Aufgabe 1 – Latenzmodell

- a) Berechnen Sie die end-to-end Latenz, falls als Switching-Strategie circuit switching verwendet wird.

end-to-end Latenz:

Sender OH + time of flight + transmission time + routing time +
Receiver OH =

$$10 \text{ ns} + 60 \text{ ns} + 800 \text{ ns} + 120 \text{ ns} + 10 \text{ ns} = 1000 \text{ ns}$$

Aufgabe 1 – Latenzmodell

b) Gehen Sie nun davon aus, dass als Switching-Strategie packet switching im store-and-forward Modus verwendet wird. Berechnen Sie ebenfalls die end-to-end Latenz.

- Jedes Schaltelement speichert komplettes Paket bevor es weitergeleitet wird
- Route wird nicht vorher festgelegt
- ⇒ Time of flight abhängig von Paketgröße
- ⇒ Routing time nicht mehr von time of flight abhängig

Time of flight:

$$\begin{aligned} & \text{Anzahl Schaltelemente} \cdot \frac{\text{Paketgröße}}{\text{Größe eines Phits}} \cdot \text{Zykluszeit} = \\ & = 6 \cdot 80 \cdot 10 \text{ ns} = 4800 \text{ ns} \end{aligned}$$

Aufgabe 1 – Latenzmodell

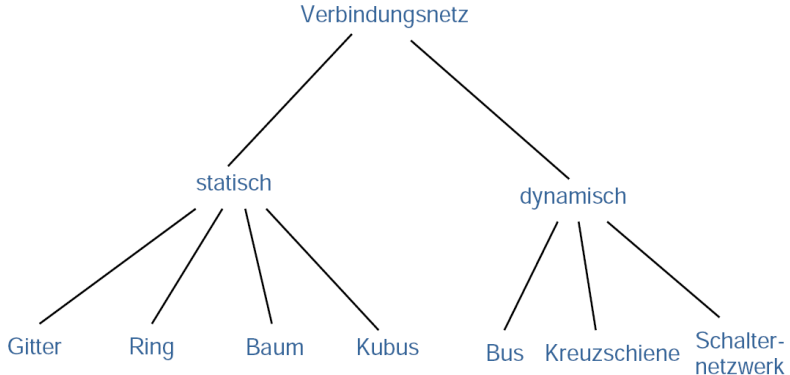
- b) Gehen Sie nun davon aus, dass als Switching-Strategie packet switching im store-and-forward Modus verwendet wird. Berechnen Sie ebenfalls die end-to-end Latenz.

Routing time:

$$\text{Anzahl Schaltelemente} \cdot \text{Routing OH} = 6 \cdot 10 \text{ ns} = 60 \text{ ns}$$

end-to-end Latenz:

$$10 \text{ ns} + 4800 \text{ ns} + 800 \text{ ns} + 60 \text{ ns} + 10 \text{ ns} = 5680 \text{ ns}$$



Statische Verbindungsstrukturen

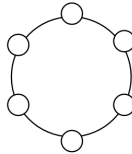
- In statischen Netzen existieren fest installierte Verbindungen zwischen Paaren von Netzknoten
- Steuerung des Verbindungsaufbaus ist Teil der Knoten

Dynamische Verbindungsstrukturen

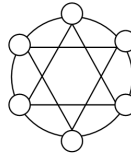
- Dynamische Netze enthalten eine Komponente „Schaltnetz“, an die alle Knoten über Ein- und Ausgänge angeschlossen sind.
- Direkte, fest installierte Verbindungen zwischen den Knoten existieren nicht.
- Alle notwendigen Steuerungsfunktionen sind im Schaltnetz konzentriert

Statische Verbindungsstrukturen

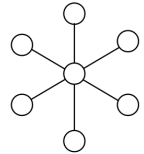
- Kette
- Ring
- Chordaler Ring
- Stern
- Baum
- Fat-Tree
- Gitter



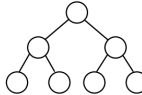
Ring



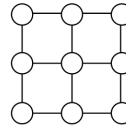
Chordaler Ring



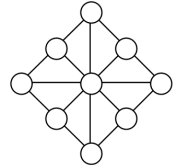
Stern



Baum



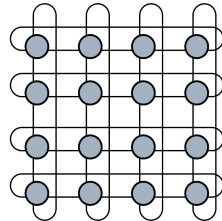
Gitter mit vier
Nachbarknoten



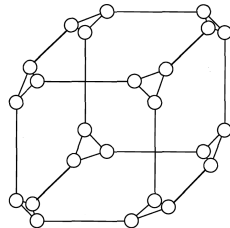
Gitter mit acht
Nachbarknoten

- Torus
- Pyramide
- Würfel
- n-dimensionaler Hyperwürfel
- K-ärer n-Kubus
- Ring-Würfel-Netzwerk
Cube-Connected-Cycle (CCC)

2-dim. Torus



CCC

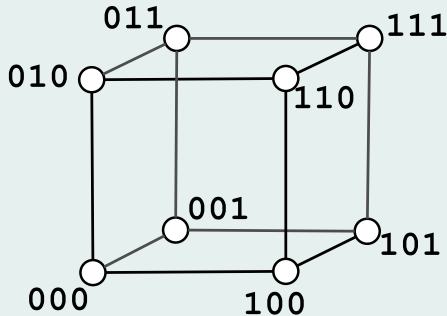


K-ärer n-Kubus (Cubes, Würfel)

- Allgemeine Form eines Kubus-Verbindungsnetzwerkes
- Ringe, Gitter oder Hyperkubi sind eine Teilmenge der Klasse der K-ären n-Kubus-Netzwerke
 - n ist die Dimension
 - Der Radius K ist die Anzahl der Knoten, die einen Zyklus in einer Dimension bilden (Rückwärtskanten)
- Enthält $N = K^n$ Knoten
- Die Knoten werden über eine n -stellige K -äre Zahl der Form $a_0, a_1, \dots, a_{n-1}$ adressiert
 - Jede Stelle a_i mit $0 \leq a_i < K$ stellt die Position des Knotens in der entsprechenden i -ten Dimension dar mit $0 \leq i \leq n-1$
 - Von einem Knoten mit Adresse $a_0, a_1, \dots, a_{n-1}$ kann ein Nachbarknoten in der i -ten Dimension mit $a_0, a_1, \dots, (a_i \pm 1) \bmod K, \dots, a_{n-1}$ erreicht werden
- Knotengrad ist $2n$ und der Diameter ist $n \lfloor \frac{K}{2} \rfloor$

K-ärer n-Kubus – Beispiele

- **K=2, n=3** (hier vereinfacht ohne Rückwärtskanten)
- Adresse: 3-stellige 2-äre (binäre) Zahl $a_0 a_1 a_2$
- a_i mit $0 \leq a_i < 2$



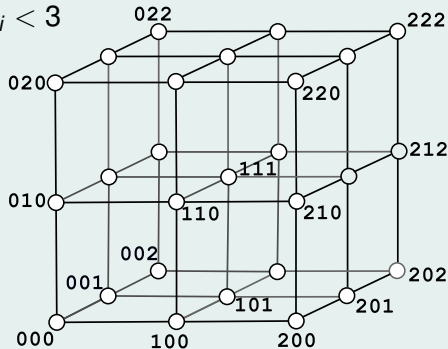
K-ärer n-Kubus – Beispiele

■ **K=3, n=3**

⇒ 3D-Torus (hier vereinfacht ohne Rückwärtskanten)

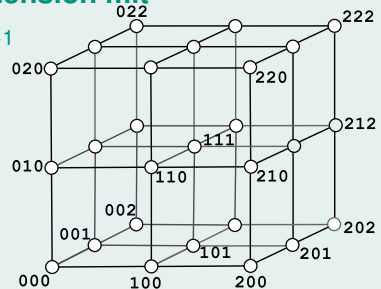
■ Adresse: 3-stellige 3-äre Zahl $a_0 a_1 a_2$

■ a_i mit $0 \leq a_i < 3$



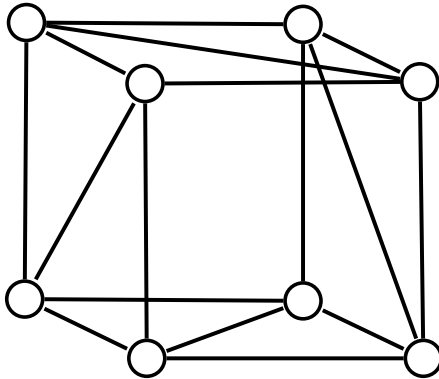
K-ärer n-Kubus – Beispiele

- **K=3, n=3**, 3D-Torus (im Bild vereinfacht o. Rückwärtskanten)
- Adresse: 3-stellige 3-äre Zahl $a_0 a_1 a_2$
- a_i mit $0 \leq a_i < 3$
- Von einem Knoten mit Adresse $a_0, a_1, \dots, a_{n-1}$ kann ein Nachbarknoten in der i -ten Dimension mit $a_0, a_1, \dots, (a_i \pm 1) \bmod k, \dots, a_{n-1}$ erreicht werden
- Von 110 $\Rightarrow$ 100
 $a_1 = 1 \Rightarrow (a_1 - 1) \bmod 3 = 0$
- Von 210 $\Rightarrow$ 010
 $a_0 = 2 \Rightarrow (a_0 + 1) \bmod 3 = 0$



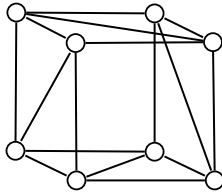
Aufgabe 2 – Statische Verbindungsstrukturen

Gegeben sei ein Verbindungsnetzwerk mit der nachfolgend dargestellten Topologie:



Aufgabe 1 – Statische Verbindungsstrukturen

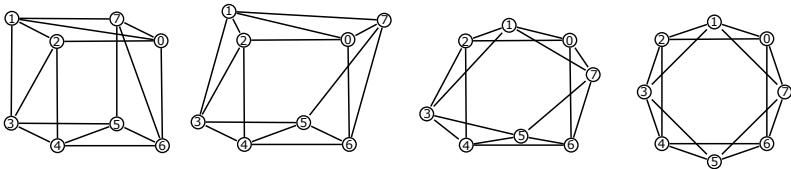
- a) Bestimmen Sie den Verbindungsgrad, den Diameter und die minimale Bisektionsbreite.



Verbindungsgrad: 4
Durchmesser: 2
min. Bisektionsbreite: 6

b) Um welche Art eines Verbindungsnetzwerkes handelt es sich in diesem Fall?

Chordaler Ring mit Knotengrad 4



c) **Liegt Redundanz vor? Wenn ja, wieviele Verbindungsleitungen können ausfallen bevor eine Verbindung zwischen zwei beliebigen Knoten nicht mehr geschaltet werden kann?**

- Es liegt Redundanz vor.
- Da der Verbindungsgrad jedes Knotens 4 ist und bidirektionale Leitungen verwendet werden, können bis zu drei Leitungen ausfallen und dennoch jeder Knoten von einem anderen erreicht werden.
Anmerkung: Hier ist die minimale Anzahl von Kanten gesucht, die ausfallen dürfen, bevor ein Knoten nicht mehr erreichbar ist.
- Allerdings kann beim Ausfall einer Kante der Durchmesser steigen, das heißt es könnten längere Wege notwendig sein.

- d) **Vergleichen Sie diese Netzwerktopologie mit den Topologien (unidirektionaler) Ring, 2D-Gitter, (binärer) Baum und Hyperkubus in den Punkten Verbindungsgrad, Durchmesser und minimaler Bisektionsbreite.**

$N = \# \text{ Knoten}$

	Aufgabe a)	Ring	2D-Gitter	(binärer) Baum	(n-dim) Hyperkubus
Knotenzahl	N	N	$N = n^2$	N	$N = 2^n$
Verbindungsgrad	4	2	$(2 -) 4$	$(1 -) 3$	$\log_2 N = n$
Durchmesser	$\lfloor \sqrt{N} \rfloor$	$N/2$	$2(n - 1)$	$2(\lceil \log_2 N \rceil - 1)$	$\log_2 N = n$
min. Bisektionsbreite	6	2	n	1	$2^{n-1} = N/2$

e) **Lange Zeit war ein Hyperkubus die häufigste Verbindungsstruktur bei nachrichtengekoppelten Multiprozessorsystemen. Wie viele Knoten müssen bei einem Hyperkubus für eine Erweiterung hinzugefügt werden? Was stellen Sie dabei für den Verbindungsgrad fest und was hat das für Auswirkungen auf den Aufbau und die Erweiterbarkeit des Rechners?**

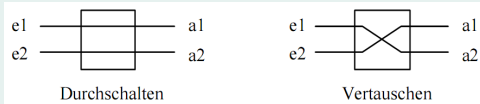
- Jede Erweiterung benötigt eine Verdopplung der Prozessorenanzahl ($N = 2^n$)
- Der Verbindungsgrad der Knoten steigt bei jeder Erweiterung um 1
- Rechner sind deshalb aus räumlichen Anordnungsgründen begrenzt

- **Bus**, Mehrfachbus

- **Kreuzschienenverteiler** (Crossbar Switch)

Alle angeschlossenen Prozessoren und Speicher können paarweise disjunkt gleichzeitig und blockierungsfrei miteinander kommunizieren.

- **Schalternetzwerke** aus Zweierschaltern



- **Permutationsnetze**

■ Permutationsnetze

- p Eingänge des Netzes können gleichzeitig auf p Ausgänge geschaltet werden

⇒ Es wird eine Permutation der Eingänge erzeugt

- **Einstufige und mehrstufige Permutationsnetze** enthalten eine bzw. mehrere Spalten von Zweierschaltern

■ Reguläre Permutationsnetzwerke

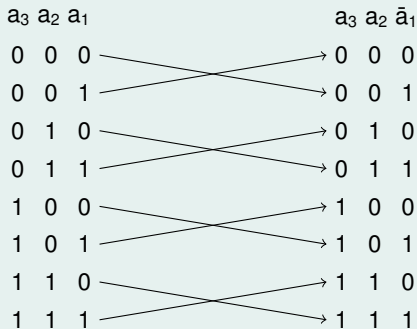
- p Eingänge
- p Ausgänge
- k Stufen mit je $p/2$ Zwischenschaltern
- p normalerweise eine Zweierpotenz

- **Irreguläre Permutationsnetzwerke** weisen gegenüber der regulären Struktur Lücken auf

Tauschpermutation T

Negation des niedrigwertigsten Adressbits

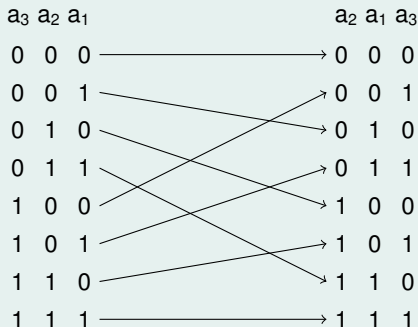
$$T(a_n, a_{n-1}, \dots, a_2, a_1) = (a_n, a_{n-1}, \dots, a_2, \bar{a}_1)$$



Mischpermutation M (Perfect Shuffle)

Kreisverschiebung der Adressbits

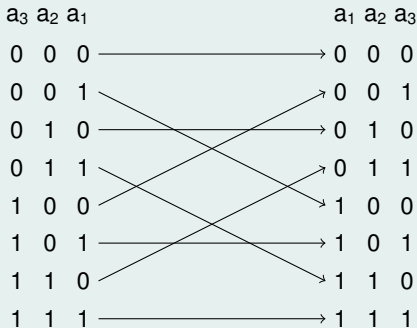
$$M(a_n, a_{n-1}, \dots, a_2, a_1) = (a_{n-1}, \dots, a_2, a_1, a_n)$$



Kreuzpermutation K (Butterfly)

Vertauschen des hochwertigsten mit dem niedrigwertigsten Adressbit

$$K(a_n, a_{n-1}, \dots, a_2, a_1) = (a_1, a_{n-1}, \dots, a_3, a_2, a_n)$$



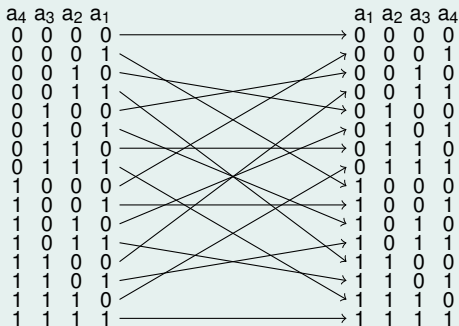
Umkehrpermutation U

Spiegelung aller Adreßbits um die Mitte der Adressbitfolge:

$$U(a_n, a_{n-1}, \dots, a_2, a_1) = (a_1, a_2, \dots, a_{n-1}, a_n)$$

- ⇒ Für $n = 2$ und $n = 3$ ergibt sich dasselbe Grundmuster wie bei der Kreuzpermutation!
- Für $n \geq 4$ unterscheiden sich Umkehr- und Kreuzpermutation

Umkehrpermutation U für $n = 4$

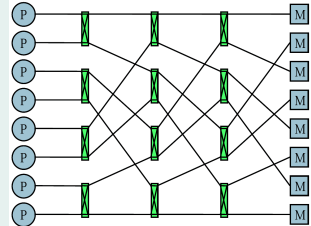


Mehrstufige Permutationsnetzwerke:

- jeweils aus einem bestimmten Grundmuster aufgebaut
- oft mit einer der eben vorgestellten Permutationen
- statt Zweierschalter auch vollwertige Crossbar-Switches als Schaltelemente

Beispiele:

- Omega-Netzwerk
 - Mischpermutation
- Switching-Banyan-Netzwerk
 - Kreuzpermutation
- Benes-Netzwerk
 - rekursiver Aufbau

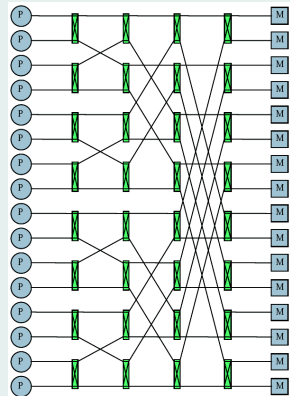


Mehrstufige Permutationsnetzwerke:

- jeweils aus einem bestimmten Grundmuster aufgebaut
- oft mit einer der eben vorgestellten Permutationen
- statt Zweierschalter auch vollwertige Crossbar-Switche als Schaltelemente

Beispiele:

- Omega-Netzwerk
 - Mischpermutation
- Switching-Banyan-Netzwerk
 - Kreuzpermutation
- Benes-Netzwerk
 - rekursiver Aufbau



Mehrstufige Permutationsnetzwerke:

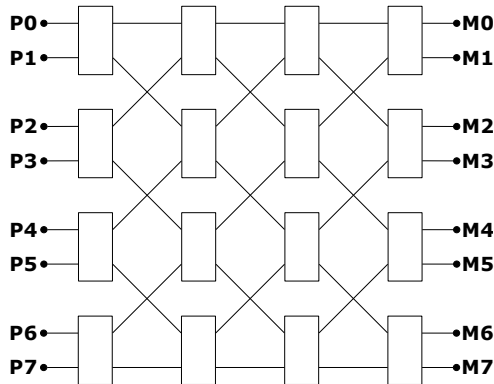
- jeweils aus einem bestimmten Grundmuster aufgebaut
- oft mit einer der eben vorgestellten Permutationen
- statt Zweierschalter auch vollwertige Crossbar-Switche als Schaltelemente

Beispiele:

- Omega-Netzwerk
 - Mischpermutation
- Switching-Banyan-Netzwerk
 - Kreuzpermutation
- Benes-Netzwerk
 - rekursiver Aufbau

Aufgabe 3 – Dyn. Verbindungsstrukturen

Gegeben sei ein dynamisches Verbindungsnetzwerk, das 8 Prozessoren (P0 – P7) mit 8 Speichern (M0 – M7) wie folgt über einen Verbund von Zweierschaltern verbindet:



Aufgabe 3 – Dyn. Verbindungsstrukturen

- a) **Kann zwischen jedem Prozessor- und Speicherpaar eine Verbindung hergestellt werden?**

Ja!

Hier war nicht nach gleichzeitig möglichen Verbindungen gefragt (vgl. Teilaufgabe b über alle Permutationen)

b) Kann jede Permutation generiert werden? Begründen Sie Ihre Antwort!

Nein! Beweis durch Widerspruch.

Annahme: jede Permutation kann generiert werden.

Gesucht: mindestens eine Permutation, für die die Annahme nicht gilt.

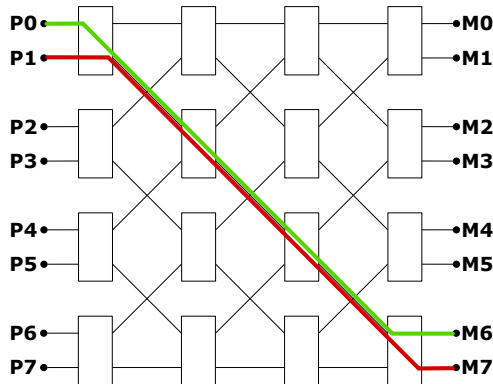
- Bei einer paarweisen Mischpermutation (Kreisverschiebung), hier also Verbindung von P0 und P1 mit M6 bzw. M7 gibt es nur einen möglichen Verbindungsweg, der gleichzeitig für beide Verbindungen benutzt werden müßte

⇒ Blockierung

Aufgabe 3 – Dyn. Verbindungsstrukturen

- c) Was ist die minimale Verbindungszahl ab der eine Blockierung auftritt? Geben Sie ein Beispiel an.

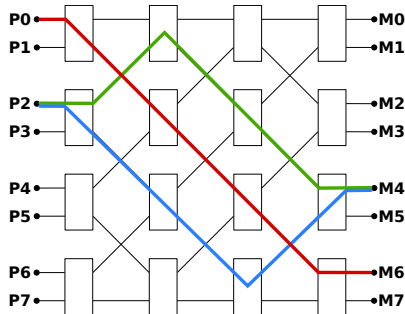
Schon bei zwei Verbindungen kann eine Blockierung auftreten:
z.B. bei $P0 \rightarrow M6$ und $P1 \rightarrow M7$



d) Ist das Netzwerk redundant? Begründen Sie Ihre Antwort.

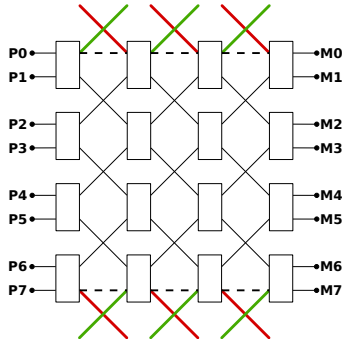
Nein!

Auf der einen Seite gibt es für bestimmte Paare mehrere Möglichkeiten (vgl. $P2 \rightarrow M4$), aber ebenso gibt es Paare, bei denen schon der Ausfall einer Verbindung die Weiterleitung ausschließt (z.B. $P0 \rightarrow M6$).



■ Wie könnten manche der Problem mit diesem Netzwerk vermeiden/verringert werden?

⇒ Verbindungen vom oberen Rand nach unten (vgl. n-dim Torus)



■ Achtung, durch das Weglassen der gestrichelten Verbindungen wäre z.B. auch keine Verbindung von P0 zu M0 mehr möglich!

Motivation

- Top500-Liste
- Welchen Rechner kaufen?
- Was für eine Leistung ist notwendig?
- Unterstützte Programmiermodelle
- Skalierbarkeit
- Abhängigkeiten vom Verbindungsnetz

FLOPS

$$\text{MFLOPS} = \frac{\text{Anzahl der ausgeführten Gleitkommainstruktionen}}{10^6 \times \text{Ausführungszeit}}$$

- Maßzahl für die Operationsleistung (Gleitkomma-Verarbeitung)
- MFLOPS, GFLOPS, TFLOPS, PFLOPS,...

Aussagekraft der Top 500-Liste

- Leistungsfähigkeit von vielen Faktoren abhängig
 - LINPACK-Benchmark (optimiert für Listenplatz)
 - Rechenleistung $\Leftrightarrow$ Energieverbrauch
 - Auslastung im regulären Betrieb
 - Programmierbarkeit
 - Real zu berechnende Probleme
 - Optimierung für bestimmte Berechnungen

Frage

- Welches ist der bessere Rechner?

⇒ **Oft keine allgemeingültige Antwort möglich!**

Aufgabe 4 – Leistungsmessung

In der 2x jährlich erscheinenden Top500-Liste werden jeweils die zum Zeitpunkt der Veröffentlichung 500 schnellsten Rechner der Welt aufgelistet. Zur Bestimmung der Rechenleistung wird dafür auf den Systemen der High-Performance LINPACK Benchmark ausgeführt.

- a) **Wie aussagekräftig sind die Ergebnisse der Messungen?**
- b) **Was ist der Nachteil der Leistungsbestimmung mit dem LINPACK Benchmark?**
- c) **Welche Benchmarks werden ebenfalls für die Leistungsmessung von Supercomputern verwendet? Und welche sind dabei von zunehmendem Interesse?**
- d) **Gibt es noch weitere Listen mit Top-Rechnern?**

a) Wie aussagekräftig sind die Ergebnisse der Messungen?

- Wert ausschließlich von LINPACK-Benchmark
- Optimierung der Systeme für genau diesen Benchmark und die Top-Position der Liste
- Länder-/Kontinent-Rivalität
- Keine Aussage über reale Nutzbarkeit / Programmierbarkeit
- In Realität oft andere Anforderungen an Systeme:
 - Hoher Durchsatz an Daten, geringe Latenz
 - Spezialberechnungen mit Beschleunigern oder optimierten Prozessoren
 - Einfache und flexible Nutzbarkeit
 - Aufwand für Optimierung
 - Hoher Durchsatz an Jobs

b) Was ist der Nachteil der Leistungsbestimmung mit dem LINPACK Benchmark?

Nachteile:

- LINPACK Benchmark ist > 37 Jahre alt (Top500 21,5 Jahre)
- Schwerpunkt auf Floating-Point-Operationen ($O(n^3)$), Datenbewegungen ($O(n^2)$)
- Entspricht immer weniger heutigen realen Anwendungen
- Werte sind Peak-FLOPS-Werte (normal nur 1/2 oder 2/3 von Maximum)
- Beschränkt Einsatz von neuen Architekturen
- Nutzbarkeit des Systems wird nicht gemessen
- Marketing-Tool
- Aussage über Rechner nur anhand einer Zahl

Quelle: „HPCG: One Year Later“, <https://software.sandia.gov/hpcg/>

b) Was ist der Nachteil der Leistungsbestimmung mit dem LINPACK Benchmark?

Sehr nachteiliges:

- Testet nicht die gesamte Architektur sondern nur einen Teilaspekt
- Beschränkt die Technologie- und Architekturmöglichkeiten für HPC-Systementwickler
- ⇒ Ausrichtung der Entwicklung für diesen Benchmark
- Benchmarks über Floating-Point-Berechnungen sind immer weniger aussagekräftig
- Datenintensive Tasks nehmen immer mehr zu

Quelle: „HPCG: One Year Later“, <https://software.sandia.gov/hpcg/>

c) **Welche Benchmarks werden ebenfalls für die Leistungsmessung von Supercomputern verwendet? Und welche sind dabei von zunehmendem Interesse?**

- Graph 500: Big Data Computing
Cybersecurity, Medical Informatics, Data Enrichment, Social Networks, and Symbolic Networks
- HPCG: High Performance Conjugate Gradient
löst $Ax = b$, große lineare Gleichungssysteme
Verschiedene Kommunikationspattern, kollektive Operationen, Speicherbandbreite, ...
- HPC Challenge: Verschiedene Benchmarks
- Livermore Loops
- NAS Parallel Benchmarks
- Dhrystone, Whetstone
- SPEC-hpc
- ...

d) Gibt es noch weitere Listen mit Top-Rechnern?

- Green 500: Energy-Aware HPC
<http://www.green500.org/>
- Graph 500: Big Data Computing
<http://www.graph500.org/>
- Green Graph 500: Energy-Aware Big Data Computing
<http://green.graph500.org/>
- Top500 HP-LINPACK vs. HPCG
<https://software.sandia.gov/hpcg/>
- ...

Zu vergleichende Parallelrechner:

- JUGENE BlueGene/P in Jülich
 - siehe Foliensatz 8, Folien 2-5 ff
 - 825,5 TFLOPS, 294.912 Prozessoren (bzw. CPU-Kerne)

- HP XC6000 am KIT
 - 1,9 TFLOPS, 282 Prozessoren
 - Netzwerk siehe Foliensatz 5, Folien 2-34 ff

a) **Wieviel GFLOPS trägt jeder einzelne CPU-Kern zur theoretischen Spitzenleistung bei?**

■ JUGENE BlueGene/P:

$825.500 \text{ GFLOPS} / 294.912 \text{ Kerne} = 2,80 \text{ GFLOPS/Kern}$

■ HP XC6000:

$1.900 \text{ GFLOPS} / (101 * 2 + 10 * 8) \text{ Proz} = 6,74 \text{ GFLOPS/Proz}$

■ **Achtung, dies sind sehr theoretische und vereinfachte Werte!**

b) Was für ein Netzwerktyp/-struktur wird verwendet?
(Topologie, Hersteller, statisches oder dynamisches Netz,...)

■ JUGENE BlueGene/P:

3-dimensionaler Torus, Eigenentwicklung von IBM, statisches Netz

■ HP XC6000:

Fat-Tree (Baumstruktur), Quadrics QsNet II Interconnect, dynamisches Netz, Rechnerknoten sind nicht im Netzwerk auf verschiedenen Ebenen verteilt

c) Wie groß ist der Durchmesser, d.h. die längste Verbindung zwischen zwei Knoten?

■ JUGENE BlueGene/P:

Kantenlänge eines 3-dim. Würfels: $\sqrt[3]{294912} \approx 67$

⇒ Durchmesser von 3D-Torus: $3 * 67 / 2 \approx 100$

Bei einem Torus wird im Vergleich zu einem Gitter der Durchmesser auf Grund der Rückwärtskanten halbiert.

Achtung, dies ist eine Schätzung ohne Berücksichtigung des tatsächlichen Netzwerkaufbaus!

■ HP XC6000:

Aufsteigen im Baum bis zur Wurzel und zurück: 4

d) **Vergleichen Sie Bandbreite, Latenz und Blockierungsfreiheit der beiden Netzwerke.**

■ JUGENE BlueGene/P:

Netzwerk ist nicht blockierungsfrei, Bandbreitenengpässe können auftreten, die Latenz ist unterschiedlich je nach Verbindung

■ HP XC6000:

Netzwerk ist blockierungsfrei, Bandbreite von mehr als 800 MB/s, geringe Latenz

e) Gibt es einen Flaschenhals?

- JUGENE BlueGene/P:
Prinzipiell nein.
Je nach Wegwahlverfahren können aber Probleme auftreten.
- HP XC6000:
Nein, da ein „Dynamic Fat-Tree“ verwendet wird, bei dem jede Permutation geschaltet werden kann

- f) **Bewerten Sie die Skalierbarkeit und Erweiterbarkeit der beiden Netzwerkvarianten.**
- JUGENE BlueGene/P:
Sehr gut, das Netzwerk kann einfach um eine Ebene erweitert werden, prinzipiell unbeschränkt
 - HP XC6000:
Sehr schlecht, erweiterbar um jeweils eine 2-er-Potenz, maximal 4096 angeschlossene Rechenknoten, d.h. maximal ≈ 40000 CPUs je nach Rechenknoten

g) Nehmen Sie an, die Prozessorenzahl des HP XC6000 würde an die Größenordnung der Prozessorenzahl des BlueGene/P angepasst. Welches Problem hinsichtlich der Netzwerkkommunikation ergibt sich hierbei? Insbesondere welche Veränderungen am Netzwerk müssten durchgeführt werden, damit es die Anforderungen hinsichtlich Blockierungsfreiheit weiterhin erfüllt?

- Netzwerk besteht aus drei Schichten miteinander verknüpfter Switches
- Ebene 1 - Switche haben genausoviel Verbindungen zur nächsten Ebene wie angebundene Rechenknoten
- In jeder Ebene nimmt die Portzahl quadratisch zu
- ⇒ Netzwerkgröße limitiert durch die Größe (Portzahl) der Switche
- ⇒ Nach Erweiterung auf ~ 200000 Prozessoren müssten Switche mit mehr als 20000 Port verwendet werden.

h) Welche Vereinfachungen im Netzwerk könnten gemacht werden, um den Aufwand für Netzwerkhardware zu verringern und was wären die Auswirkungen hiervon?

- Ausdünnung der Verbindungen in Richtung Wurzel des Baumes
 - ⇒ Keine Blockierungsfreiheit mehr
 - ⇒ Durchsatz verringert sich
 - ⇒ Redundanz geringer
- Intelligente Routing-Verfahren
- Analyse und Anpassung an bestimmte Kommunikationspattern

a) **Warum sind vollständige statische Verbindungsnetze nicht praktikabel in Parallelrechnern?**

Antwort:

Bei vollständigen statischen Verbindungsnetzen ist jeder Knoten direkt mit jedem anderen Knoten verbunden, weshalb die Netzwerkkosten quadratisch mit der Anzahl der Knoten steigt.

- c) Geben Sie die allgemeine Formel zur Berechnung der Anzahl der für ein Omega-Netzwerk nötigen Crossbar Switches mit dem Switching-Grad k (k entspricht der Anzahl der Ein-/Ausgänge) abhängig von der Knotenzahl N an.

Formel:

$$\text{Anzahl Switches} = \frac{N \cdot \log_k N}{k}$$

- d) **Nennen Sie zwei Gründe, warum sich die Durchschalte-/Leitungsvermittlung nicht für Systeme mit kurzen Nachrichten und einer hohen Netzauslastung eignet.**
- Teurer Verbindungsaufbau, der sich bei kurzen Nachrichten nicht amortisiert
 - Blockiert Leitungen während der kompletten Kommunikation, was bei hoher Netzauslastung zu langen Wartezeiten führen kann

d) Was ist der Hauptunterschied zwischen den Übertragungsmodi „store and forward“ und „cut through“?

Antwort:

Beim „store and forward“-Modus wird die Nachricht in jedem Zwischenknoten zunächst komplett in Empfang genommen und vollständig zwischengespeichert. Erst wenn die komplette Nachricht erhalten und gespeichert wurde, wird sie an den nächsten Knoten weiter übertragen. Beim „cut through“-Modus ist dies nicht der Fall. Hier können die einzelnen Teile der Nachricht direkt an den nächsten Knoten weitergeleitet werden.

a) **Aus welchen Bestandteilen setzt sich die Übertragungszeit T_{msg} einer Nachricht zusammen?**

Die Übertragungszeit setzt sich zusammen aus:

- Startzeit t_s , also der Zeit, die benötigt wird um die Kommunikation zu initiieren
- Transferzeit t_w , also der Zeit, die benötigt wird um die Datenwörter physikalisch zu übertragen

ebenfalls korrekt:

- Kanalverzögerung
- Schalt-/Routing-Verzögerung
- Blockierungszeit

d) **Nennen Sie jeweils zwei statische und dynamische Verbindungsstrukturen.**

Statisch:

- Ring
- Baum
- Torus
- ...

Dynamisch:

- Bus
- Kreuzschiene
- Schaltnetzwerk
- ...

e) **Nennen Sie zwei Vermittlungstypen des Datentransfers für Verbindungsstrukturen.**

- Durchschalte- oder Leitungsvermittlung
- Paketvermittlung

Übung #7 – 14.07.2016

- **Cache-Kohärenz**

Übung #8 – 21.07.2016

- **Vektorrechner**
- Klausuraufgaben
- ...

Zentralübung Rechnerstrukturen im SS 2016

Verbindungsstrukturen

Thomas Becker, Prof. Dr. Wolfgang Karl

Lehrstuhl für Rechnerarchitektur und Parallelverarbeitung

30. Juni 2016

