

# Kapitel 4

# Rechnerinterne Kommunikationssysteme

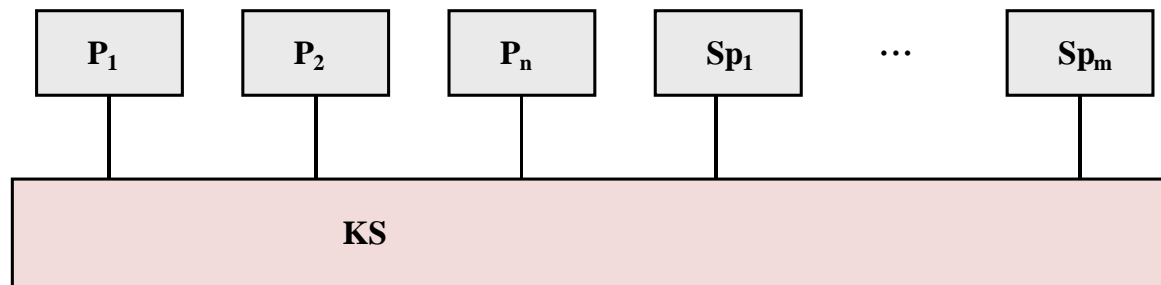
- |     |   |    |
|-----|---|----|
| 4.1 | Einsatzgebiete, grundsätzliche Probleme und Lösungsansätze bei rechnerinterner Datenübertragung | 1  |
| 4.2 | Infrastrukturen für rechnerinterne Kommunikation  | 7  |
| 4.3 | Architekturen von Vermittlungsrechnern  | 23 |

# 4. Rechnerinterne Kommunikationssysteme

## 4.1 Einsatzgebiete, grundsätzliche Probleme und Lösungsansätze bei rechnerinterner Datenübertragung

Zur Bedeutung der rechnerinternen DÜ :

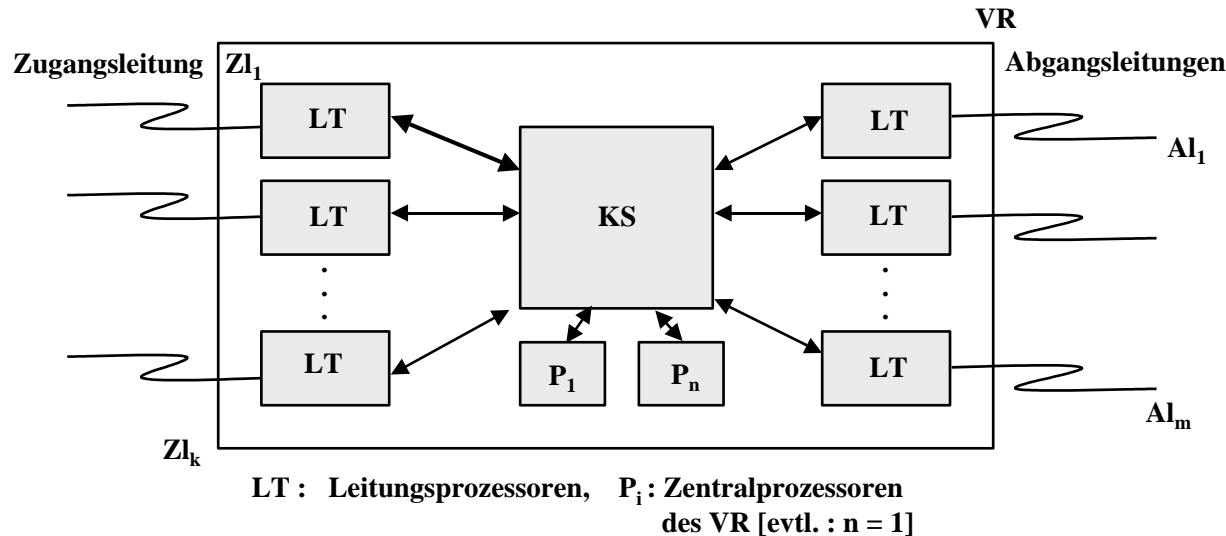
- relevant für alle **Rechensysteme**, z.B.
  - Register-Register-Transfers,
  - Kommunikation CPU  $\leftrightarrow$  Cache/HSP (Hauptspeicher) : der “*von Neumann-Flaschenhals (bottleneck)*”
- besonders relevant für
  - **Multiprozessorsysteme/Parallelrechner** :



$P_i$ : Prozessoren     $Sp_i$ : Speichermoduln    KS: (rechnerinternes) Kommunikationssystem

# Rechnerinterne DÜ (Fortsetzung)

## - Vermittlungsrechner (VR) :



- im allg. :  $k = m$
- $Al_i = Zl_i$  bei halb- oder vollduplex-Leitungen
- Aufgabe von Vermittlungsrechner
  - bei *Leitungsvermittlung/Durchschaltetechnik* sowie bei *Paketvermittlung mit virtuellen Verbindungen* (vgl. GSS-Vorl.?!): eingehende Daten einer physikal. oder virtuellen Verbindung  $Zl_i$  allesamt auf  $Al_j$ ,  $j=j(i)$ , abzusenden
  - bei *Datagrammtechnik* (vgl. GSS-Vorl.?!): evtl. Weiterleitung der DEen (Pakete) abhängig von Routing-Entscheidung

# Probleme und Anforderungen an rechnerinterne DÜ :

- hoher Durchsatz gefordert (insbesondere bei Kommunikation zwischen Zentral- bzw. E/A-Prozessoren und Hsp. sowie in schnellen Vermittlungsrechnern → vgl. ATM-Switches)
- geringe Verzögerungszeit (z.B. resultierend aus Wartezeit bei Zugriff auf gemeinsames ÜM)
- relativ geringe Komplexität → VLSI-Realisierbarkeit, hohe Leistungsanforderungen (s.o.)
- keine aufwändige Wegeermittlung (z.B. alle Dateneinheiten bei  $Zl_i$  für gegebenes Ziel  $x$  über  $Al_x$  weitergeleitet)
- Einsatz von Standardlösungen (insbesondere bei Kommunikation Zentraleinheit ↔ E/A-Subsystem, bei Systembussen, etc.)
- möglichst hohe Verfügbarkeit des rechnerinternen Kommunikationssystems

# Vereinfachungen gegenüber DÜ in Rechnernetzen :

- typischerweise *weniger Signalverfälschungen* bei rechner-interner DÜ (aber : in LANs/WANs → optische DÜ !)
- rechnerintern → *sehr geringe Signallaufzeiten* (häufig vernachlässigbar)
- *parallele DÜ* und somit in der Regel schneller
- *zentralisierte Zugriffskontrolle* bei gemeinsamen ÜM eher akzeptabel als in Rechnernetzen, überdies : spezielles Kommunikationssystem zur Realisierung der Zugriffskontrolle im allgemeinen einsetzbar (z.B. Steuerbus)

# Kommunizierende Komponenten bei rechnerinterner Datenübertragung und übliche Topologieformen

- **Kommunizierende Komponenten**  
(auf PMS-“processor-memory-switch“ –Ebene) :
  - **Prozessoren** (Zentral ~, E/A ~, Leitungs ~)
  - **Speichermodule** (Hauptspeichermodule, Cache)
  - **LAN-Adapter**  
o.ä.
- **Topologieformen**, u.a.:
  - (a) **individuelle Leitungen** → z.B. Punkt-zu-Punkt-Verbindungen oder vollständige Vermaschung
  - (b) **(Schalt-) Stern** → z.B. Kreuzschienenverteiler, Banyan-Netze
  - (c) **Bus** → z.B. als Systembus
  - (d) **Ring** → z.B. zum Anschluss von Peripherie an Rechnern
  - (e) **gemeinsamer Speicher** → z.B. in Multiprozessorsystem
  - (f) **Hypercube/Hyperwürfel** → z.B. als Kommunikationsinfrastruktur in Parallelrechner



## 4.2 Infrastrukturen für rechnerinterne Kommunikation

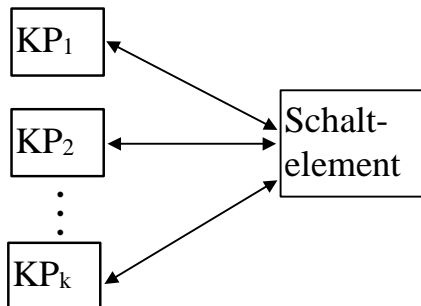
- Verfeinerte Darstellung zu **(Schalt-) Stern**  
wesentliches Merkmal für Schaltstern :  
*zentrales Schaltelement*



**Vorteile** des Schaltsterns:

- weitgehend triviale Zugriffskontrolle
- einfache Adressierung, triviale Wegeermittlung

**Grundsätzlicher Aufbau** des Schaltsterns:

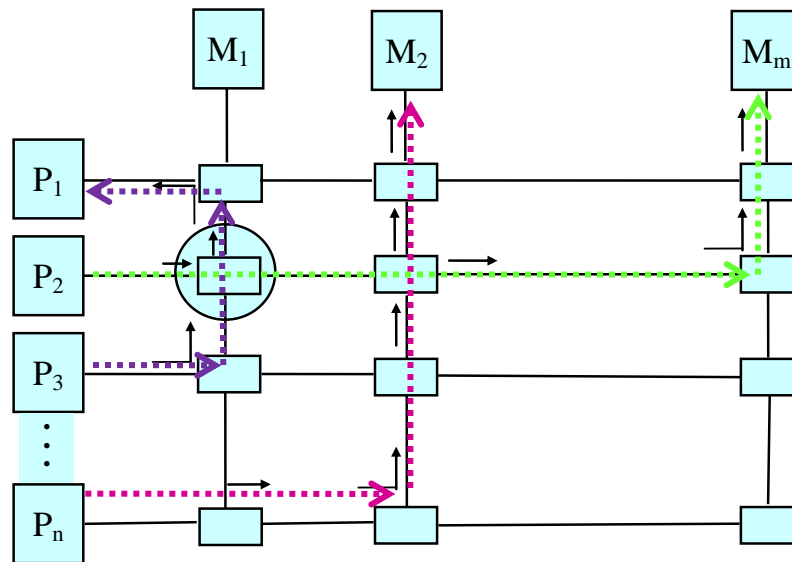


KP: *Kommunikationspartner*  
(z.B. Prozessoren, Speicher-  
moduln, Rechner)

# Realisierung von Raummultiplex-Schaltsternen

**Beispiel B1 : Kreuzschienenverteiler** (z.B. in Multiprozessorsystemen)

→ vgl. Abschn. 4.3



P ≡ Prozessor  
(z.B. CPU,  
E/A-Proz.)

M ≡ Speichermodul  
*nota bene:* P/M ≡ Leitungen  
in Abschn.4.3

Exist. Kommunikations-  
beziehungen:

- $P_1 \leftrightarrow P_3$
- $P_2 \leftrightarrow M_m$
- $P_n \leftrightarrow M_2$

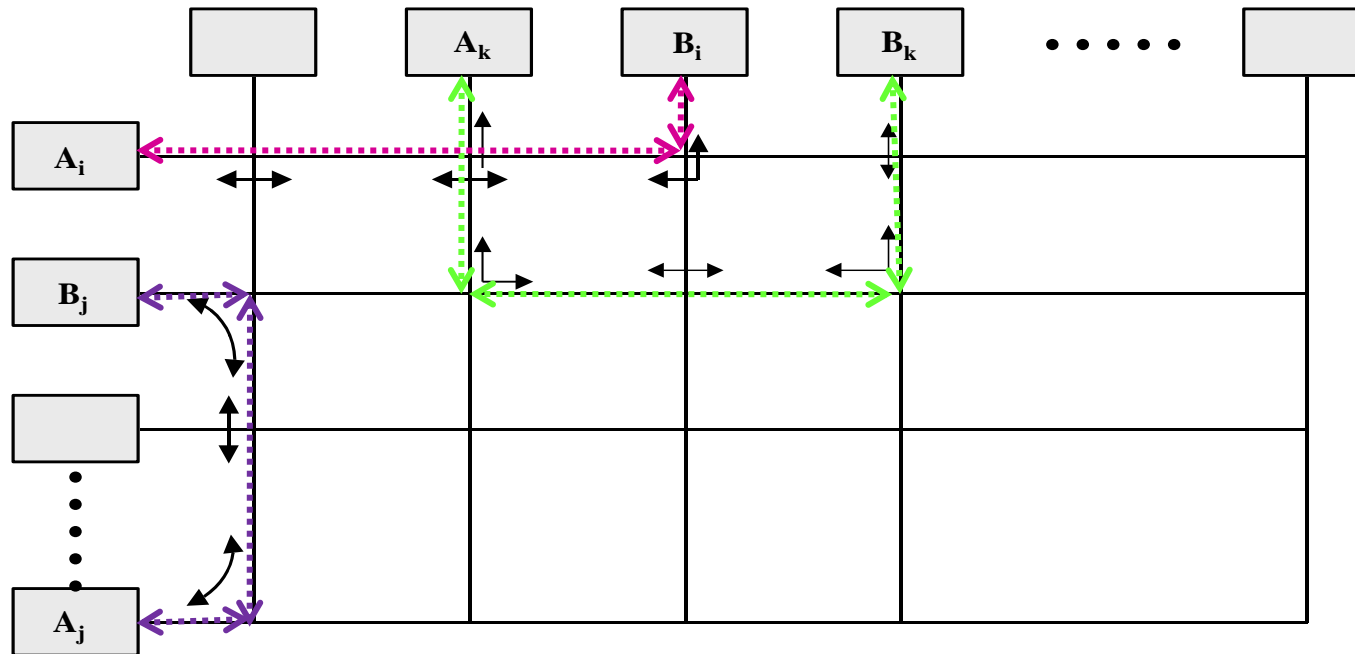


Koppeleinrichtung (Verfeinerung von ):

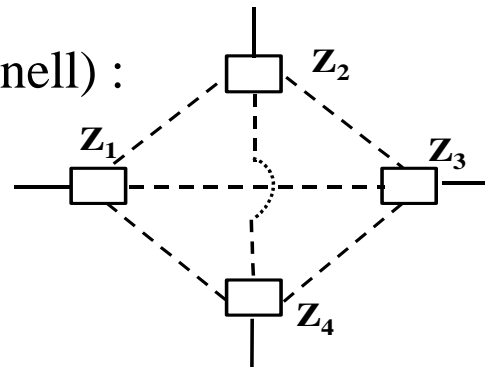
siehe Verfeinerung der Kreuzungspunkte (konzeptionell) auf nächster Folie



# Der Kreuzschienenverteiler / CROSSBAR-SWITCH (Forts.)



**Kreuzungspunkte** (konzeptionell) :



⇒ Bedingung für Nutzung eines Kreuzungspunktes :  
 nur max. 1 Datenstrom (für gegebenes Sender-Empfänger-Paar)  
 fließt über jeden der Zugangspunkte  $Z_1, \dots, Z_4$

## Vorteile Kreuzschienenverteiler :

- nach Festlegung des, sodann festen, Datenflusses für sämtliche Kreuzungspunkte : Wegeermittlung gelöst (solange keine Neufestlegung getroffen wird); ABER : ggf. temporäre Blockierung von Pfaden
- Existenz zahlreicher paralleler Pfade, allerdings nur sehr eingeschränkte Ausfalltoleranz, jedoch insgesamt hoher Durchsatz; ABER: nur 1 Ein-/Ausgang pro Kommunikationspartner (-modul) → ist nicht notwendigerweise ein Nachteil, z.B. nicht für Komponenten mit „mutual exclusion“-Zugriff (wie Speichermoduln)

*nota bene* : Struktur prädestiniert für länger andauernde Verbindungen

# Beispiel:

## B2 : **Raumkoppelstufe** zur Realisierung von Leitungs-/Durchschaltevermittlung

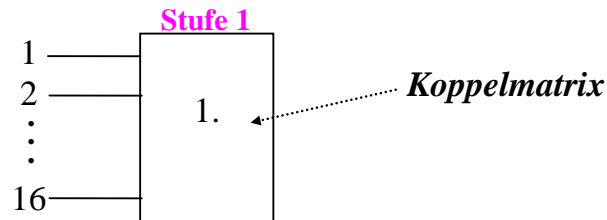
(z.B. in Vermittlungsrechnern in WANs) → siehe Abschn. 6.8.

→ vgl. **“(n stage) Banyan Network“**

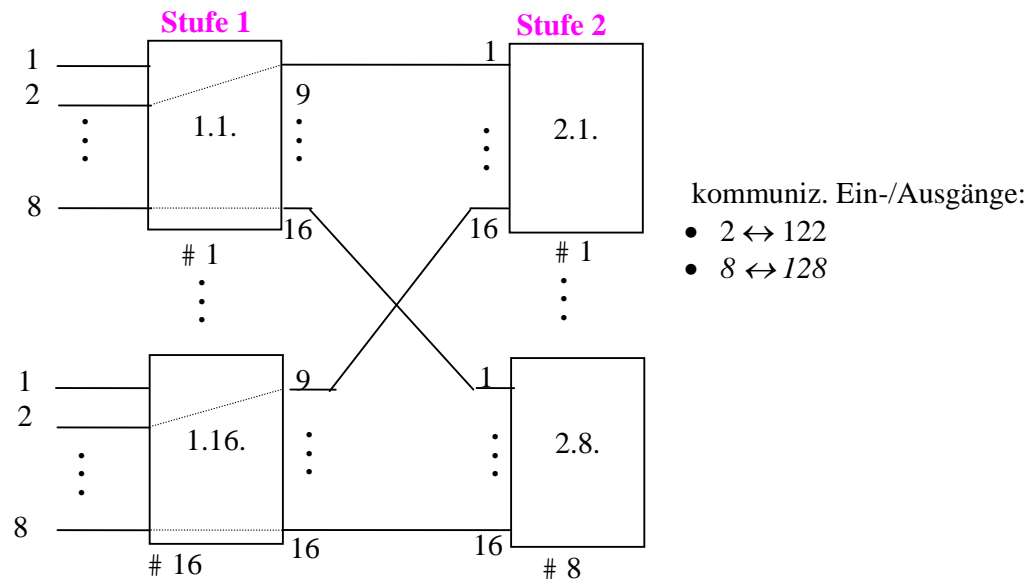
**Basiskomponente** : Koppelmatrix mit Multiplexern als Koppellementen  
(z.B. 16 Ein-/Ausgänge, s. Variante I)



**Variante I :**

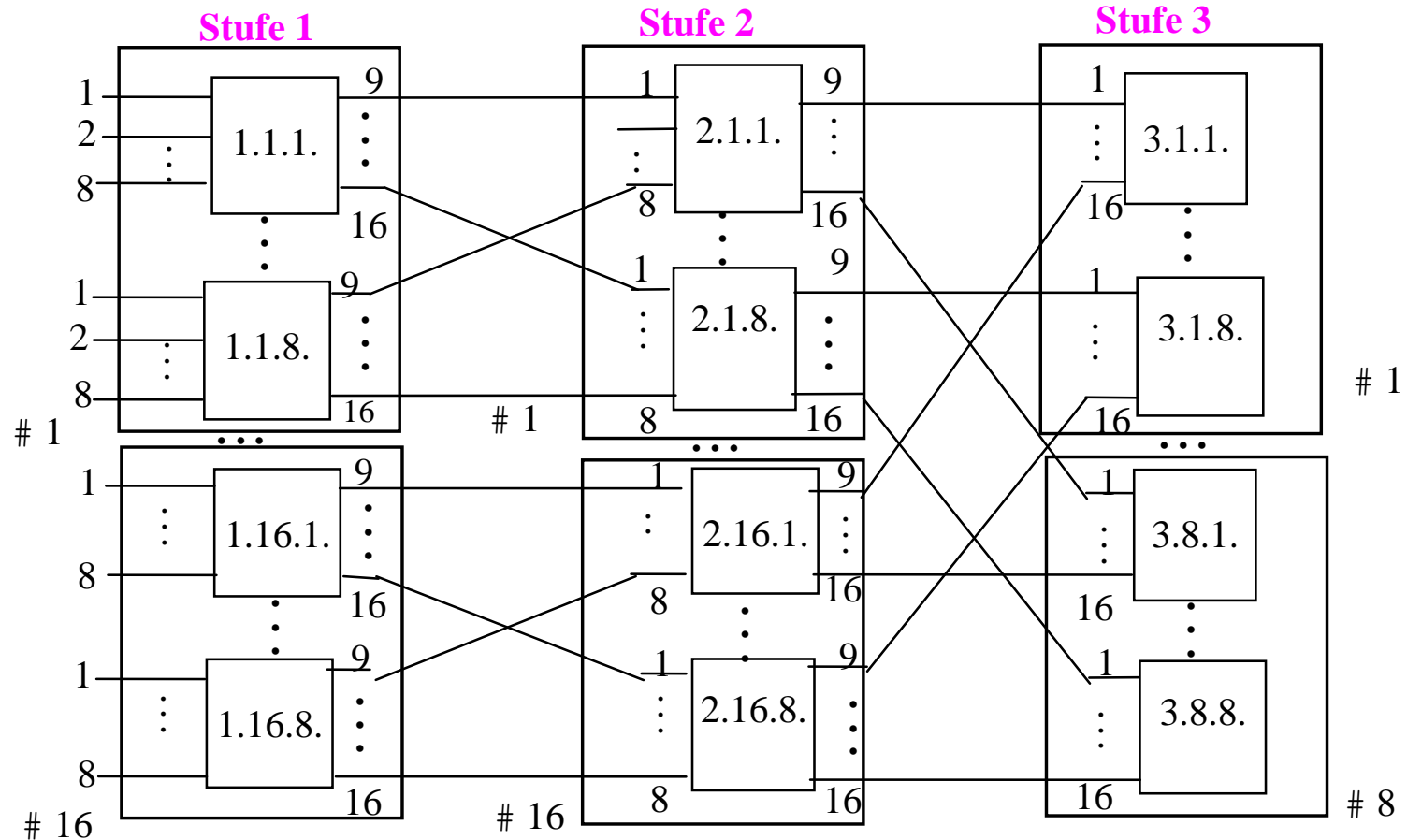


**Variante II :** (128 Ein-/Ausgänge) :

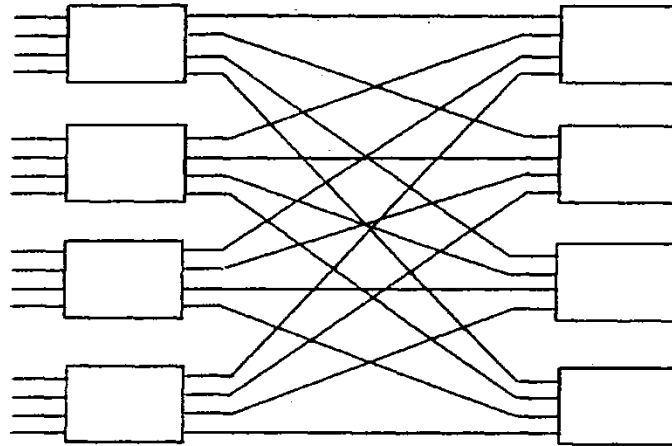


# „(n stage) Banyan Network“ (Forts.):

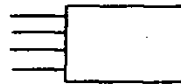
**Variante III :** (1024 Ein-/Ausgänge) :



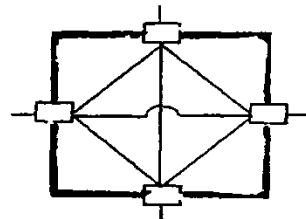
- (1) Zurückführung eines Systems mit 16 Ein-/Ausgängen auf elementarere Struktur



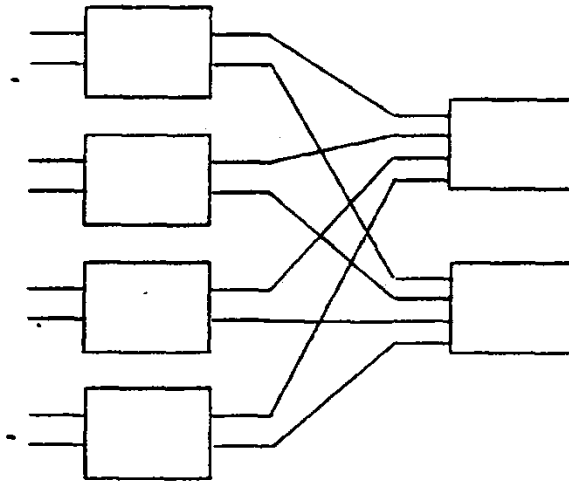
Mögliche „Innen-Struktur“ für :



ist die folgende :



- (2) Zurückführung eines Systems mit 8 Ein-/Ausgängen auf Strukturen mit 4 Ein-/Ausgängen



Bem.: Strukturen mit 4 Ein-/Ausgängen sind hinreichend elementar (s.o.)

## ➤ Verfeinerte Darstellung zu **Bus**

*Zentrales Problem* von Bussystemen :  
***Zugriffskontrolle*** (vgl. Kap. 5.)

Einsatzgebiete, z.B.

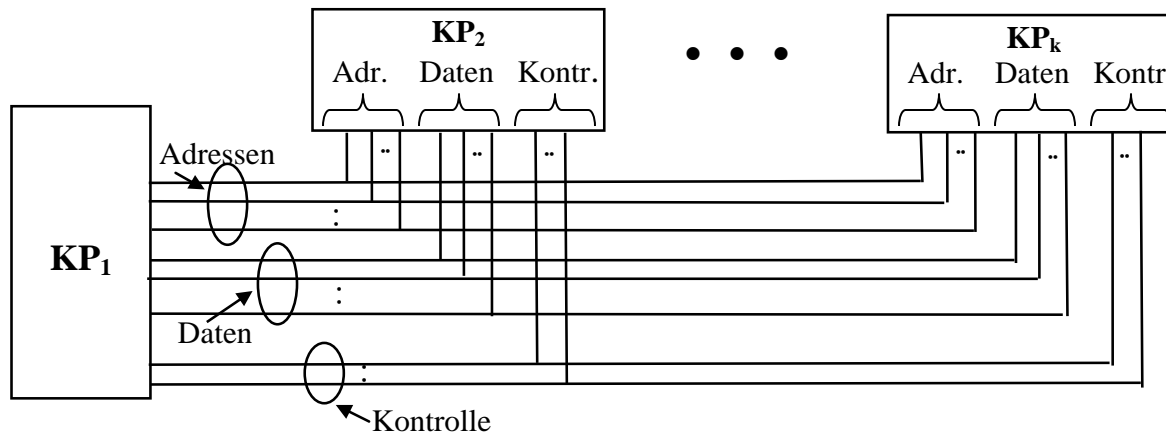
- (rechnerinterner) „Systembus“ zur Interkonnektion von Prozessoren und Speichermoduln → auch für Multiprozessorsysteme
- Bus zur Interkonnektion von Stationen in LANs

***Klassifikation*** der Realisierungsvarianten von Bussystemen :

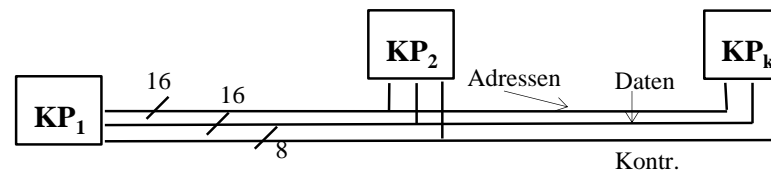
- ***Zugriffskontrolle***
  - Aufforderung
  - zufällig
  - Reservierung
- *Art der* ***Datenübertragung***
  - bitseriell
  - bitparallel
- ***Verwendungszweck*** des Busses
  - Datenbus
  - Adressbus
  - Steuerbus



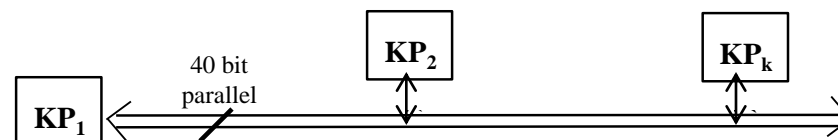
# Unterschiedl. Detaillierungsgrad für die Darstellung der Verbindungsstruktur



Vergrößerung :



weitere Vergrößerung :





## ➤ Verfeinerte Darstellung zu **Ring**

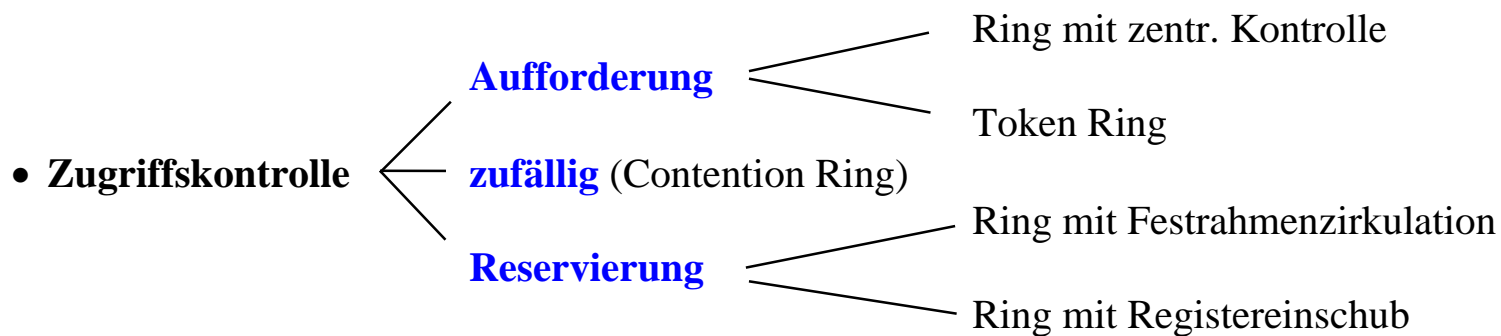
*Zentrales Problem* von Ringnetzen :

***Zugriffskontrolle*** (vgl. Kapitel 5)

Einsatzgebiete, z.B.

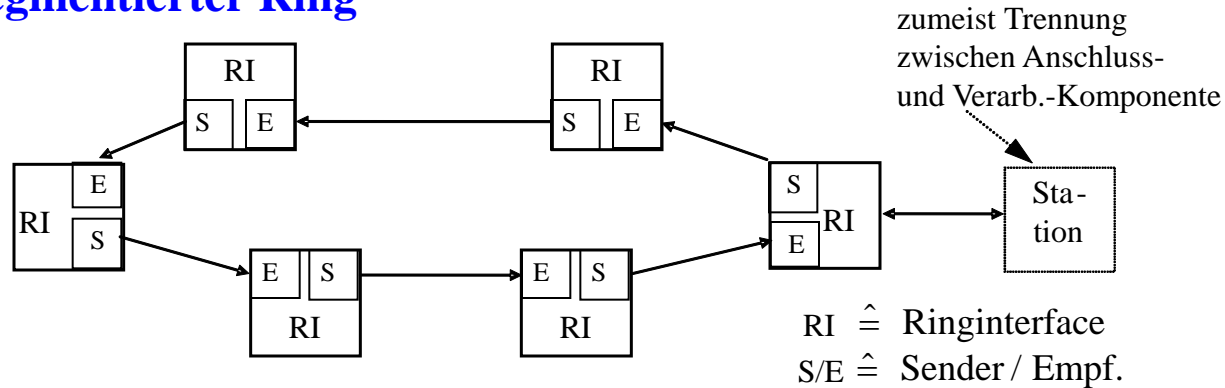
- Anschluss von Peripherie an Rechner
- Ring zur Interkonnektion von Prozessoren in Multiprozessorsystemen und von Stationen in LANs

***Klassifikation*** der Realisierungsvarianten von Ringnetzen :

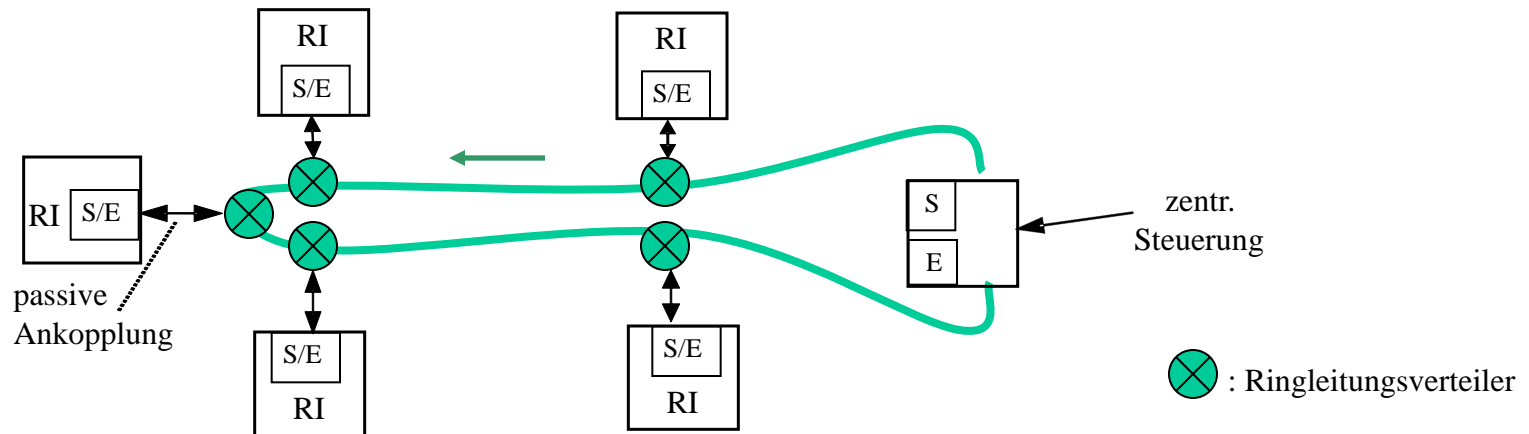


# Varianten für die Verbindung der Kommunikationspartner in Ringnetzen

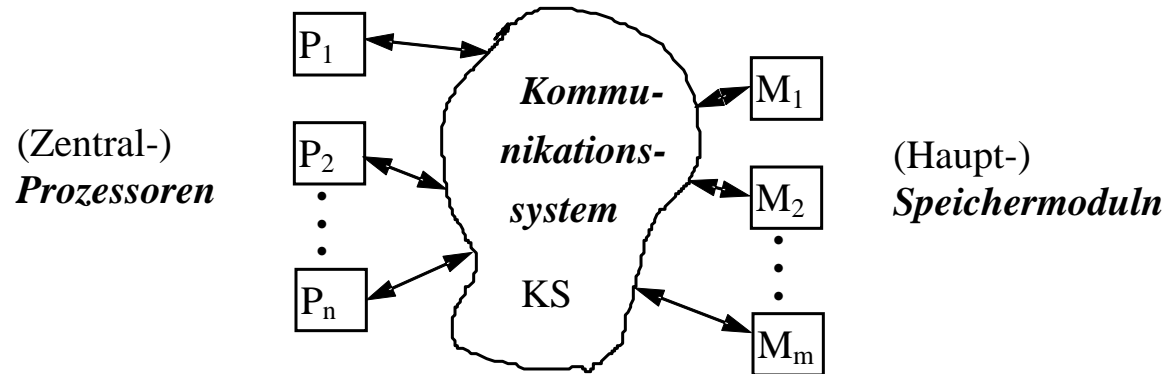
## (d1) Segmentierter Ring



## (d2) Unsegmentierter Ring



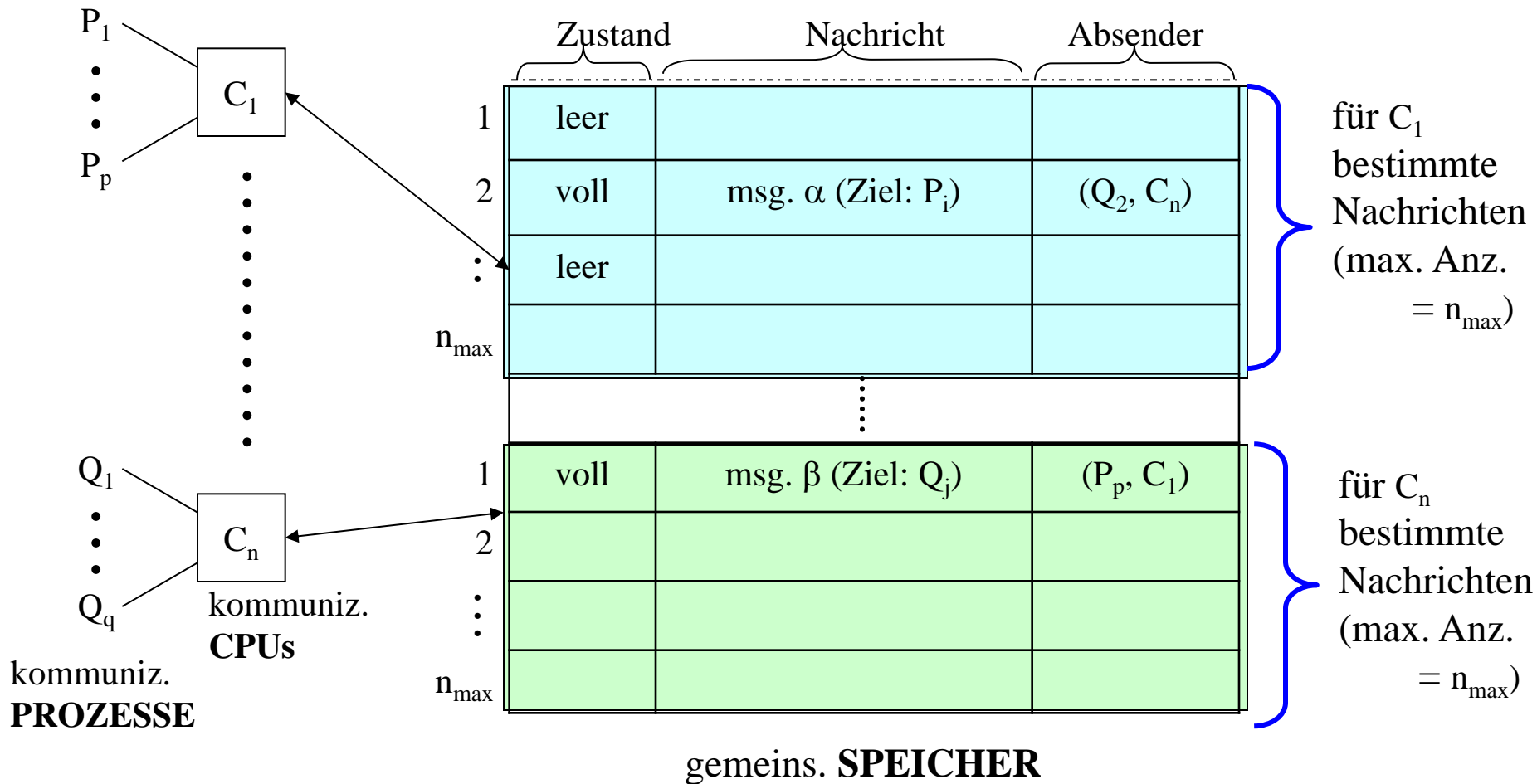
# Verfeinerte Darstellung zu **gemeinsamer Speicher**



**Zentrale Probleme** der Kommunikation über gemeins. Speicher :

- (1) Realisierung eines geeigneten KS, z.B.
  - individ. Leitungen
  - Kreuzschienenverteiler
  - Bus
 (Beschreibung s.o.)
- (2) Setzen von temporären Zugriffssperren bei Speichermoduln
  - („mutual exclusion“ - MUTEX)
 ⇒ Reduktion der Anzahl benötigter Sperren durch stärkere Speichermodularisierung !

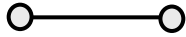
# Beispiel einer geeigneten Strukturierung des gemeinsamen Speichers für eine Interprozesskommunikation :



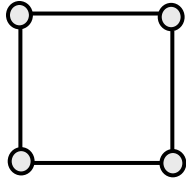
*nota bene* : resultierende „Producer-Consumer-Beziehungen“

# Verfeinerte Darstellung zu Hyperwürfel / HYPERCUBE

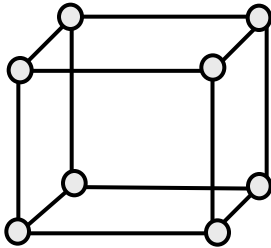
$n = 1 :$



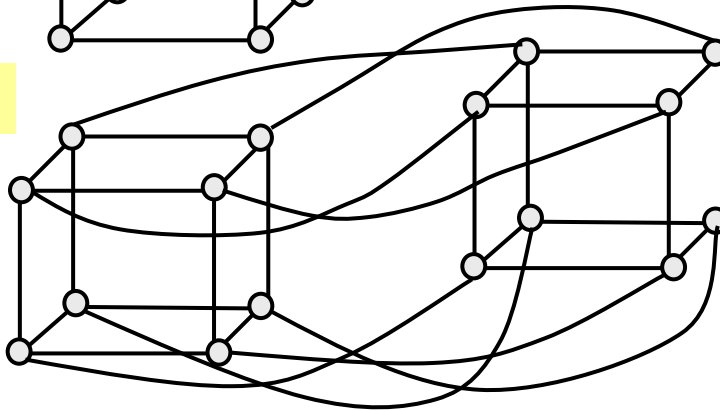
$n = 2 :$



$n = 3 :$



$n = 4 :$



**Dimension  $n$  :**

- $2^n$  Knoten

- $n$  ausgehende Verbindungen von jedem Knoten,

ergo :  $n \cdot 2^n / 2 = n \cdot 2^{n-1}$  Verbindungen insgesamt



# Vorteile Hypercube :

- relativ einfache Wegeermittlung
- relativ gute Zuverlässigkeit durch Alternativpfade
- geringe Anzahl von “*hops*“ (Knotendurchquerungen) zu jedem Ziel, max.  $n$  “hops“ bei Dimens. =  $n$
- relativ geringe Anzahl benötigter Verbindungen

nota bene : Struktur prädestiniert für Austausch einzelner Dateneinheiten (z.B. Datenpakete); Nutzung von Hypercubes insbes. in Multiprozessorsystemen

## 4.3 Architekturen von Vermittlungsrechnern

Zentrale Anforderungen an heutige Vermittlungsrechner :

(höchst) effiziente und zuverlässige Weiterleitung von Dateneinheiten

→ häufig: **Spezialrechner** !

ABER: „im Kommen“ sind auch **Software-Router** ohne Spezialhardware, vgl. hierzu DFG-Projekt **MEMPHIS** bei TKRN seit 3/2012

**Beispiele** : IP-Router, ATM-Vermittlungsrechner, Fast Ethernet-Switch oder -Hub (zu entspr. Protokollen, vgl. Kap. 5/6)

Offensichtlich : Starker Einfluss der Vermittlungstechnik (Details, vgl. Kap. 6), u.a. mit Extremen wie

- ***Durchschaltetechnik***

→ evtl. gesamter Verkehr einer Zugangsleitung auf genau eine Abgangsleitung zu transportieren

- ***Datagrammtechnik***

→ evtl. adaptive Wegewahl mit Wegeermittlung für jedes Paket/Datagramm [sehr hoher Aufwand pro Paket !]

# Neue **Tendenz** bei Vermittlungsrechnern :

## Optische Vermittlung

(bei opt. Signalübertragung über Zu-/Abgangsleitungen)

→ in spe : Tbit/s an vermittelbarem Durchsatz erzielbar

In der Folge indes :

Fokus auf heutigen Vermittlungsrechnern

(d.h. *nicht-optische* Vermittlung)

*nota bene* :

Zahlreiche der Kommunikationssystem-Infrastrukturen aus Abschn. 4.2 nutzbar (aber zu erweitern wegen erhöhten Leistungs-/Zuverlässigkeitsanforderungen)



# Klassifikation von Topologien im Kontext von Vermittlungsrechnern

- *Stufe* (*stage*) : ein- oder mehrstufige Schaltfunktionalität
- *Pfad* (*path*) : zeitgleich nur eine oder mehrere Übertragungen (von Ein- zu Ausgang) durch Struktur unterstützt

→ für Klasseneinteilung verwendet:

- Single- versus Multi-Stage  
sowie
- Single- versus Multi-Path



# TOPOLOGY CLASSIFICATION



| <u>Class</u>  | <u>Schematic Example</u> |
|---|--------------------------|
| <b>SHARED MEDIUM</b><br>SINGLE-STAGE<br>SINGLE-PATH<br><ul style="list-style-type: none"> <li>• Bus</li> <li>• Ring</li> <li>• Star coupler</li> </ul>  |                          |
| <b>CROSSBAR MATRIX</b><br>SINGLE-STAGE<br>MULTI-PATH  |                          |
| <b>NODAL NETWORK</b><br>MULTI-STAGE<br>MULTI-PATH<br><ul style="list-style-type: none"> <li>• Banyan, Delta, Omega, ...</li> <li>• Distribution Network/Banyan</li> <li>• Expanded Banyan</li> <li>• Parallel Banyans</li> <li>• Sort/Banyan (Batcher/Banyan)</li> <li>• Clos, Benes, Richard, ...</li> </ul> |                          |

# Klassifikation bezüglich Platzierung von Warteschlangen

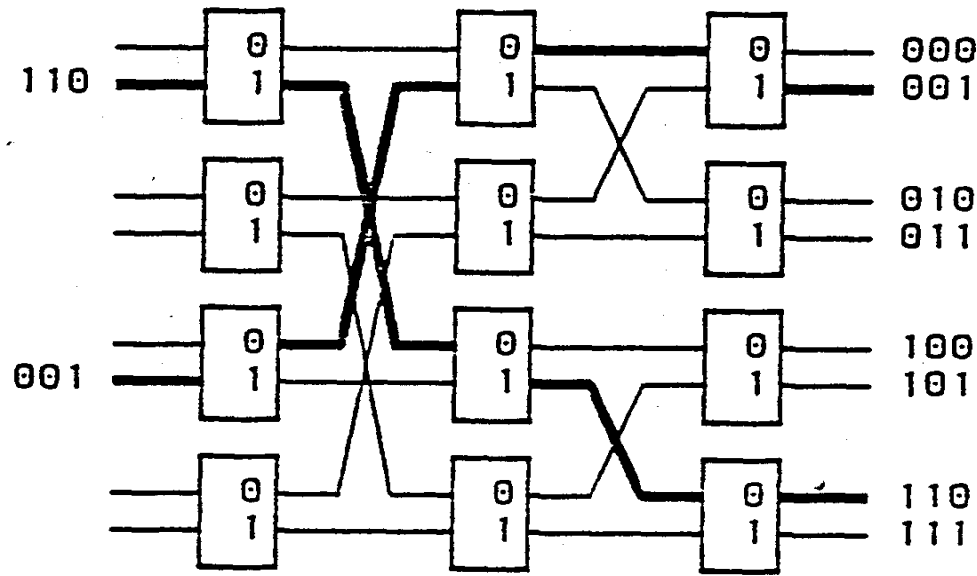


## QUEUEING CLASSIFICATION

| <u>Class</u>   | <u>Schematic Example</u> |
|--|--------------------------|
| <b>INPUT QUEUEING</b> <ul style="list-style-type: none"> <li>• Crossbar with Input Queues</li> <li>• Sort/Banyan with Input Queues</li> <li>• <math>K = 1</math> (typical)</li> <li>• <math>K &gt; 1</math></li> </ul>             |                          |
| <b>OUTPUT QUEUEING</b> <ul style="list-style-type: none"> <li>• Crossbar with Output Queues</li> <li>• Bus with Output Queues</li> <li>• <math>L = N</math> (typical)</li> <li>• <math>L &lt; N</math></li> </ul>                  |                          |
| <b>CENTRALIZED QUEUEING</b>  |                          |
| <b>DISTRIBUTED QUEUEING</b> <ul style="list-style-type: none"> <li>• Classical Buffered Banyan</li> <li>• Banyan, Richard, ... with Input-Buffered Modules</li> <li>• Banyan, Richard, ... with Output-Buffered Modules</li> </ul> |                          |
| <b>COMBINATIONS</b> <ul style="list-style-type: none"> <li>• Input and Output Queues</li> <li>• Buffered Banyan and Input Queues</li> </ul>  |                          |

# Erweiterung der Banyan-Netze (vgl. Abschn. 4.2) für Vermittlungsrechner mit Paket- oder (ATM-) Zellenvermittlung

## ➤ Struktur eines Banyan-Netzes :



## ➤ Funktion :

- Paketkopf enthält binäre Zieladresse (Id. der Ausgangsleitung)
- erstes Adressbit bestimmt in jedem Knoten (des Banyan-Netzes) eindeutig den weiteren Weg ("*self-routed packet-switching*")
- erstes Adressbit in jedem Knoten eliminiert nach vollzogener Wegeermittlung

➤ **Charakteristika von Banyan-Netzen :**

- $\lg(N)$  Stufen,  $N/2$  Knoten pro Stufe
- weniger Knoten im Vergleich zu (rechteckigem) Kreuzschienenverteiler  
aber: kompliziertere Knoten

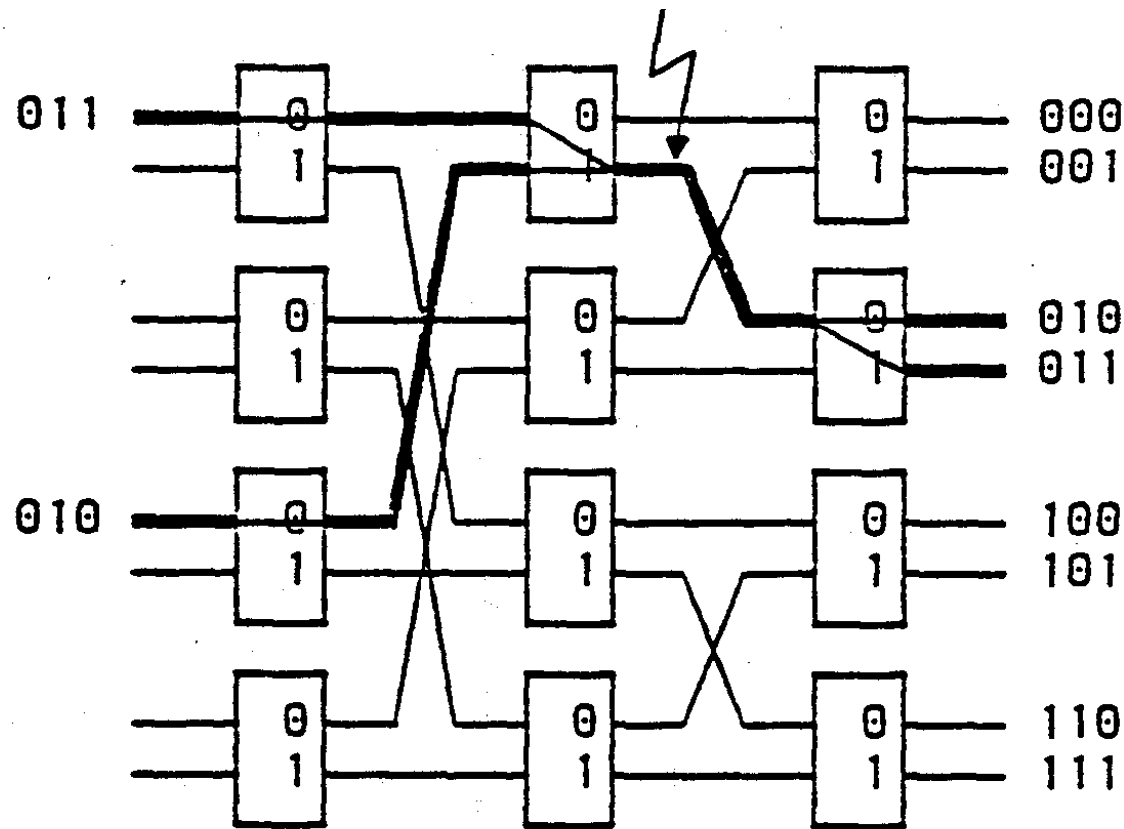
| N x N     | Knoten | Kreuzungspunkte |
|-----------|--------|-----------------|
| 8 x 8     | 12     | 64              |
| 64 x 64   | 192    | 4.096           |
| 128 x 128 | 448    | 16.384          |

- modularer Aufbau, VLSI-Implementierung möglich
- wahlweise synchroner oder asynchroner Betrieb
- nur max. 1 Pfad zwischen jedem Paar von Ein-/Ausgängen

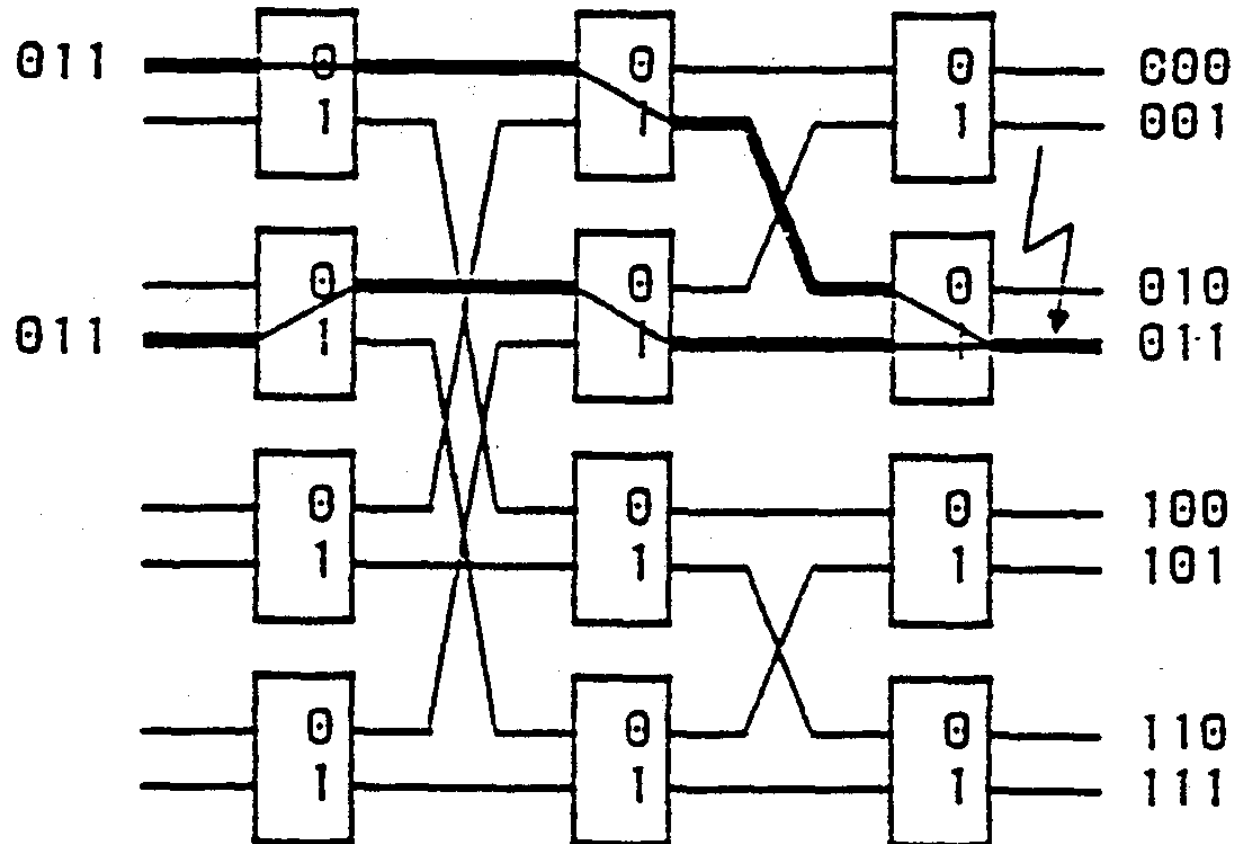
➤ **Nachteile :** BLOCKIERUNGEN (auf internen Leitungen sowie an Ausgängen) möglich

# Blockierungen in Banyan-Netzen

- Blockierung auf einer **internen Leitung**



- Blockierung an einer **Ausgangsleitung**



⇒ Verbesserungen zur **Reduktion oder Vermeidung von Blockierungen** :

- **Paketpuffer** in allen Knoten → Buffered Banyan  
(Klassisch : zusätzliche Speicher für genau 1 Paket pro Knotenzugangsleitung)
- **“handshake“** zwischen Stufen und Eingangswarteschlangen  
(Drosselung des Zuflusses an Daten)
- **Erhöhung** der **internen Geschwindigkeit** (für Datenweiterleitung)
- **Vorschalten** eines **Verteilnetzes** (“distribution network“) vor das Banyan-Netz
- paralleler Einsatz **multipler Banyan-Netze**





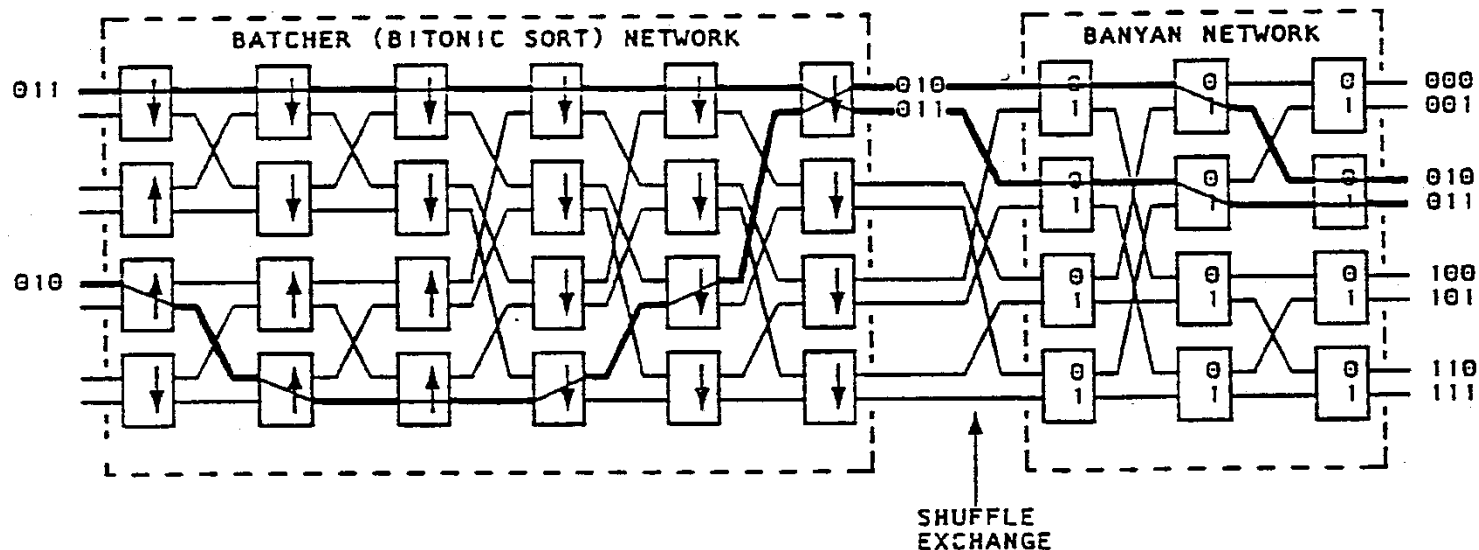
# Sort/Banyan-Netze

➤ Grundlegende Idee für Sort/Banyan-Netze (d.h. Banyan-Netze mit Vorsortieren der Dateneinheiten) :

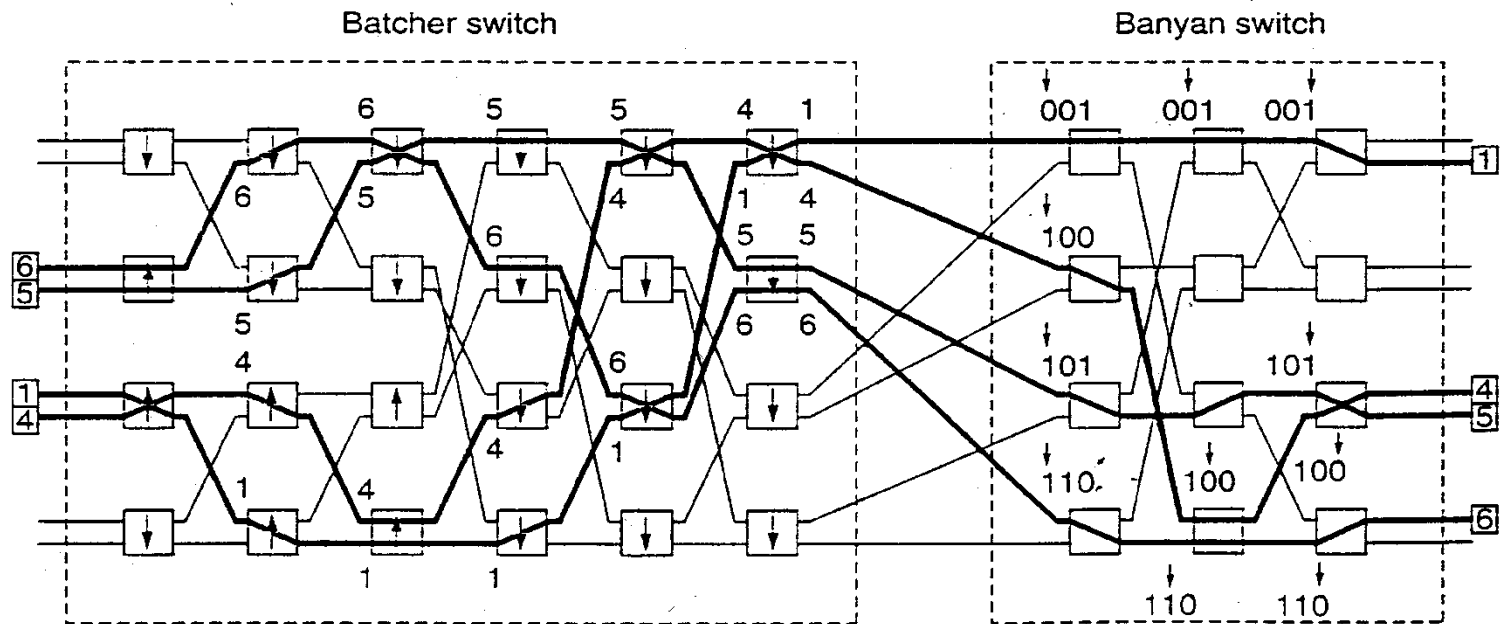
→ *interne Blockierung* in Banyan-Netzen *vermeidbar*, wenn die Pakete :

- vorsortiert werden gemäß ihrer Zieladresse (Id. Ausgangsleitung),
- nach Sortiervorgang in geeigneter Weise an das Banyan-Netz übergeben werden (mittels “shuffle exchange“ bzw. “shuffle network“),
- allesamt für paarweise disjunkte Ausgangsleitungen bestimmt sind.

➤ Struktur eines Sort/Banyan-Netzes



- **Beispiel** eines konkreten Sortiervorgangs (mit 4 Paketen, Zellen, o.ä.)



- Einige **Charakteristika von Sort/Banyan-Netzen:**

### *Batcher-Stufen*

- $\frac{1}{2} ((\text{Id } N)^2 + \text{Id } N)$  Batcher-Stufen und  $\text{Id}(N)$  Banyan-Stufen
- $N/2$  Knoten pro Stufe
- bei fester Paketlänge : synchroner Betrieb



# Erweiterung der Kreuzschienenverteiler (vgl. Abschn. 4.2) für Vermittlungsrechner

## Idee :

Platzierung von Warteschlangen an Zugangsleitungen oder an die Kreuzungspunkte des Kreuzschienenverteilers

| <u>Class</u>  | <u>Principle Structure</u> |
|---|----------------------------|
| <b><u>SINGLE INPUT QUEUES.</u></b><br><b><u>NxN MATRIX</u></b><br><br><b>EXAMPLE</b> <ul style="list-style-type: none"> <li>• PHILIPS patent</li> </ul>                 |                            |
| <b><u>MULTIPLE INPUT QUEUES.</u></b><br><b><u>KNxN MATRIX</u></b><br><br><b>EXAMPLE</b> <ul style="list-style-type: none"> <li>• SIEMENS ATD switch (S.u.)</li> </ul>   |                            |
| <b><u>CROSSPOINT QUEUES.</u></b><br><b><u>NxN MATRIX</u></b><br><br><b>EXAMPLE</b> <ul style="list-style-type: none"> <li>• FUJITSU Bus Matrix Switch module</li> </ul> |                            |

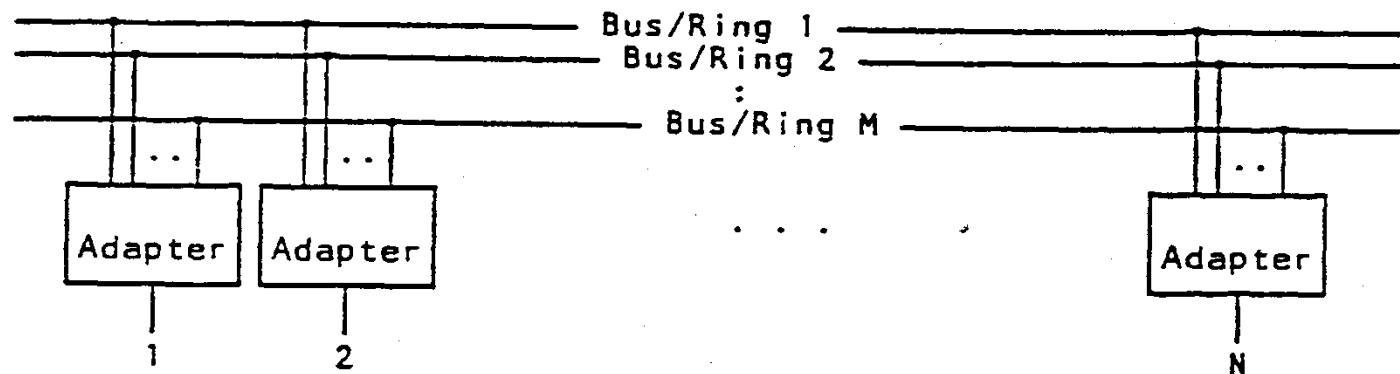


# Erweiterung von Bussystemen/Ringnetzen (vgl. Abschn. 4.2) für Vermittlungsrechner

## ➤ Medien

- *einziges* Bussystem, Ringnetz oder Sternkoppler bei opt. DÜ
- *multiple* Bussysteme, Ringnetze oder opt. Sternkoppler

## ➤ Prinzipielle Struktur:



➤ **Mögliche Vorteile :**

- Leistungsgewinn :  $M$ -facher maximaler Durchsatz in System ohne Ausfall, indes Paketübertragungszeit nicht reduziert bei nur einer Sendung
- Verfügbarkeitsvorteil : **“gracefully degrading system“**  
(bei  $\leq M-1$  ausgefallenen Bussystemen bzw. Ringnetzen)

➤ **Erhöhter Aufwand durch :**

- höheren “Verdrahtungsaufwand“
- höhere Komplexität der Adapter (u.a. separate Zugriffskontrolle für jeden Bus bzw. jedes Ringnetz)
- $M$  Ein-/Ausgänge zu den Bus-/Ring-Subsystemen benötigt pro Adapter