



Theorie und Einsatz von Verbindungseinrichtungen in parallelen Rechnersystemen

Dynamische Verbindungsnetzwerke

29. Juni 2012

Andy Georgi

INF 1046

Nöthnitzer Straße 46

01187 Dresden

0351 - 463 38783

Agenda

- 1 Einführung
- 2 Klassifizierung
- 3 Einstufige Netze
- 4 Mehrstufige Einpfadnetze
- 5 Mehrstufige Mehrpfadnetze
- 6 Literaturverzeichnis

Agenda

1 Einführung

- Steuerung von Verbindungen mit Hilfe von aktiven Koppelementen
- Topologische Parameter *Grad* und *Durchmesser* nicht maßgebend

- Allgemeines Modell eines dynamischen Verbindungsnetzwerks:
 - Verteilung der Koppellemente auf k Schaltstufen
 - Schaltstufe j mit $0 \leq j < k$ enthält u_j Koppellemente
 - Verbindung der Schaltstufen über statische Verbindungsstrukturen
 - Ein Koppellement $S_{i,j}$ besitzt $E_{i,j}$ Eingänge und $A_{i,j}$ Ausgänge

Einführung III

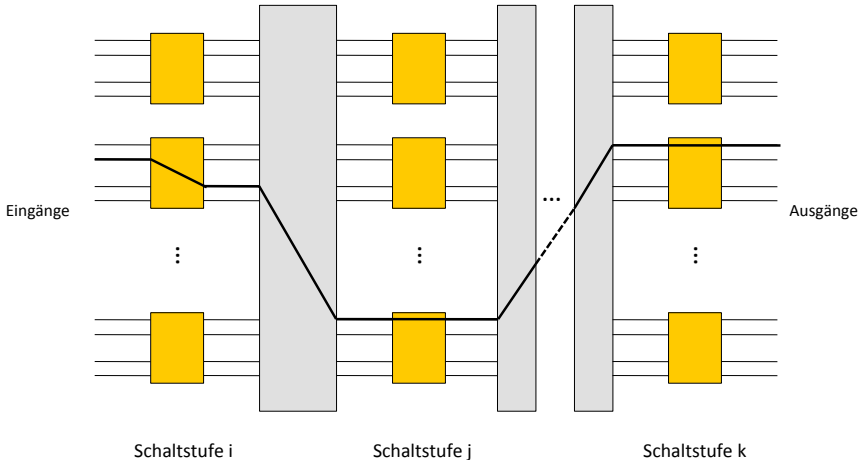


Abbildung: Allgemeines Modell eines dynamischen Verbindungsnetzwerks

- 2 Klassifizierung
 - Einstufige vs. mehrstufige Netze
 - Einpfad- vs. Mehrpfadnetze
 - Blockierend vs. blockierungsfrei vs. rearrangierbar

Einstufige vs. mehrstufige Netze

Abhängig von der Anzahl der Stufen werden Netze mit $k = 1$ als *einstufig* (*single stage*) und Netze mit $k > 1$ als *mehrstufig* (*multi-stage*) bezeichnet.

Einfad- vs. Mehrpfadnetze

Existiert genau ein Weg von jedem Knoten zu jedem anderen Knoten, so spricht man von *Einfadnetzen* (*single path interconnects*). Verfügt das Netz dagegen über alternative Wege zwischen einem Sender und einem Empfänger, so handelt es sich um ein *Mehrfadnetz* (*multiple path interconnect*).

Blockierend vs. blockierungsfrei vs. rearrangierbar

In Abhängigkeit der erlaubten Permutationsmöglichkeiten ist ein Netz *blockierend*, wenn zu einem Zeitpunkt nicht alle Eingänge auf jeden Ausgang abgebildet werden können, *rearrangierbar*, wenn ggf. nach einer Rekonfigurationen alle Permutationen erlaubt sind oder *blockierungsfrei*, wenn bereits ohne Rekonfiguration bestehender Verbindungen jeder Eingang auf einen beliebigen Ausgang abgebildet werden kann.

- 3 Einstufige Netze
 - Kreuzschienenverteiler
 - Shuffle-Exchange-Netz

Definition

Kreuzschienenverteiler (Crossbars) bestehen aus horizontalen und vertikalen Bussystemen. An jedem Kreuzungspunkt (*Koppelpunkt*) befindet sich ein Schalter mit dessen Hilfe der vertikal verlaufende Bus mit dem horizontal verlaufenden Bus verbunden werden kann.

Kreuzschienenverteiler II

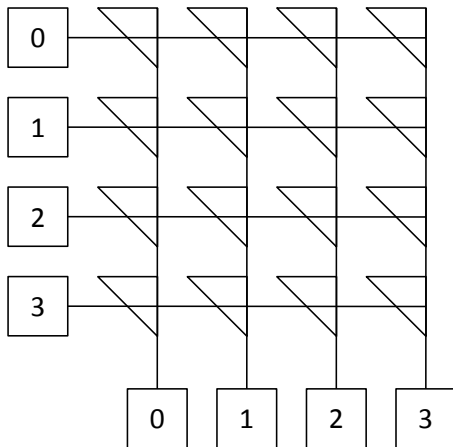


Abbildung: 4x4 Kreuzschienenverteiler

Kreuzschienenverteiler II

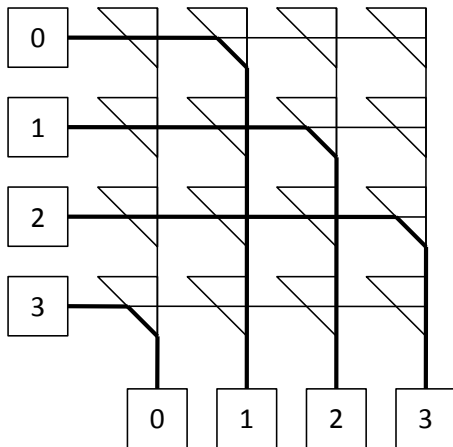


Abbildung: 4x4 Kreuzschienenverteiler

Kreuzschienenverteiler III

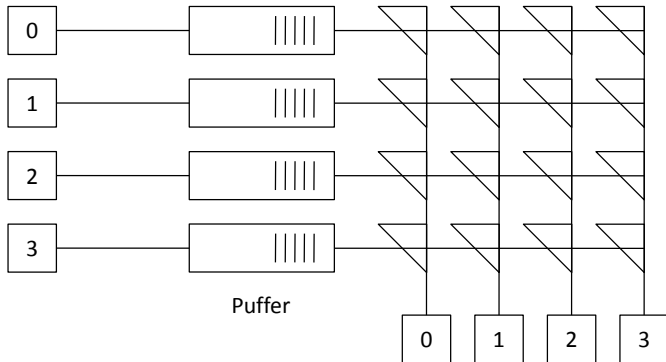


Abbildung: 4x4 KSV mit Eingangspufferung

Kreuzschienenverteiler III

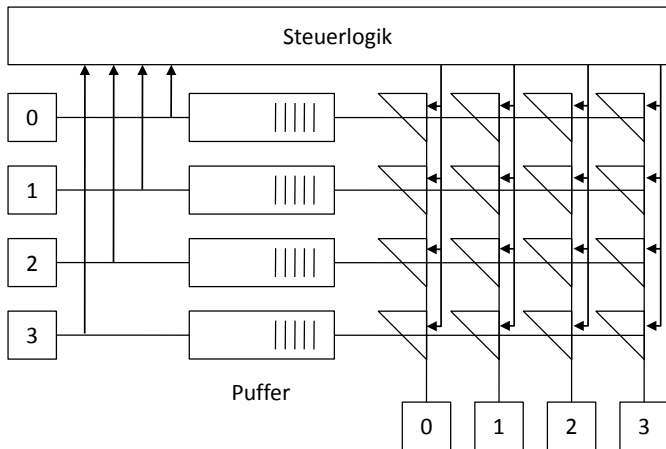


Abbildung: 4x4 KSV mit Eingangspufferung und Steuerlogik

- *Shuffle*:
 - Verschiebung des höchstwertigen Bits auf die niederwertigste Position
 - Realisierung durch die Verbindungsstruktur
- *Exchange*:
 - Invertierung des niederwertigsten Bits
 - Umsetzung durch aktive Koppelemente

Shuffle-Exchange-Netz II

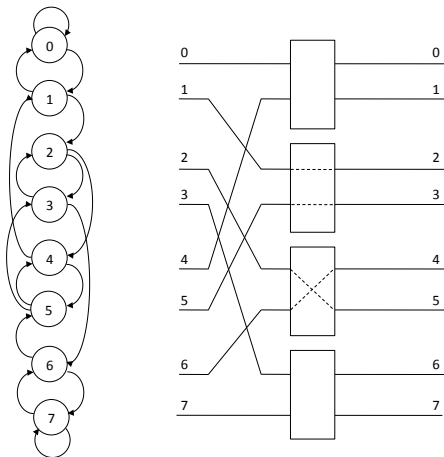


Abbildung: Statisches (links) und dynamisches (rechts) Shuffle-Exchange-Netz

- ④ Mehrstufige Einpfadnetze
 - Banyan-Netz
 - Omega-Netz
 - Generalized-Cube-Netz

Definition

Banyan-Netze [GoL73] werden durch ihre Graphenrepräsentation definiert. Der Graph eines *Banyan-Netzes* besteht aus drei verschiedenen Knoten:

- *Basisknoten*: Knoten ohne Eingangskanten
- *Zwischenknoten*: Knoten mit Ein- und Ausgangskanten
- *Apex-Knoten*: Knoten ohne Ausgangskanten

Die fundamentale Eigenschaft des *Banyan-Graphen* besteht darin, dass exakt ein Weg von jedem *Basisknoten* zu jedem *Apex-Knoten* existiert.

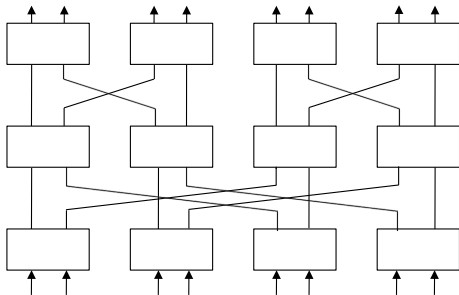
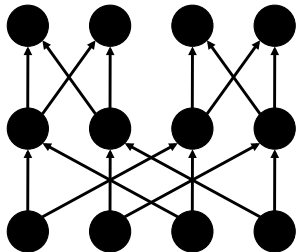


Abbildung: Banyan-Graph (links) und Banyan-Netz (rechts)

Definition

Ein *Omega-Netz* [Law75] mit N Ein- und Ausgängen besteht aus $n = \lg(N)$ Schaltstufen, mit jeweils $N/2$ Beta-Zellen, wobei die Leitungsführung zwischen den Stufen der *Shuffle-Exchange-Funktion* entspricht. Vom Eingang zum Ausgang sind die Stufen in absteigender Reihenfolge, von $n - 1$ bis 0 , nummeriert. Die Indizierung der Ein- und Ausgänge erfolgt hingegen in aufsteigender Reihenfolge von 0 bis $N - 1$ und ist in allen Stufen identisch.

- Verteilte Steuerung durch die zu vermittelnden Nachrichten
- Gegeben:
 - Quelle $Q = q_{n-1} q_{n-2} \dots q_1 q_0$
 - Senke $S = s_{n-1} s_{n-2} \dots s_1 s_0$
- Gesucht:
 - Routing-Tag $T = t_{n-1} t_{n-2} \dots t_1 t_0$

XOR-Routing

Bei Einsatz des *XOR-Routings* wird das n Bit lange Routing-Tag T aus der bitweisen Exklusiv-Oder-Verknüpfung von Q und S gebildet:

$$T = Q \oplus S = t_{n-1} t_{n-2} \dots t_1 t_0$$

Destination Routing

Bei Verwendung des *Destination Routings* impliziert die Zieladresse das Routing-Tag. Das Koppелеlement der Stufe k untersucht dabei die Bitposition k von S und leitet die Nachricht an den oberen Ausgang, wenn $s_k = 0$ oder an den unteren Ausgang wenn $s_k = 1$.

Konflikterkennung

Ein Konflikt tritt auf, sobald zwei oder mehr Kommunikationen an Stufe k gleichzeitig den selben Ausgangsport P_k nutzen wollen. Ist die Quelle Q mit der Senke S verbunden und soll nun die Quelle Q' mit der Senke S' verbunden werden, tritt demzufolge an Stufe k genau dann ein Konflikt auf, wenn $P_k = P'_k$.

Definition

Generalized-Cube-Netze [SiS87] sind topologisch äquivalent zu Omega-Netzen. Die Indizierung der Schaltstufen erfolgt vom Eingang zum Ausgang in absteigender Reihenfolge. Die Indizes der Ein- und Ausgänge liegen zwischen 0 und $N - 1$, wobei sich diese an einer Beta-Zelle in Stufe k ausschließlich in Bit k unterscheiden:

$$\begin{aligned}P_{up} &= p_{n-1} p_{n-2} \dots p_k \dots p_1 p_0 \\P_{down} &= p_{n-1} p_{n-2} \dots \bar{p}_k \dots p_1 p_0\end{aligned}$$

Zudem ist der Ausgang P der Stufe k immer mit dem Eingang P der Stufe $k - 1$ verbunden.

- Beibehaltung des Index bei einer *straight*-Operation
- Ein *exchange* entspricht der *cube*-Funktion
- Fallunterscheidung am Koppellement der Stufe k :
 - $q_k = s_k$: *straight*
 - $q_k \neq s_k$: *cross*

- Weitere, zu den Omega-Netzen, topologisch äquivalente Verbindungsnetzwerke:
 - Indirect-Binary-n-Cube-Netz [Pea77]
 - Flip-Netz [Bat76]
 - Baseline-Netz [WuF80]

- 5 Mehrstufige Mehrpfadnetze
 - Clos-Netz
 - Beneš-Netz

Definition

Das *Clos-Netz* [Clo53] besteht aus drei Schaltstufen, wobei in Stufe k die Anzahl der Koppellemente durch den Parameter r_k , die Anzahl der Eingänge pro Koppellement durch m_k und die der Ausgänge durch n_k definiert ist. Weiterhin gilt $m_2 = r_1$, $m_3 = r_2$, $n_1 = r_2$ und $n_2 = r_3$, womit ein dreistufiges Netz durch die fünf Parameter m_1 , n_3 , r_1 , r_2 und r_3 vollständig definiert wird. Die Verbindung der Schaltstufen erfolgt über die Perfect-Shuffle-Funktion, wobei jedes Koppellement der ersten und letzten Stufe mit jedem Koppellement der mittleren Stufe verbunden wird.

Clos-Netz II

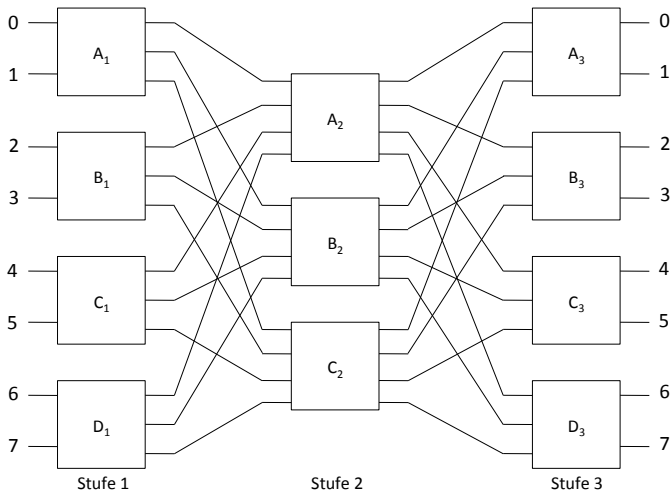


Abbildung: Symmetrisches Clos-Netz mit $q = 4$, $n = 2$ und $r_2 = 3$

Theorem (1)

Ein Clos-Netz ist dann und nur dann streng blockierungsfrei für 1-zu-1-Verbindungen, wenn $r_2 \geq m_1 + n_3 - 1$. Ein symmetrisches Netz ist demnach genau dann streng blockierungsfrei, wenn $r_2 \geq 2n - 1$.

Theorem (2)

Ein Clos-Netz ist genau dann rearrangierbar, wenn $r_2 \geq \max(m_1, n_3)$. Ein symmetrisches Netz mit $m_1 = n_3 = n$ ist demnach genau dann rearrangierbar, wenn $r_2 \geq n$.

- Aufbau einer Verbindung zwischen den Kopelementen A und B
- Umsetzung mit Hilfe der *Paull'schen Verbindungsmatrix* [Pau62]
- Da $r_2 \geq \max(m_1, n_3)$ gilt eine der folgenden Bedingungen:
 - ① Es existiert mind. ein KE der mittleren Stufe welches weder in Reihe A noch in Spalte B existiert
 - ② In Reihe A existiert ein KE C der mittleren Stufe, das nicht in Spalte B erscheint, und es existiert in Spalte B ein KE D der mittleren Stufe das nicht in Reihe A existiert

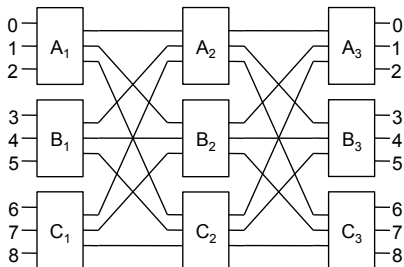
Fall 1:

- Aufbau der Verbindung über das KE welches weder in Reihe A noch in Spalte B existiert

Fall 2:

- Suchen des Eintrags C in Reihe A der nicht in Spalte B existiert
- Suchen des Eintrags D in Spalte B der nicht in Reihe A existiert
- Abwechselnde Fortsetzung des Vorgangs bis kein C oder D mehr auf gleicher Reihe bzw. Spalte gefunden wird
- Rekonfiguration des Netzes indem entlang des Suchwegs alle C durch D und alle D durch C ersetzt werden
- Aufbau der Verbindung über KE D welches jetzt weder in Reihe A noch in Spalte B vorkommt

Rekonfigurationsbeispiel I

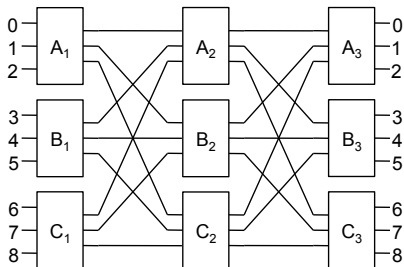


Pauli'sche Matrix:

	A ₃	B ₃	C ₃
A ₁	A ₂		
B ₁		A ₂	B ₂
C ₁			C ₂

Abbildung: Verbindungsaufbau von Knoten 1 zu Knoten 8 führt zu Blockierung

Rekonfigurationsbeispiel II



Pauli'sche Matrix:

	A ₃	B ₃	C ₃
A ₁	A ₂		B ₂
B ₁		B ₂	A ₂
C ₁			C ₂

Abbildung: Rearrangierung bestehender Verbindungen führt zu Blockierungsfreiheit

- Ziel: Verringerung der Komplexität der KE eines Clos-Netzes
- Umsetzung: Rekursiver Aufbau aus 2×2 Crossbars

Allgemeine Konstruktion

Die Grundlage des Konstruktionsprinzips eines *Beneš-Netzes* [Ben64, Ben65] bildet ein dreistufiges symmetrisches Clos-Netz mit N Ein- und Ausgängen. Ein- und Ausgangsstufen sind aus Betazellen aufgebaut, woraus unmittelbar folgt, dass die mittlere Stufe aus zwei $N/2 \times N/2$ Koppелеlementen bestehen muss. Diese wird rekursiv ersetzt bis für $N = 2^n$ Ein- und Ausgänge die $2n - 1$ Stufen erreicht wurden.

Beispiel I - Beneš-Netz mit $N=8$

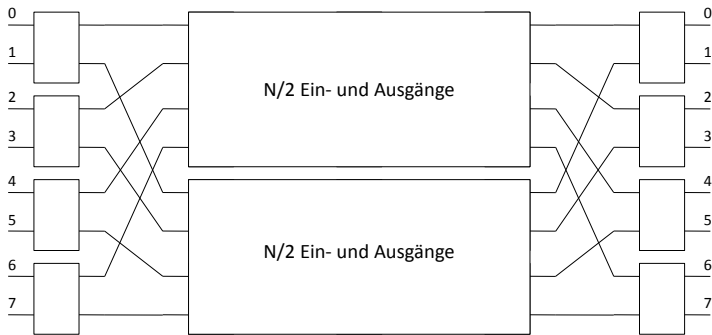


Abbildung: Erster Rekursionsschritt

Beispiel II - Beneš-Netz mit $N=8$

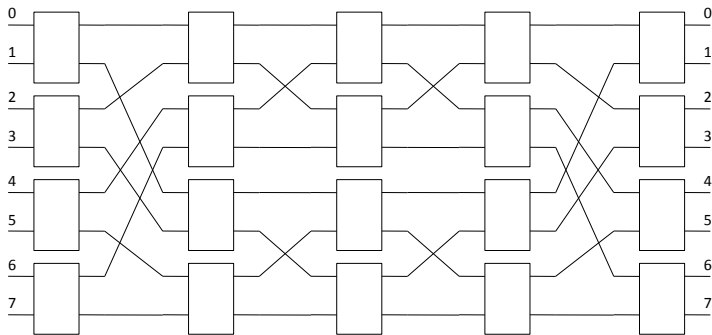


Abbildung: Zweiter Rekursionsschritt

Schleifenalgorithmus [OpT71]:

① Initialisierung:

Der Algorithmus beginnt mit dem Eingangskoppelement 0, welches mit S bezeichnet wird.

② Vorwärtsschleife:

Ein nicht verbundener Eingang von S wird über das obere KE der mittleren Stufe mit dem korrekten Ausgang am 2×2 -KE D verbunden. Falls keine Verbindung erforderlich ist, gehe zu Schritt 4.

③ Rückwärtsschleife:

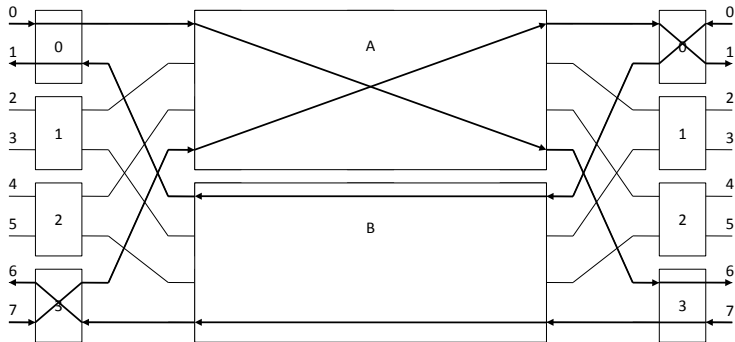
Der benachbarte Ausgang von D wird über das untere KE der mittleren Stufe mit dem korrekten Eingang am Koppelement S' verbunden. Ist keine Verbindung erforderlich, gehe zu Schritt 4. Andernfalls setze $S = S'$ und gehe zu Schritt 2.

- Beginne neue Schleife:





Sind alle notwendigen Verbindungen aufgebaut, beende den Algorithmus. Andernfalls wähle ein noch nicht voll konfiguriertes Eingangskoppelement als S und gehe zu Schritt 2.





Beispiel - Schleifenalgorithmus





Verbindungen in einem Beneš-Netz mit $N=8$: $\{(0,6),(1,0),(6,7),(7,1)\}$



6 Literaturverzeichnis

-  [BuC91] J. R. Burke, C. Chen, T.-Y. Lee, D. P. Agrawal
Performance analysis of single stage interconnection networks, 1991.
IEEE Transactions on Computers, Band C-40, S. 357-365
-  [WuF81] C.-L. Wu, T. Y. Feng
The universality of the shuffle-exchange network, 1981.
IEEE Transactions on Computers, Band C-30, S. 324-332
-  [GoL73] G. R. Goke, G. J. Lipovski
Banyan networks for partitioning multiprocessor systems, 1973.
First Annual Symposium on Computer Architecture, S. 21-28
-  [Law75] D. H. Lawrie
Access and alignment of data in an array processor, 1975.
IEEE Transactions on Computers, Band C-24, S. 1145-1155

-  [SiS87] H. J. Siegel, S. D. Smith
Study of multistage SIMD interconnection networks, 1987.
Fifth Annual Symposium on Computer Architecture, S. 223-229
-  [Pea77] M. C. Pease III
The indirect binary n-cube microprocessor array, 1977.
IEEE Transactions on Computers, Band C-26, S. 458-473
-  [Bat76] K. E. Batcher
The flip network in STARAN, 1976.
International Conference on Parallel Processing, S. 65-71
-  [WuF80] C.-L. Wu, T. Y. Feng
On a class of multistage interconnection networks, 1980.
IEEE Transactions on Computers, Band C-29 S. 694-702

-  [Clo53] C. Clos
A study of non-blocking switching networks, 1953.
Bell System Technical Journal, Band 32, S. 406-424
-  [Pau62] M. C. Paull
Reswitching of connection networks, 1962.
Bell System Technical Journal, Band 41, S. 833-855
-  [Ben64] V. E. Benes
Optimal rearrangeable multistage interconnection networks, 1964.
Bell System Technical Journal, Band 41, S. 1641-1656
-  [Ben65] V. E. Benes
Mathematical Theory of Connecting Networks and Telephone Traffic,
1965.
Academic Press, New York



[OpT71] D. C. Opferman, N.T. Tsao-Wu

On a class of rearrangeable switching networks, 1971.

Bell System Technical Journal, Band 50, S. 1579-1600