

ÜBERSICHTSBLATT: VERTEILTE HARDWARE-ARCHITEKTUREN

In dieser Übersicht sollen die verschiedenen hardwareseitigen Konzepte zur Realisierung verteilter Rechensysteme untersucht werden. Diese Konzepte lassen sich in verschiedene Klassen einteilen.

Klassifikation nach Flynn:

Hierbei werden Instruktions- und Datenströme auf denen Instruktionen arbeiten, als Kennzeichen der Arbeitsweise einer Rechnerarchitektur betrachtet. Man unterscheidet nach Lynn die folgenden Klassen von Hardwarearchitekturen:

SISD:

Single Instruction Stream / Single Data Stream
Dies ist die traditionelle Architektur die in der Regel mit Uniprozessor-Technologie ausgeführt wird.

SIMD:

Single Instruction Stream / Multiple Data Streams
Vektorrechner: Parallele Ausführung einer Instruktion auf mehreren Vektorelementen.

MISD:

Multiple Instruction Streams / Single Data Stream
Diese Klasse wird von den meisten Autoren als leer angesehen.

MIMD:

Multiple Instruction Streams / Multiple Data Streams:
Systeme mit mehreren eng oder lose gekoppelten Prozessoren, parallele und verteilte Rechensysteme. Diese Klasse ist nach Rashid in einige weitere Unterklassen aufgeteilt.

Rashid unterscheidet die einzelnen Architekturen hier noch einmal anhand der Speicherzugriffe:

UMA (Uniform Memory Access):

Hierbei unterhalten alle Prozessoren einen gemeinsamen Adressraum (enge Kopplung). Hieraus ergibt sich eine identische Zugriffszeit für alle Prozessoren auf alle Speicherzelle.

NUMA (Non-Uniform Memory Access):

Auch hier liegt eine enge Kopplung vor. Allerdings ist der gemeinsame Speicher auf verteilten Systemen realisiert. Dadurch entstehen „entfernte“ Speicherzugriffe und die Zugriffszeit der einzelnen Prozessoren auf verschiedene Speicherzellen ist unterschiedlich lang.

NORMA (No Remote Memory Access):

Hierbei gibt es gar keinen gemeinsamen Adressraum mehr. Man spricht hier auch von einer losen Kopplung.

ÜBERSICHTSBLATT: VERTEILTE HARDWARE-ARCHITEKTUREN

Multiprozessor-System:

Unter einem Multiprozessor-System versteht man generell ein MIMD-Rechensystem mit eng gekoppelten identischen Prozessoren. Hierbei kommt es für die Definition allerdings nicht darauf an, ob es sich um eine UMA oder NUMA Architektur handelt. Die Einzelnen Prozessoren kommunizieren hierbei über den gemeinsamen Speicher.

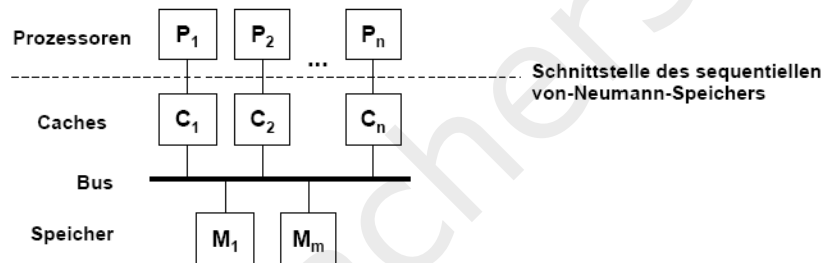
UMA wird hierbei eher für kleine Multiprozessor-Systeme genutzt, für größere Systeme greift man auf die neuere NUMA-Architektur zurück.

Vorteil: Leicht zu programmieren

Realisierungsalternativen für enge Kopplung:

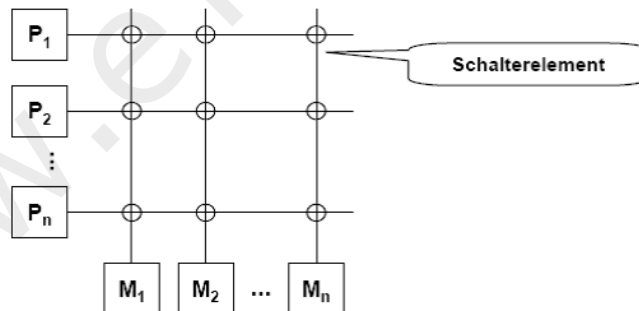
1-Bussysteme:

Dies ist die klassische Form der Realisierung. Alle Prozessoren greifen hier über einen gemeinsamen Bus auf die verteilten Speicher zu.



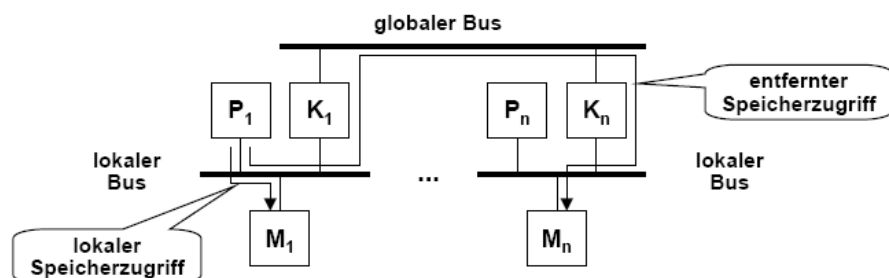
Switch-Basiertes Verbindungsnetzwerk:

Hier geschieht der Speicherzugriff über ein Schalternetz. Durch schließen und öffnen von Schaltern kann hier der jeweils kürzeste Weg zum Speicher realisiert werden.



Hierarchisches Bus-System:

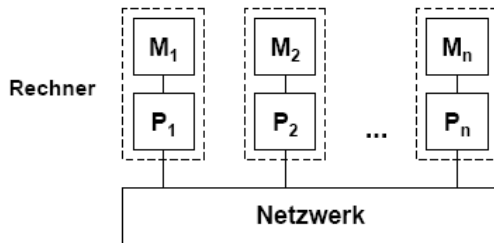
Hierbei werden jeweils noch zusätzliche Kommunikations-Prozessoren eingeführt. Des Weiteren gibt es einen lokalen Bus, auf dem lokale Speicherzugriffe realisiert werden. Und einen globalen Bus über den mittels der Kommunikations-Prozessoren die globalen Speicherzugriffe realisiert werden.



Multicomputer-System:

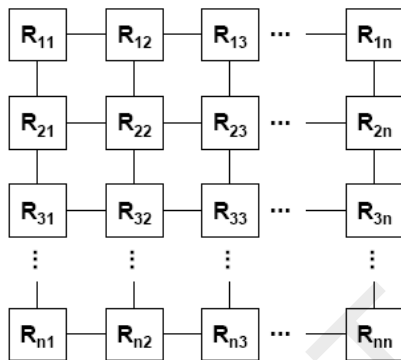
Hierunter versteht man ein MIMD-Rechensystem mit loser Kopplung. Es gibt also keinen gemeinsamen Speicher mehr und so ist die Kommunikation der einzelnen Prozessoren untereinander auf das so genannte Message-Passing angewiesen. Dies bedeutet in der Praxis dass die einzelnen Rechner mit ihren lokalen Speichern an ein Netzwerk angeschlossen sind über welches der Nachrichtenaustausch vorgenommen wird.

Allgemeine Darstellung:

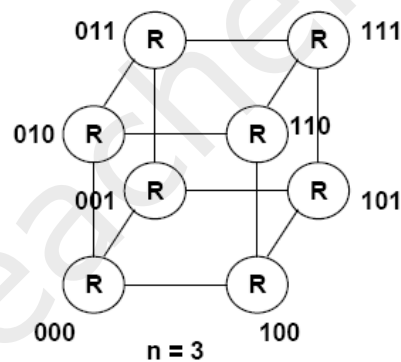


Für diese Netzwerke sind neben dem oben dargestellten Ethernet auch andere Topologien möglich.

Gitter-Topologie:



Hypercube-Topologie:



Vorteil: Leicht auf eine große Anzahl von Prozessoren erweiterbar.