

Proseminar Rechnerarchitekturen

Parallelcomputer: Multiprozessorsysteme

Stefan Schumacher, <stefan@net-tex.de>, PGP Key <0xB3FBAE33>
<http://www.net-tex.de/uni>

Id: mps-folien.tex,v 1.1.32 2003/11/14 21:14:34 stefan stable

Gliederung

- Motivation
- Taxonomie
- Vektorrechner
- Feldrechner / systolisches Array
- Busbasierte Multiprozessorsysteme
- Schalterbasierte Multiprozessorsysteme
- Speicherarchitekturen & -verwaltung

Warum Multiprozessorsysteme?

- höhere Leistung
- bessere Skalierbarkeit
- besseres Preis/Leistungsverhältnis
- verteilte Nutzung von Daten und Ressourcen
- Sicherheit dank Redundanz
- einfachere Kommunikation

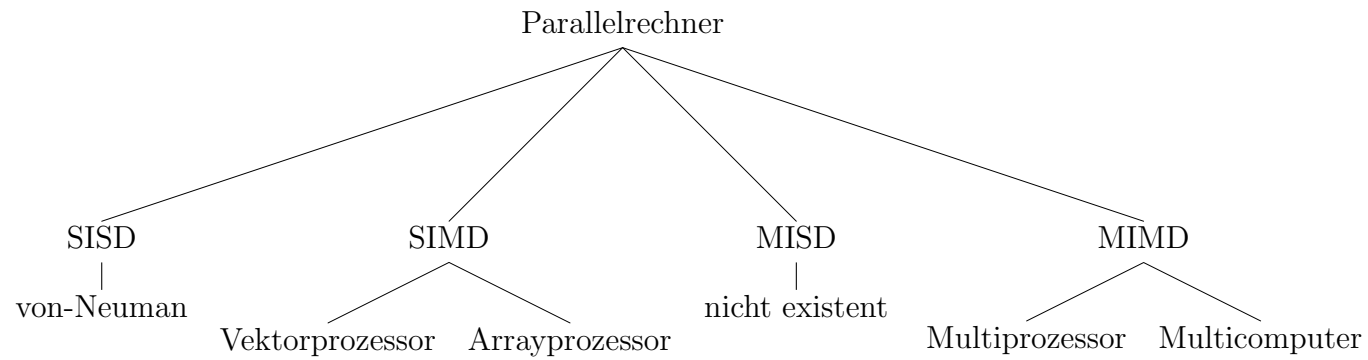
Nachteile

- Verbindungsnetzwerk ist Single Point of Failure
- Verbindungsnetzwerk meist Bottleneck

Hardwarekonzepte

- Kategorisierung nach Verbindung und Kommunikation
- Einteilung nach Flynn
- SISD : Single Input Single Data
- SIMD : Single Input Multiple Data
- MISD : Multiple Input Single Data
- MIMD : Multiple Input Multiple Data Stream

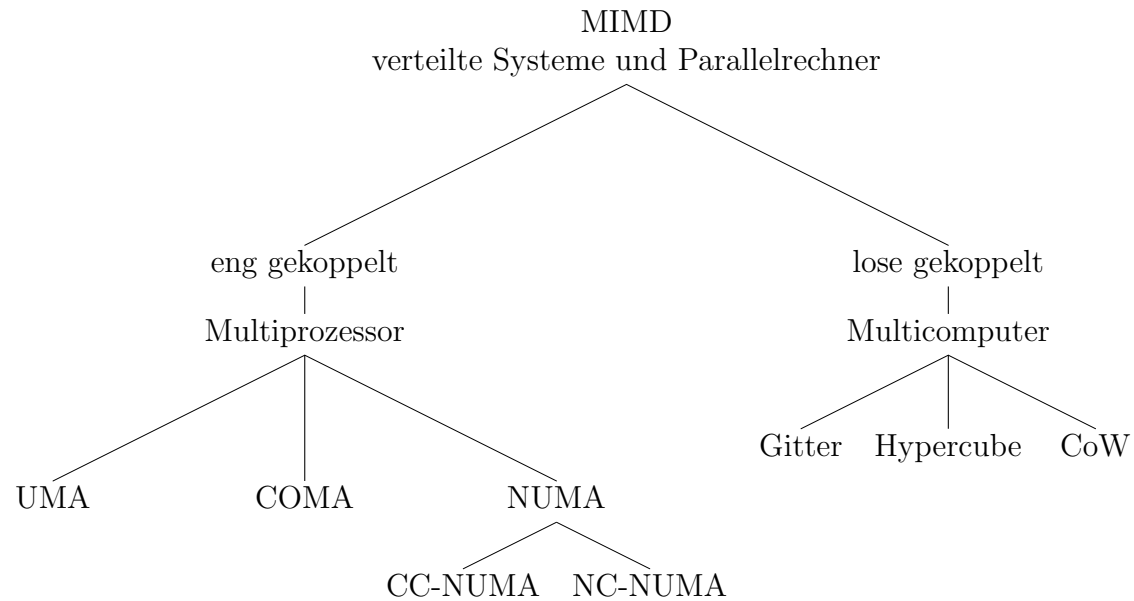
Flynn-Schema



Tanenbaum-**Erweiterung**

- erweitert Flynn-Schema bezgl. verteilter Systeme
- eng gekoppelte Systeme
- lose gekoppelte Systeme
- weitere Unterteilung bezgl. Kommunikation und Cache
- Busbasiert
- Schalterbasiert

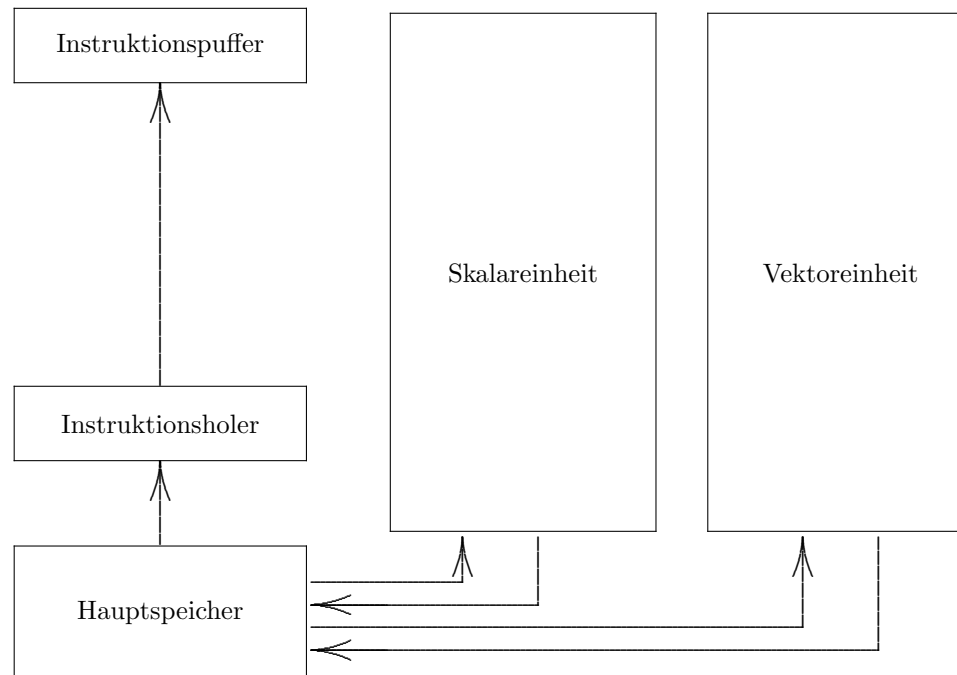
Tanenbaum-**S**chema



Vektorprozessoren

- Parallelisierbarkeit von Vektoroperationen
- bestehen aus Skalareinheit und Vektoreinheit
- Parallelisierbarkeit transparent in Hardware (Pipeline)
- Vorteil: gewohnt sequentiell programmierbar
- einige Compiler können parallelisieren

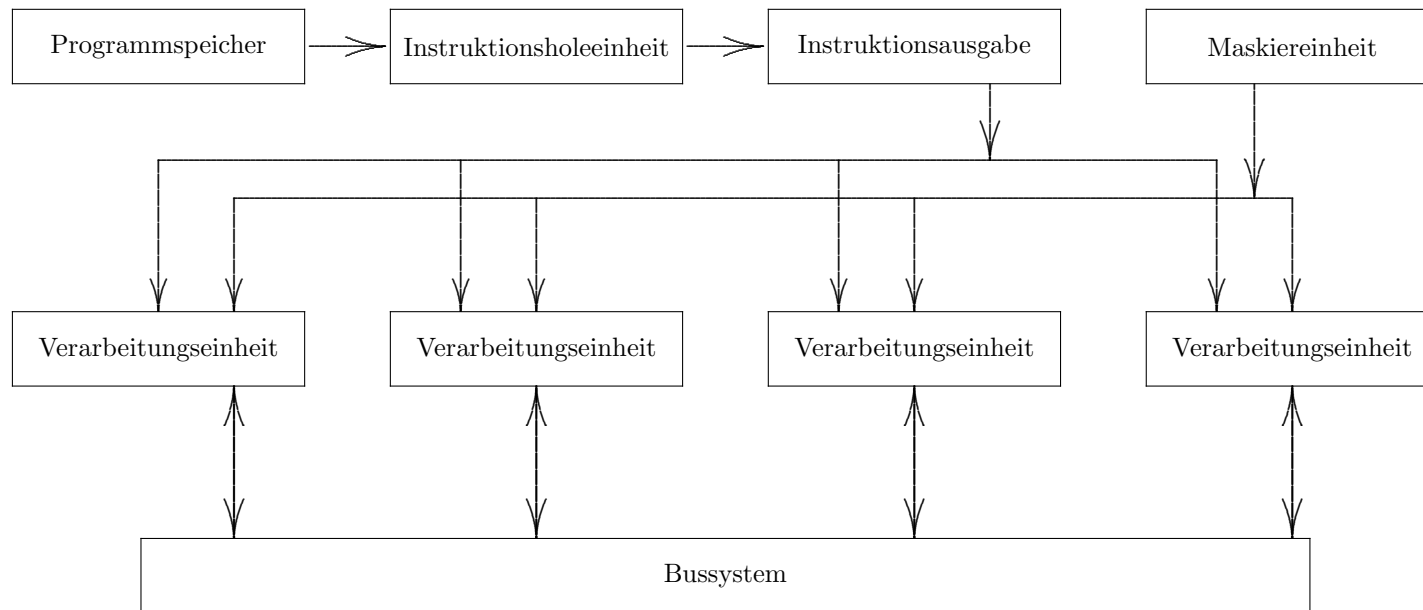
Vektorrechner



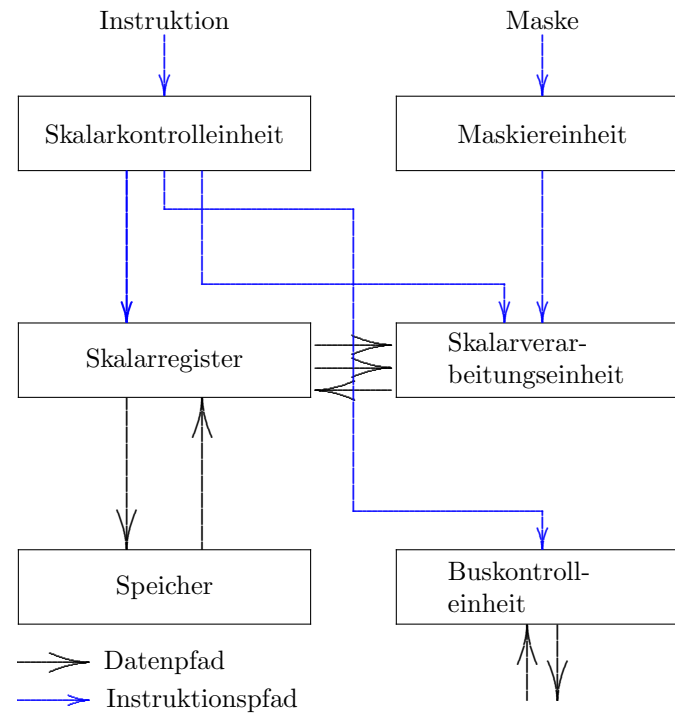
Feldprozessoren

- einen Prozess auf verschiedene Mengen von gleichförmigen Daten anwenden
- einer Instruktionseinheit instruiert Befehl
- mehrere nebenläufige spezialisierte CPUs
- Abarbeitung des Befehls im Gleichtakt auf eigenes Datum

Feldprozessoren



Feldprocessorverarbeitungseinheit



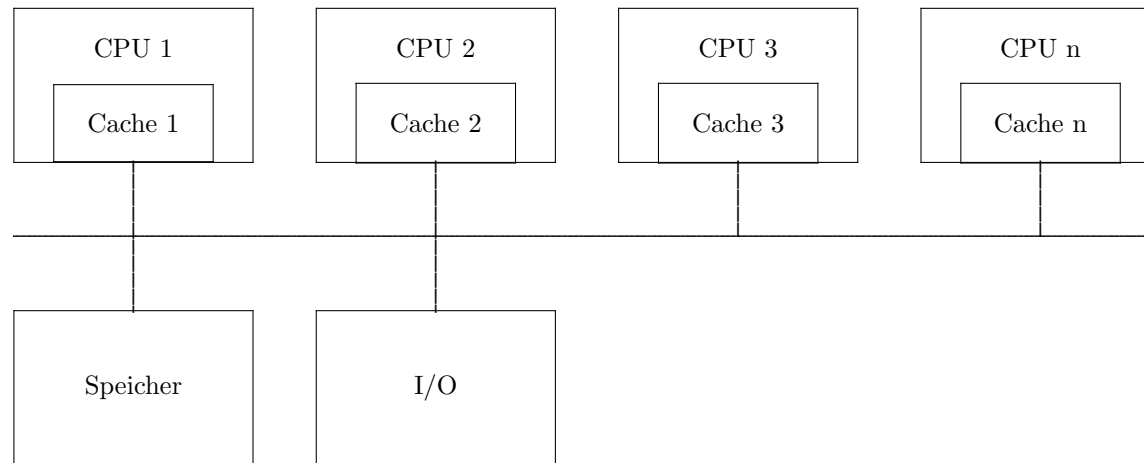
systemisches Array

- besondere Form des Feldrechners
- zwei- oder dreidimensionale Verknüpfung von VE
- taktsynchrones Pipelineverfahren

Busbasierte Multiprozessorsysteme

- n Prozessoren, 1 Speichermodul, verbunden über Bus
- **Problem:** Bus ist Flaschenhals, skaliert suboptimal
- Lösung: lokaler Cache speichert Blöcke zwischen

Cache



Cache

- **Problem:** Cachekoherenz
- Datum in allen Speichern/Caches gleich
- Lösung 1: Write-Through-Cache
- Datum wird sofort in alle Speicher geschrieben
- **Problem:** Schreibperformance leidet
- Lösung 2: Snoopy Cache
- jeder Cache überwacht Bus auf Schreiboperation
- bei Treffer eigener gecacheter Daten wird aktualisiert

Cache

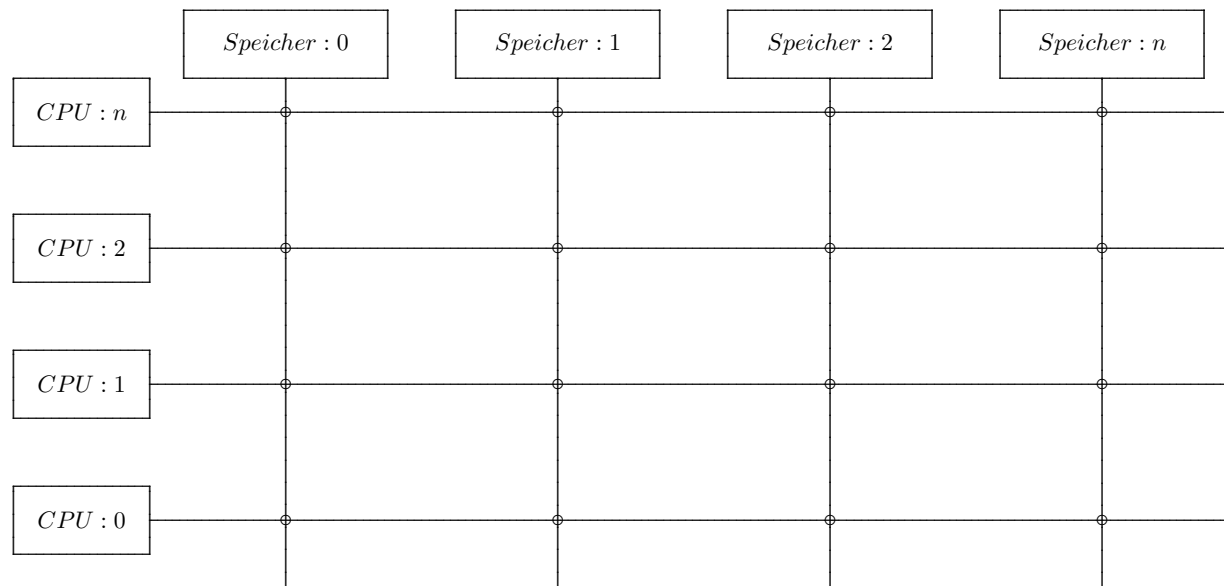
- Lösung 3: Write-Back-Cache
- Datum wird nicht geändert, sondern als geändert gekennzeichnet
- Lösung 4: MESI
- Modified: Cache geändert, Hauptspeicher ungültig
- Exclusive: Zeile liegt nur im Cache und ist unverändert
- Shared: Zeile liegt auch in anderen Caches
- Invalid: Zeile nicht im Cache verfügbar

Schalterbasierte Multiprozessorsysteme

- Verschaltung von Speicher und CPUs, mehr als ein Bus
- Kreuzschienenverteiler (Crossbarswitch)
- Omeganetzwerk

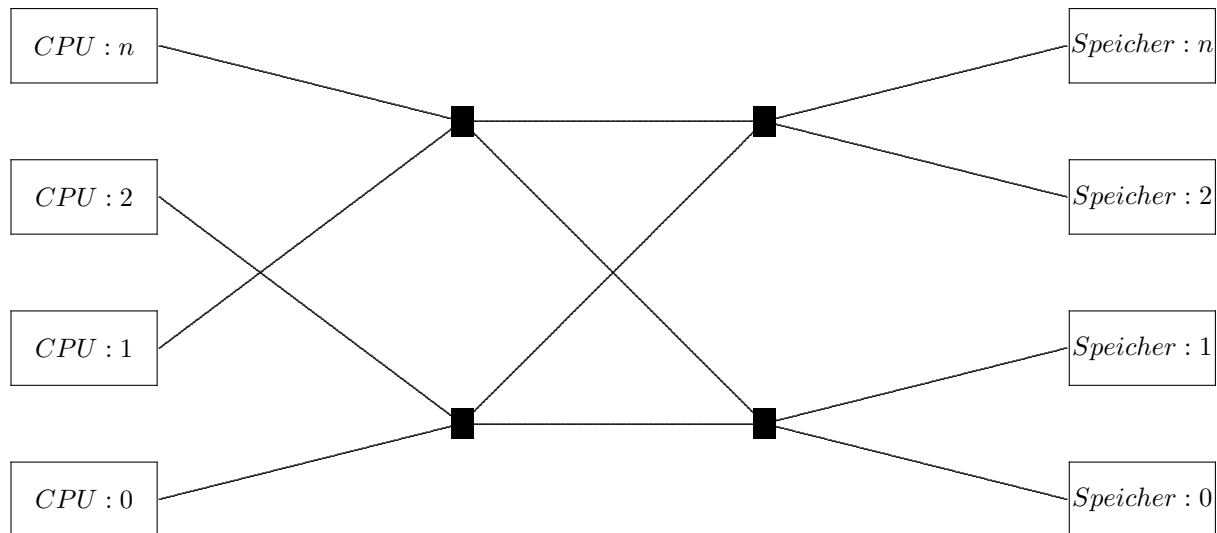
Kreuzschienenverteiler

Kreuzschienenverteiler in 4x4 Matrix



Omega Netzwerk

Omeganetz mit 4 2x2-Schaltern



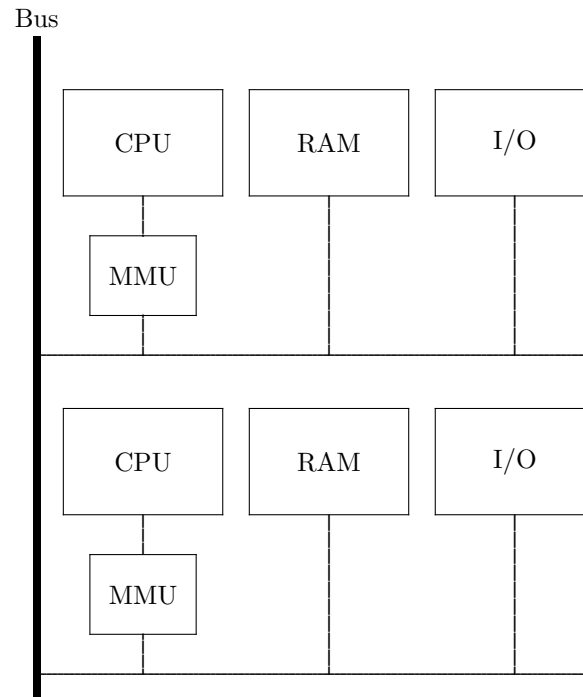
UMA

- UMA : Uniform Memory Architecture
- alle CPU mit Speicher verbunden
- siehe Einprozessorsysteme
- gleichförmiger Zugriff auf Speicher
- geringe Latenzzeit
- Beispiel: SMP

NUMA

- NUMA : Non Uniform Memory Architecture
- nichtgleichförmiger Zugriff auf lokalen oder entfernten RAM
- Zugriff auf entfernten RAM möglich, aber langsam
- Beispiel: CM*
- Cluster aus CPU, MMU, I/O und RAM
- per Bus verbunden

CM*



ccNUMA

- ccNUMA : cache coherent NUMA
- erweitern NUMA um koheränten Caches
- Snooping Caches, belasten Bus
- verzeichnisbasierte Multiprozessoren
- Datenbank über Cacheverteilung

COMA

- COMA: Cache Only Memory Architecture
- RAM einer CPU als lokaler Cache
- physischer Adressraum → Zeilen
- kann im gesamten System migrieren,
benötigt spezielle Speicherverwaltung