Das MEMSY Experimentiersystem

Wolfgang Stukenbrock

Thomas Thiel

Stefan Turowski

IMMD IV

TR-I4-92-11.slides: 27.03.95 13:27

Querschnittsthema MEMSY des SFB 182

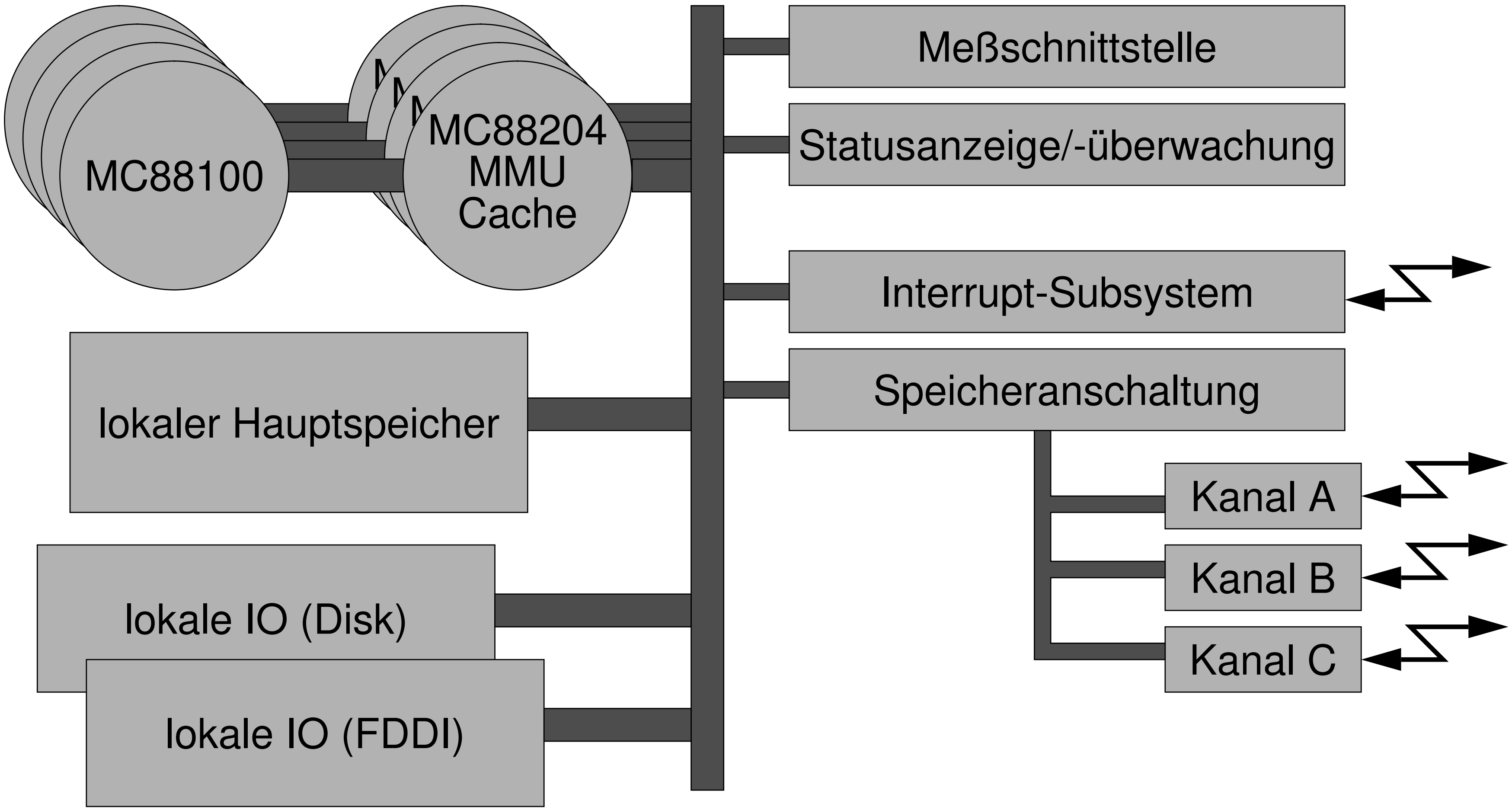
Design und Aufbau	IMMD III und IMMD IV
Zusätzliche HW-Komponenten	IMMD III
Betriebssystem, Laufzeitsystem, Simulator, Debug-Hardware	IMMD IV
Messungen	IMMD VII
Optisches Bussystem	Lst. für Angewandte Optik
Bisherige Anwendungen	IMMD III IMMD IV Lst. für Strömungsmechanik Lst. für Theoretische Chemie

Überblick

- ❑ Hardware Architektur
- ❑ Programmiermodell
- ❑ Betriebssystem Architektur

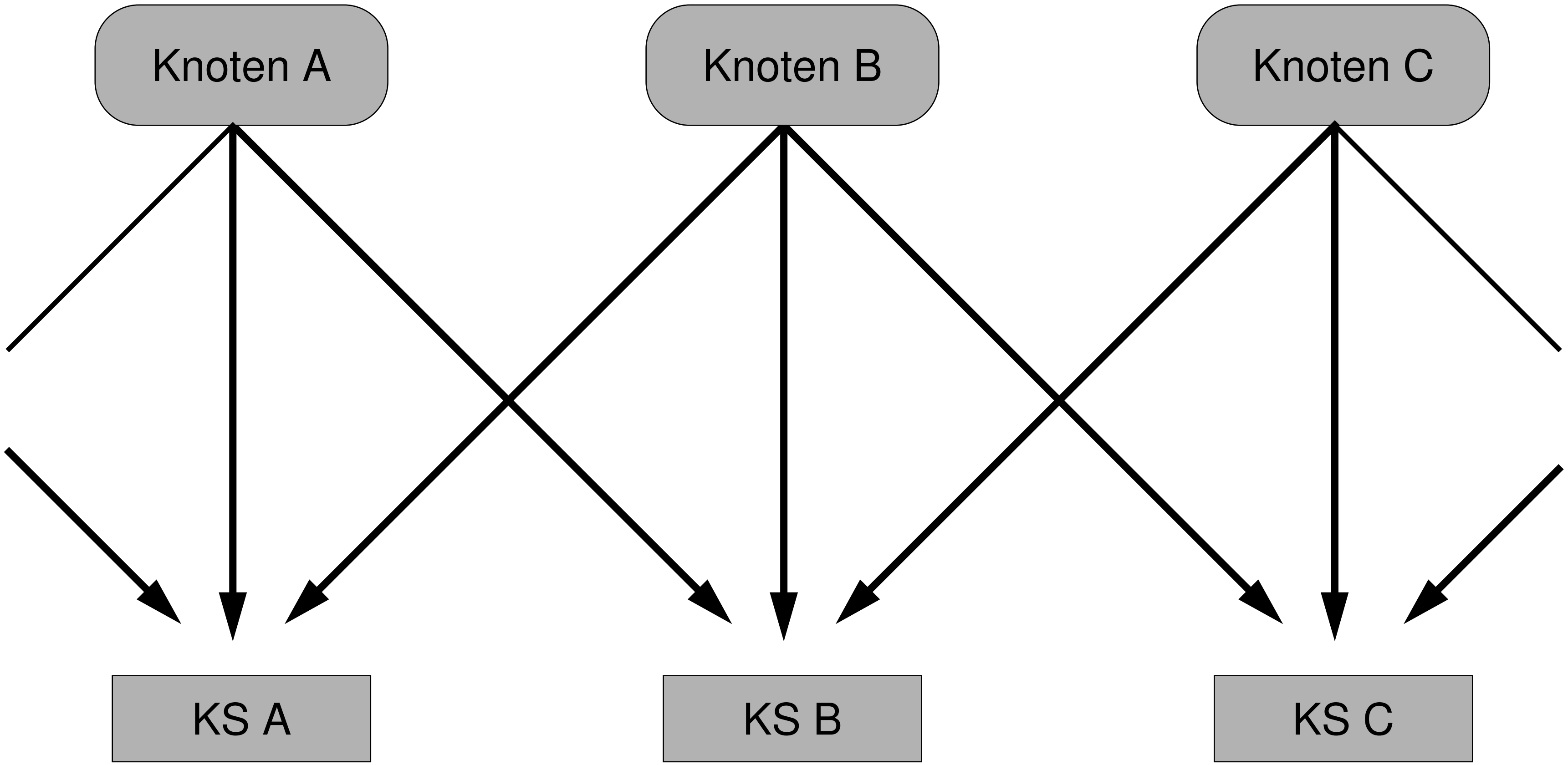
MEMSY - Hardware Architektur

- Aufbau eines MEMSY-Knotens



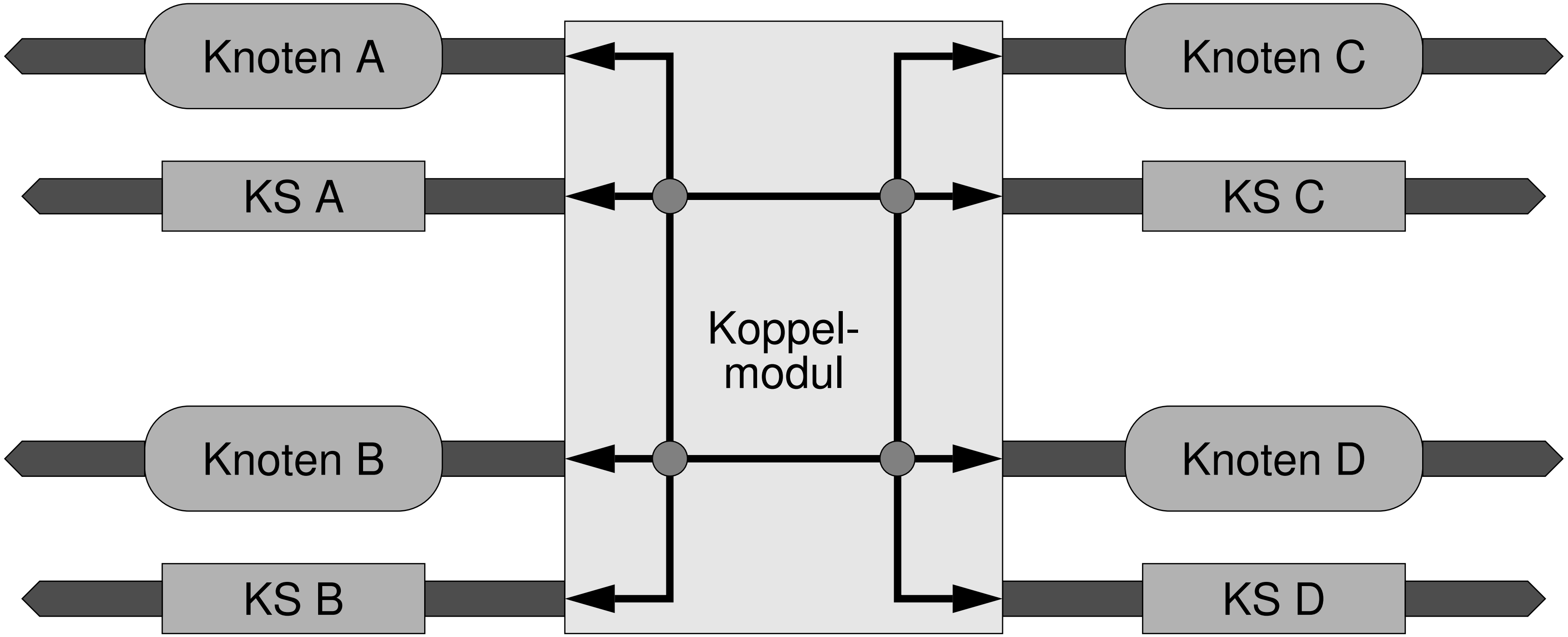
MEMSY - Hardware Architektur

- Prinzip der Speicherkopplung



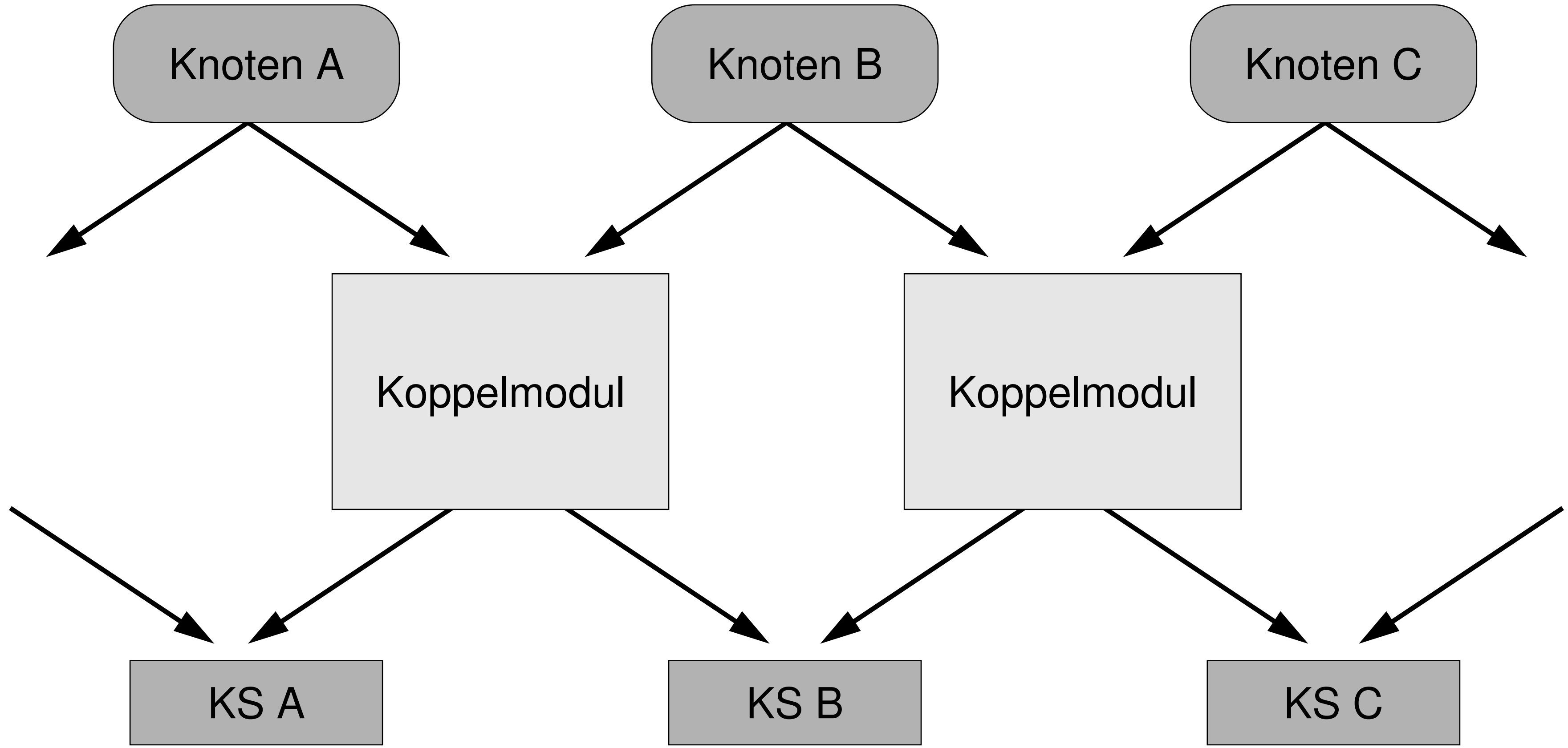
MEMSY - Hardware Architektur

❏ Speicherkopplung innerhalb der Ebene



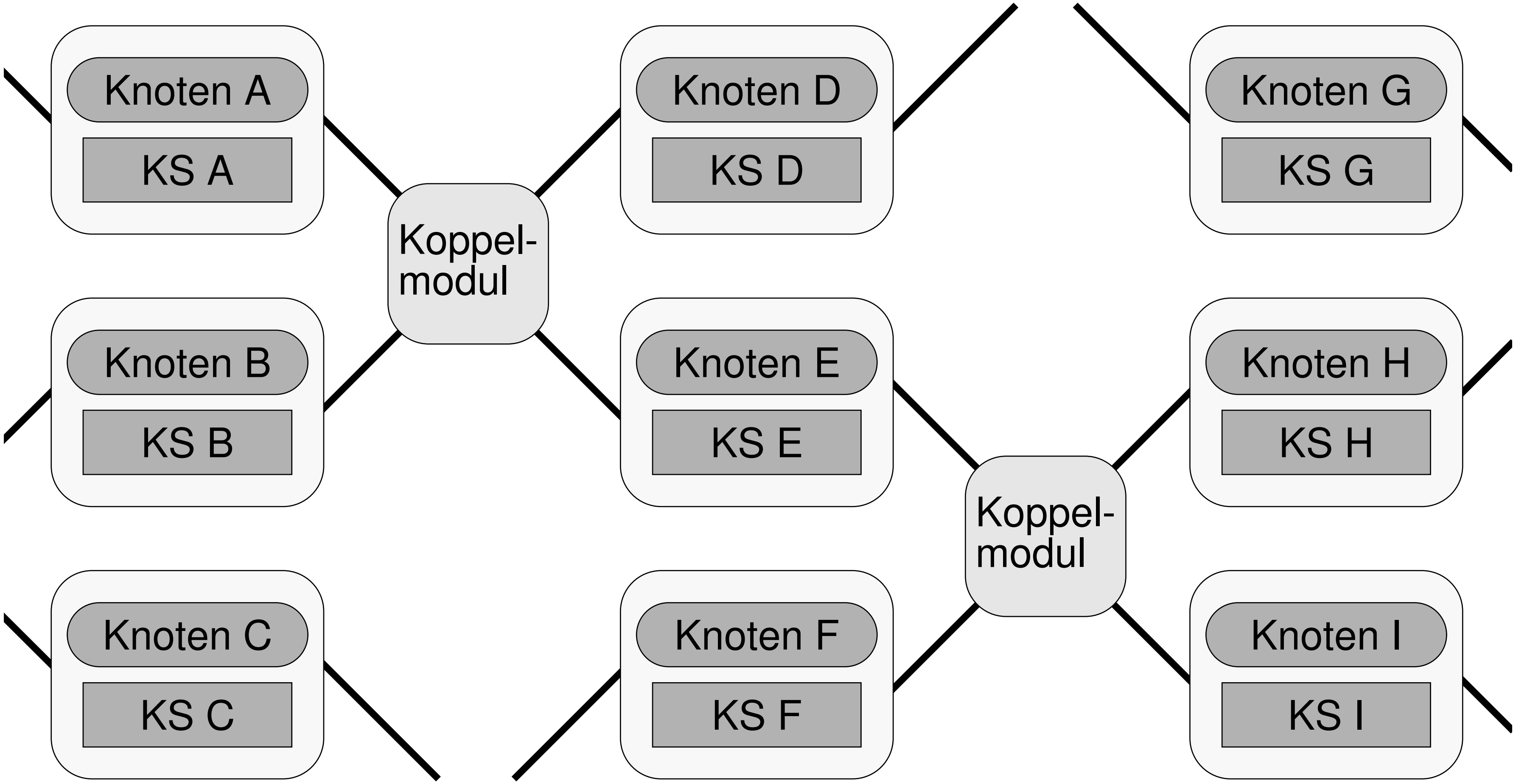
MEMSY - Hardware Architektur

- Realisierung der Speicherkopplung



MEMSY - Hardware Architektur

❏ Realisierung der Speicherkopplung in der Ebene (Draufsicht)



MEMSY - Hardware Architektur

❏ Prozessor MC88100/MC88204

- 25 MIPS (peak) ca. 18 MIPS Dauerleistung
- 12,5 MFLOPS (peak) ca. 10 MFLOPS Dauerleistung
- 64 kByte physikalischen Cache für Daten (Konsistenz-Protokoll in HW)
- 64 kByte physikalischen Cache für Code (Konsistenz-Protokoll in HW)

❏ Leistungsdaten je Knoten (4 CPU’s, lokale Festplatte, FDDI-Interface)

- 100 MIPS (peak) ca. 72 MIPS Dauerleistung
- 50 MFLOPS (peak) ca. 40 MFLOPS Dauerleistung
- 32 MByte lokaler Hauptspeicher (auf 192 MByte ausbaubar)
- 4 MByte Kommunikationsspeicher (auf 16 MByte ausbaubar)
- 512 MByte Festplatte (IO-Bandbreite 5 MByte/s - auf 15 MByte/s ausbaubar)
- FDDI-Interface mit 100 MBit/s Bandbreite

MEMSY - Hardware Architektur

❏ Leistungsdaten 20iger System

- 2000 MIPS (peak) ca. 1440 MIPS Dauerleistung
- 1000 MFLOPS (peak) ca. 800 MFLOPS Dauerleistung
- 10 MByte physikalischen Cache
- 640 MByte lokaler Hauptspeicher (auf 3840 MByte ausbaubar)
- 80 MByte Kommunikationsspeicher (auf 320 MByte ausbaubar)
- 10 GByte lokale Plattenkapazität (nahezu beliebig ausbaubar)
  - weitere Plattenspeicher auf dem Vor-/Steuerrechner
- 100 MByte/s IO-Bandbreite zu den Festplatten (auf 300 MByte/s ausbaubar)
- 100 MBit/s Bandbreite (FDDI) zum Datentransport von und zu dem System
  - Vervielfachung der Bandbreite durch den Einsatz mehrerer FDDI-Ringe
- optisches Experimental-Bussystem (z.Zt. vorraussichtlich 5 MBit)
- 400 MByte/s Bandbreite zu den lokalen Hauptspeichern
- 177 MByte/s Bandbreite zu den Kommunikationsspeichern

Programmiermodell

- ❑ Programmiersprachen C und C++
- ❑ explizite Verwaltung des Kommunikationsspeichers durch die Anwendungen
- ❑ Adressierung der Knoten relativ über Himmelsrichtungen oder absolut
- ❑ Angebot an verschiedenen Kommunikationsmechanismen
  - Shared Memory
  - Messages
  - Signale
  - Spinlocks
  - Semaphore
  - Transport
- ❑ Simulator

Programmiermodell

- ❑ Shared Memory
    - Create + Attach Mechanismus (Segmente)
    - Zugriff über Zeiger (C!)
    - dynamisches Erzeugen und Vernichten von Segmenten
    - Cache Kontrolle
- ❑ Messages
    - kurze (2 Worte) Nachrichten
    - blockierend oder nicht blockierend
    - Absender kann festgestellt werden
    - für Koordinierungszwecke
- ❑ Signale
    - Unterbrechungen zur Behandlung von Ausnahmesituationen

Programmiermodell

- ❑ Spinlocks
  - Koordinierungsvariablen im Kommunikationsspeicher
  - implementiert durch spezielle Prozessorinstruktion (XMEM)
  - “busy-wait” - für kurze kritische Abschnitte
  - timeout - Mechanismus
- ❑ Semaphore
  - globale Koordinierungsvariablen
  - implementiert durch das Betriebssystem
  - kein “busy-wait”
- ❑ Transport
  - effizienter Transport von größeren Datenmengen
  - implementiert durch das Betriebssystem
  - dynamische Auswahl der besten Alternative

Programmiermodell

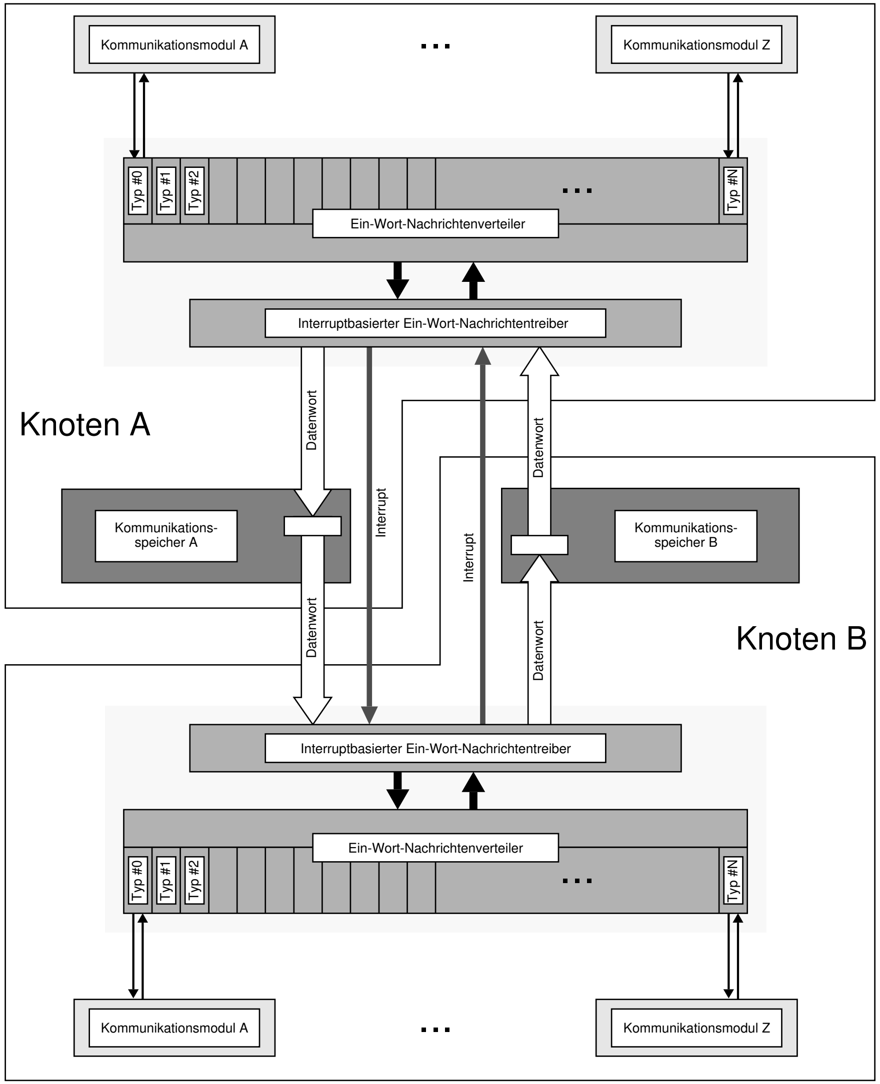
- Informationsmechanismen
  - Prozessoridentifikation und Position im Feld
  - Systemzustand
- I/O
  - normale UNIX-I/O (fopen, fread, fwrite, ...)
  - knotenlokale Datenbereiche
  - globale Datenbereiche
- fork
- Simulator
  - Testen der Anwendungen auf einer Sequent S2000/700 mit 16 Prozessoren

Betriebssystem Architektur

- ❑ Standard UNIX System V3.2 von MOTOROLA (multiprozessorfähig)
- ❑ Erlanger Erweiterungen:
  - Verwaltung des Kommunikationsspeichers
  - Interrupt und Nachrichtenmechanismus
  - Unterstützung der Funktionen des Programmiermodells
  - Anwendungsunterstützung (mehrere Anwendungen gleichzeitig!)
  - Fehlererkennungs- und Fehlerbehebungsmaßnahmen
  - erweitertes Scheduling

Betriebssystem Architektur

Interruptmechanismus

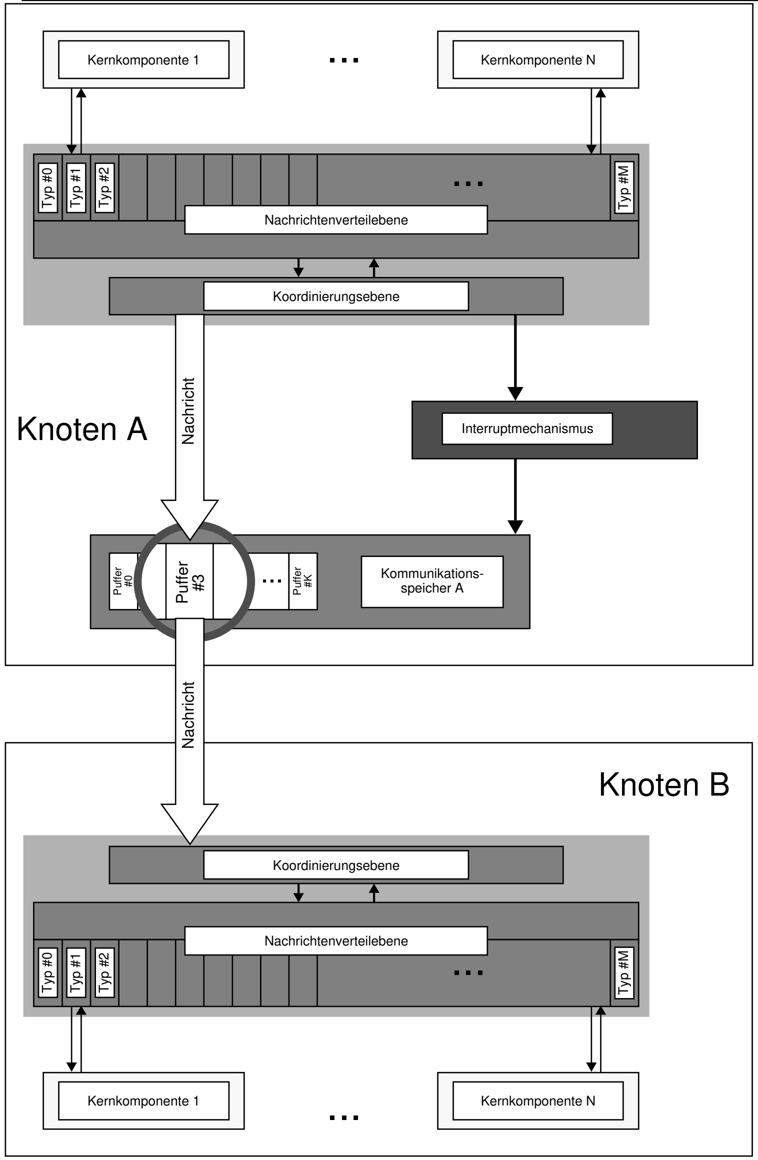


- Ein-Wort-Nachrichtenverteiler
  - Flußkontrolle
  - Zustellung der Datenworte
  - Statusmeldungen
  - statistische Information

- Ein-Wort-Nachrichtentreiber
  - Verbindungsaufbau
  - Verbindungsüberwachung
  - Pufferung (zur Entkopplung)

BS Architektur

Kommunikation durch Message-Passing
(Kommunikationsmodul INC)



- ❑ Nachrichtenverteilebene
  - Pufferung empfangener Nachrichten
  - Verteilung empfangener Nachrichten
- ❑ Koordinierungsebene
  - Verbindungsüberwachung
  - Flußkontrolle
  - Sicherstellung der Resource-Konsistenz
  - Verwaltung der Nachrichtenpuffer
  - Statusmeldungen
  - statistische Informationen

Betriebssystem Architektur

- ❑ Erweiterung des SYSTEM V Schedulingmechanismus durch eine zusätzliche *Applikationswarteschlange*
  - Applikationsprozesse werden ausschließlich in der Applikationswarteschlange eingetragen
  - Alle sonstigen UNIX-Prozesse werden in der ”Standard”-Systemwarteschlange eingetragen
  - Prozesse mit Systempriorität werden vor allen anderen Prozessen bedient
  - Lauffähige Prozesse der Applikation mit der höchsten Applikationspriorität werden bevorzugt

Stand der Arbeiten (Juli 1992)

- ❑ 4er System lauffähig mit Testanwendungen
- ❑ 20er System im Aufbau - geplant lauffähig Ende September