

# Neurocomputer-Architekturen

Innovative Rechnerarchitekturen  
Matthias Jauernig (B. Sc.), 12.06.07

# Überblick

---

1. Künstliche Neuronale Netze
2. Möglichkeiten der Parallelisierung
3. Neurocomputer-Architekturen
  - ▶ Klassifizierung
  - ▶ Kurzbeschreibung der einzelnen Klassen
  - ▶ Konkretes Beispiel: CNAPS-1064 Neuro-Chip
4. Zusammenfassung und Ausblick



---

# (1) Künstliche Neuronale Netze

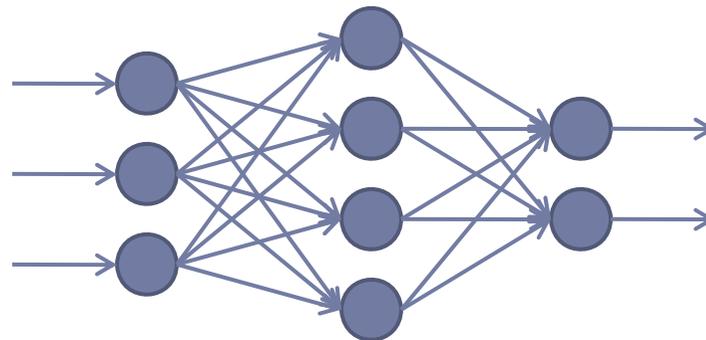
---



# KNNs allgemein

---

- ▶ Allg.: KNN = gerichteter Graph:
  - ▶ Kanten: gewichtet → Gewichte = implizites Wissen
  - ▶ Knoten: „Neuronen“ → primitive signalverarbeitende Elemente (Prozessoren)
- ▶ Beispiel: Vorwärtsgerichtetes Netz

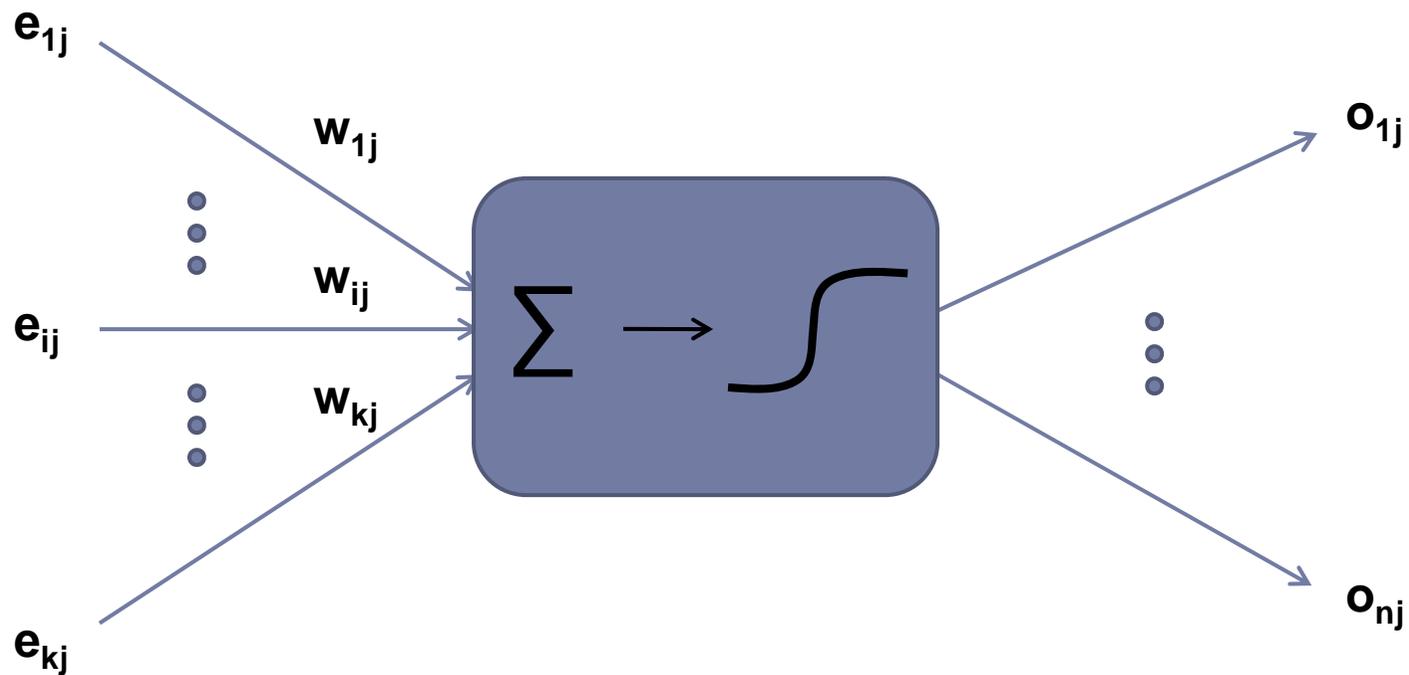


(1)

## KNNs allgemein

---

- ▶ Eingabe-/Ausgabe-Charakteristika eines Neurons  $j$ :



# KNNs allgemein

---

- ▶ **Unterteilung in Lernphase und Arbeitsphase**
  - ▶ **Lernphase:** überwachtes Lernen → KNN wird mit bekannten Paaren von Ein-/Ausgabedaten gefüttert, „lernt“ durch Fehler-Minimierung Eingaben zu Ausgaben zuzuordnen (am bekanntesten: Backpropagation)
    - Hoffnung: wenn es für die Trainingsdaten klappt, dann auch für bislang unbekannte Daten
  - ▶ **Arbeitsphase:** dem KNN wird eine Eingabe präsentiert, es ermittelt durch Verarbeitung dieser Eingabe eine Ausgabe



# KNNs allgemein

---

- ▶ Anwendungsgebiete (Auszug):
  - ▶ Optimierung
  - ▶ Robotik
  - ▶ Vorhersagen (Wetter, Börsenkurse etc.)
  - ▶ Klassifikations- / Erkennungs-Aufgaben:
    - ▶ Bilderkennung: Nummernschilder
    - ▶ Texterkennung: Buch-Digitalisierung, Unterschriften-Erkennung
    - ▶ Gesichtserkennung
    - ▶ Spracherkennung
    - ▶ Freund-/Feind-Erkennung im Militär
    - ▶ Erkennung von Kreditkarten-Missbrauch



# Ziel von Neurocomputern

---

- ▶ Parallelisierte Abarbeitung von Berechnungen (vgl. Vorbild Gehirn)
- ▶ Geschwindigkeitssteigerung im Lernprozess sowie im Arbeitsprozess
- ▶ Einsatz von KNNs in Systemen, wo Geschwindigkeit relevant ist (z.B. in Bereichen der Bildanalyse, Gesichtserkennung, Unterschriftenerkennung, Spracherkennung etc.)



---

## (2) Möglichkeiten der Parallelisierung

---



## Parallelisierungsmöglichkeiten

---

- ▶ Hier am Beispiel von vorwärtsgerichteten Netzen
- ▶ KNNs ideal für Implementierung auf Parallelrechnern
  - ▶ Große Zahl primitiver Neuronen und Verbindungen
  - ▶ Vorwärtsgerichtete Netze sehr regelmäßig → einfache Abbildung auf Parallel-Architekturen möglich
- ▶ Einschränkungen der Parallelisierbarkeit durch Datenabhängigkeiten
- ▶ Meist Kombination mehrerer Methoden, um hohe Auslastung zu erreichen und Flaschenhalse zu vermeiden

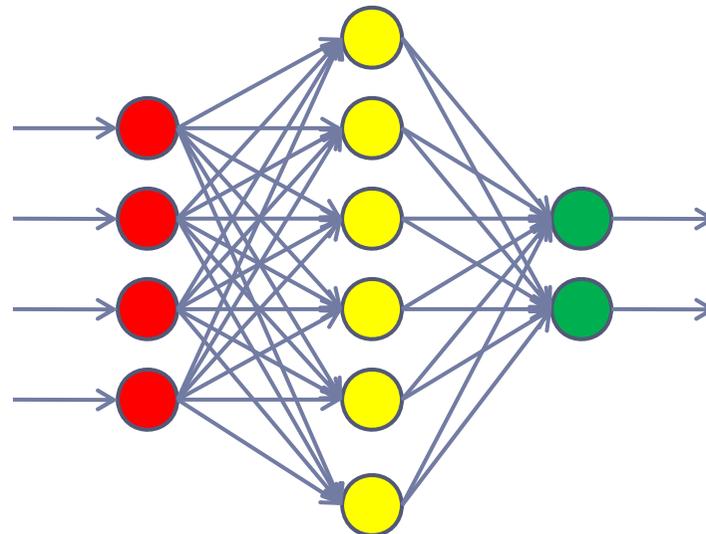


# Parallelisierungsmöglichkeiten

---

## ▶ Knoten-Parallelität:

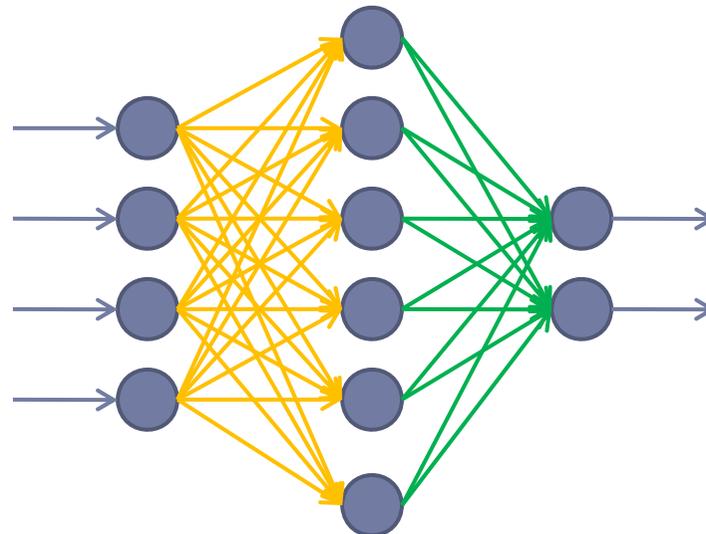
- ▶ Parallele Berechnungen der Aktivierungen und Ausgaben von Knoten einer Schicht



# Parallelisierungsmöglichkeiten

---

- ▶ **Kanten-Parallelität:**
  - ▶ Parallele Berechnungen der Kantenwerte

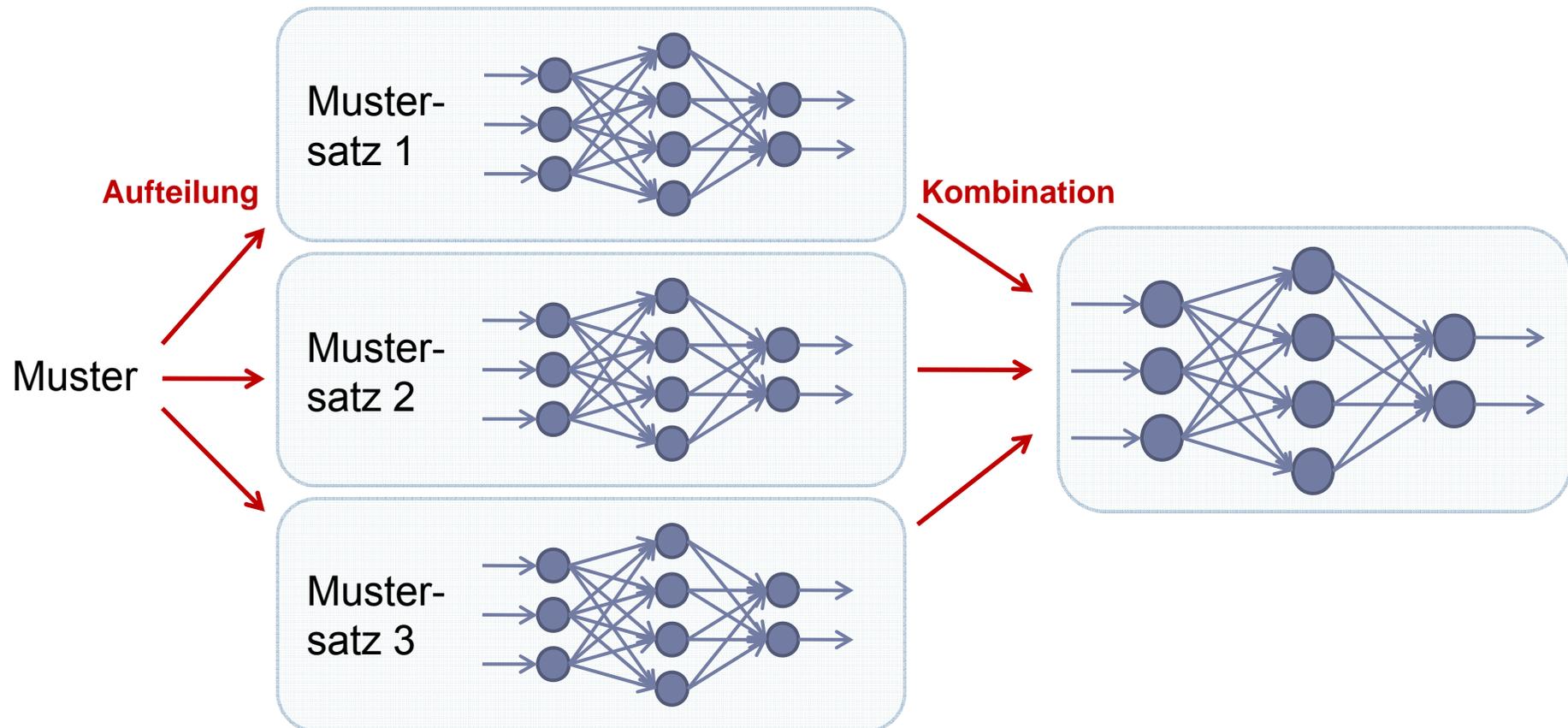


# Parallelisierungsmöglichkeiten

---

## ▶ Trainingsmuster-Parallelität:

- ▶ Kopien des Netzwerks, jede Kopie mit unterschiedlichen Mustern angelernt, zum Schluss Kombination der KNNs



---

## (3) Neurocomputer-Architekturen

---



# Neurocomputer-Architekturen

---

## ▶ Klassifizierung:

- ▶ Universalrechner
- ▶ MIMD/SIMD-Parallelrechner
- ▶ Coprozessor-Boards
- ▶ Neurocomputer aus Standardbausteinen
- ▶ VLSI-Neurocomputer mit Spezialhardware
- ▶ VLSI-Neuro-Chips (digital/analog)



**Steigende  
Spezialisierung**



# Neurocomputer-Architekturen

---

- ▶ **Geringe Spezialisierung:**
  - ▶ Universal einsetzbare Systeme
  - ▶ Meist Gleitkommaarithmetik
  - ▶ Hohe Genauigkeit
- ▶ **Mit steigender Spezialisierung:**
  - ▶ Dedizierte Systeme
  - ▶ Leistungsfähigere replizierte Hardware
  - ▶ Festkommaarithmetik mit 4-16 Bits
  - ▶ Geringere Genauigkeit
  - ▶ i.d.R. steigende Verarbeitungsleistung



# Neurocomputer-Architekturen

---

- ▶ **Coprozessor-Boards:**
  - ▶ Unterstützen einen Host-Rechner bei Berechnungen in neuronalen Netzen
  - ▶ Verbindung mit Host als Steckkarte, über Peripherie-Schnittstelle oder über ein Netzwerk
  - ▶ Beispiel: Nestor/Intel Ni1000 Recognition Accelerator
    - ▶ ISA-Board
    - ▶ Ursprünglicher Preis: 10.000 \$
    - ▶ Kernstück ist digitaler VLSI-Neuro-Chip Ni1000



# Neurocomputer-Architekturen

---

- ▶ **Neurocomputer aus Standardbausteinen:**
  - ▶ Komplette Rechner, meist mit mehreren Boards, die wiederum mehrere Prozessoren beinhalten
  - ▶ Beispiele: ICSI RAP, Fujitsu Neurocomputer: beinhalten Standard-Signalprozessoren, die durch eine Ring-Topologie miteinander verbunden sind



# Neurocomputer-Architekturen

---

- ▶ **VLSI-Neurocomputer mit Spezialhardware:**
  - ▶ Beinhalten spezielle Neuro-Chips: liefern noch höhere Leistungen, sind aber in ihrer Genauigkeit reduziert



# Neurocomputer-Architekturen

---

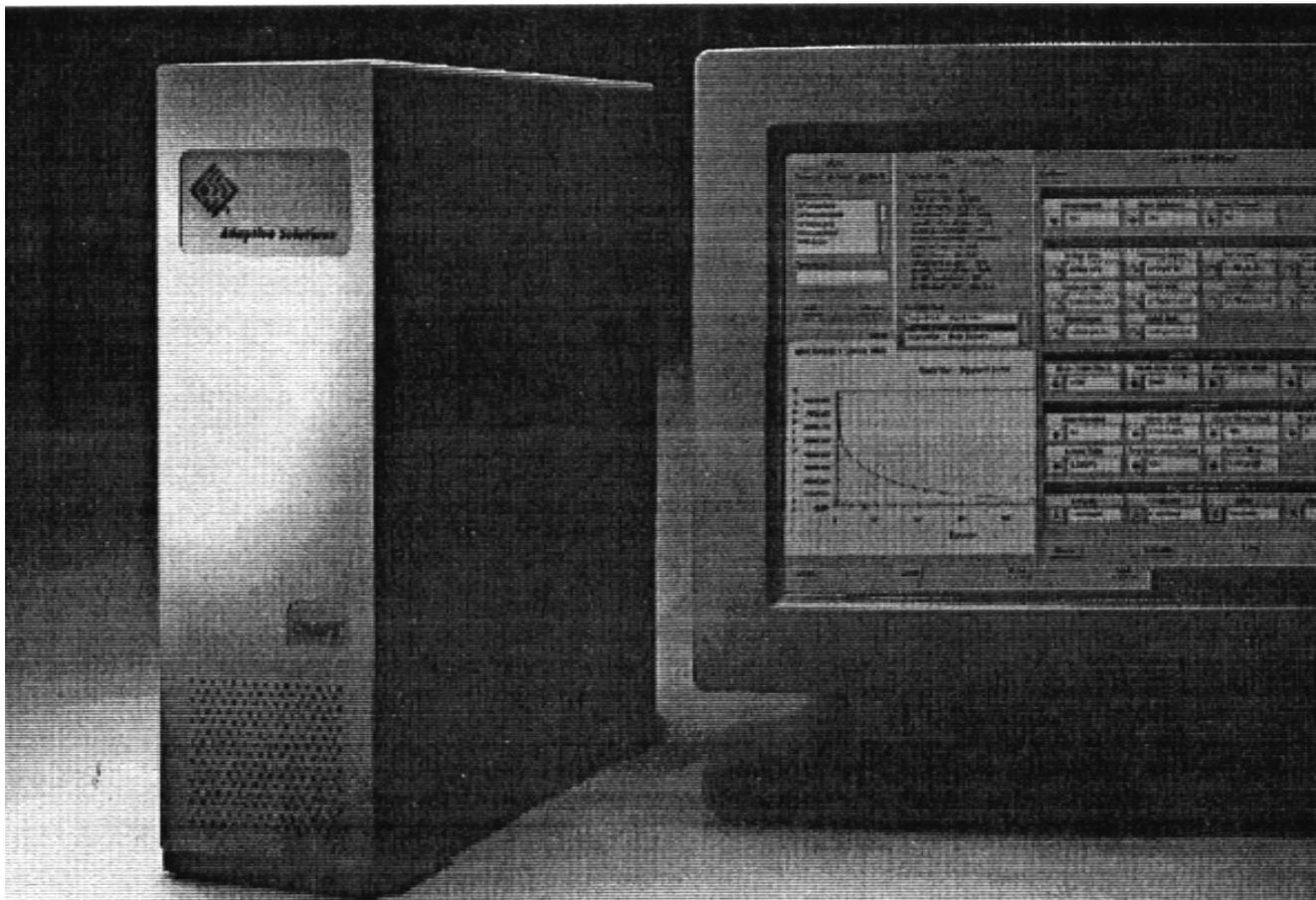
- ▶ Beispiel für VLSI-Neurocomputer: Adaptive Solutions „CNAPS Server II“
  - ▶ Taktfrequenz von 20 MHz
  - ▶ Beinhaltet ein CNAPS/VME-Board mit bis zu 4 CNAPS-1064-Neuro-Chips → jeder dieser Chips besitzt 64 PEs
  - ▶ Erweiterung auf 8 Chips mit 512 PEs möglich
  - ▶ Grafische Anwendungsumgebung vorhanden
  - ▶ Entwicklungsumgebung erlaubt Implementierung eigener Lernverfahren in einem parallelen C oder in Assembler



# Neurocomputer-Architekturen

---

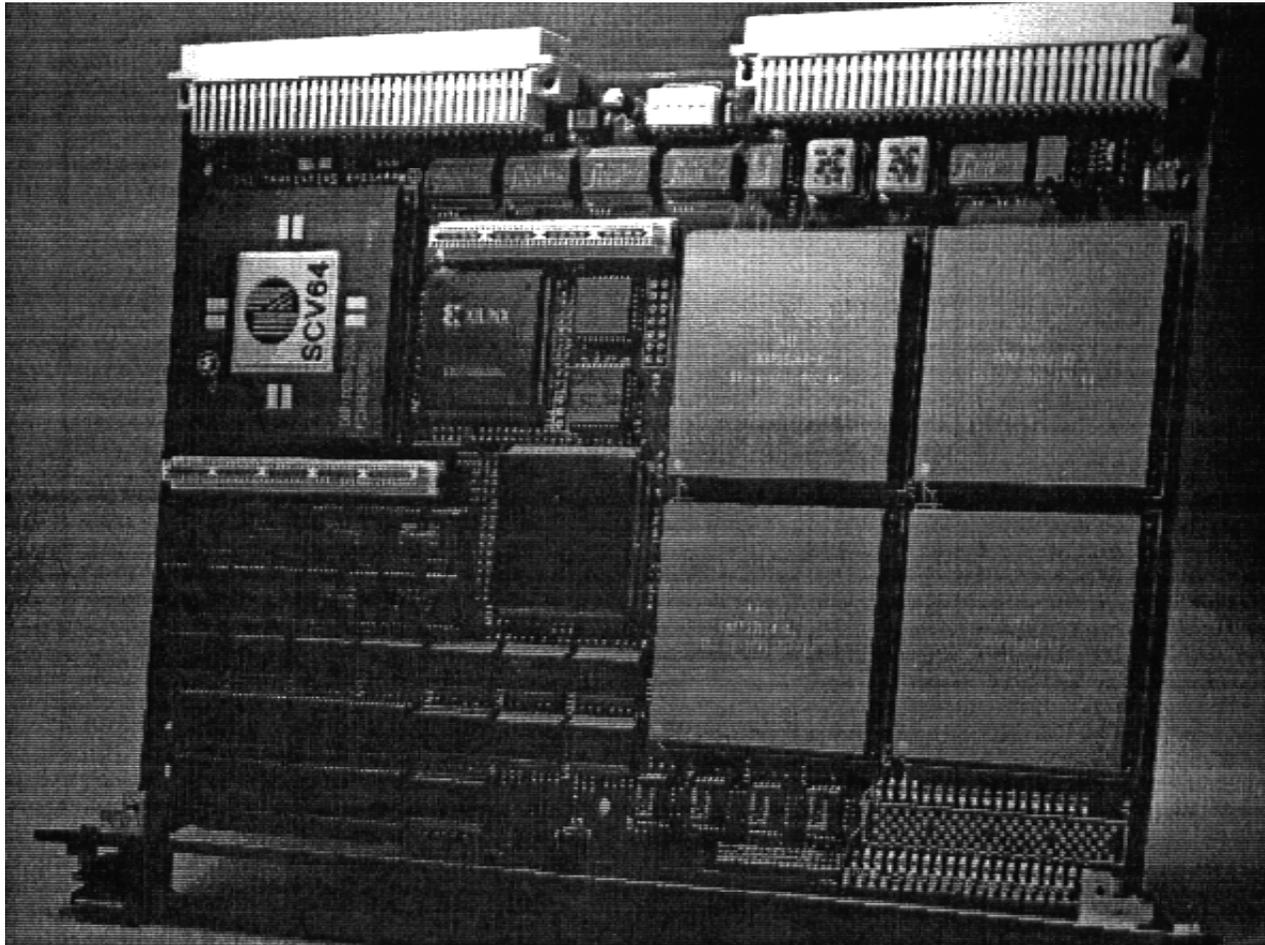
## ▶ CNAPS Server II:



# Neurocomputer-Architekturen

---

- ▶ CNAPS/VME-Board:



# Neurocomputer-Architekturen

---

- ▶ **VLSI-Neuro-Chips:**
  - ▶ Unterteilung in digital und analog
  - ▶ Digitale Chips: ab 8 Bit Genauigkeit, Flexibilität hinsichtlich verschiedener neuronaler Modelle
  - ▶ Analoge Chips: wesentliche neuronale Berechnungen und Speicherung der Gewichte sind analog realisiert, geringere Genauigkeit, aber höhere Geschwindigkeit
  - ▶ Oft statt rein analog: hybride Architekturen



# Neurocomputer-Architekturen

---

- ▶ **Beispiel für digitalen Neuro-Chip: CNAPS-1064**
  - ▶ Fertigung in CMOS ULSI-Technik (Ultra Large Scale Integration)
  - ▶ Ca. 11 Mio. Transistoren
  - ▶ Nach Fertigung 80 Prozessoren, von denen jedoch meist nicht alle fehlerfrei sind (Größe und Dichte des Chips)
  - ▶ Aus den 80 PEs werden 64 ausgewählt, die übrigen werden elektrisch getrennt (Zerstörung von Mikrosicherungen)

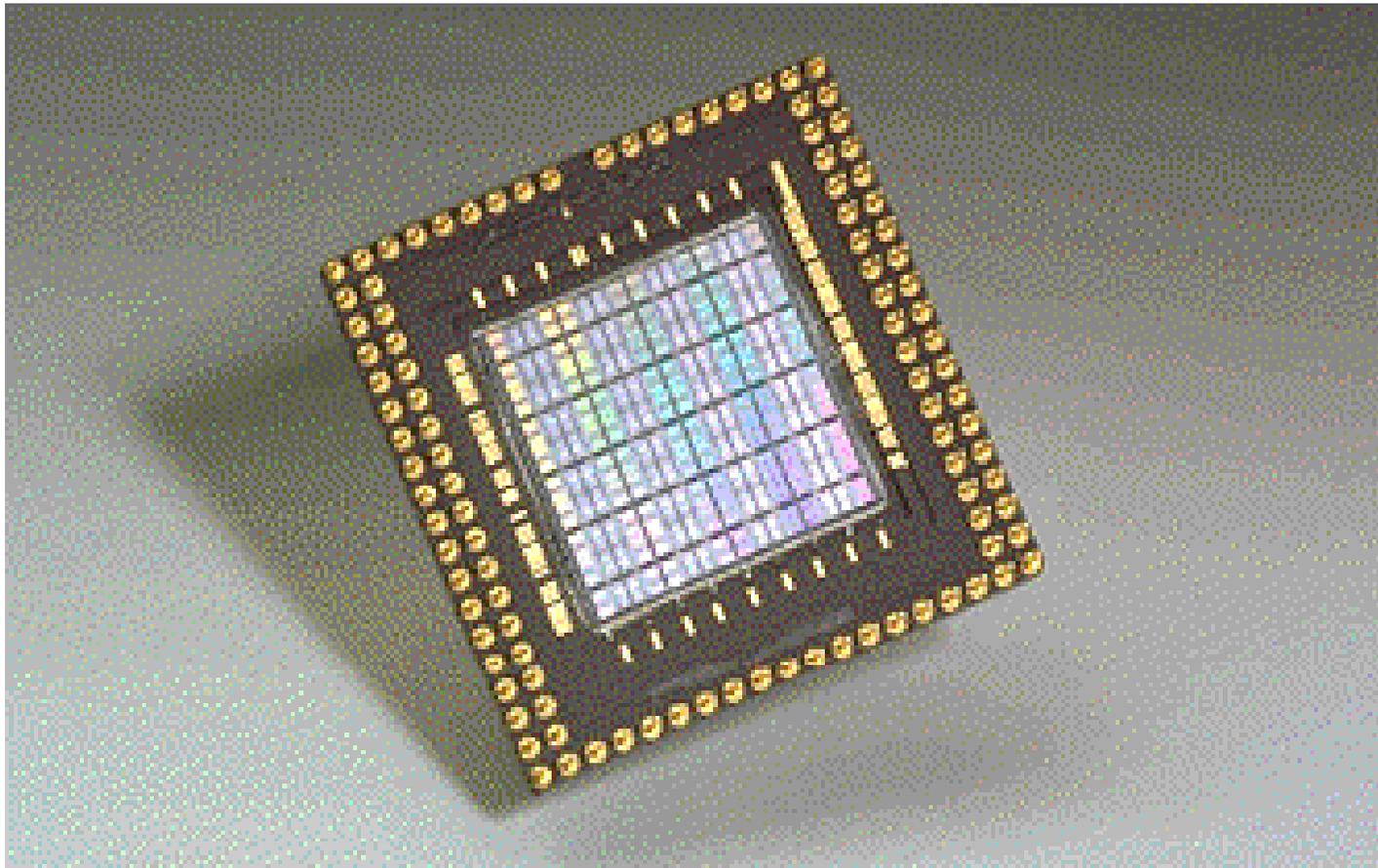


(3)

## CNAPS-1064 Neuro-Chip

---

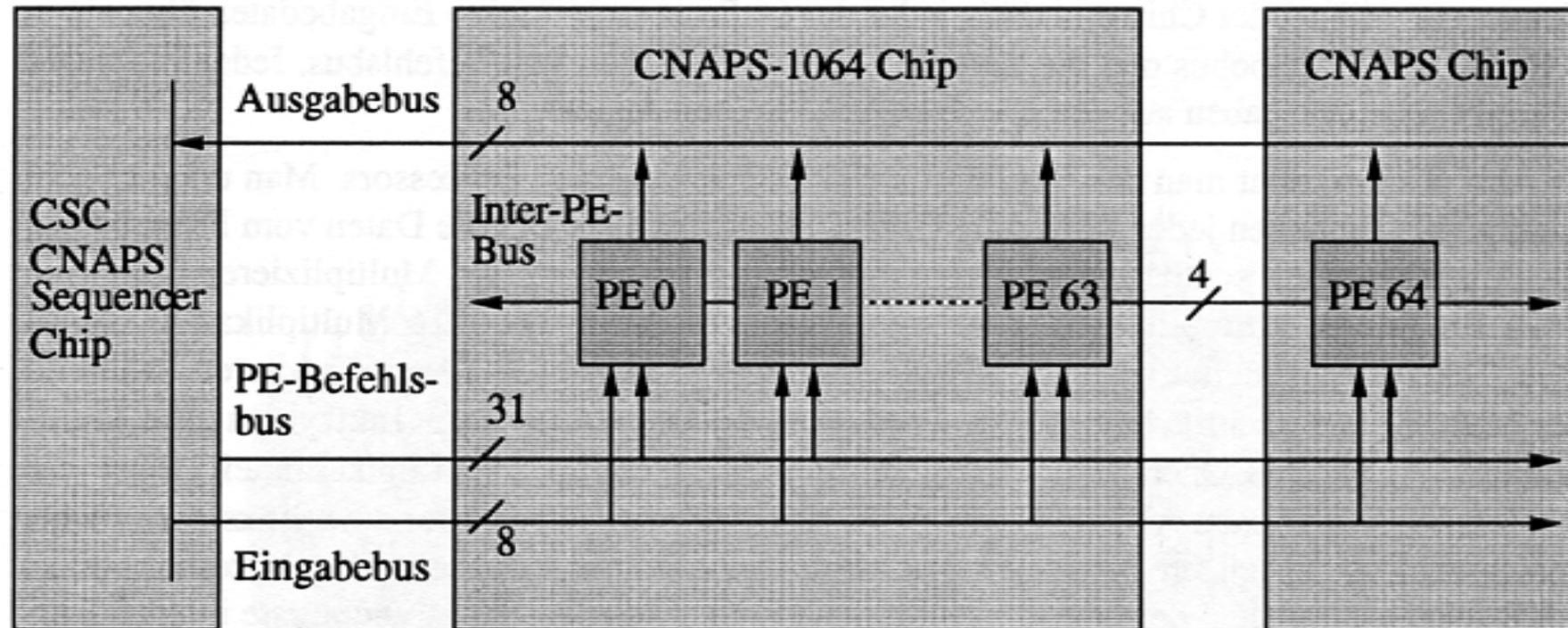
- ▶ CNAPS-1064 Chip mit 80 Prozessoren:



(3)

# CNAPS-1064 Neuro-Chip

- ▶ Schematischer Aufbau des CNAPS-1064:



## CNAPS-1064 Neuro-Chip

---

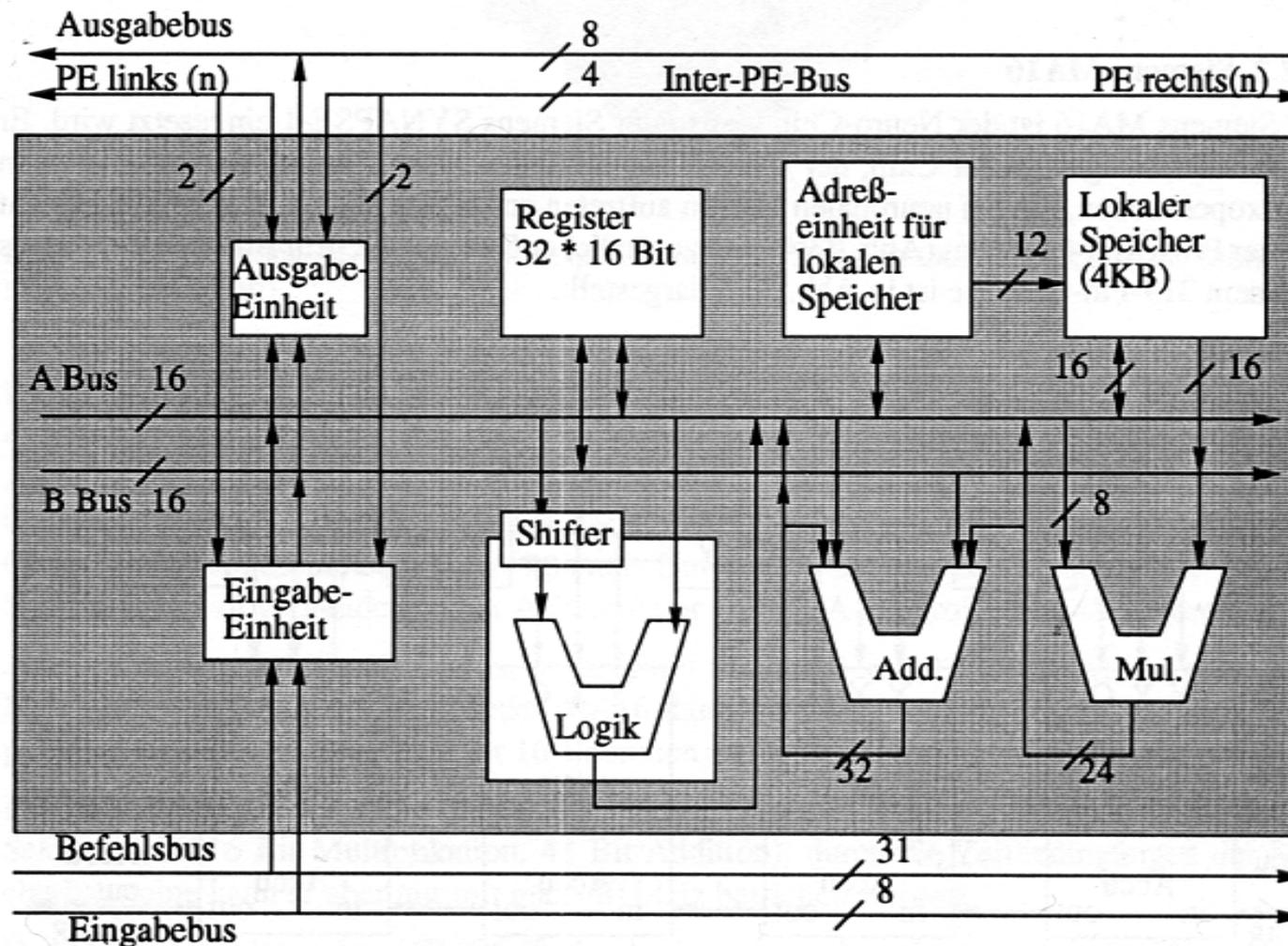
- ▶ Sequencer verteilt an jeden Chip und die 64 darin enthaltenen PEs:
  - ▶ Eingabedaten über 8 Bit Eingabebus
  - ▶ Befehle über 31 Bit Befehlsbus
- ▶ Jeder Prozessor schreibt seine Ausgaben auf 8 Bit Ausgabebus



(3)

# CNAPS-1064 Neuro-Chip

- ▶ Aufbau eines einzelnen Prozessors:



# CNAPS-1064 Neuro-Chip

---

- ▶ Lediglich Integer-Operationen
- ▶ Jede PE hat 8 funktionale Einheiten
  1. Eingabe-Einheit
  2. Ausgabe-Einheit
  3. Multiplizierer: 16-Bit-Integer-Multiplikation
  4. Addierer: 32-Bit-Integer-Addition
  5. Logik-Einheit: AND, OR, XOR, Shifting
  6. Registereinheit: 32 Register von 16 Bit Breite
  7. Lokaler Speicher: 4 KB pro PE
  8. Adresseinheit: adressiert Speicher
- ▶ A- und B-Bus: interne Busverbindungen, 16 Bit breit



---

## (4) Zusammenfassung und Ausblick

---



## Zusammenfassung

---

- ▶ KNNs sind vielseitig einsetzbar
- ▶ Ausgezeichnet für Parallelisierung geeignet
  - ▶ Verschiedenste Arten der Parallelisierbarkeit
- ▶ Vielzahl an Neuro-Architekturen
  - ▶ Unterscheiden sich im Wesentlichen durch den Grad der Spezialisierung
  - ▶ Einsatz dort, wo es auf Geschwindigkeit ankommt



# Ausblick

---

- ▶ Recherche nach aktuellen Architekturen schwierig
  - ▶ Hersteller legen Technologien nicht offen
  - ▶ Viele Neurocomputer-Firmen sind nach einem Boom Anfang der 1990er Jahre wieder vom Markt verschwunden
  - ▶ Die Entwicklung scheint vorerst stehen geblieben zu sein; Gründe:
    - ▶ Zu geringer Markt
    - ▶ Hoher Aufwand, hohe Kosten, geringe Einnahmen
    - ▶ Steigende Rechnerleistungen scheinen Neuro-Architekturen überflüssig zu machen → stimmt aber nicht: Aufwand steigt exponentiell mit Netzgröße an
    - ▶ KNN-Euphorie ist Vergangenheit



## Ausblick

---

- ▶ **Perspektivisch: Renaissance von Neuro-Arch. bei der Entwicklung intelligenter Roboter und Systeme**
  - ▶ Architekturen abhängig von den entwickelten Modellen: modular oder monolithisch
  - ▶ *Modular*: dedizierte Neuro-Rechner z.B. bei der Bilderkennung, Sprachverarbeitung, Situationsbewertung etc. – Controller-Board könnte diese verarbeiteten Sensor-Informationen kombinieren und Reaktionen auslösen → in Echtzeit notwendig
  - ▶ *Monolithisch*: gesamtheitliches Hirnmodell notwendig, massiv paralleler Neuro-Rechner denkbar



---

**Ende**

---

