

Parallel & Distributed Computing

(Vorlesung WS 2025/26)

Technische Fakultät
Uni Bielefeld

Dirk Freilöb



Parallel & Distr. Comp.

Literatur:

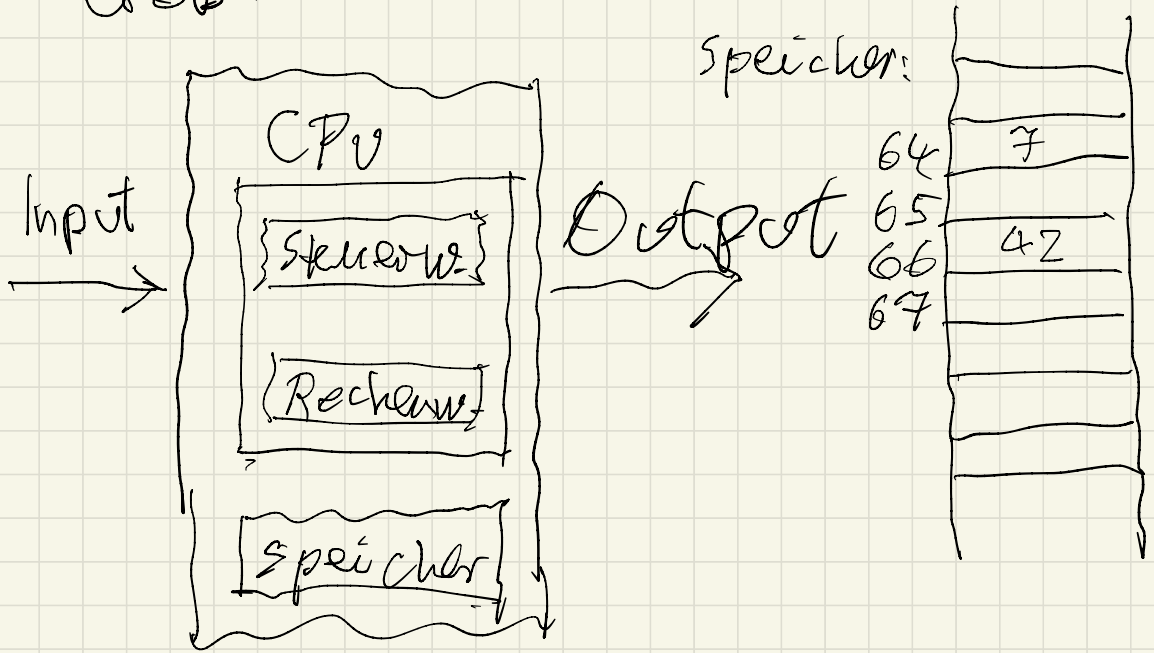
- Rauber & Rünger: Parallel Progr. (Unibibel.)
- Skript: Tautou Skript
„Parallel ~~verb.~~“ Uni Lüneburg

Plan:

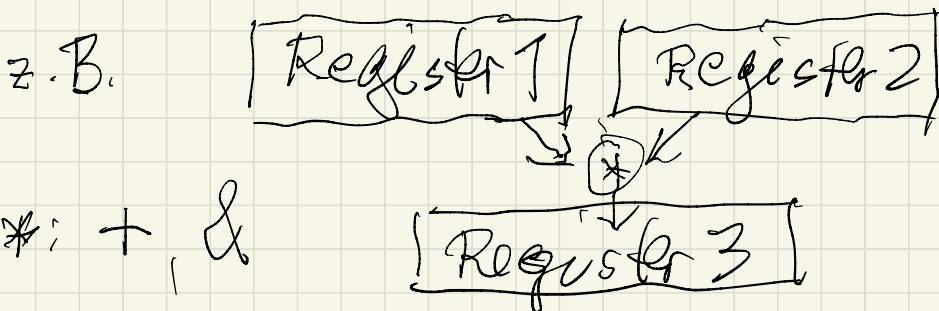
- Was ist ein Parallel-Rechner?
- Programmiermodelle
- ein Beispiel von Lösern für lineare Gl.-systeme.
- ... und andere

Kap. 1. Parallelrechner

(Viele) heutige Rechner beruhen auf der von-Neumann-Architektur
Grob:



Rechenwerk (ALU)



Steuerwerk:

regelt Programm-
ablauf

Befehlszähler

Befehlsregister

Decoder

heute im Rechner z.B.

- CPU: viel Steuerwerk,
wenig Rechenwerk
- GPU: wenig Steuerwerk,
viel Rechenwerk

Beschleunigungspotenzial:

1. Taktrate erhöhen
2. mehrere Aktionen pro Takt
3. Engpässe beseitigen
(z.B. Speicherzugriff)

zu ②: „Parallelisierung“

Idee: viele Aktionen pro Takt.

Historisch:

„Digitalrechner“: immer 0-1

also nur 1 Bit pro

Leitung / Speicherzelle...

- 1-Bit-Rechner: gab es nie

Zuse Z 3 (1943) simultan 22 bit

- 4-Bit: CPU: Intel 4040

aber eigtl. Speicher, Daten-

Leitungen: 8-Bit

- 8-Bit CPU: Intel 8080

1974 Altair 8800

(Apple II, Commodore: Moto-
rale)

Mit 8 bit kann ich aber
z.B. nur $256 = 2^8$ Speicher-
zellen (direkt) ansprechen.

16-bit ab ca 1978

aber zunächst nur in CPU,

32-bit (Intel, Motorola)

Vorteile:

- Speicheradressen bis 4 GB
- Integer $-2^{31}, \dots, 2^{31}-1$
- float 8 Dezimalstellen

Nachteile: dasselbe.

64-Bit ab 2000er

AMD vs Intel

bis zu 4 EB = 10^6 TB

Speicherzellen direkt ansprechbar.

Verbesserungen seitdem in
anderen Bereichen.

Z.B. Taktrate steigern

(z.B. Ende 80er: 2 MHz
Mitte 00er: 2 GHz
heute: 4 GHz)

physische Grenze
ist erreicht

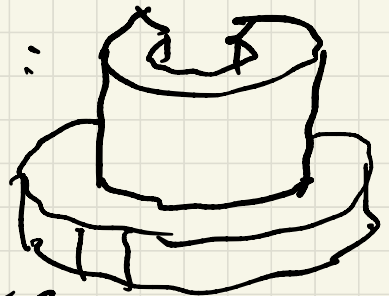
Pipelining

Vektorrechner:

Cray Supercomputer:
1976-88

• die schnellsten

Cray-2 1,9 G-Flop



1.1. Vektorrechner

intern 64-Bit, mehrere CPUs,
die je 8-Bit verarbeiten.

Gut z.B. für viele gleiche Op.
auf großen Datenmengen:

$$a_i \leftarrow a_i + 5 \quad 1 \leq i \leq 10^6$$

(dazu: Klug Speicher verwalten)

1.2 Pipelining

In real besteht 1 Befehl je
aus mehreren Schritten: z.B.

Fetch: lies Befehl ADD

Decode: dekodiere

Execute: ausführen

Write

Verteile diese Teile auf
mehrere CPUs:

