

Lernzettel

Rechnerorganisation

Universität: Technische Universität Berlin
Kurs/Modul: Rechnerorganisation
Erstellungsdatum: September 20, 2025



Zielorientierte Lerninhalte, kostenlos!
Entdecke zugeschnittene Materialien für deine Kurse:

<https://study.AllWeCanLearn.com>

Rechnerorganisation

Lernzettel: Rechnerorganisation

(1) Einordnung und Zielsetzung. Rechnerorganisation beschreibt Aufbau, Funktionsweise und Optimierung digitaler Systeme auf Mikroarchitektur- und Befehlssatz-Ebene. Sie verbindet logische Schaltungen mit der Ausführung von Programmen und erklärt, wie Maschinenbefehle in Hardware umgesetzt werden. Zentrale Aspekte sind Von-Neumann-Architektur, ISA, Mikroarchitektur, Pipeline, Speichersysteme und Ein-/Ausgabe.

(2) Von-Neumann-Rechner und Grundbausteine. Ein typischer Von-Neumann-Rechner besteht aus: - Rechenwerk (ALU), Steuerwerk (Kontrolle), Speicher (RAM), - Eingabe/Ausgabe-Geräte und Bus-Systeme. Die Grundoperationen werden durch Fetch-Decode-Execute-Schleifen realisiert, in denen Programme aus dem Speicher gelesen, dekodiert und ausgeführt werden.

Fetch: $IR \leftarrow \text{Memory}[PC]$

Decode/Execute: opcode, operanden, Kontrollsignale

(3) Befehlssatzarchitektur (ISA) und Assemblersprache. Der ISA definiert, welche Befehle vorhanden sind, wie Befehlssätze aufgebaut sind, welche Adressierungsarten unterstützt werden und wie viele Register verfügbar sind. Typische Unterscheidungen: - RISC vs. CISC: einfache, regelmäßige Befehlsformate (RISC) gegenüber komplexeren Befehlen (CISC). - Adressierungsarten: Immediate, Direct, Register, Indirect, Relative.

Adressierungsarten: Immediate ($\#imm$), Direct(Adr), Register(R_n), Indirect($[R_n]$), Relative($PC \pm \text{offset}$)

Assemblersprache verwandelt menschlich lesbare Mnemonics in Maschinencode. Wichtige Konzepte: Labels, Sprungkonstrukte, Bedingungsprüfungen, Pseudooperationen (DW, DB, ORG) und Makros.

(4) Mikroarchitektur und Registertransferebene (RTL). Auf RTL-Ebene werden Mikrooperationen beschrieben, die zwischen Registern und dem Rechenwerk stattfinden, z. B.

$R_dest \leftarrow \text{ALU}(R_src1, R_src2)$

Kontrollsignale steuern die Ausführung dieser Mikrooperationen. Zentrale Konzepte: - Registertransfer-Language (RTL) - Mikroarchitektur-Entscheidungen: Einzelzyklus vs. Mehrzyklus, Pipeline - Pipeline-Hazards: Datenhazards, Steuerhazards, Strukturhazards - Hazard-Lösungen: Forwarding, Stalling, Branch Prediction

(5) Rechenleistung, Zahlendarstellungen und Mikroalgorithmen. - SPEC-Benchmarks dienen der Leistungsbewertung moderner Systeme. - Amdahl's Law beschreibt Maximalgeschwindigkeit durch zu beschleunigende Anteile:

$$S = \frac{1}{(1 - p) + \frac{p}{s}}$$

mit p dem Anteil der beschleunigten Teilaufgabe und s dem Speedup dieses Anteils.

- Zahlendarstellungen: Ganzzahlen in Zweierkomplement, Fest- und Fließkomma.

Zweierkomplement (n Bits): $[-2^{n-1}, 2^{n-1} - 1]$

2's Kompl.: $x \mapsto x \mod 2^n$

(6) Aufbau eines einfachen Prozessor-Entwurfs. - Von-Neumann-Rechner-Verständnis: Zentrale Bausteine, Datenpfade, Kontrollen. - Mehrzyklus-Implementierung: mehrere Takte pro Befehl, flexiblere Nutzung von Ressourcen. - Einzelzyklus-Ansatz ist schneller in der Praxis oft durch begrenzte Taktpfade eingeschränkt.

(7) Fließbandverarbeitung, Pipeline und Pipeline-Konflikte. - Pipeline-Aufbau: fetch, decode, execute, memory, write-back in verschiedenen Stufen. - Vorteile: höhere Durchsatzrate bei regelmäßigem Code. - Konflikte/Hazards: - Datenhazards: Weiterverwendung von Ergebnissen aus vorherigen Befehlen. - Steuerhazards: Sprungbefehle beeinflussen die Pipelinestruktur. - Strukturhazards: Ressourcenknappheit (z. B. nur eine ALU).

Lösungen umfassen Forwarding, Branch Prediction, NOPs bzw. Stalling.

(8) Speicherhierarchie, Caches und virtueller Speicher. - Speicherhierarchie: Register → L1-Cache → L2/L3-Cache → Hauptspeicher → Massenspeicher. - Lokalisierungsprinzipien: Temporal- und Lokationslokalität. - Cache-Operationen: Hit/Miss, Cache-Coherence (bei Mehrkernsystemen). - Virtueller Speicher: Paging, Seitentabellen, Translation Lookaside Buffer (TLB).

(9) Ein-/Ausgabe (I/O), Adressierung, Synchronisation und direkter Speichzugriff. - I/O-Techniken: Speicher-mapped I/O, Programmed I/O, DMA (Direct Memory Access). - Adressierung von Peripherie: I/O-Adressräume, Ports. - Synchronisation: Interrupts, Semaphoren, Sperren. - Direkter Speichzugriff reduziert CPU-Last durch Übertragung von Daten direkt zwischen Peripherie und Speicher.

(10) Zusammenfassung und Ausblick. - Rechnerorganisation verbindet logische Schaltungen mit Maschinensprache. - Verständnis von ISA, Mikroarchitektur, Pipeline, Cache und I/O ist grundlegend für das Entwerfen sicherer, effizienter Rechnerstrukturen. - Zukünftige Themen: sichere Architektur, Erweiterungen der Maschinenbefehlsbearbeitung, optimierte Speicherkohärenz, fortgeschrittene Cache-Strategien, speichernahe Programmierung.