Lernzettel

CPU-Komponenten und Arbeitsweise: Register, ALU, Steuereinheit, Cache und Pipelines

Universität: Technische Universität Berlin
Kurs/Modul: Technische Grundlagen der Informatik
Erstellungsdatum: September 19, 2025

Lernzettel: CPU-Komponenten und Arbeitsweise: Register, ALU, Steuereinheit, Cache und Pipelines

(1) Register. Register sind die kleinsten, schnellsten Speicher im Prozessor. Sie halten Operanden, Zwischenergebnisse und Adressinformationen während der Ausführung von Befehlen fest. Typische Typen:

- Allgemeine Register (General-Purpose Registers, GPR) wie R0, R1, ...
- Spezialregister: Programmzähler (PC), Befehlsregister (IR), Adressregister (MAR), Speicherregister (MDR)
- Status-/Flags-Register (z. B. ZF, CF, SF, OF) zur Angabe von Rechenresultaten

Vorteile: extrem niedrige Latenz, hoher Durchsatz, unmittelbarer Zugriff zur ALU. Nachteile: begrenzte Anzahl an Registern, teurer in der Hierarchie als Speicher.

(2) ALU – Arithmetisch-Logische Einheit. Die ALU führt arithmetische und logische Operationen auf Operanden aus. Typische Operationen:

- Arithmetik: Addition, Subtraktion, ggf. Multiplikation/D division
- Logik: UND, ODER, XOR, NICHT
- Verschiebungen/Rotationen: Links-/Rechts-Shift, Rotate

Ausgabe der ALU geht in Register oder in Cache-line-weise Zwischenspeicher. Die Steuerlogik koordiniert, welche Operanden in welchem Register liegen und wohin das Ergebnis geschrieben wird. Die Leistungsfähigkeit der ALU bestimmt maßgeblich den Grunddurchsatz der CPU.

(3) Steuereinheit. Die Steuereinheit orchestriert den Ablauf der Befehle und den Datenfluss im Prozessor. Sie entscheidet, welche Signale an die Register, die ALU, den Speicher und die Cache-Hierarchie gesendet werden.

- Hardwired Control: feste Schaltlogik, schneller, aber weniger flexibel
- Mikroprogrammierte Control: Mikroanweisungen steuern das Verhalten, flexibler

Typischer Ablauf eines Befehlszyklus:

- Fetch: Nächsten Befehl aus dem Speicher in das Befehlsregister
- Decode: Befehl wird decodiert, benötigte Operanden identifiziert
- Execute: ALU-Operation oder Speicherzugriff wird ausgeführt
- Memory/Write-back: Speicherzugriffe erfolgen; Ergebnisse werden in Register geschrieben

In modernen CPUs arbeiten Steuereinheit und Pipeline eng zusammen, um Parallelität zu nutzen und den Taktzyklus auszulasten.

(4) Cache. Cache ist ein schneller Zwischenspeicher zwischen CPU und Hauptspeicher. Er reduziert Zugriffszeiten durch räumliche und zeitliche Lokalität.

- Hierarchie: L1 Daten- und/oder Instruktions-Cache, ggf. L2/L3

- Struktur: Cache ist in Zeilen (Lines) und Blöcke organisiert
- Operationen: Hit bedeutet schneller Zugriff, Miss erfordert Aufruf aus dem nächsten Speicherebenen
- Strategien: Schreibstrategie (Write-Back vs Write-Through), Vorkehrungen gegen Misses
- Parameter: Größe, Blockgröße, assoziativität beeinflussen Leistung und Komplexität

Vorteil: dramatischer Durchsatzanstieg durch Reduktion der Speicherlatenzen; Nachteil: Komplexität, mögliche Cache-Misses, Cache-Kohärenz in Mehrkernsystemen.

(5) Pipelines. Pipelines zerlegen die Befehlsausführung in mehrere aufeinanderfolgende Stufen, um parallel mehrere Befehle zu bearbeiten. Standard-Stufen (Beispiel einer 5-Stufen-Pipeline):

- Fetch (F): Befehl aus dem Speicher
- Decode (D): Befehl decodieren, Operanden bestimmen
- Execute (E): ALU-Operation durchführen
- Memory (M): Speicherzugriff, falls nötig
- Write-back (W): Ergebniss in Register schreiben

Vorteil: höhere Throughput, potenziell mehrere Befehle pro Taktzyklus. Risiken:

- Data Hazards: Abhängigkeiten zwischen aufeinanderfolgenden Befehlen
- Control Hazards: Sprung- bzw. Verzweigungsbefehle
- Structural Hazards: begrenzte Ressourcen

Lösungen:

- Datenweiterleitung (Forwarding)
- Blöcke/Staus an kritischen Stellen (Stalling)
- Spekulative Ausführung und Branch Prediction

Fortgeschrittene Designs können mehrere Befehle pro Taktzyklus bearbeiten (Superskalare Implementierungen) oder unterschiedliche Pipelines parallel nutzen.