Rechnerorganisation
Sommersemester 2023 – 9. Vorlesung

Prof. Stefan Roth, Ph.D.

Technische Universität Darmstadt

26. Juni 2023

Inhalt

Mikroarchitektur [HH16, S. 385 – 484]

- Einführung in die Mikroarchitektur
- Eintakt-Prozessor
- Mehrtakt-Prozessor
- Pipeline-Prozessor
- Analyse der Rechenleistung
- Ausnahmebehandlung
- Weiterführende Themen

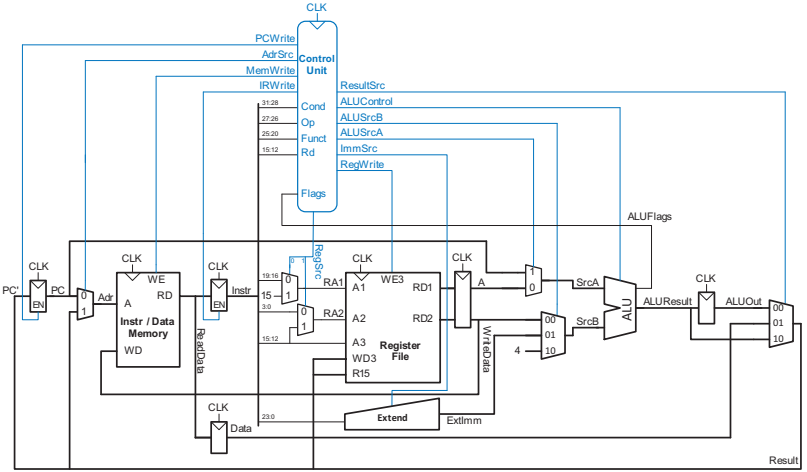
Wiederholung: Befehlsausführung & Mehrtakt-Prozessor

Interpretation und Ausführung eines Befehls

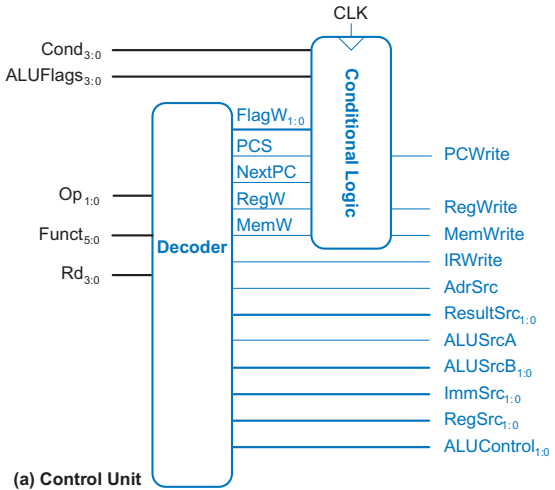
- Wie funktioniert die Verarbeitung eines Befehls bzw. der Daten in einem Rechnersystem?
  - Sowohl der Befehl, als auch die Daten stehen in dem Speicher(-system).
- Die Aufgabe des Prozessors bzw. des Steuerwerks ist es, die Befehle aus dem Speicher zu lesen (bzw. die Anweisungen dafür zu geben). Diesen Vorgang bezeichnet man als **Befehlsholphase**.[1]
- Wenn der Befehl geholt ist und in einem Register steht, muss er dekodiert werden. Diesen Vorgang nennt man **Befehlsdekodierung**.[2]
- Als letztes wird der Befehl ausgeführt (**Befehlsausführung**).[3] Danach wird der nächste Befehl aus dem Speicher geholt.
- Man spricht auch von den *drei Phasen der Befehlsausführung*.

[1] engl.: instruction fetch
[2] engl.: instruction decode
[3] engl.: instruction execute

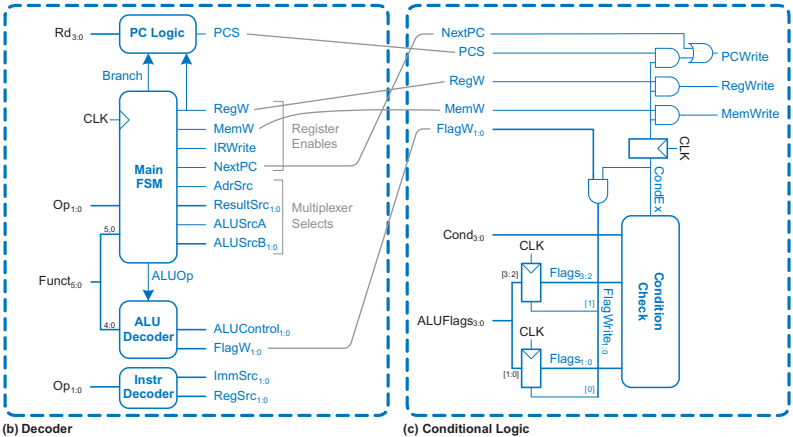
Mehrtakt-Prozessor: Datenpfad und Kontrolleinheit



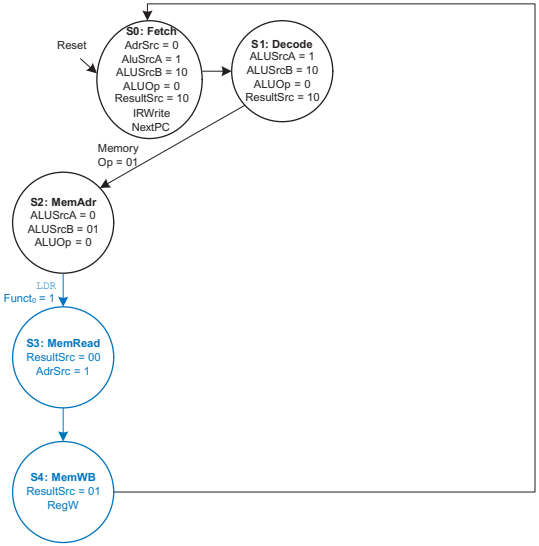
Mehrtakt-Prozessor: Kontrolleinheit



Mehrtakt-Prozessor: Kontrolleinheit - Detail



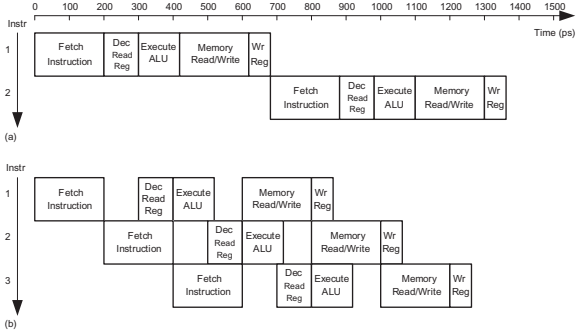
Mehrtakt-Prozessor: Steuerwerk, Zustände für `ldr`



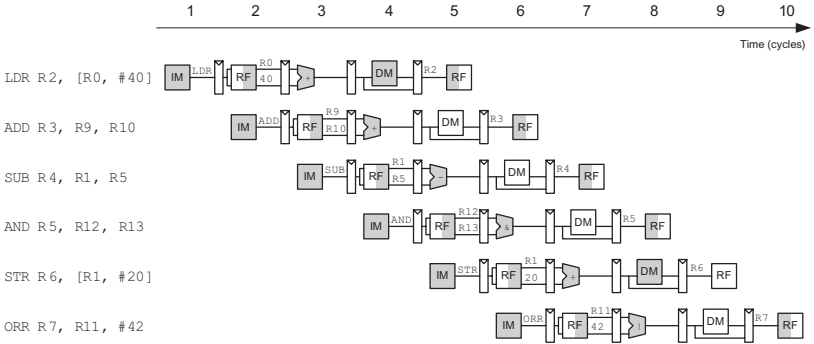
Pipeline-Prozessor

Pipeline-Prozessor, Befehlsausführung

- Die Phasen der Befehlsausführung werden beim ARM® Prozessor in fünf Phasen eingeteilt
  - Instruction Fetch
  - Instruction Decode, Read Register
  - Execute ALU
  - Memory Read/Write
  - Write Register

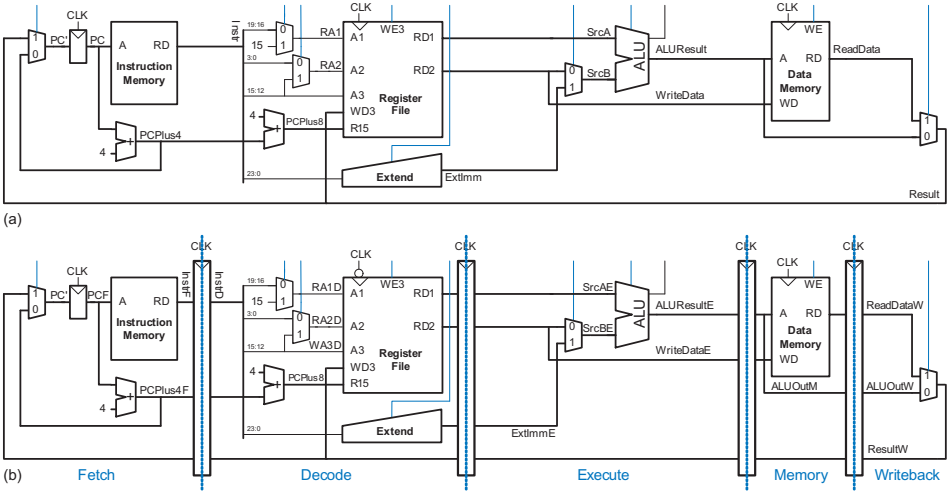


Pipeline-Prozessor: abstrakte Darstellung

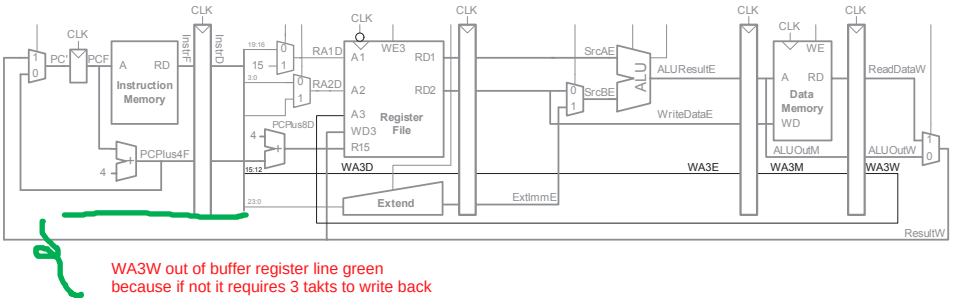


- Vereinfachte Darstellung des Datenpfads der Mikroarchitektur
- Abkürzungen:
  - IM – Instruction Memory (Befehlsspeicher)
  - RF – Register File (Registerfeld)
  - DM – Data Memory (Datenspeicher)

Pipeline-Prozessor: Datenpfad – erster Versuch

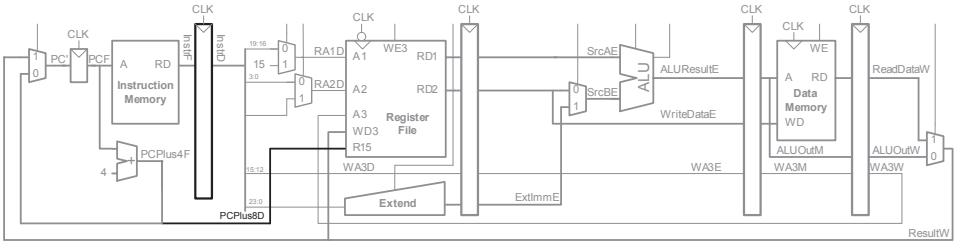


Pipeline-Prozessor: Datenpfad – korrigiert



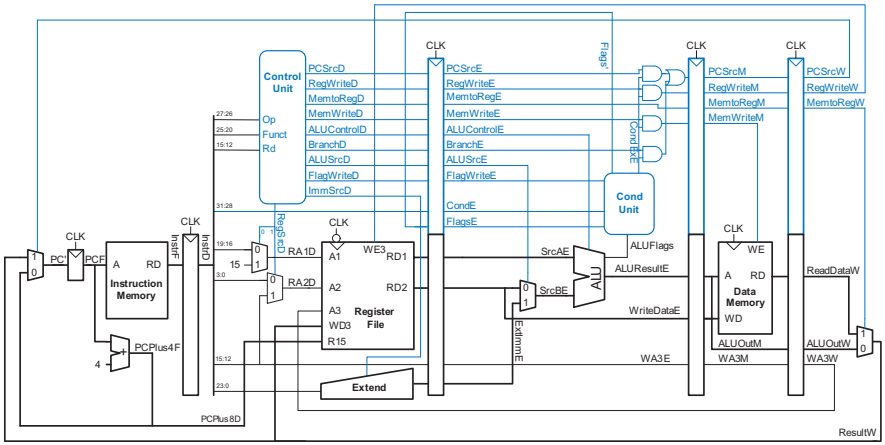
- Die Adresse des Zielregister muss zusammen mit dem Datum des Zielregisters angelegt werden. Deshalb wird auch die Adresse des Zielregisters (WA3D) über die Pipelinestufen geführt.

Pipeline-Prozessor: Datenpfad – Optimierungen



- Optimierung der Program Counter Logik durch Eliminierung eines Registers und eines Addierers

Pipeline-Prozessor: Datenpfad und Kontrolleinheit

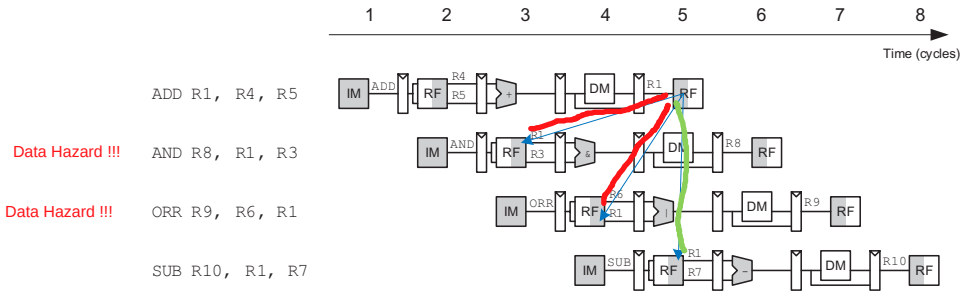


Hazards

Abhängigkeiten zwischen Pipeline-Stufen – Hazards

- Treten auf wenn eine
  - Instruktion vom Ergebnis einer vorhergehenden abhängt
  - ... diese aber noch kein Ergebnis geliefert hat
- Arten von Hazards
  - Data Hazard: z. B. Neuer Wert von Register noch nicht in Registerfeld eingetragen
  - Control Hazard: Unklar welche Instruktion als nächstes ausgeführt werden muss (tritt bei Verzweigungen auf)

Data Hazard



nur bei SUB R10,
R1, R7 wird kein Data Hazard auftreten

- Hier: Read-after-Write Hazard (RAW) – `r1` „muss vor Lesen geschrieben werden“

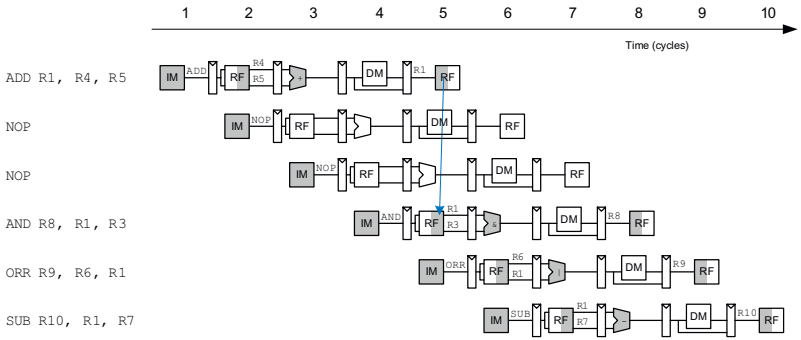
Lösung:
Software: andere Befehle inzwischen schreiben, sodass keine Abhängigkeit auftritt
oder nops (Wartezeit-Befehl)

Umgang mit Data Hazards

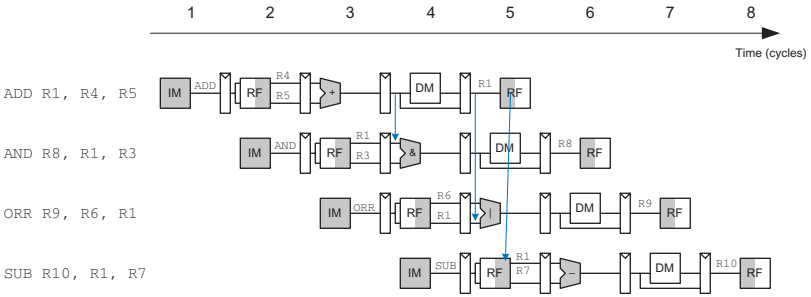
- Möglichkeiten
  - Plane Wartezeiten von Anfang an ein
    - Füge nops[4] zur Compile-Zeit ein
    - scheduling (Ablaufplanung)
  - Stelle Maschinencode zur Compile-Zeit um
    - scheduling / reordering
  - Leite Daten zur Laufzeit schneller über Abkürzungen weiter
    - bypassing / forwarding
  - Halte Prozessor zur Laufzeit an bis Daten da sind
    - stalling

[4] no operations

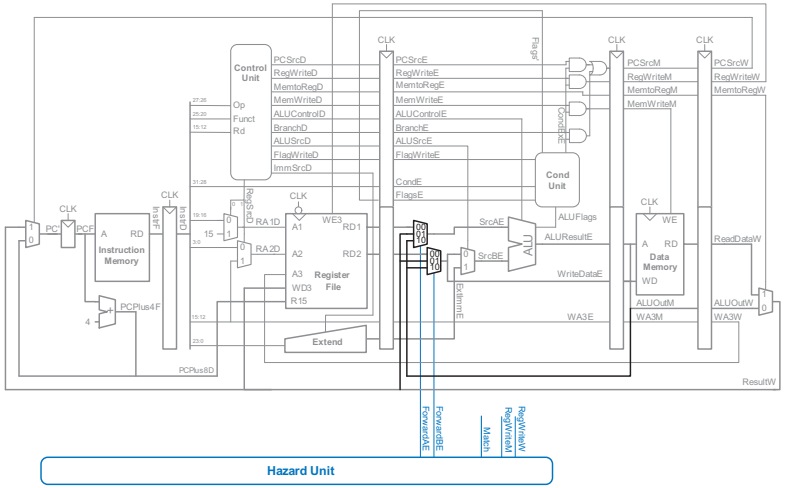
Data Hazard – nops einfügen



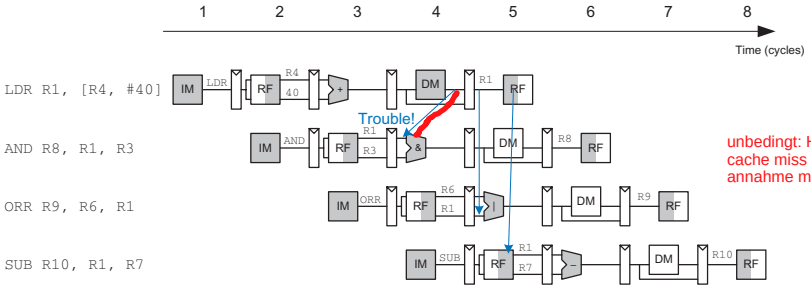
Data Hazard – bypassing / forwarding



Pipeline-Prozessor: Datenpfad, Kontrolleinheit, Hazard-Unit

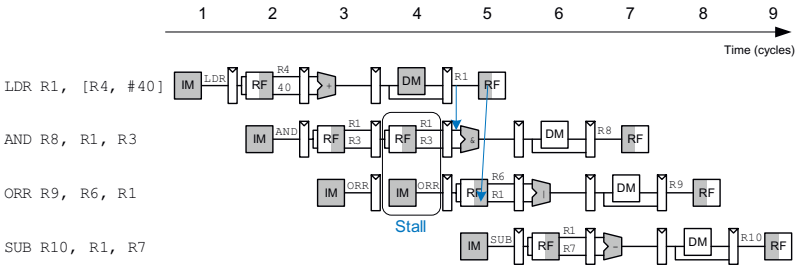


Data Hazard – ldr

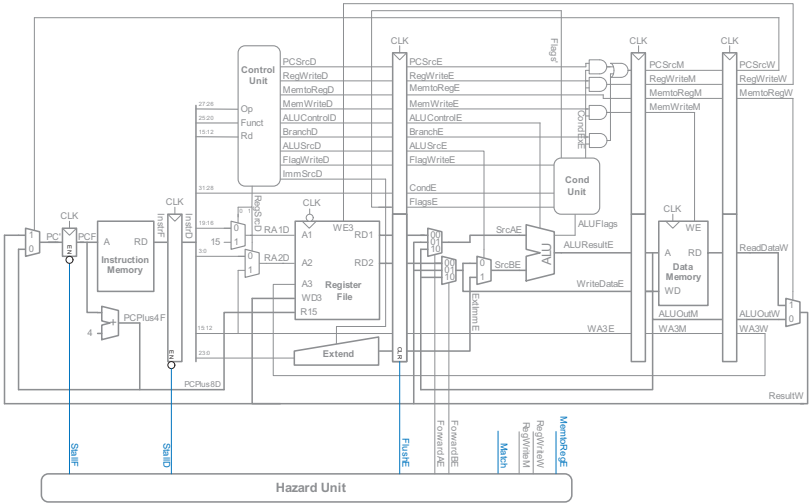


unbedingt: Hardware Lösung denn die cache miss in DM ist not messbar (muss annahme machen)

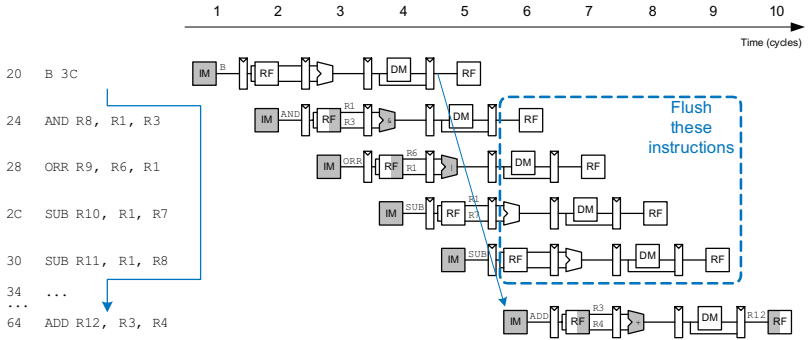
Data Hazard – stall



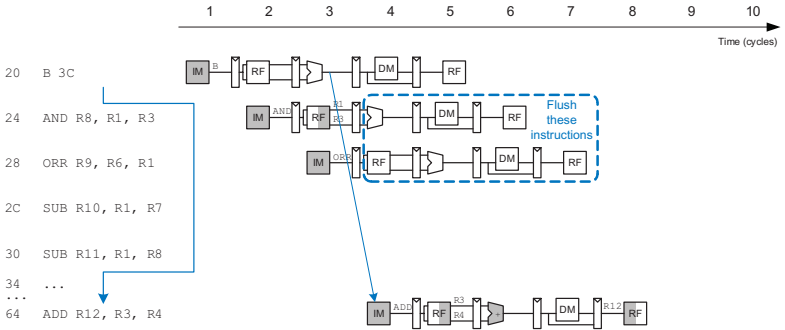
Pipeline-Prozessor: Datenpfad, Kontrolleinheit, Hazard- & Stall-Unit



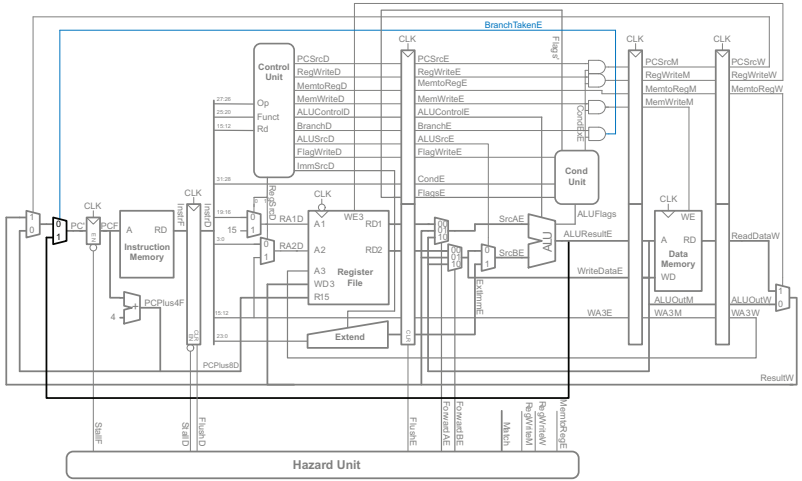
Control Hazards



Control Hazards



Pipeline-Prozessor: Datenpfad, Kontrolleinheit, Hazard, Stall & Controll-Unit



Zusammenfassung und Ausblick

Zusammenfassung und Ausblick

Zusammenfassung

- Mikroarchitekturen von Rechnersystemen
- Pipeline-Prozessor

Ausblick

- Analyse der Rechenleistung
- Ausnahmebehandlung
- Weiterführende Themen

Lernkontrolle

- Die Ideen zur Entwicklung eines Pipeline-Prozessor kann ich nachvollziehen ☑
- Die Entstehung der unterschiedlichen Hazards habe ich verstanden und kann diese bei der Programmierung in Assembler vermeiden ☑
- ...

Literatur

Literatur

[BO10] Bryant, Randal E. und David R. O´Hallaron: *Computer Systems - A Programmer´s Perspective*.
Prentice Hall, 2010.

[HH16] Harris, David Money und Sarah L. Harris: *Digital Design and Computer Architecture, ARM® Edition*.
Morgan Kaufmann, 2016.