

Technische Grundlagen der Informatik – Kapitel 7



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Prof. Dr. Jan Peters
Fachgebiet Intelligente Autonome Systeme (IAS)
Fachbereich Informatik

WS 14/15



Meisenantworten ... vielen Dank für die Hinweise!

- *Frohes neues Jahr! Ich hoffe, Ihr hattet gute Weihnachten?*
- *Vielen Danke für die Weihnachtsgruesse!*
- *Deutlicher Artikulieren – versuche ich!*
- *Danke für Klassewitze (besonders, Halloween Oct 31 = Dec 25 Weihnachten)*
- *Kapitel 5 ins Wiki – hochgeladen! Sorry, aber unsere Tutoren haben Schwierigkeiten nachzukommen.*
- *Unfaire Snickerverteilung – sorry!*
- *Wieder Lehrewiki-Fragen! Sind zurück! Schön, dass sie vermisst wurden!*
- *Meisenantworten vermisst? Sind zurück!*
- *Antworten von Kommilitonen wiederholen. Bemühe mich!*
- *Projekt “Sniggers-shooter” betreue ich gerne als Bachelor-Arbeit!*
- **Weitere drei Meisenantworten folgen morgen...**

Kapitel 7: Themen

- **Einführung in die Mikroarchitektur**
- **Analyse der Rechenleistung**
- **Ein-Takt-Prozessor**
- **Mehrtakt-Prozessor**
- **Pipeline-Prozessor**
- **Ausnahmebehandlung**
- **Weiterführende Themen**

Einleitung

- Mikroarchitektur
 - Hardware-Implementierung einer Architektur
- Prozessor:
 - Datenpfad: funktionale Blöcke
 - Steuerwerk: Steuersignale

Anwendungs-Software	Programme
Betriebs-systeme	Gerätetreiber
Architektur	Instruktionen Register
Mikro-architektur	Datenpfade Steuerwerke
Logik	Addierer Speicher
Digital-schaltungen	AND Gatter NOT Gatter
Analog-schaltungen	Verstärker Filter
Bauelemente	Transistoren Dioden
Physik	Elektronen

LEHRE WIKI
TEST IN
DREI
FOLIEN

- Mehrere Implementierungen für eine Architektur
 - Ein-Takt
 - Jede Instruktion wird in einem Takt ausgeführt
 - Mehrtakt
 - Jede Instruktion wird in Teilschritte zerlegt
 - Pipelined
 - Jede Instruktion wird in Teilschritte zerlegt
 - Mehrere Instruktionen werden gleichzeitig ausgeführt

Rechenleistung eines Prozessors

- Ausführungszeit eines Programms

$$\text{Ausführungszeit} = (\# \text{ Instruktionen})(\text{Takte/Instruktion})(\text{Sekunden/Takt})$$

- Definitionen:
 - Takte/Instruktion = CPI (*cycles per instruction*)
 - Sekunden/Takt = Taktperiode
 - $1/\text{CPI} = \text{Instruktionen/Takt} = \text{IPC}$ (*instructions per cycle*)
- Herausforderung: Einhalten zusätzlicher Anforderungen
 - Kosten
 - Energiebedarf
 - Rechenleistung

- Bitte jetzt auf LEHRE WIKI eine Frage beantworten!

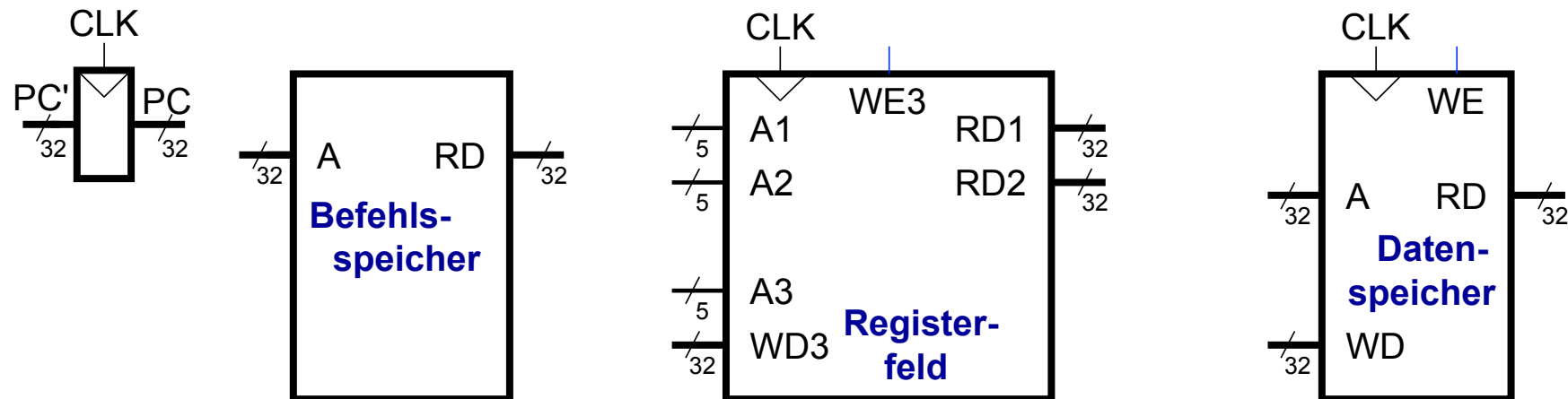
Unser erster MIPS Prozessor

- Zunächst **Untermenge** des MIPS Befehlssatzes:
 - R-Typ Befehle: `and`, `or`, `add`, `sub`, `slt`
 - Speicherbefehle: `lw`, `sw`
 - Bedingte Verzweigungen: `beq`
- **Später** hinzunehmen: `addi` und `j`

Architekturzustand

- Auf Ebene der **Architektur** sichtbare Daten
 - Für den Programmierer **zugänglich**
- Bestimmen **vollständigen** Zustand der Architektur
 - PC
 - 32 Register
 - Speicher

Elemente des MIPS Architekturzustands



Ein-Takt MIPS Prozessor

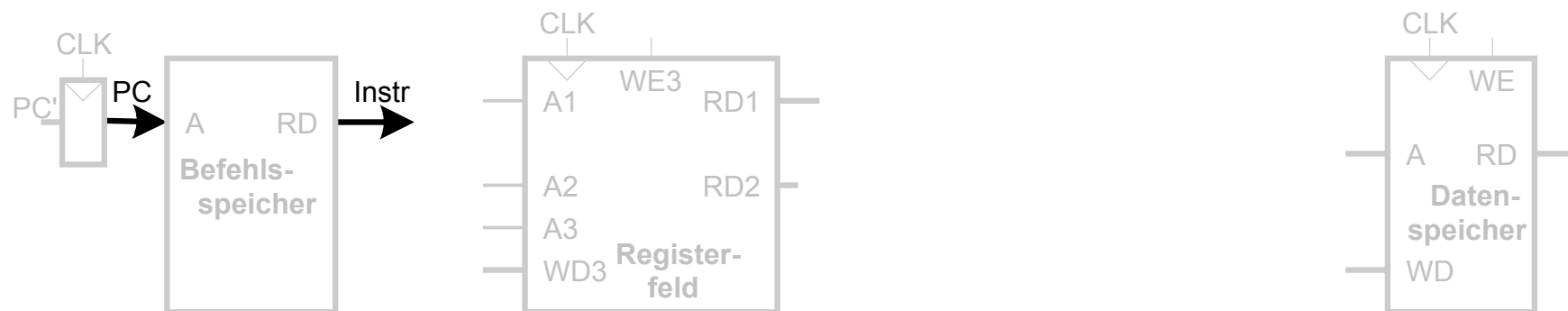


- Datenpfad
- Steuerwerk



Ein-Takt Datenpfad: Holen eines 1w Befehls

- Ein *load word* Befehl (1w) soll ausgeführt werden
- **Schritt 1:** Hole Instruktion

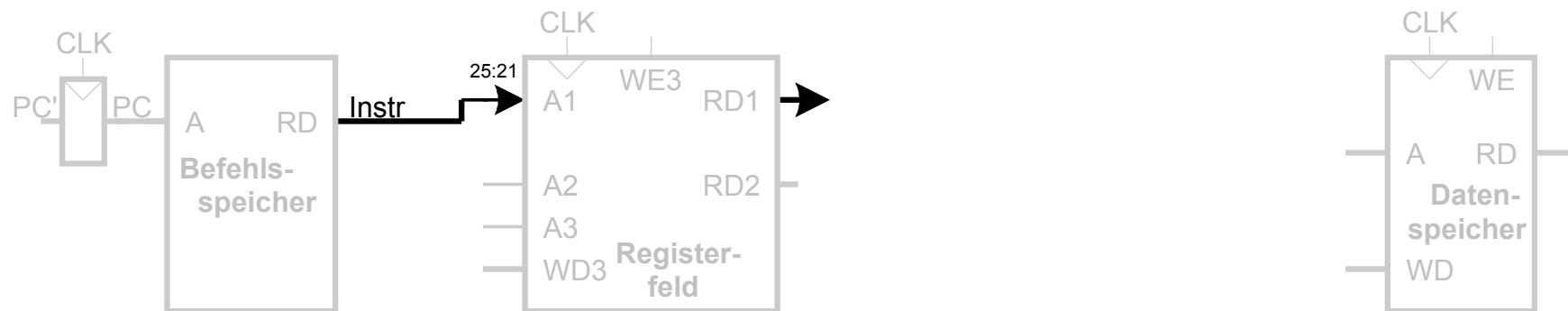


Ein-Takt Datenpfad: Lesen des Registers für 1w



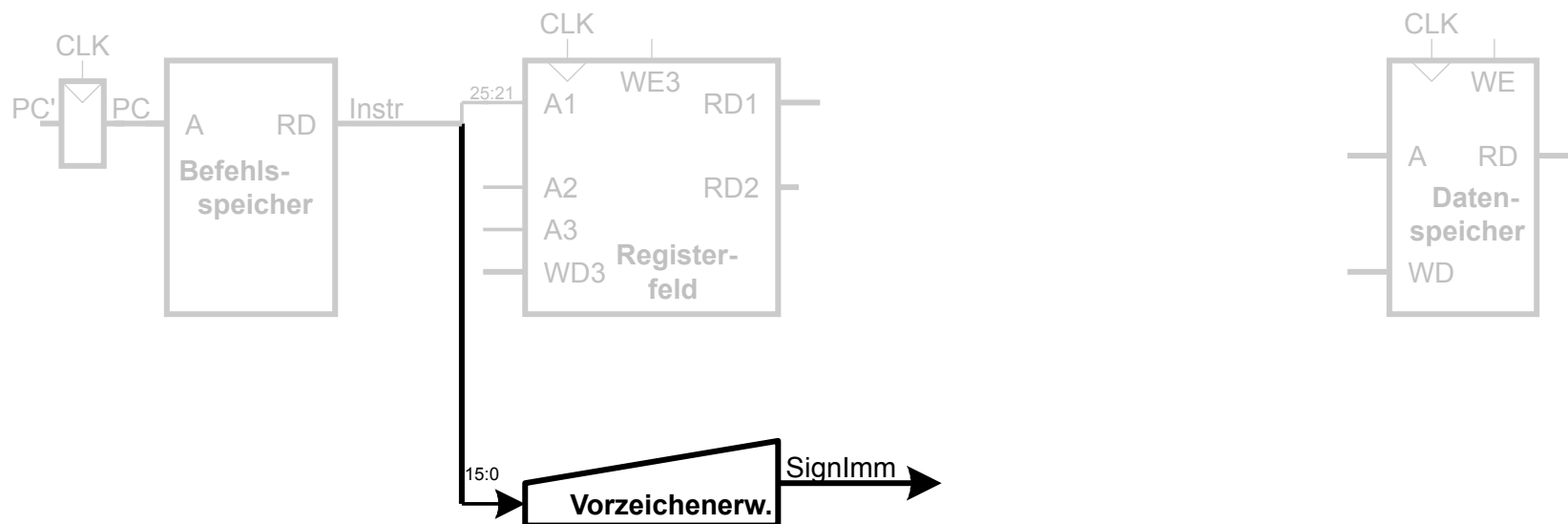
TECHNISCHE
UNIVERSITÄT
DARMSTADT

- **Schritt 2:** Lese Quelloperand aus Registerfeld



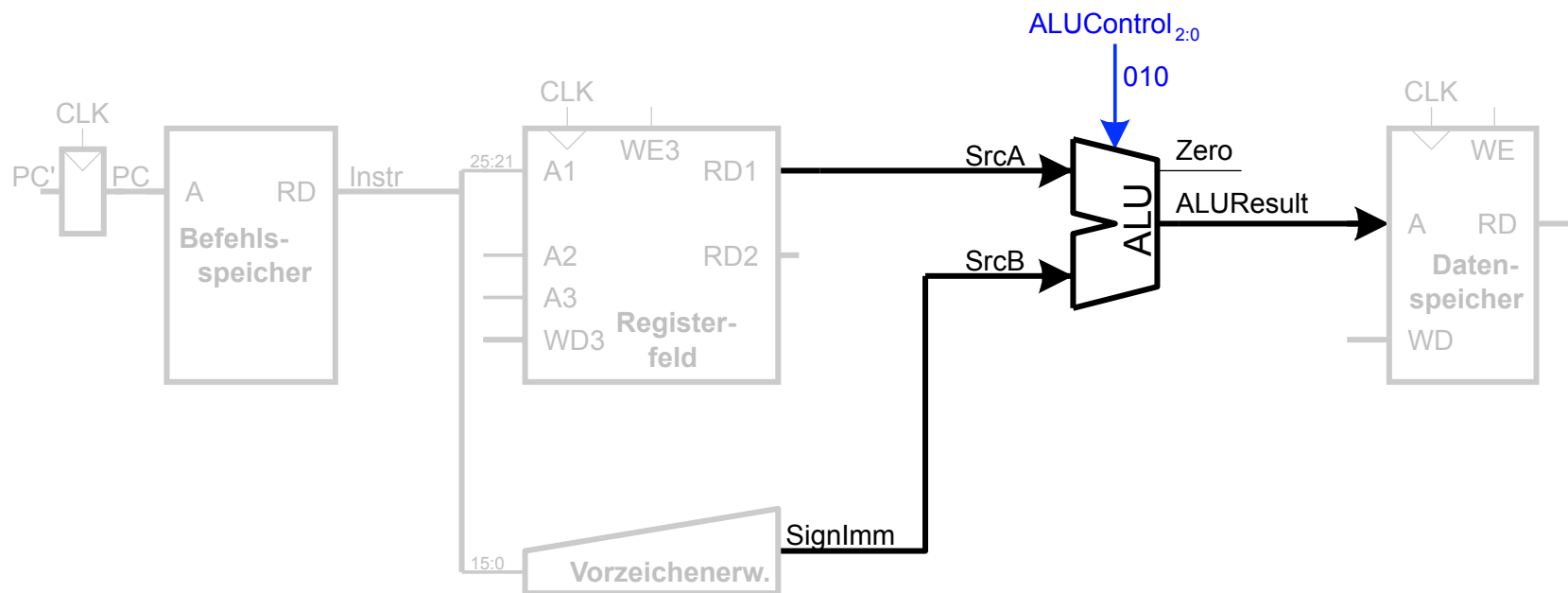
Ein-Takt Datenpfad: Behandle 1w Direktwert

- **Schritt 3:** Vorzeichenerweitere den 16b Direktwert auf 32b Signal `SignImm`



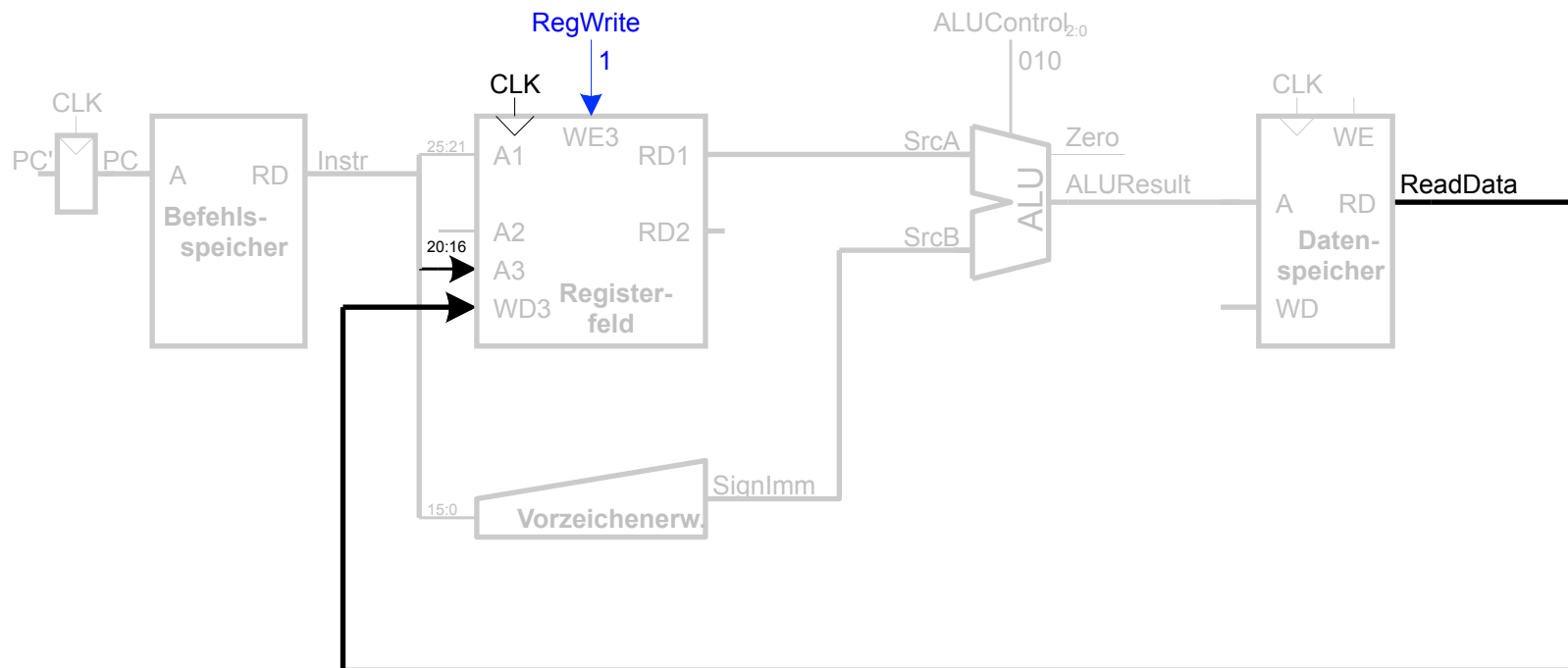
Ein-Takt Datenpfad: Berechne 1w Zieladresse

- **Schritt 4:** Berechne die effektive Speicheradresse



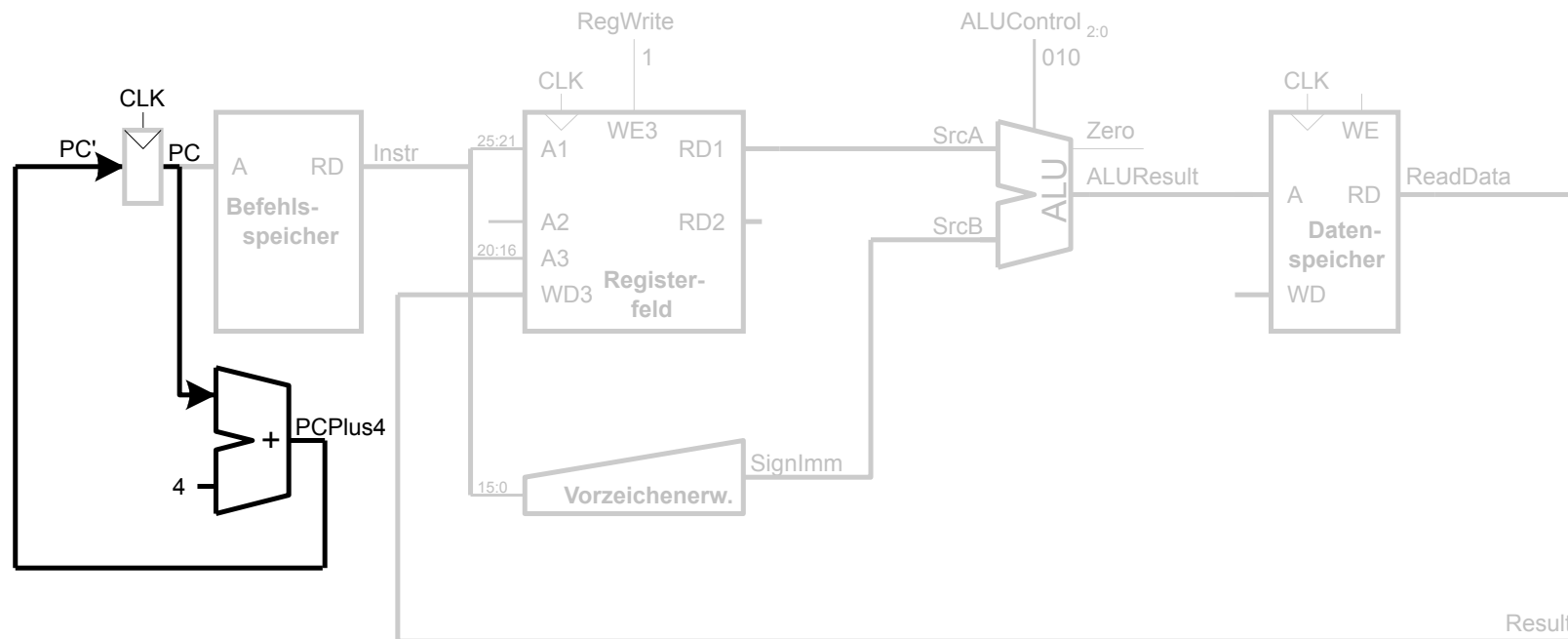
Ein-Takt Datenpfad: Lese Speicher mit 1w

- **Schritt 5:** Lese Daten aus Speicher und schreibe sie ins passende Register



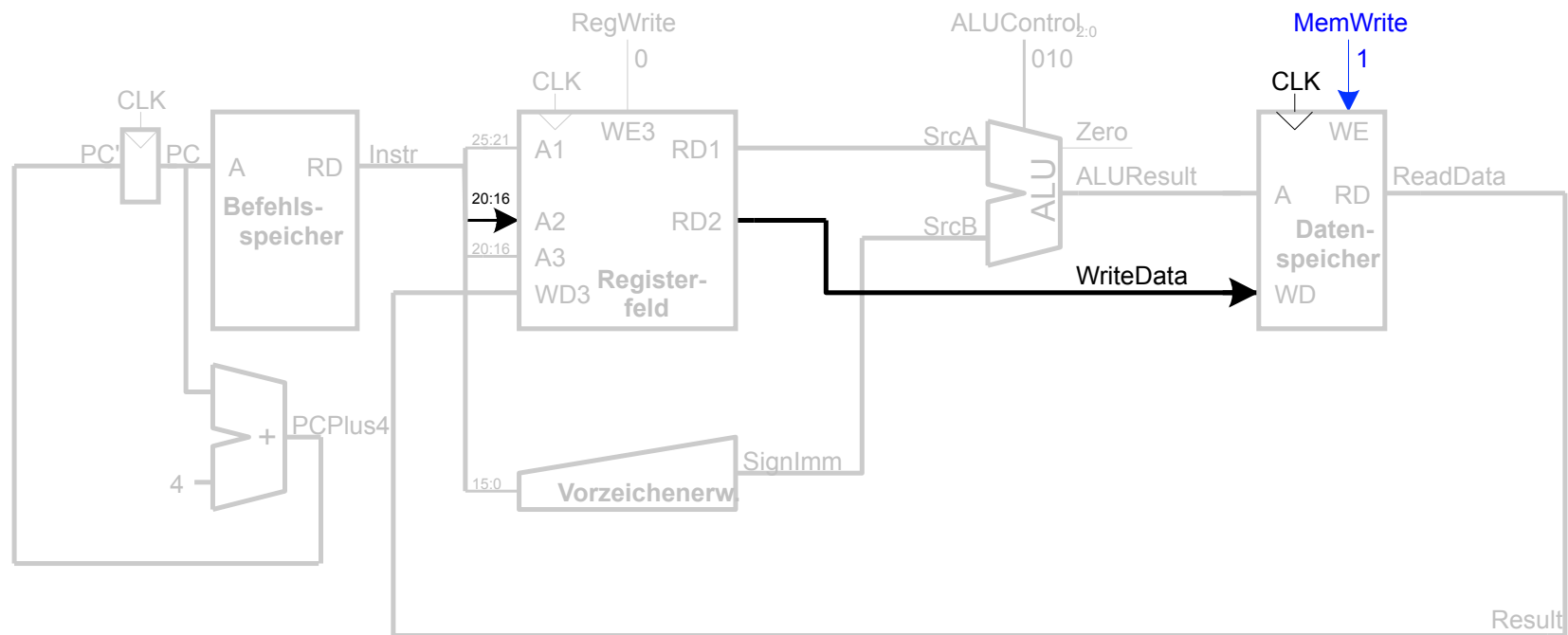
Ein-Takt Datenpfad : Erhöhe PC nach 1w

- **Schritt 6:** Bestimme Adresse des nächsten Befehls



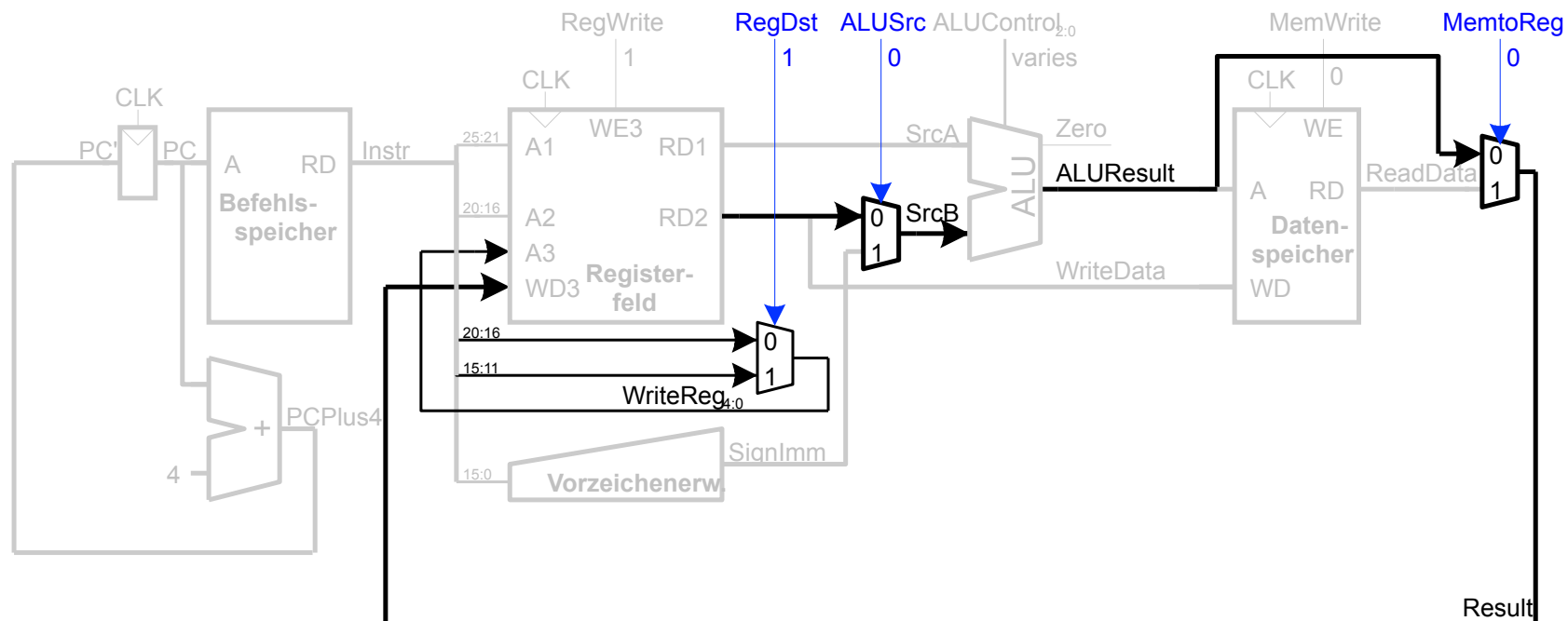
Ein-Takt Datenpfad: *sw*

- Schreibe Daten aus *rt* in den Speicher



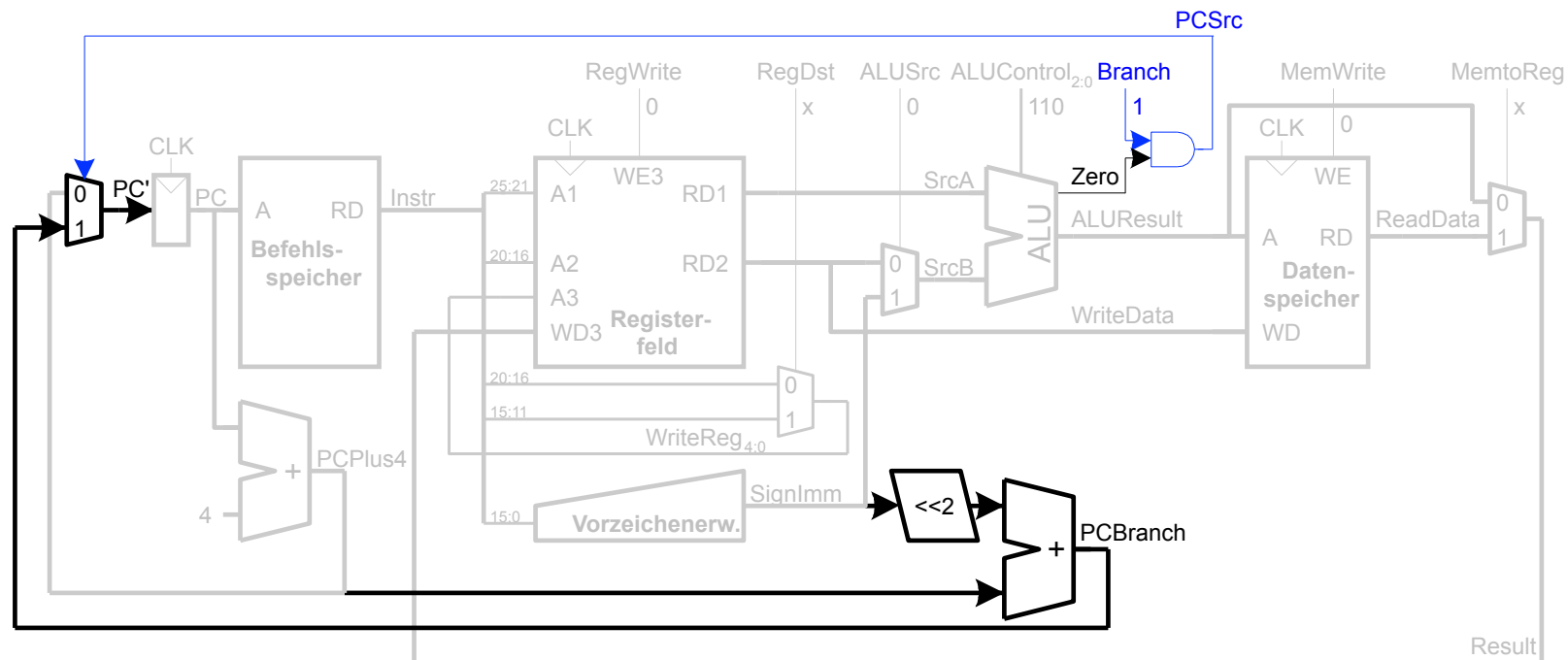
Ein-Takt Datenpfad: Instruktionen vom R-Typ

- Lese aus *rs* und *rt*
- Schreibe *ALUResult* ins Registerfeld
- Schreibe nach *rd* (statt nach *rt* wie bei *sw*)



Ein-Takt Datenpfad: beq

- Prüfe ob Werte in `rs` und `rt` gleich sind
- Bestimme Adresse von Sprungziel (*branch target adress, BTA*):
$$BTA = (\text{vorzeichenerweiterter Direktwert} \ll 2) + (PC+4)$$



Vollständiger Ein-Takt-Prozessor

