

Rechnerorganisation

Prof. Dr. Wolfgang Karl

Vorlesung im Wintersemester 2025/2026 – Foliensatz: RO25-FS08

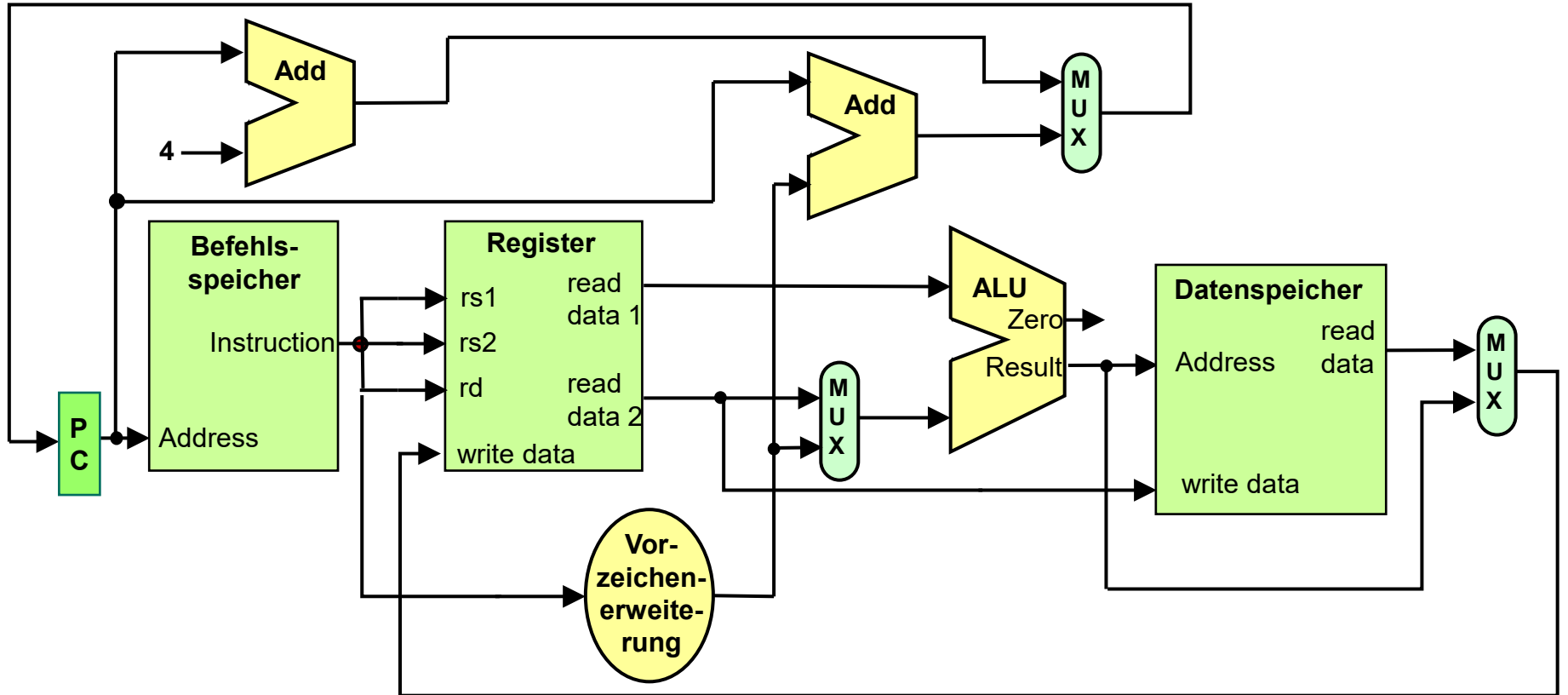


Kapitel 6

Prozessororganisation (Fortsetzung)

- Grundlagen Pipelining des Maschinenbefehlszyklus
- Pipeline-Hemmnisse, Pipeline-Konflikte
- **Pipeline-Organisation RISC-V**

Datenpfad für RISC-V: Einzyklen-Implementierung



Pipelining

■ Befehlsabarbeitung

■ Fallstudie RISC-V: Phasen des Maschinenbefehlszyklus

1. Hole Befehl aus dem Speicher (Befehl holen, instruction fetch, **IF**)
2. Dekodiere Befehl und lese Register (Befehl dekodieren, instruction decode, **ID**)
3. Führe Operation aus oder berechne Adresse (Befehl ausführen, instruction execute, **EX**)
4. Greife auf Datum im Speicher zu, falls notwendig (Speicherzugriff, memory access, **MA**)
5. Schreibe Ergebnis in ein Register, falls notwendig (Ergebnis in Register schreiben, write back, **WB**)

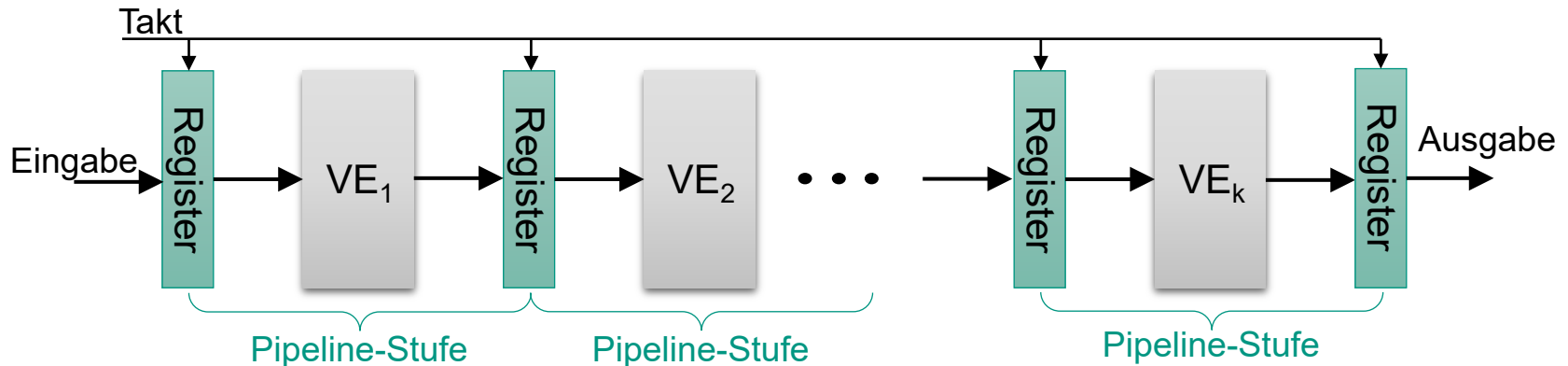
Pipelining

■ Befehlsabarbeitung

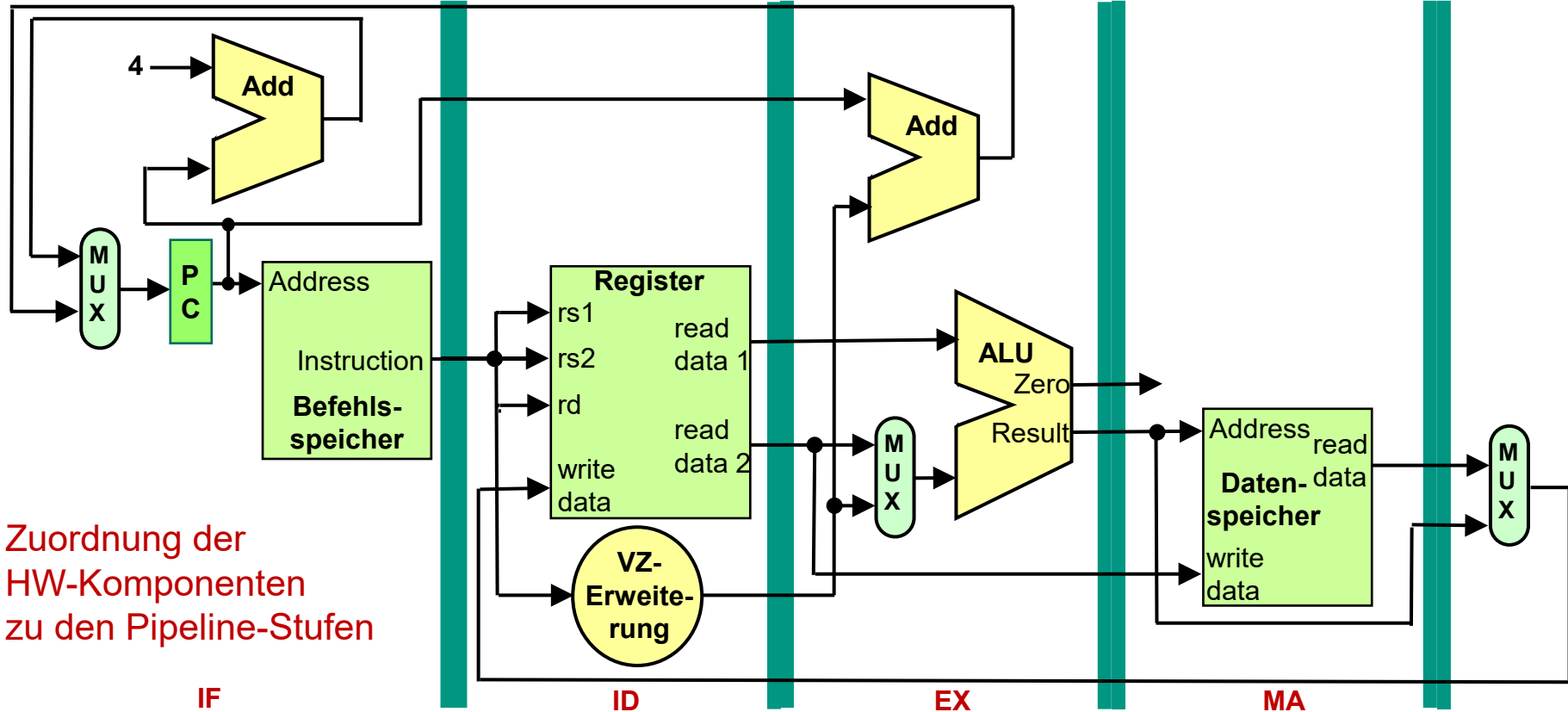
■ Pipelining des Maschinenbefehlszyklus (Instruction Pipelining)

- Gesamtheit der Verarbeitungseinheiten VE_1, VE_2, \dots, VE_k
- Pipeline-Stufen sind durch Register getrennt: taktsynchrone Verarbeitung;

■ k-stufige Pipeline:



Datenpfad für RISC-V: Pipeline-Implementierung



Zuordnung der
HW-Komponenten
zu den Pipeline-Stufen

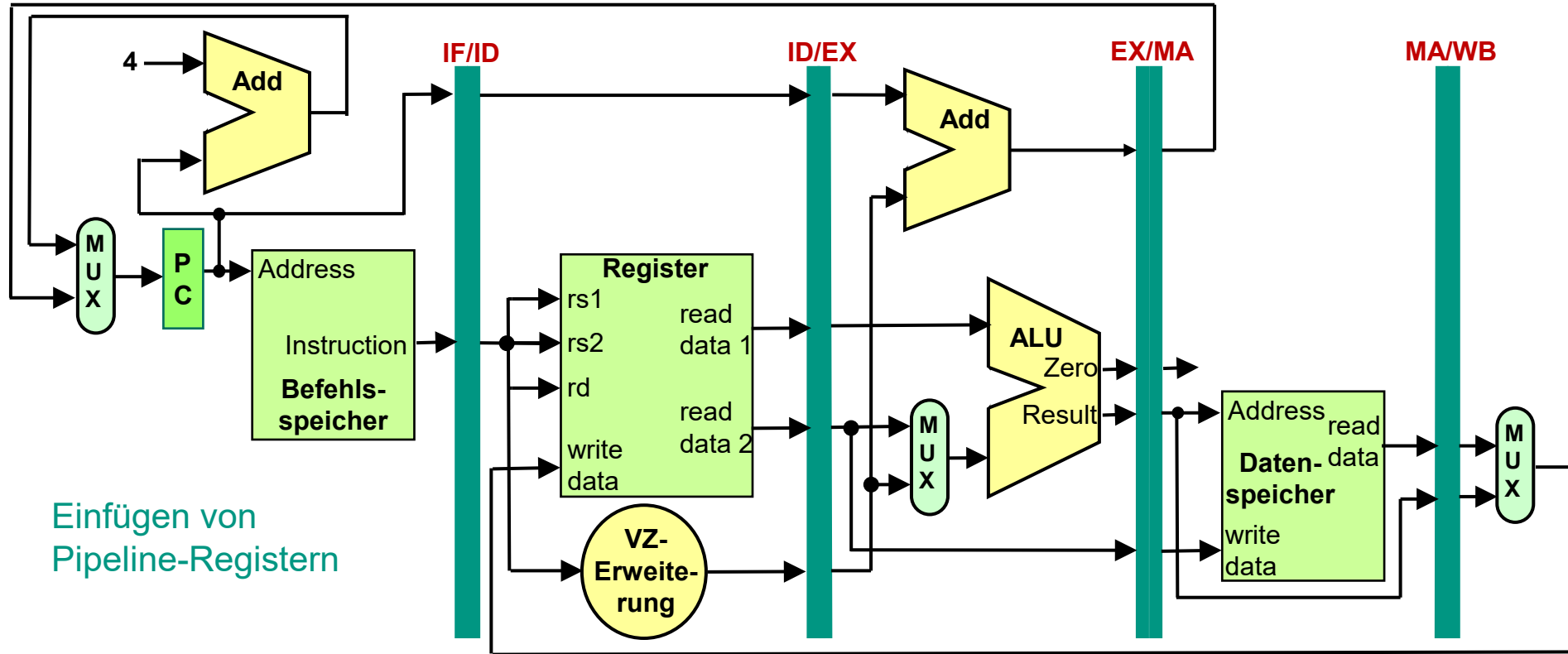
IF

ID

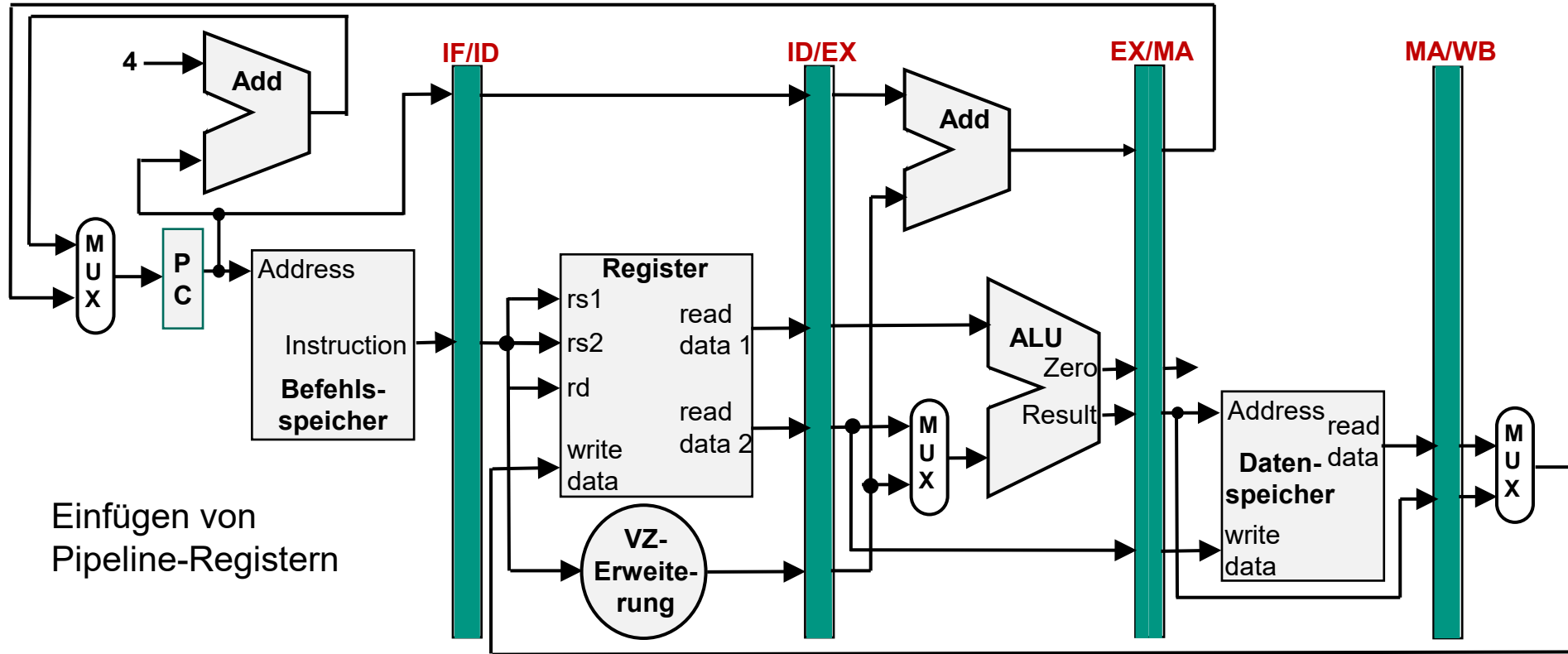
EX

MA

Datenpfad für RISC-V: Pipeline-Implementierung

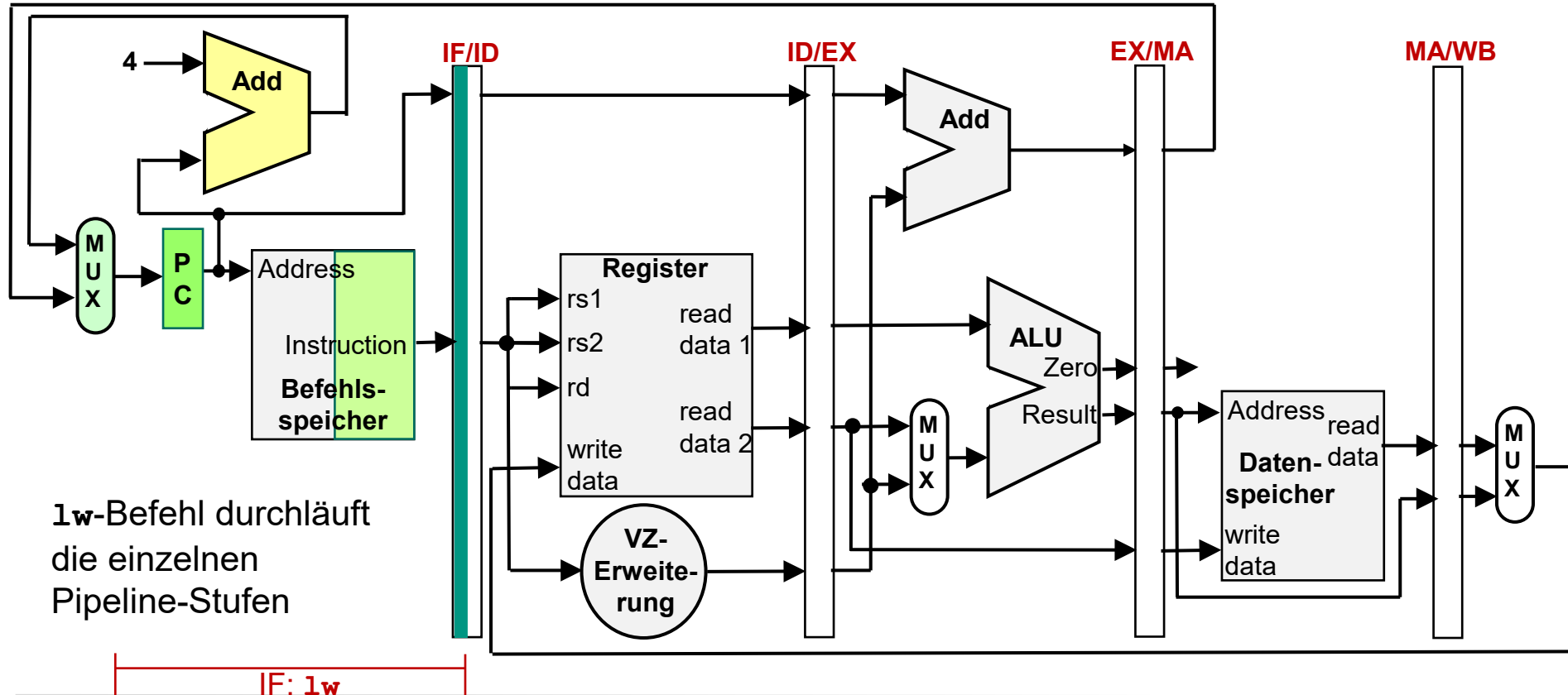


Datenpfad für RISC-V: Pipeline-Implementierung

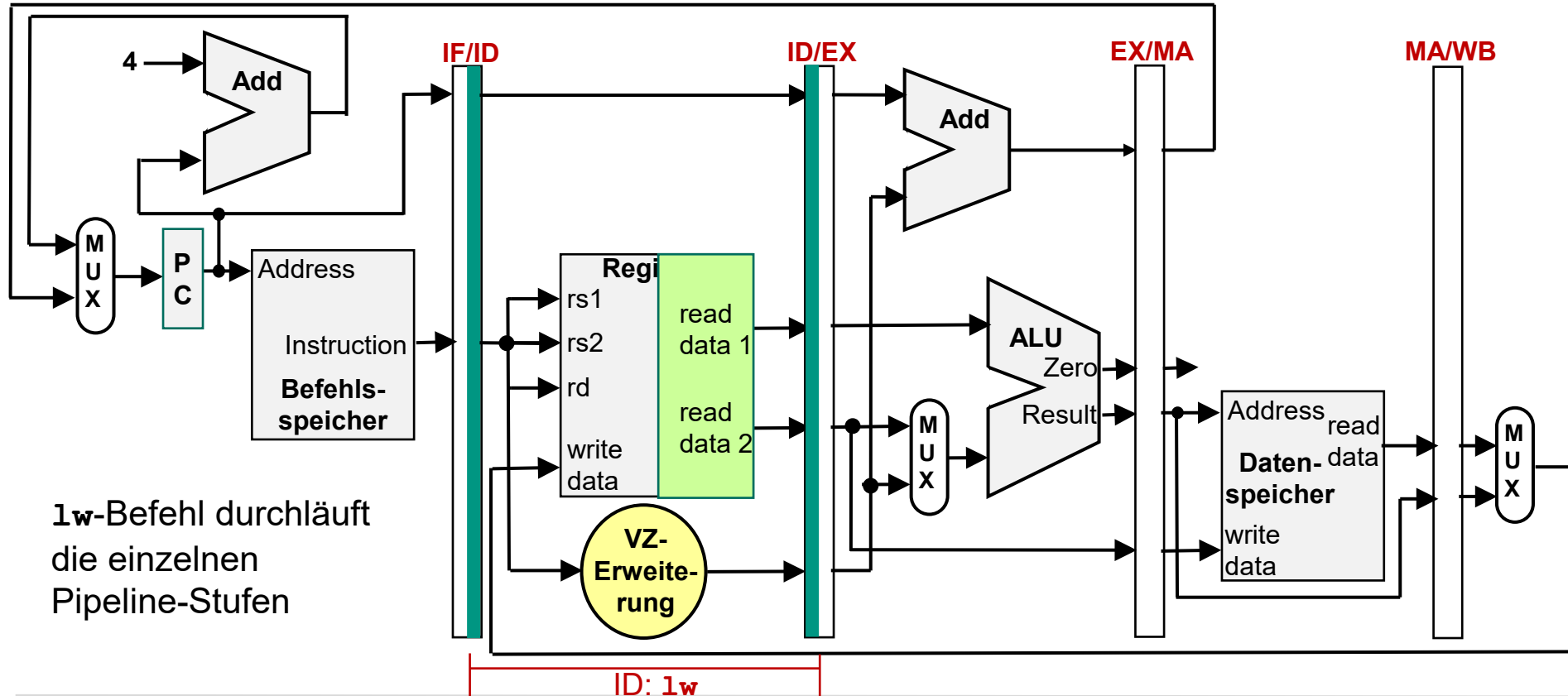


Einfügen von
Pipeline-Registern

Datenpfad für RISC-V: Pipeline-Implementierung

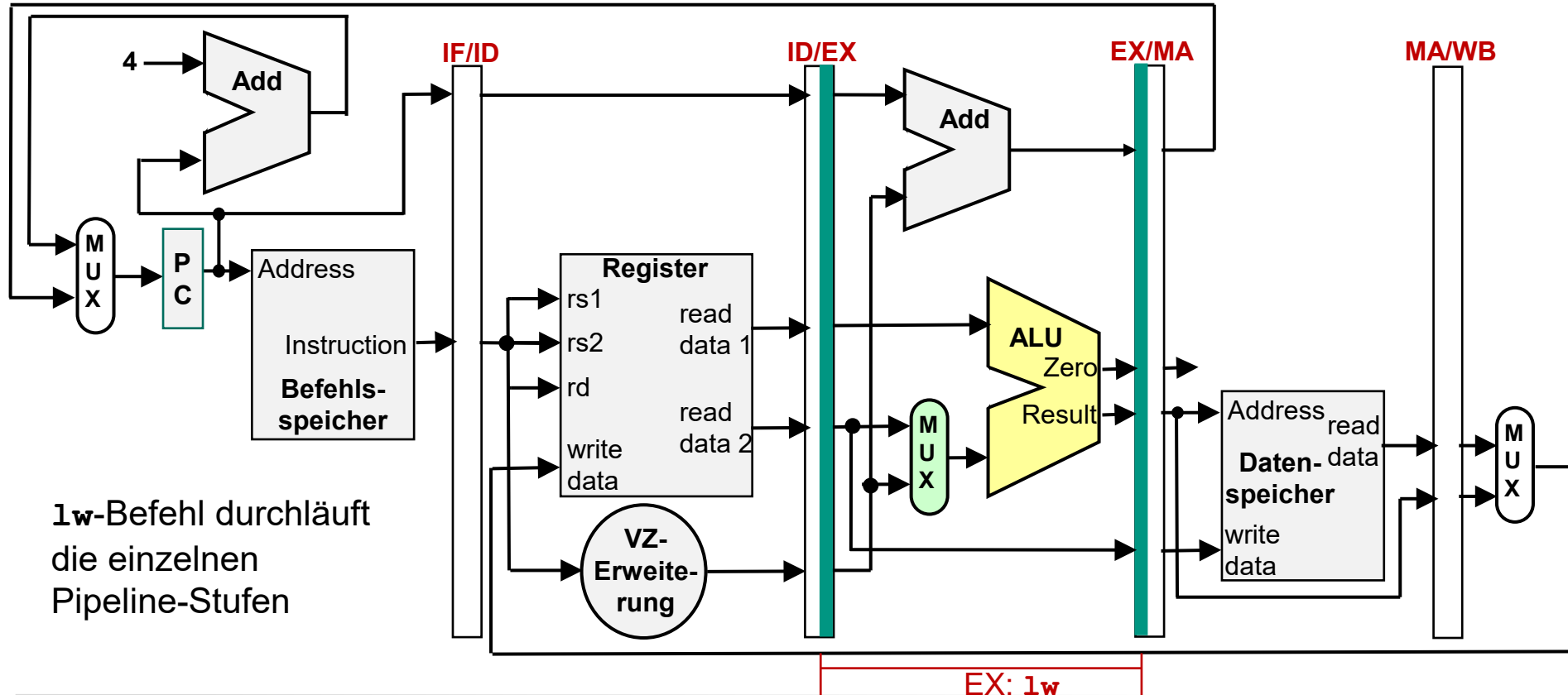


Datenpfad für RISC-V: Pipeline-Implementierung



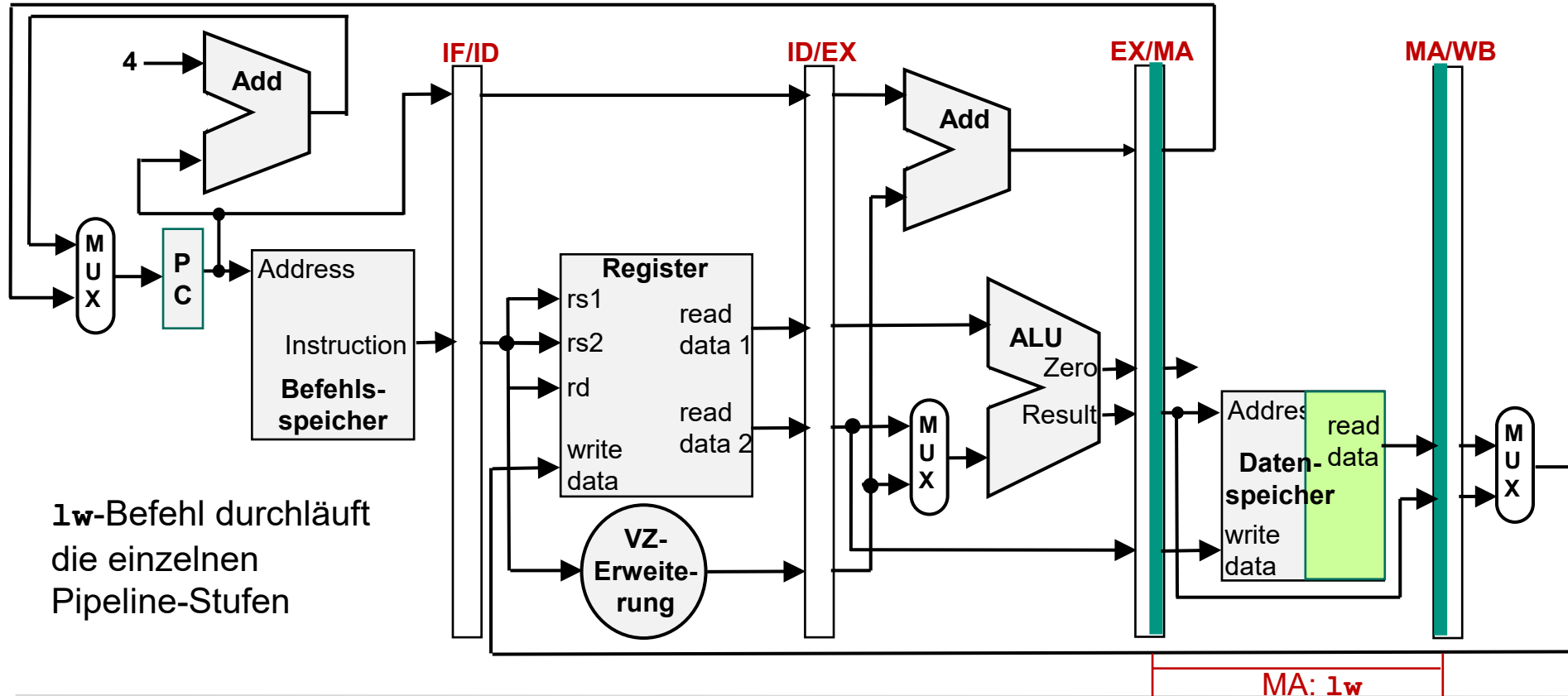
1w-Befehl durchläuft die einzelnen Pipeline-Stufen

Datenpfad für RISC-V: Pipeline-Implementierung



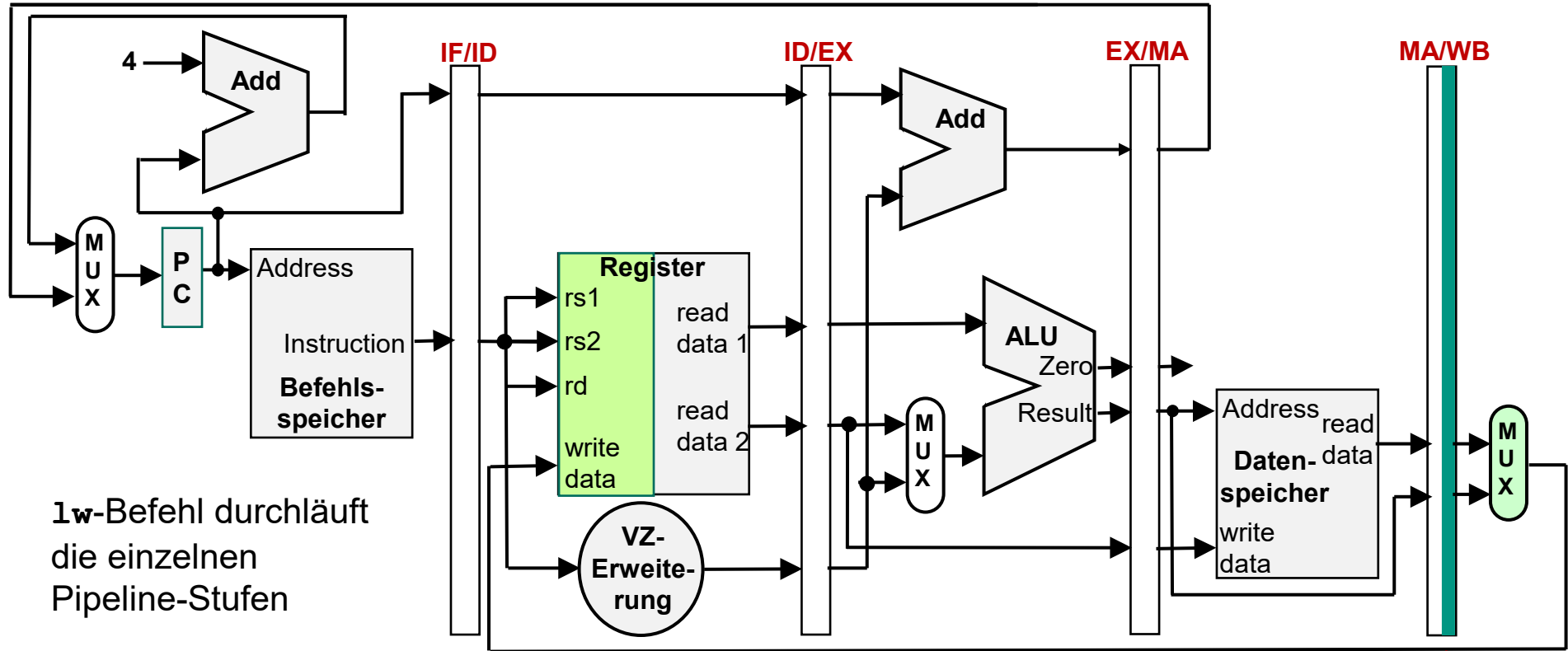
1w-Befehl durchläuft die einzelnen Pipeline-Stufen

Datenpfad für RISC-V: Pipeline-Implementierung



1w-Befehl durchläuft die einzelnen Pipeline-Stufen

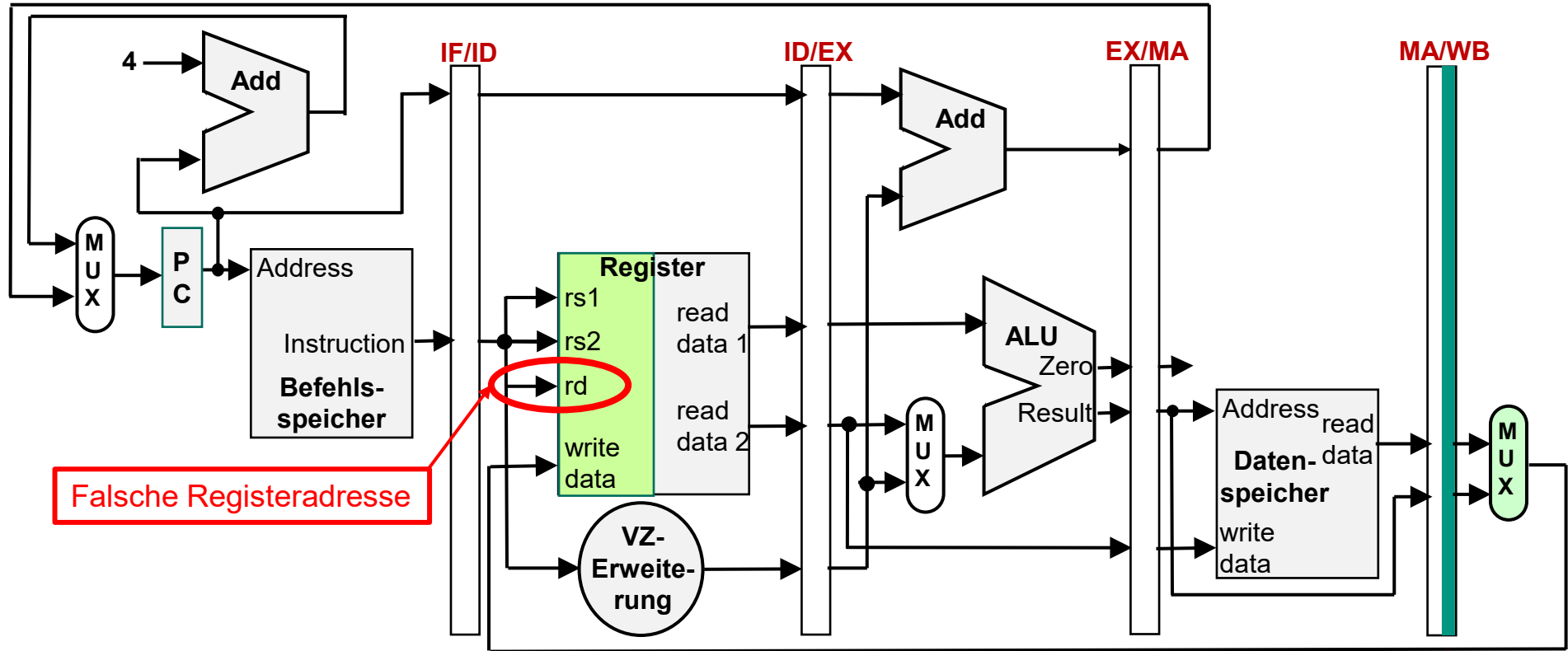
Datenpfad für RISC-V: Pipeline-Implementierung



1w-Befehl durchläuft die einzelnen Pipeline-Stufen

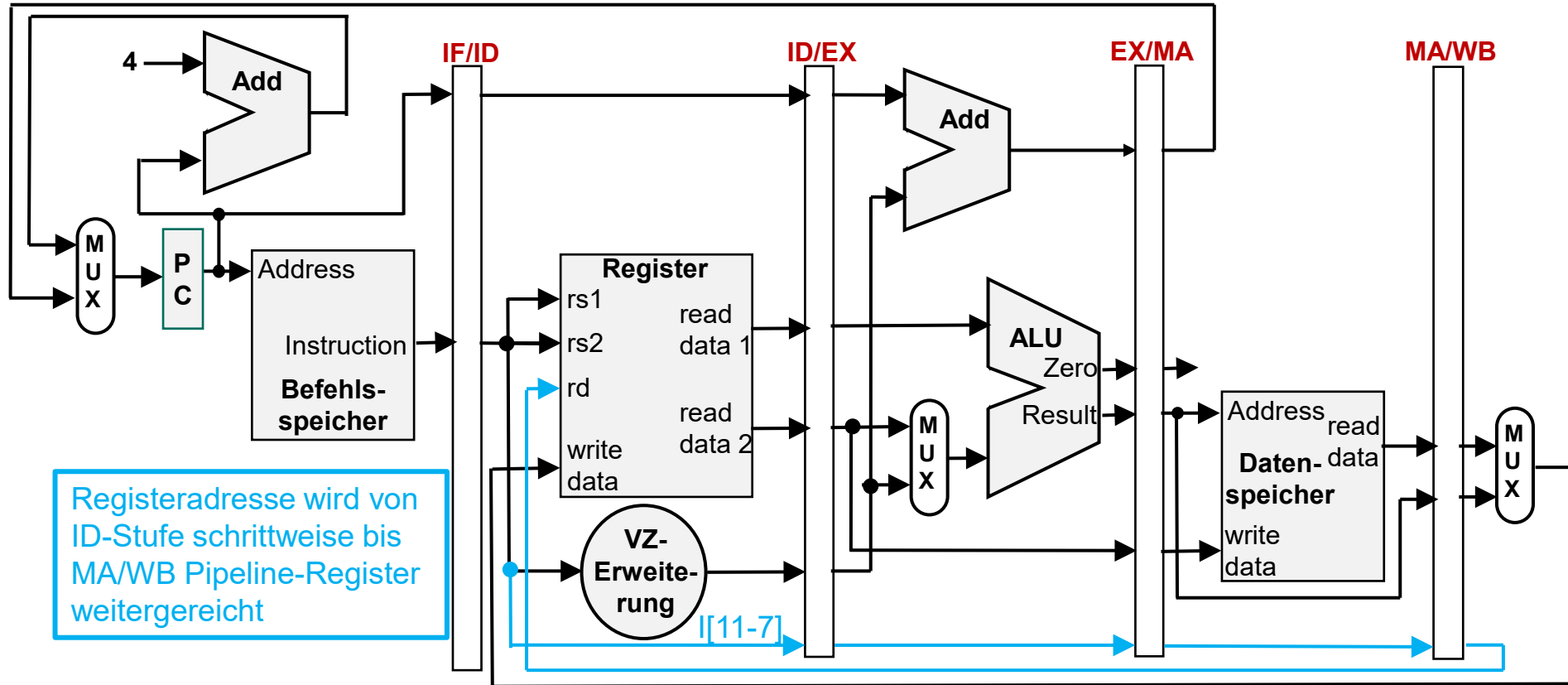
WB: 1w

Datenpfad für RISC-V: Pipeline-Implementierung



WB: 1w

Datenpfad für RISC-V: Pipeline-Implementierung



Registeradresse wird von ID-Stufe schrittweise bis MA/WB Pipeline-Register weitergereicht

Pipelining

■ Befehlsabarbeitung

■ Fallstudie RISC-V: Phasen des Maschinenbefehlszyklus für `lw`-Befehl

1. Hole Befehl aus dem Speicher (Befehl holen, instruction fetch, **IF**)
 - Der Befehl wird mit der Adresse im PC aus dem Befehlsspeicher geholt und in das IF/ID-Pipelinerregister geladen.
 - Der PC wird um 4 inkrementiert und in den PC geschrieben, damit die Adresse für den nächsten Taktzyklus bereit steht. Der Inhalt des PC wird ebenso in das IF/ID-Pipelinerregister geladen, falls er für nachfolgende Befehle (z.B. `beq`) benötigt wird.

Pipelining

■ Befehlsabarbeitung

■ Fallstudie RISC-V: Phasen des Maschinenbefehlszyklus für $1w$ -Befehl

2. Dekodiere Befehl und lese Register (Befehl dekodieren, instruction decode, **ID**)
 - Das IF/ID-Pipelineregister liefert die Registeradresse des zu lesenden Registers sowie die Konstante, die als Eingabe für die Vorzeichenerweiterung dient. Der aus dem adressierten Register gelesene Wert sowie die vorzeichenerweiterte 32-Bit-Konstante werden in das ID/EX-Pipelineregister geschrieben.

Pipelining

■ Befehlsabarbeitung

■ Fallstudie RISC-V: Phasen des Maschinenbefehlszyklus für $1w$ -Befehl

3. Führe die Operation aus oder berechne Adresse (Befehl ausführen, instruction execute, **EX**)
 - Es wird der Inhalt des Registers sowie die vorzeichenerweiterte 32-Bit-Konstante vom ID/EX-Pipeline-Register gelesen. Beide Operanden werden in der ALU addiert. Das Ergebnis wird in das EX/MA-Pipeline-Register geschrieben

Pipelining

■ Befehlsabarbeitung

■ Fallstudie RISC-V: Phasen des Maschinenbefehlszyklus für lw -Befehl

4. Greife auf Datum im Speicher zu, falls notwendig (Speicherzugriff, memory access, **MA**)
 - Der Lade-Befehl liest den Inhalt der adressierten Speicherzelle des Datenspeichers. Die Adresse kommt vom EX/MA-Pipeline-Register. Das Ergebnis wird in das MA/WB-Pipelinerregister geschrieben.

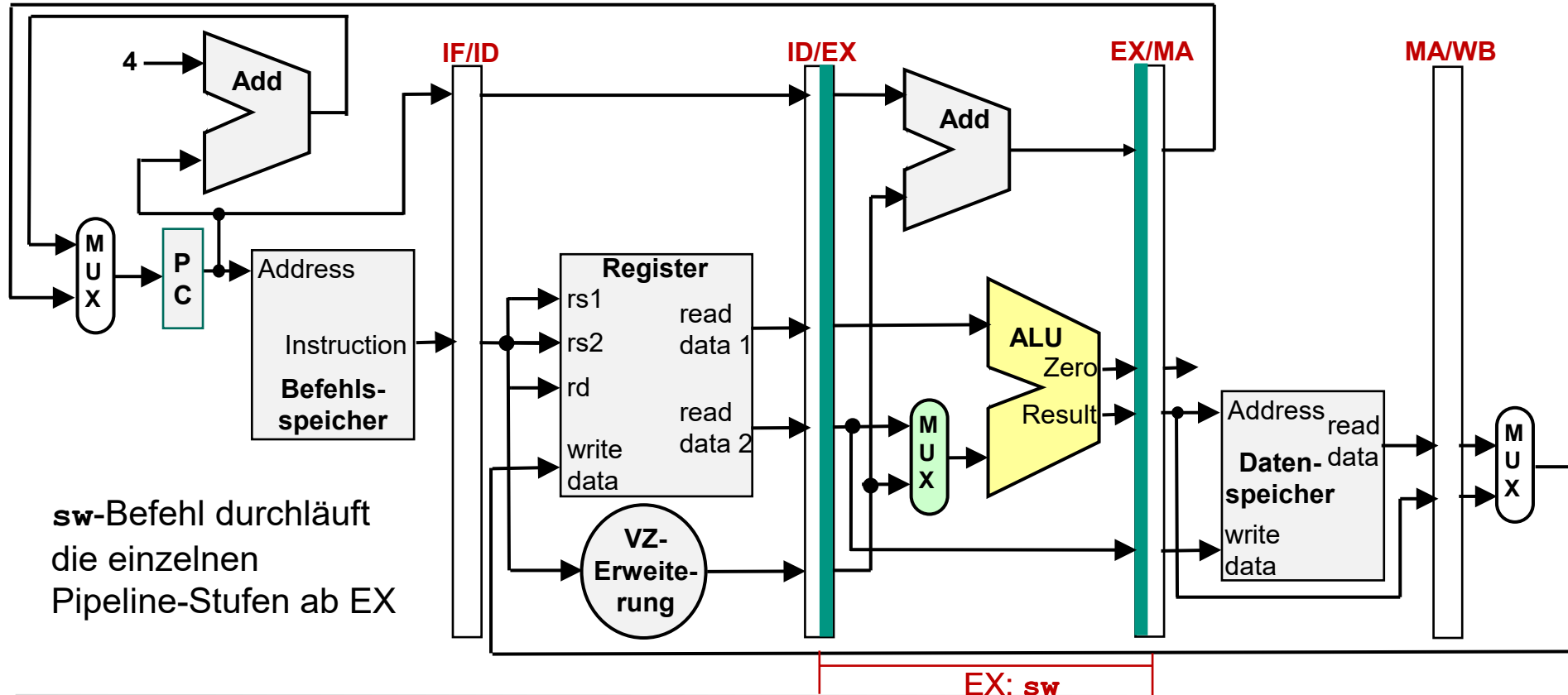
Pipelining

■ Befehlsabarbeitung

■ Fallstudie RISC-V: Phasen des Maschinenbefehlszyklus für $1w$ -Befehl

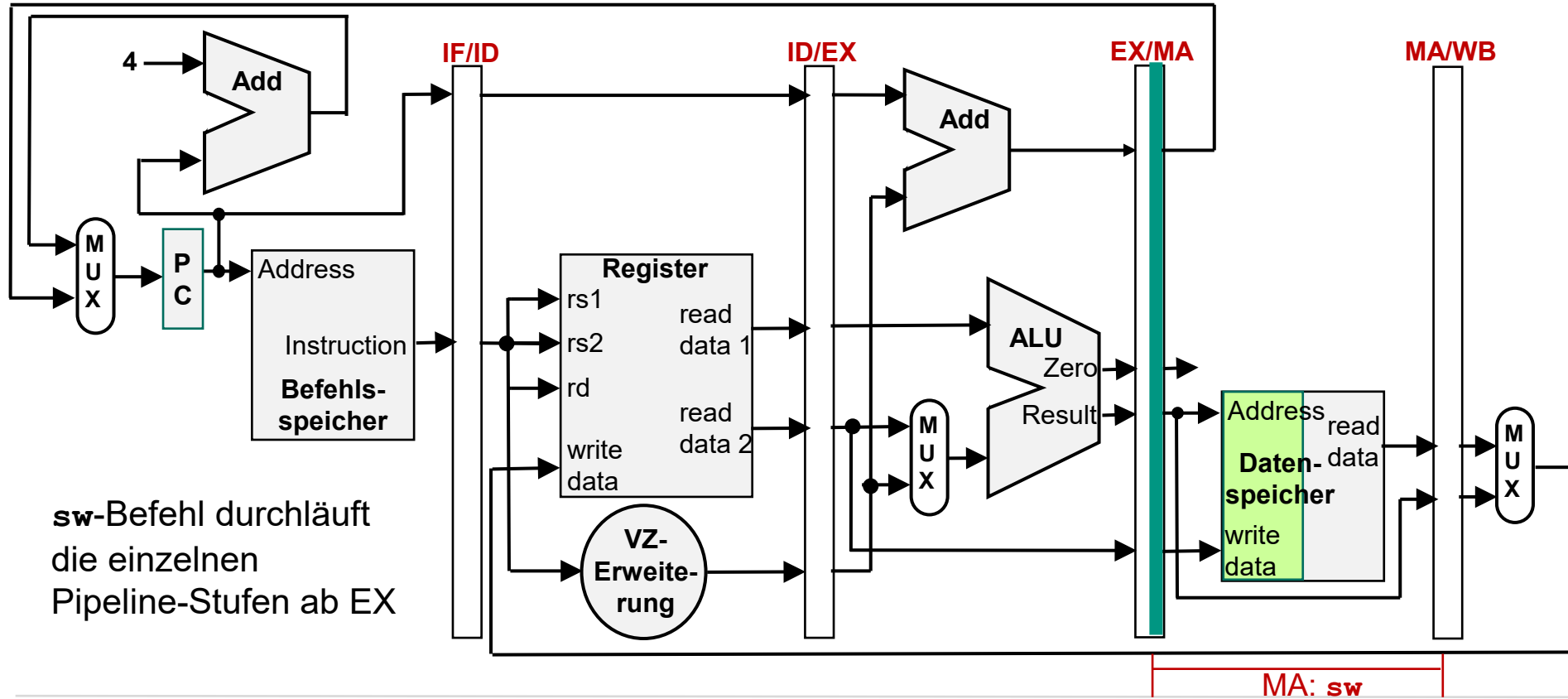
5. Schreibe Ergebnis in ein Register, falls notwendig (Ergebnis in Register schreiben, write back, **WB**)
 - Die Daten werden vom MA/WB-Pipelinerregister gelesen und in das Zielregister geschrieben.

Datenpfad für RISC-V: Pipeline-Implementierung



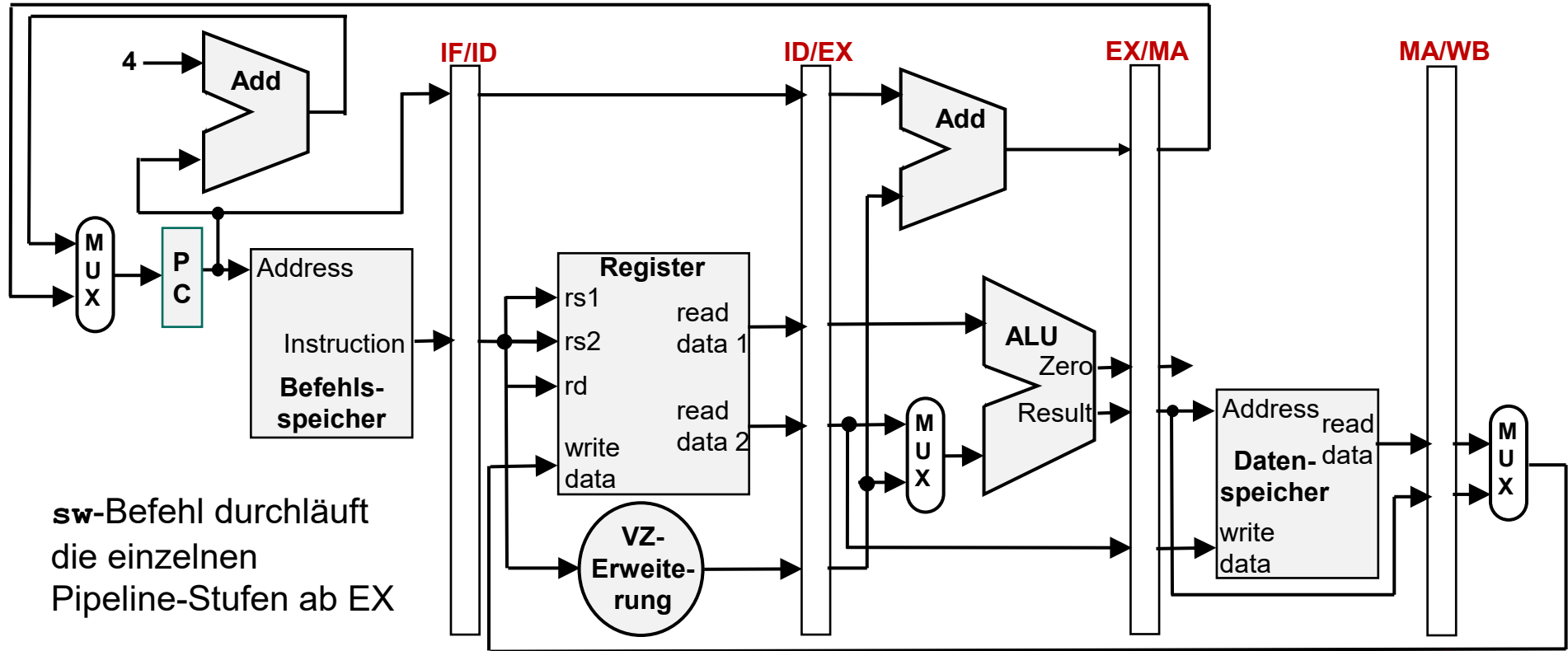
sw-Befehl durchläuft die einzelnen Pipeline-Stufen ab EX

Datenpfad für RISC-V: Pipeline-Implementierung



sw-Befehl durchläuft die einzelnen Pipeline-Stufen ab EX

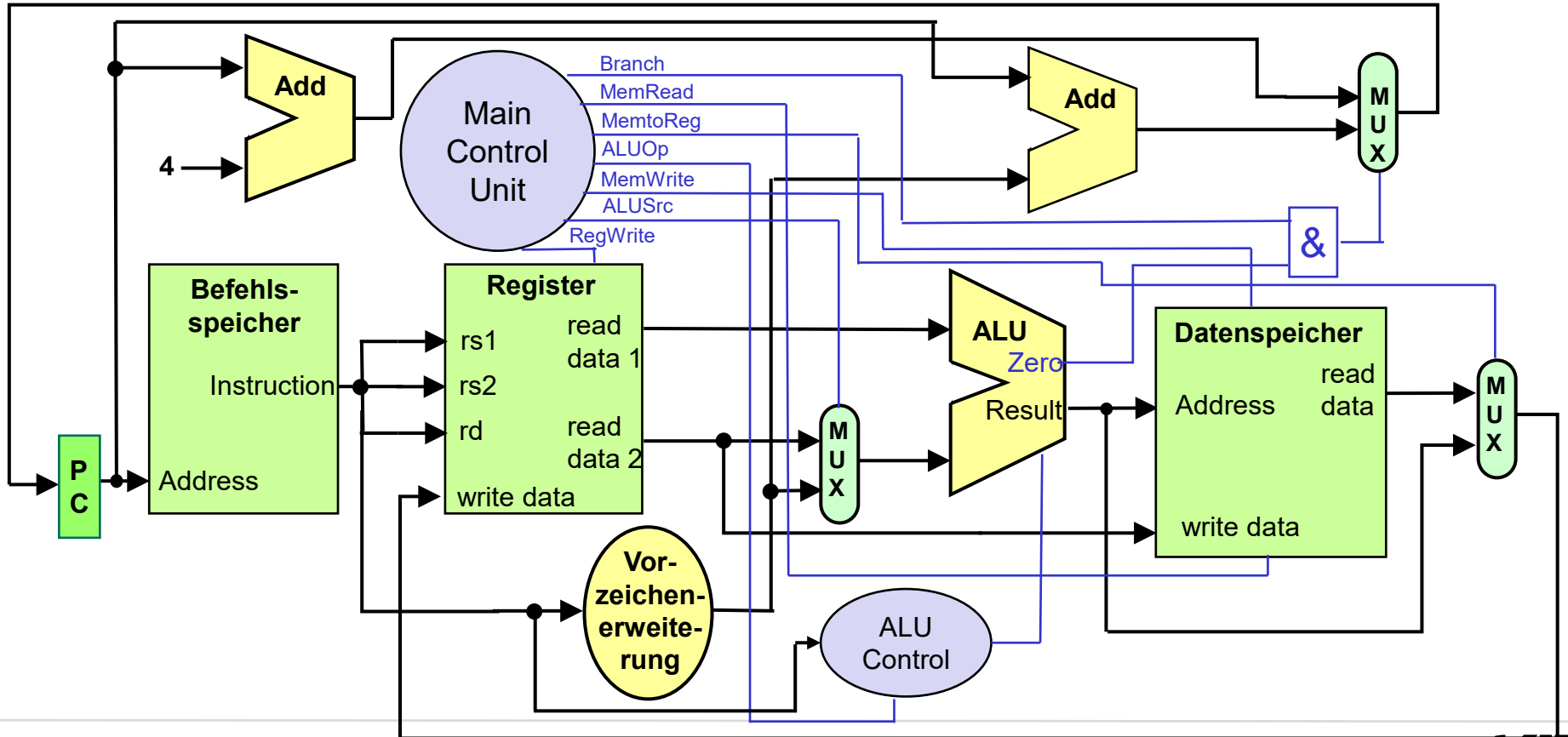
Datenpfad für RISC-V: Pipeline-Implementierung



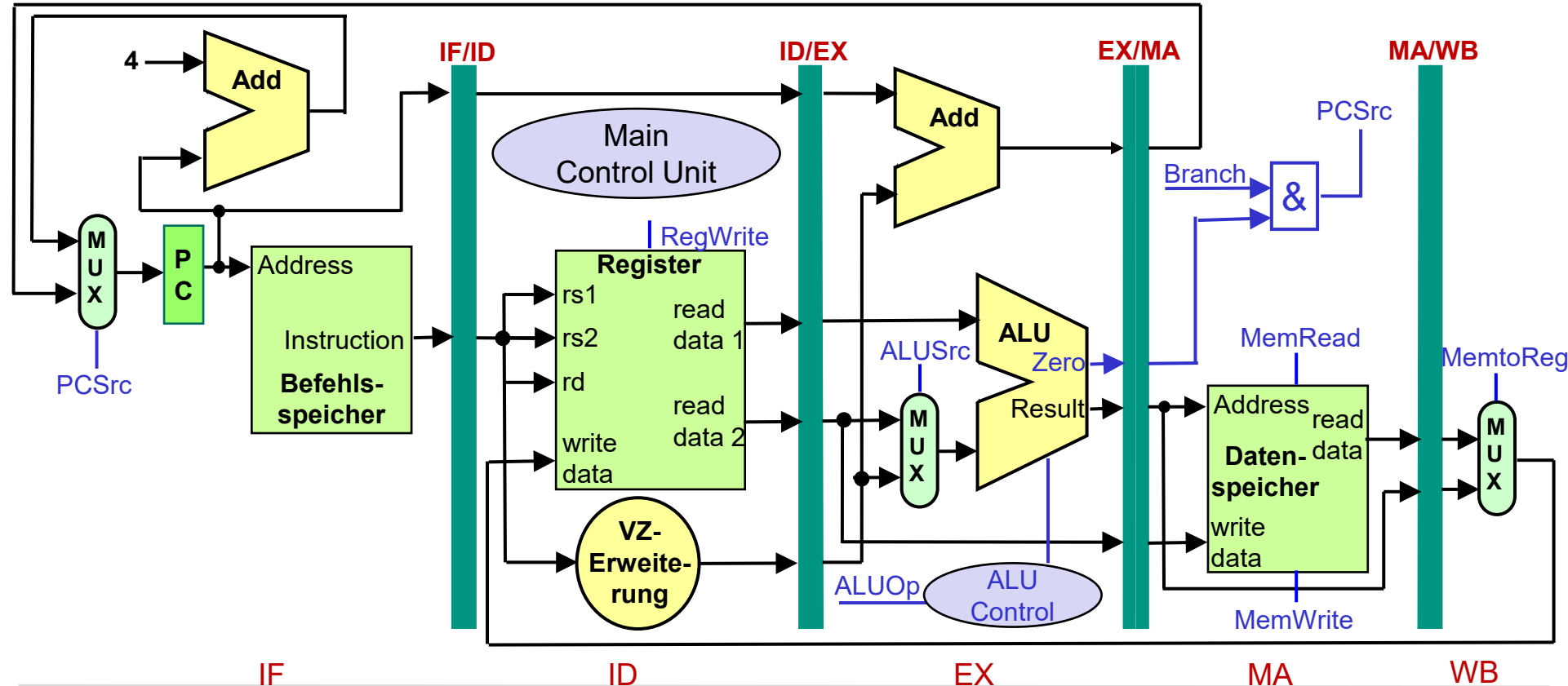
sw-Befehl durchläuft die einzelnen Pipeline-Stufen ab EX

WB: sw

Datenpfad für RISC-V mit Steuerung



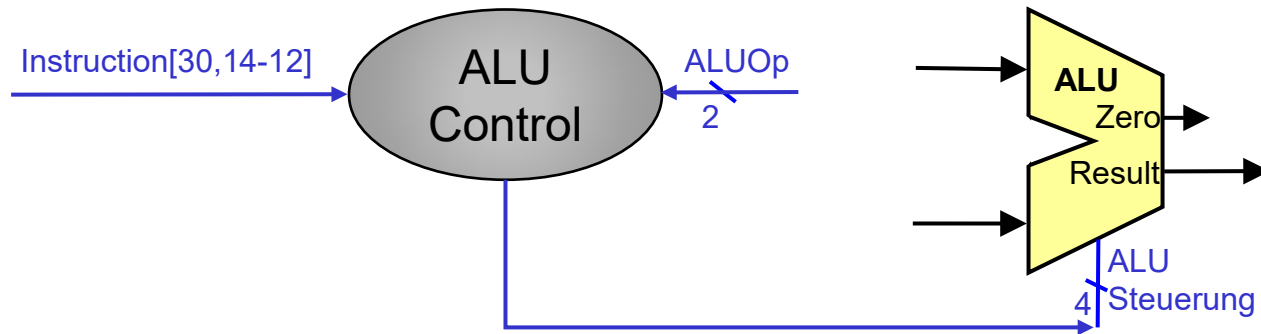
Datenpfad für RISC-V mit Steuerung



Datenpfad für RISC-V: Pipeline-Implementierung

■ ALU Control

- Belegung der ALU Steuerung wird über die Einheit ALU Control in Abhängigkeit der Signale $I[30, 14-12]$ und $ALUOp1$ und $ALUOp2$ bestimmt.
 - Siehe Foliensatz RS25-FS-06: S. 28



Datenpfad für RISC-V: Pipeline-Implementierung

■ ALU Control

■ Belegung der ALU Steuerung (Eingang[3:0])

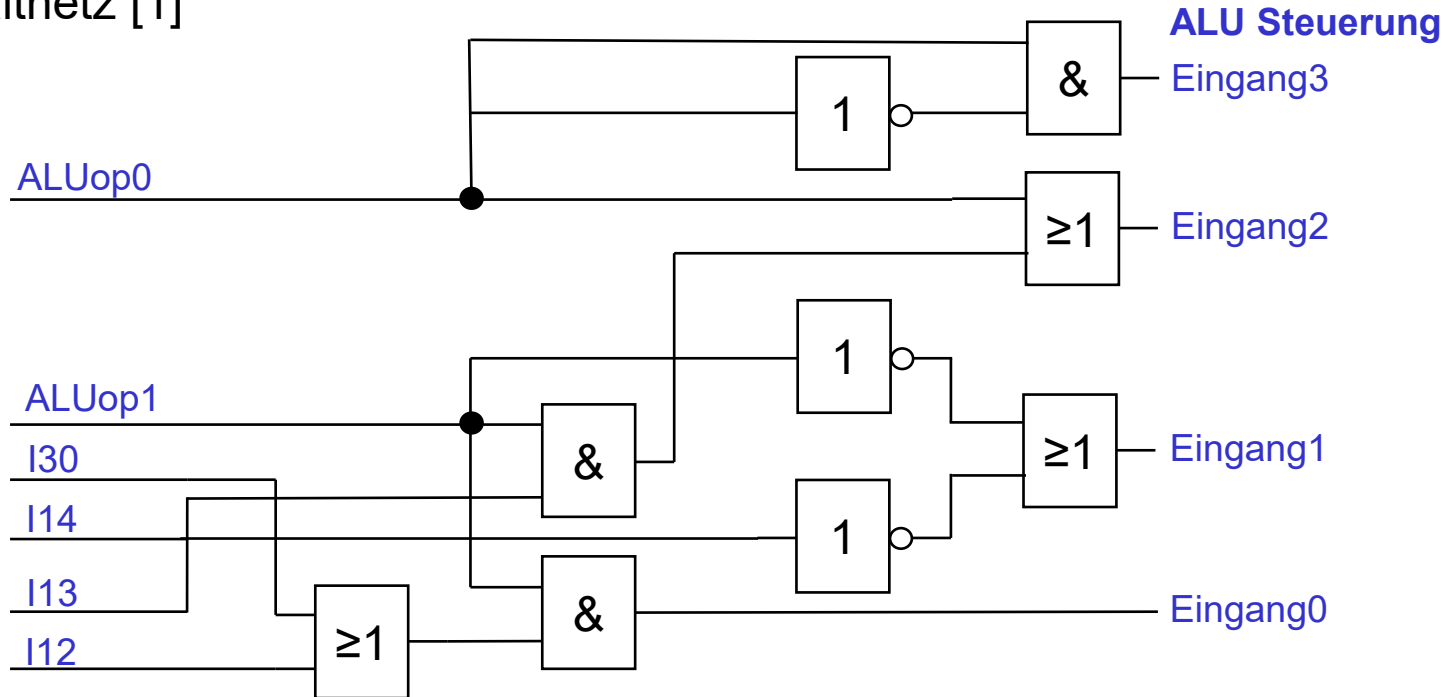
- Bedeutung der Steuersignale ändern sich nicht (siehe Foliensatz RS252-FS-06: S. 30)

Opcode des Befehls	ALUOp	Operation	funct7-Feld (Bits 31:25)	funct3-Feld (Bits 14:12)	ALU Operation	ALU Steuerung (Eingang)
lw	00	load word	XXXXXXXX	XXX	add	0010
sw	00	store word	XXXXXXXX	XXX	add	0010
beq	01	branch if equal	XXXXXXXX	XXX	sub	0110
R-Type	10	Addition	0000000	000	add	0010
R-Type	10	Subtraktion	0100000	000	sub	0110
R-Type	10	AND	0000000	111	and	0000
R-Type	10	OR	0000000	110	or	0001

Datenpfad für RISC-V: Pipeline-Implementierung

■ ALU Control

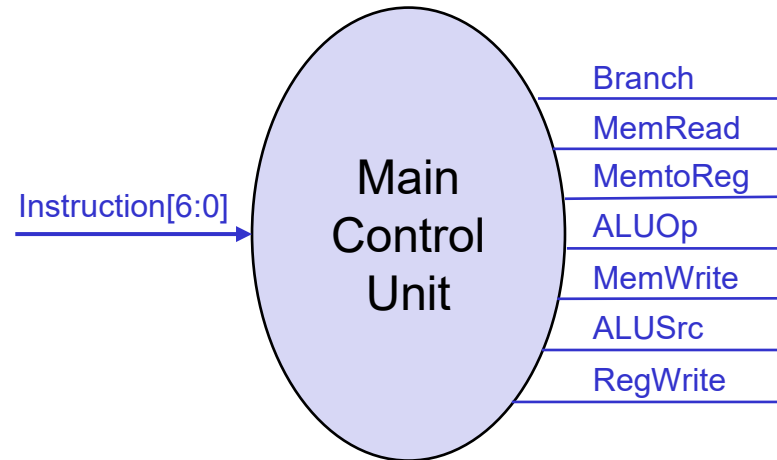
■ Schaltnetz [1]



[1] (siehe Foliensatz RS25-FS-06: S. 32)

Datenpfad für RISC-V: Pipeline-Implementierung

■ Main Control Unit (Control)



[1] (siehe Foliensatz RS25-FS-06: S. 32)

Datenpfad für RISC-V: Pipeline-Implementierung

- Main Control Unit (Control)
 - Bedeutung der Steuersignale

Signalname	Wirkung, wenn logisch 0	Wirkung, wenn logisch 1
RegWrite	Keine	Das am rd -Eingang adressierte Register wird mit dem Wert am write data Eingang beschrieben
ALUSrc	Als zweiter ALU-Operand wird der am read data 2 Ausgang der Registerdatei ausgewählt	Als zweiter ALU-Operand wird die vorzeichenerweiterte 12-Bit-Konstante des Befehls ausgewählt
PCSrc	Der neue Wert des Befehlszählers (PC) wird durch den Ausgangswert des Addierers ersetzt, der PC+4 berechnet	Der neue Wert des Befehlszählers (PC) wird durch den Ausgangswert des Addierers ersetzt, der das Sprungziel berechnet

[1] (siehe Foliensatz RS25-FS-06: S. 35)

Datenpfad für RISC-V: Pipeline-Implementierung

- **Main Control Unit (Control)**
 - Bedeutung der Steuersignale

Signalname	Wirkung, wenn logisch 0	Wirkung, wenn logisch 1
MemRead	Keine	Mit der am Address -Eingang des Datenspeichers anliegenden Adresse wird die Speicherzelle ausgewählt, deren Inhalt über den read data Ausgang ausgegeben wird
MemWrite	Keine	Mit der am Address -Eingang des Datenspeichers anliegenden Adresse wird die Speicherzelle ausgewählt, deren Inhalt mit dem am write data Eingang anliegenden Wert überschrieben wird
MemToReg	Der am write data Eingang der Registerdatei anzulegende Wert kommt vom ALU Ausgang	Der am write data Eingang der Registerdatei anzulegende Wert kommt vom read data Ausgang des Datenspeichers

[1] (siehe Foliensatz RS25-FS-06: S. 36)

Datenpfad für RISC-V: Pipeline-Implementierung

■ Main Control Unit (Control)

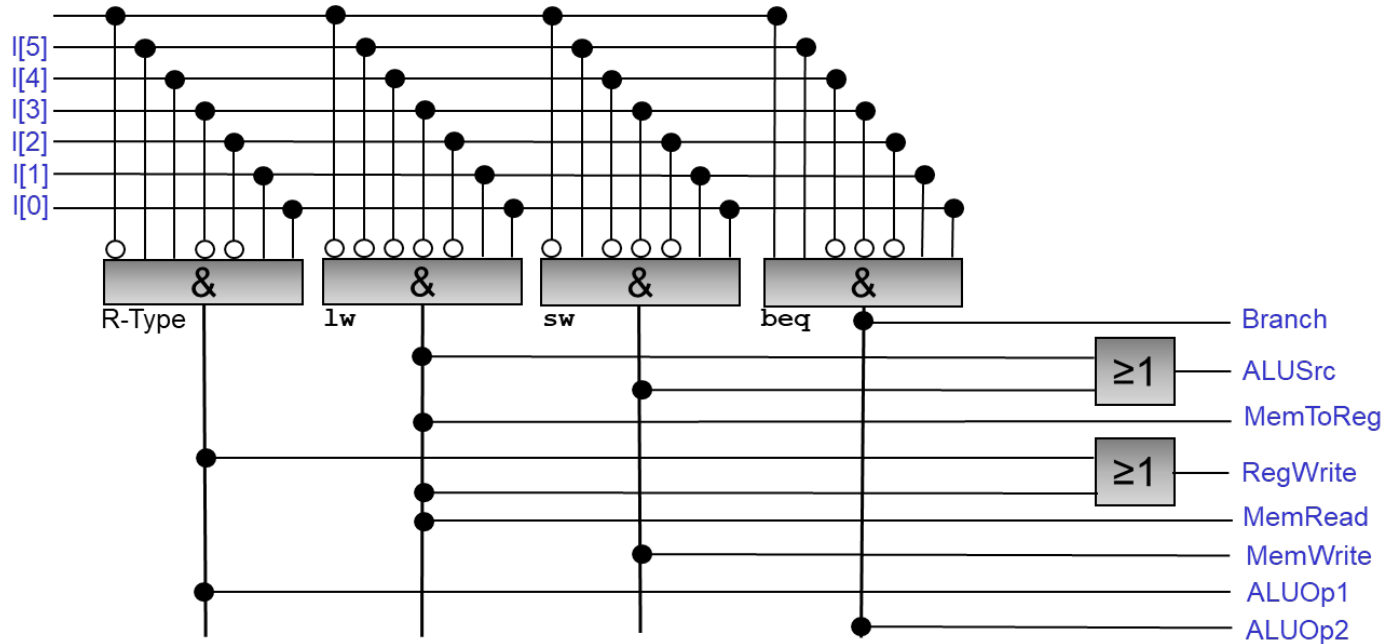
- Belegung der Steuersignale: Zuordnung zu Pipelinestufen
 - Wahrheitstabelle 2 (Foliensatz RS25-FS-06: S. 51)
 - Kodierung Befehl: Wahrheitstabelle 1 (Foliensatz RS25-FS-06: S. 50)

Befehl	EX: Ausführung / Adressrechnung Steuersignale		MA: Speicherzugriff Steuersignale			WB: Rückschreiben Steuersignale	
	ALU-Op	ALUSrc	Branch	Mem- Read	Mem- Write	Reg- Write	Mem- ToReg
R-Type	10	0	0	0	0	1	0
lw	00	1	0	1	0	1	1
sw	00	1	0	0	1	0	X
beq	01	0	1	0	0	0	X

Datenpfad für RISC-V: Pipeline-Implementierung

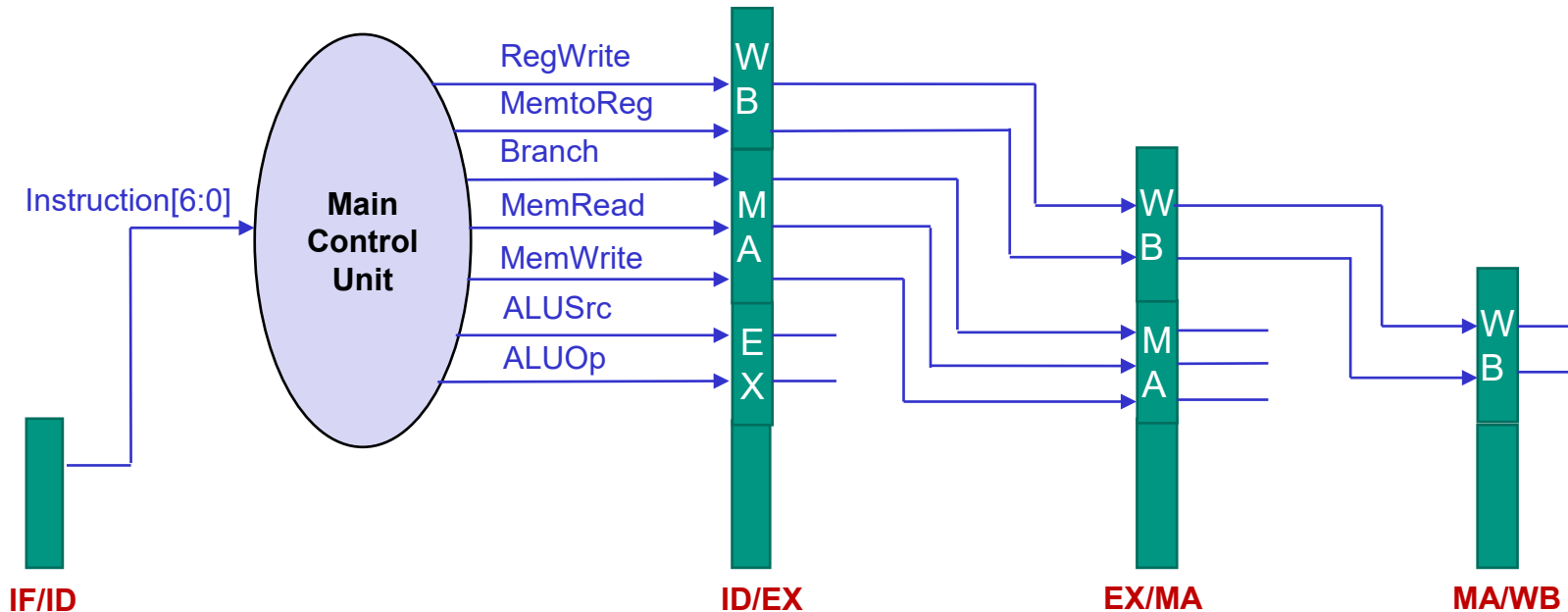
■ Main Control Unit (Control)

■ Implementierung als PLA



Datenpfad für RISC-V: Pipeline-Implementierung

- **Main Control Unit (Control) für RISC-V-Pipeline**
 - Weiterleiten der Steuersignale über die Pipelineregister



Datenpfad für RISC-V: Pipeline-Implementierung

■ Pipeline-Hemmnisse

■ Beispiel:

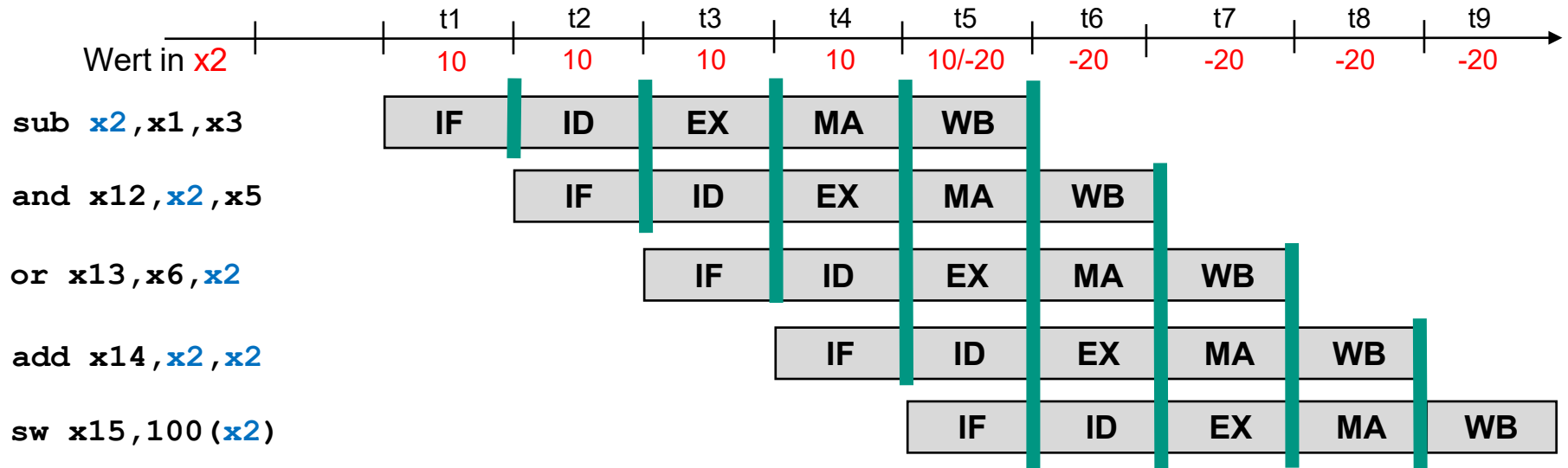
```
sub  x2, x1, x3
and  x12, x2, x5
or   x13, x6, x2
add  x14, x2, x2
sw   x15, 100 (x2)
```

Echte Datenabhängigkeiten:

Der Befehl `sub` schreibt das Ergebnis in das Register `x2`.
Die folgenden Befehle benötigen den Wert als Operand.

Datenpfad für RISC-V: Pipeline-Implementierung

- Pipeline-Hemmnisse
 - Darstellung in Pipeline

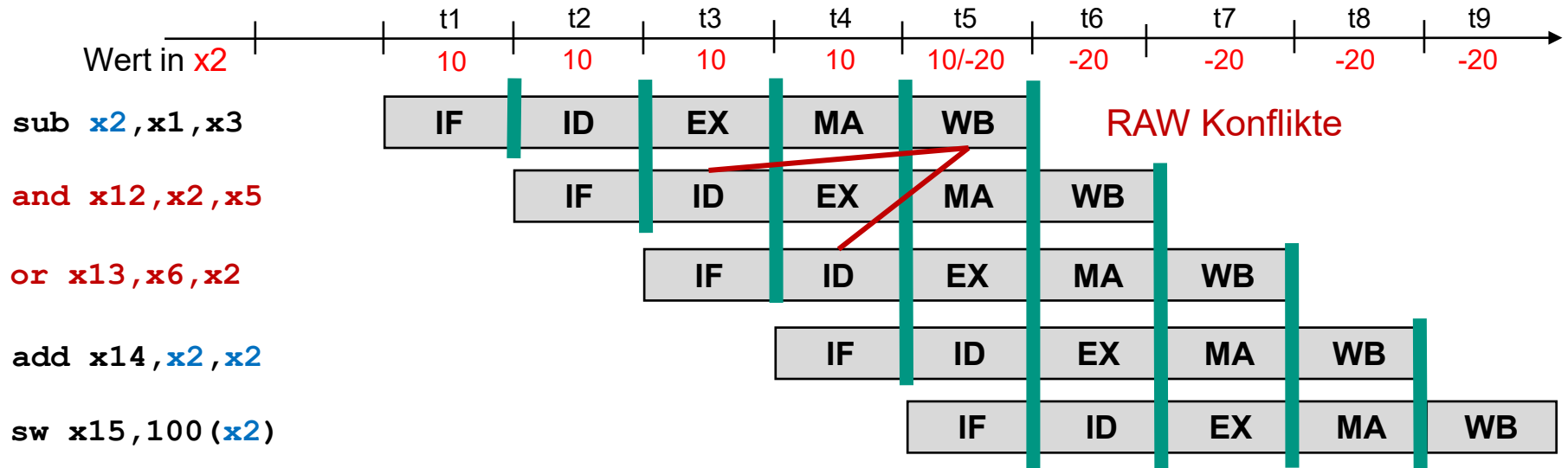


Datenpfad für RISC-V: Pipeline-Implementierung

■ Pipeline-Hemmnisse:

■ Darstellung in der Pipeline

■ Lese-nach-Schreibe-Konflikt (RAW)

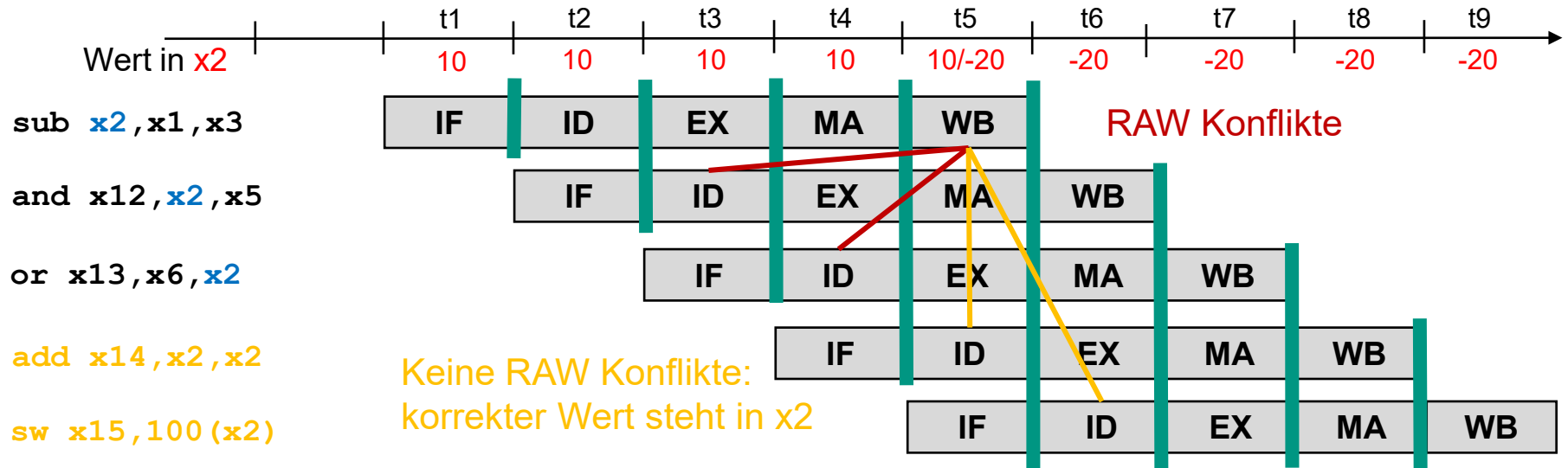


Datenpfad für RISC-V: Pipeline-Implementierung

■ Pipeline-Hemmnisse:

■ Darstellung in der Pipeline

■ Kein Lese-nach-Schreib-Konflikt

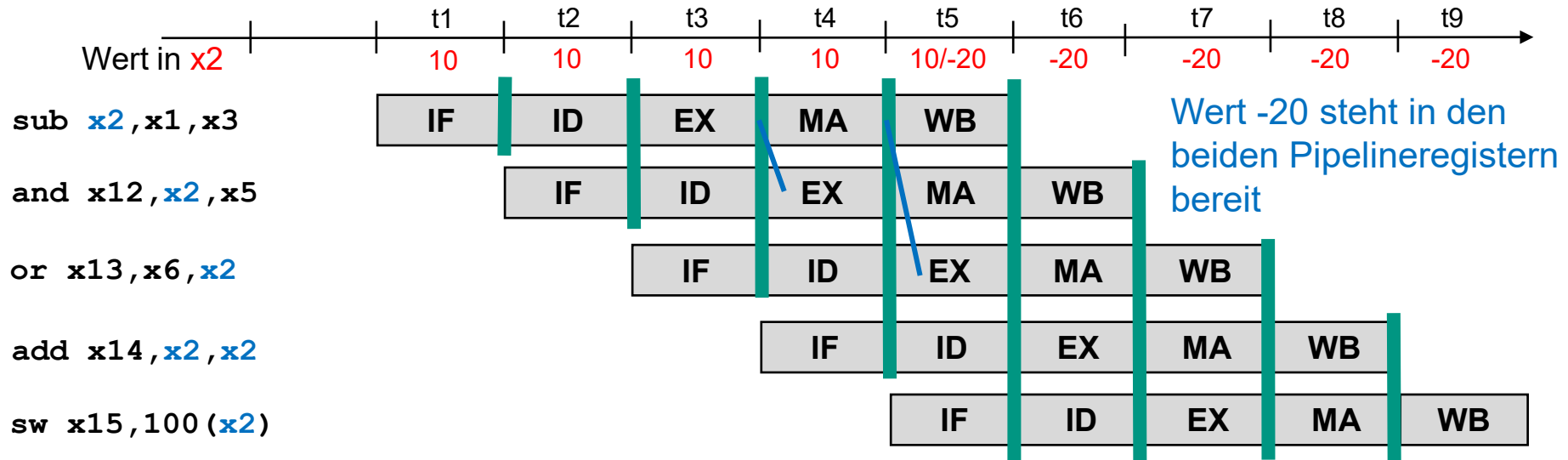


Datenpfad für RISC-V: Pipeline-Implementierung

■ Pipeline-Hemmnisse:

■ Konfliktauflösung durch Forwarding

- Bereitstellen des Ergebnisses für nachfolgende Befehle

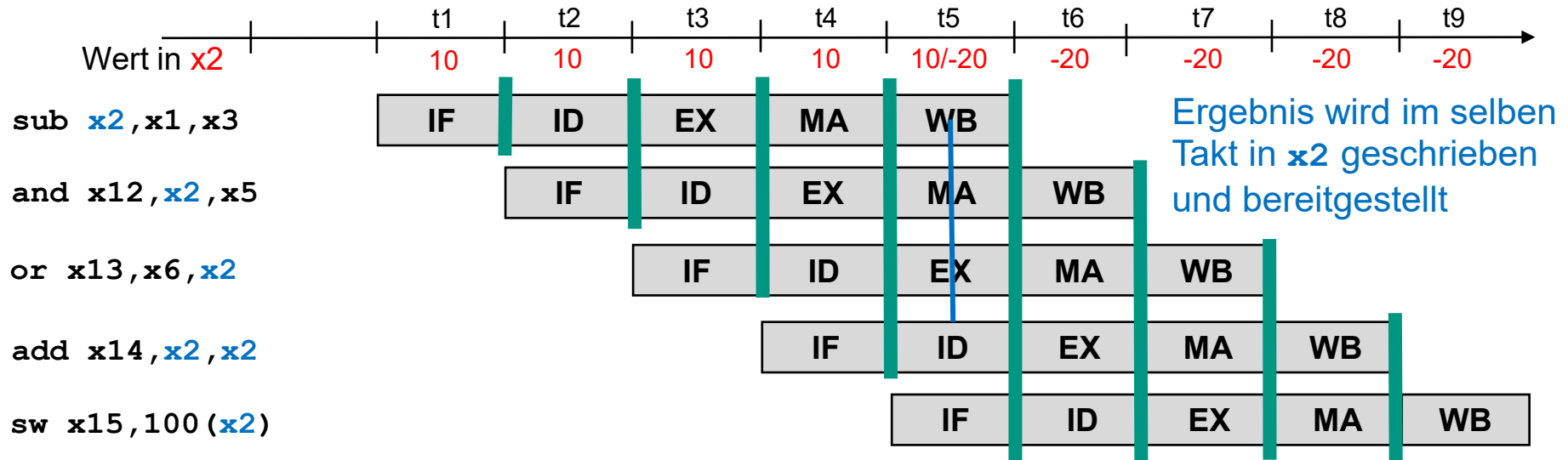


Datenpfad für RISC-V: Pipeline-Implementierung

■ Pipeline-Hemmnisse:

■ Konfliktauflösung durch Forwarding

- Bereitstellen des Ergebnisses für nachfolgende Befehle



Datenpfad für RISC-V: Pipeline-Implementierung

■ Erkennen von RAW-Konflikten

- Pipelineregister puffern Informationen, die in den nachfolgenden Pipelinestufen benötigt werden
 - Beispiel: IF/ID Pipelineregister: speichert den geholten Befehl in entsprechenden Feldern
 - Notation:
 - Für die Felder in den Pipelineregistern werden Namen eingeführt:
 - Das Feld im Pipelineregister **ID/EX**, das die Registeradresse enthält, die im rs1-Feld des Befehls steht, heisst: **ID/EX.RegisterRs1**
 - ... analog für rs2

Datenpfad für RISC-V: Pipeline-Implementierung

■ Erkennen von RAW-Konflikten

■ Bedingungen für RAW-Konflikte in der Pipeline:

- (1) EX/MA.RegisterRd = ID/EX.RegisterRs1
- (2) EX/MA.RegisterRd = ID/EX.RegisterRs2
- (3) MA/WB.RegisterRd = ID/EX.RegisterRs1
- (4) MA/WB.RegisterRd = ID/EX.RegisterRs2

Datenpfad für RISC-V: Pipeline-Implementierung

■ Erkennen von RAW-Konflikten

■ Bedingungen für RAW-Konflikte in der Pipeline:

- (1) EX/MA.RegisterRd = ID/EX.RegisterRs1
- (2) EX/MA.RegisterRd = ID/EX.RegisterRs2
- (3) MA/WB.RegisterRd = ID/EX.RegisterRs1
- (4) MA/WB.RegisterRd = ID/EX.RegisterRs2

■ Beispiel:

```
sub  x2, x1, x3
and  x12, x2, x5
or   x13, x6, x2
add  x14, x2, x2
sw   x15, 100(x2)
```

RAW-Konflikt kann entdeckt werden, wenn der `and`-Befehl in EX-Stufe ist und der vorangehende `sub`-Befehl in der MA-Stufe ist.

Datenpfad für RISC-V: Pipeline-Implementierung

■ Erkennen von RAW-Konflikten

■ Bedingungen für RAW-Konflikte in der Pipeline:

- (1) EX/MA.RegisterRd = ID/EX.RegisterRs1
- (2) EX/MA.RegisterRd = ID/EX.RegisterRs2
- (3) MA/WB.RegisterRd = ID/EX.RegisterRs1
- (4) MA/WB.RegisterRd = ID/EX.RegisterRs2

■ Beispiel:

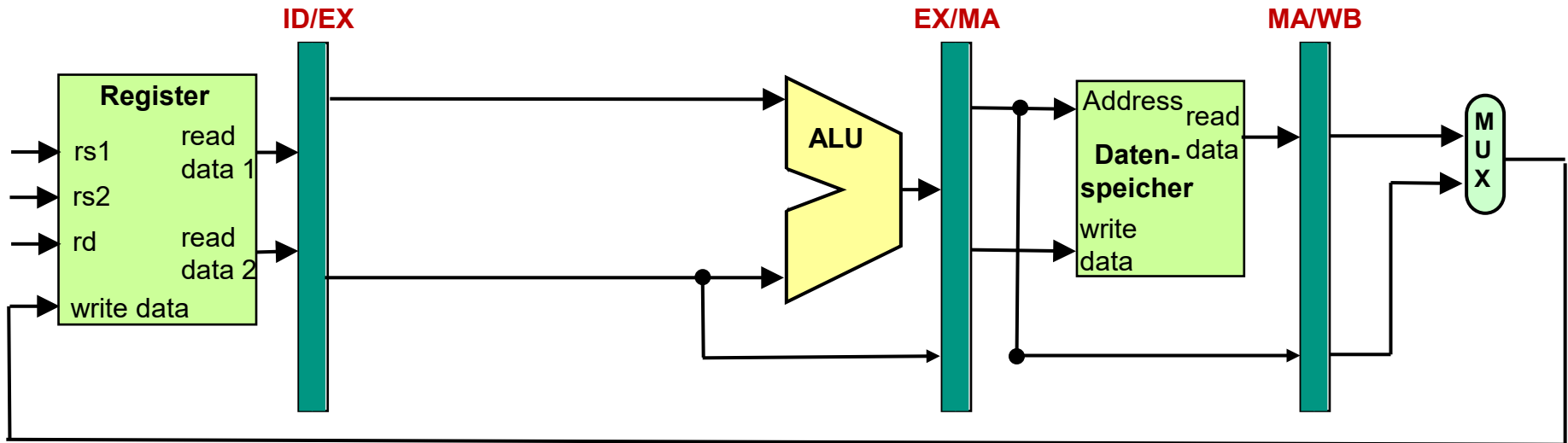
```
sub  x2, x1, x3  
and  x12, x2, x5  
or   x13, x6, x2  
add  x14, x2, x2  
sw   x15, 100(x2)
```

Bedingung:
EX/MA.RegisterRd = ID/EX.RegisterRs1 = x2

Datenpfad für RISC-V: Pipeline-Implementierung

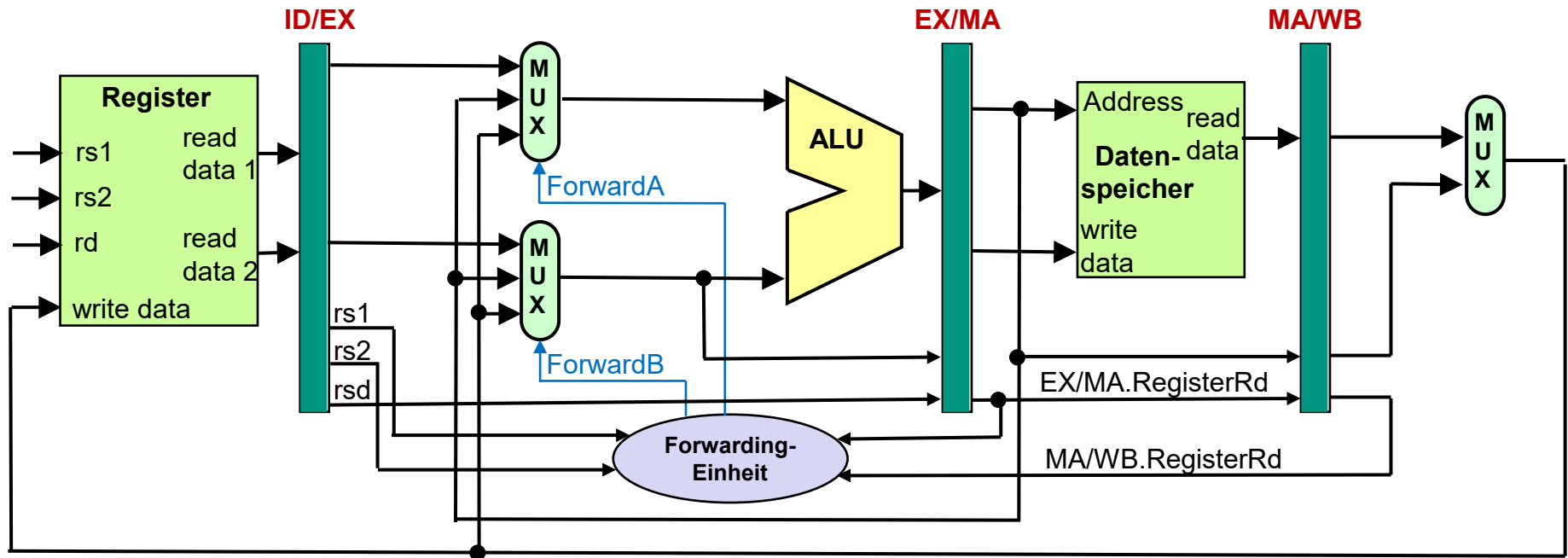
■ Auflösen von RAW-Konflikten: Result-Forwarding

- Datenpfad ohne Forwarding-Logik (vereinfachte Darstellung)



Datenpfad für RISC-V: Pipeline-Implementierung

- Auflösen von RAW-Konflikten: Result-Forwarding
 - Datenpfad mit Forwarding-Logik (vereinfachte Darstellung)



Datenpfad für RISC-V: Pipeline-Implementierung

■ Auflösen von RAW-Konflikten: Result-Forwarding

■ Forwarding-Einheit

- Steuersignale **ForwardA** und **ForwardB** steuern die Multiplexer zur Auswahl der an die ALU-Eingänge anzulegenden Quellen
 - Inhalte der Register
 - Wert über Forwarding-Pfad
- Teil der EX-Stufe

Datenpfad für RISC-V: Pipeline-Implementierung

■ Auflösen von RAW-Konflikten: Result-Forwarding

■ Forwarding-Einheit

- Steuersignale **ForwardA** und **ForwardB**

Mux-Steuerung	Quelle	Wirkung
ForwardA=00	ID/EX	Der 1. ALU-Operand kommt von der Registerdatei
ForwardA=10	EX/MA	Der 1. ALU-Operand ist das von der ALU kommende Ergebnis (forwarding)
ForwardA=01	MA/WB	Der 1. ALU-Operand ist der vom Speicher kommende Wert oder ist ein früheres ALU-Ergebnis (forwarding)
ForwardB=00	ID/EX	Der 2. ALU-Operand kommt von der Registerdatei (forwarding)
ForwardB=10	EX/MA	Der 2. ALU-Operand ist das von der ALU kommende Ergebnis (forwarding)
ForwardB=01	MA/WB	Der 2. ALU-Operand ist der vom Speicher kommende Wert oder ist ein früheres ALU-Ergebnis (forwarding)

Datenpfad für RISC-V: Pipeline-Implementierung

■ Auflösen von RAW-Konflikten: Result-Forwarding

■ Forwarding-Einheit

- Bedingungen zur Erkennung von RAW-Konflikte und Steuersignale zu deren Auflösung

■ Konflikt in EX Stufe

```
if (EX/MA.RegWrite  
and (EX/MA.RegisterRd ≠ 0)  
and (EX/MA.RegisterRd = ID/EX.RegisterRs1)) ForwardA = 10
```

```
if (EX/MA.RegWrite  
and (EX/MA.RegisterRd ≠ 0)  
and (EX/MA.RegisterRd = ID/EX.RegisterRs2)) ForwardB = 10
```

Datenpfad für RISC-V: Pipeline-Implementierung

■ Auflösen von RAW-Konflikten: Result-Forwarding

■ Forwarding-Einheit

- Bedingungen zur Erkennung von RAW-Konflikte und Steuersignale zu deren Auflösung

■ Konflikt in MA Stufe

```
if (MA/WB.RegWrite  
and (MA/WB.RegisterRd ≠ 0)  
and (MA/WB.RegisterRd = ID/EX.RegisterRs1)) ForwardA = 01
```

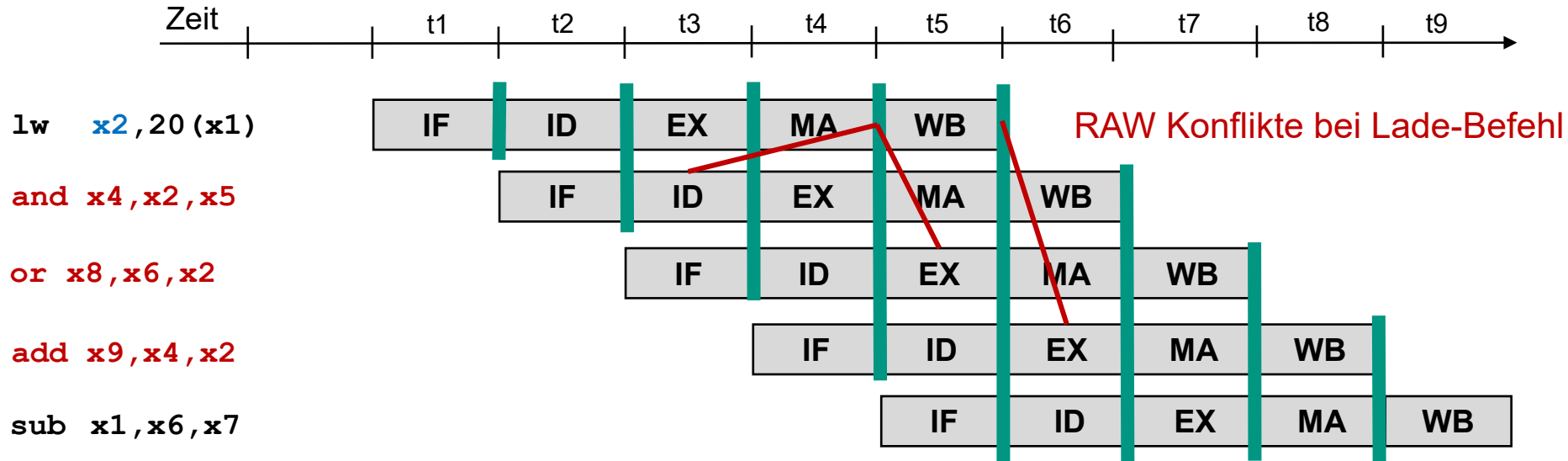
```
if (MA/WB.RegWrite  
and (MA/WB.RegisterRd ≠ 0)  
and (MA/WB.RegisterRd = ID/EX.RegisterRs2)) ForwardB = 01
```

Datenpfad für RISC-V: Pipeline-Implementierung

Erkennen von RAW-Konflikten bei Lade-Befehle

Darstellung in der Pipeline

Lese-nach-Schreib-Konflikt (RAW)

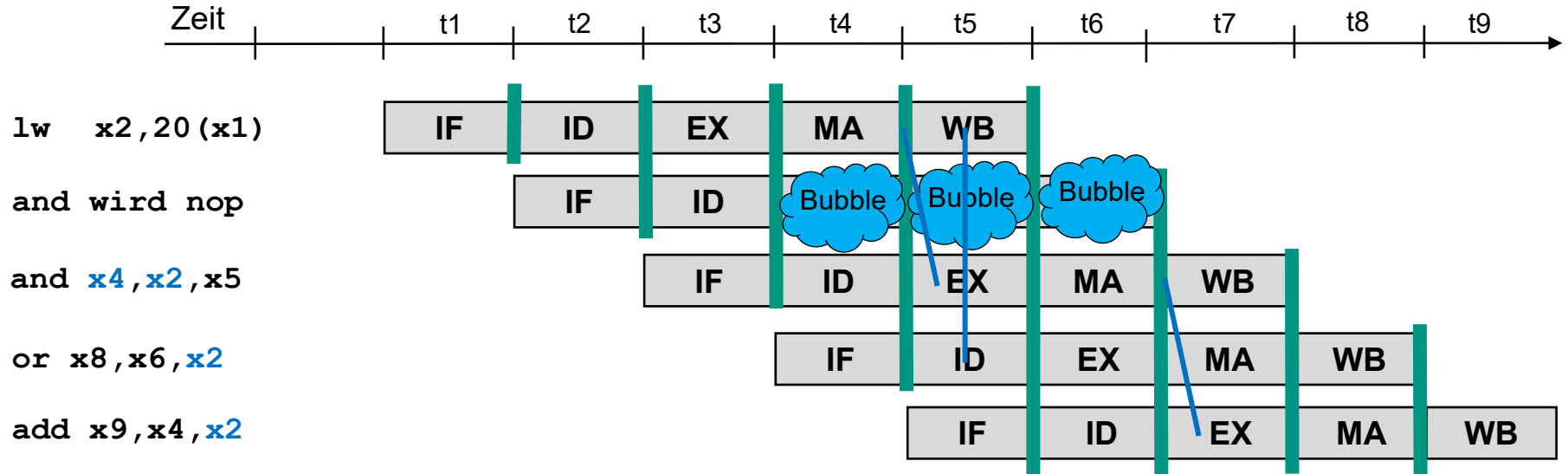


Datenpfad für RISC-V: Pipeline-Implementierung

■ Auflösen von RAW-Konflikten: Load-Forwarding

■ Darstellung in der Pipeline

- Einfügen von Leeroperationen / Verzögerungsphasen



Datenpfad für RISC-V: Pipeline-Implementierung

■ Load-Forwarding

■ Hazard-Detection-Einheit

- Erkennung von RAW-Konflikten bei Lade-Befehlen:
 - Falls ein RAW-Konflikt bei einem Lade-Befehl erkannt wird, werden Wartezyklen zwischen dem Lade-Befehl und dem von diesem abhängigen Befehl eingefügt
 - Teil der Dekodier-Stufe

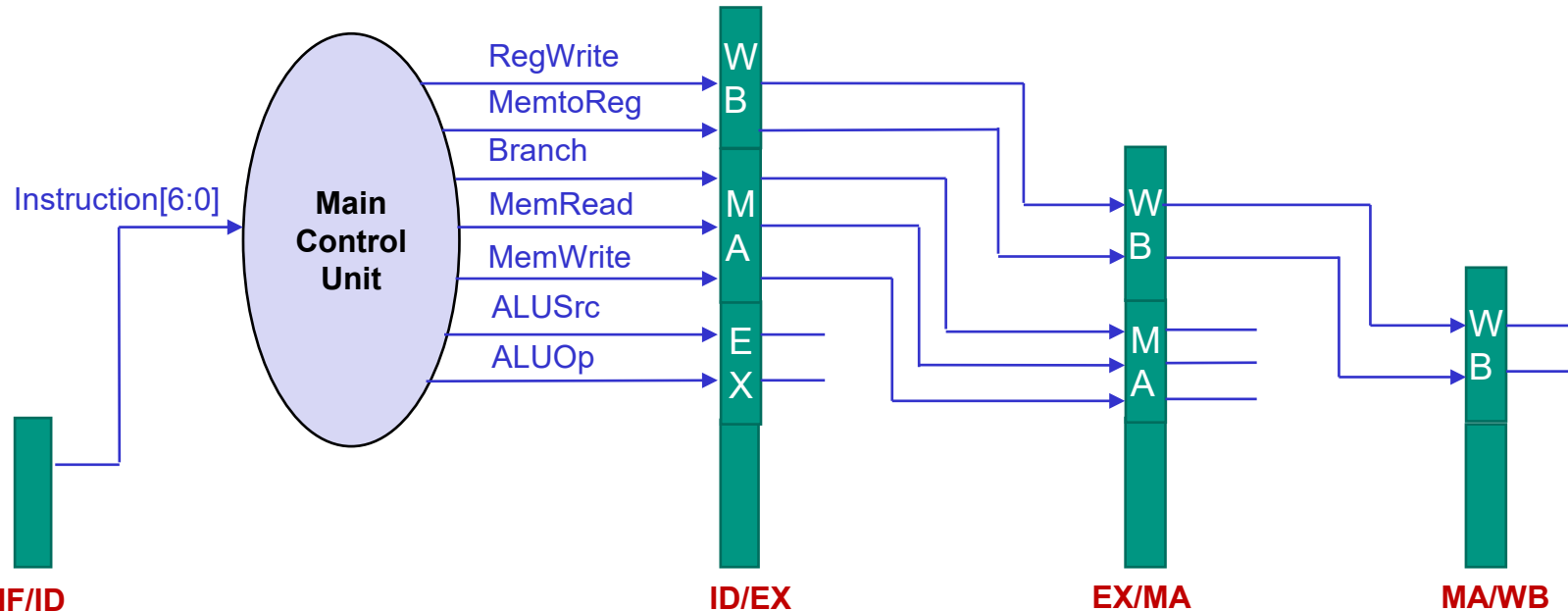
```
if (ID/EX.MemRead
and ((ID/EX.RegisterRd = IF/ID.RegisterRs1) or
     (ID/EX.RegisterRd = IF/ID.RegisterRs2)))
stall Pipeline
```

Datenpfad für RISC-V: Pipeline-Implementierung

■ Load-Forwarding

■ Main Control Unit (Control)

- Weiterleiten der Steuersignale über die Pipelineregister (siehe Folie 113)



Datenpfad für RISC-V: Pipeline-Implementierung

■ Load-Forwarding

■ Main Control Unit (Control) und Hazard Detection Unit

