

Mikroprozessor

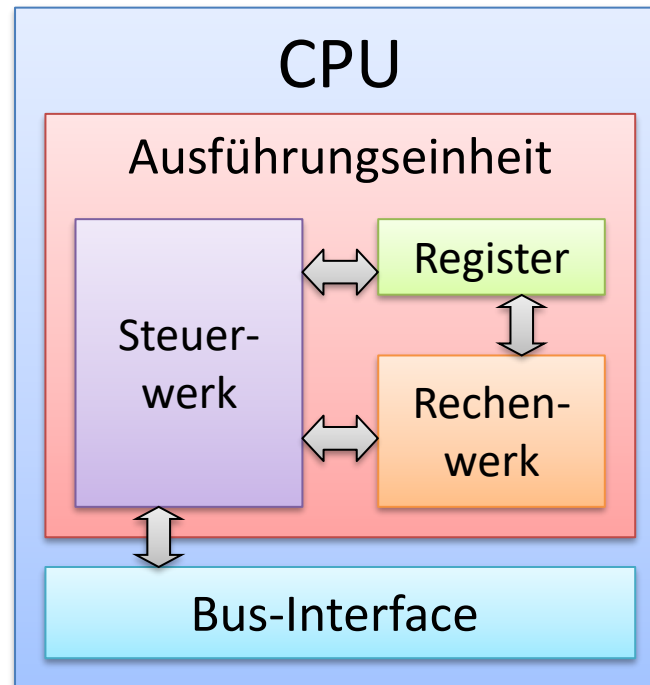
Netzwerke und Embedded Systems

1. Jahrgang

Wolfgang Neff

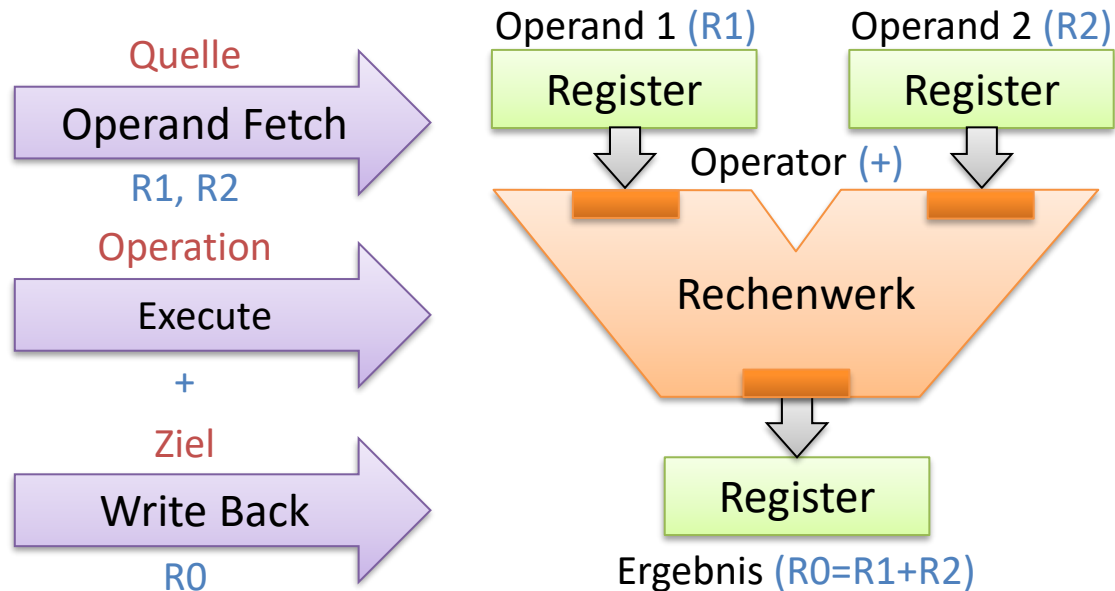
Mikroprozessor (1)

- Blockdiagramm
 - Ausführungseinheit
 - Steuerwerk
 - Register
 - Rechenwerk
 - ADD, SUB etc.
 - NOT, AND etc.
 - Bus-Interface



Mikroprozessor (2)

- Rechenwerk (ALU, Arithmetic logic unit)

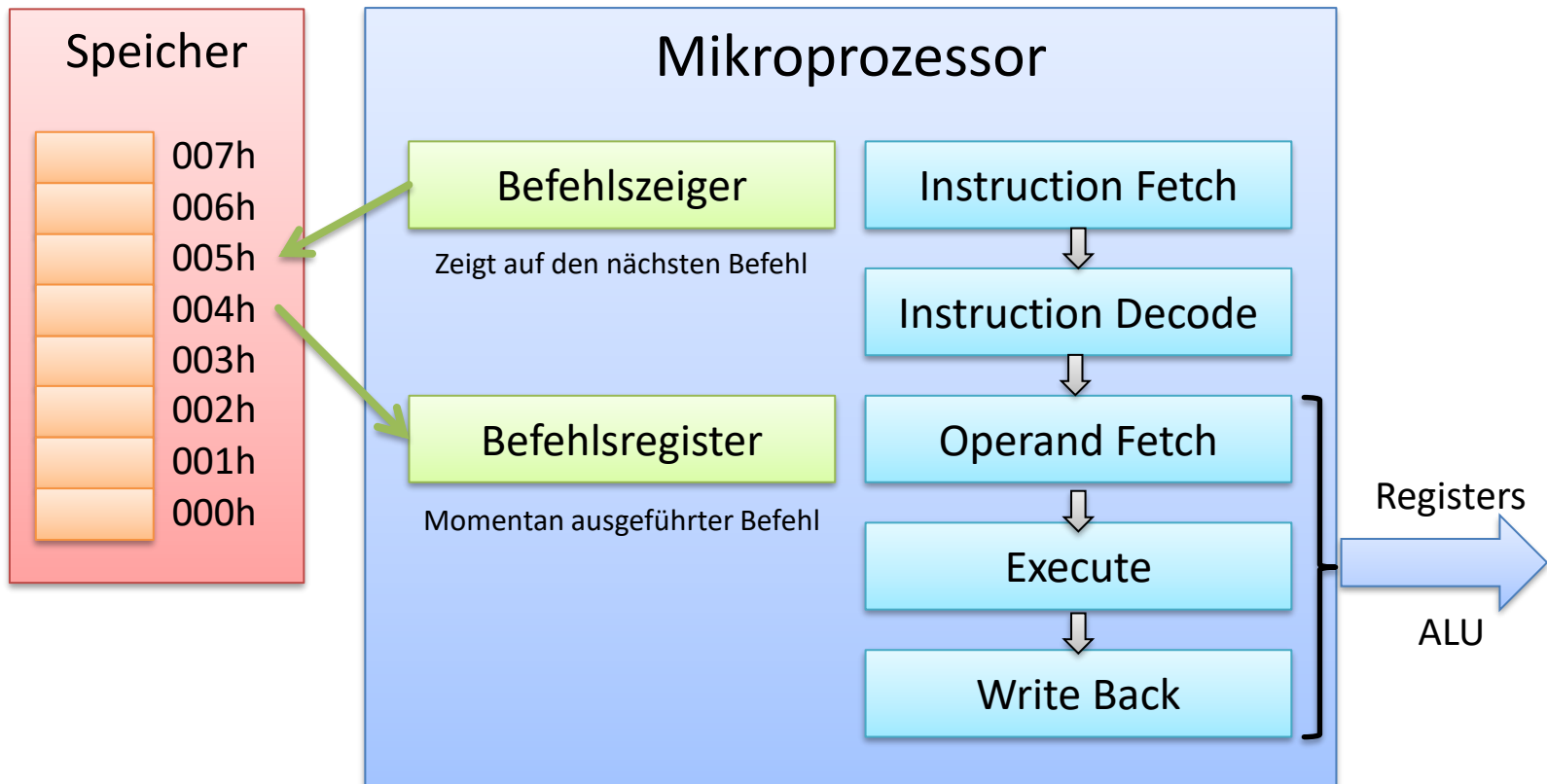


Ergebnis = Operand1 *Operator* Operand2

Ziel = Quelle1 *Operator* Quelle2

Mikroprozessor (3)

- Aufbau des Steuerwerks



Mikroprozessor (4)

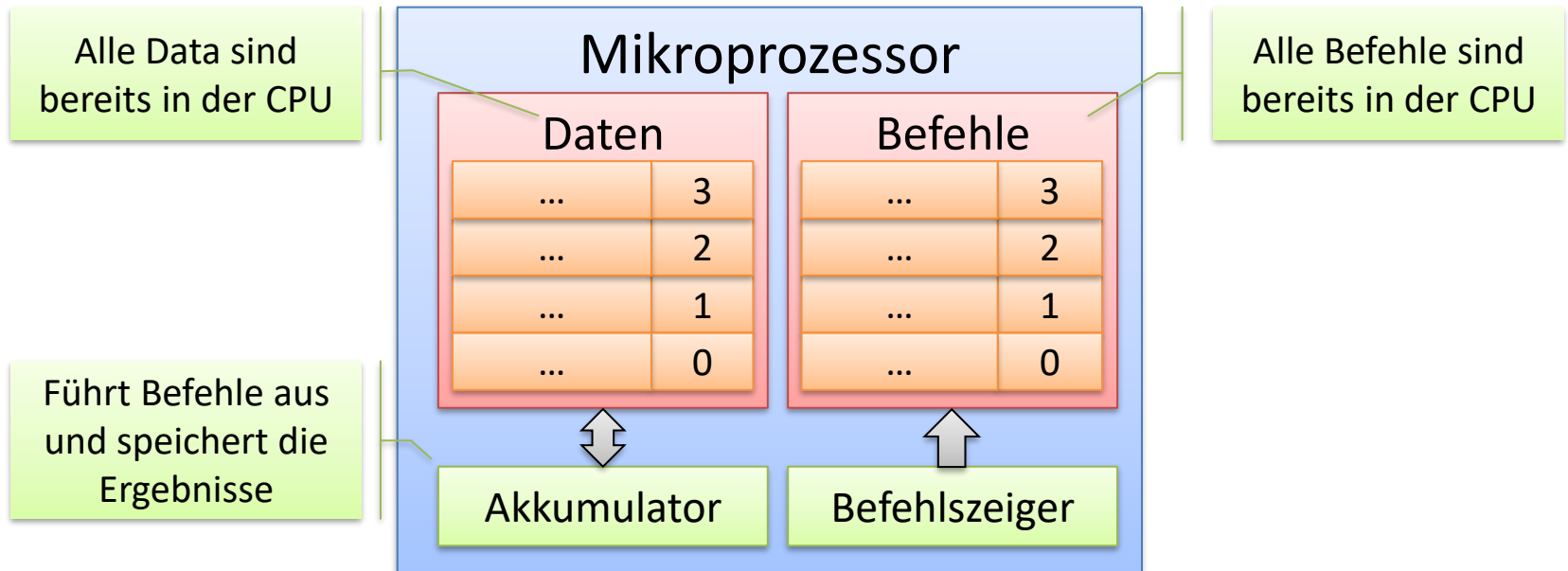
- Befehlspipeline

Befehl	Schritt						
1	IF	ID	OF	EX	WB		
2		IF	ID	OF	EX	WB	
3			IF	ID	OF	EX	WB
4				IF	ID	OF	EX
5					IF	ID	OF
Takt	1	2	3	4	5	6	7

Parallele Ausführung von fünf Befehlen

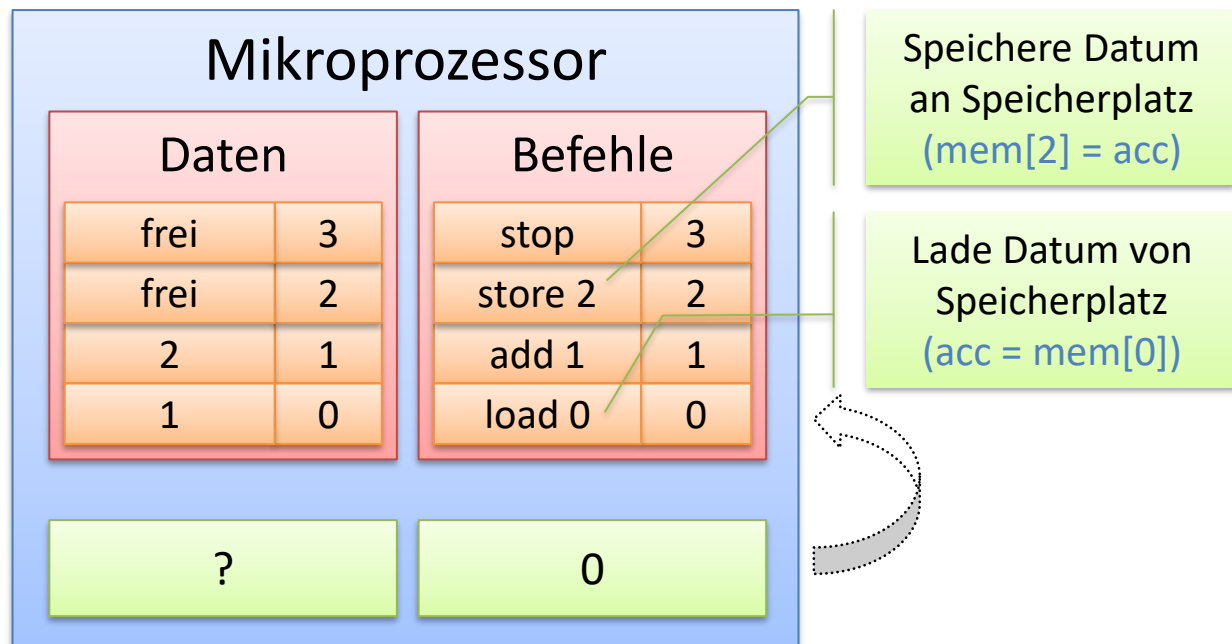
Registermaschinen (1)

- Theoretisches Model



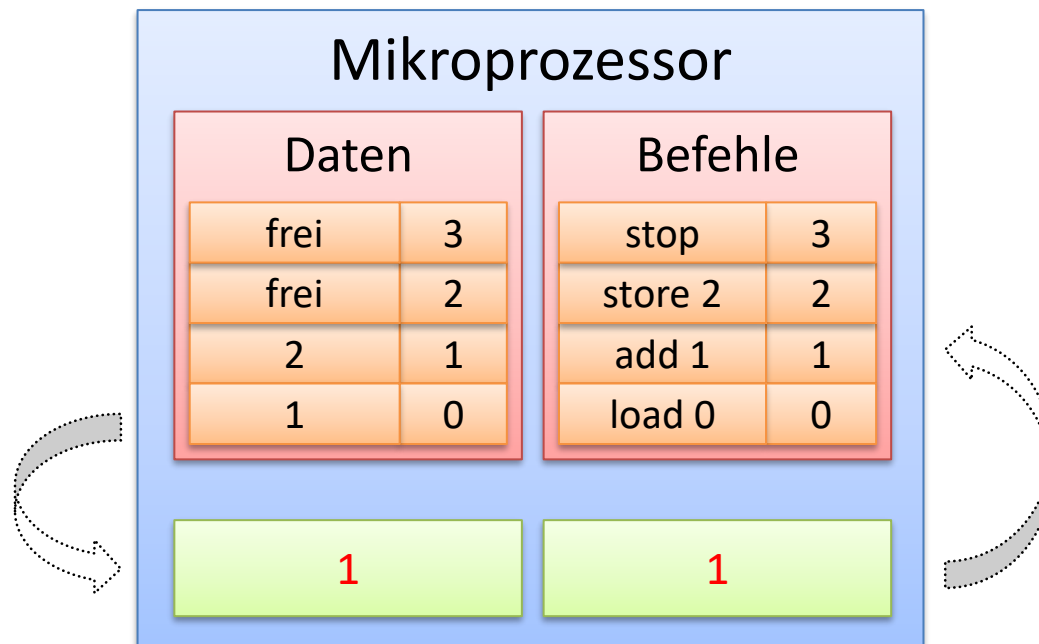
Registermaschinen (2)

- Programmausführung



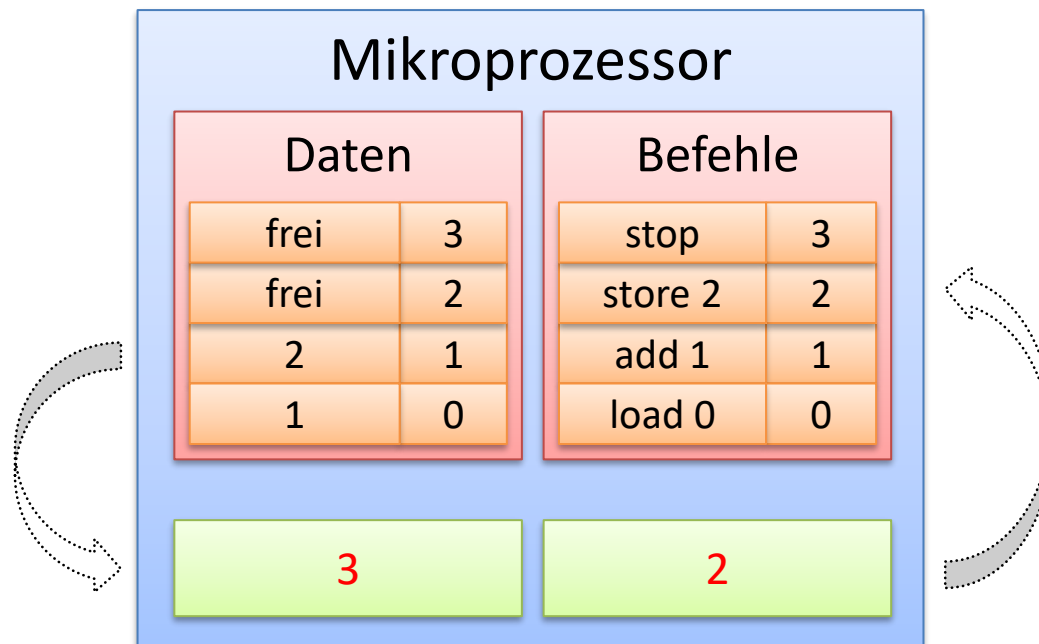
Registermaschinen (2)

- Programmausführung (Fortsetzung)



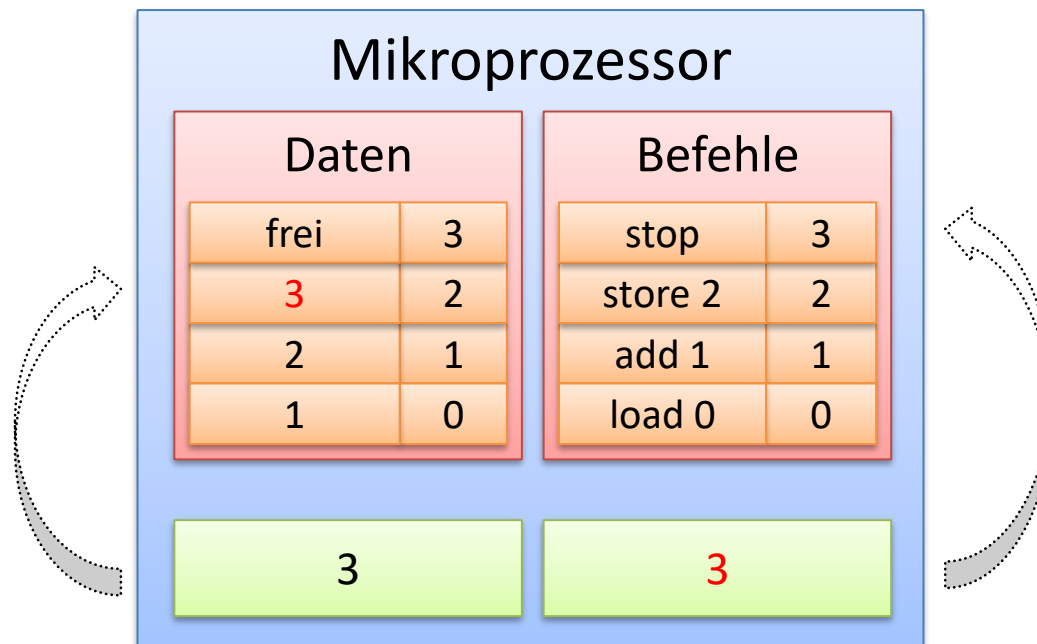
Registermaschinen (2)

- Programmausführung (Fortsetzung)



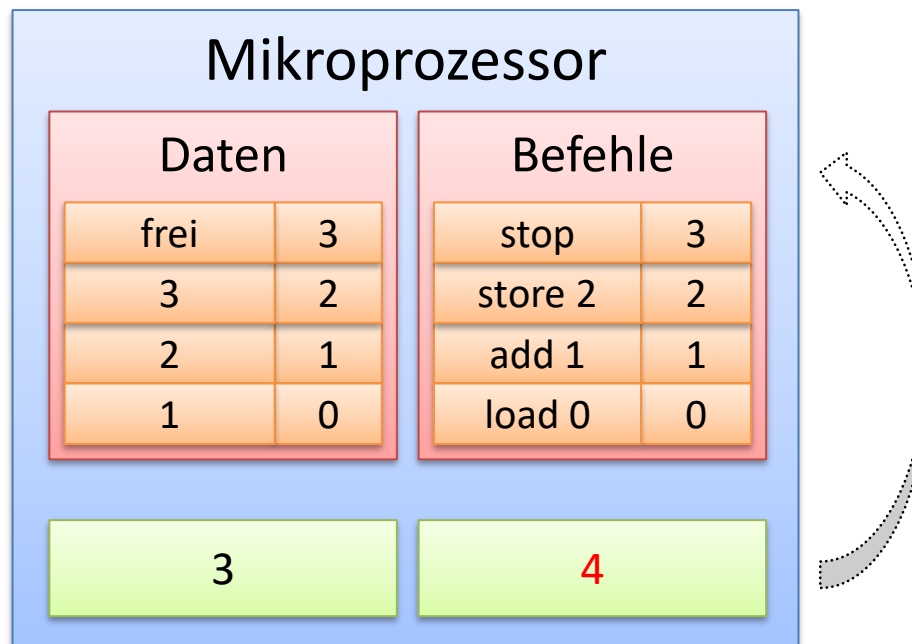
Registermaschinen (2)

- Programmausführung (Fortsetzung)



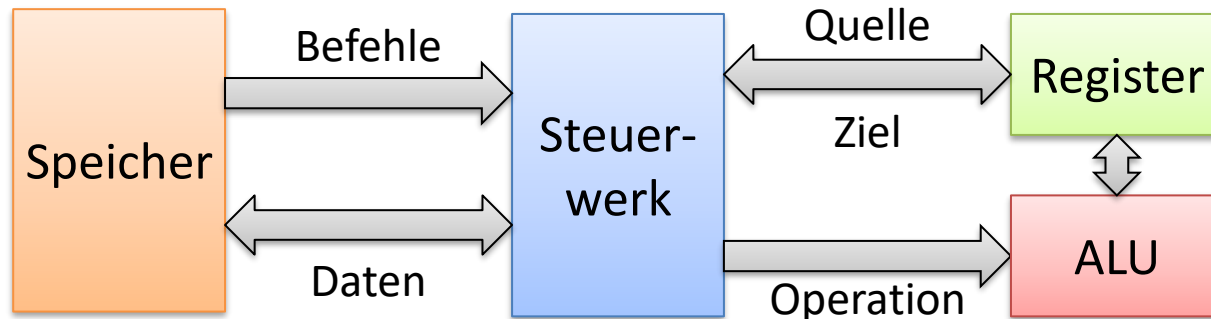
Registermaschinen (2)

- Programmausführung (Fortsetzung)



Befehlssatz (1)

- Befehle
 - Grundlegende Bauteile eines Programms
 - Befinden sich im Speicher
 - Geben Operation, Quelle und Ziel an

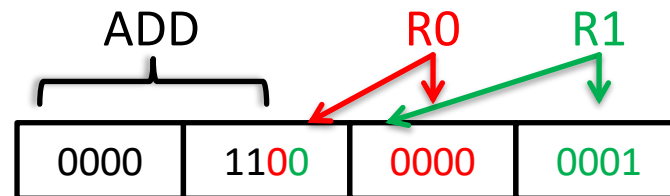


Befehlssatz (2)

- Maschinensprache
 - Spezifisch für jeden Mikroprozessor
 - Wird direkt vom Prozessors verstanden
 - Durch numerische Operationscodes dargestellt

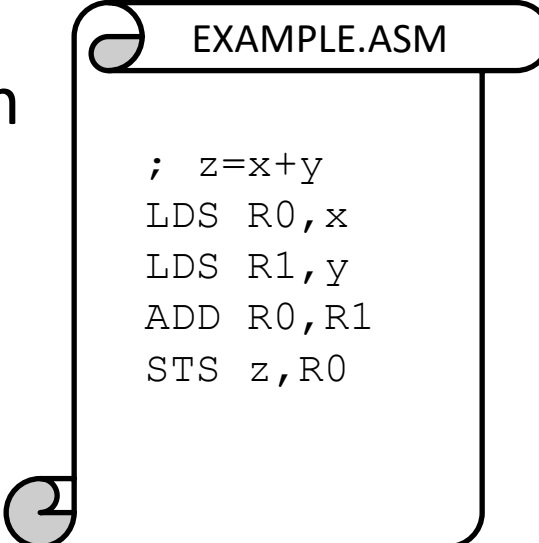
- Beispiel:

- Operation: ADD
- Quelle: R0, R1
- Ziel: R0
- Opcode: 0C01_{hex}



Befehlssatz (3)

- Assemblersprache
 - Opcodes werden durch Mnemonik ersetzt
 - Quelltext wird lesbar
 - Muss kompiliert werden
 - Vorteile:
 - Einfacher zu lesen
 - Kommentare erlaubt
 - Variablennamen
 - Sprungmarken



```
EXAMPLE.ASM  
  
; z=x+y  
LDS R0, x  
LDS R1, y  
ADD R0, R1  
STS z, R0
```

Befehlssatz (4)

- Befehlssatz

- Befehle für das Rechenwerk

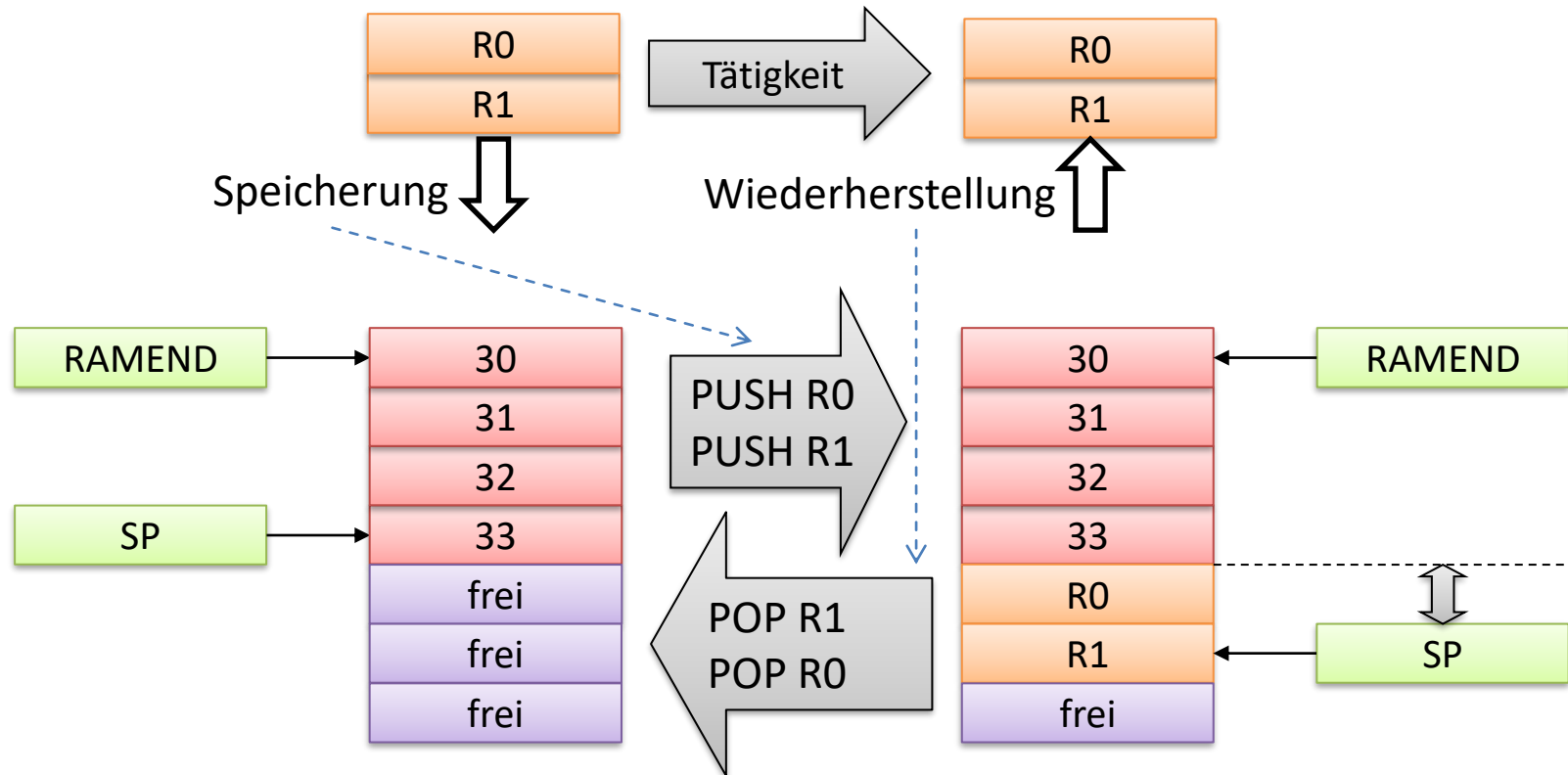
- Arithmetische Befehle : *ADD, SUB, MUL, CP, ...*
 - Logische Befehle : *AND, OR, EOR, COM, ...*
 - Befehle zum Schieben and Rotieren: *LSL, LSR, ROL, ...*
 - Befehle zur Bitmanipulation: *SBI, CBI, CLI, SEI, ...*

- Befehle für das Steuerwerk

- Datenbewegungsbefehle: *MOV, LD, ST, PUSH, ...*
 - Sprungbefehle: *RCALL, RET, BRcc, ...*

Stack (1)

- Funktionsweise



Stack (2)

- Zusammenfassung
 - Temporäres Speichern von Daten
 - LIFO-Speicher (Last in first out)
 - Wächst meist von oben nach unten
 - Der Stackzeiger zeigt auf das letzte Element
 - Nur zwei Grundfunktionen
 - PUSH: Verringert Stackzeiger, legt Element auf Stack
 - POP: Nimmt Element vom Stack, erhöht Stackzeiger
 - PUSH/POP: Predekrement / Postinkrement

