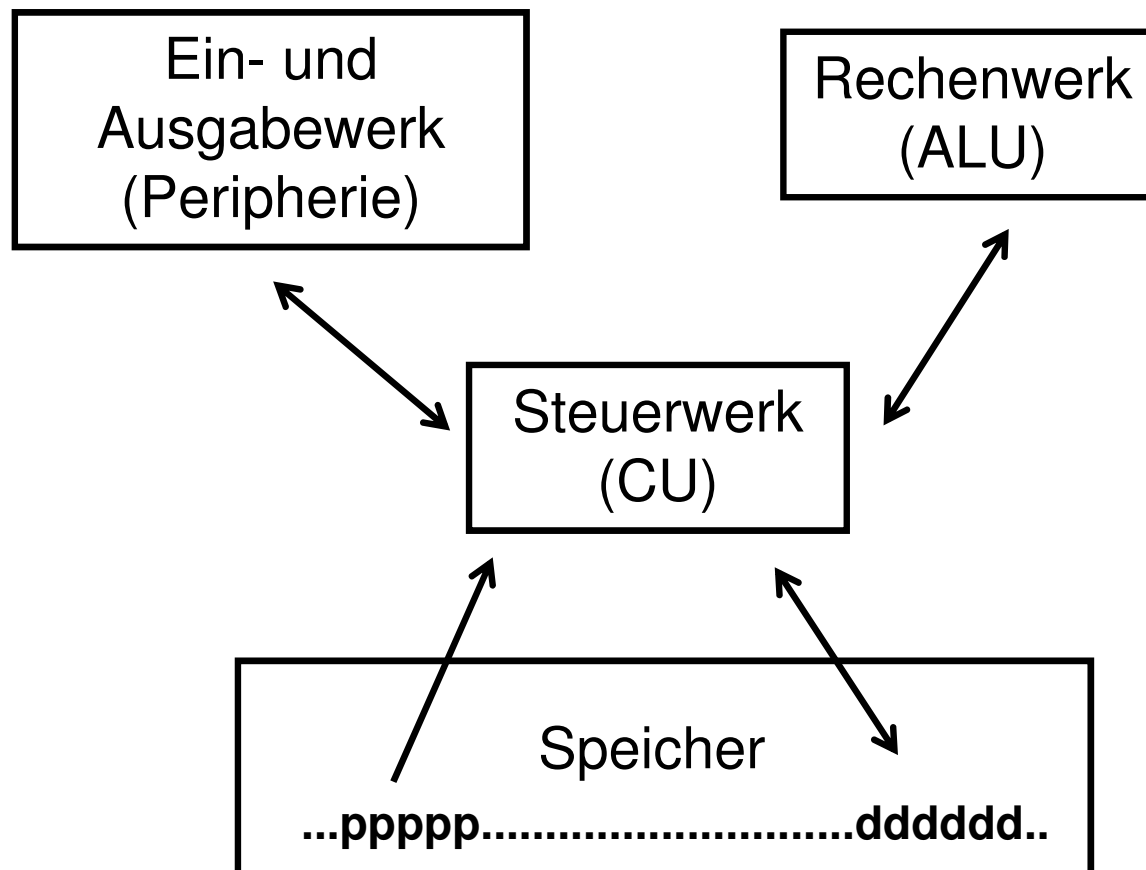




- Bisher wichtig: Konstruktionsprinzip des Rechenwerkes und Leitwerkes.
- Neu: Größerer Arbeitsspeicher





- Speicher beinhaltet sowohl die Daten zur Programmsteuerung (pppppp) als auch die Verarbeitungsdaten (dddddd).
- Der Speicher besteht aus einer geordneten Folge von Speicherplätzen, die mit Hilfe einer Nummer bzw. Adresse identifiziert werden können.
- Die Inhalte der Speicherplätze sind binär kodierte Daten. Sie können gelesen, aber auch überschrieben werden.
- Speicherzugriffe erfordern neben der Angabe einer Zugriffsfunktion (Lesen/Schreiben) unbedingt die Angabe einer Speicherplatzadresse.
- Das Rechenwerk (ALU=Arithmetic Logic Unit) kann durch das Steuerwerk bereitgestellte Daten bearbeiten, z.B. Zahlen addieren.
- Mit Hilfe des Ein- und Ausgabewerkes kann der Computer Daten von der Außenwelt anfordern bzw. an diese ausgeben.
- Das Steuerwerk (CU=Control Unit) hat die Aufgabe, die automatische Programmabarbeitung zu organisieren.

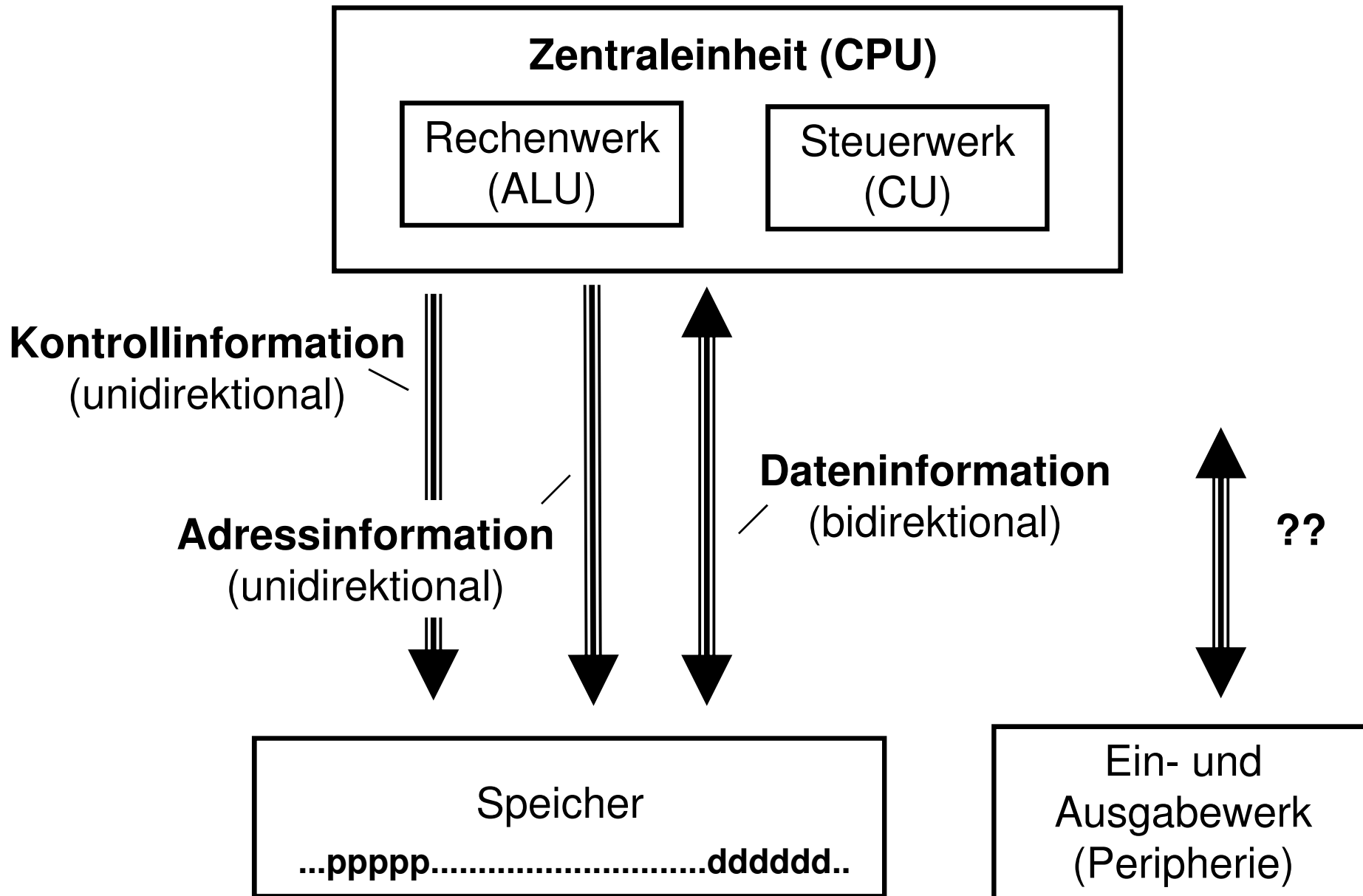


- Programm besteht aus sinnvoller zeitlichen Abfolge elementarer Anweisungen an das Steuerwerk, sog. **Maschinenbefehle**.
- Voraussetzung bei Programmstart: binär kodierten Maschinenbefehle des Programms und zugehörigen Verarbeitungsdaten sind im Speicher.
- Der Bediener stellt die Adresse des ersten zu bearbeitenden Befehles am Steuerwerk ein und startet die Abarbeitung.
- Das Steuerwerk muss dann zyklisch:
 - den aktuellen Maschinenbefehl im Speicher lesen
 - den Befehl interpretieren
 - ggf. Operanden im Speicher lesen
 - die Bearbeitung durch Rechen- oder Ein-/Ausgabewerk anstoßen
 - Adresse des vorgesehenen Nachfolgebefehls bestimmen.
- Ein spezieller HALT-Befehl kann zum Beenden des Zyklus führen.



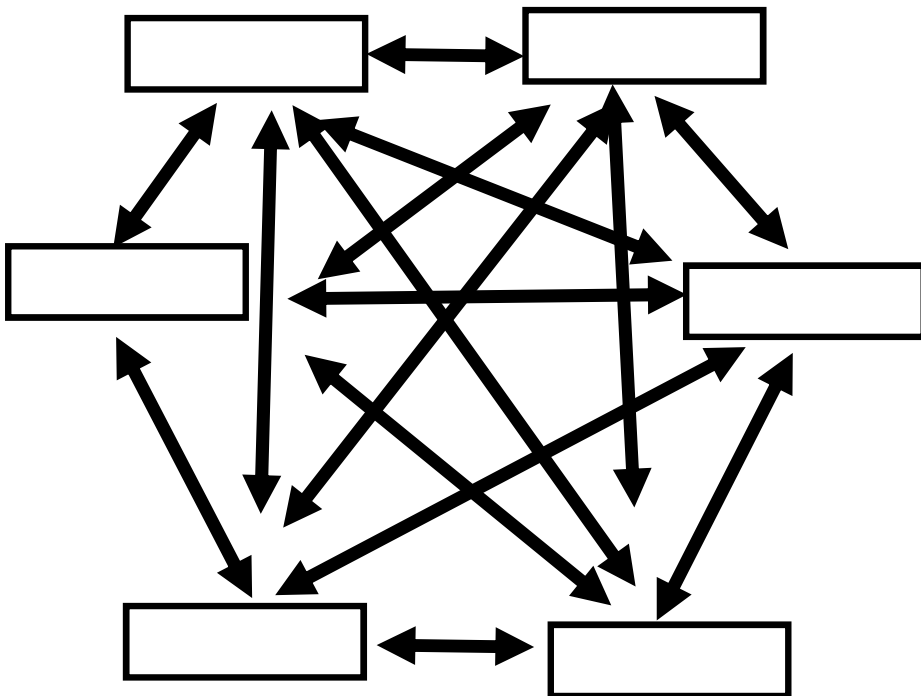
- Die Menge aller Maschinenbefehle eines Rechners wird als **Befehlssatz** bezeichnet.
- Befehle können in den folgenden Klassen zusammengefasst werden:
 - Arithmetische Befehle (z.B. Addition)
 - Logische Befehle (z.B. UND)
 - Bitmanipulationsbefehle (z.B. Shift)
 - Sprung- und Testbefehle (z.B. JMP, BRZ)
 - Transportbefehle (z.B. LDA)
 - Steuerbefehle (z.B. HALT, NOP)

Kommunikation zwischen CPU und Speicher

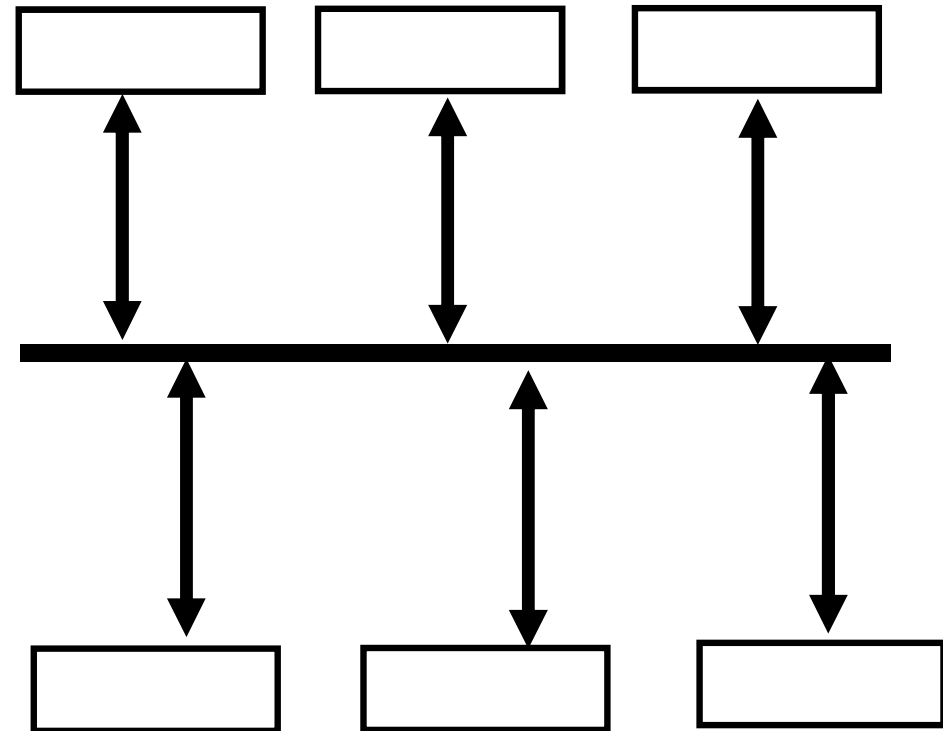




Direkte Kommunikation:

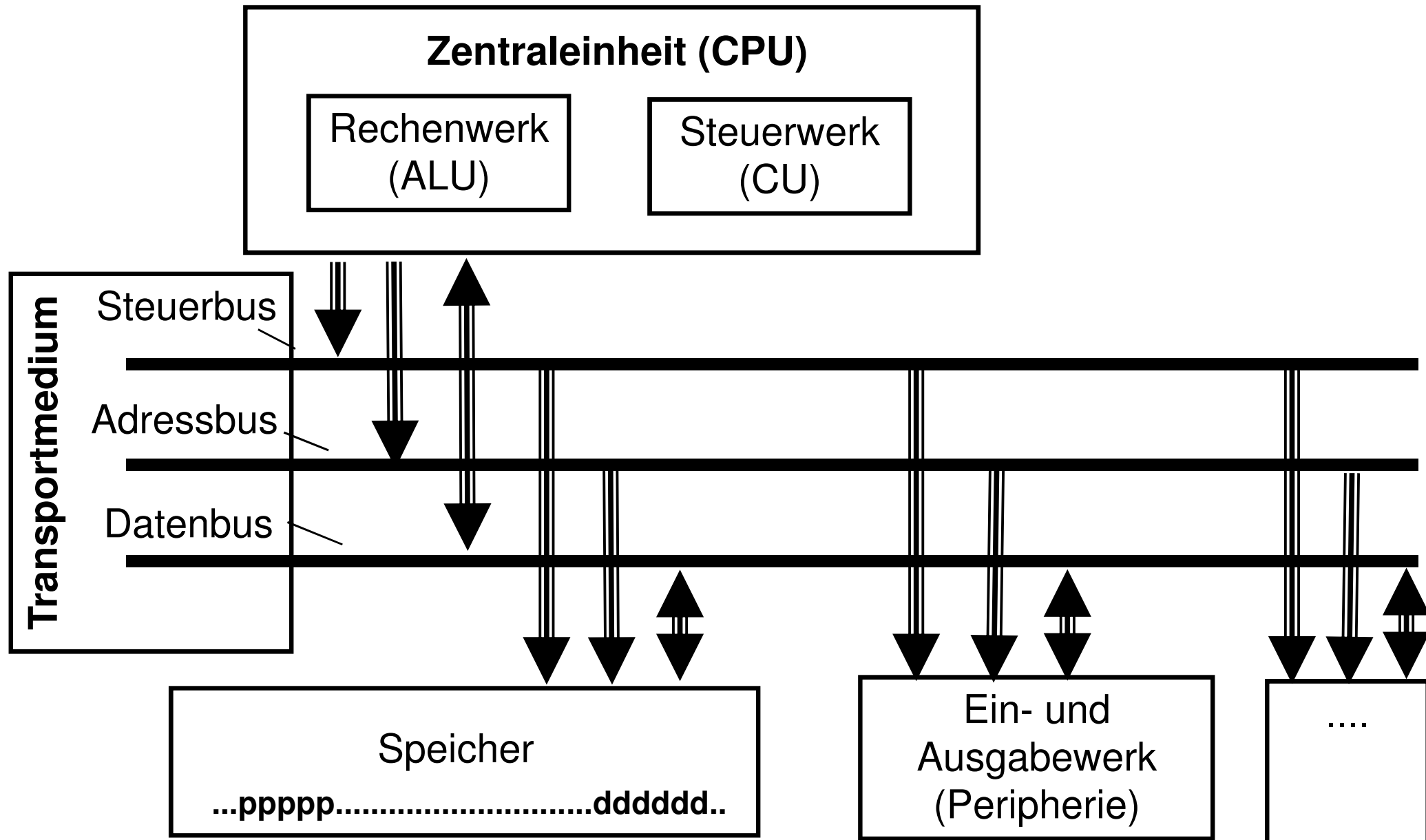


Indirekte Kommunikation:

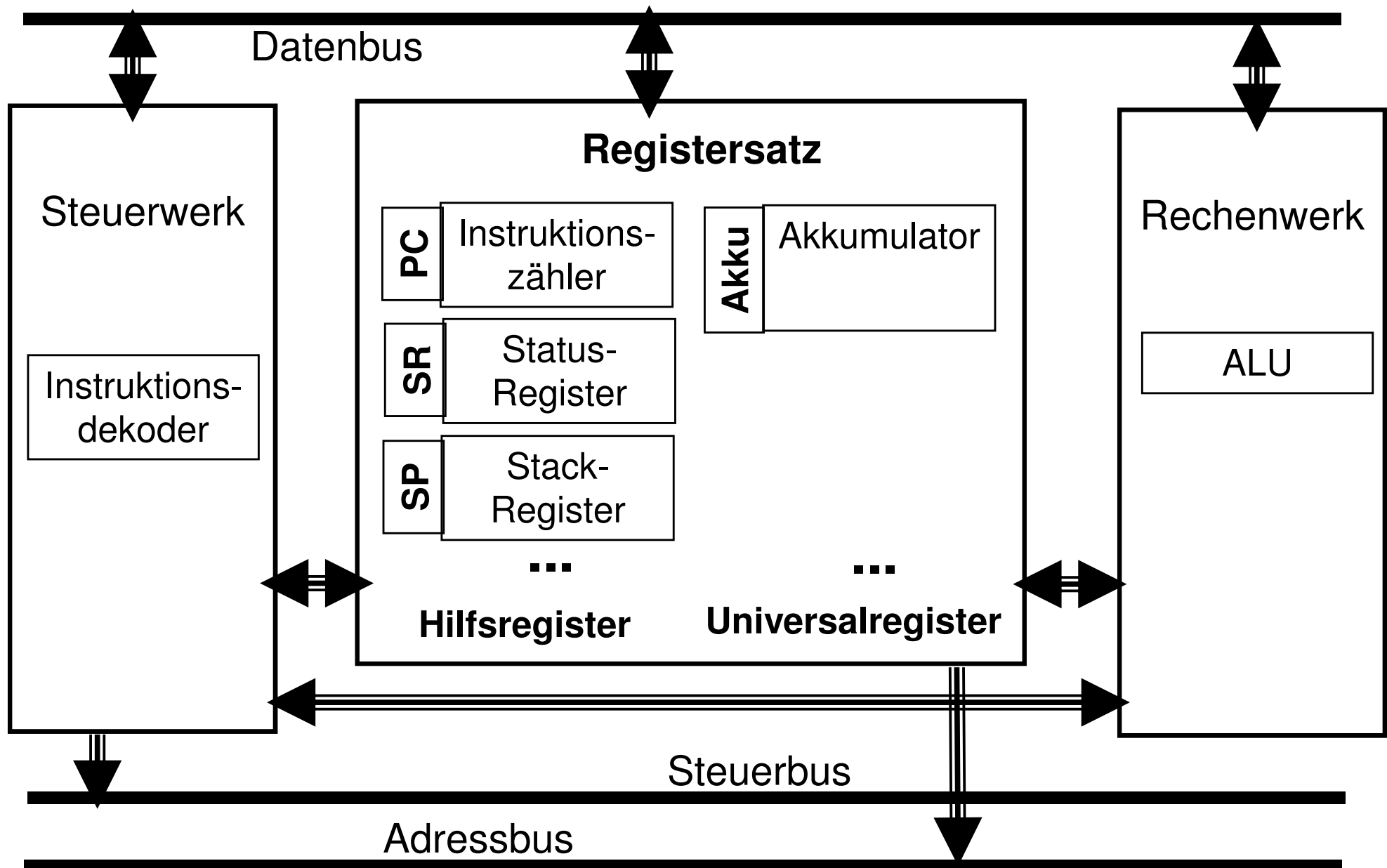




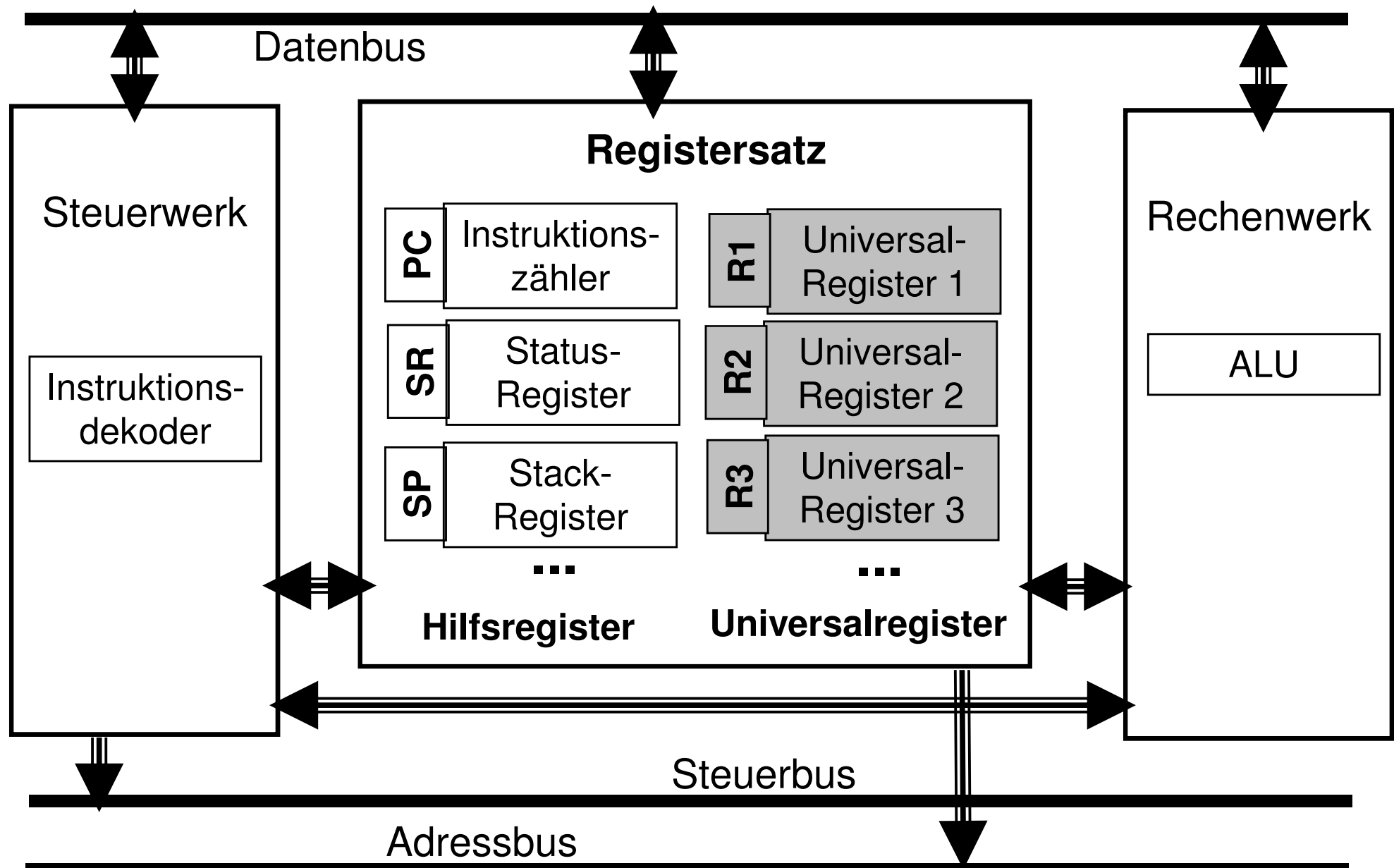
- Datenaustausch zwischen Digitalen Baugruppen und Bausteinen
- Unidirektionaler Bus in einer Richtung
- Bidirektionaler Bus in beiden Richtungen
- Parallele Bussysteme (8, 16, 32, 64-Bit) bestehen aus mehreren Sammelleitungen
- Serielle Bussysteme
(Sender legt einzelne Bits nacheinander auf die Leitung)
- Maximal eine Sender, n Empfänger



Aufbau der CPU



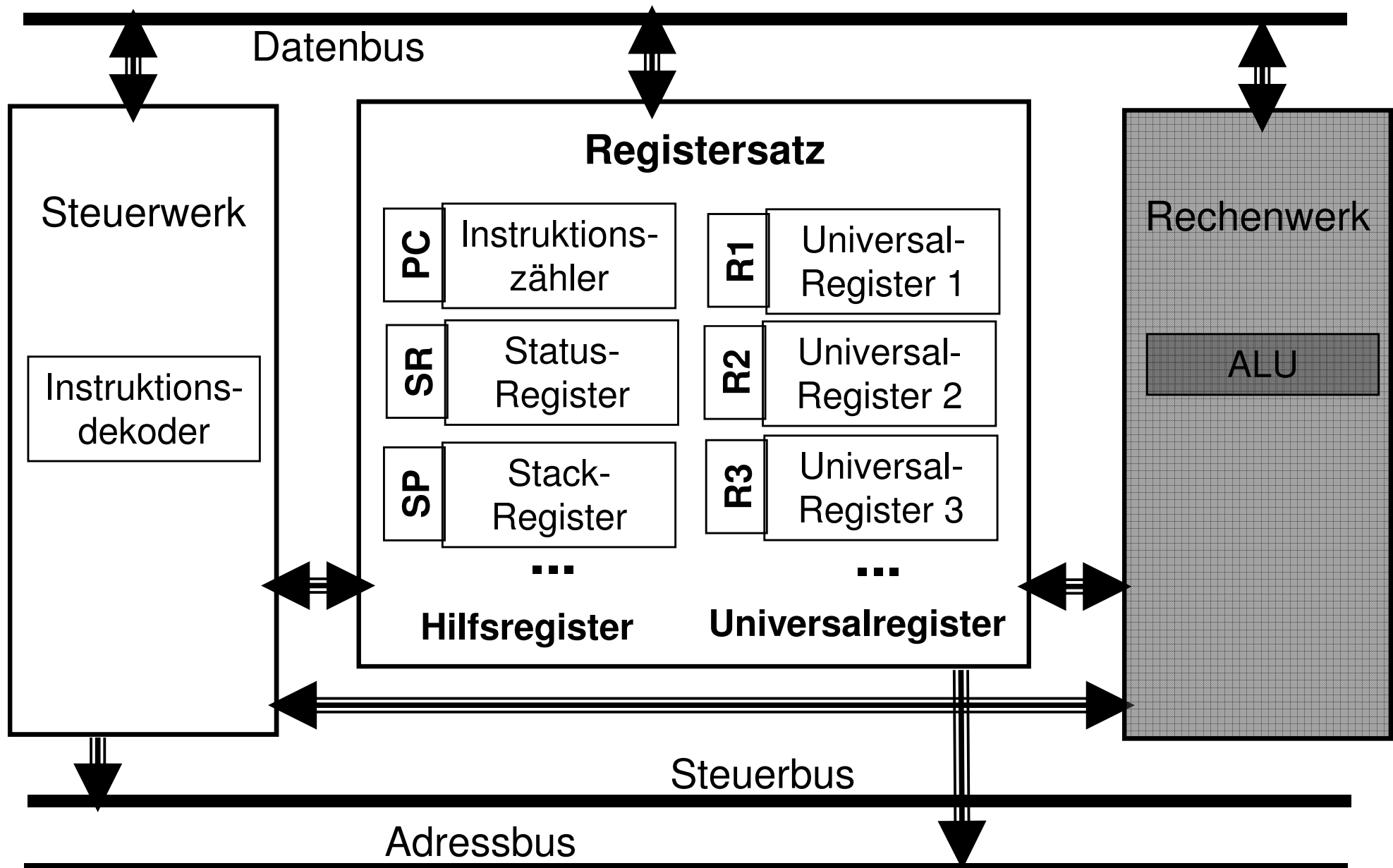
Aufbau der CPU



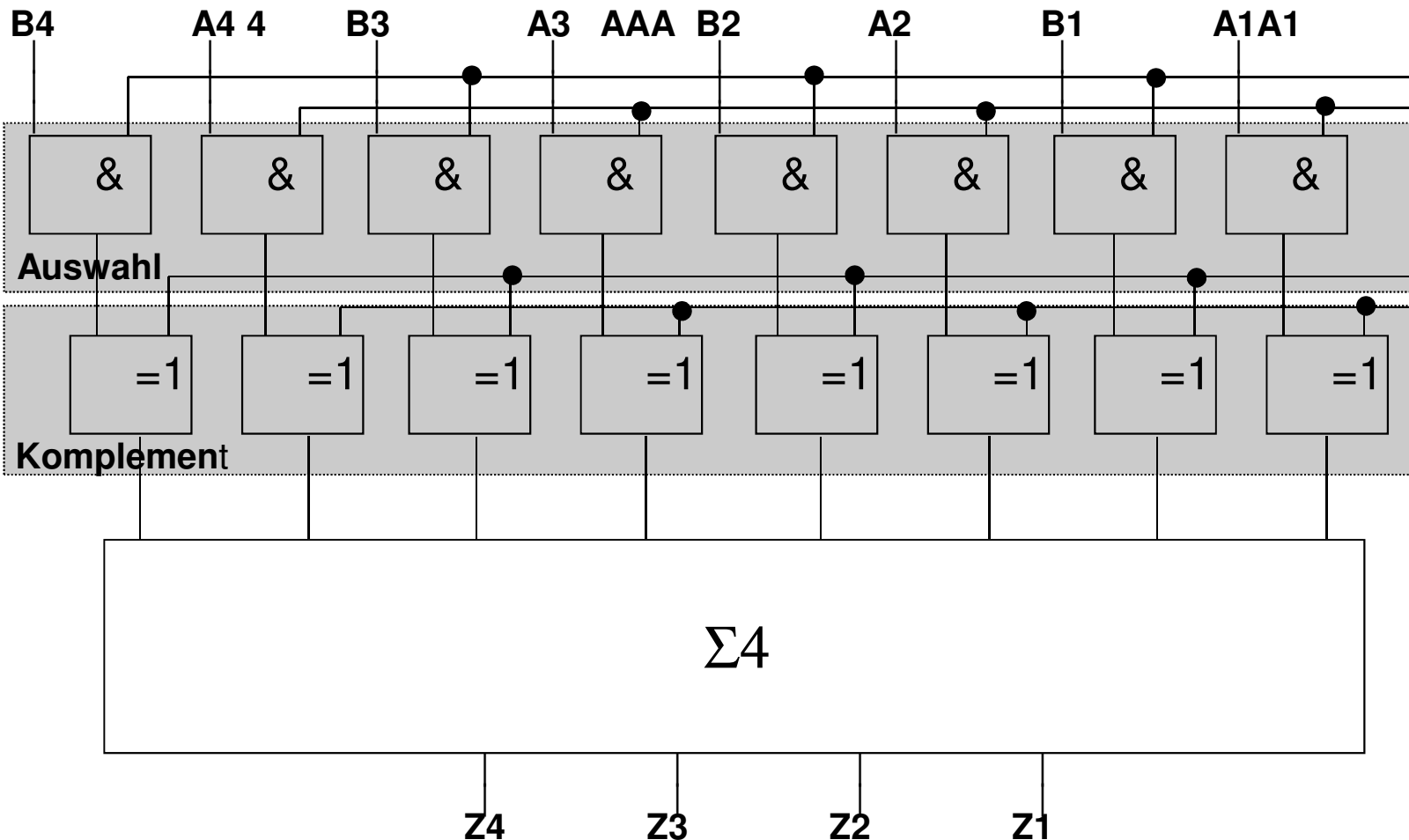


- Instruktionszähler (PC-Register):
 - Speicheradresse zum nächsten ausführbaren Befehl.
 - Zum Laden eines Befehls wird Registerinhalt zur Adressierung verwendet und anschließend um eins erhöht.
 - Sprung an eine feste Speicherstelle wird durch einfaches Überschreiben des PC-Registers erreicht.
- Statusregister (SR):
 - Wird vom Rechenwerk beschrieben
 - z.B. Carry-Bit, Negativ-Bit, Zero-Bit
- Stapelregister (Stack Pointer (SP)):
 - Zur Verwaltung von Unterprogrammen

Aufbau der CPU

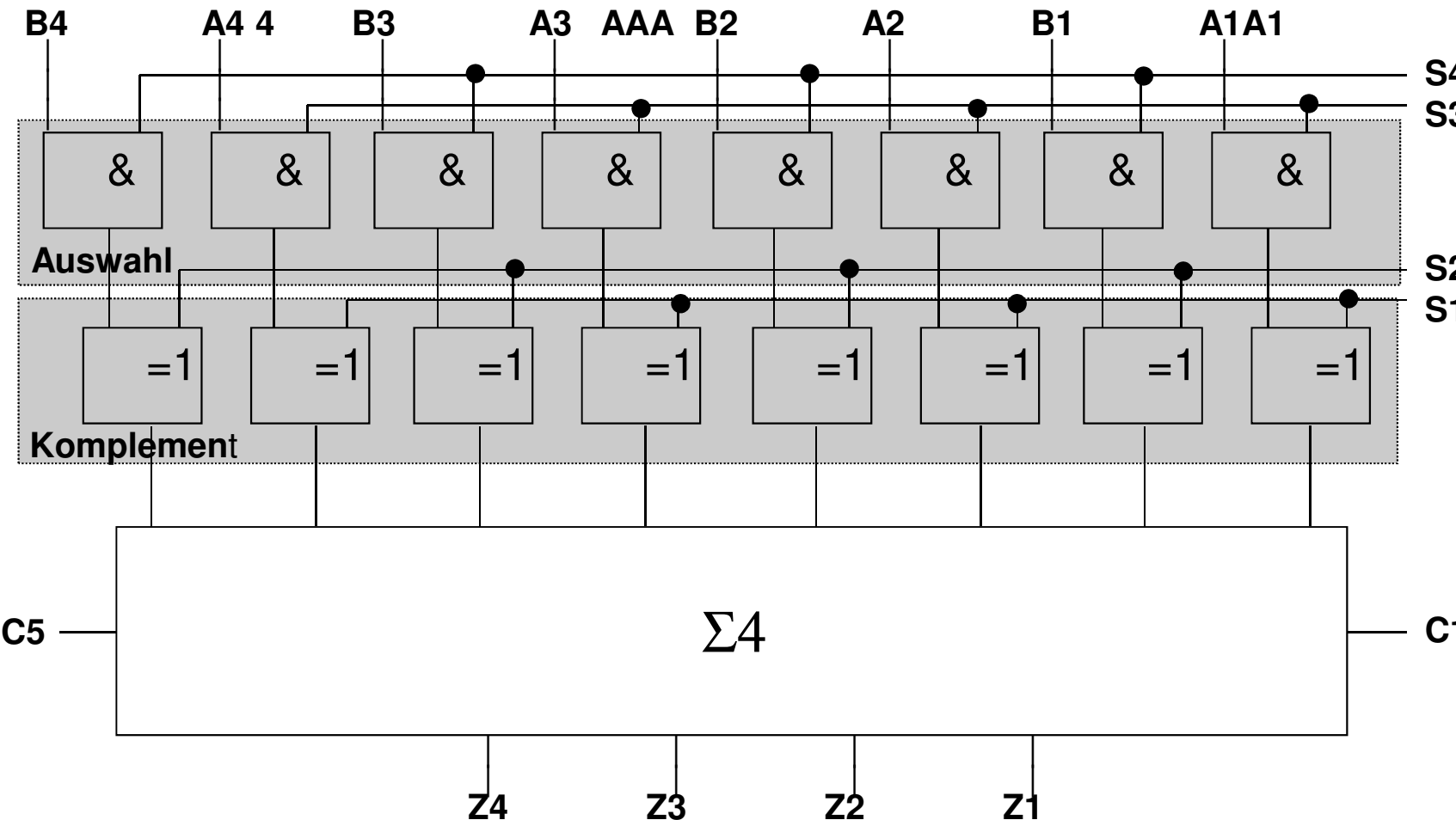


Addierer-Subtrahierer



	S4	S3	S2	S1	Z
0	0	0	0	0	0
0	0	0	1	0	-1
0	0	1	0	0	-1
0	0	1	1	0	-2
0	1	0	0	0	+A
0	1	0	1	0	-A-1
0	1	1	0	0	+A-1
0	1	1	1	0	-A-2
1	0	0	0	0	+B
1	0	0	1	0	+B-1
1	0	1	0	0	-B-1
1	0	1	1	0	-B-2
1	1	0	0	0	+A+B
1	1	0	1	0	+B-A-1
1	1	1	0	0	+A-B-1
1	1	1	1	0	-A-B-2

Addierer-Subtrahierer



	S4	S3	S2	S1	Z
	0	0	0	0	C1
	0	0	0	1	C1-1
	0	0	1	0	C1-1
	0	0	1	1	C1-2
	0	1	0	0	C1+A
	0	1	0	1	C1-A-1
	0	1	1	0	C1+A-1
	0	1	1	1	C1-A-2
	1	0	0	0	C1+B
	1	0	0	1	C1+B-1
	1	0	1	0	C1-B-1
	1	0	1	1	C1-B-2
	1	1	0	0	C1+A+B
	1	1	0	1	C1+B-A-1
	1	1	1	0	C1+A-B-1
	1	1	1	1	C1-A-B-2



Wahlweise:

- Addierer-Subtrahierer
- Und-Verknüpfung
- Oder-Verknüpfung
- Exklusiv-Oder-Verknüpfung

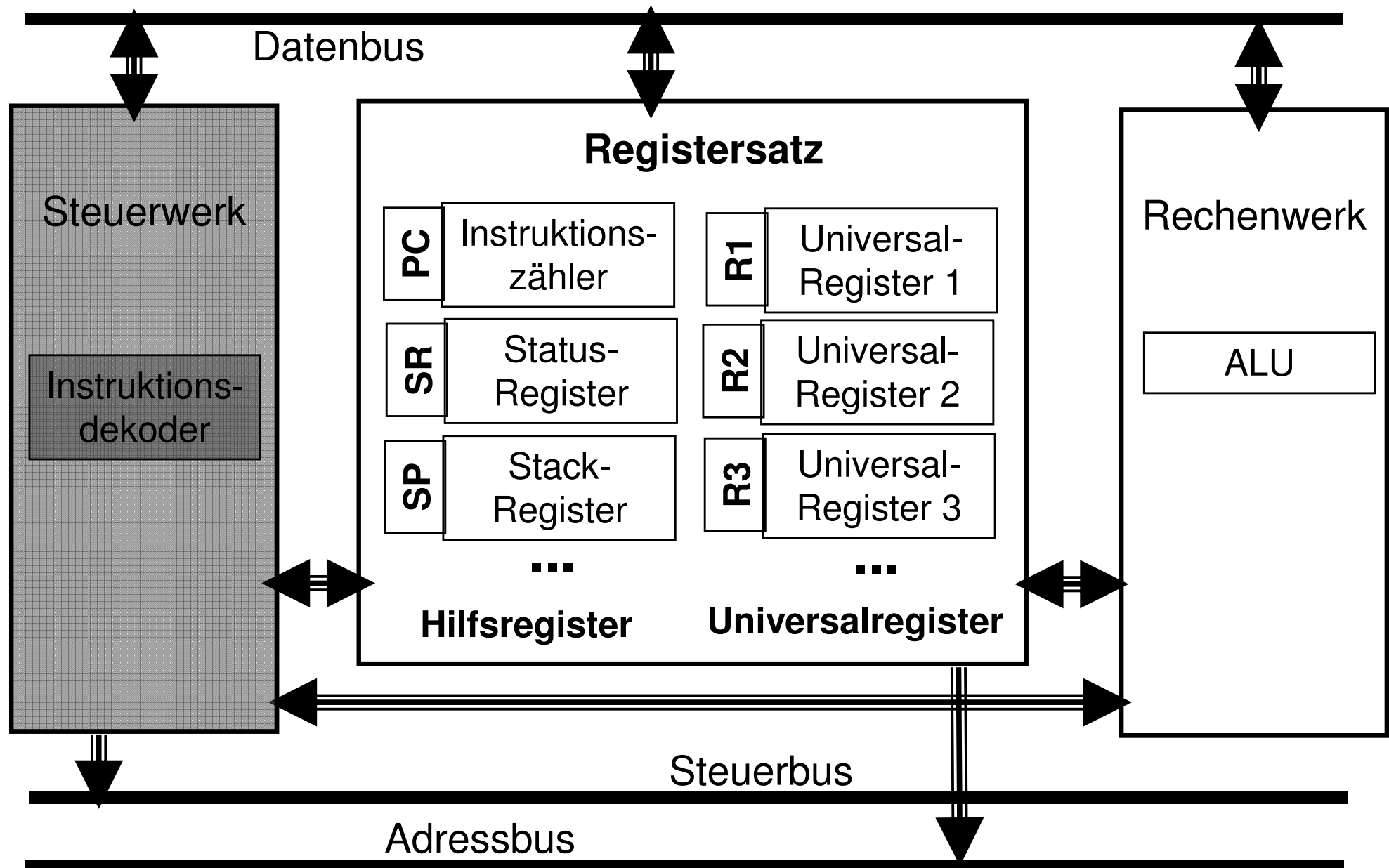
2 zusätzliche Steuerbits zur Umschaltung der 4 Modi

Insgesamt gibt es damit (inklusive C1) 128 „Befehle“

Benötigt werden (ohne C1) nur 13 Befehle, deshalb:

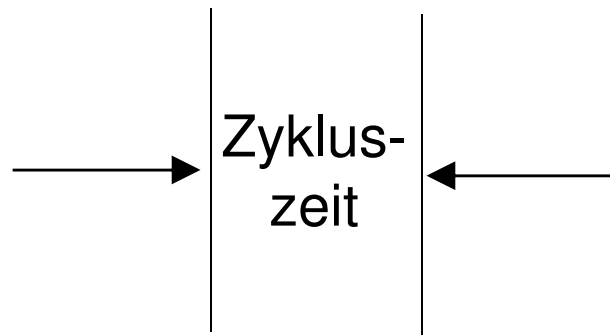
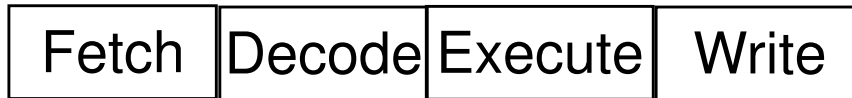
Umcodierung mit 4-Bit-Code

Aufbau der CPU





1. Befehl:



2. Befehl:



- Befehlsholphase (Fetch)
- Decodierphase (Decode)
- Ausführungsphase (Execute)
- Speicherphase (Write)

Befehlsliste des Modellprozessors



Nr.	Befehl	Code	Beschreibung
0	NOP	0000	Wartezyklus (no Operation)
1	LDA #n	0001	Lädt den Akkumulator mit dem Wert n
2	LDA (n)	0010	Lädt den Akkumulator mit dem Inhalt der Speicherstelle n
3	STA n	0011	Lädt den Akkumulatorinhalt in die Speicherstelle n
4	ADD #n	0100	Erhöht den Akkumulator um den Wert n
5	ADD (n)	0101	Erhöht den Akkumulator um den Inhalt der Speicherstelle n
6	SUB #n	0110	Erniedrigt den Akkumulator um den Wert n
7	SUB (n)	0111	Erniedrigt den Akkumulator um den Inhalt der Speicherstelle n
8	JMP n	1000	Lädt den Instruktionszähler mit dem Wert n
9	BRZ #n	1001	Addiere n auf den Instr.-Zähler, falls das Zero-Bit gesetzt ist
10	BRC #n	1010	Addiere n auf den Instr.-Zähler, falls das Carry-Bit gesetzt ist
11	BRN #n	1011	Addiere n auf den Instr.-Zähler, falls das Negations-Bit gesetzt ist



- 4-Bit Prozessor (Registerbreite 4 Bit)
- 4-Bit Adressbus (16 Speicherplätze)
- 4(8)-Bit Datenbus
- 4-Bit Instruktionsformat
- 4-Bit Akkumulator: Setzen /Erhalten/Akkumulieren (Addition/Subtraktion)
- 4 Bit Programmzähler Laden/ Inkrementieren / Akkumulieren
- keine Universalregister, kein Stack-Pointer
- 3-Bit Statusregister Zero, Carry, Negative
- 1 Befehl pro Taktzyklus