

- im Altertum: ABAKUS als Rechenhilfe
- mechanische Additionsmaschinen (vor etwa 300 Jahren):
- Schickard'sche Rechenuhr 4 Grundrechenarten (Prof. für biblische Sprachen in Tübingen, 1623)
- PASCAL (1623-1662) Radgetriebe

Programmsteuerung seit etwa 100 Jahren:

- England: Babbage (1792-1871): Pläne für Rechenwerk, Steuerwerk, Speicher
- Hollerith (1900): Lochkarten; 1928 erste funktionsfähige Maschinen

Elektrische Rechner: Erster Relais-Rechner mit Festprogrammierung

Konrad Zuse (Bauingenieur) 1935 †1995



Mathematische Modellierung kontinuierlicher Signale:

Laplace

} Transformation: Veröffentlicht im 19. Jahrhundert

Fourier

Jean Baptiste Joseph, Baron de **Fourier** war Gouverneur in Unterägypten unter Napoleon

1801 Rückkehr nach Frankreich

1822 umfangreiches Werk über Wärmefluß an Kanonenrohren

→ *Theorie der Fourierreihe*

→ später: Diskrete FT (DFT)

Pierre Simon, Marquis de **Laplace** war theoretischer Astronom

um 1800: Veröffentlichung der *Laplacetransformation*

später: Diskrete LT (z-Transformation)

1965 Cooley & Tukey: **FFT-Algorithmus** *Fast Fourier Transform* $\equiv$ schnelle DFT

(Ursprünge finden sich bei den Mathematikern Runge und Gauß $\approx$ 1900)

1980 – 1982 => die ersten „Digitalen Signalprozessoren“

AMI S2811

Intel 2920

NEC μ PD7720

TI TMS32010

Heute:

Motorola

▪ DSP 56300 24bit-Festkomma-DSP (Motorola) 100MHz $\Leftrightarrow$ 100MIPS

▪ MSC8101 **StarCore SC140** 1200 **True** DSP MIPS oder 3000 RISC MIPS bei 300 MHz Taktfrequenz

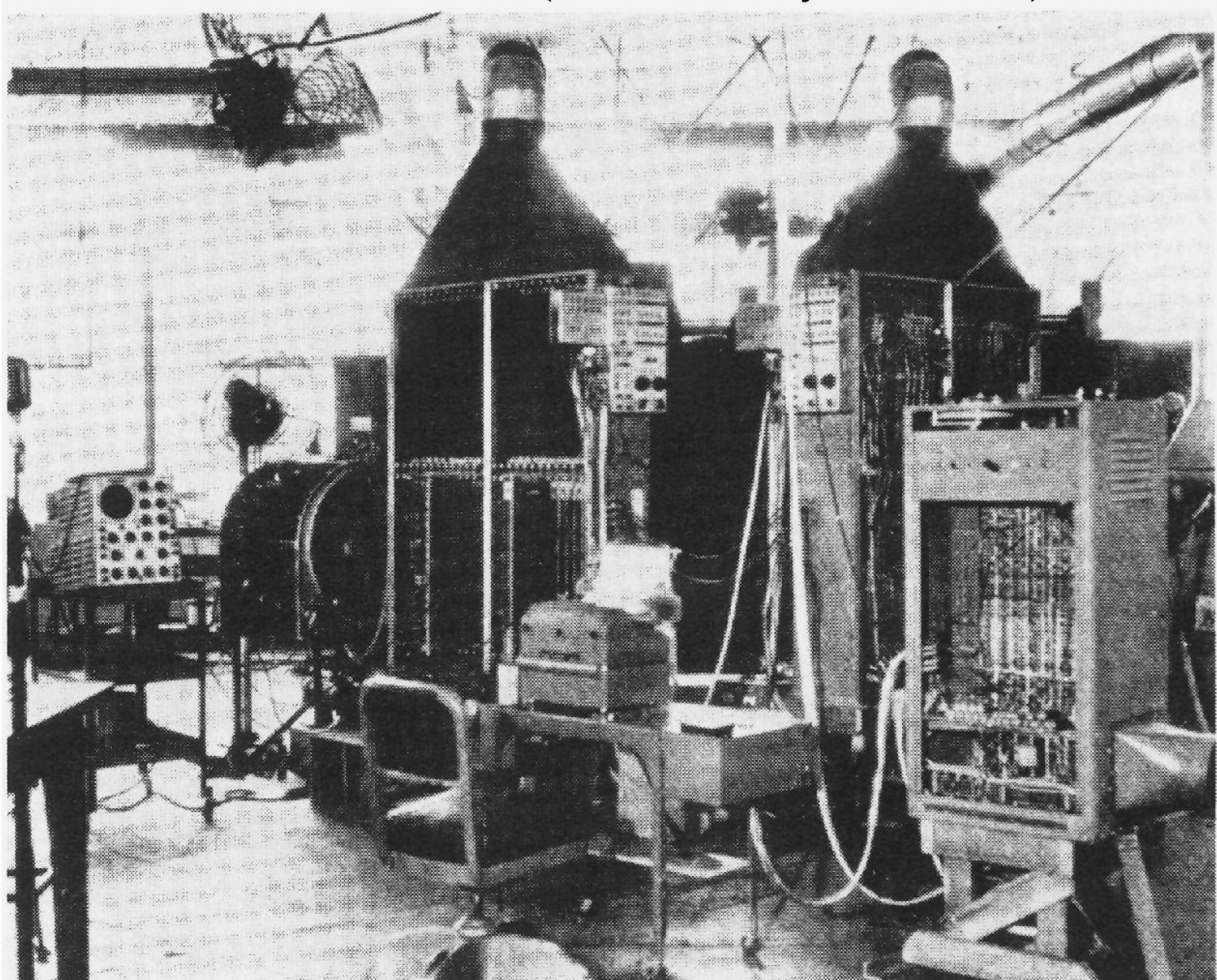
Texas Instruments

▪ TMS 320C6201 „16bit“-Festkomma-DSP 200MHz $\Leftrightarrow$ 1600MIPS

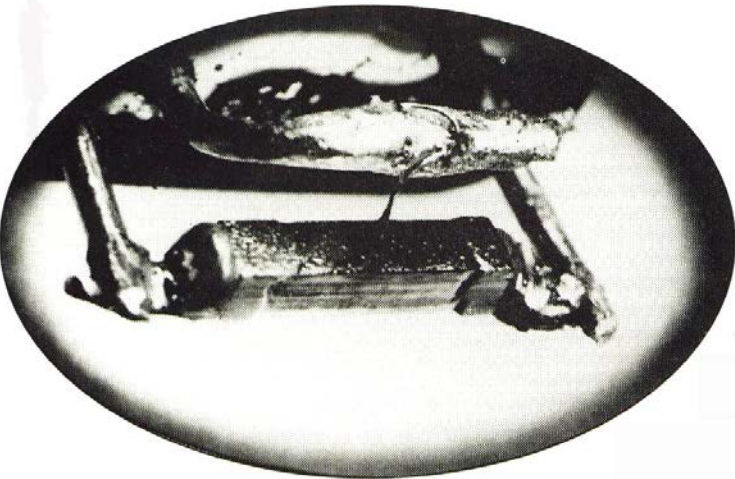
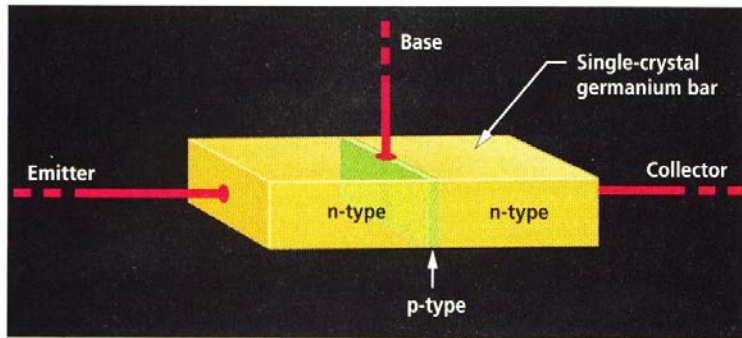
▪ TMS320DM642 – max. 600 MHz Taktfrequenz $\Leftrightarrow$ 4800 MIPS

Elektronische Rechner:

Erster Röhrenrechner: ENIAC (Univ. of Pennsylvania, 1946) fürs Militär



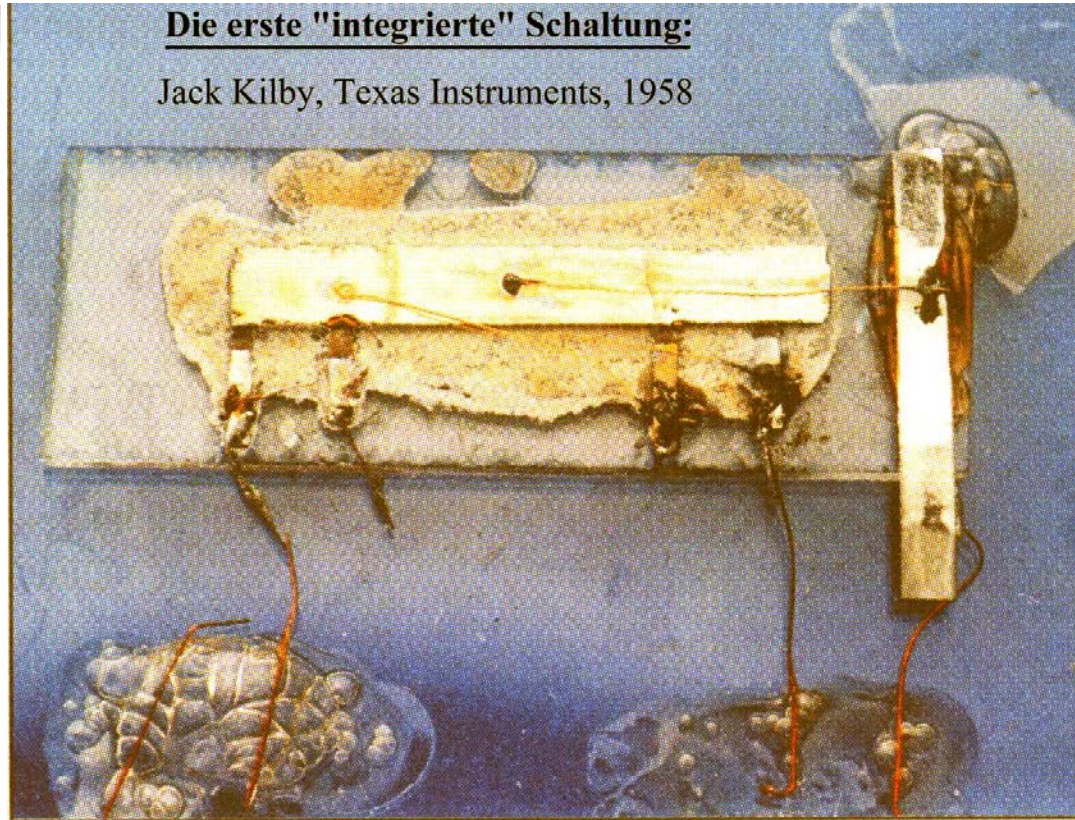
Erster Transistor: Shockley, Bardeen, Brattain (1948)



Der Beginn der Integrationstechnik

Die erste "integrierte" Schaltung:

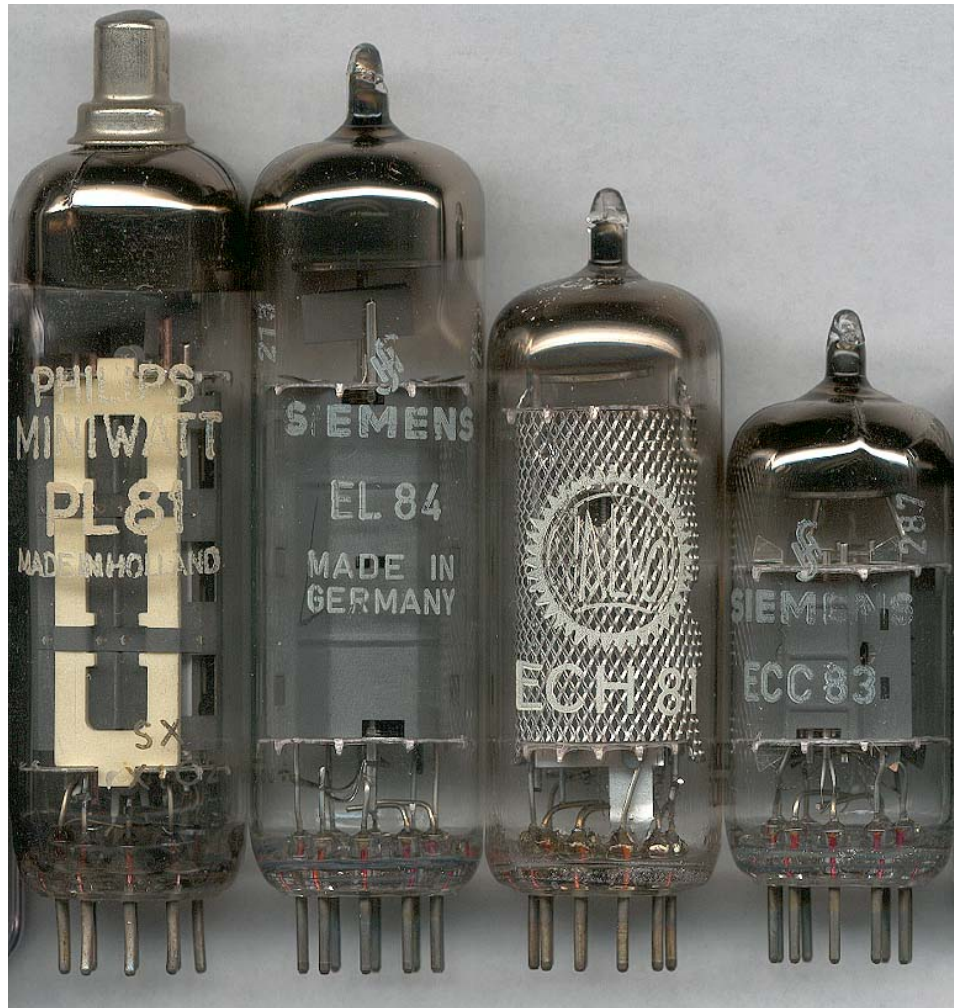
Jack Kilby, Texas Instruments, 1958



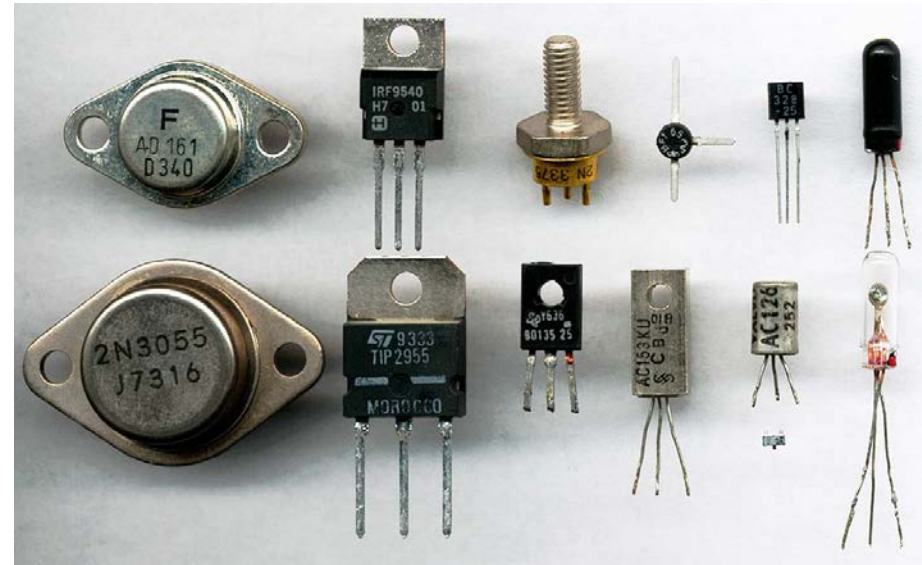
Meilensteine der Integrationstechnik:

- 1958 Jack Kilby (Texas Instruments): Multivibrator mit 4 Transistoren
- 1993 über 3 Mio. Transistoren auf einem Chip (Intel Pentium-Prozessor)
- 2004 AMD Athlon: 40 Mio. Transistoren, Intel Pentium 4: ca. 50 Mio. Transistoren

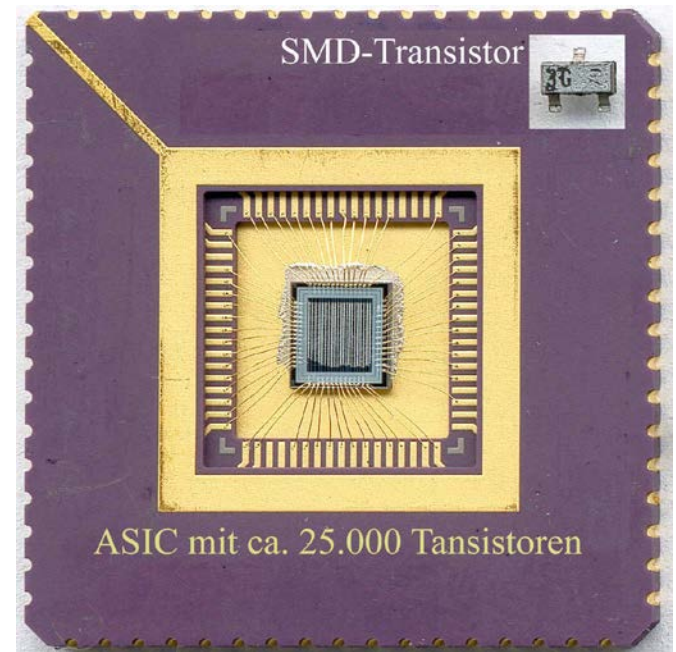
Elektronenröhren



einzelne Transistoren



Anwendungsspezifische integrierte Schaltung



Analogrechner

- Es werden Schaltungen aufgebaut, bei denen Spannungen, Ströme über Differentialgleichungssysteme verknüpft sind.
- Man hat gleiche Gesetzmäßigkeiten wie in vielen anderen physikalischen Systemen, z.B. mechanischen.
- Die Simulationsmodelle sind anschaulich und der Denkweise des Ingenieurs angepaßt.
- Die Programmierung eines zu lösenden Problems erfolgt über einfache lineare Transformationen, d.h. es müssen im wesentlichen Maßstabsfaktoren bestimmt und eingestellt werden.
- Die Ausgabe von Ergebnissen erfolgt am Oszillographen, über Plotter oder X-Y-Schreiber.

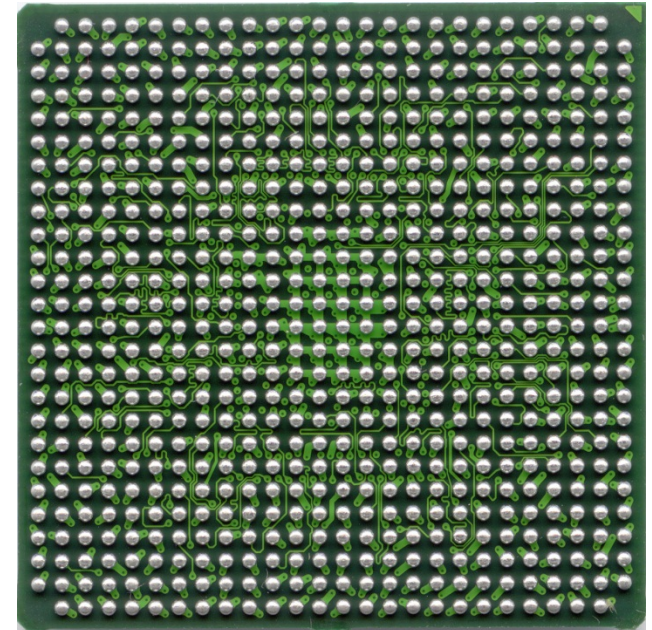
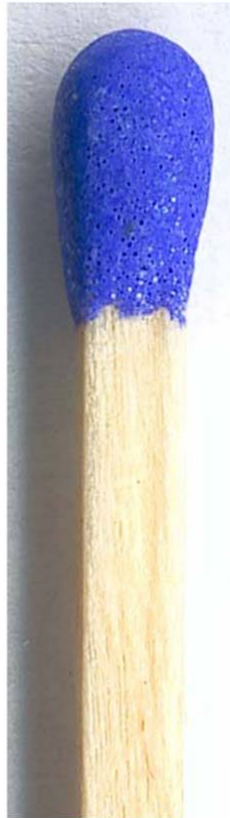
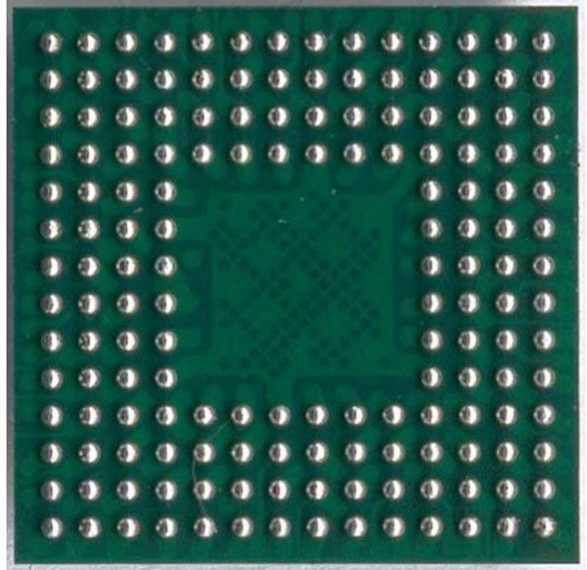
Digitalrechner

- dimensionslose Zahlen mit vielen Stellen als Rechengrößen
- Wertebereiche der physikalischen Größen müssen passend transformiert werden
- die Genauigkeit ist durch Rechenzeit praktisch beliebig steigerbar
- Informationen können gespeichert, sortiert und logisch verknüpft werden
- kein Genauigkeitsverlust bei Datenübertragung
- Programmierung ist aufwendig und es werden gute Algorithmen benötigt
- in der Regel ein einziges Rechenwerk, das seriell arbeitet, verfügbar
- Simulation und Regelung in Echtzeit sind problematisch

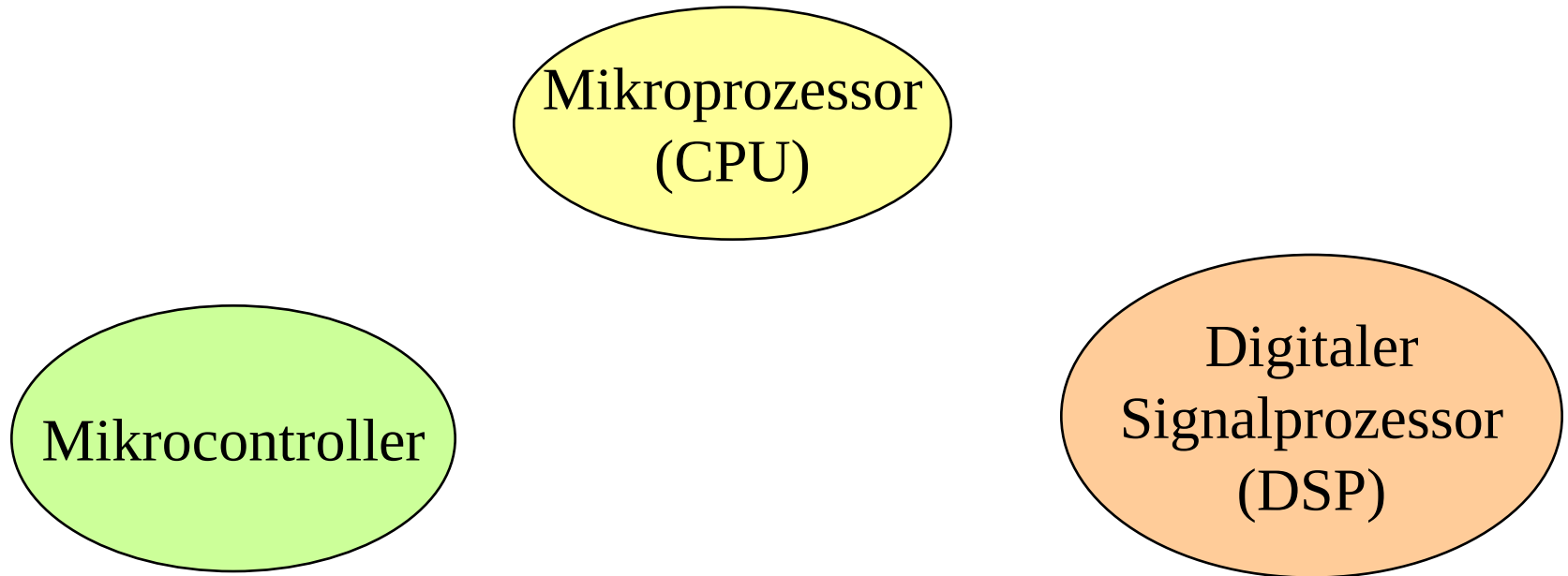
Vergleich von „Computergenerationen“

	ENIAC (1946)	8080-System (1973)	Pentium 4 im PC (2004)
Volumen:	100 m ³	0,3 l	40l
Masse:	30.000kg	500g	20kg
Leistungsaufnahme:	174kW	2,5W	300W
ROM:	16kbit	16kbit	
RAM:	1kbit	8kbit	>4 Gbit
aktive Elemente:	18.000 Röhren	20.000 Transistoren	> 50 Mio. Transistoren
Addieren von 2 12-stelligen Ziffern	200µs	100µs	< 250ps
MTBF	1h	> 1 Jahr	> 1 Jahr
Kosten:	500.000 € (Material)	ca. 50 €	ca. 1000 €

Heutige Gehäuseformen: DSP und FPGA im „Ball Grid Array“ (BGA) Gehäuse

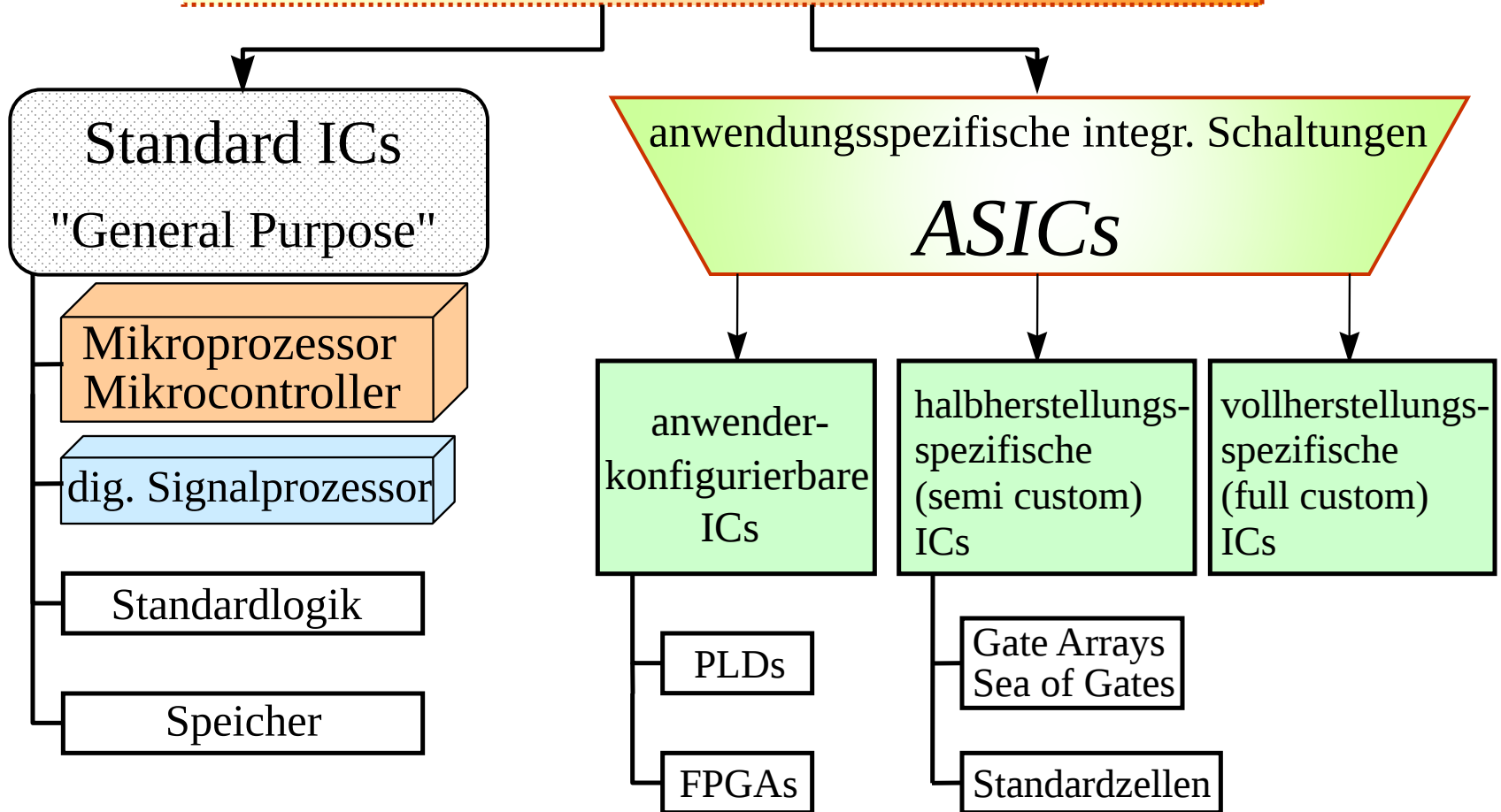


Die 'Welt' der Mikrorechner



- **Mikroprozessor** – auch CPU genannt – hierbei fehlen Speicher und Peripheriekomponenten
- **Mikrocontroller** – enthält alle Komponenten eines selbständigen Rechnersystems
- **DSP ≡ digitaler Signalprozessor** – bietet besonders hohe Rechenleistung, gekoppelt mit schnellen, parallelen Datentransfers für „Echtzeitaufgaben“

Integrierte Schaltungen (ICs)



Aufbau eines Mikrorechnersystems mit Mikroprozessor

Bussystem: Datenbus, Adressbus, Steuerbus



Mikroprozessor (CPU)

bewirkt:

Datentransfer

Arithmetik

logische Verknüpfungen

enthält:

ALU, Register,

Ablaufsteuerung

Ein/Ausgabe-Einheiten



Programmspeicher:

„nur lesbar“

ROM, PROM,

EPROM, EEPROM



Datenspeicher:

les- und beschreibbar

RAM, statisch, dynamisch

Mikrocontroller (MC)

enthalten:

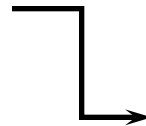
CPU Programmspeicher, (kleine) Datenspeicher, Zähler (Timer), E/A-Einheiten

MC stellen vollständige, autonome Rechnersysteme „in einem Chip“ dar

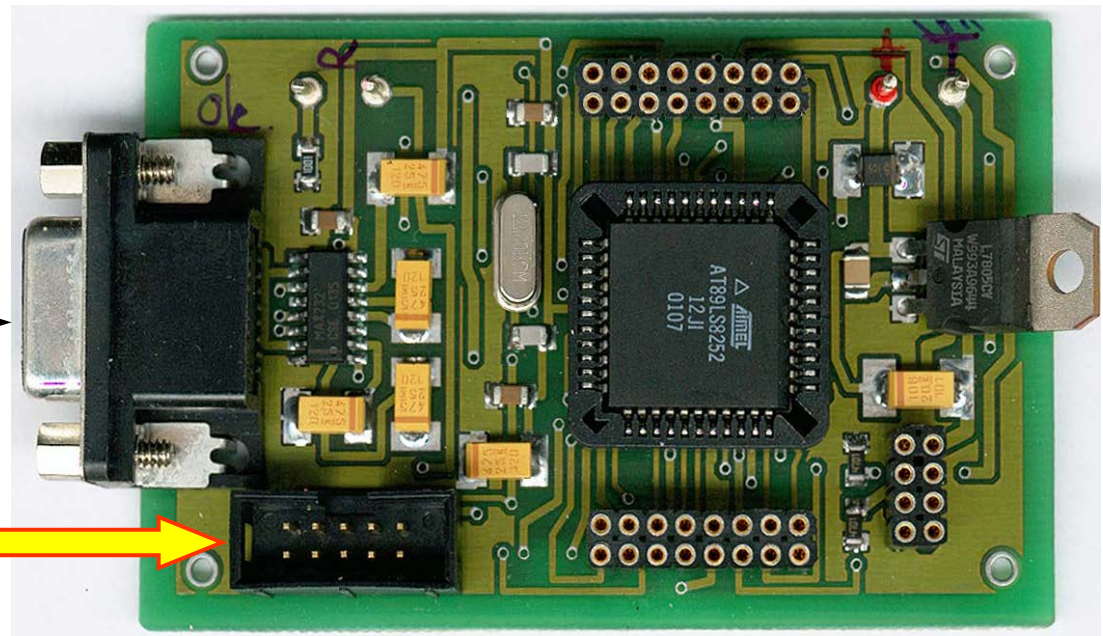


Beispiel einer MC-Universalplatine

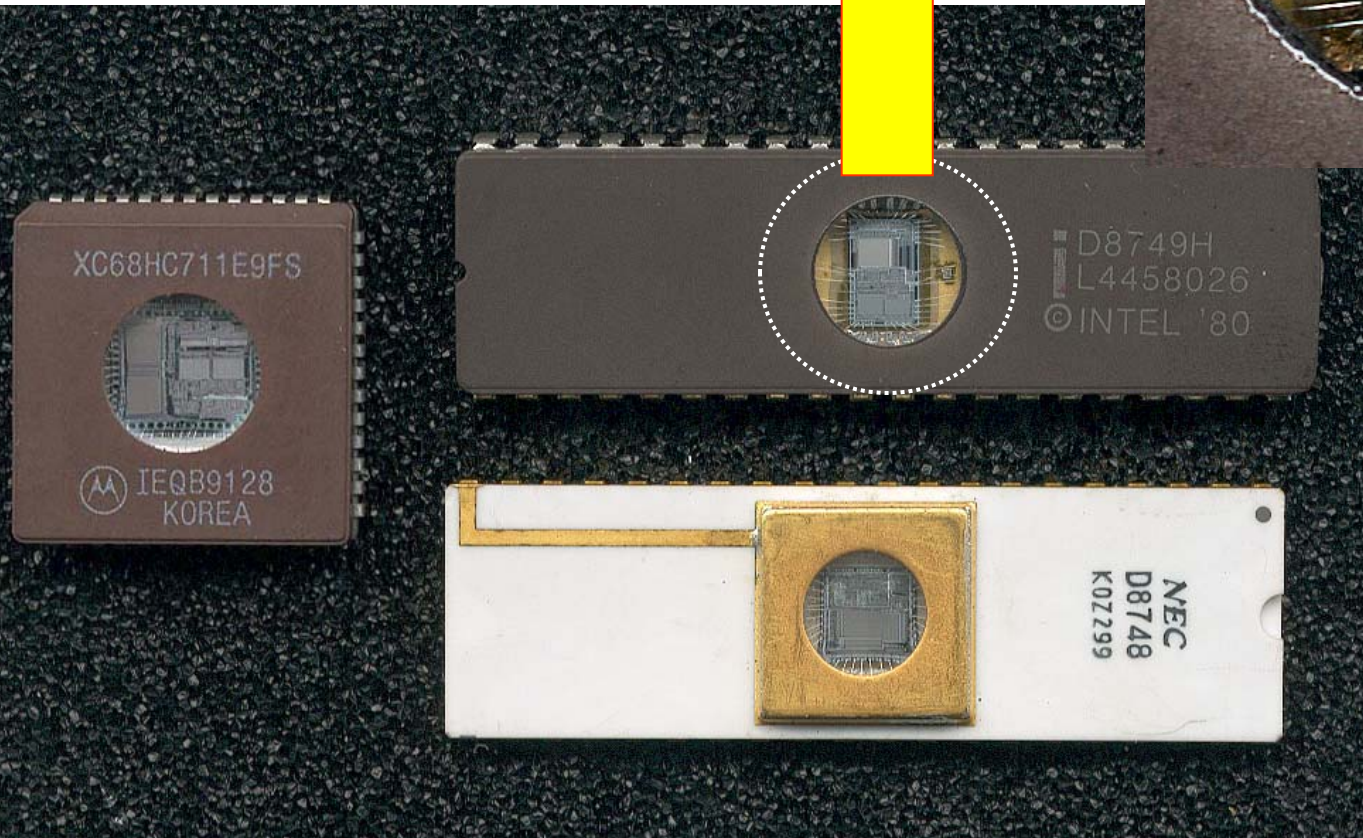
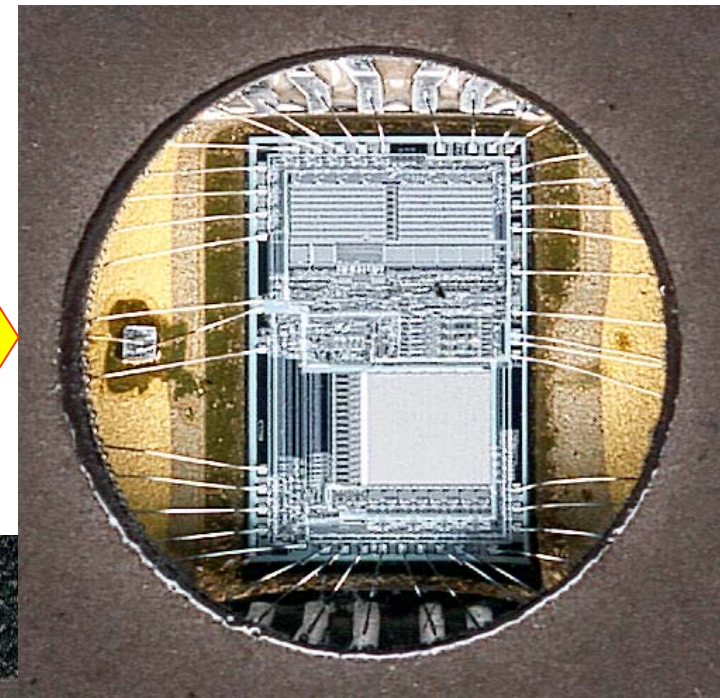
serielle (V.24 oder RS232)
Schnittstelle



Programmierinterface
ISP (In System Programmer)

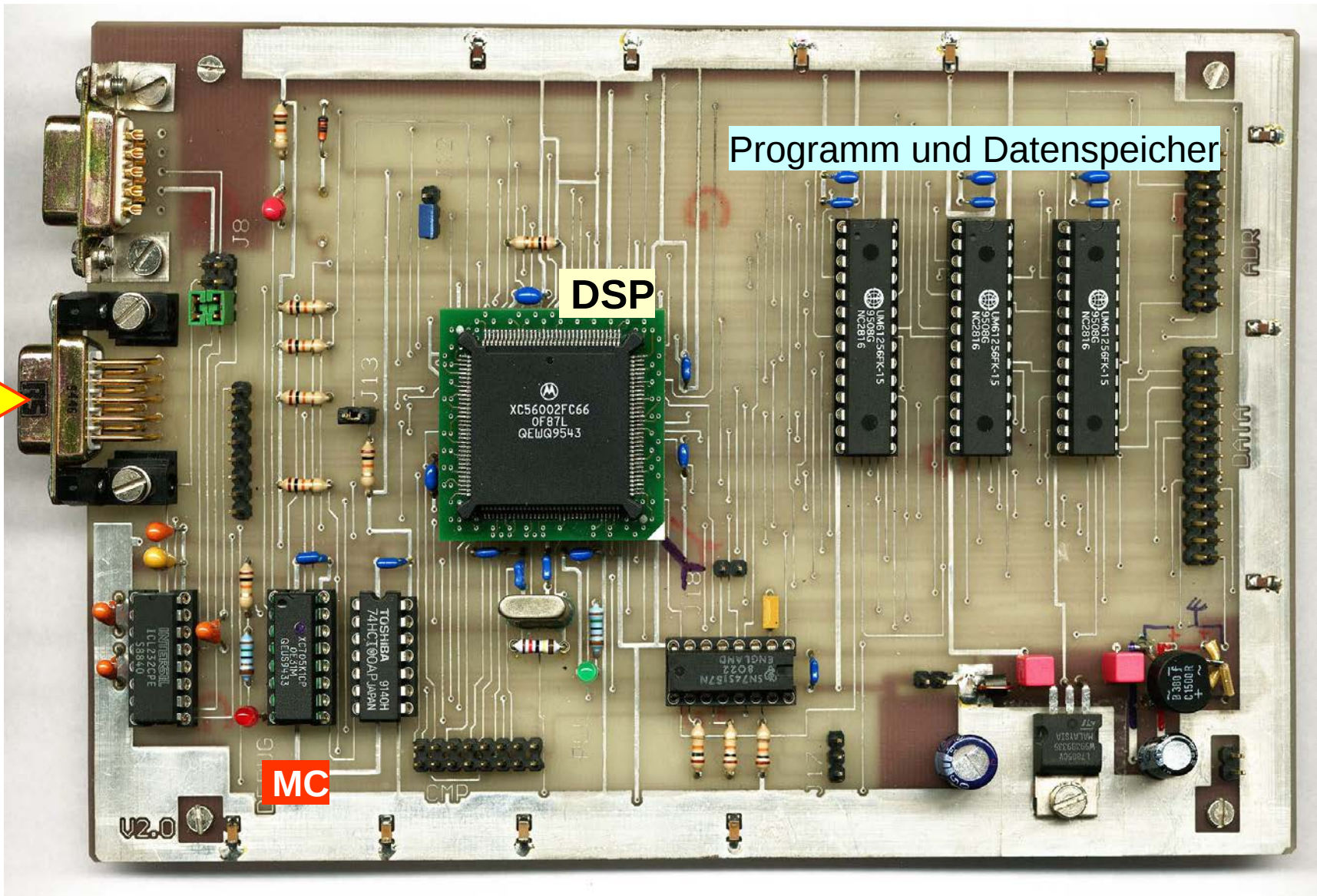


Mikrocontroller (MC) mit EPROM als Programmspeicher

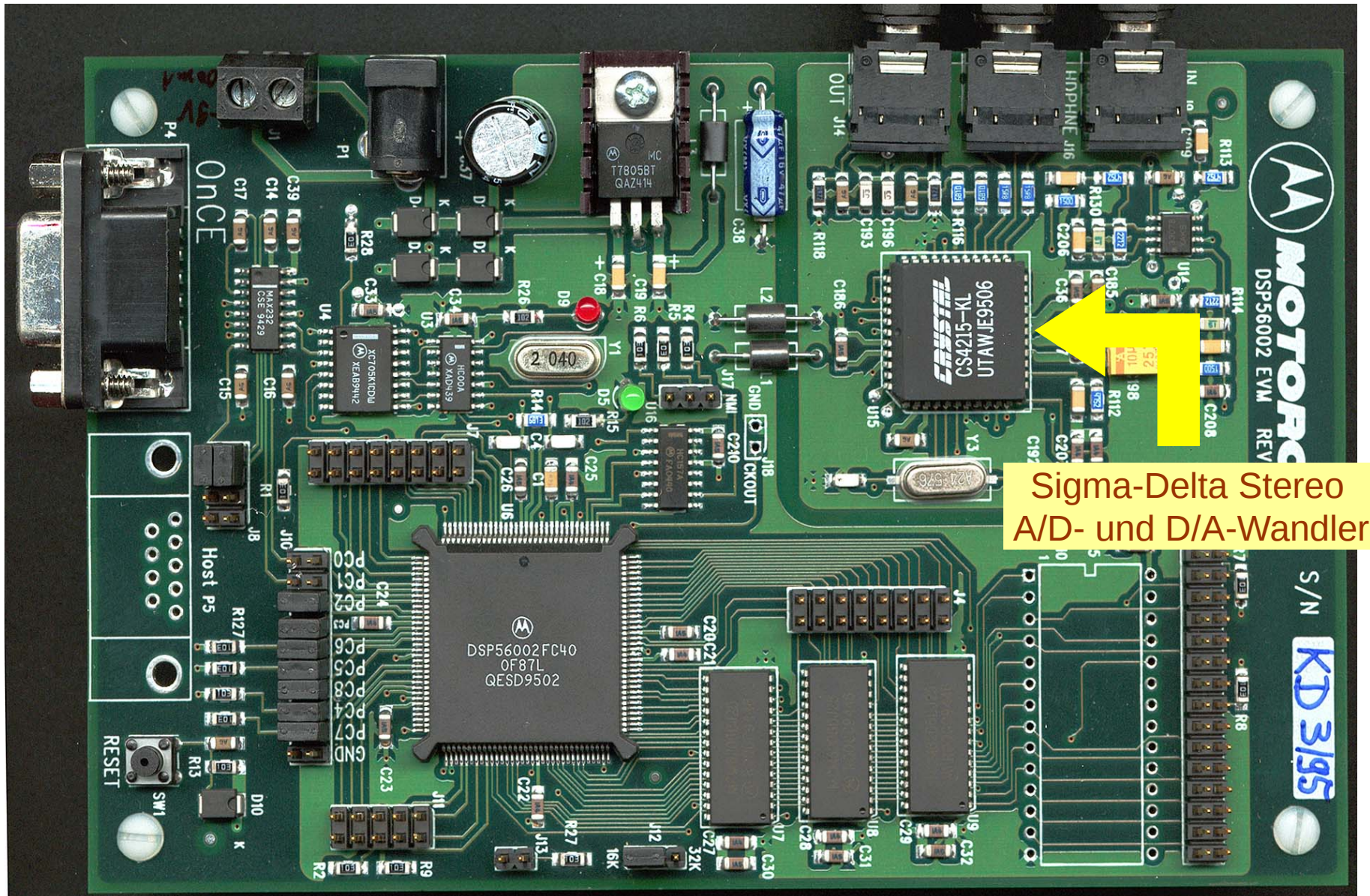


Aufbau eines digitalen Signalprozessorsystems (DSP 56002) mit Programmierereinrichtung und „Emulator“

Programmier- und Datenschnittstelle

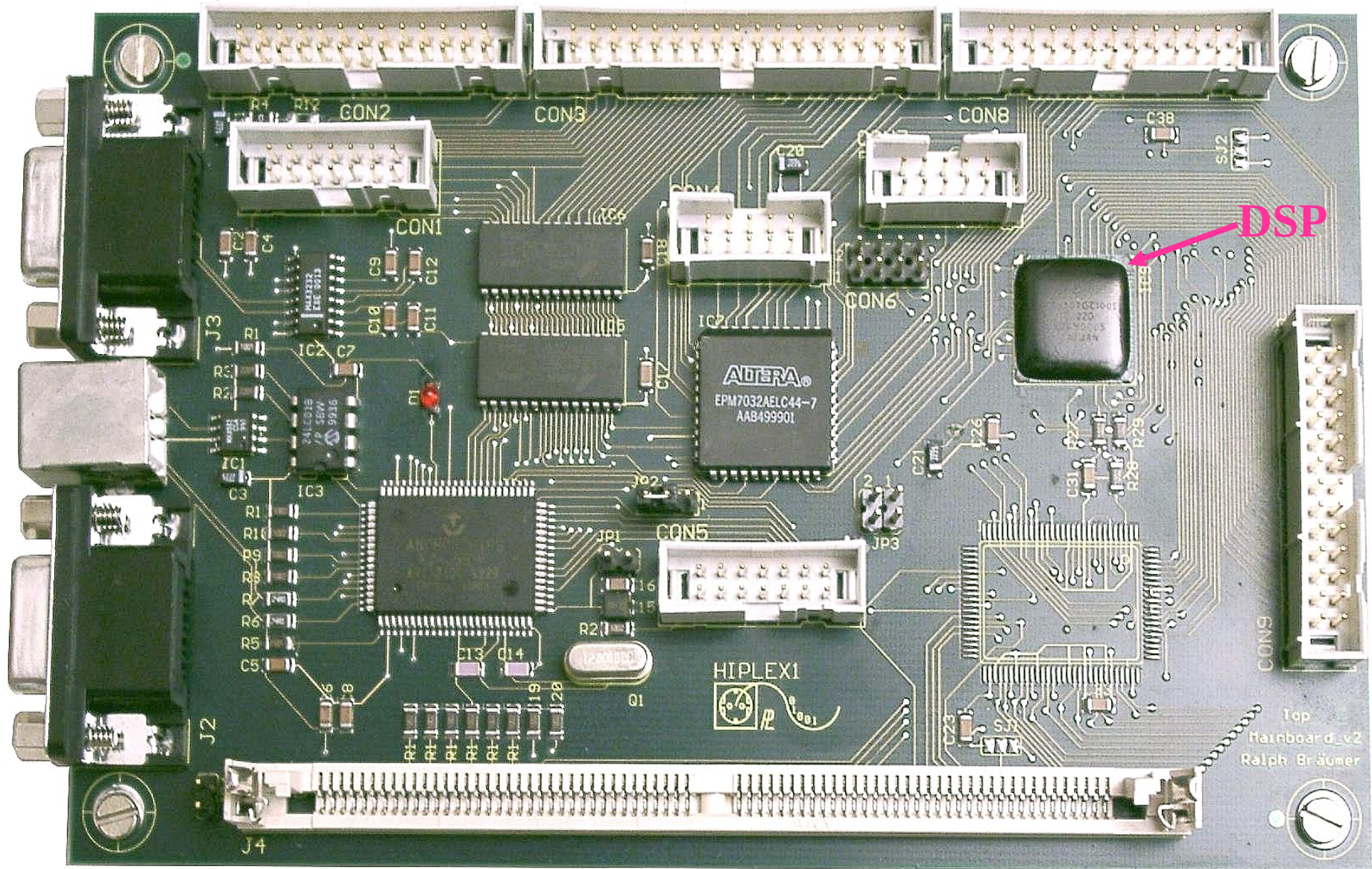


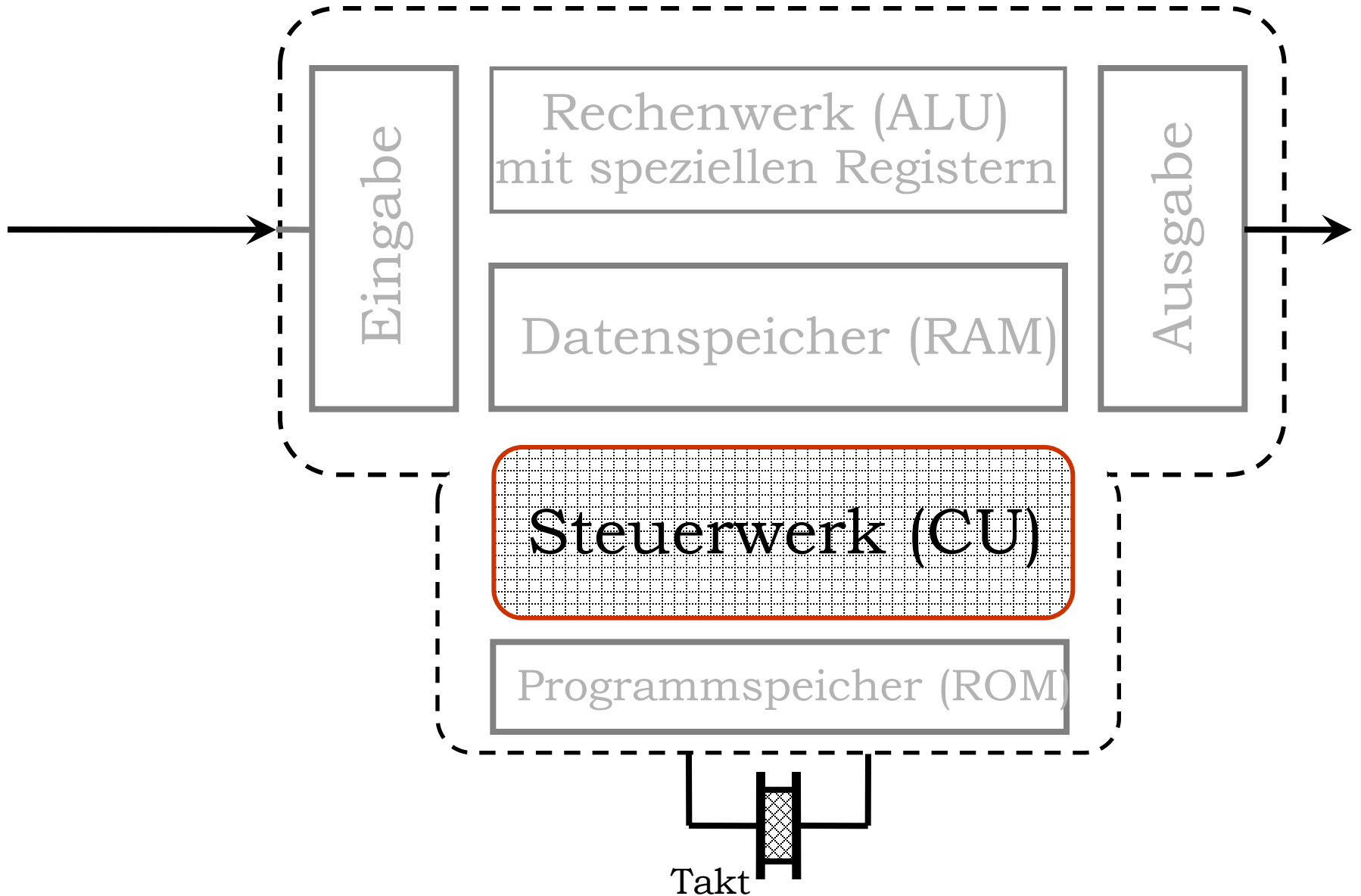
DSP 56002 Experimentiersystem für Audio-Signalverarbeitung



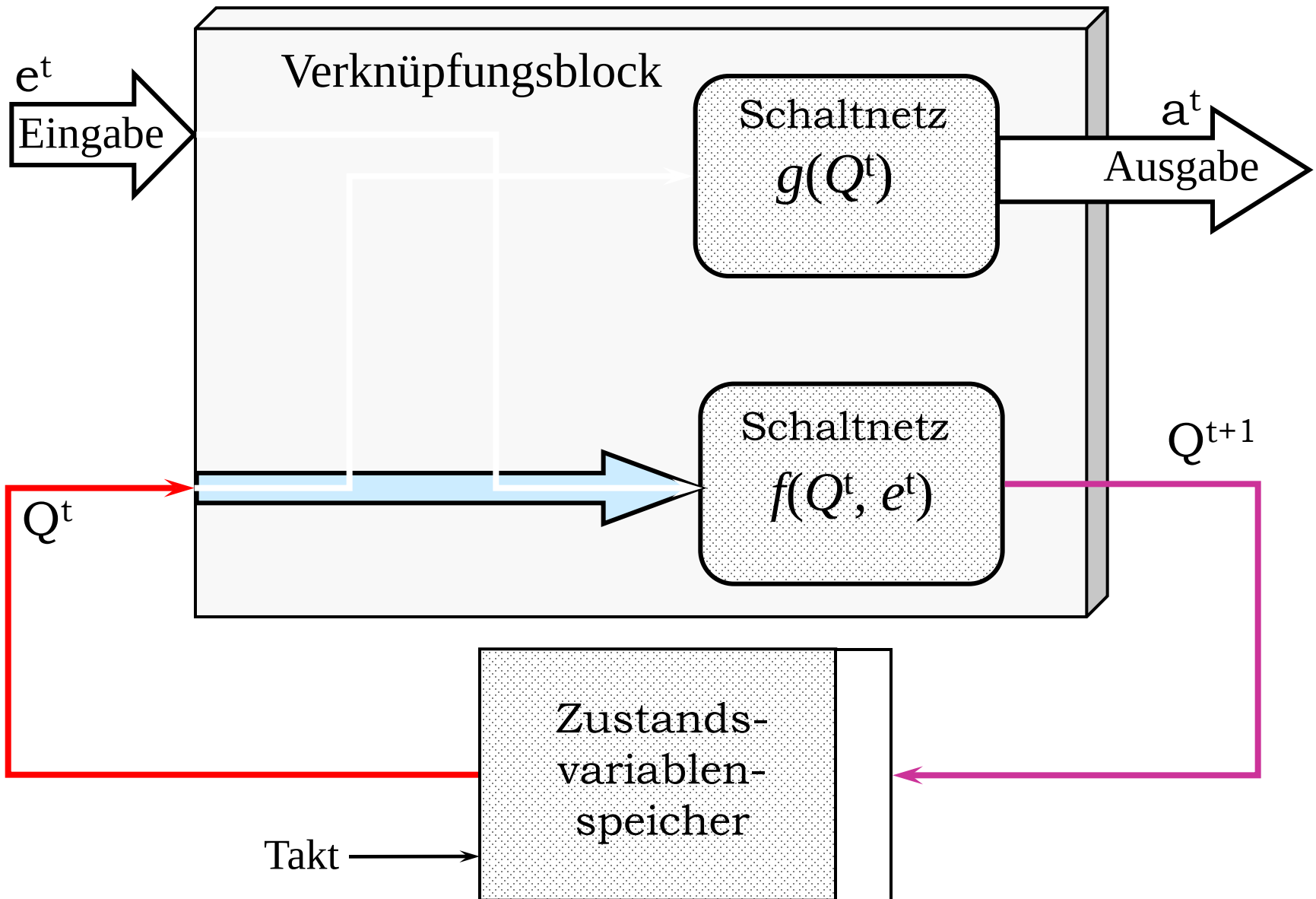
Sigma-Delta Stereo
A/D- und D/A-Wandler

OFDM-System mit DSP und anwendungsspezifischer Logik

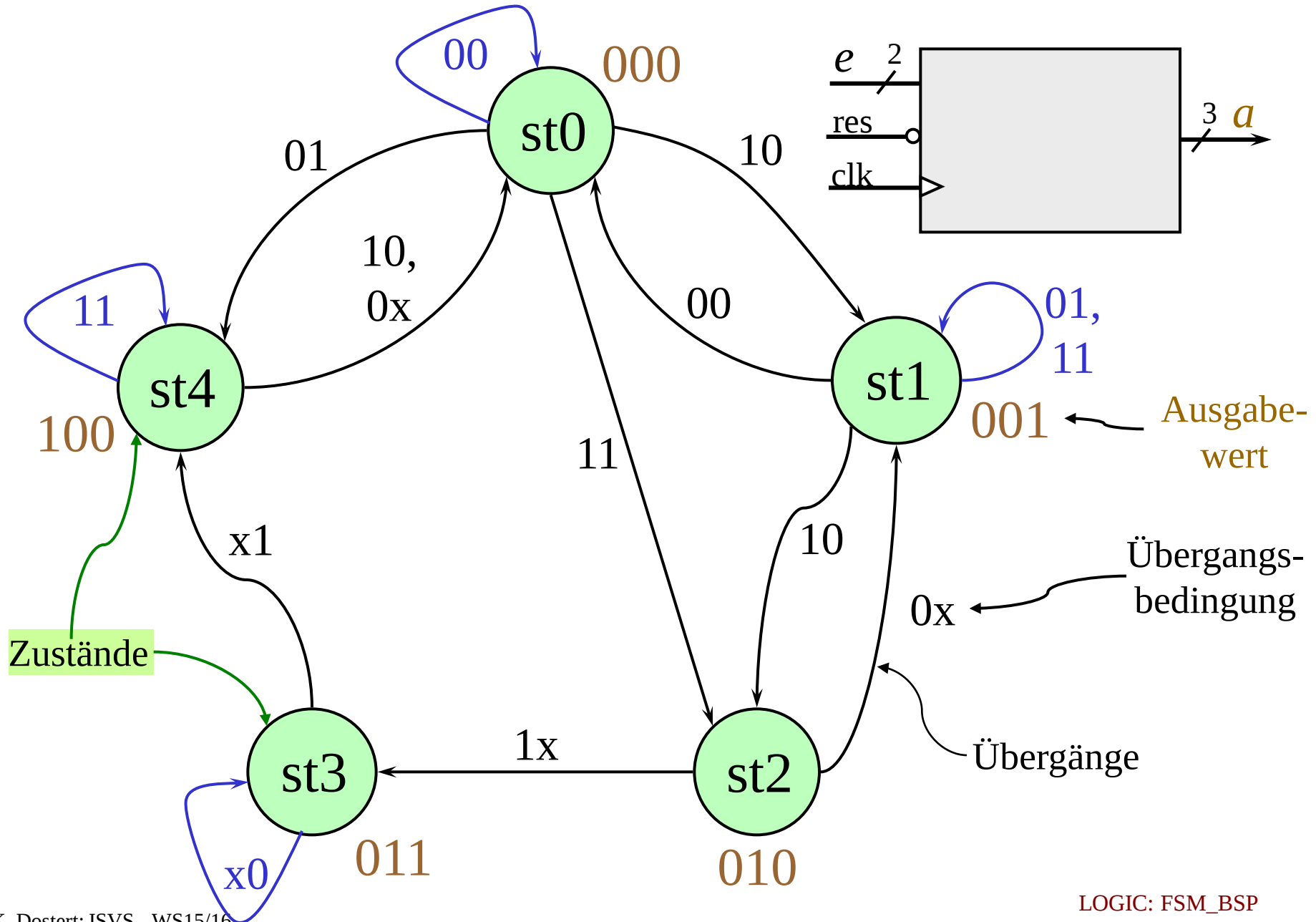




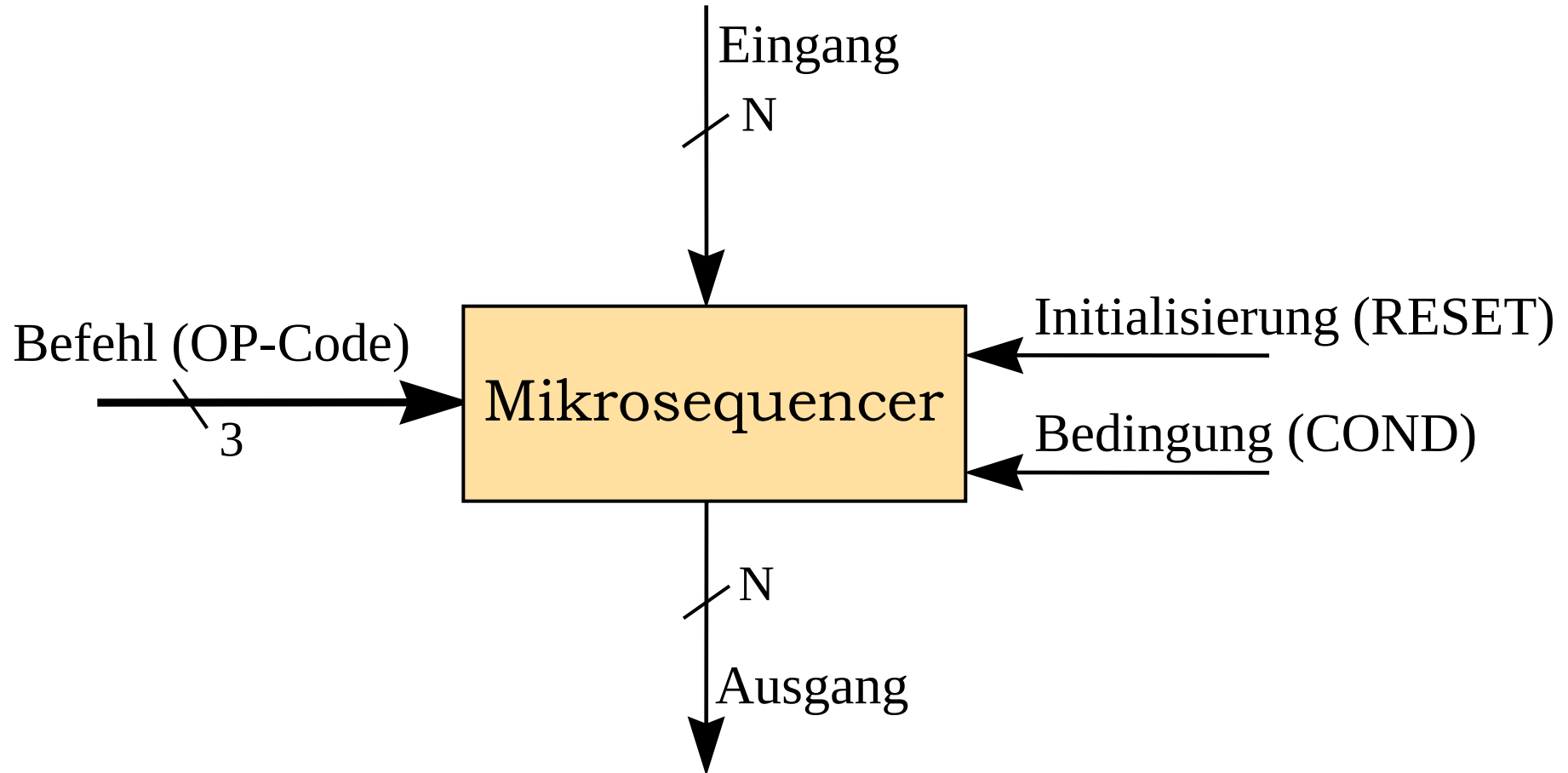
Der Moore-Automat

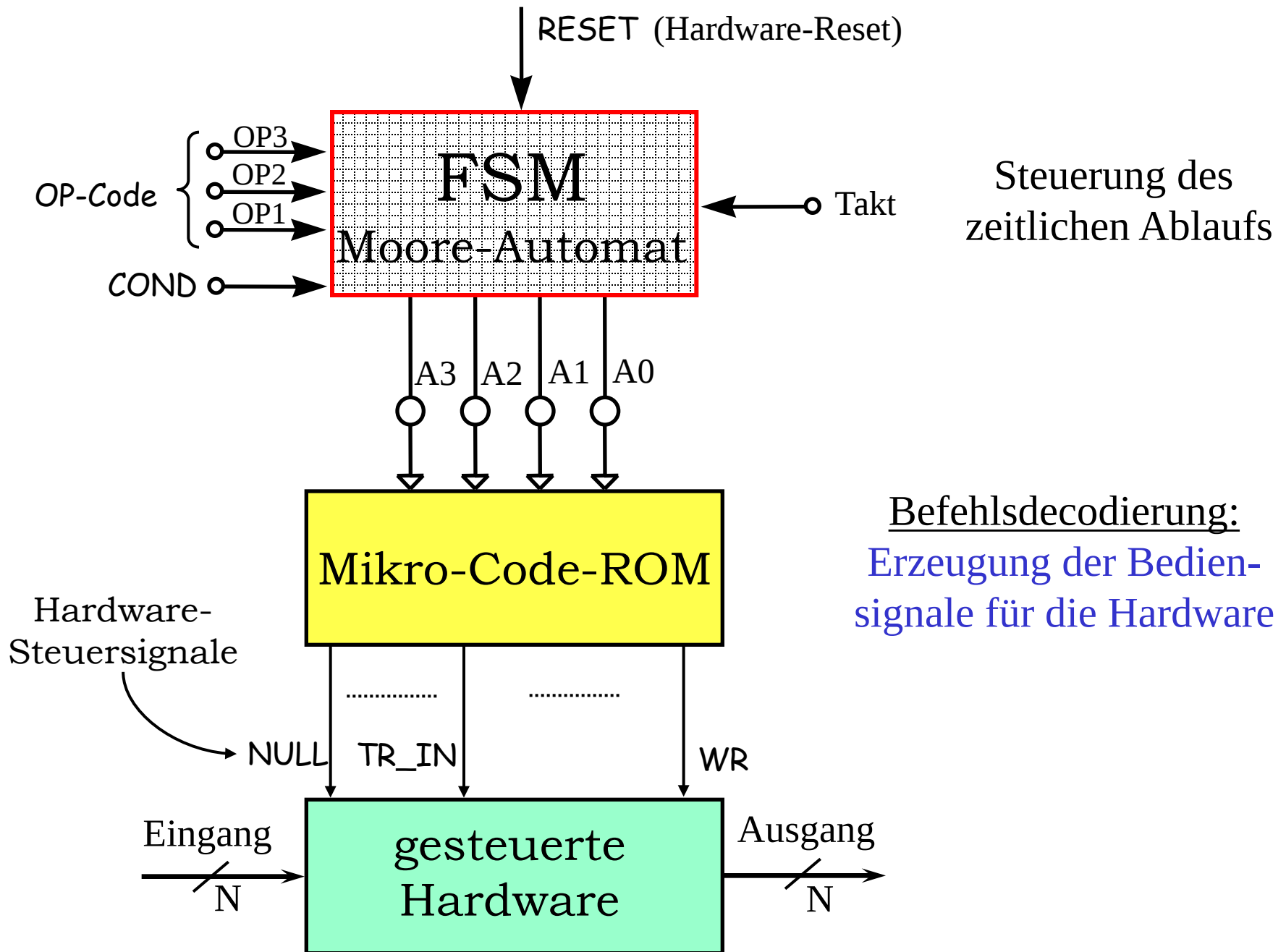


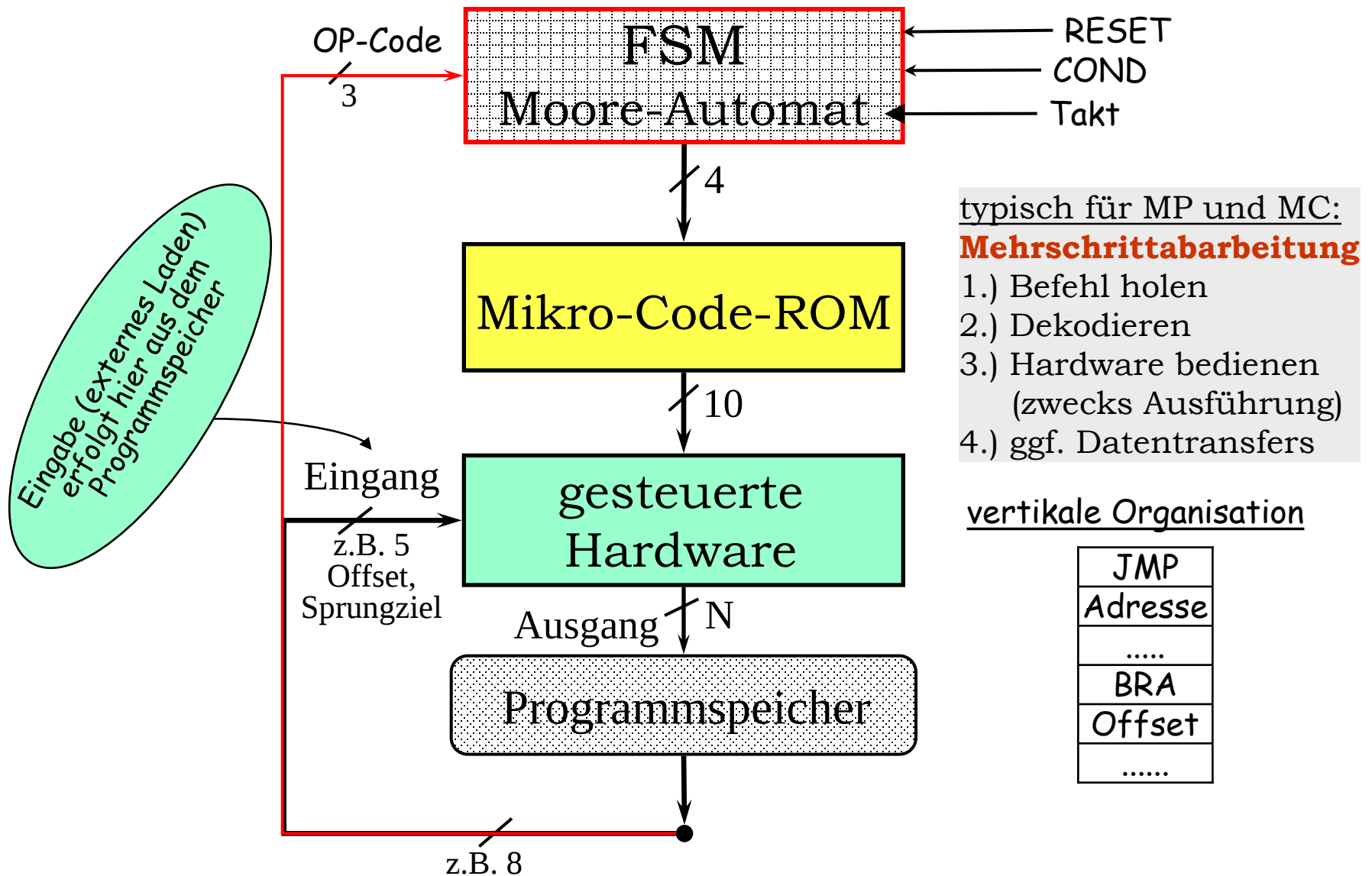
Zustands-Übergangsdiagramm für einen Automaten



Ausschnitt aus dem Steuerwerk eines Mikrorechners





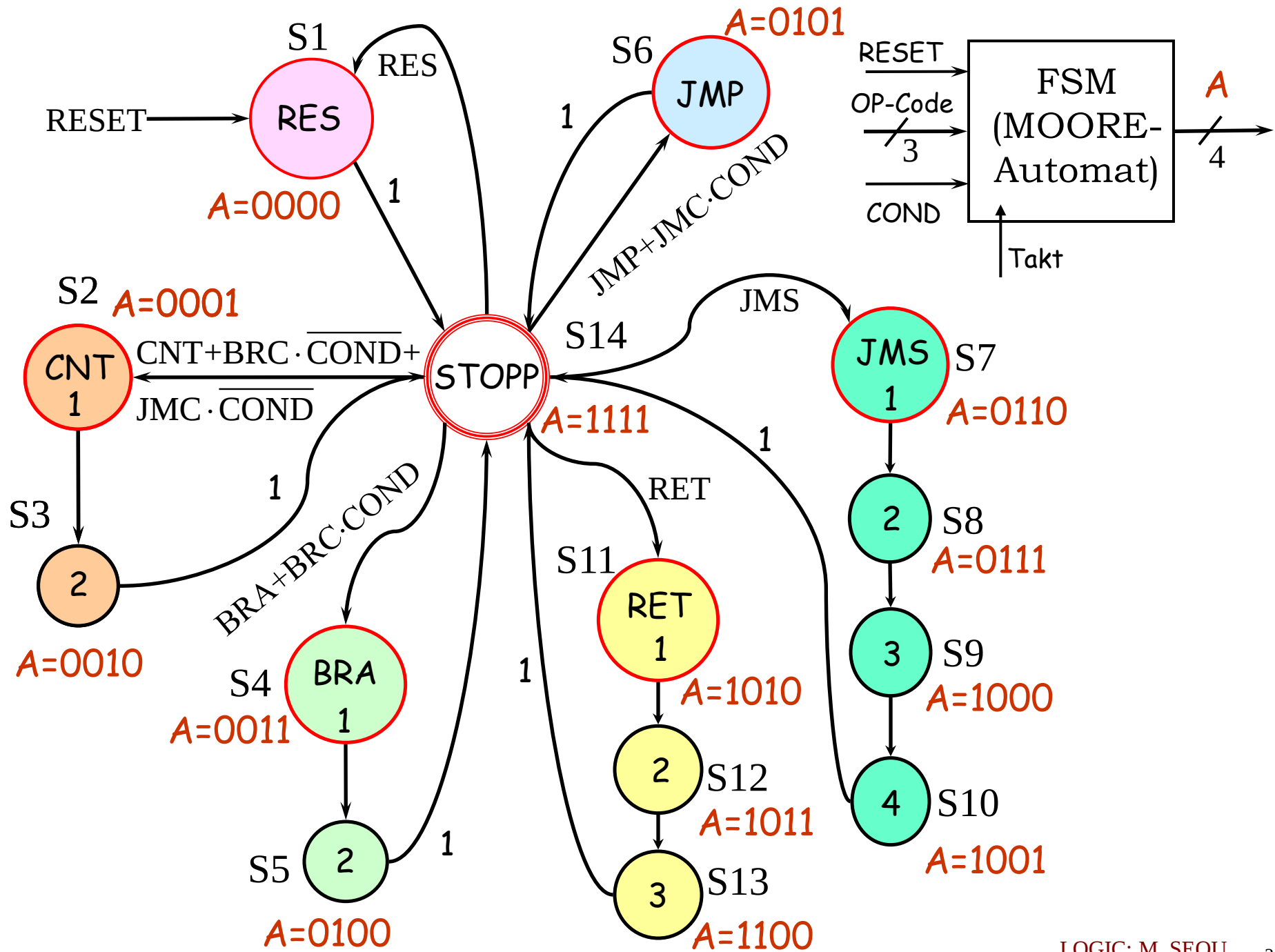


typisch für DSP:
Einschrittverarbeitung mit parallelen Datentransfers

horizontale Organisation (VLIW*-Prinzip)

Adresse(n) / Daten	OP-Code(s)
--------------------	------------

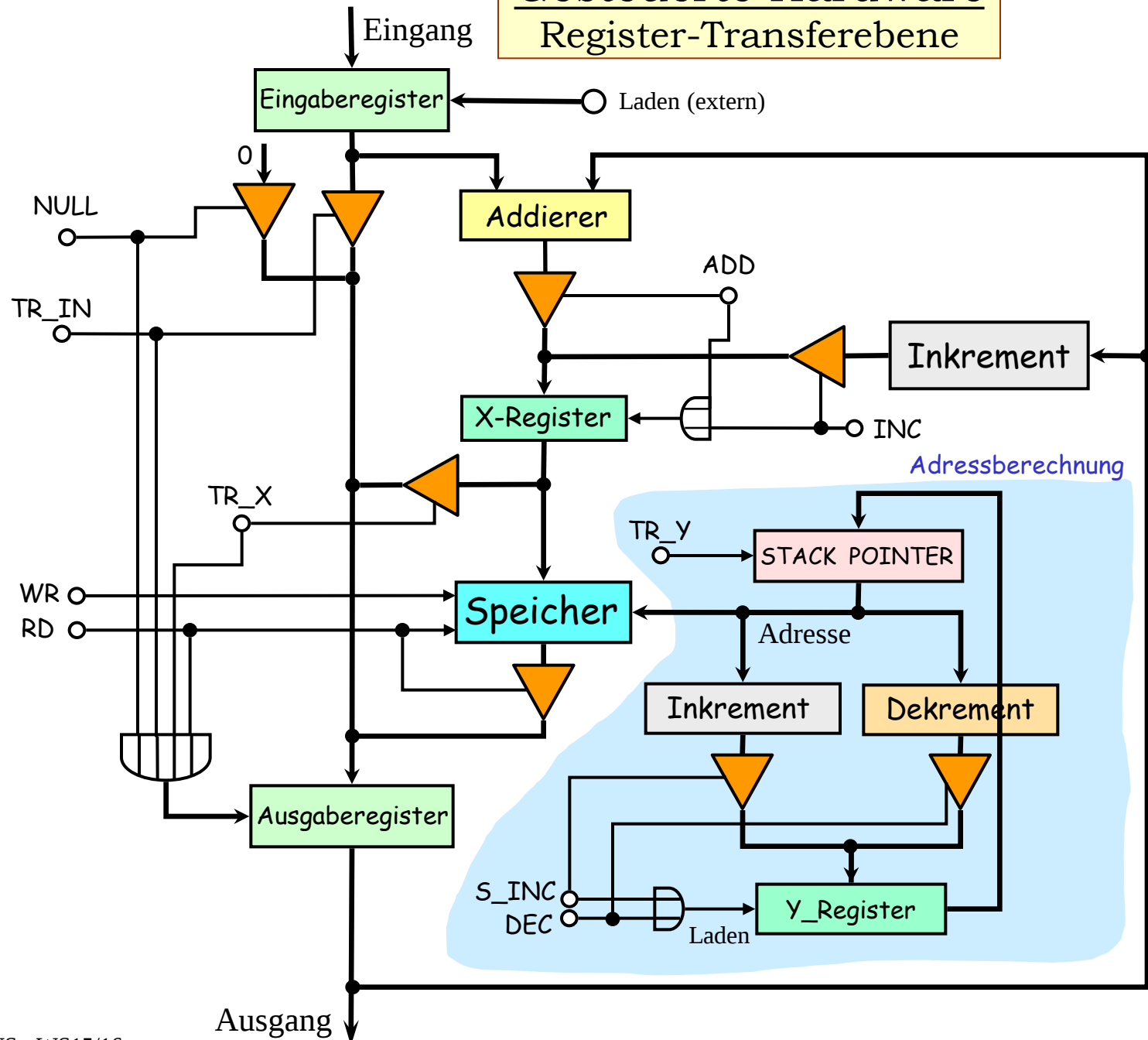
*very long instruction word,
 (heute bis 256bit)



Liste der implementierten Befehle

Befehl	Code	Mnemonic	Beschreibung
reset	000	RES	transfer null-vector to output
continue	001	CNT	increment output
branch	010	BRA	output:= output + input
branch conditionally	011	BRC	output:= output + input, if COND=1, otherwise: increment
jump	100	JMP	output:= input
jump conditionally	101	JMC	output:= input, if COND = 1 otherwise: increment output
jump to subroutine	110	JMS	output:= output +1 $\Rightarrow$ save to STACK, then output:= input
return	111	RET	fetch last STACK entry and transfer to output

Gesteuerte Hardware Register-Transferebene

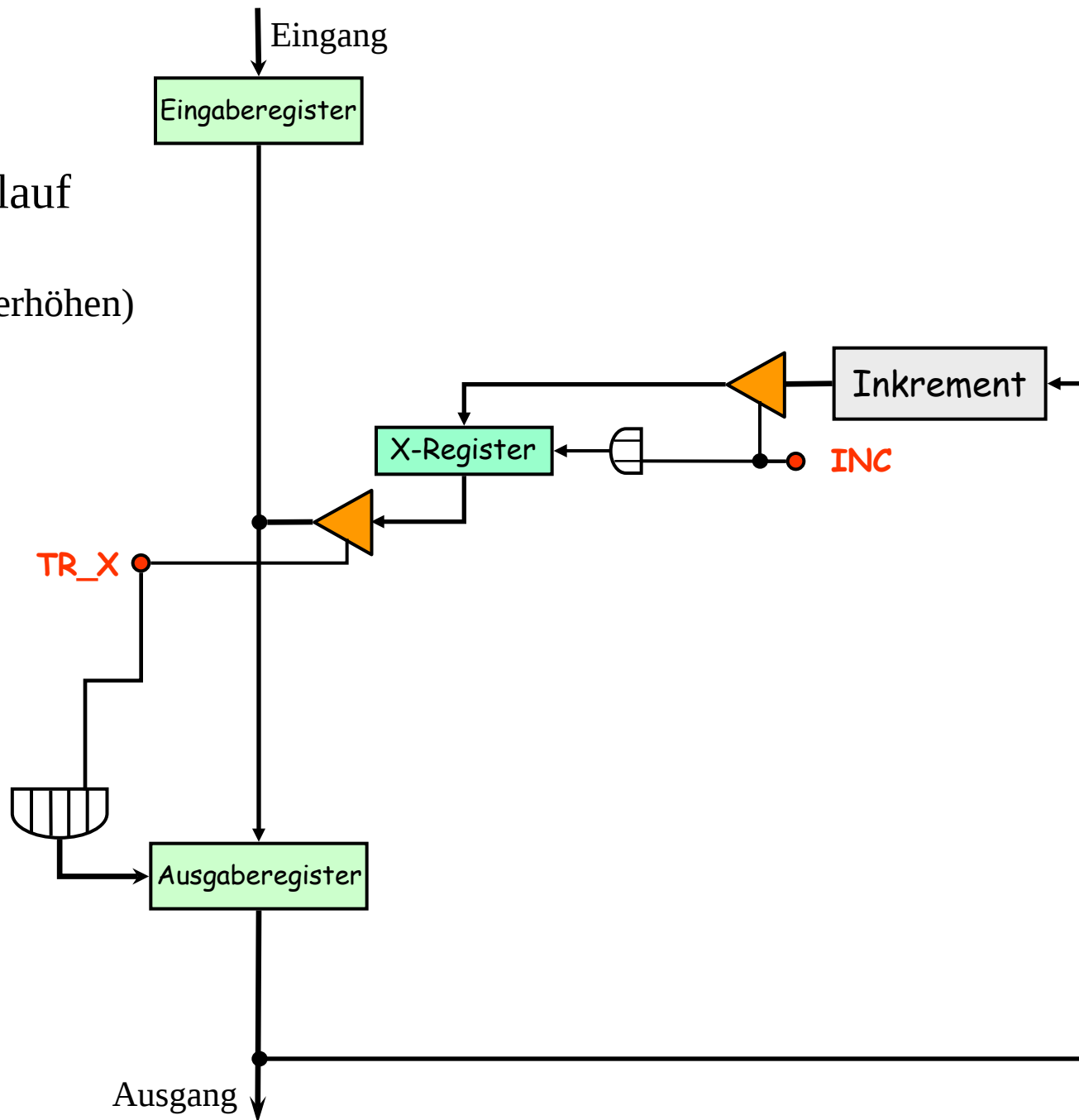


Mikroprogramm zur Hardwaresteuerung

Steuersignale➔

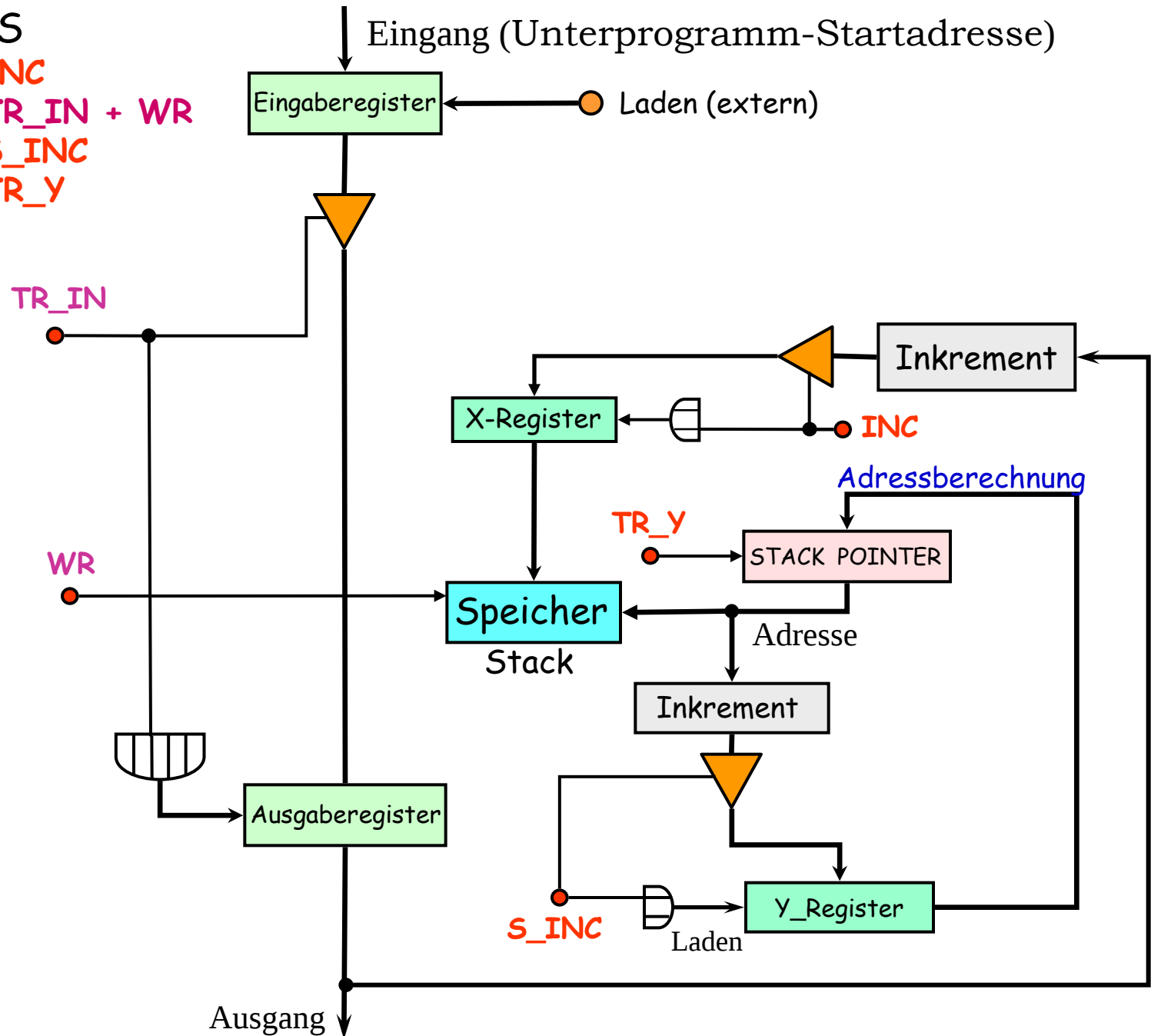
OP-Code	Null	TR_IN	TR_X	ADD	INC	S_INC	DEC	TR_Y	RD	WR
RES	1	0	0	0	0	0	0	0	0	0
CNT	0	0	0	0	1	0	0	0	0	0
	0	0	1	0	0	0	0	0	0	0
BRA	0	0	0	1	0	0	0	0	0	0
	0	0	1	0	0	0	0	0	0	0
JMP	0	1	0	0	0	0	0	0	0	0
JMS	0	0	0	0	1	0	0	0	0	0
	0	1	0	0	0	0	0	0	0	1
	0	0	0	0	0	1	0	0	0	0
	0	0	0	0	0	0	0	1	0	0
RET	0	0	0	0	0	0	1	0	0	0
	0	0	0	0	0	0	0	1	0	0
	0	0	0	0	0	0	0	0	1	0

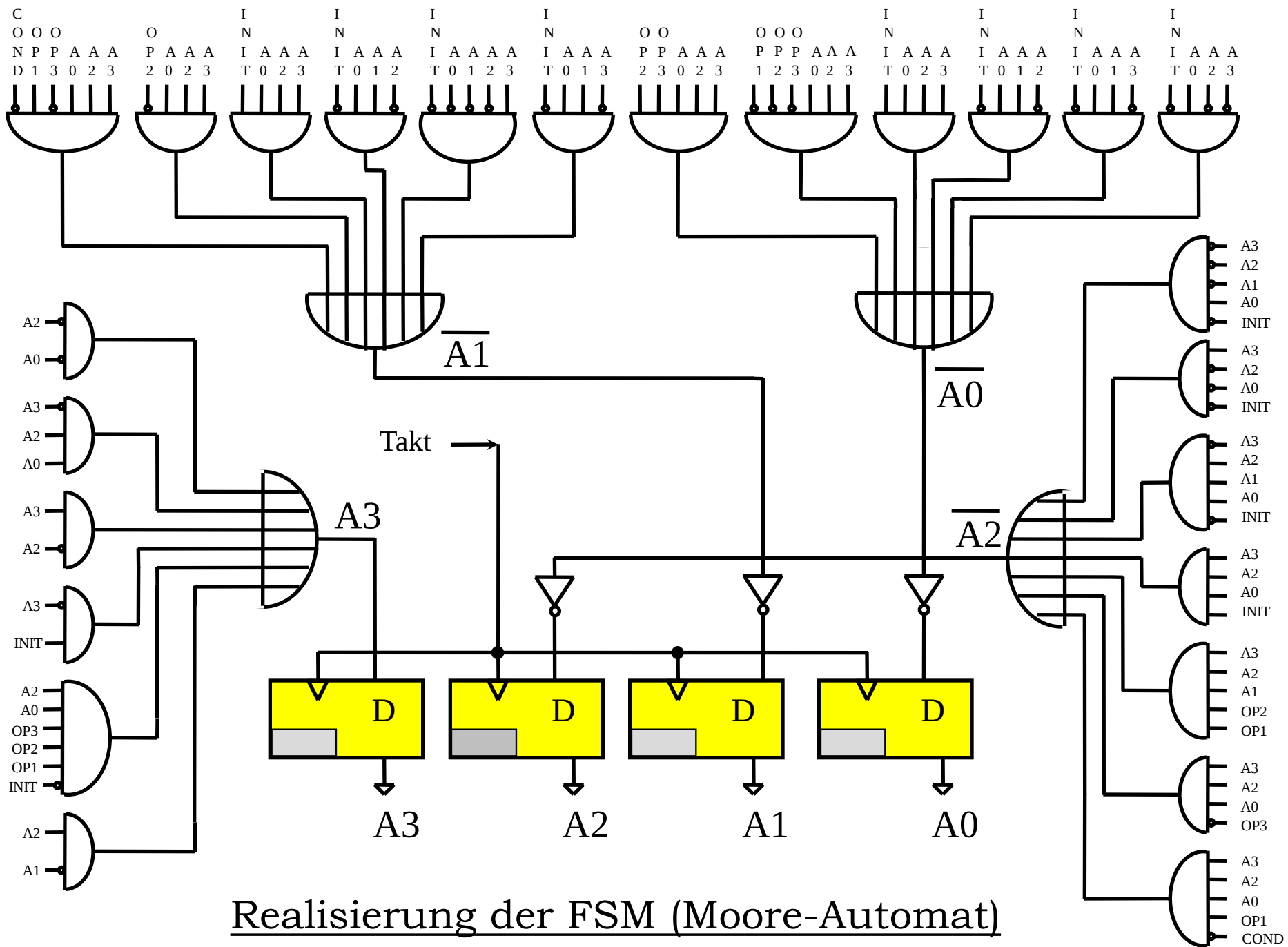
Befehlsablauf
CNT
(Ausgang um 1 erhöhen)



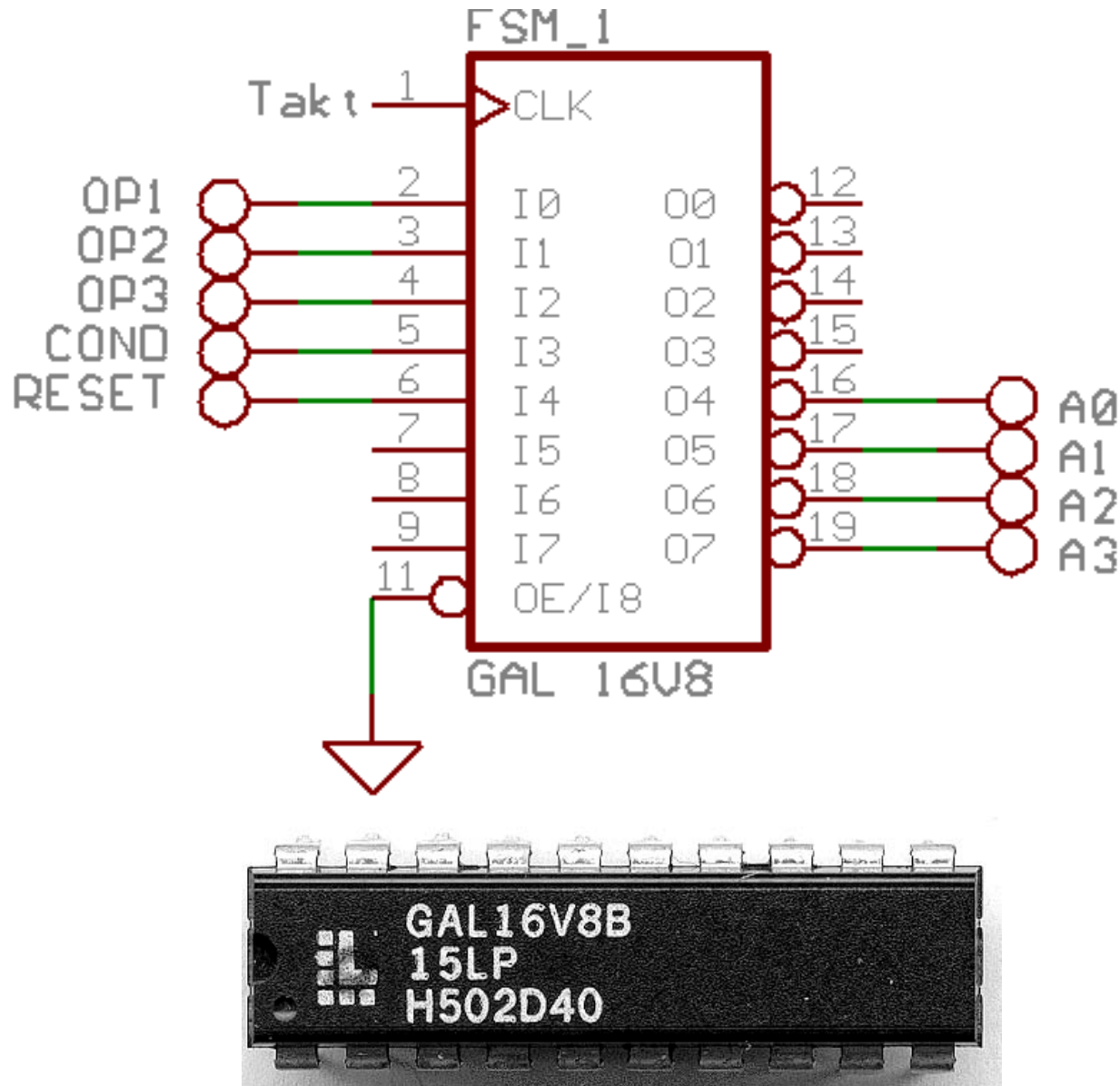
JMS

1. INC
2. TR_IN + WR
3. S_INC
4. TR_Y





Implementierung der FSM in einem einfachen PLD (Programmable Logic Device)



binäre Zahlendarstellung zur Basis “2“

$$Z_b = \sum_{i=0}^{N-1} x_i \cdot 2^i = x_0 \cdot 2^0 + x_1 \cdot 2^1 + x_2 \cdot 2^2 + \dots + x_{N-1} \cdot 2^{N-1}, \quad \text{mit } x_i \in \{0,1\}$$

Schreibweise als „Binärvektor“ und Komplement

$$\mathbf{p} = \sum_{i=0}^{N-1} p_i \cdot 2^i \Rightarrow \mathbf{p} + \bar{\mathbf{p}} = 2^N - 1$$

Beispiel

p	10011101011001
$\bar{\mathbf{p}}$	01100010100110
$\mathbf{p} + \bar{\mathbf{p}}$	11111111111111

Zweierkomplementdarstellung

$$-\mathbf{p} - \bar{\mathbf{p}} = -2^N + 1, \text{ oder } -\mathbf{p} = -2^N + \bar{\mathbf{p}} + 1$$

Beispiel für die Verwendung des Zweierkomplements

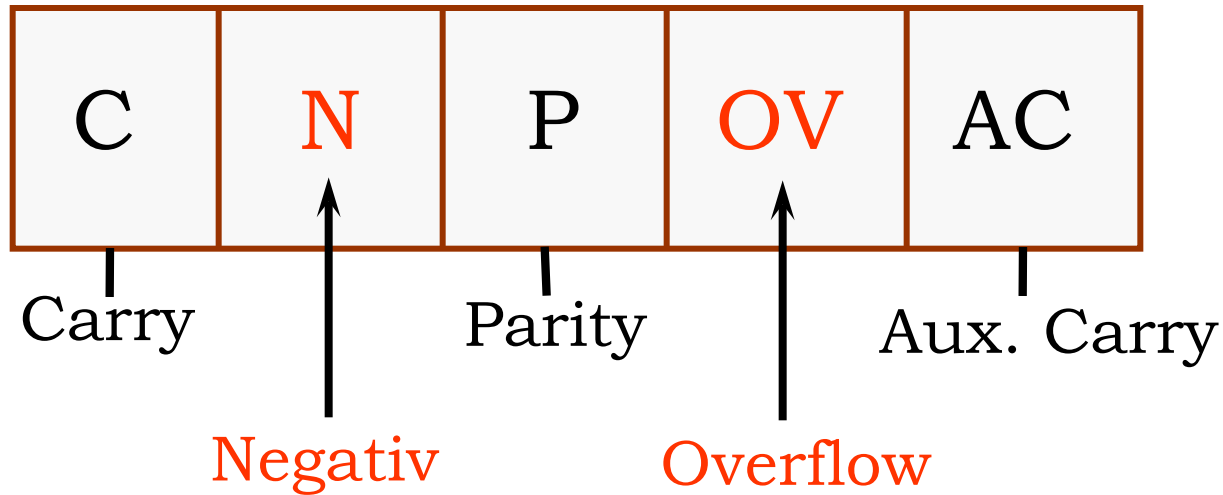
9	1001	binäre Addition	
7	0111	9	1001
-7	1000+1-2 ⁴	-7	-2 ⁴ +1001
		Summe:	+2 ⁴ -2 ⁴ +0010
		9-7=2	0010

Dezimalwert einer Zweierkomplementzahl

$$Z_b = -x_N \cdot 2^N + \sum_{i=0}^{N-1} x_i \cdot 2^i$$

Wozu sind „Flags“ (Markierungsbits) nötig?

	0	1	1	1	1	1	1	1	127
+	0	0	0	0	0	0	0	1	1
=	1	0	0	0	0	0	0	0	-128



Was ist „Sign-Extension“ (Vorzeichenerweiterung)

- bei positiven Zahlen ist der Sachverhalt trivial, da Nullen vorangestellt werden
- bei negativen Zahlen werden Einsen vorangestellt

$$11\ 0111\ 1111 \equiv -512 + 256 + 127 = -129$$

oder

$$111\ 0111\ 1111 \equiv -1024 + 512 + 256 + 127 = -129$$

oder

$$1111\ 0111\ 1111 \equiv -2048 + 1024 + 512 + 256 + 127 = -129$$

jedes weitere Bit liefert den Beitrag

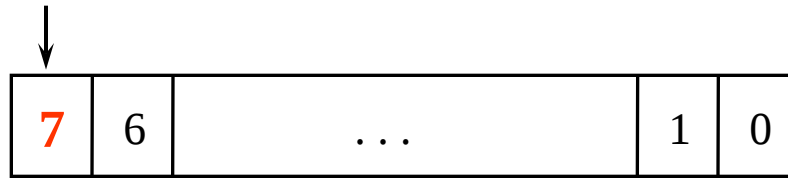
$$-2^{N+1} + 2^N = -2^N - 2^N + 2^N = -2^N$$

⇒ es ergibt sich keine Veränderung

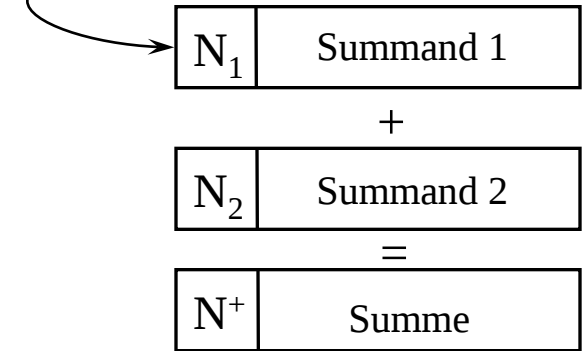
Fallunterscheidungen hinsichtlich möglicher Additionsfehler

Summanden	nach Addition	Bemerkung
Fall I + +	Carry ist immer '0'	
	wenn $N^+ = 0 \Rightarrow OV = 0$	wenn $N^+ = 1 \Rightarrow OV = 1 \Rightarrow$ Korrektur nötig
Fall II - -	Carry ist immer '1'	
	wenn $N^+ = 1 \Rightarrow OV = 0$	wenn $N^+ = 0 \Rightarrow OV = 1 \Rightarrow$ Korrektur nötig
Fall III + -	$C = 1 \Rightarrow$ Summe positiv $N^+ = 0$	immer $OV = 0$, d.h. immer korrekt
	$C = 0 \Rightarrow$ Summe negativ $N^+ = 1$	

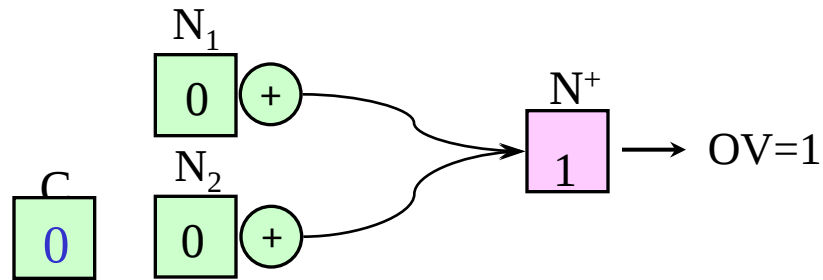
N-Flag im 8bit Akkumulator



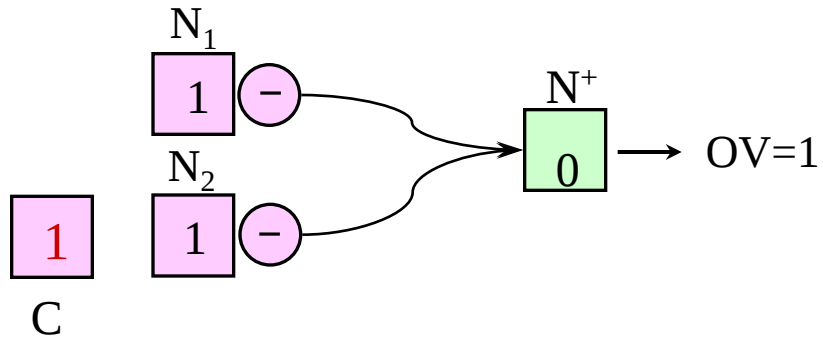
Negativ-Flags



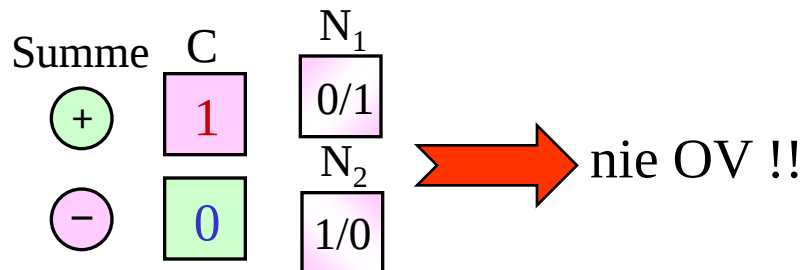
Fall I



Fall II

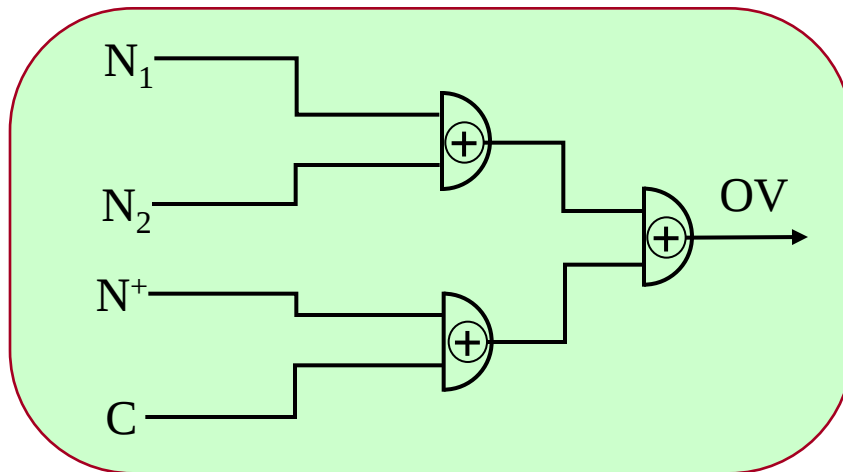


Fall III



N ₁	N ₂	N ⁺
0 (+)	0 (+)	1 → OV
1 (-)	1 (-)	0 → OV

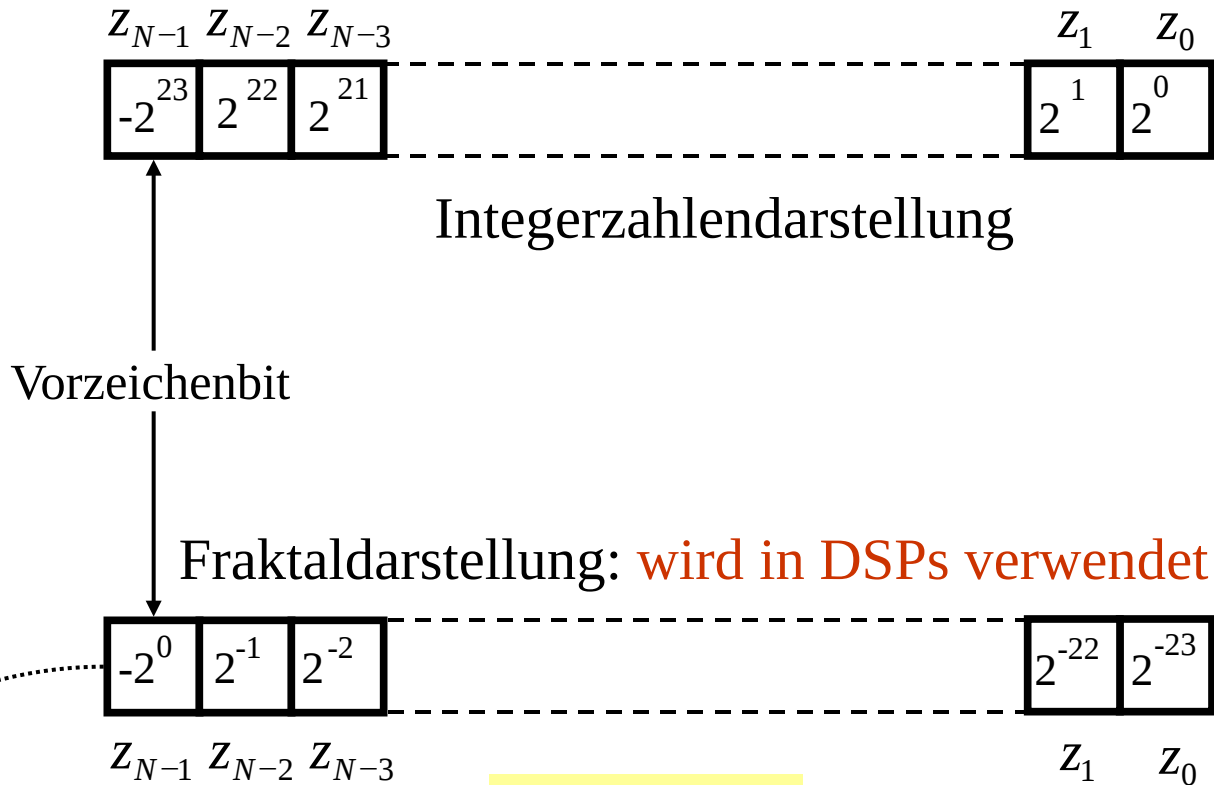
Realisierung der OV-Flag-Erzeugung



$$OV = N_1 \oplus N_2 \oplus N^+ \oplus C$$

	N_1	N_2	N^+	C	OV
Fall I	0	0	0 Fehler wenn '1'	0 immer	0 1
Fall II	1	1	1 Fehler wenn '0'	1 immer	0 1
Fall III	0 1	1 0	0 (pos.) 1 (neg.)	1 0	0

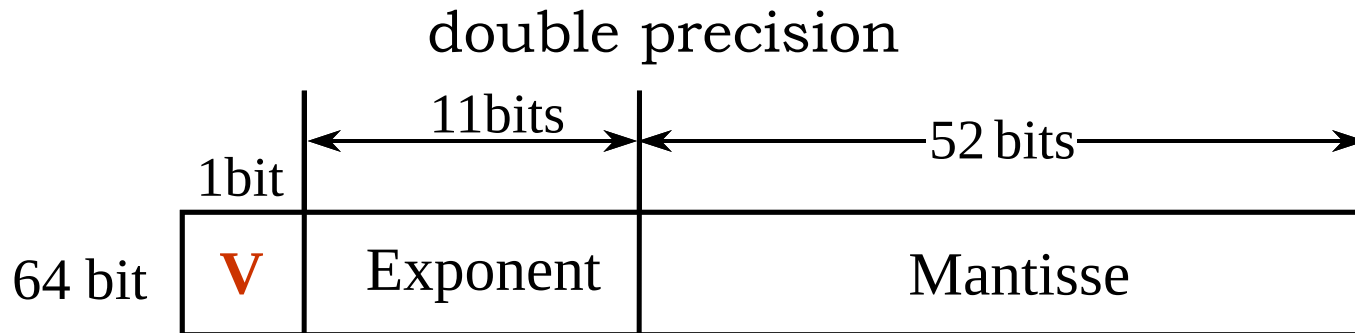
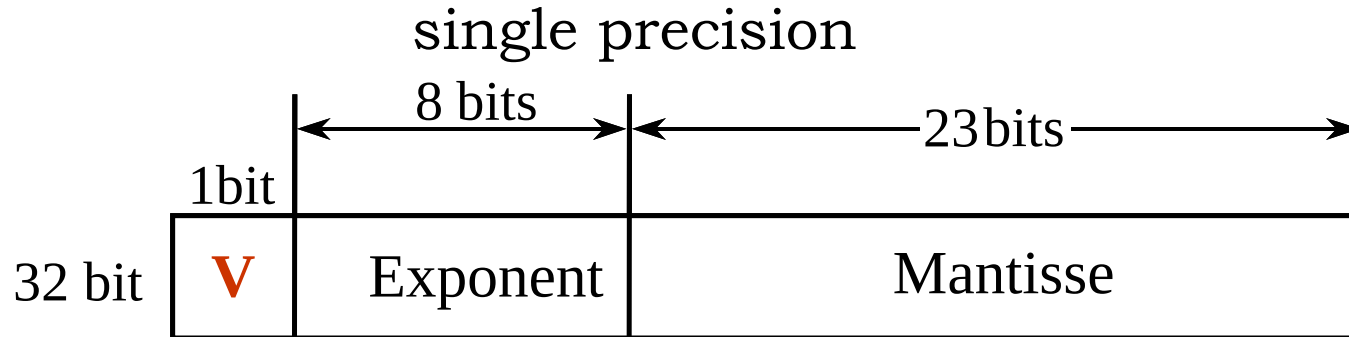
Übergang von der Integer- zur Fraktalzahlendarstellung auf Zweierkomplementbasis



Dezimalwert

$$Z_{\text{fr}} = \frac{Z_i}{2^{N-1}} = \frac{-z_{N-1} \cdot 2^{N-1} + \sum_{i=0}^{N-2} z_i \cdot 2^i}{2^{N-1}} = -z_{N-1} \cdot 2^0 + \sum_{i=0}^{N-2} z_i \cdot 2^{i-N+1}$$

Der IEEE 754 Floating Point Standard



Dezimalwert

$$Z_d = (-1)^V \cdot M \cdot 2^{E_{eff}}$$

$$\text{mit } E_{eff} = Z_E - Bias$$

Bias:= 127 bzw. 1023

Mantisse

$$M = 1 + \sum_{i=1}^N m_i \cdot 2^{-i} ; \quad \text{z.B. } N=23 \text{ bzw. } 52$$

↑ erscheint nicht in der Darstellung

$$M = 1 + 0,5m_1 + 0,25m_2 + \dots m_N \cdot 2^{-N}$$

Beispiele für 'single precision'

$$\begin{array}{c}
 + \\
 0 \\
 \hline
 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \\
 \hline
 131 \\
 - 127 \\
 \hline
 4
 \end{array}
 \cdot
 \begin{array}{c}
 0.5 \\
 \hline
 1 \ 0 \ \dots \ 0 \\
 \hline
 \end{array}
 = +1.5 \cdot 2^4 = +24$$

hexadezimal: 41 C0 00 00

größte Zahl

$$\begin{array}{c}
 0 \\
 \hline
 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \\
 \hline
 254 \\
 - 127 \\
 \hline
 127
 \end{array}
 \cdot
 \begin{array}{c}
 0.5 \quad \dots \quad 2^{-23} \\
 \hline
 1 \ 1 \ \dots \ 1 \\
 \hline
 = 2^0 - 2^{-23}
 \end{array}
 = (1 + 2^0 - 2^{-23}) \cdot 2^{127} \approx 2^{128} \triangleq 3.4 \cdot 10^{38}$$

hexadezimal: 7F 7F FF FF

kleinste Zahl (betragsmäßig)

$$\begin{array}{c}
 + \\
 0 \\
 \hline
 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \\
 \hline
 1 \\
 - 127 \\
 \hline
 -126
 \end{array}
 \cdot
 \begin{array}{c}
 0.5 \\
 \hline
 0 \ 0 \ \dots \ 0 \\
 \hline
 \end{array}
 = \pm 1 \cdot 2^{-126} \triangleq 1.175 \cdot 10^{-38}$$

hexadezimal: 00 80 00 00
80 80 00 00

'Ausnahmen'

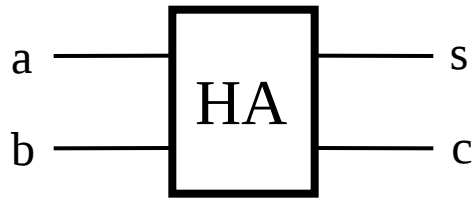
$$\begin{array}{c}
 0 \\
 1
 \end{array}
 \begin{array}{c}
 \text{Exponent} \\
 \hline
 0 \ \dots \ 0 \\
 \hline
 -127
 \end{array}
 \cdot
 \begin{array}{c}
 \text{Mantisse} \\
 \hline
 0 \ \dots \ 0 \\
 \hline
 \end{array}
 \pm 0$$

Hexadezimal: 00 00 00 00
80 00 00 00

$$\begin{array}{c}
 0 \\
 1
 \end{array}
 \begin{array}{c}
 \text{Exponent} \\
 \hline
 1 \ \dots \ 1 \\
 \hline
 \frac{255}{-127} \\
 +128
 \end{array}
 \cdot
 \begin{array}{c}
 \text{Mantisse} \\
 \hline
 0 \ \dots \ 0 \\
 \hline
 \end{array}
 \pm \infty$$

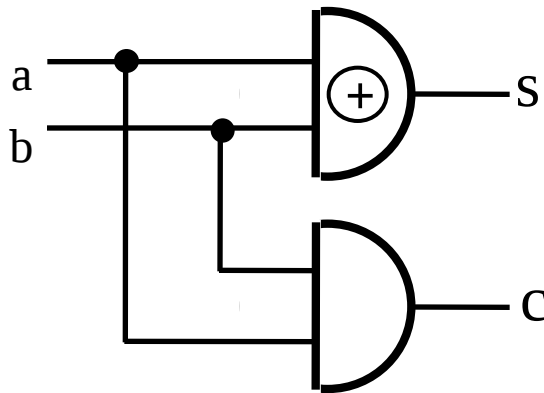
hexadezimal: 7F 80 00 00
FF 80 00 00

Halbaddierer

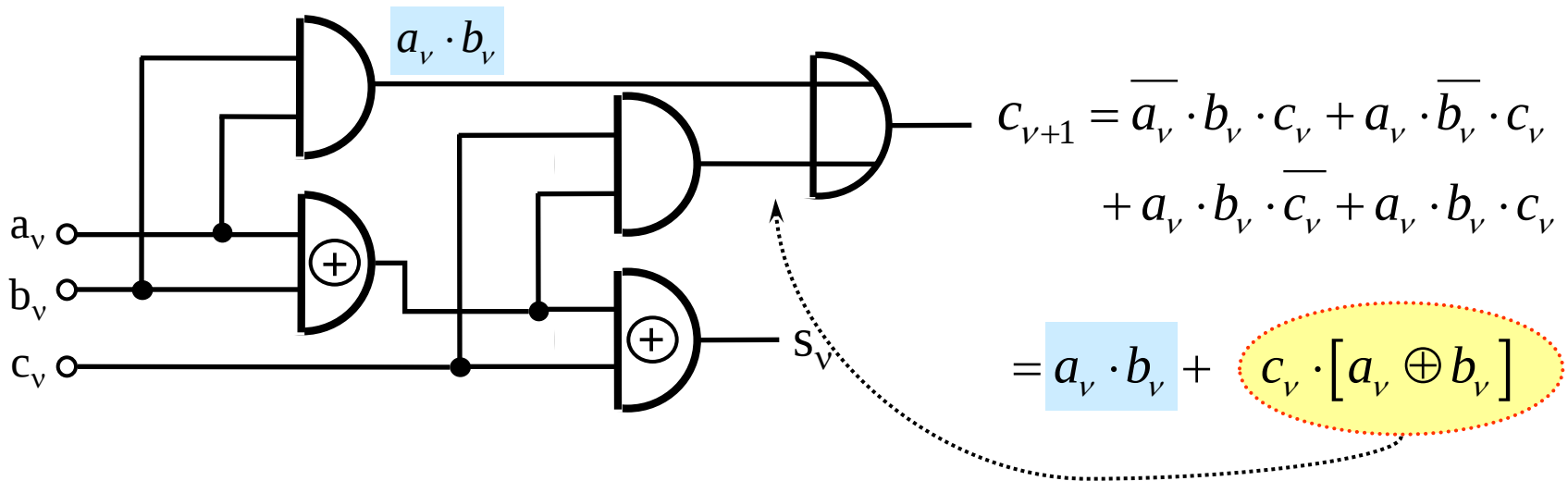
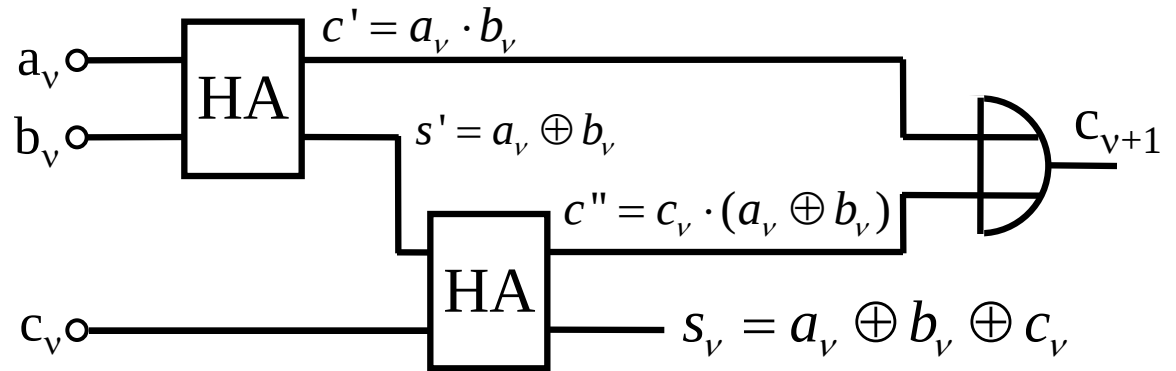
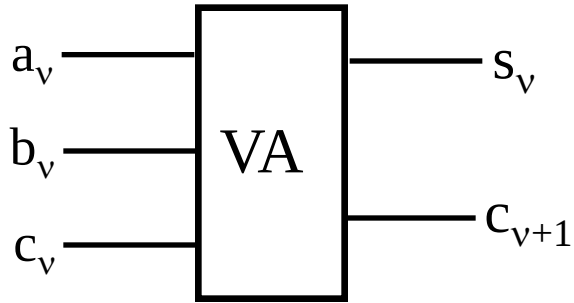


$$s = a \cdot \bar{b} + b \cdot \bar{a} = a \oplus b \Rightarrow \text{EXOR-Funktion}$$

$$c = a \cdot b \Rightarrow \text{UND-Funktion}$$

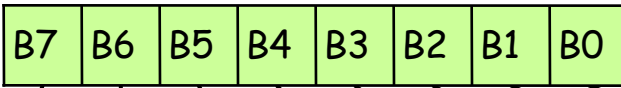


Volladdierer

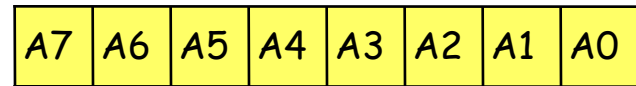


8-bit-Ripple-Carry-Addierer

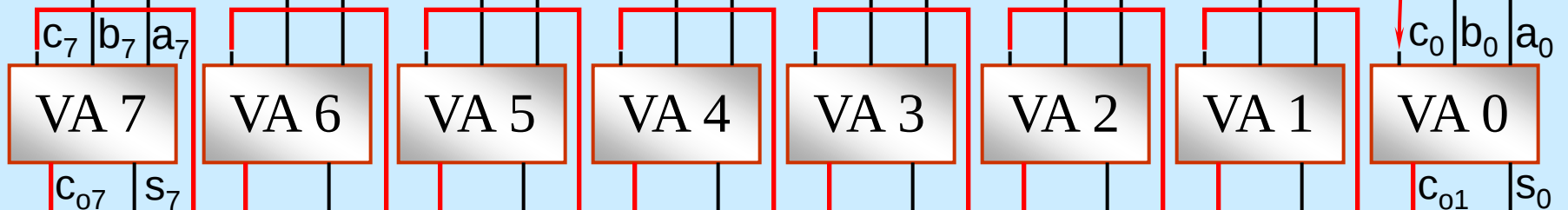
B-Operandenregister



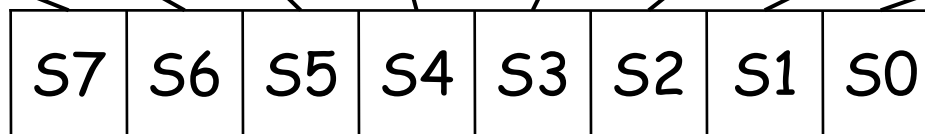
A-Operandenregister



INC

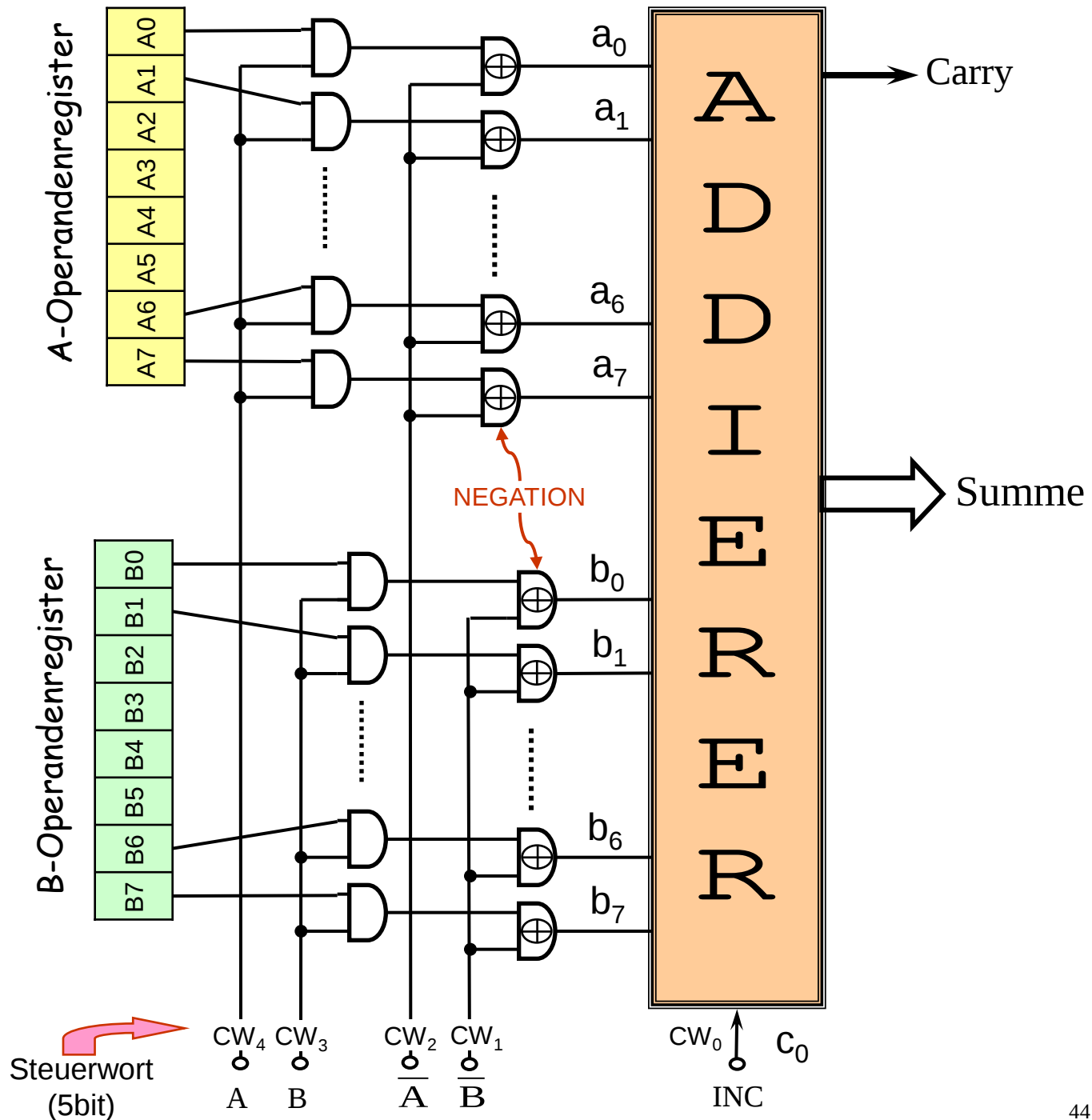


Addierer

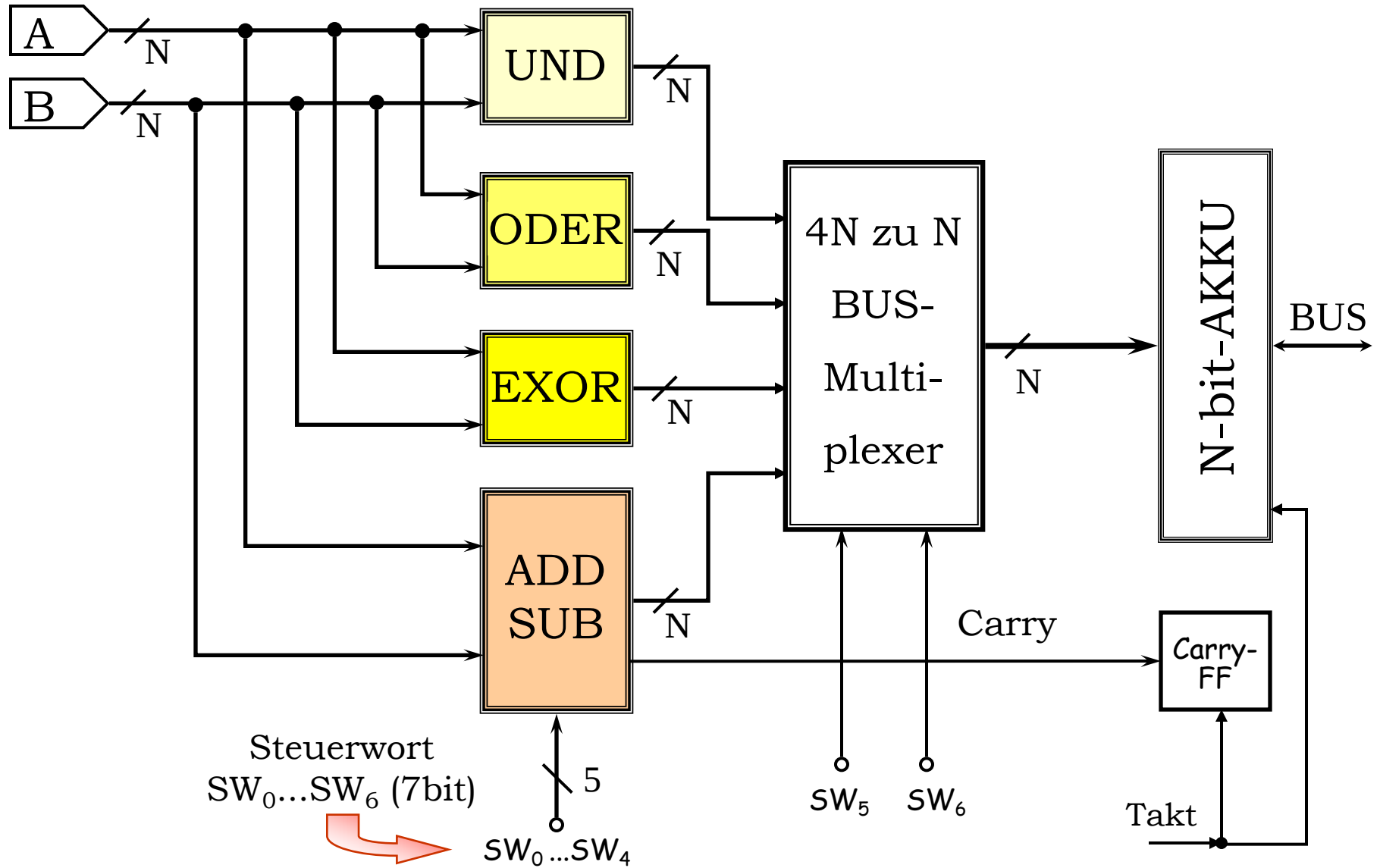


Summenregister

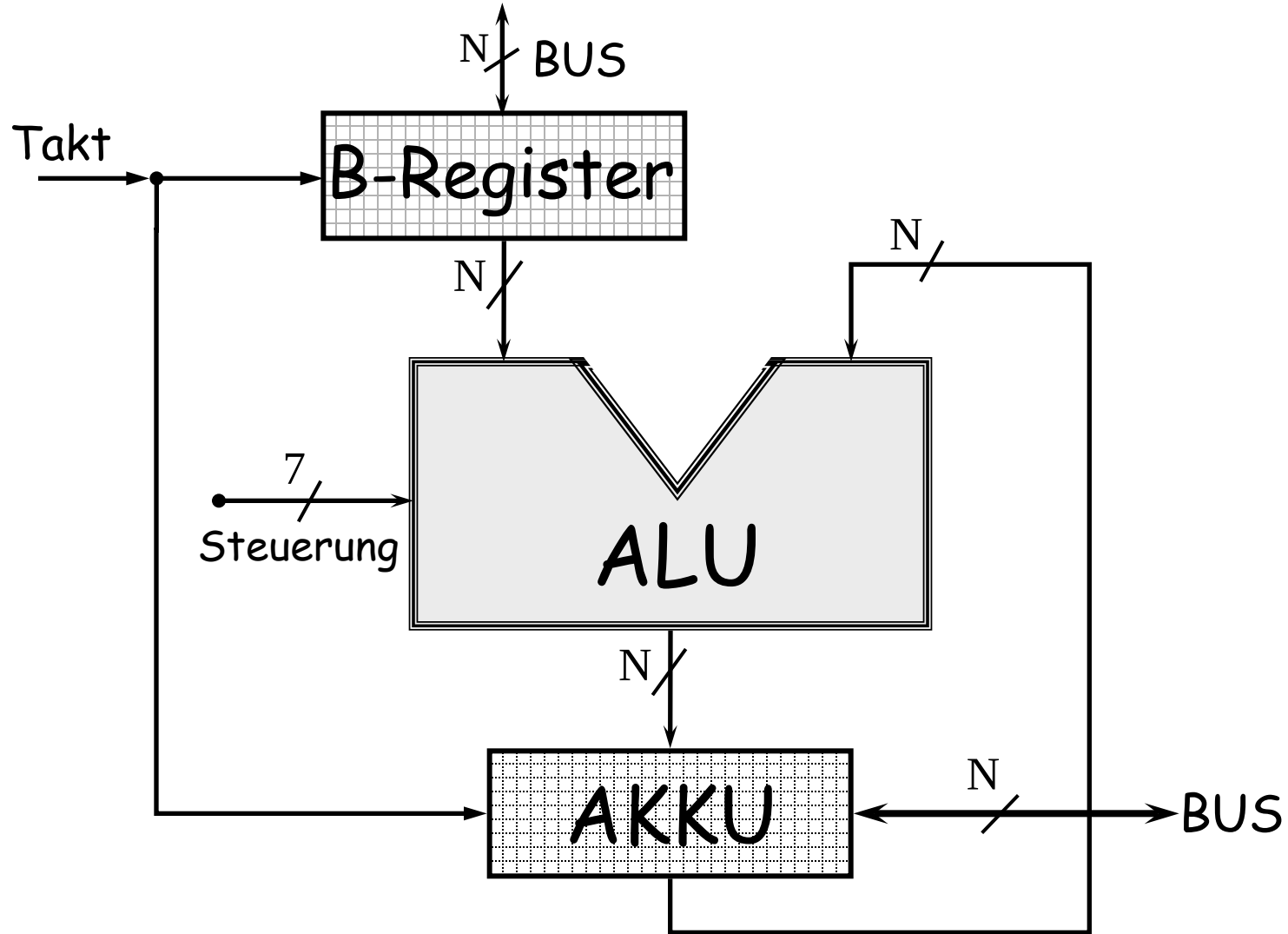
steuerbarer
Addierer/
Subtrahierer



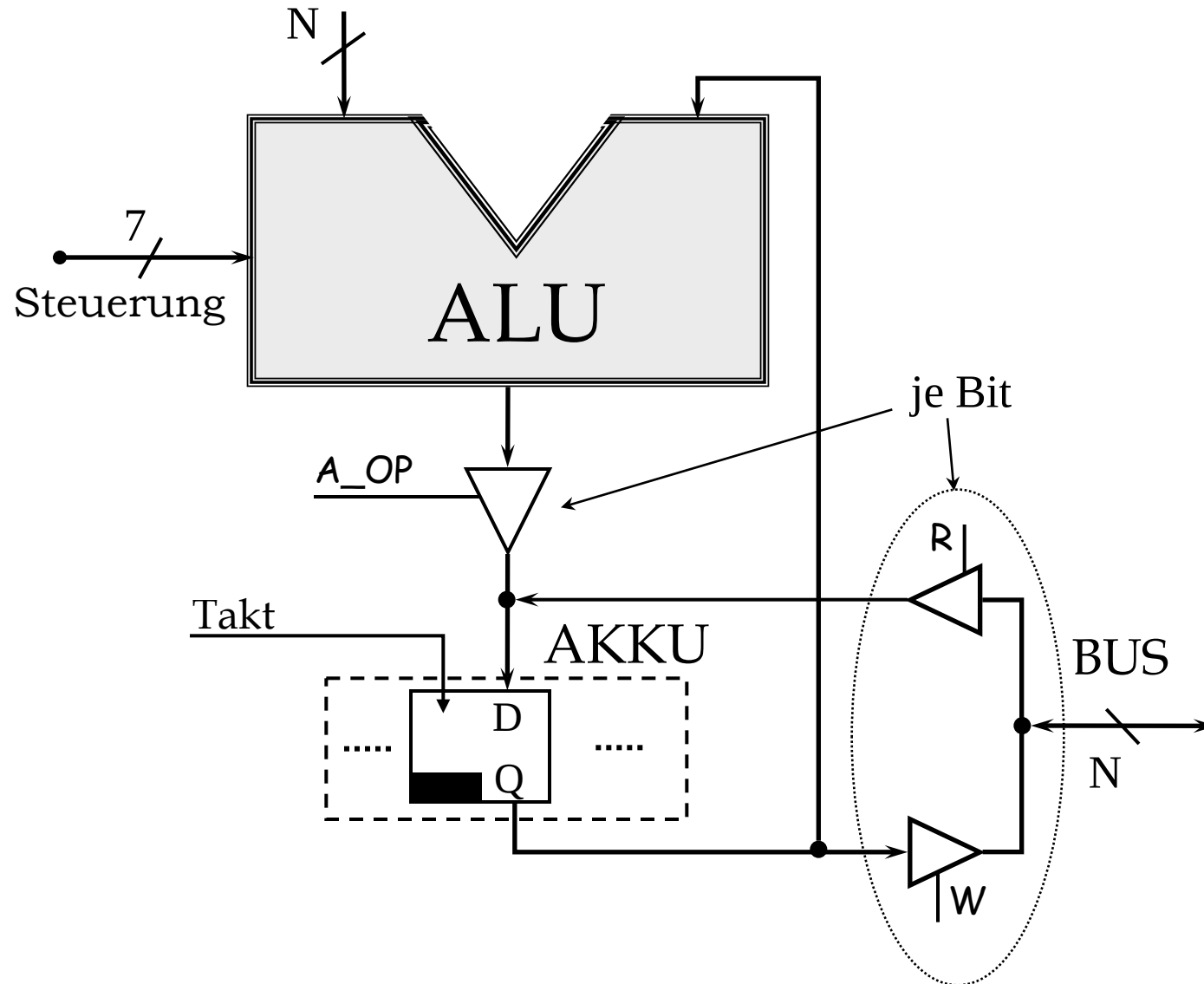
ALU-Aufbau für eine N -bit-Maschine



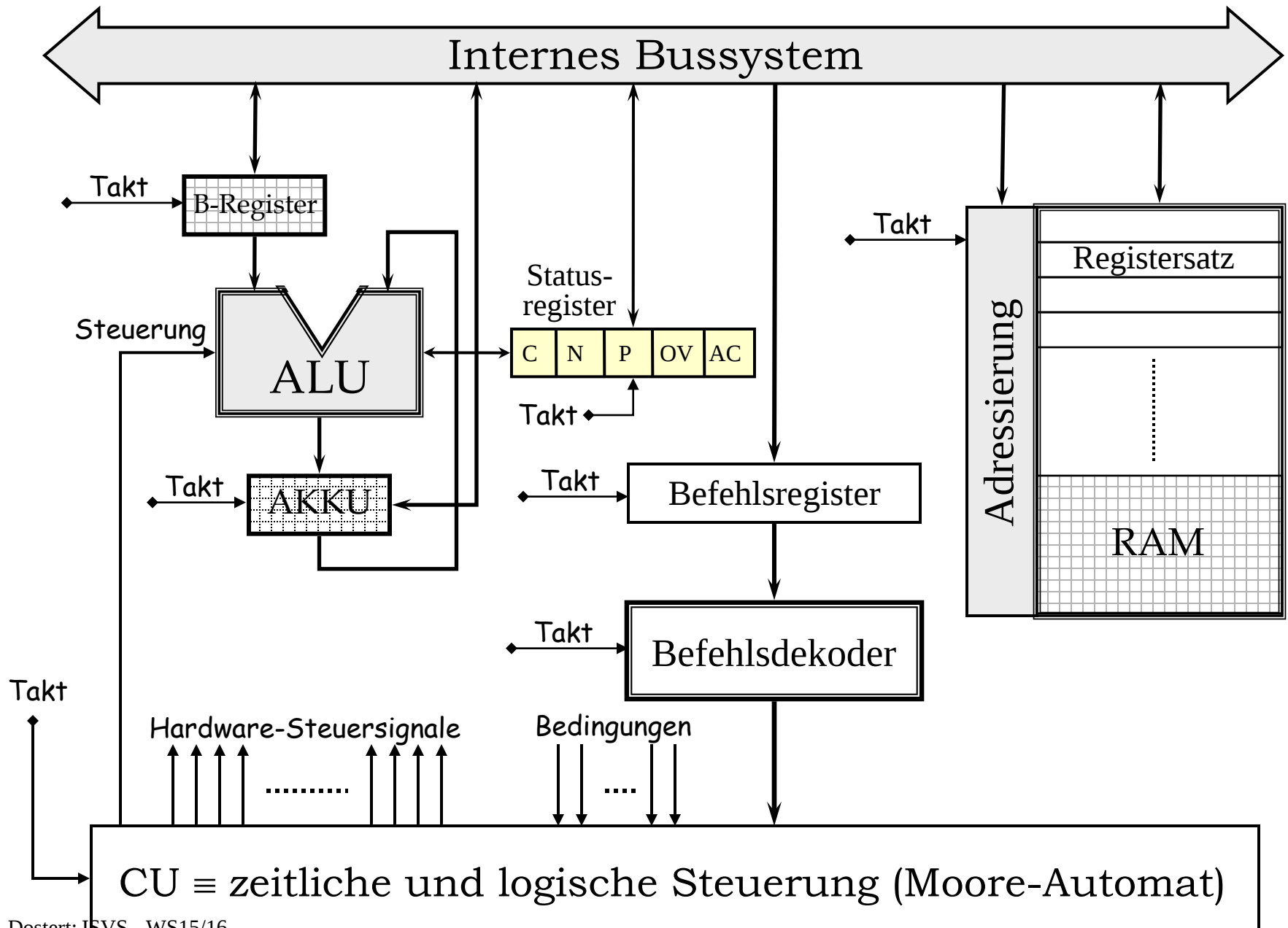
Die ALU im Verbund mit den wichtigsten Arbeitsregistern

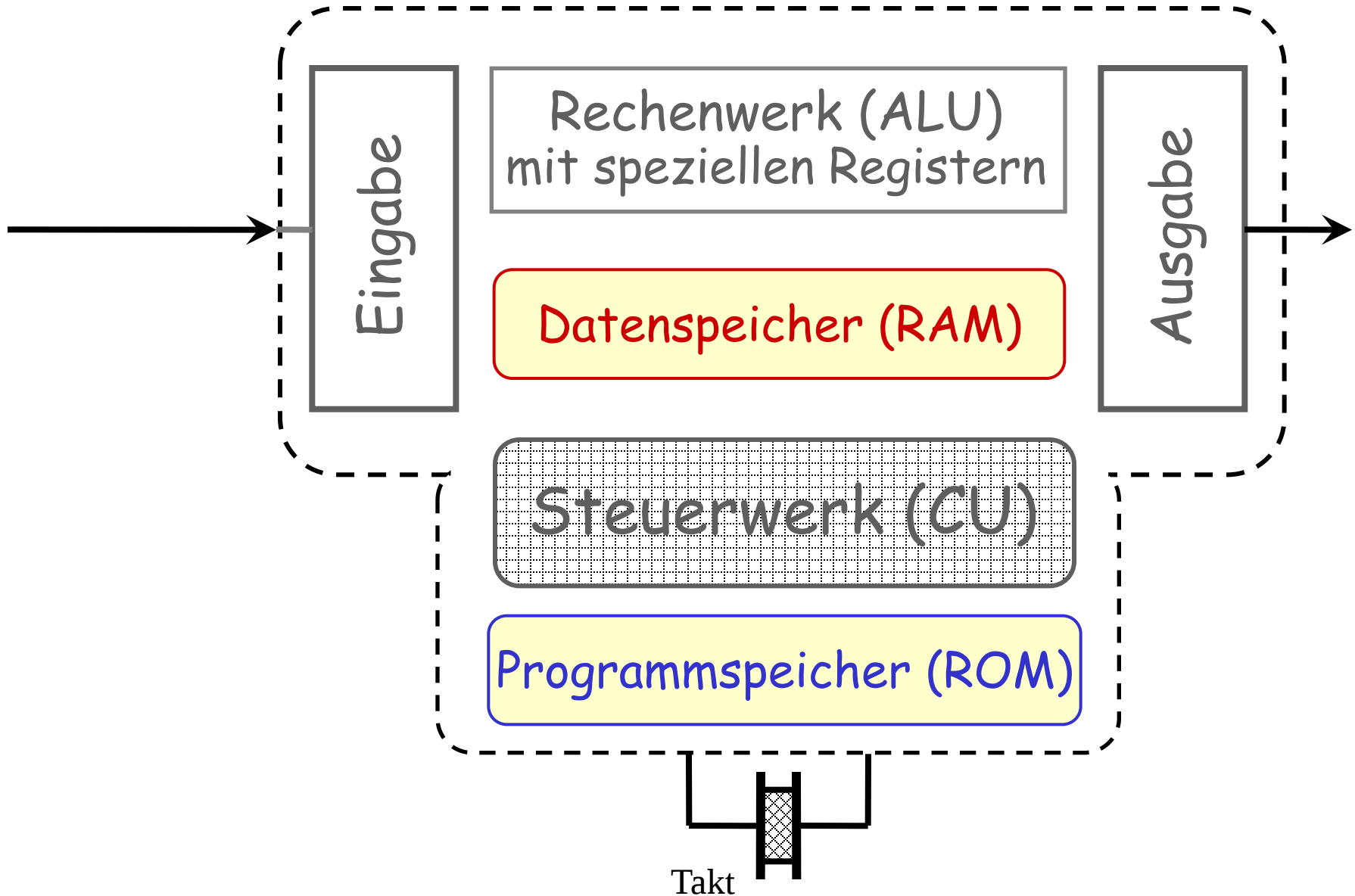


Einzelheiten der Datenflusssteuerung am Akkumulator

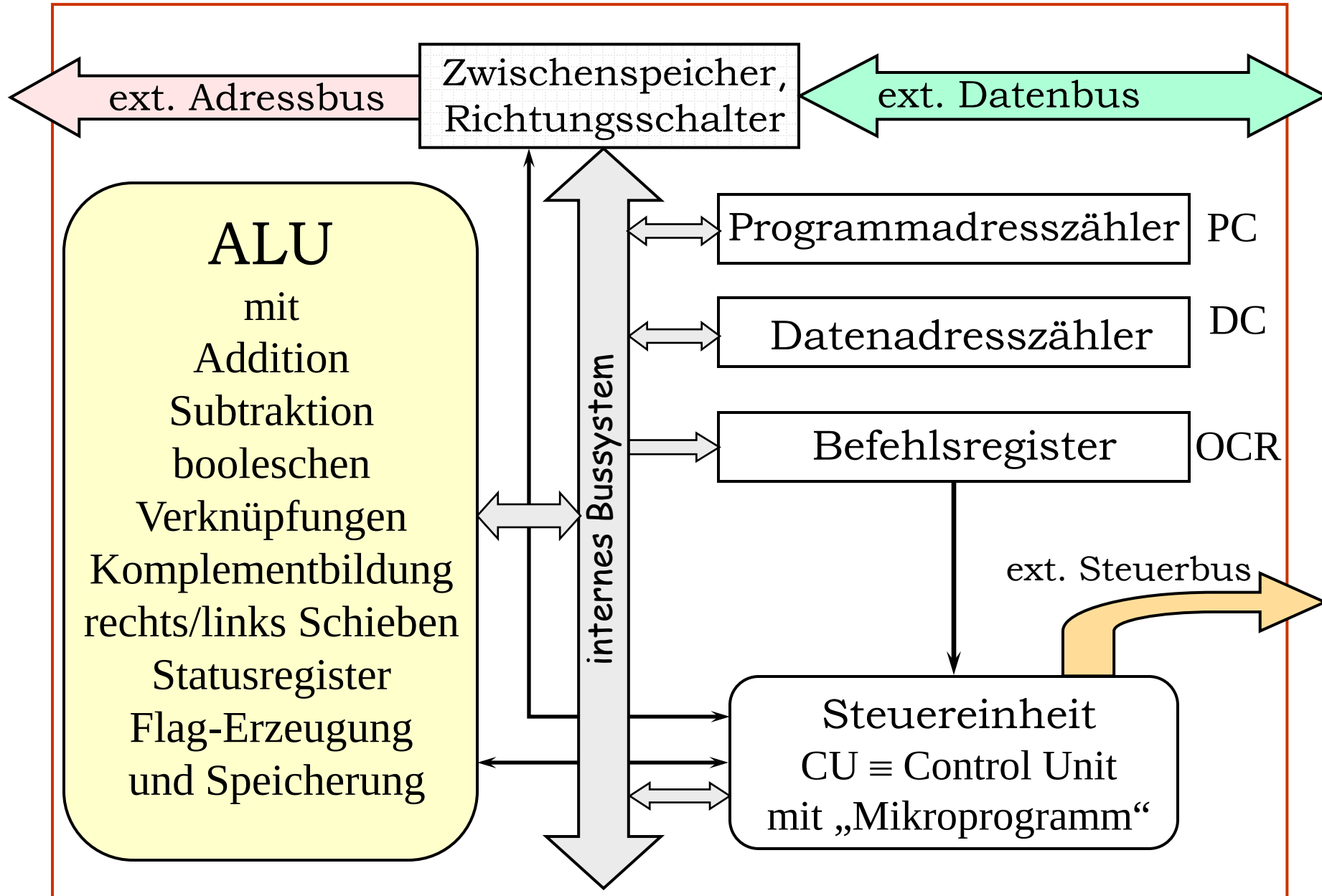


Innerer Grundaufbau eines Mikroprozessors

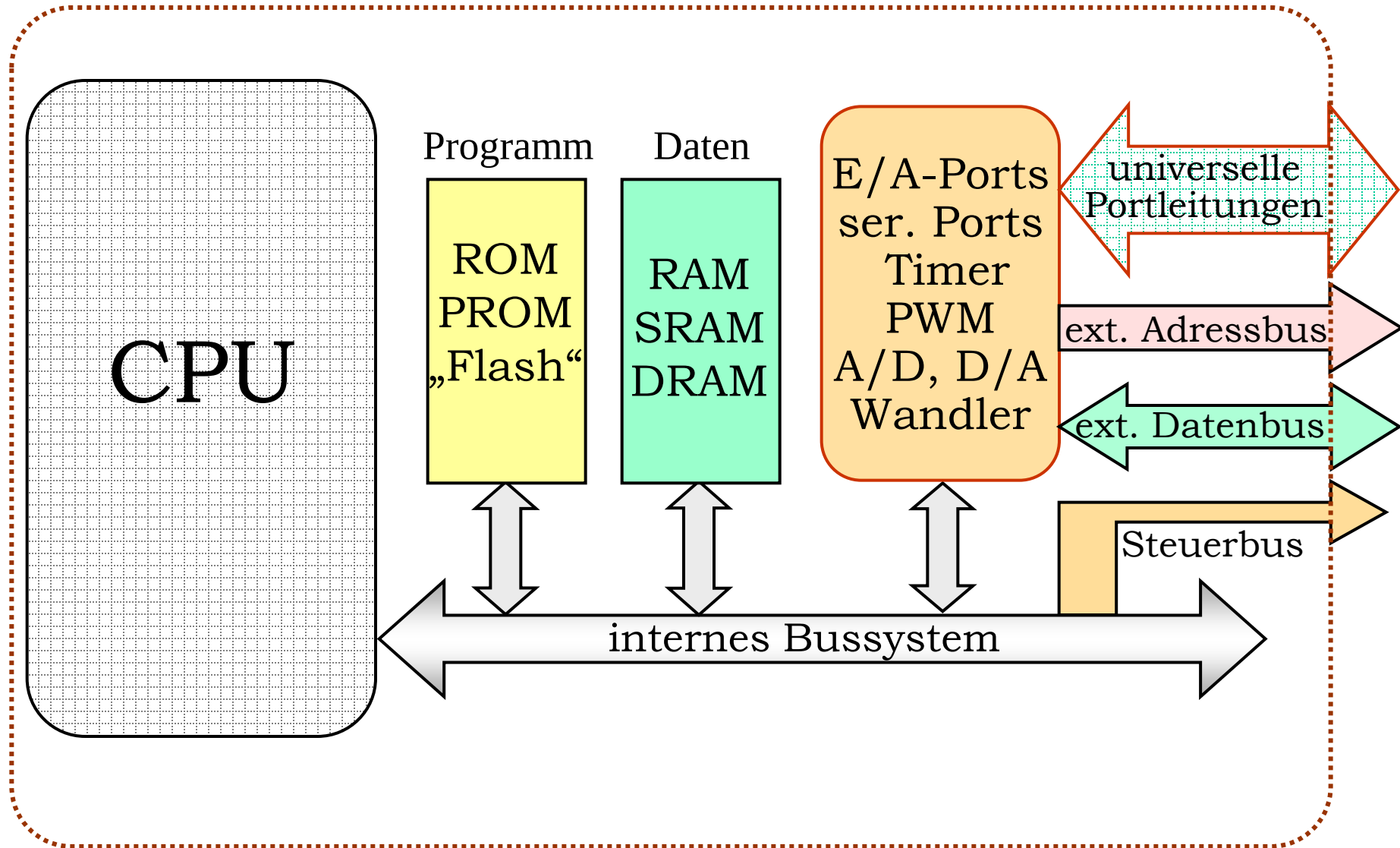




Mikroprozessor (MP oder CPU)



Mikrocontroller (MC)



➤ Festwertspeicher

behalten Information auch ohne Stromversorgung
(meist nur lesbar) **ROM** ≡ **Read Only Memory**

➤ Programmierbare Festwertspeicher

„schnell“ lesbar, löschar und „langsam“ beschreibbar
PROM ≡ **Programmable Read Only Memory**

➤ **EPROM** ≡ **Erasable Programmable Read Only Memory**
Löschen mit UV-Licht

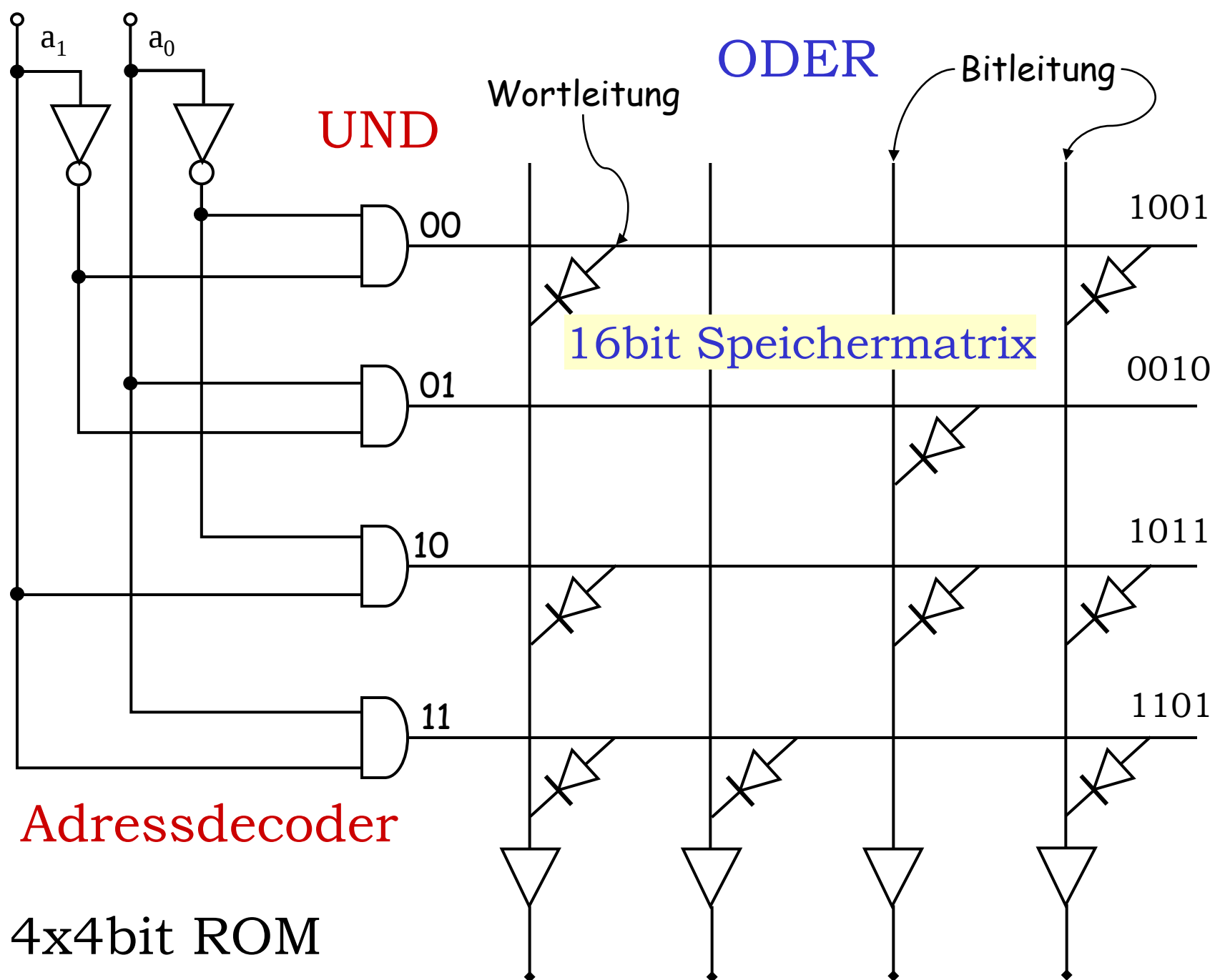
➤ **EEPROM** ≡ **Electrically Erasable Programmable Read Only Memory**
elektrisch löschar- und wiederbeschreibbar

- **FLASH** ≡ **EEPROM** mit besonderer Blockstruktur, die sehr schnelles Löschen ermöglicht

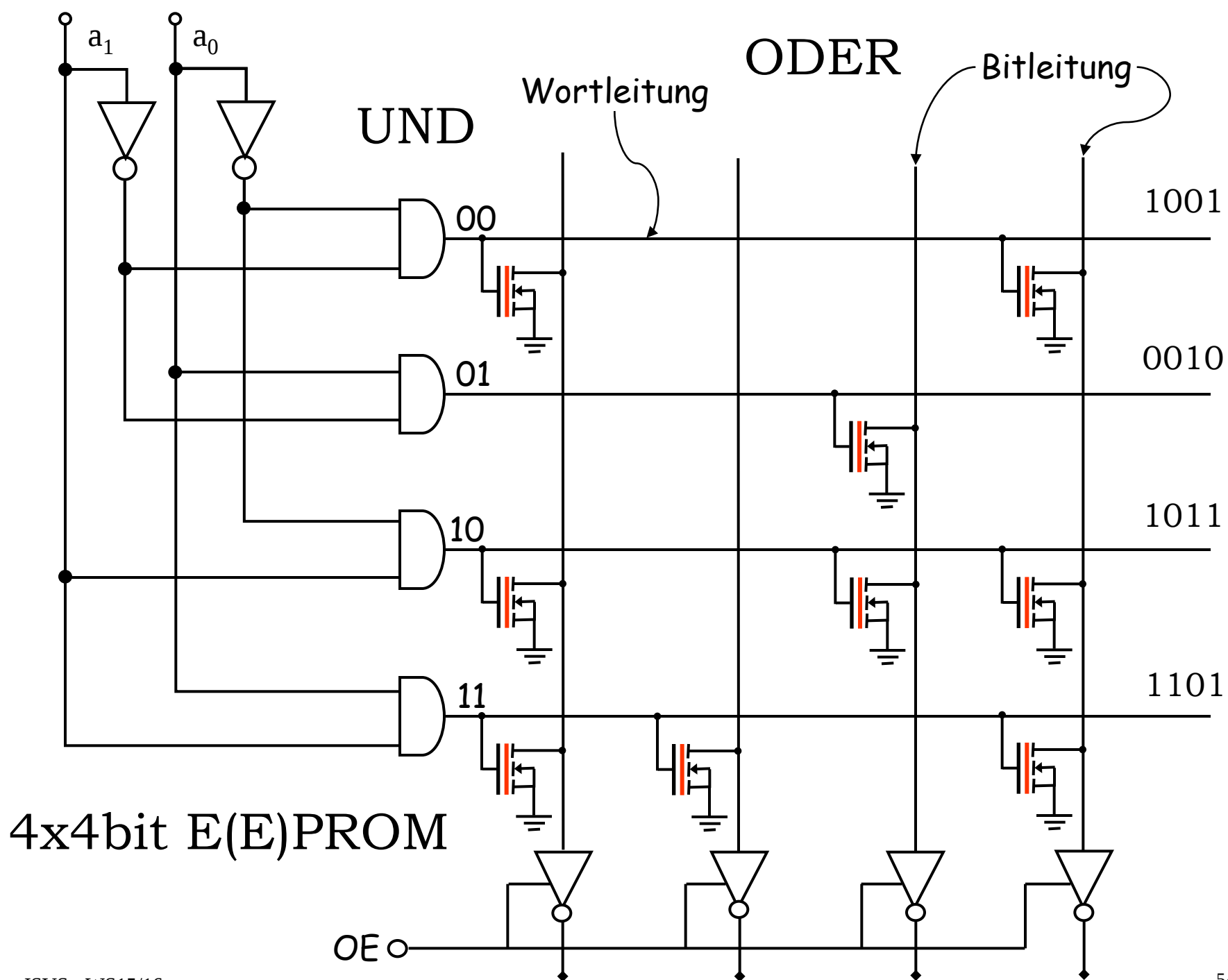
➤ Schreib/Lesespeicher mit wahlfreiem Zugriff

RAM ≡ **Random Access Memory**

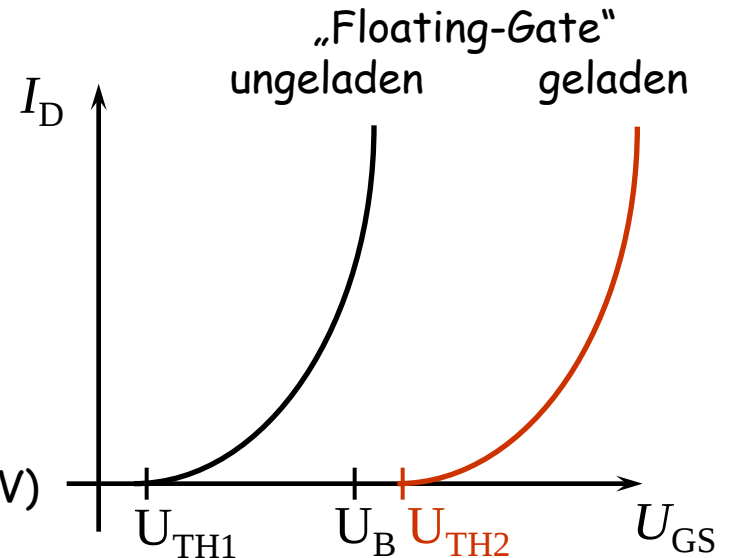
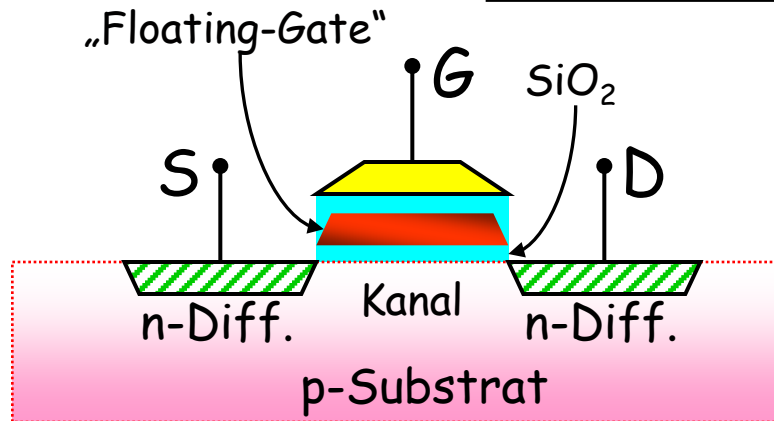
Register, Register-File (Akkumulator) aus Flip-Flops aufgebaut
statisches RAM, dynamisches RAM



4x4bit ROM



EPROM und EEPROM-Prinzip



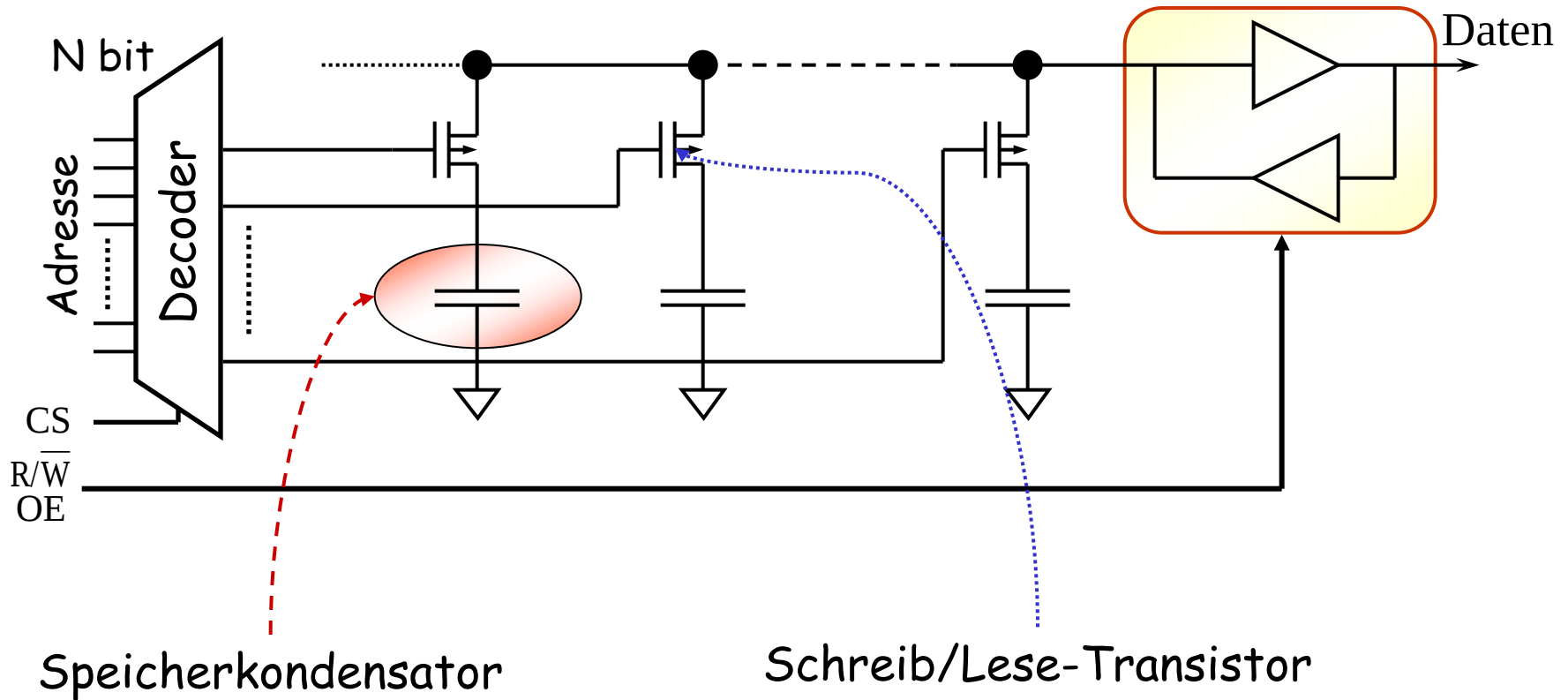
Laden des „Floating-Gate“ entweder mit „heissen“ Elektronen oder durch Tunnel-Effekt (Barriere: 3,25eV)

Entladen: Mittels UV-Licht beim EPROM, durch Gegenfeld beim EEPROM (Flash-Speicher)

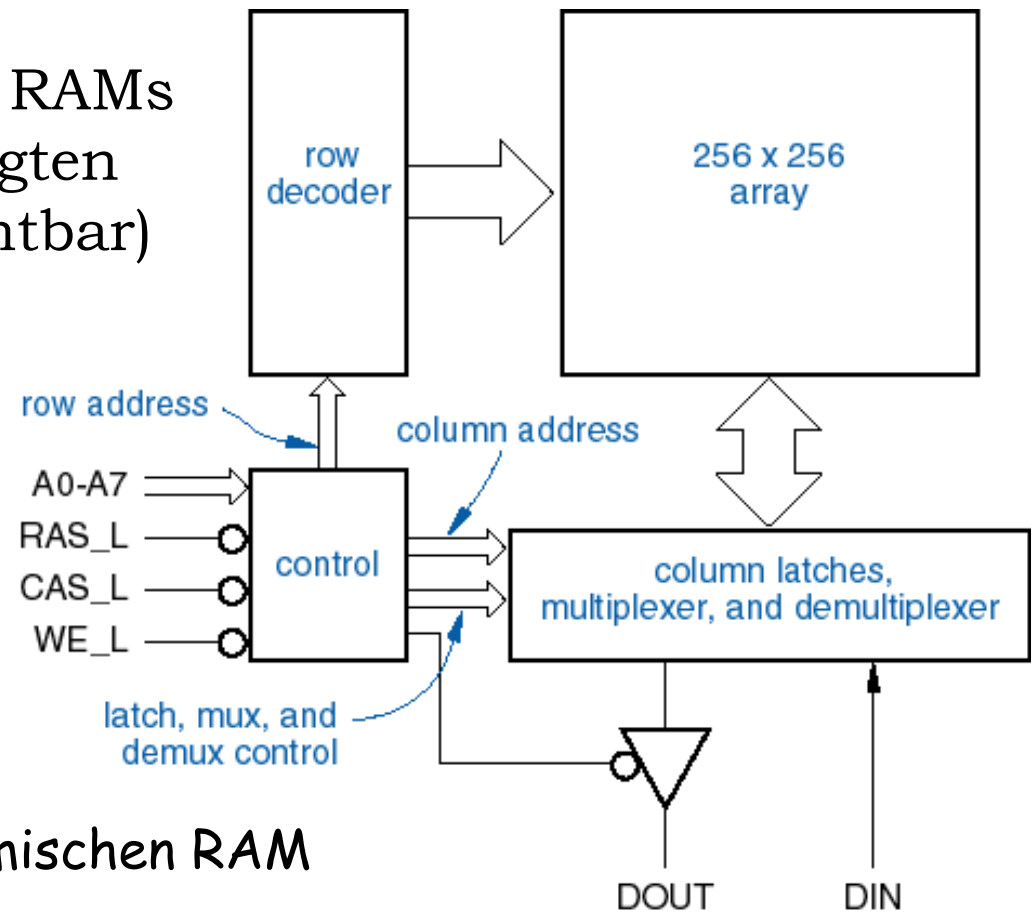
Gate und Floating-Gate bestehen aus Polysilizium und sind in isolierendem SiO₂ eingebettet. Es gibt 2 Mechanismen, durch die Elektronen die Energiebarriere (z.B. 3,5eV) überwinden können:

1. Durch hohe Feldstärke im Kanal erreichen die Elektronen eine genügend hohe kinetische Energie. Im Transistor fließt dabei ein Drain-Source-Strom
2. Bei stromlosem Transistor wird eine hohe Spannung an das Gate angelegt, so daß aufgrund extremer Feldstärke das sogenannte „Fowler-Northeim-Tunneling“ einsetzt, wobei Elektronen den Isolator „durchtunneln“ und auf das Floating-Gate gelangen.

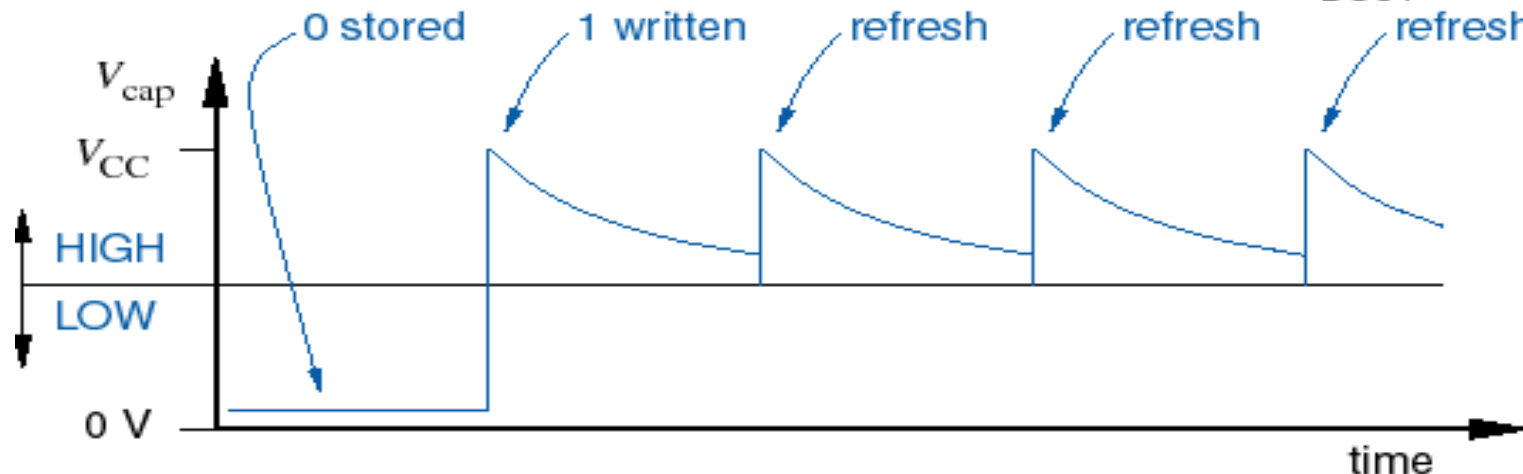
Prinzipaufbau eines dynamischen RAMs



Aufbau eines dynamischen RAMs mit den zum Betrieb benötigten Signalen (Refresh nicht sichtbar)



Refresh-Vorgang beim dynamischen RAM



Die dynamische RAM-Zelle ist rein analog in Form eines Kondensators auf Silizium (z.B. Polysilizium - SiO_2 - Substrat) aufgebaut. Die Kapazität liegt im Bereich von Femtofarad ($1 \text{ fF} = 10^{-15} \text{ F}$)

Beispiel für $C = 1 \text{ fF}$ bei einer Spannung von $3,3 \text{ V}$

$$U = \frac{Q}{C} \quad U_{\text{sw}} = \frac{3,3 \text{ V}}{2} \Rightarrow Q = 10^{-15} \frac{\text{As}}{\text{V}} \cdot 1,65 \text{ V} = 1,65 \cdot 10^{-15} \text{ C}$$

$e = 1,6 \cdot 10^{-19} \text{ C}$ [Elektronenladung in C (Coulomb) oder As]

$\Rightarrow Z \approx 10^4$ Elektronen sind für die Schwellspannung nötig, d.h. 20.000 für einen guten H-Pegel

Beim Lesen wird die Information zerstört und muss deshalb nach Regenerierung sofort wieder eingeschrieben werden.

Heutige DRAM-Zellen müssen nur ca. alle 60ms aufgefrischt werden. Was das physikalisch für die Transistorqualität bedeutet, macht ein Zahlenbeispiel klar:

$$I = \frac{\Delta Q}{\Delta t} = \frac{10^4 \cdot 1,6 \cdot 10^{-19} \text{ As}}{60 \cdot 10^{-3} \text{ s}} = \frac{1,6}{6} \cdot 10^{-13} \text{ A} = 0,027 \text{ pA}$$

➤ **Der Leckstrom muss unter $0,03 \text{ pA}$ bleiben!**

Schreib/Lesespeicher mit wahlfreiem Zugriff

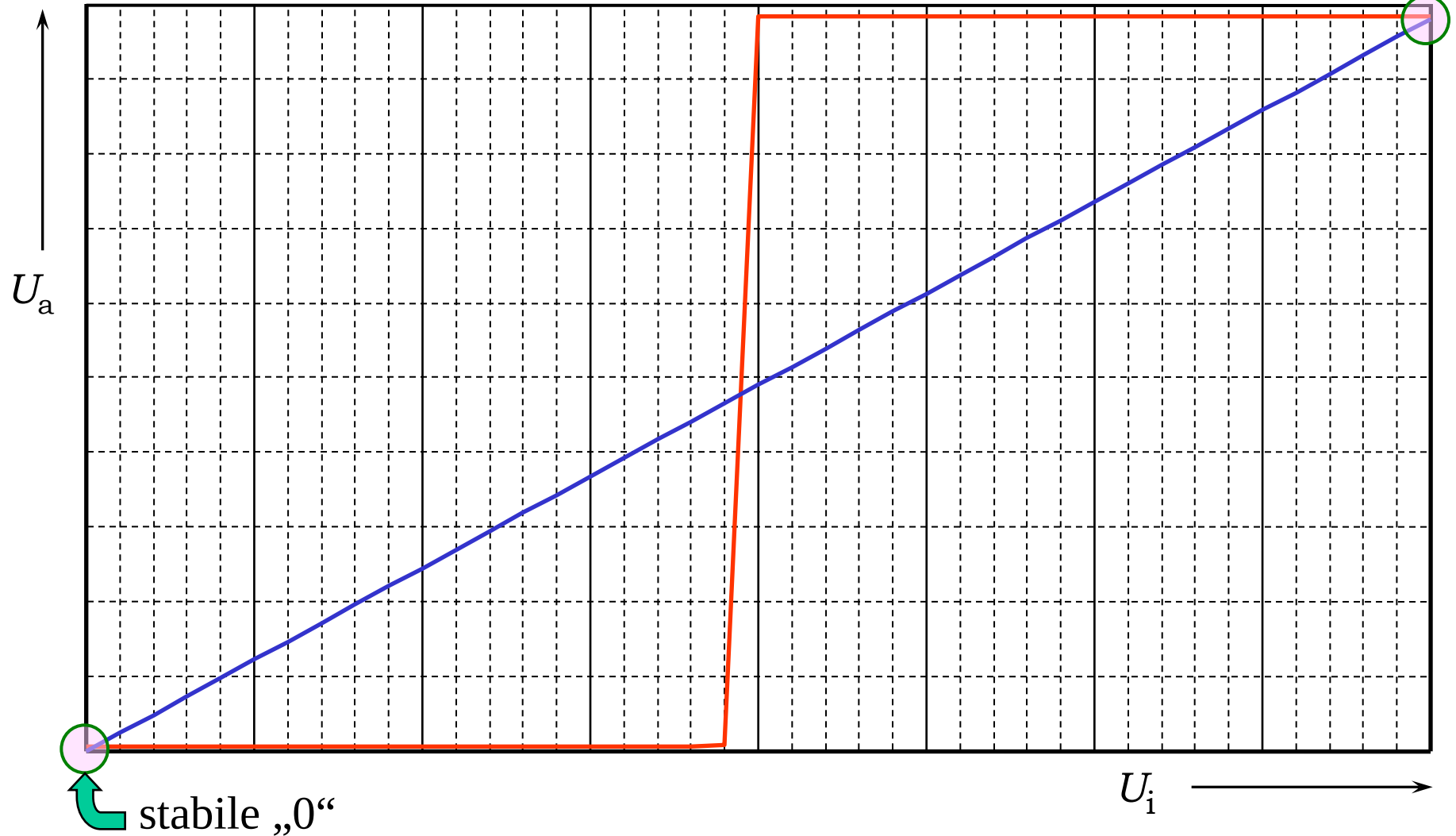
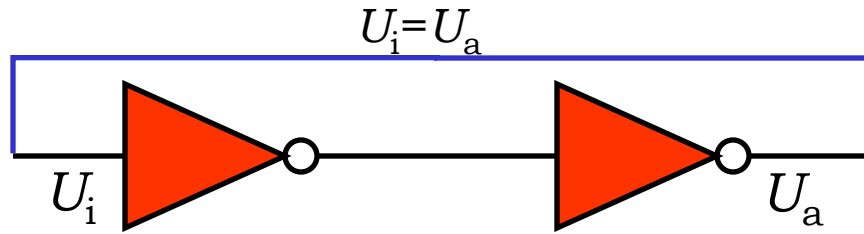
- bilden die Arbeitsspeicher in Mikrorechnersystemen können sehr schnell gelesen und wieder beschrieben werden - einzelne Speicherstellen können geändert werden
- Zahl der Lösch- und Schreibvorgänge ist unbegrenzt
- Der Kernbereich einer CPU (Central Processing Unit Mikroprozessor) verfügt in der Regel über mindestens **8 Register**, die jeweils ein **Maschinenwort** (8, 16, 32bit) oder gelegentlich auch das Doppelte davon fassen können.

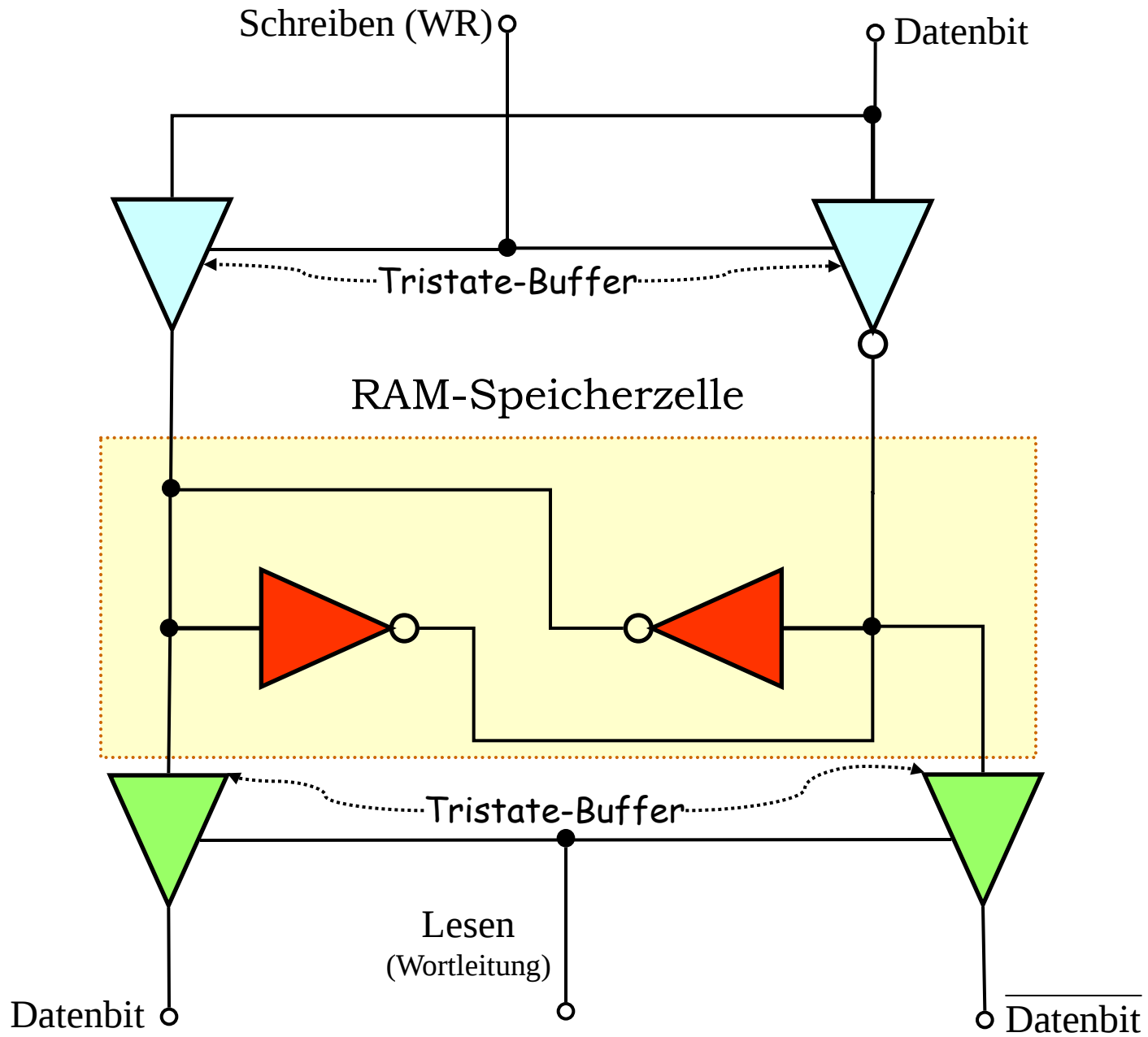
Nahezu alle Datenverarbeitungsvorgänge nutzen Register:

Beispiel für einen Datentransfer **MOV R1, A** ;A nach R1 bringen

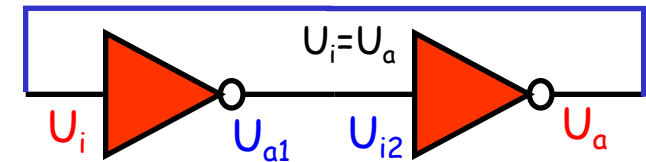
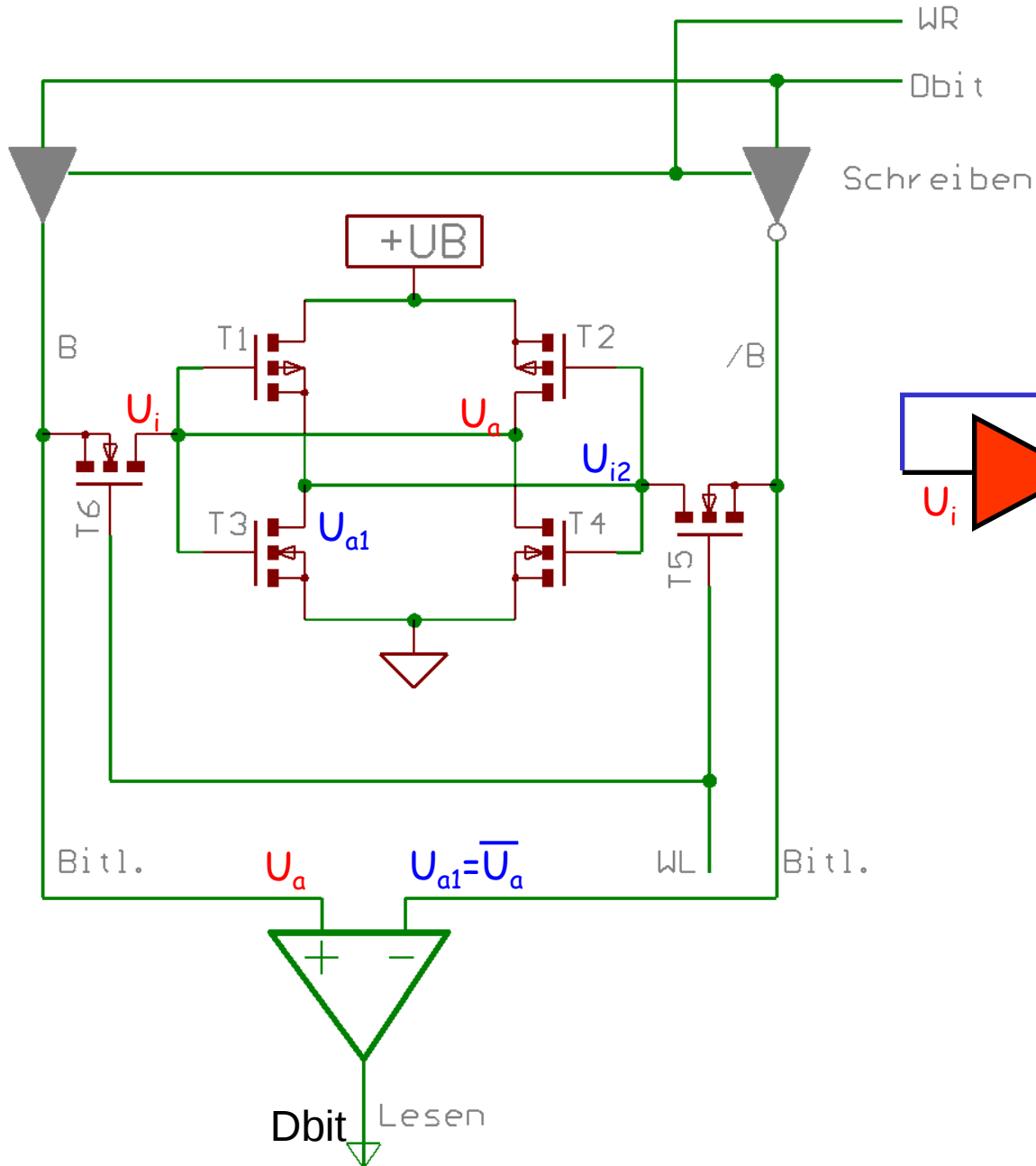
Beispiel für eine Addition **ADD A, R2** ; A:=A+R2

RAM-Speicherzelle



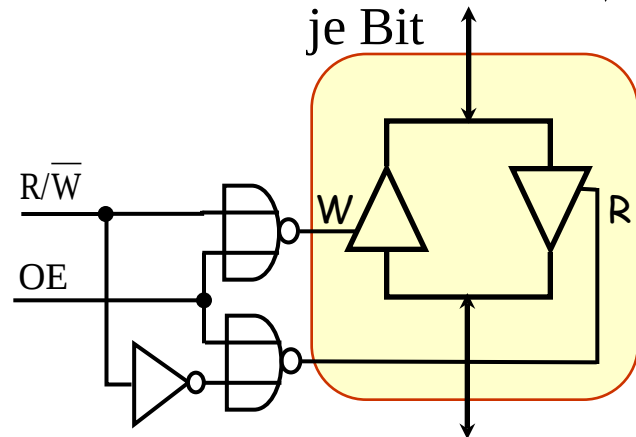
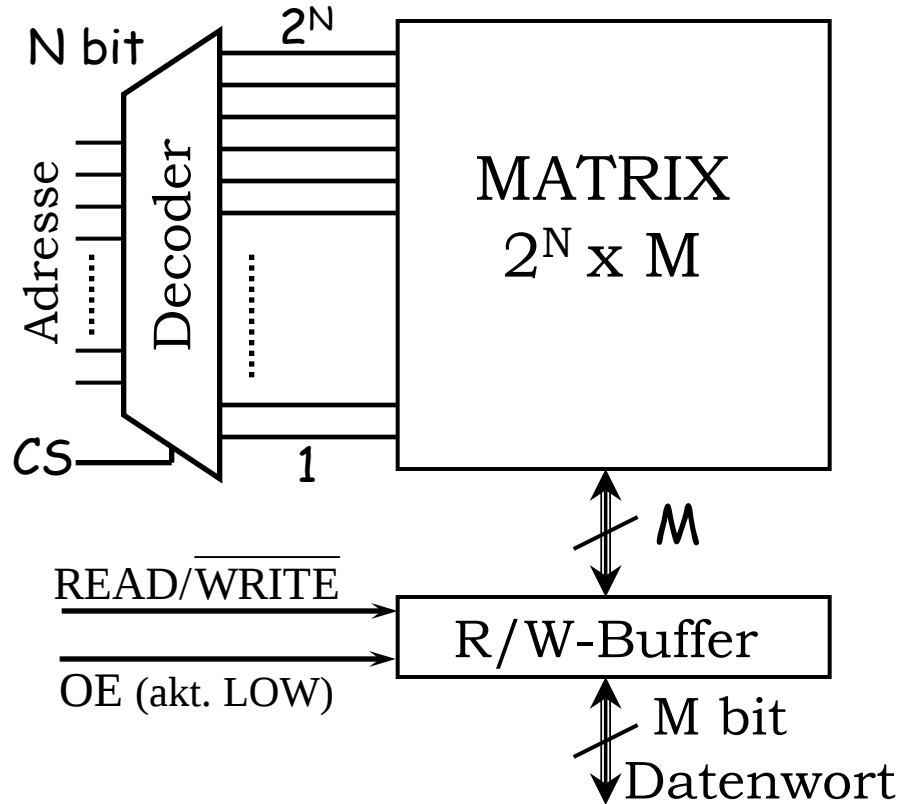


Vollständiger Aufbau einer statischen RAM-Zelle

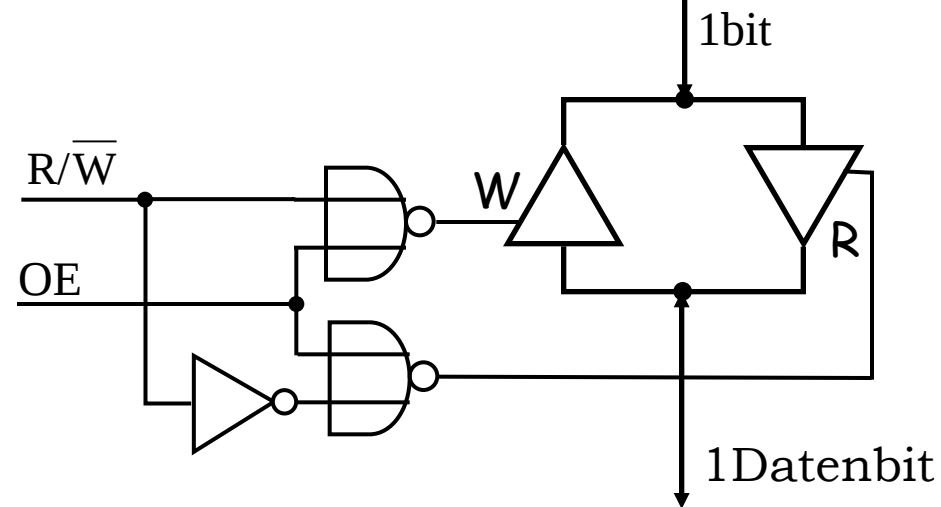
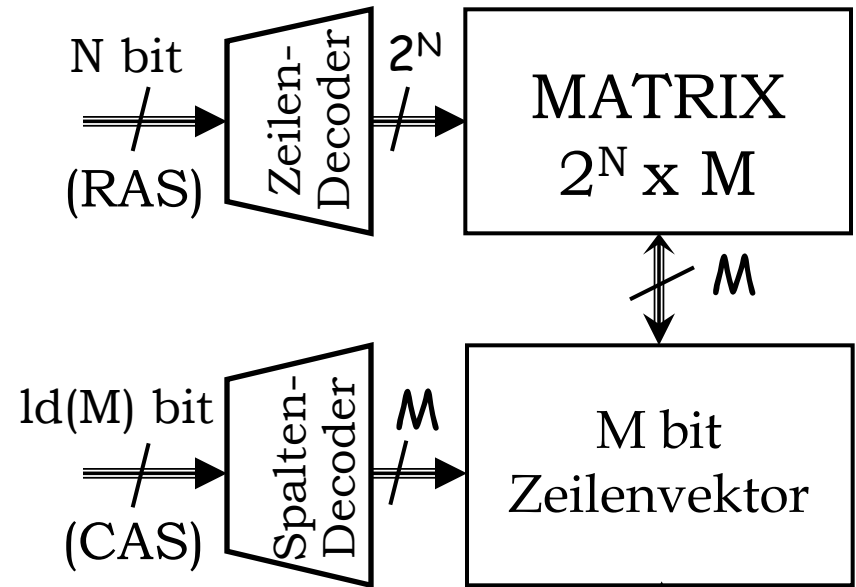


Allgemeine Speicherstrukturen: Wort- bzw. Bitorganisation

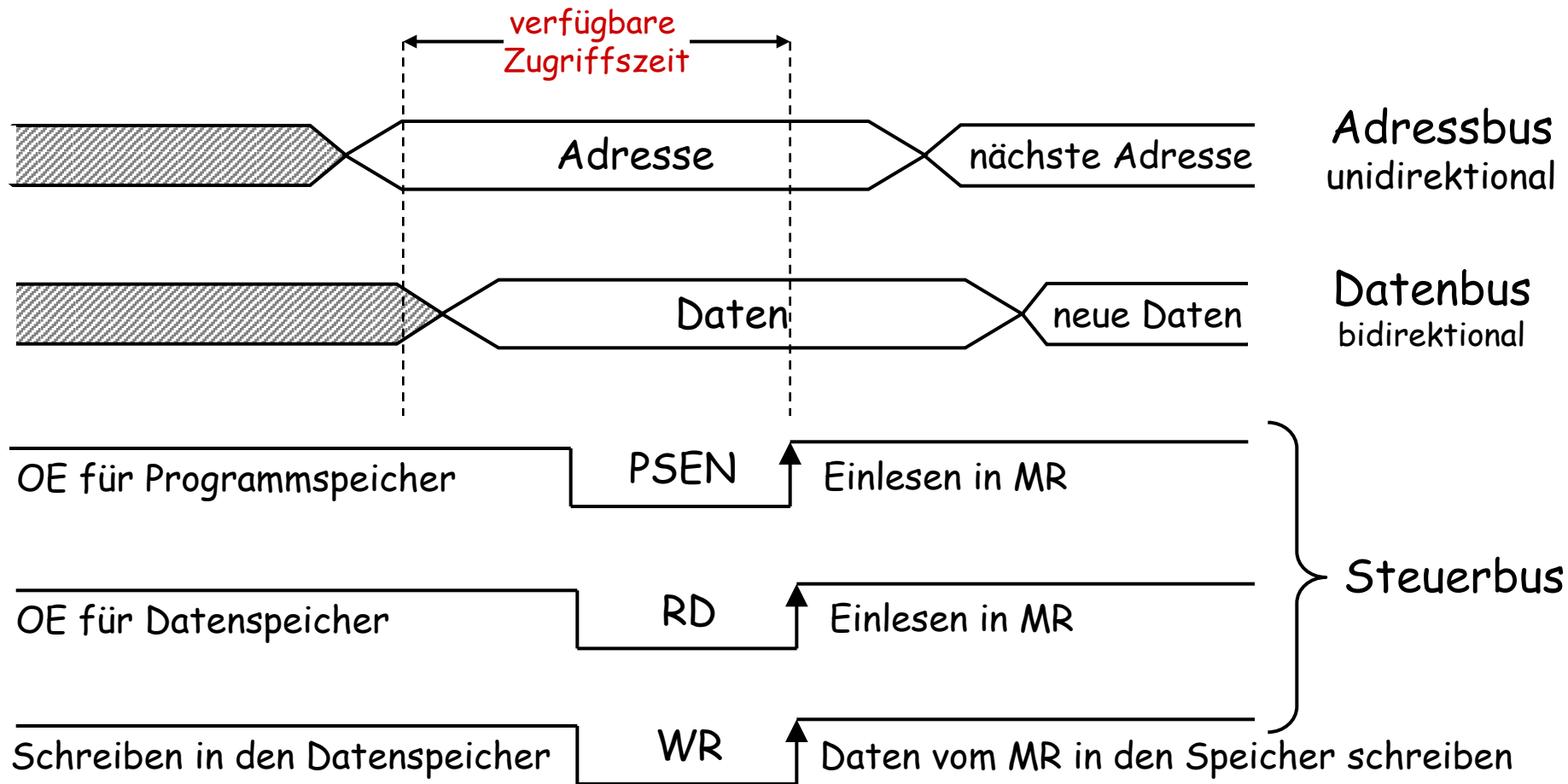
wortorganisiert



bitorganisiert

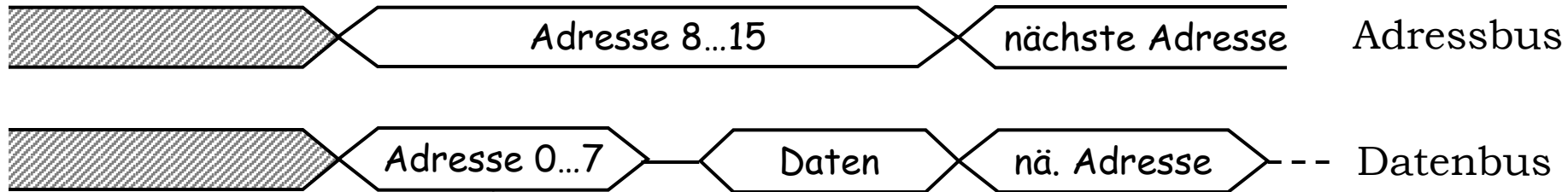


Datentransfer zwischen Speicher und Mikrorechner

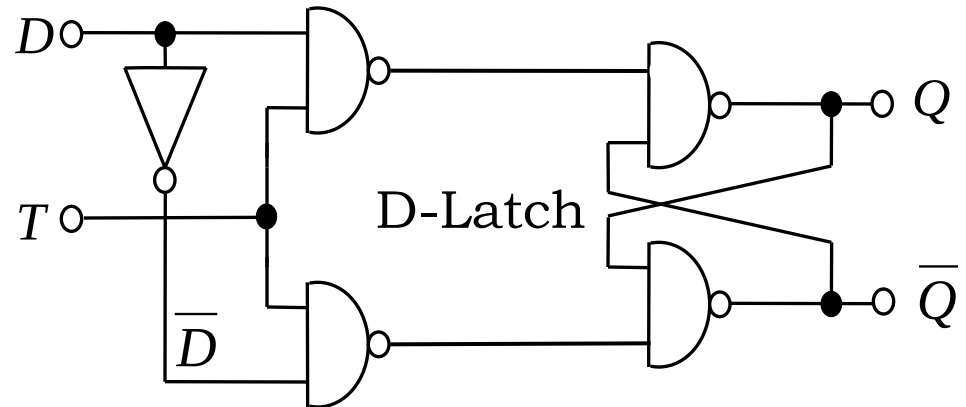
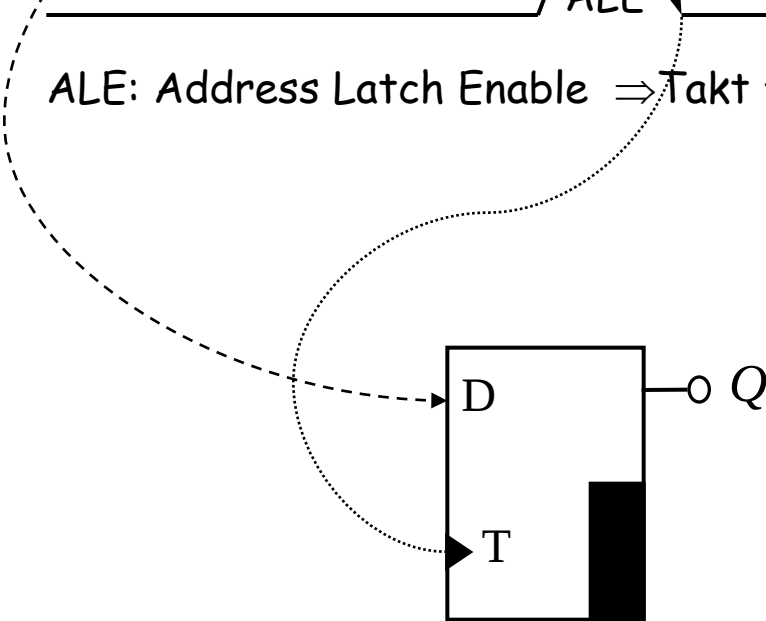


PSEN: Program Storage Enable
OE: Output Enable
RD: Read
WR: Write

Adress- und Datenbusmultiplex bei Mikrocontrollern (spart Leitungen)

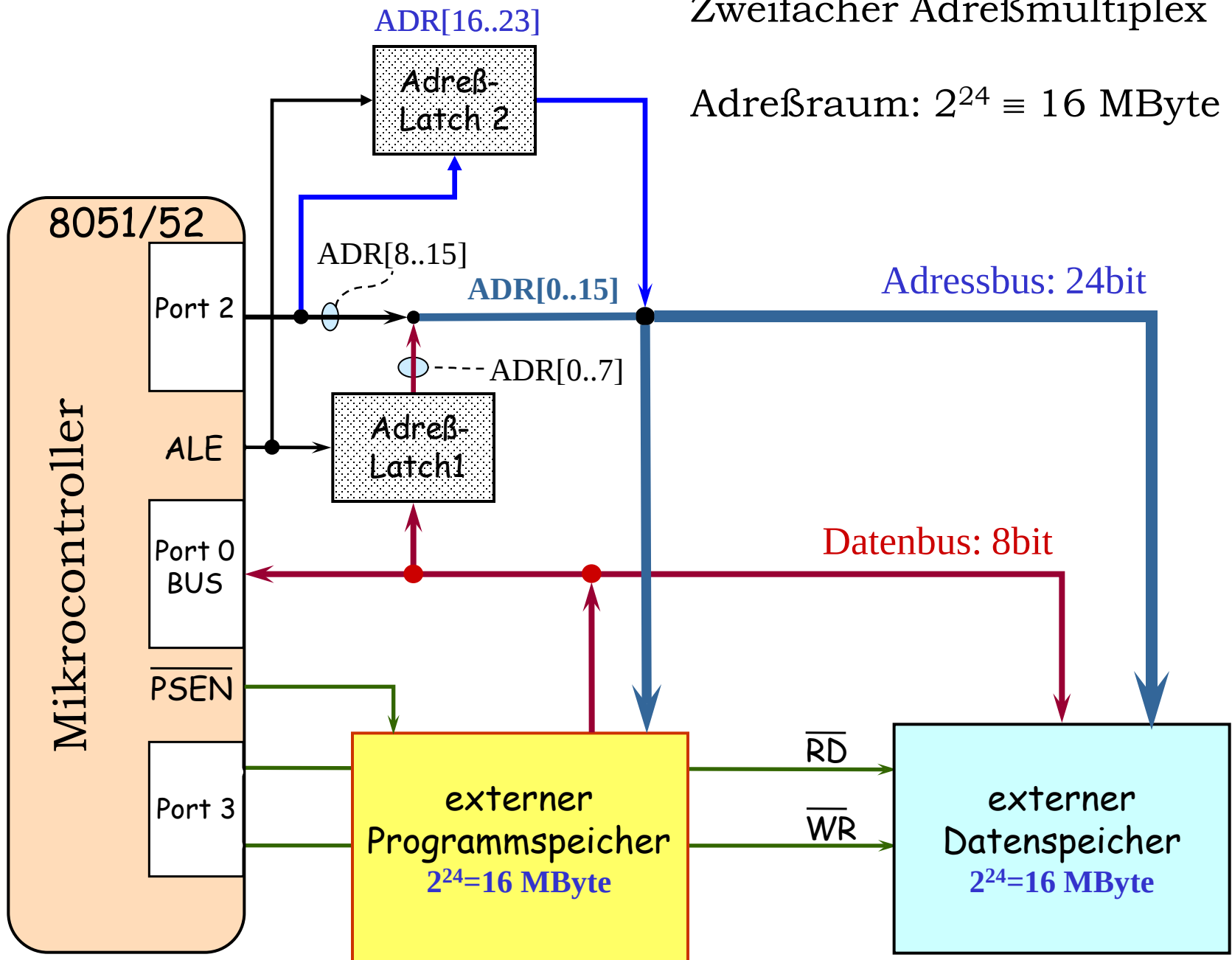


ALE: Address Latch Enable $\Rightarrow$ Takt für Adreßzwischenpeicher (D-Latch)

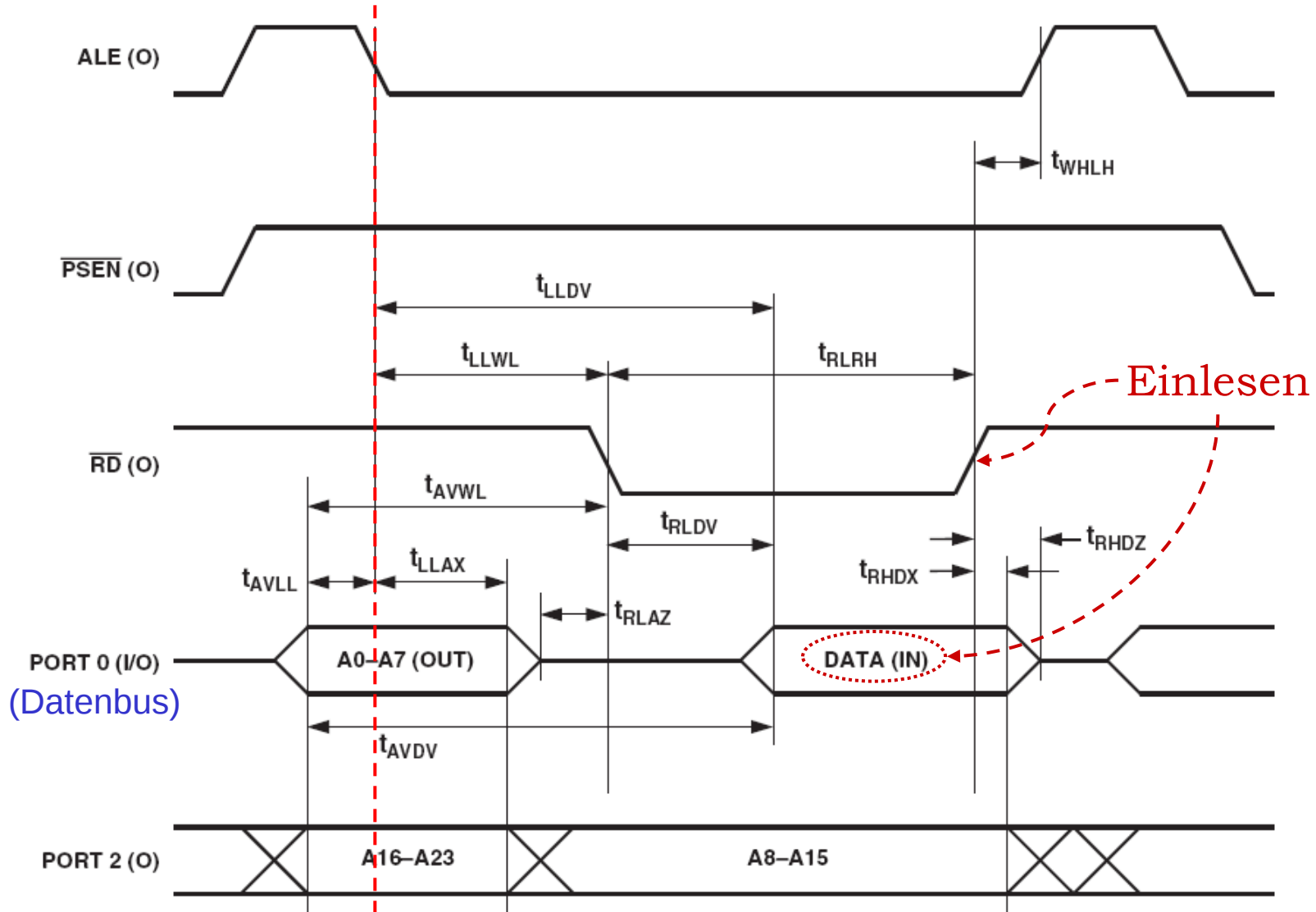


Zweifacher Adreßmultiplex

Adreßraum: $2^{24} \equiv 16 \text{ MByte}$



Lesevorgang aus einem Datenspeicher mit zweifachem Multiplex an Port 0 (Datenbus) und an Port 2



Adressbits 0...7 und 16...23 in Latches speichern