

## Grundlagen der Elektronik für Medienwissenschaftler Teil 4

# Digitale bis Rechen-Technik

Seminar HU; WS 2012/13

Prof. Dr. Horst Völz

Diese Einführung benutzt hauptsächlich:

Völz, H.: Elektronik - Grundlagen - Prinzipien - Zusammenhänge. 5. Aufl. Akademie Verlag, Berlin 1989.

Es ist vollständig als PDF-Datei downloadbar von [horstvoelz.de](http://horstvoelz.de)

Weitere Literatur befindet sich am Ende.

Dieses Material wurde heruntergeladen von [horstvoelz.de](http://horstvoelz.de)

Für privaten Gebrauch ist es frei nutzbar.

Bei Publikationen, Vorträgen usw. ist die Angabe der Quelle notwendig.

Bei kommerzieller Nutzung ist eine Abstimmung mit mir erforderlich.

Die Bilder sind in höherer Qualität ca. 2000×3000 Pixel oder \*.cdr Version 12 bzw. 16 verfügbar.

Prof. Dr. Horst Völz, Koppenstr. 59, 10243 Berlin, Tel./Fax 030 288 617 08

Email: [h.voelz \(at\) online.de](mailto:h.voelz@online.de)

# Zusammenfassung

Der Übergang von kontinuierlich nach digital erzwingt einen verbotenen Signalbereich zwischen 0 und 1. Es wird gefragt, ob er dem verbotenen Dritten der Logik zugeordnet werden kann. Nach Einführung der möglichen Signalkennlinie werden weitere Eigenschaften von kontinuierlich und digital verglichen. Es folgt die Einführung der kombinatorischen Schaltungen, ihrer Grenzen, Basis-Systeme und technischen Varianten. Als komplexe Bausteine werden u. a. Coder, Decoder, Multiplexer und Demultiplexer sowie PLA, PROM und PAL vorgestellt. Die Einteilung der digitalen Bausteine führt zur Dreiteilung in kombinatorische, sequentielle Bausteine und Speicher. Die große Vielfalt der Speicher erzwingt selbst bei starker Einschränkung einen ziemlich umfangreichen Teil. Es folgen wichtige sequentielle Schaltungen einschließlich der verschiedenen Automaten- und Rechner-Strukturen. Als ergänzende Systeme werden nur kurz AD-Wandler, Kompression, Fehlerbehandlung und Kryptografie behandelt. Abschließend erfolgt eine sehr kurze Behandlung der allgemeinen Grundlagen. Sie beginnt mit den Widersprüchen der Logik, geht über Axiomatik, GOEDEL-Theorem, TURING-Automat und CHURCH-These und endet bei N- und NP-Problemen.

# Gliederung

## 1. Grundlagen

2. Kombinatorische Bausteine

3. Die drei typischen Digitalerschaltungen

4. Speicher

5. Sequentielle Schaltungen

6. Automaten und Rechner

7. Ergänzende Systeme

8. Berechenbarkeit

# Kontinuierlich $\Leftrightarrow$ digital

Fast immer fallen *zunächst kontinuierliche Signale* an.

Sie sind stets mit *Störungen* überlagert (s. Teil 3).

Der Übergang zu digitalen Signalen erfordert daher *Toleranzbereiche*.

Sie sind nebenstehend für die üblichen TTL-Pegel dargestellt.

Aus einem kontinuierlichen Pegel folgt daher *nicht entweder 0 oder 1*.

Es muss vielmehr immer einen zusätzlichen *dont-care-Bereich* geben.

Das widerspricht dem „*ausgeschlossenen Dritten*“ der Logik (s. Ende).

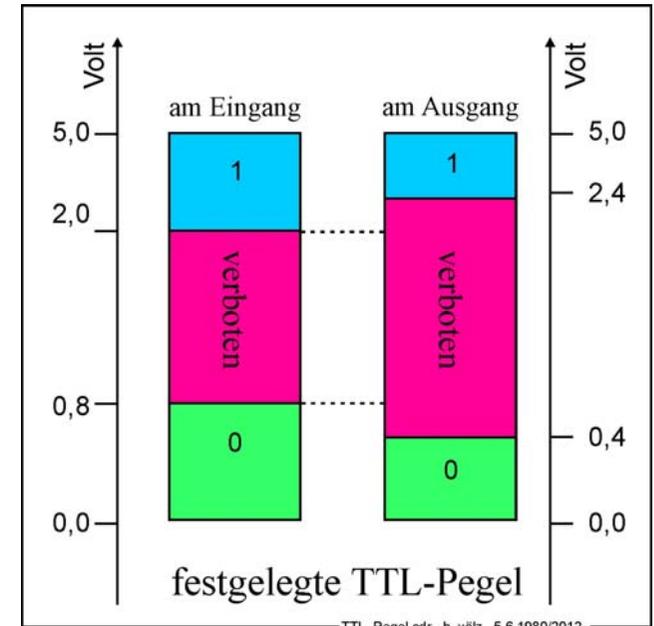
Er ist aber *technisch unbedingt notwendig!*

Dazu gehören auch die typischen *technischen Kennlinien* (s. u.).

Der exakt schaltende Übergang in Spalte 4 ist technisch jedoch nicht möglich.

Doch zunächst müssen einige allgemeine *Grundlagen* behandelt werden.

Erst dann ist der Übergang zur *digitalen Elektronik und Rechentechnik* möglich.



# Typische Kennlinien

Der Übergang in Spalte 4 ist nur theoretisch wichtig, technisch ist er wegen Störungen nie exakt realisierbar.

		kontinuierlich			diskret, digital		
		Toleranzbereich					
		linear	nichtlinear	hysteretisch	schaltend	speichernd	
passiv		klassische Naturbeschreibung			Magnetband	flüssig ↔ fest (Taste)	Ferritkern Kippschalter
aktiv		Verstärker				Relais	Telefonrelais

Klassifikation.cdr h. vözl 9.2.11

# Vergleiche

Der Vergleich von kontinuierlich und digital betrifft hauptsächlich zwei Aspekte.

1. die mathematischen, theoretischen *Grundlagen* und
2. deren technische Realisierung durch (elektronische) *Bauelemente* und *Schaltungen*.

	<b>kontinuierlich</b>	<b>digital</b>
<b>Bauelemente Schaltungen</b>	R, L, C, Verstärker, Oszillatoren usw.	logische Elemente, Speicher, Automaten usw.
<b>mathematische Funktionen</b>	Grundlegend: +, ×, -, /; Verknüpfung Beispiele: $x^2$ , $y^x$ , $y = f(x)$	YES, NOT, AND, OR, XOR, NAND, NOR; logisch: $\wedge$ , $\vee$ , $\neg$ , Mengen: $\cap$ , $\cup$

Für die wichtigsten digitalen Bezüge gilt außerdem (Eselsbrücke für  $\vee$  lateinisch vel):

<b>Mengenlehre</b>		<b>formale Logik</b>		<b>BOOLE-sche Algebra</b>	
verbal	Symbol	verbal	Symbol	Wert	Benennung
Vereinigung	$\cup$	oder	$\vee$	OR	Disjunktion
Durchschnitt	$\cap$	und	$\wedge$	AND	Konjunktion
Leere Menge	$\emptyset$	falsch	f	low (Pegel)	0
Allmenge	1	wahr	w	high (Pegel)	1

# Kontinuierlich $\Leftrightarrow$ digitale Funktionen

Bei *kontinuierlichen Signalen, reellen Zahlen* bewirken die Eingangs-Variablen  $x, y$  usw. über die kontinuierliche Funktionen  $f()$  die dazugehörenden möglichen Werten  $f(x)$  bzw.  $f(x, y)$ .

$$f(x) = x^2, \sqrt{x}, \sin(x), \log(x), e^x \text{ usw.}$$

$$f(x, y) = x + y, x \cdot y, x / y, x^y, \sqrt[x]{y}, \log_y(x), \text{ usw.}$$

Wegen der unendlichen vielen reellen Zahlen  $-\infty \leq x, y \leq +\infty$  gibt es erst recht viele Funktion  $f(x), f(x, y)$ . Das ist noch mehr ausgeprägt bei mehr als zwei Variablen  $x_1, x_2 \dots x_n$ .

Für *binäre Zahlen* und die dazugehörenden *logischen Funktionen* sind die Zusammenhänge deutlich einfacher:

Variablen                      mögliche Zustände

$$x \notin \{0, 1\} \text{ und } f(x) \notin \{00, 01, 10, 11\}.$$

$$x \notin \{x_1, x_2\} \text{ und } f(x_1, x_2) \notin \{0000, 0001, \dots, 1111\}.$$

Infolge dieser Einschränkungen existieren auch nur wenige *digitale Gesetze*  $f() = \text{Operationen}$  als Formel-Äquivalent.

Die wichtigsten sind einstellig: YES, NOT und zweistellig: AND, OR, NAND NOR, XOR.

Für alle existieren einfache Schaltzeichen.

Sie lassen sich gut durch ein Kästchen mit 1 bzw. 2 Tasten und einem leuchtenden Lämpchen veranschaulichen.

Bei den zweistelligen Operationen ist auch eine mengen-theoretische Interpretation möglich.

# kombinatorische Schaltungen voll reversibel

YES (on)			leuchtet = ein	
NOT (off)			geht aus	
AND			bei beiden ein	
OR			bei einer oder beiden ein	
NAND			bei beiden aus	
NOR			bei mindestens einen aus	
XOR			bei nur einer ein	

XOR entspricht der Wechselschaltung der Elektrotechnik!

# Logische = kombinatorische Bauelemente

Das typische kombinatorische Bauelement zeigt die nebenstehende Skizze

Es besitzt  $n$  **Eingänge**  $x_i$ , deren Werte (0, 1) das **Ausgangssignal**  $y$  erzeugen.

Die **Verzögerung** zwischen Eingang und Ausgang soll theoretisch Null sein.

In der Praxis existiert jedoch der möglichst kleine Wert  $\Delta t$ .

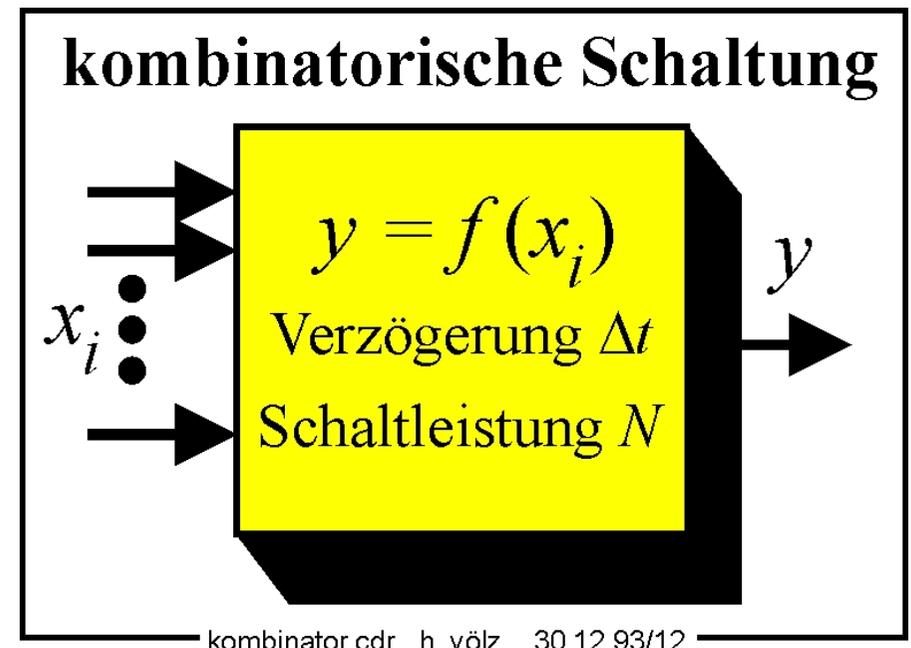
Beim Schalten wird eine möglichst kleine **Leistung**  $N$  verbraucht, theoretisch erwünscht Null.

Aus beiden folgt das **Schaltzeit-Leistungs-Produkt** in Joule.

Es ist typisch für die verschiedenen **Baureihen**.

Z. T. hängt es von der benutzten **Taktfrequenz** ab.

Mit der Entwicklung konnte sein Wert ständig verringert werden.



# Digitale Operationen

Allgemein muss die Komplexität der digitalen Operationen 3 Größen erfassen:

1. Die Anzahl  $n$  der verfügbaren **Eingangsvariablen**  $x_n$ .
2. Die dazugehörenden **möglichen Zustände** (Ausgangswerte)  $2^n$ .
3. Die Anzahl der **möglichen Verknüpfungen**  $2^{2^n}$  zu den möglichen Ausgangszuständen.

Die Tabelle zeigt, wie schnell mit den Eingangsvariablen die Anzahl der möglichen Verknüpfungen zunimmt.

Variablen	1	2	3	4	5	6	7
Zustände	2	4	8	16	32	64	128
Verknüpfungen	4	16	256	65 536	$4,29 \cdot 10^9$	$1,84 \cdot 10^{19}$	$3,40 \cdot 10^{38}$

U. a. haben sich deshalb fast nur Varianten mit **ein** und **zwei Eingangsvariablen** und deren Verknüpfungen behauptet. Außerdem gibt es mit zunehmenden Eingangsvariablen viele gleichwertige Beschreibungen der gleichen Funktion.

Ein sehr einfachen Beispiel ist:  $x_1 \vee (x_1 \wedge x_2) = x_1 \vee x_2$ .

Es gibt immer **Minimalformen**. Doch in der Praxis hat sich einheitlich keine bestimmte durchsetzen können.

Eine gewisse Verbreitung hat nur die disjunktive und konjunktive Normalform erlangt.

Für Anwendung sind alle **Minimalisierungsverfahren** recht komplex.

# Basis-Systeme

Für die formelmäßige Beschreibung und damit auch für die technische Realisierung aller möglichen Verknüpfungen genügt immer eine *geringe Auswahl der logischen Operationen*.

Besonders wichtig sind dabei die folgenden Systeme:

- 3 Funktionen: **NOT** (Negation,  $\neg$ ), **AND** (Konjunktion,  $\wedge$ ) und **OR** (Disjunktion,  $\vee$ )
- 2 Funktionen: **AND** und **XOR** (exklusives OR).
- 1 Funktion: **NAND** = nicht AND: besonders wichtig bei TTL-Schaltungen.
- 1 Funktion: **NOR** = nicht OR: besonders wichtig für ECL- und I<sup>2</sup>L-Schaltkreise.

# Gliederung

1. Grundlagen

**2. Kombinatorische Bausteine**

3. Die drei typischen Digitalerschaltungen

4. Speicher

5. Sequentielle Schaltungen

6. Automaten und Rechner

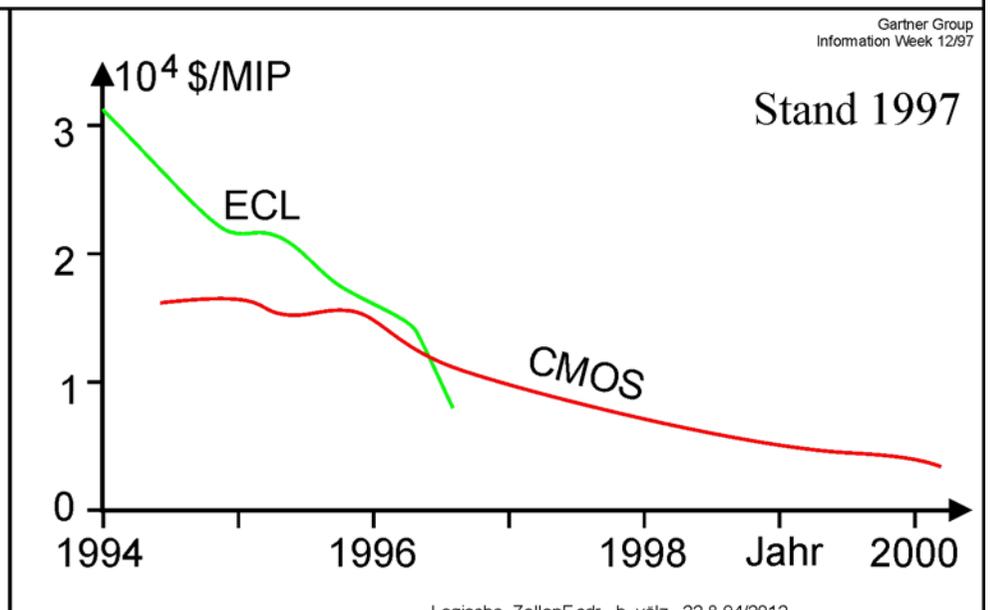
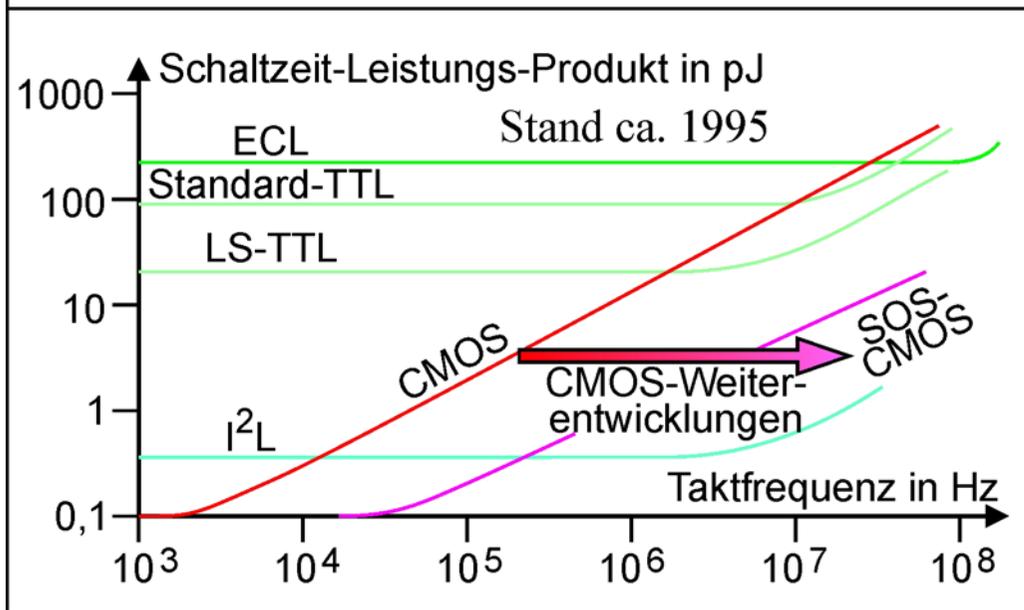
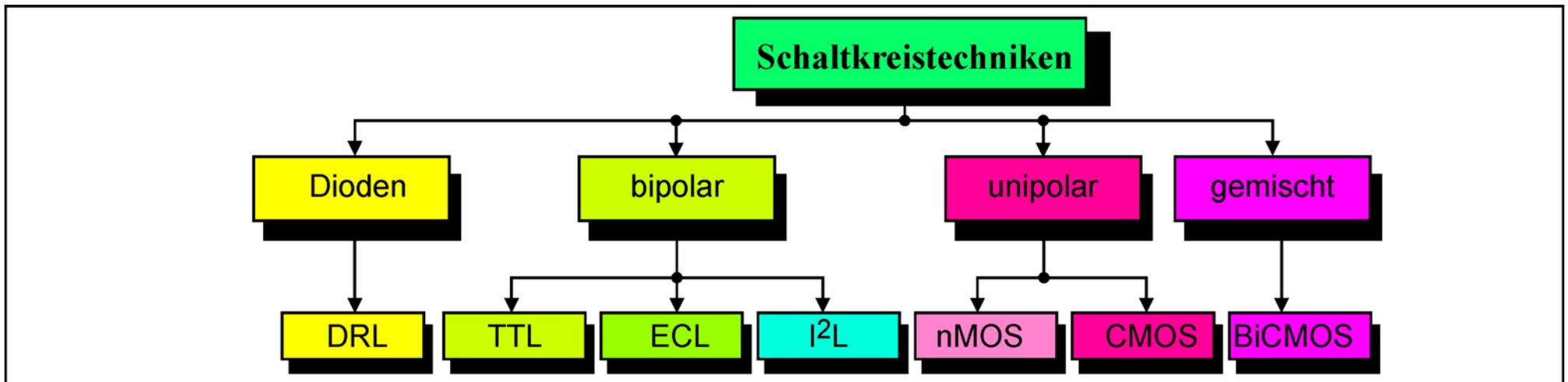
7. Ergänzende Systeme

8. Berechenbarkeit

# Technische Varianten

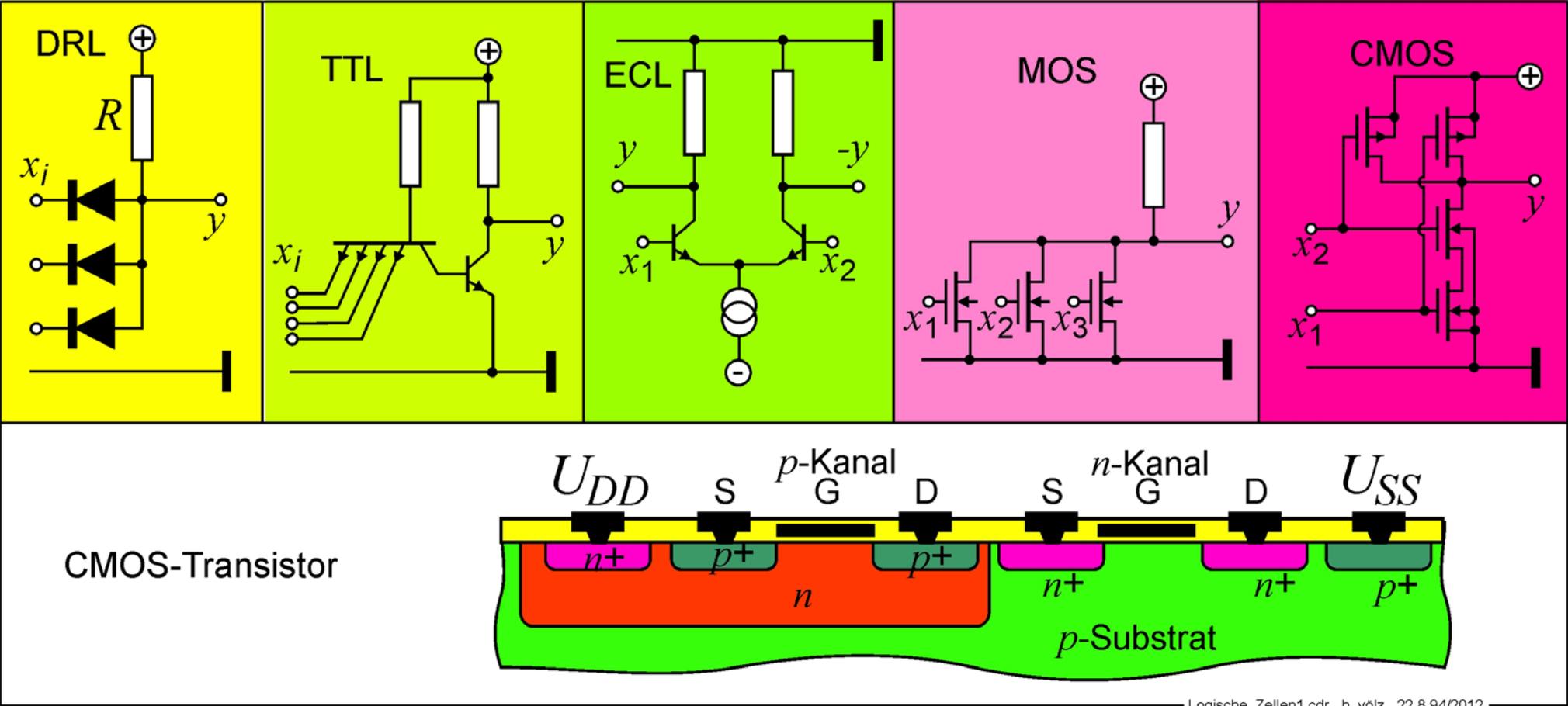
Die kombinatorischen Schaltungen werden aus Widerständen, Dioden und Transistoren aufgebaut. Im Laufe der Entwicklung entstanden dabei mehrere Baureihen, die mit Abkürzungen bezeichnet werden. Die wichtigsten sind alphabetisch geordnet (Details s. Elektronik-Buch):

<b>CMOS</b>	complimentary MOS ..... <i>wird heute vorrangig benutzt.</i>
<b>DCTL</b>	direct coupled transistor logic
<b>DRL</b>	diode resistor logic (der Begriff ist kaum gebräuchlich).
<b>DTL</b>	diode transistor logic.
<b>ECL</b>	emitter coupled logic ..... <i>war lange Zeit das schnellste System</i>
<b>I<sup>2</sup>L</b>	integrated injection logic ..... <i>ist nur integriert realisierbar, flexible im Schaltzeit-Leistungs-Produkt.</i>
<b>MOS</b>	metal oxide Silizium.
<b>RTL</b>	resistor transistor logic.
<b>TTL</b>	(T <sup>2</sup> L) transistor transistor logic... <i>ist historisch und immer noch als Bezugssystem wichtig.</i>



Logische\_ZellenF.cdr h. vözl 22.8.94/2012

# Beispiele einiger Grundstrukturen



Logische\_Zellen1.cdr h. vözl 22.8.94/2012

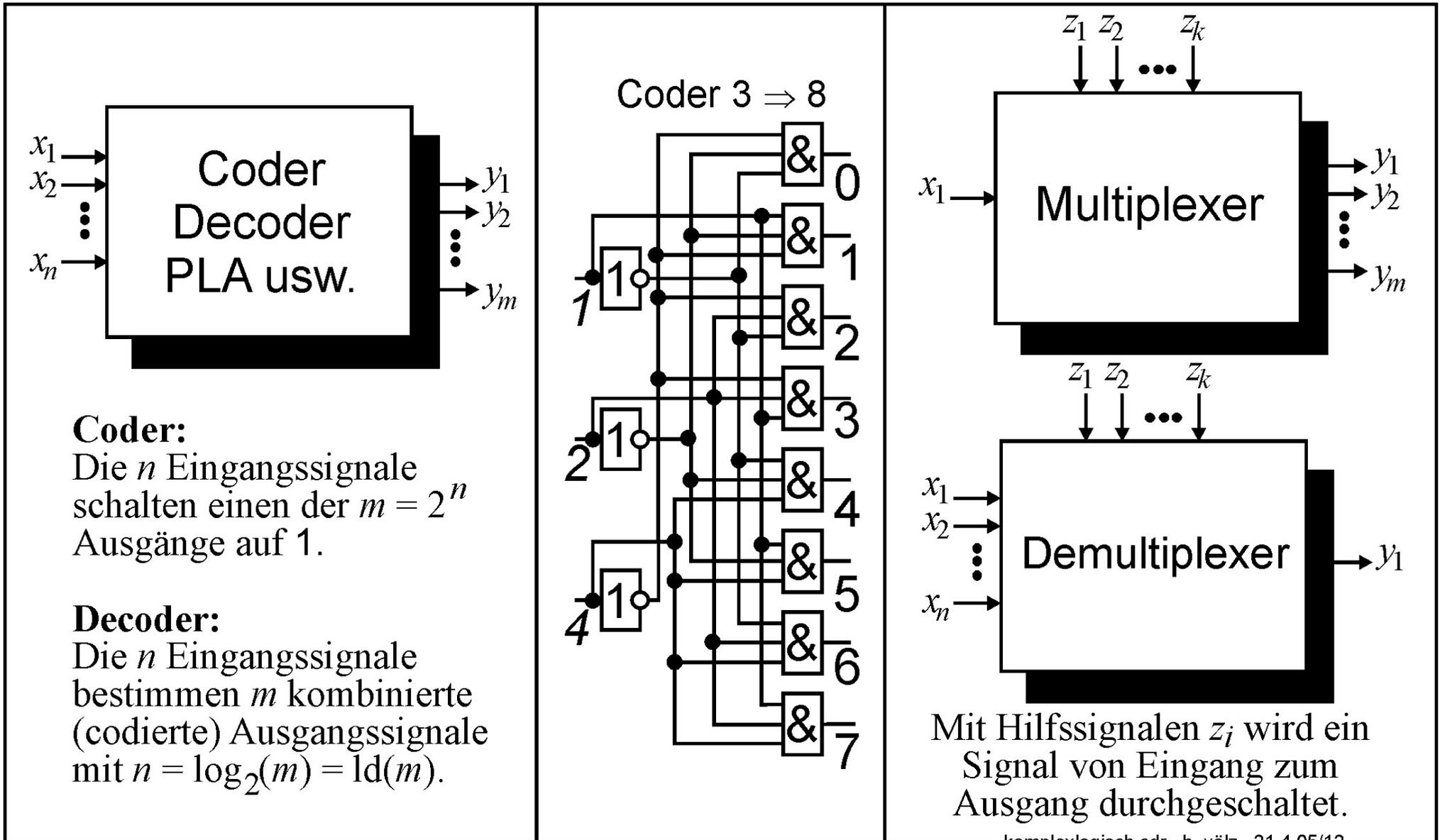
# Komplexe kombinatorische Bausteine

Es gibt eine Vielzahl zusammengesetzter logischer Bausteine.

Gegenüber den einfachen Bausteinen besitzen sie vor allem mehrere Ausgänge

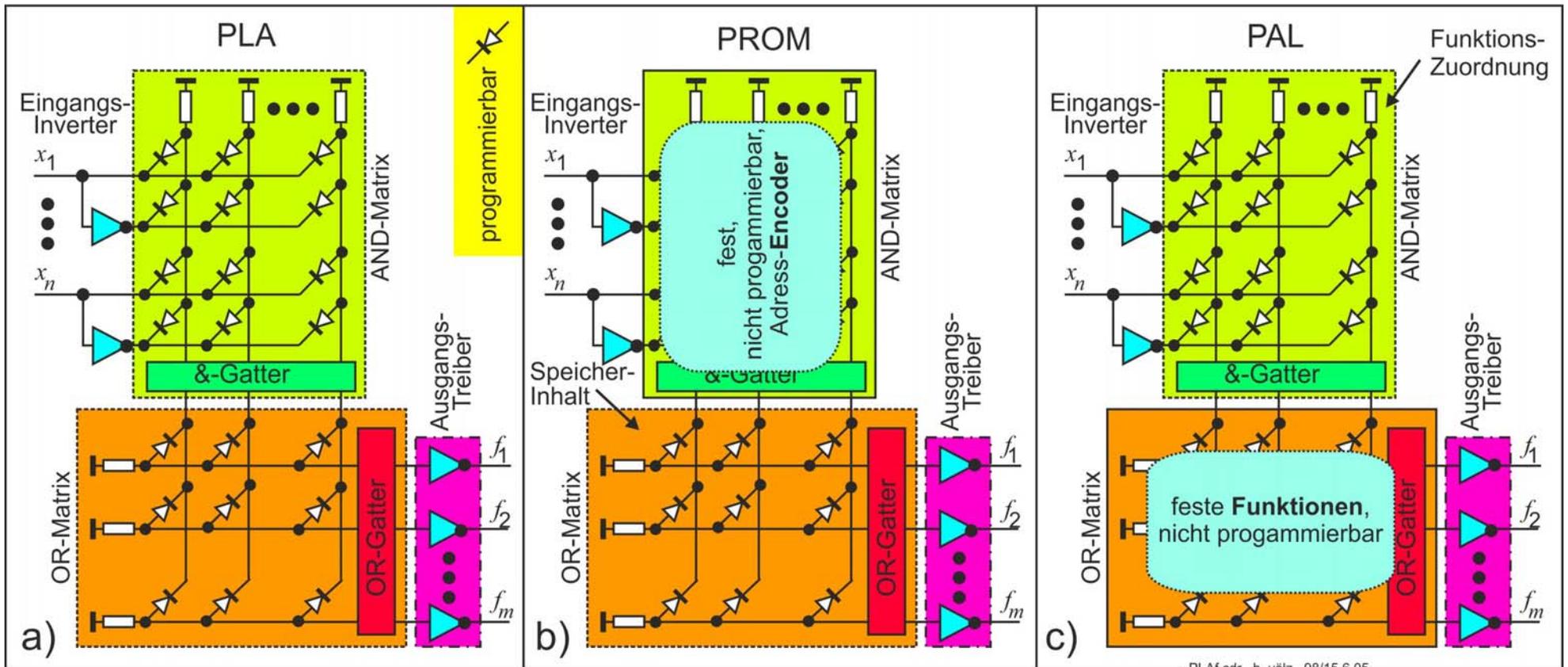
Dafür bestehen mehrere Möglichkeiten. Die wichtigsten sind:

- Es sind **mehrere** einfache logische Schaltungen **zusammengefasst**.  $y_i = f_i(x_1, x_2, \dots, x_n)$  mit  $i = 1$  bis  $m$  ( $m$  beliebig)
- Teilweise sind hierbei die **Funktionen programmierbar**, dann entstehen folgende Bausteine (übernächstes Bild):
  - PLA = programmable logic array.
  - PROM = programmable read only memory.
  - PAL = programmable array logic.
- Es erfolgt eine **Codierung** bzw. als Umkehrung eine **Decodierung**,  
Bei der Codierung wird z. B. mit  $n$  Eingängen eine Leitung von  $m = 2^n$  Ausgängen auf 1 geschaltet.  
Dieses Prinzip ist für **Speicher** sehr wichtig (s. d.).  
Ein Coder wird zuweilen auch **Encoder** genannt.
- Beim **Multiplexer** und **Demultiplexer** existieren mehrere Ausgangs- bzw. Eingangssignale.  
Sie werden mit logischen  $z$ -Signalen zu einem Ausgang bzw. von einem Eingang durchgeschaltet.



komplexlogisch.cdr h. völz 21.4.05/12

Multiplexer und Demultiplexer sind Spezialfälle!



# Gliederung

1. Grundlagen
2. Kombinatorische Bausteine
3. Die drei typischen Digitalerschaltungen
4. Speicher
5. Sequentielle Schaltungen
6. Automaten und Rechner
7. Ergänzende Systeme
8. Berechenbarkeit

# Weitgehend vollständiger Überblick

In meisten Lehrbüchern werden nur *zwei Hauptklassen* digitaler Schaltungen behandelt.

Die Bezeichnungen dafür sind leider je nach Autor recht unterschiedlich (s. u.).

Das beginnt bereits mit *digitale Schaltung = binäre Technik* usw.

Für die beiden Hauptgruppen verwende ich *kombinatorische* und *sequentielle* Schaltung

Das betont ihr *funktionelles Verhalten*.

Doch viel wesentlicher ist es, dass der *Speicher* meist nur indirekt auftritt.

Doch er ist ganz *fundamental* und bestimmt die technologische Halbleiter-Entwicklung.

Die Ursache für seine Unterbewertung könnte vielleicht seine *formale Einfachheit* sein:

Er übernimmt „einfach“ die Daten und gibt sie später bei der Wiedergabe zurück.

Allgemein wird vergessen, dass seine *Vielfalt besonders groß* ist und

dass er eine fundamentale Grundlage der *Natur* und des *Menschen* ist.

Alles *was wir wissen* geht auf gespeicherte Daten zurück.

Ohne Speicherung wissen wir nichts von *gestern*, von *uns* und können uns auch kein *morgen* vorstellen

Es gäbe nicht einmal eine *Zeit*.

Ganz deutlich zeigen das die äußerst bedauerlichen *Alzheimer-Kranken*.

Aus all diesen Gründe benutze ich eine *andere Einteilung* aller digitalen Schaltungen.

# Begriffe der binären Technik

kombinatorische  
Schaltung

sequentielle  
Schaltung

## Synonyme

z.T. mit etwas abweichender Bedeutung

Schaltwerk,  
statische Logik,  
binäre Schaltung,  
Zuordner,  
Codierer.

Schaltnetz,  
Folge-Schaltung,  
dynamische Logik.

## Theorien

Schaltalgebra  
Boolesche Algebra  
formale Logik  
Karnaugh-Diagramm

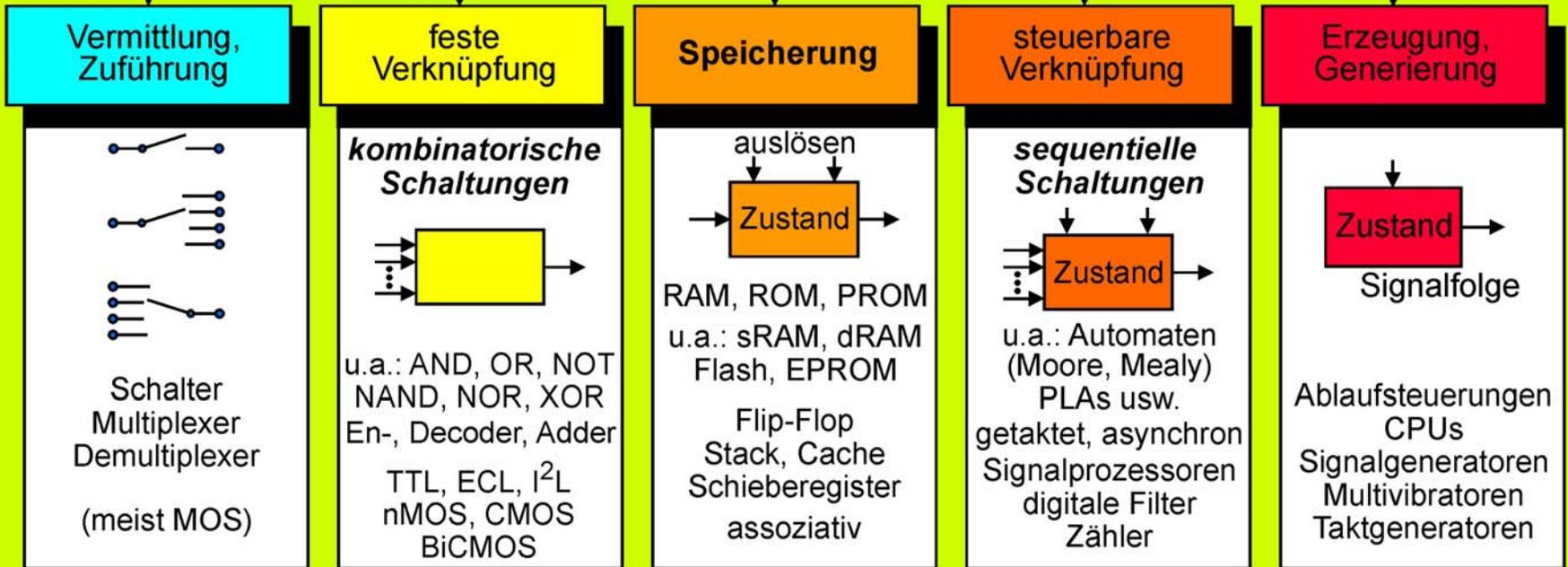
Automatentheorie  
Petrietze

binschalF.odr h. vözl 31.12.93

# Funktionen elektronischer Schaltungen

In-, Output von Signalen

wachsende Komplexität



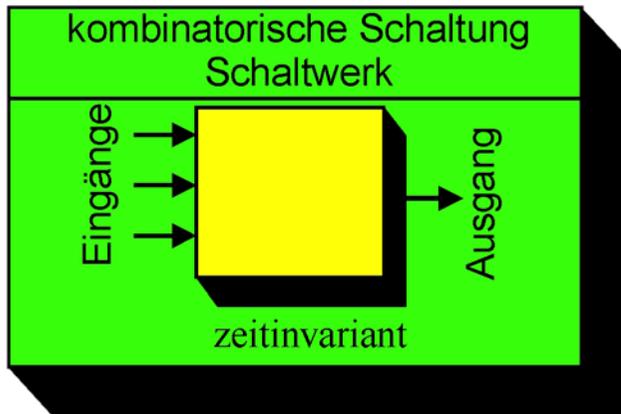
diskret, digital

kontinuierlich, analog

AD-, DA-Wandler, Begrenzer, Schmitt-Trigger, Timer u.a.

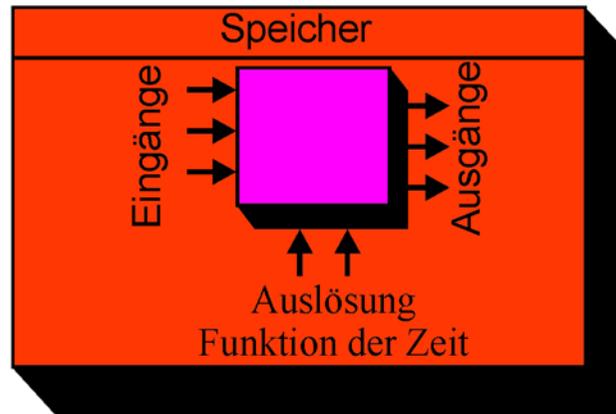
Verstärker, Modulatoren, CCD, Laufzeitketten, Filter, Oszillatoren, nichtlineare Bauelemente u.a.

# Die drei fundamentalen digitalen Systeme



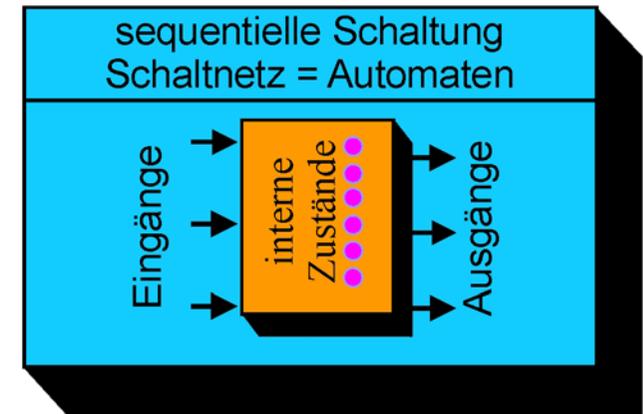
$$x_a = f(x_1, x_2, \dots, x_n)$$

ohne Zeiteinfluß



$$x_a = x_e$$

speichert Eingänge  
durch Auslösung  
zu anderen Zeiten

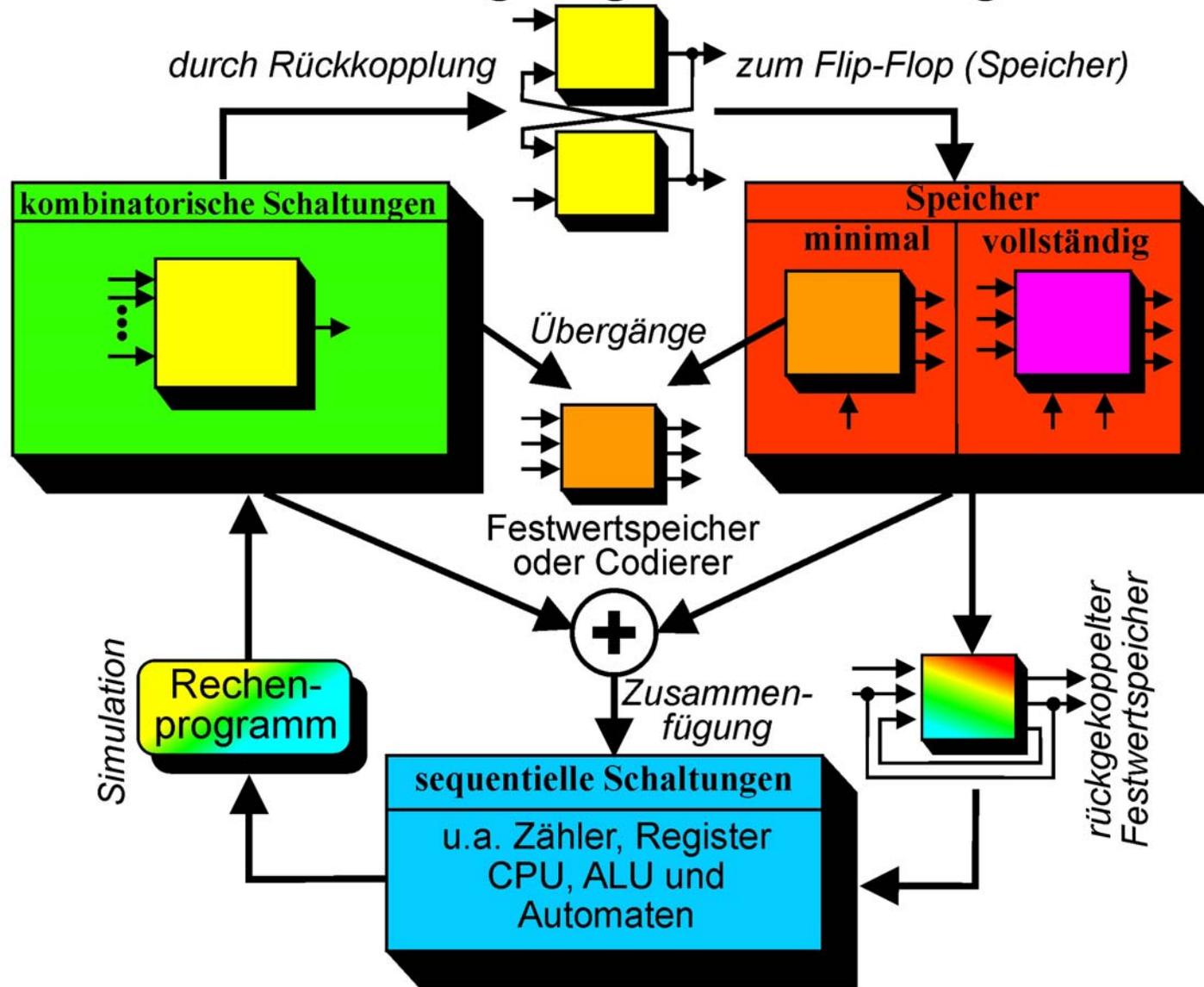


$$x_a = f(Z_i, x_e)$$

$$Z_i = g(Z_i, x_e)$$

komplexe und zeitabhängige  
Auswirkung der Eingänge

# Zusammenhänge digitaler Schaltungen



elschaltF.cdr h. vözl 30.12.93/05/09

# Gliederung

1. Grundlagen
2. Kombinatorische Bausteine
3. Die drei typischen Digitalschaltungen
- 4. Speicher**
5. Sequentielle Schaltungen
6. Automaten und Rechner
7. Ergänzende Systeme
- 8 Berechenbarkeit

# Speichern allgemein

Gespeichert werden u. a. *Stoffe, Energien* und *Information*.

Speichern ist sinnvoll, um Etwas von der *Gegenwart* für die *Zukunft* aufzuheben.

Das setzt den *Zeitpfeil* voraus.

Die Menschheit hat *Verfahren* entwickelt, um *Etwas* für die *Zukunft* zu berechnen.

Es gibt aber kaum entsprechenden Verfahren *Vergangenes zu berechnen*. Dafür werden Speicher benötigt.

Fast immer verlangt Speichern *Irreversibilität* und zwar ganz im Gegensatz zu den reversiblen Naturgesetzen.

Irreversibilität kann u. a. durch *Rückkopplung und Zusammenfügen* mehrerer reversibler Bausteine erzeugt werden.

Speichern ist stets *unvollständig*. Seine *Inhalte* betreffen:

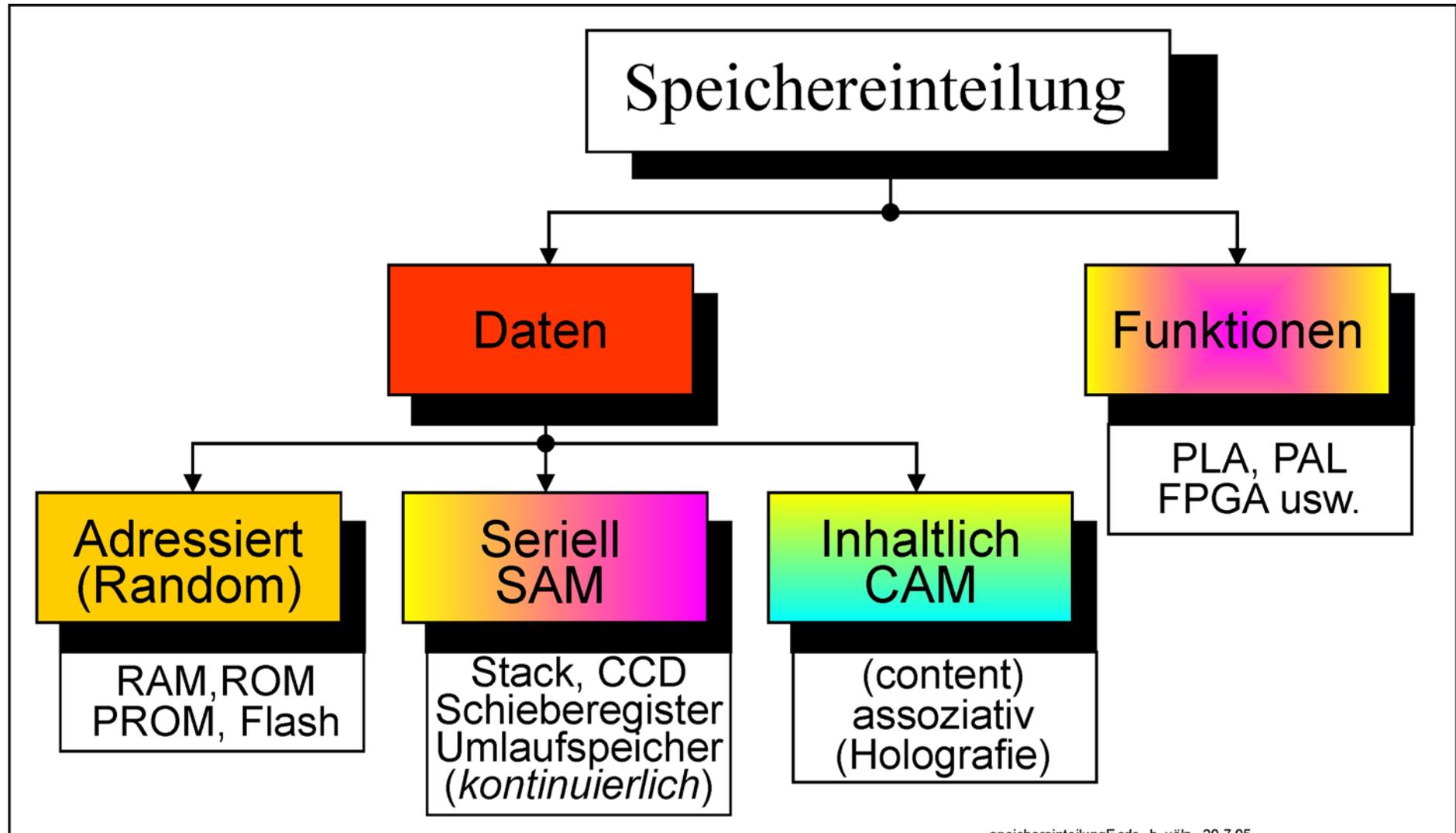
- In der Zeit beständige *Zustände*, z. B. Daten, Fakten, Texte, Bilder, Formeln usw.
- *Abläufe in der Zeit*, z. B. für Schall, Film, Video, Prozesse, Geschehen usw.

Dabei müssen unterschieden werden:

- der *Informationsträger* als das stofflich-energetische Substrat (nur dies interessiert hier!) und
- das *Informat* (Getragene) als das im Empfangssystem Bewirkte.

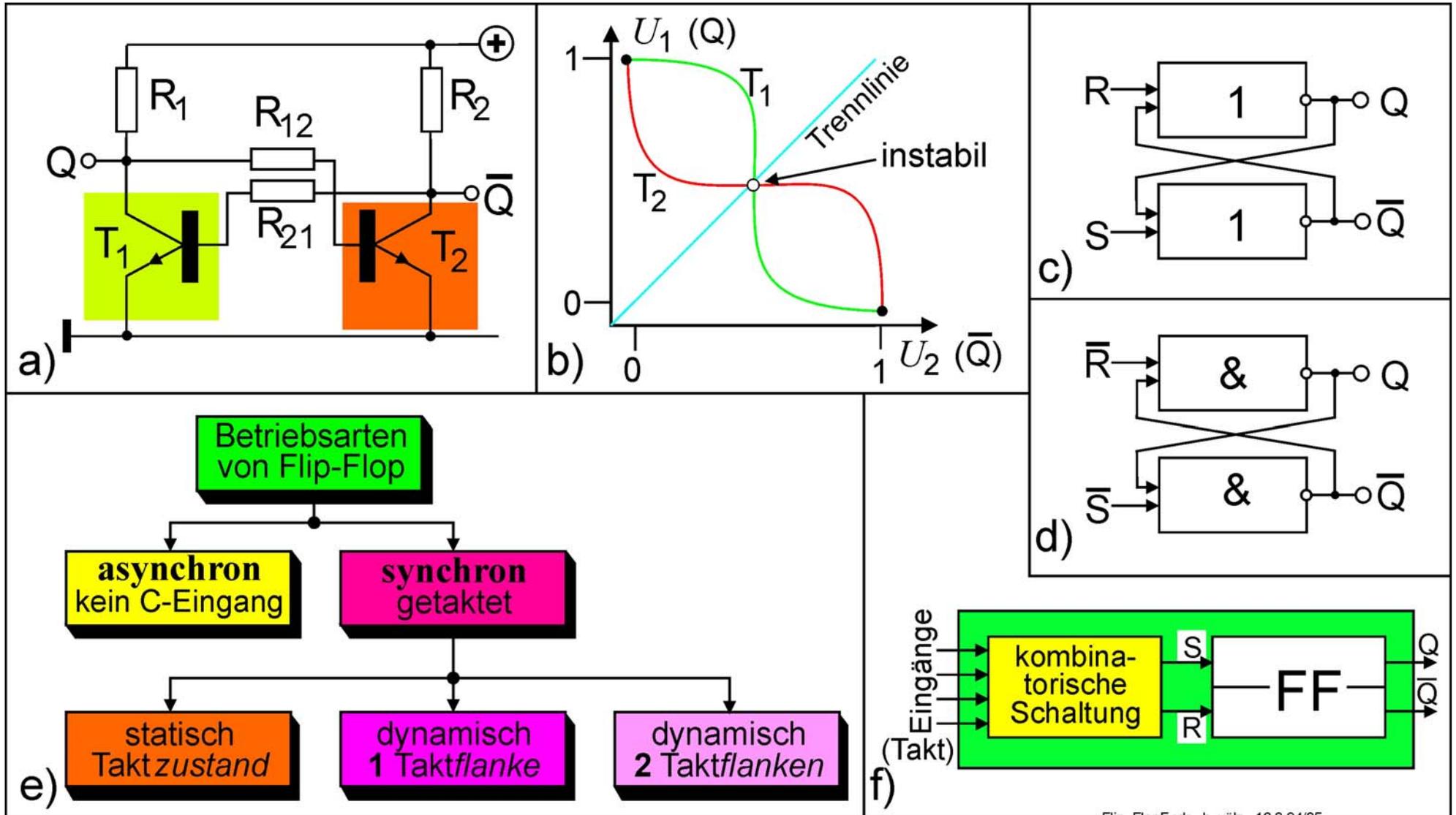
Weitere Details und Literatur u. a.: [r-h-voelz/pdf/HU/SpeichernFundament.pdf](http://r-h-voelz/pdf/HU/SpeichernFundament.pdf)

# Speichern in der Digitaltechnik



speichereinteilungF.cdr h. vözl 20.7.05

# Das Flip-Flop (FF) als Basis-Element



Flip\_FlopF.cdr h. vözl 16.8.94/05

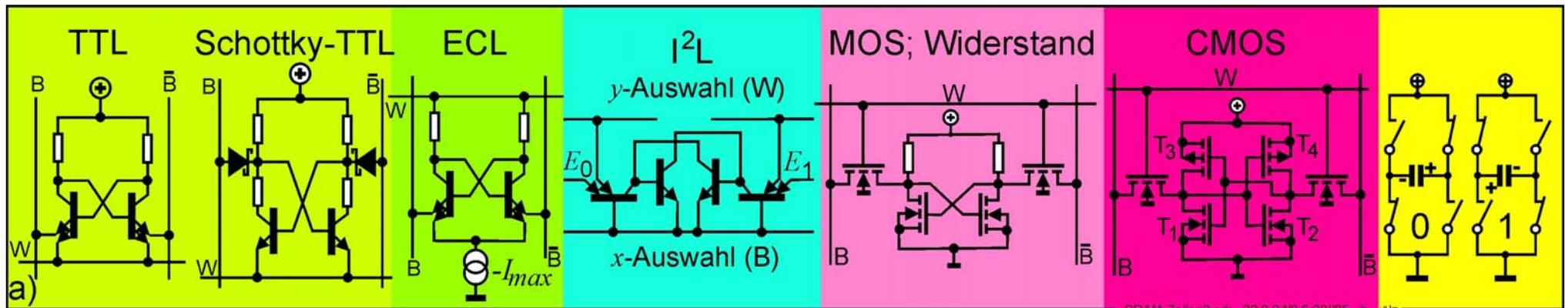
# Technologien für FF

**W** = Wort-Leitungen; **B** = Bit-Leitungen, wichtig bei *Matrix-Auswahl*; s. u.

MOS; widerstand verlangt nur **4 Transistoren**, aber ein *Widerstand* benötigt mehr *Platz* als ein Transistor  $\Rightarrow$  **6-Transistor-Zelle**.

Bei **CMOS** sind immer 2 diagonal gegenüberliegende Transistoren leitend bzw. gesperrt.

Daher besitzt CMOS *fast keinen Ruhestrom*, er wird nur beim Schalten benötigt: Auf- und Entladen der *Kondensatoren*.



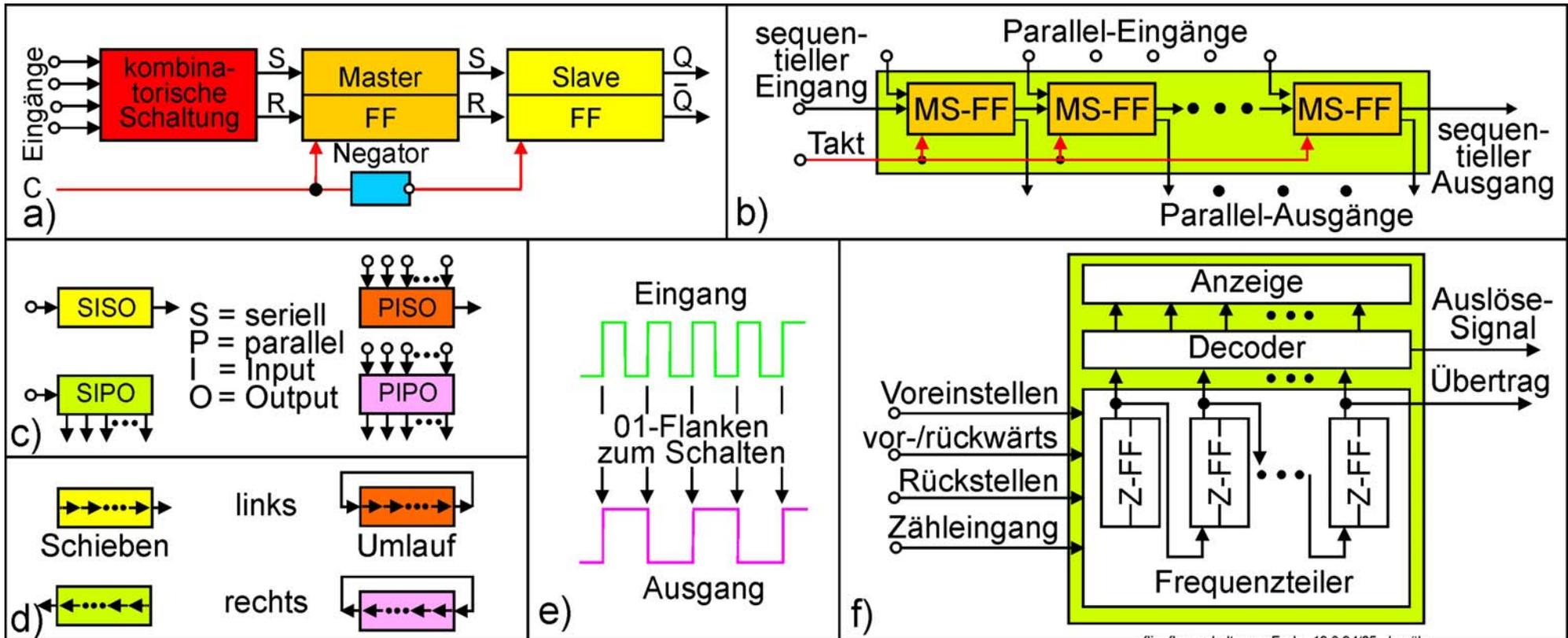
SRAM-Zellen2.cdr 22.8.94/9.5.98/05 h. vözl

# Anwendungen von FF

**Master-Slave-FF** zur Vermeidung von Problemen mit Taktunsicherheiten

**FF-Kette** ermöglicht *Seriell- ↔ Parallel-*Wandlung und *Schieberegister* mittels Takt.

Ferner ermöglicht sie *Zähler*-Schaltungen (Takt- = Frequenz-Teiler), benötigt zusätzliche *Decoder* zur Anzeige.



# Folgen großer Speicherkapazität

Zunächst waren nur wenige Bit (FF) *auf einem Chip*.

Dabei reichte für die *Auswahl einer Speicherzelle* eine Aktivierungsleitung

Doch bereits bei **16 Bit** wären 16 Auswahlleitungen, 2-mal Stromversorgung sowie Ein- und Ausgangleitung = **20 Leitungen** erforderlich.

Das war jedoch mit der beschränkten *Pin-Anzahl* (= äußere Anschlüsse) eines Chips nicht möglich.

Inzwischen sind zwar um 100 Pin möglich, aber die Speicherkapazität liegt bei GBit  $\Rightarrow$  **1 Milliarde Pin**.

Es mussten also andere *Auswahlmethoden* gefunden werden.

Dies waren zuerst die *Matrix-Schaltung* und dann die *Coder* und *Decoder* (Multiplexer, Demultiplexer).

Infolge der Mehrstufigkeit traten hierdurch zusätzliche Verzögerungen ein und es mussten *Sondercodier*-Schaltungen entwickelt werden.

Es folgten die Wort-Adressierung und dann das **CAS-RAS** (column bzw. row address select) mit Zwischenspeicherung.

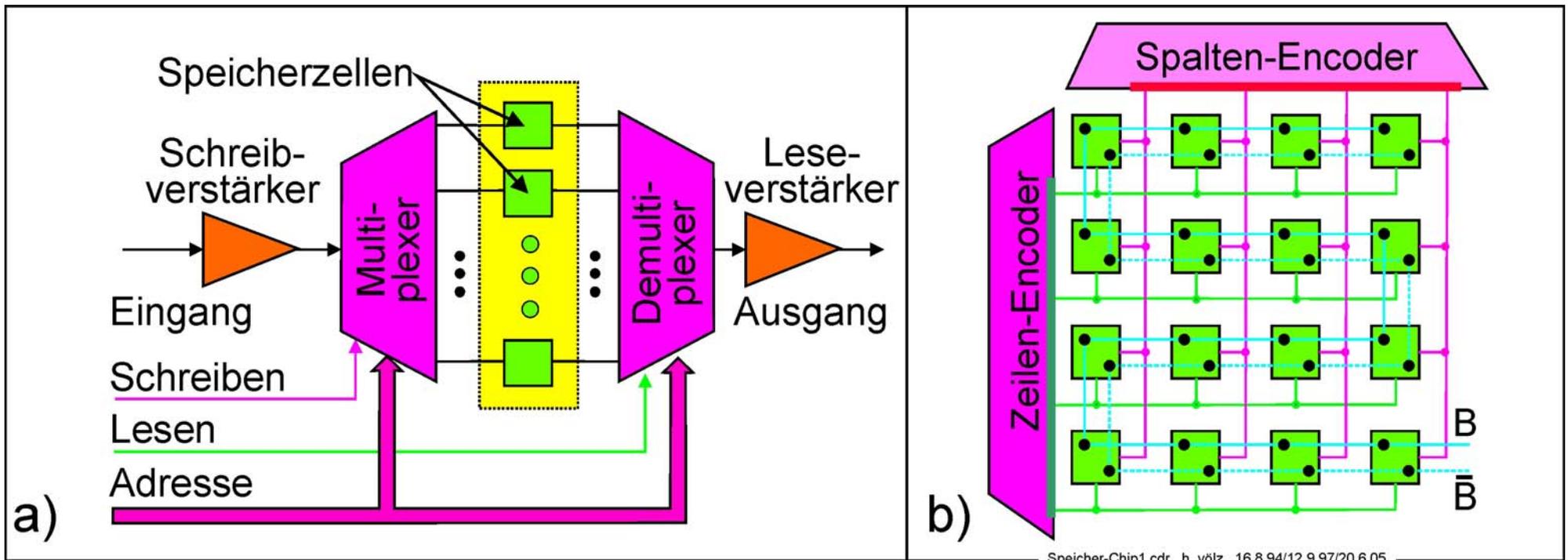
Durch das Zusammenwirken der vielen Speicherzellen wurden außerdem die *nutzbaren Signale* der FF immer kleiner.

Das erforderte *hochempfindliche Wiedergabeverstärker*.

Bei den **dRAM** (s. u.) waren noch zusätzliche *Refresh-Verstärker* und *Dummy-Zellen* notwendig.

So entstanden schließlich Speicherschaltkreise, bei denen die Speichermatrix nur noch  $\frac{1}{4}$  *der Komplexität und Fläche* ausmachte.

Links. Auswahl einzelner Speicherzellen durch Multiplexer und Demultiplexer (Coder und Decoder)  
 Rechts. Auswahl durch Matrizierung in Zeilen und Spalten.



Die vollständige Schaltung des ersten größeren Speichers 7481 in TTL-Technik für 16 Bit in Matrix-Ansteuerung.

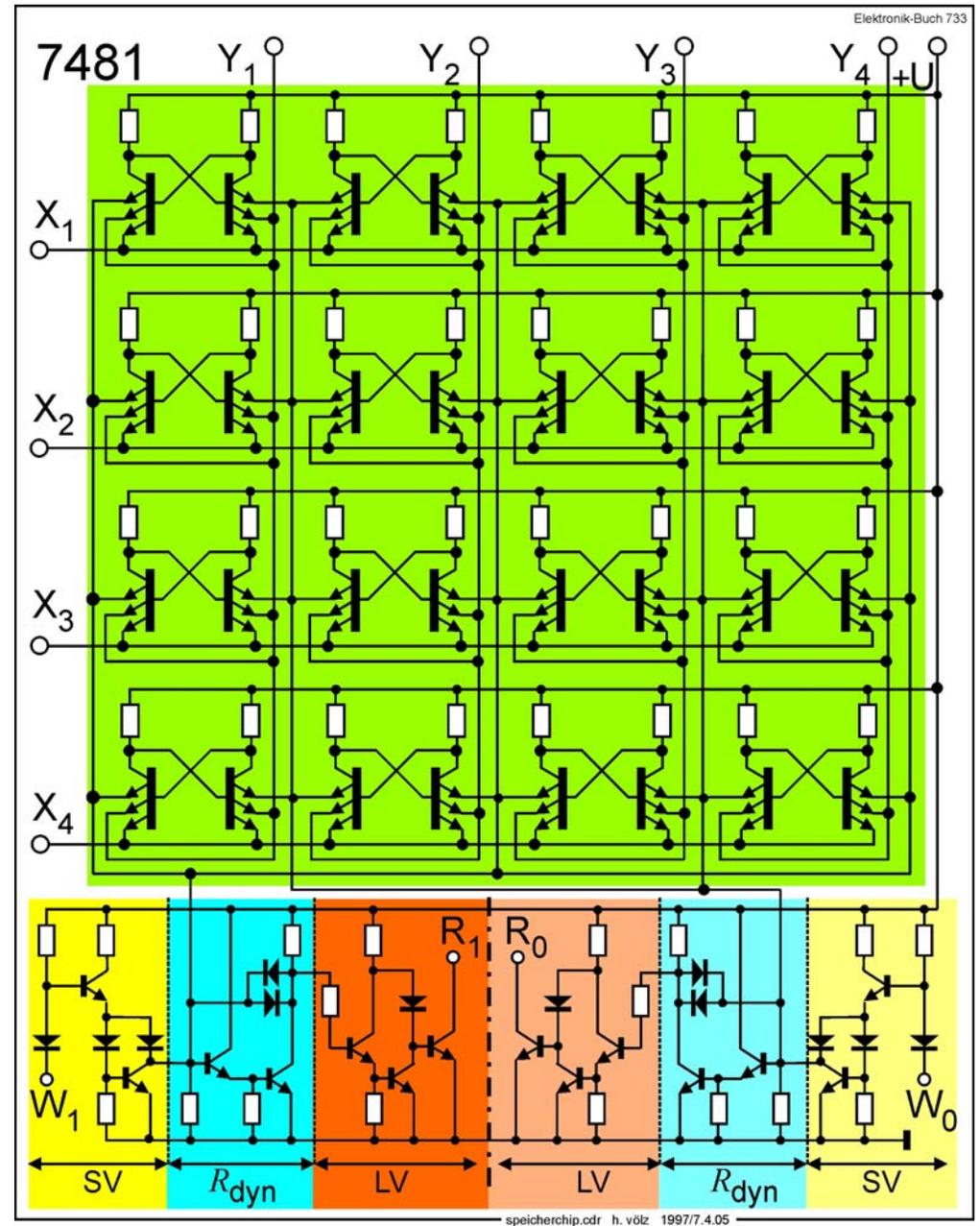
SV = Schreibverstärker.

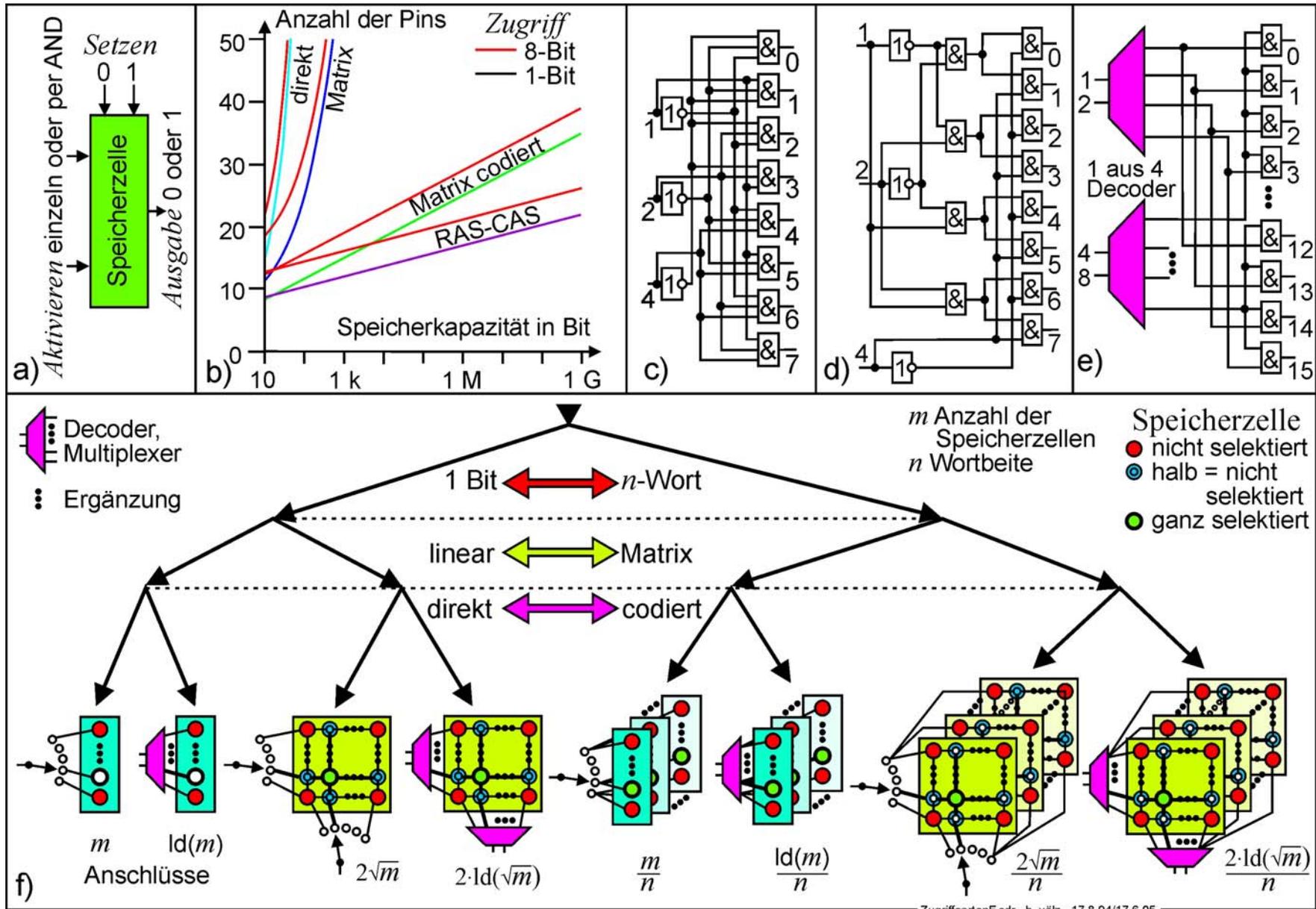
LV = Leseverstärker.

$R_{dyn}$  = elektronische (dynamischer Widerstand für die Zeilen bzw. Spalten.

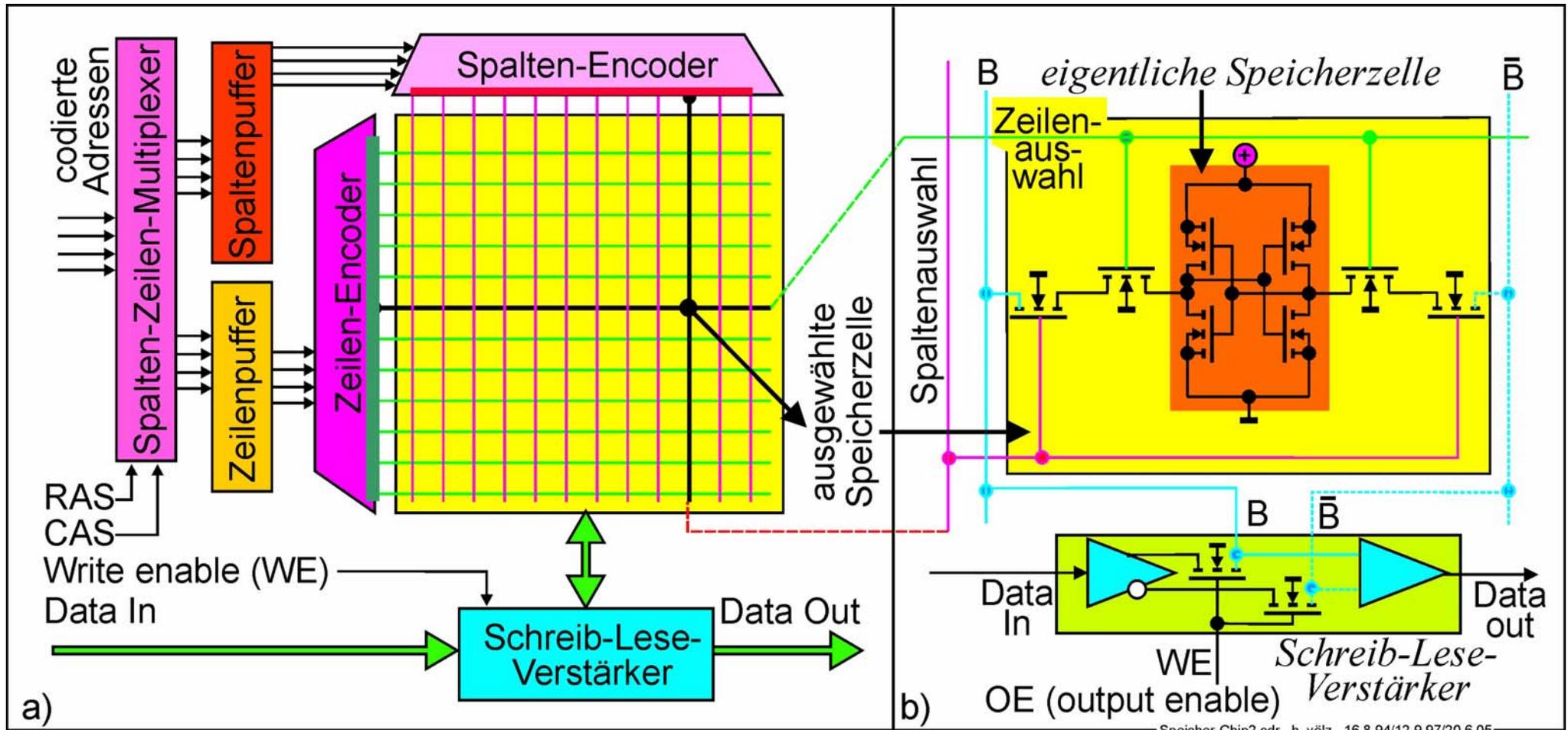
$W_0, W_1$  Schreibsignal.

$R_0, R_1$  Lesesignal (read).





# Prinzipaufbau eines heutigen sRAM-Speichers, rechts die Speicherzelle mit dem Verstärkern



# dRAM-Speicher

Der Aufwand und die Siliziumfläche (Preis) für die 6-Transistorzelle ist sehr groß.

Daher entwickelte 1966 ROBERT H. DENNARD bei IBM die 1-Transistor-Zelle, das dRAM.

Der erste 1-KBit-Speicher wurde dann im Oktober 1970 vorgestellt.

Von da an erfolgte ein sehr schneller Fortschritt zu erheblich größeren Kapazitäten.

Kondensatoren waren viel früher zur Speicherung (von Ladungen) benutzt.

Das Problem besteht in den immer vorhandenen *Verlustwiderständen*  $R$ .

Dadurch sinkt die Ladung *exponentiell* gemäß der Zeitkonstante  $T = R \cdot C$  auf  $1/e \approx 0,36$ .

Auf einem Halbleiterchip sind nur sehr kleine Kapazitäten von wenigen **fF** möglich.

So sind bei heutigen Speichern nur noch *wenige tausend Elektronen* vorhanden.

Trotz guter Isolation ist dann nach wenigen *ms* die Ladung für eine Wiedergabe zu klein.

Die Ladung muss vor dieser Zeit gelesen und dann wieder vollständig eingeschrieben werden.

Dieser *Refresh-Vorgang* muss für alle Speicherzellen im ms-Abstand wiederholt werden.

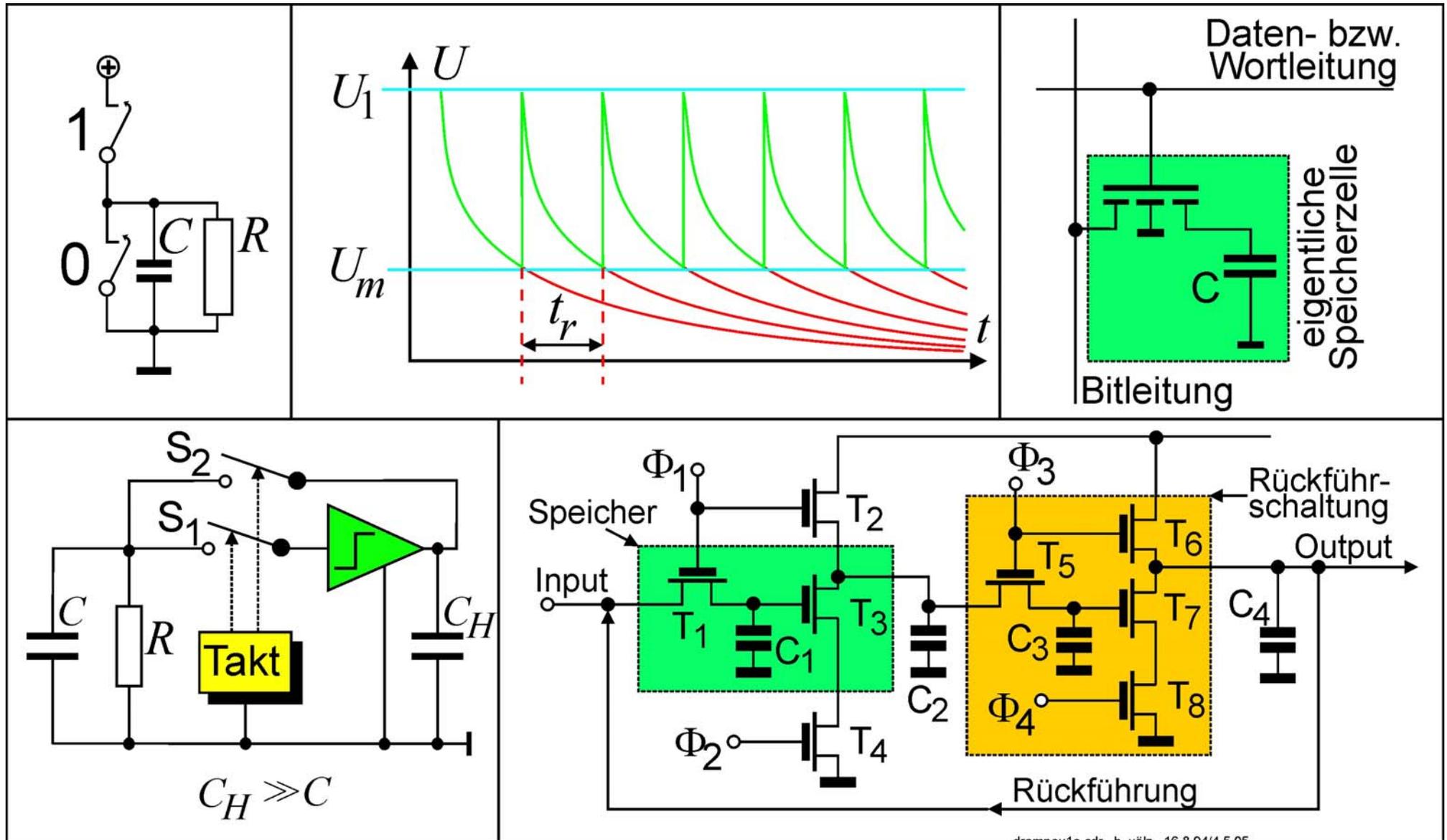
Hierbei müssen mit einem Refresh-Verstärker *alle Speicherzellen der Reihe nach* durchlaufen werden.

Er ist auch unmittelbar nach der Wiedergabe notwendig, weil dabei ein Grossteil der Ladung verbraucht wird.

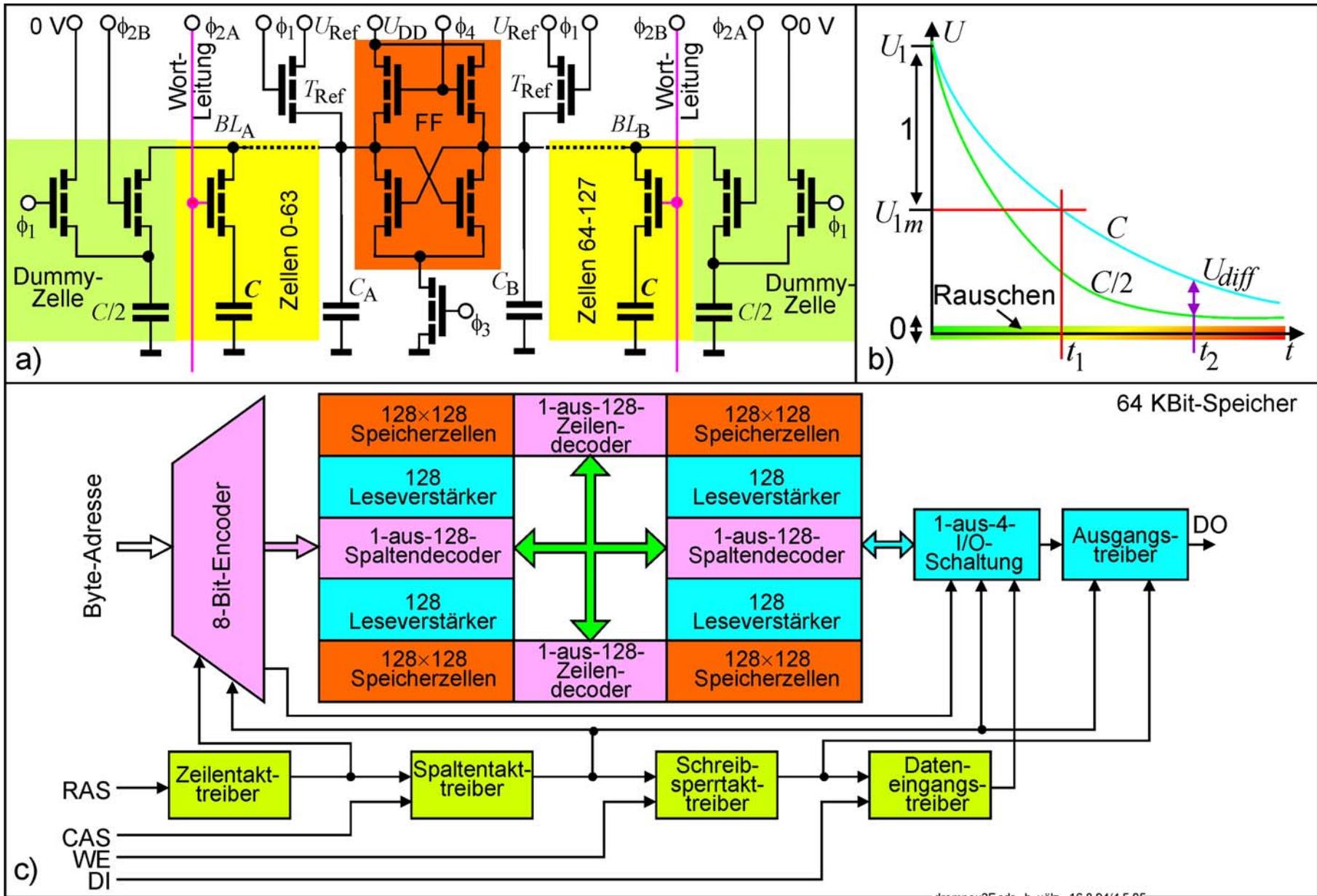
Dies bedeutet einen großen Aufwand, verlängert die *Zugriffszeit* und verbraucht *viel Energie*.

Deshalb müssen inzwischen Speicher *gekühlt* werden!

Beispiele und Grundschaltungen für einen dRAM-Speicher, rechts oben die 1-Transistorzelle, unten eine Refreshschaltung



dramneu1a.cdr h. vözl 16.8.94/4.5.05



# ROM - EPROM - Flash

Bisher wurden Speicher betrachtet, die immer wieder gelöscht und neu beschrieben werden.

Doch es gibt viele Anwendungen – sogar Wirtschafts-Gesetze – bei denen ein *nur einmaliges Einspeichern* genügt.

Dies kann dann sogar in einem *Produktionsprozess* erfolgen. Im weitesten Sinne gehören hierzu u. a. *Schallplatte* und *Buch*.

Solche Speicher heißen *Minimal-Speicher*, *Festwertspeicher* oder **ROM** (read only memory).

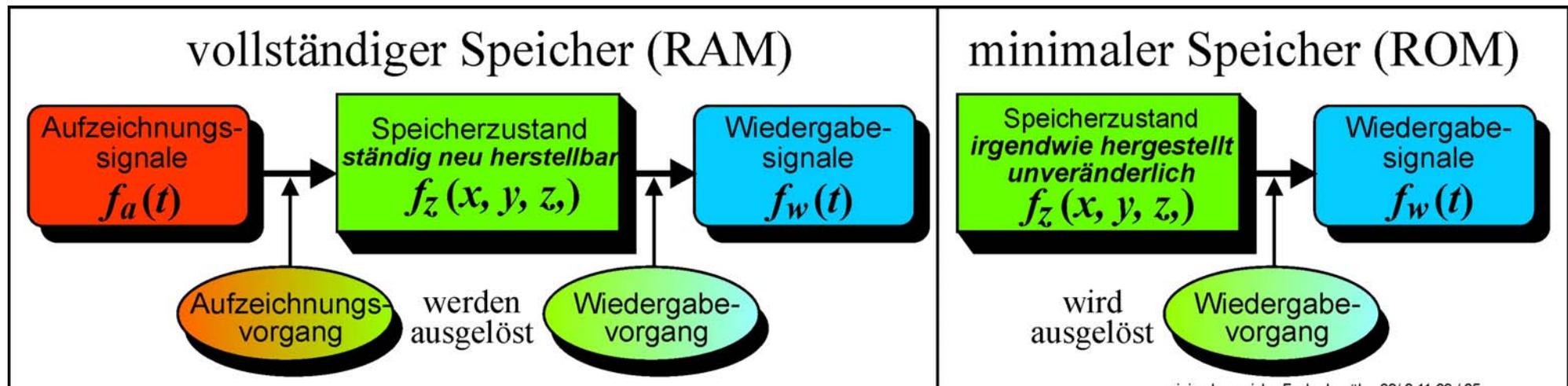
Sie enthalten die Information unabänderlich und kennen *nur den Wiedergabevorgang*.

Sie sind wesentlich einfacher aufgebaut, benötigen weniger Chipfläche und sind daher deutlich preiswerter.

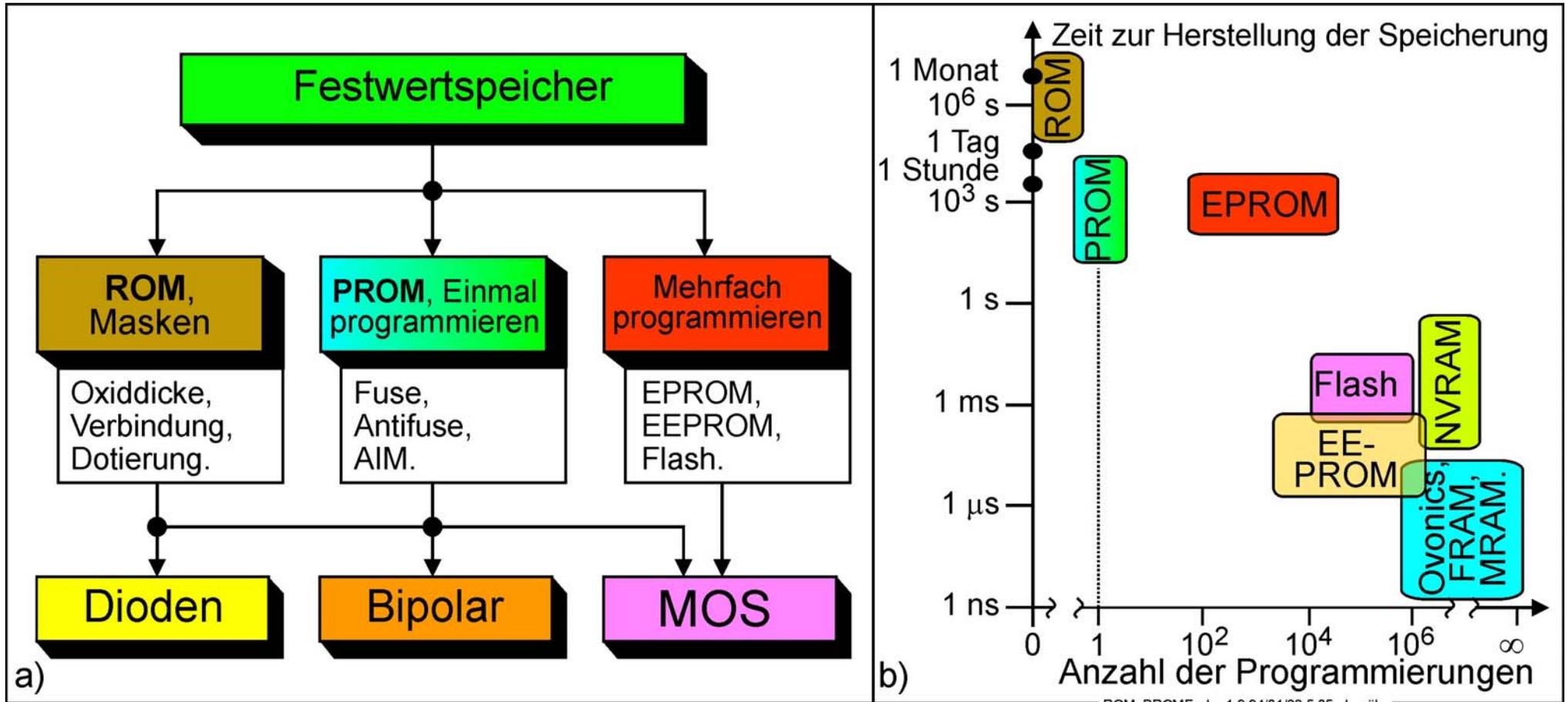
Eine Zwischenstufe zu den RAM (random access memory) sind die **PROM**, **EPROM**, **EEPROM** und **Flash**.

Sie werden als **RMM** (read mostly memory) zusammengefasst, denn sie werden nur selten beschrieben aber oft gelesen.

P = programmable, E = electrical, EE electrical erasable, Flash von englisch Blitz, schnell.

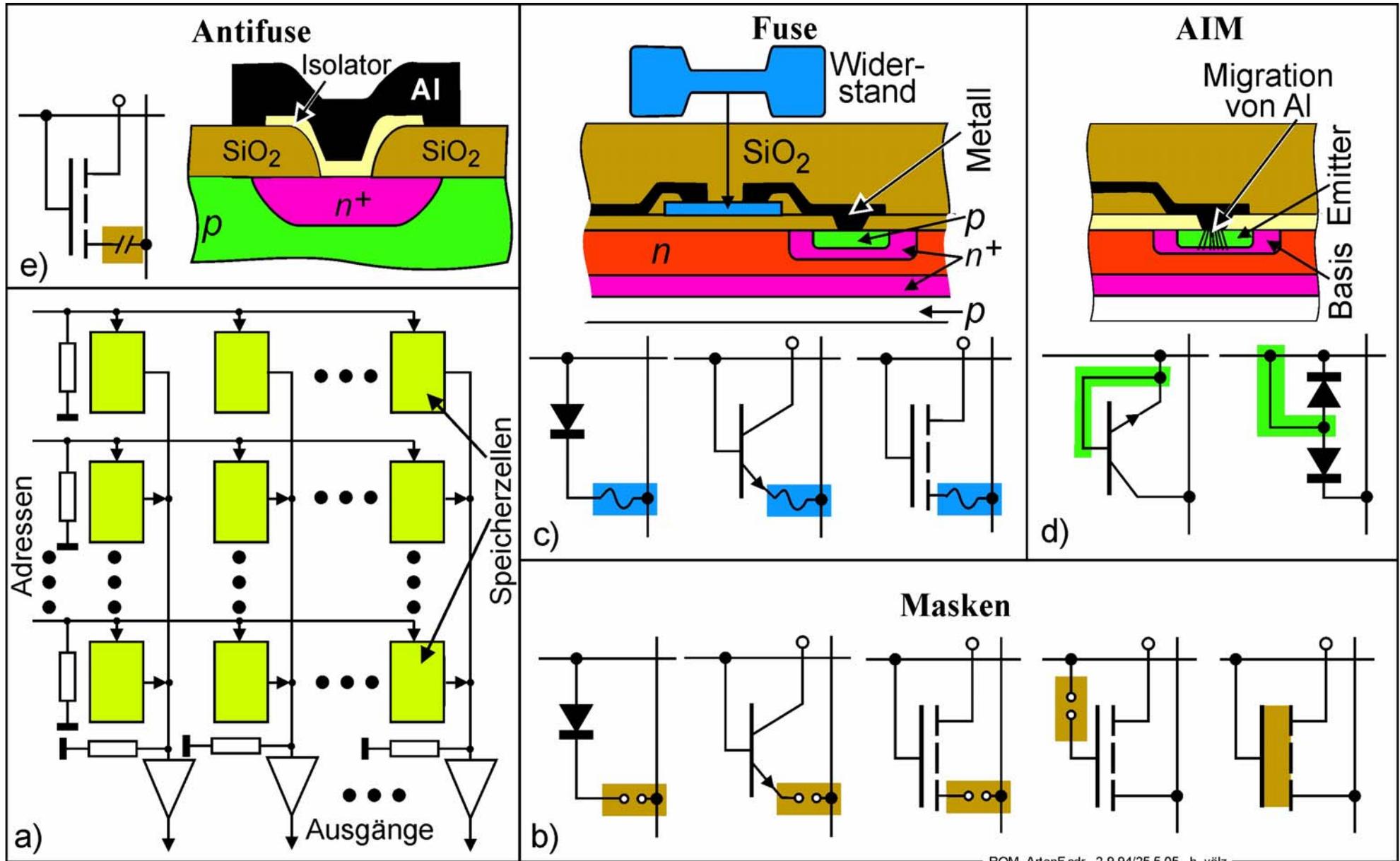


# Systematik der ROM und RMM



ROM\_PROMF.cdr 1.9.94/01/23.5.05 h. vözl

Beispiele für die wichtigsten ROM-Varianten, sie werden heute kaum noch angewendet, sind durch die Flash abgelöst.



# EPROM - EEPROM - Flash

Bei den RWW ist unter dem steuernden Gate ein *zweites Gate völlig isoliert* (isolated gate) eingefügt.

Es kann unterschiedlich *mit Ladungen belegt* werden (programmieren), dies erfolgt heute meist mit dem Tunneleffekt.

Dadurch sind zwei *Zustände* 0 oder 1, bei neueren Flash auch bis zu  $16 = 4$  Bit möglich.

Die *Isolation ist so gut*, dass eine einmal eingebrachte Ladung *Jahrzehnte* bestehen bleibt.

Das (ursprüngliche) *steuernde Gate* schaltet nur diese Speicherzelle zur Schreib- bzw. Leseleitung durch.

Um ein RWW *neu zu beschreiben*, müssen seine Speicherzellen gelöscht, die Ladungen wieder entfernt werden.

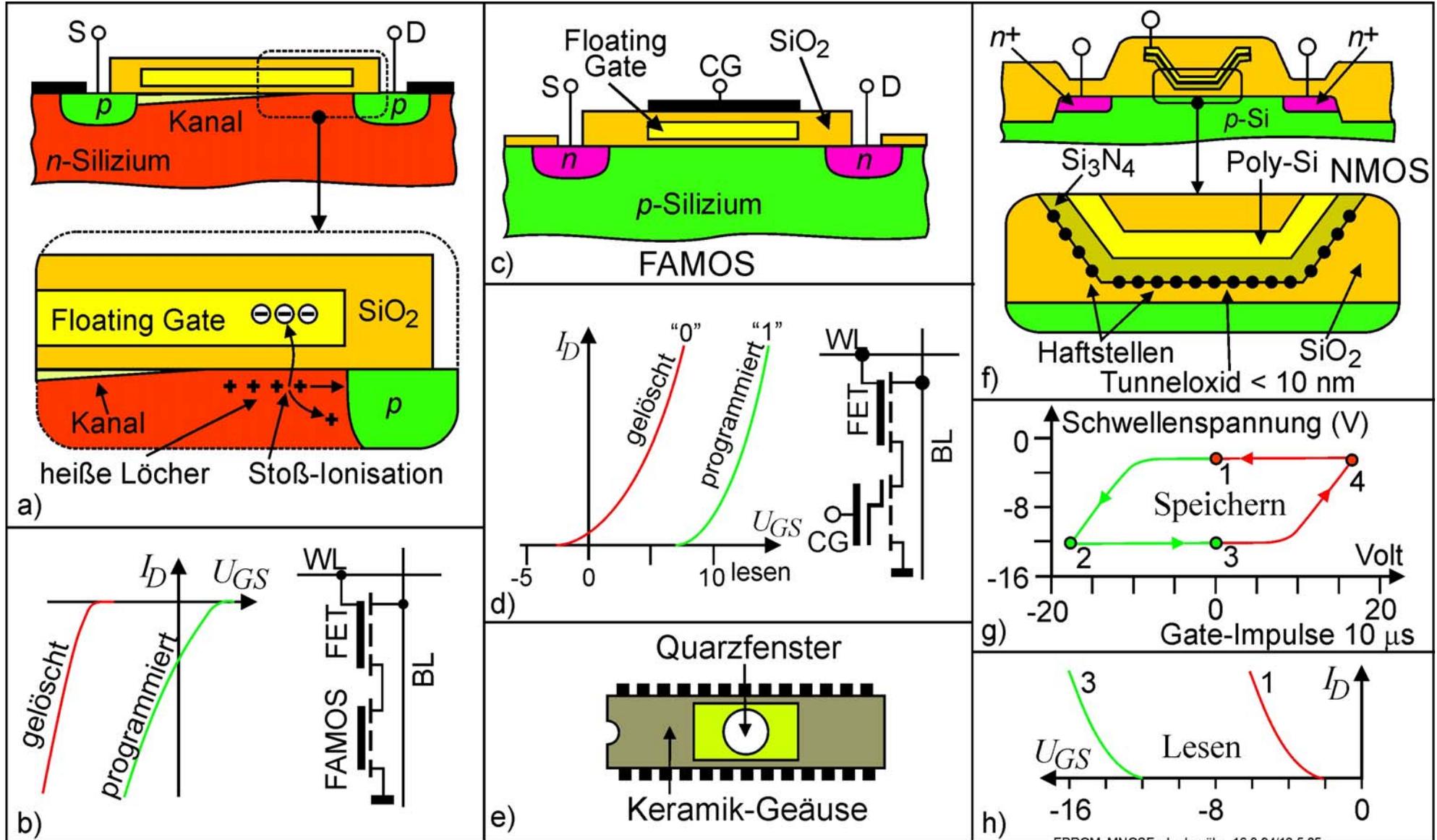
Bei den ersten **EPROM** erfolgte dies *für alle Bit* durch 10 Minuten *UV-Strahlung*.

Dazu musste das EPROM von der Leiterplatte entfernt und in ein spezielles *Löschgerät* gelegt werden.

Später gelang mit dem **EEPROM** eine elektrische Bit-weise Löschung. Doch auch diese dauerte recht lange.

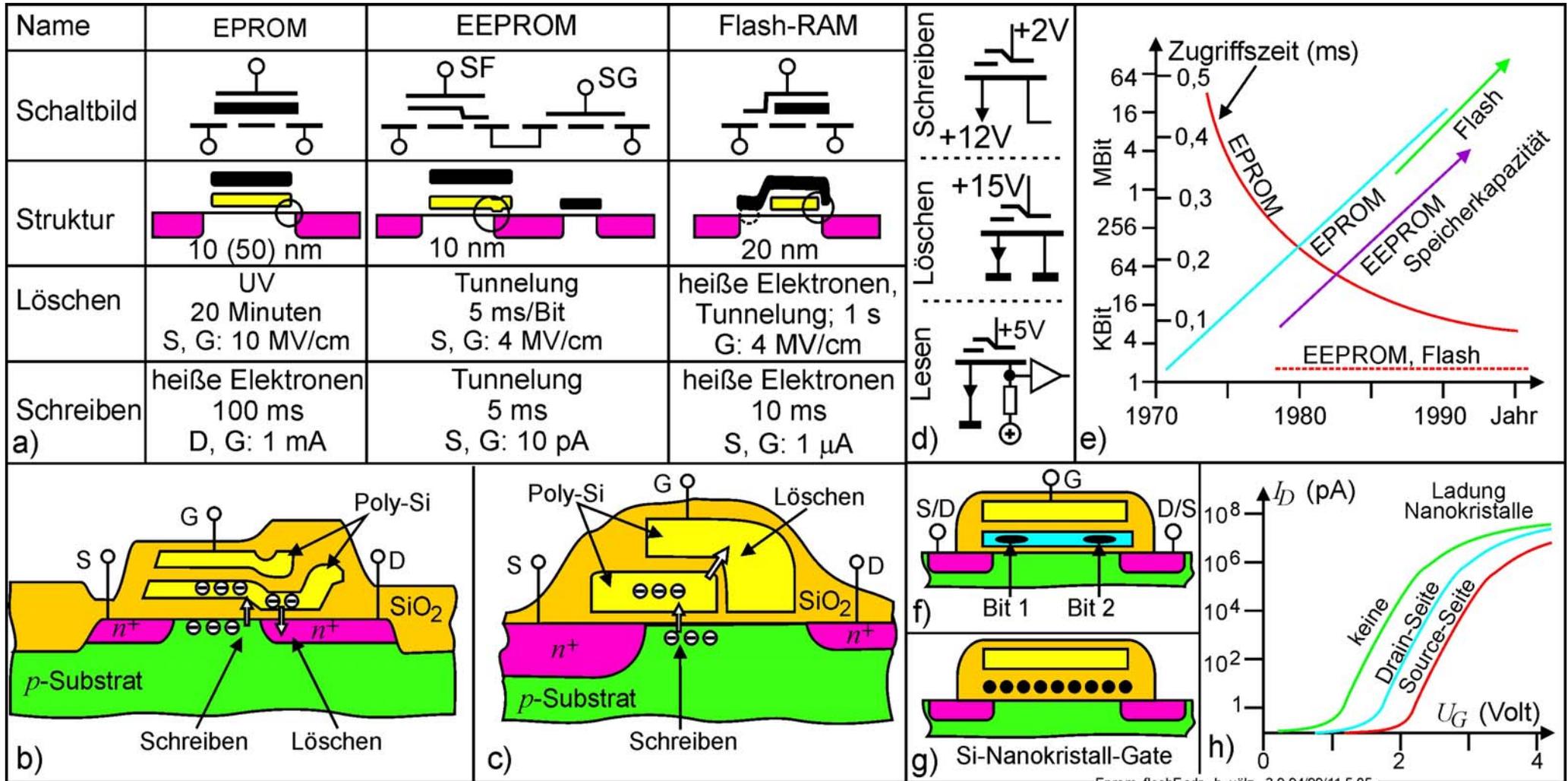
Erst mit dem **Flash** war eine *schnelle Löschung* möglich. Doch sie erfolgt immer für ganze *Blöcke*.

# UV-löschbares EPROM



EPROM\_MNOSF.cdr h. vözl 16.8.94/12.5.05

# EPROM - EEPROM - Flash



Eprom\_flashF.cdr h. vözl 3.9.94/99/11.5.05

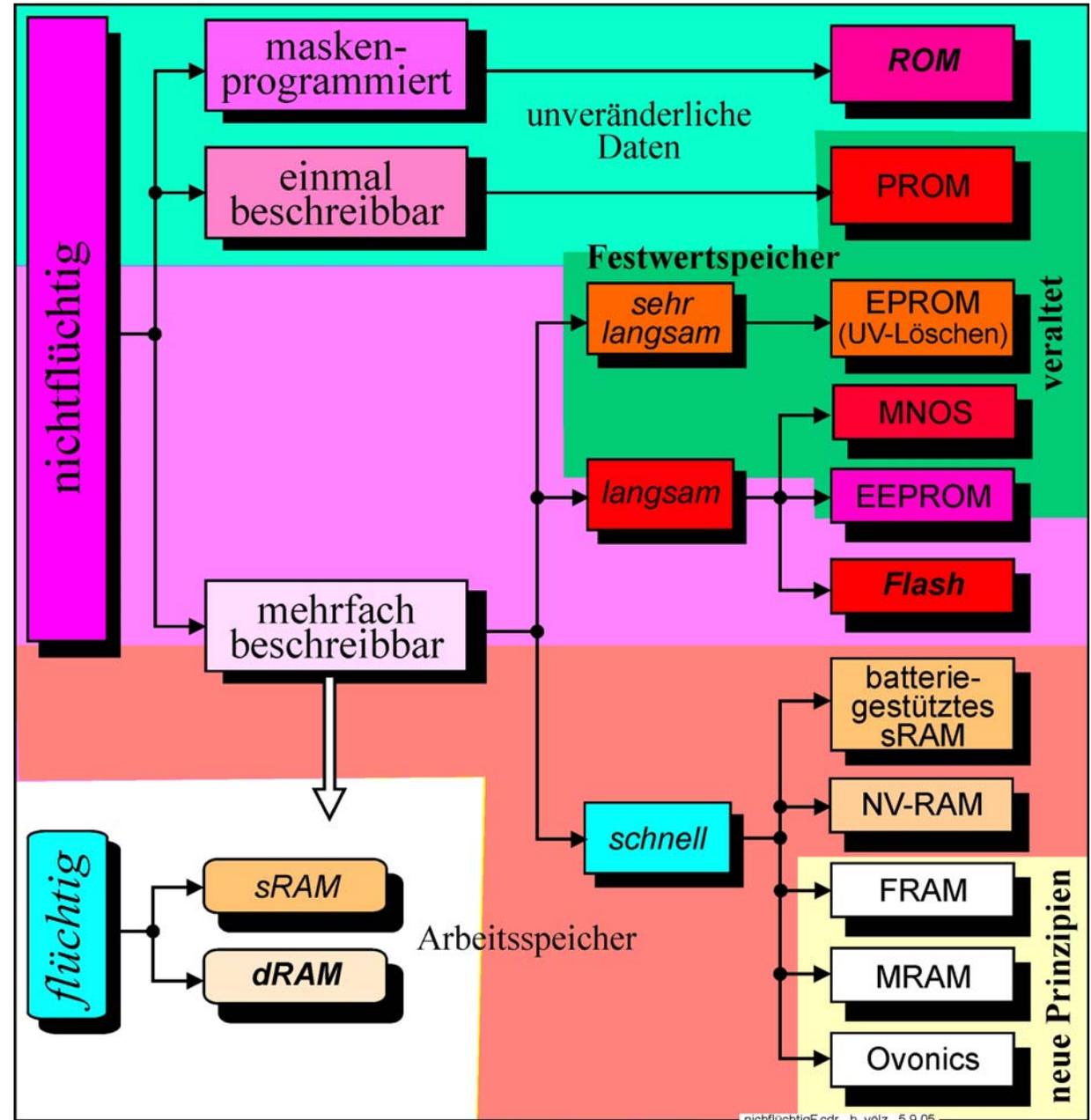
# Systematik der verschiedenen elektronischen Speicher

Die *nichtflüchtigen* Speicher behalten ihre Information auch dann, wenn die Betriebsspannung ausfällt, abgeschaltet wird.

Die *flüchtigen* Speicher verlieren beim Ausfall der Spannung ihre Information. Bei den *dRAM* ist zusätzlich ein *Refresh* notwendig,

Die *NV-RAM* (non-volatile) sind eine Kombination aus dRAM und Flash (EEPROM).

Die *gelb* hinterlegten Speicher **FRAM**, **MRAM** **Ovonisc** usw. sind eventuelle künftige Speicher. Sie sind nichtflüchtig und beruhen auf Ferroelektrizität, Magnetismus oder amorph  $\leftrightarrow$  kristallin (etwa wie die CD-RW). Es gibt noch weitere zukünftig mögliche Speicher, z. B. organische, Holographie und Tieftemperatur.



# Speicherschaltungen

Eine kleine Auswahl spezieller Speicher und Speicherschaltungen betrifft

**Buffer** oder Puffer = kleiner aber schneller Zwischenspeicher, der nur vorübergehend Daten aufnimmt, sowie ähnlich arbeiten auch **Register**.

Weitere Bausteine sind: **Cache**, **Stack** und **Assoziativspeicher**.

## Cache (*englisch* Versteck, geheimes Lager)

Bei Rechnern werden Befehle und Daten in recht **unregelmäßiger Adress-Abfolge** benötigt.

Der häufige Adresswechsel bedeutet beachtliche **Verzögerungen** (Zeit) für die Adresssuche und den Zugriff.

Mit einem Cache lässt sich das verkürzen, indem er immer größere Adressbereiche von Daten voraus bereithält.

Dazu benötigt er aber spezielle Programmierung, die ein „**richtiges Vorausdenken**“ (**look ahead**) bewirkt.

Bei der Nutzung müssen unterschieden werden:

- **Hit** (*englisch to hit* treffen, bestätigen schlagen) vorhandene Daten und
- **Miss** (*englisch to miss* vermissen, fehlen, verpassen), geforderte Daten nicht vorhanden

Das Verhältnis beider ist der „Wirkungsgrad“ des Cache

Er nimmt hängt von der Cache-Programmierung ab und nimmt begrenzt mit der Speicherkapazität zu.

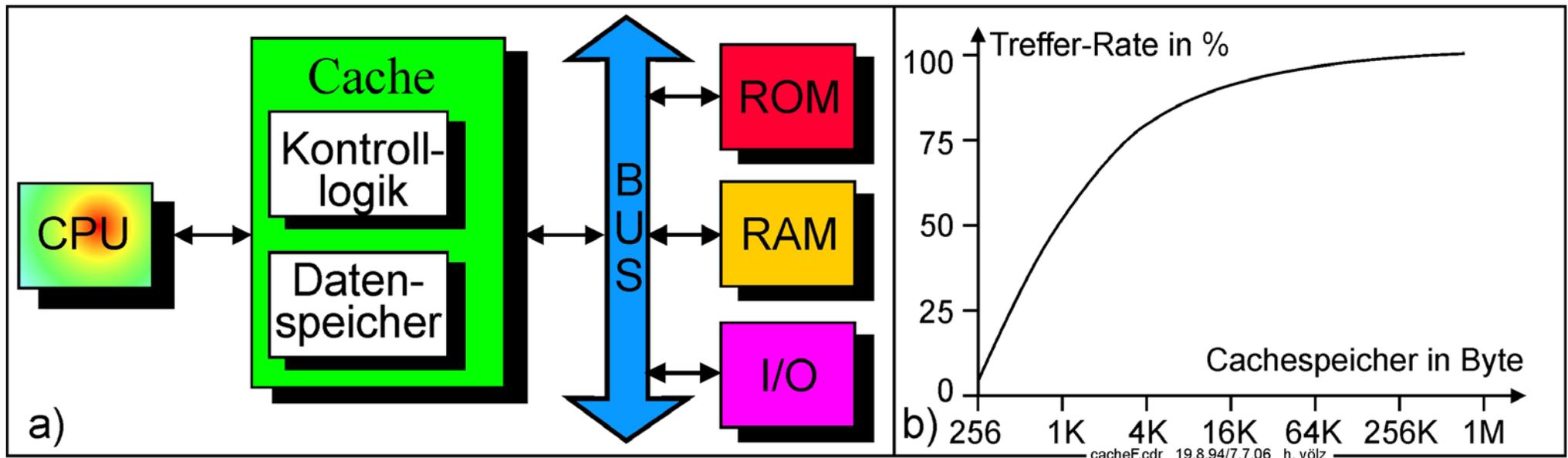
Zur Erhöhung der Hit werden häufig mehrere Cache hintereinander geschaltet.

p. s. Miss: nicht verwechseln mit Miss als Frau.

# Cache

Trefferrate = Verhältnis der erfolgreichen Hit zur Anzahl aller Zugriffe

Zu beachten ist, dass ein sehr großer Cache keinen wesentlichen Gewinn mehr bringt.



# Stack

*Englisch* Stapel, Haufen, Warteraum, Miete, Schacht, Schornstein

Im Deutschen ist auch der Begriff *Kellerspeicher* gebräuchlich (warum ist unklar)

Sein Arbeitsprinzip ist LIFO (last **in** first **out**), also ähnlich wie bei Zettel- oder Teller-Stapel.

Das Prinzip lautet: Was zuletzt abgelegt wurde muss zuerst wieder entnommen werden.

Von ihm gibt es keine echte Hardware-Variante, er wird immer programmiert.

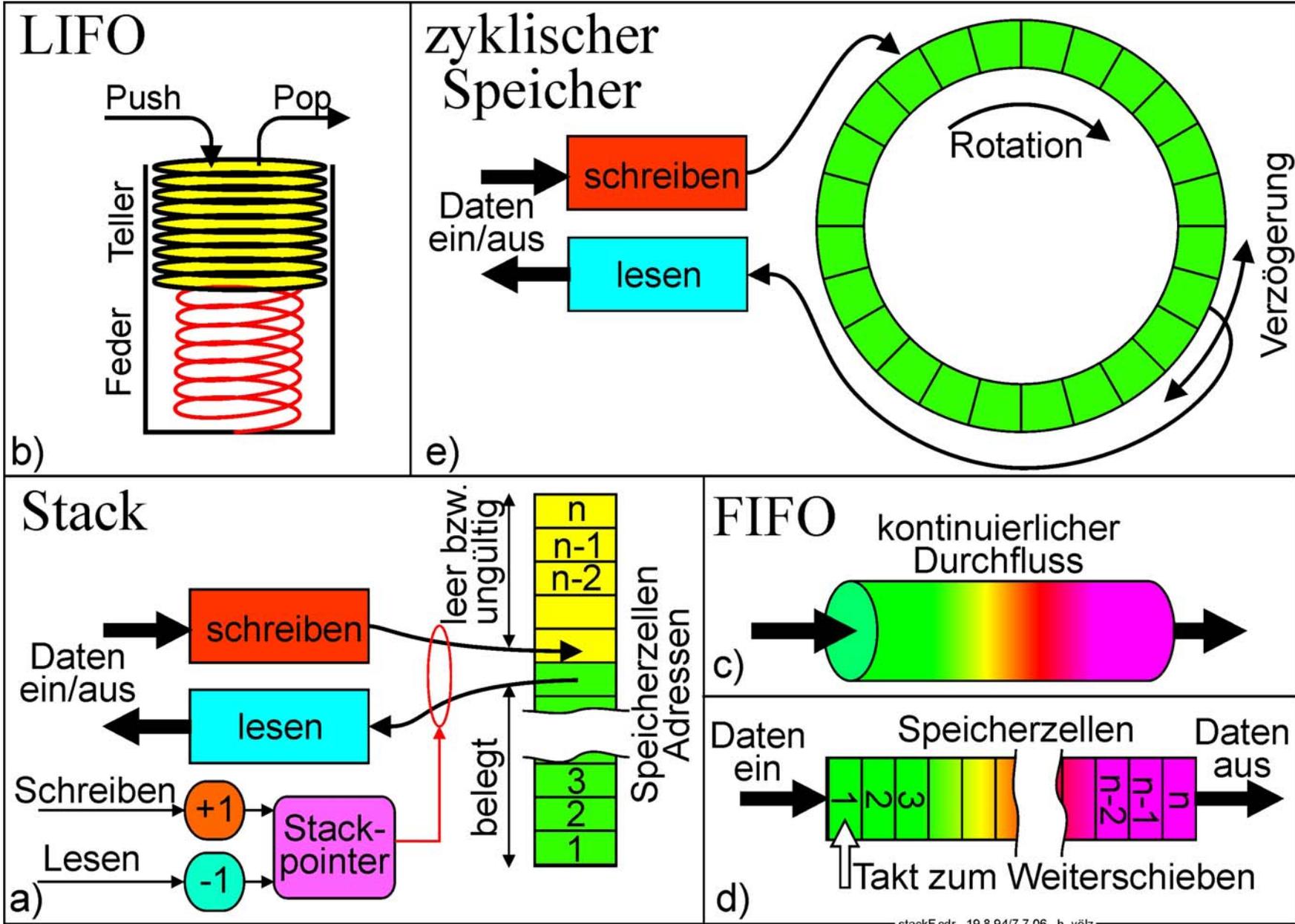
Meist werden mehrere Stack parallel benötigt, bei der Sprache FORTH sogar 4.

Eine Sondervariante ist ein *zyklischer Speicher*.

Zum Vergleich: Register usw. benutzen im Gegensatz zum Stack das FIFO (first in - first out)

Erfunden ist der Stack von WILHELM KÄMMERER (1905 – 1994) in seiner Dissertation.

KÄMMERER entwickelte 1955 bei Zeiß Jena OPREMA (**Optik-Rechen-Maschine**) auf Relais-Basis.



# Assoziativspeicher

**assoziativ** *lateinisch ad-* hin, zu und *socius* Gefährte

In der Mathematik ist das assoziative Gesetz (Verbindungsgesetz) schon lange bekannt:

$$a+b+c = (a+b)+c = a+(b+c) \text{ bzw. } a*b*c = (a*b)*c = a*(b*c).$$

Ab dem 19. Jh. vermutet die Psychologie, dass wir Wissen über inhaltliche Zusammenhänge (assoziativ) erinnern.

Angewendet wird das u. a. in der Gestalttheorie  $\Rightarrow$  Verknüpfung von Begriffen, Wörtern und Gedächtnisinhalten.

Assoziative technische Speicher = **CAM** (content addressed memory) wären wichtig für große Datenbestände.

Die Abfrage erfolgt dann **nicht sequentiell** über Adressen sondern die **parallel über die Inhalte** mittels einer Maske

Z. B. sei ein Ausschnitt einer Personendatei gemäß der folgenden Tabelle gespeichert.

Nun werde ein **unverheirateter Mann** im **Alter** von 25 bis 35 Jahren mit dem **Hobby** Foto gesucht.

In der Tabelle gibt es weitere Personendaten, wie Kinder, Name usw. die für die Suche zunächst unwichtig sind

Sie werden in der Maske mit „×“ als „*don't care*“ (englisch nicht drum kümmern) markiert und für die Suche ausgeblendet.

Alle dann noch „gültig“ bleibenden Daten werden in der letzten Spalte mit „\*“ markiert.

Dieser Prozess lässt sich vollständig parallel durchführen. Dann werden nur die markierten Zeilen sequentiell benutzt.

Nr.	Name	weiblich	Alter	ledig	Kinder	Hobby	Marke
45	Meyer	0	30	0	0	Foto	
46	Müller	1	35	1	1	Briefmarken	
47	Schulze	0	27	1	2	Theater	
48	Altmann	0	40	1	0	Foto	
49	Schmidt	1	30	0	2	Reisen	
50	Lindner	0	28	1	0	Foto	*
Maske	×	0	25 – 35	1	×	Foto	

Genau dieses Prinzip gibt es für die technische Speicherung seit den 50er Jahren eine Hardwarestruktur (s. folgendes Bild).

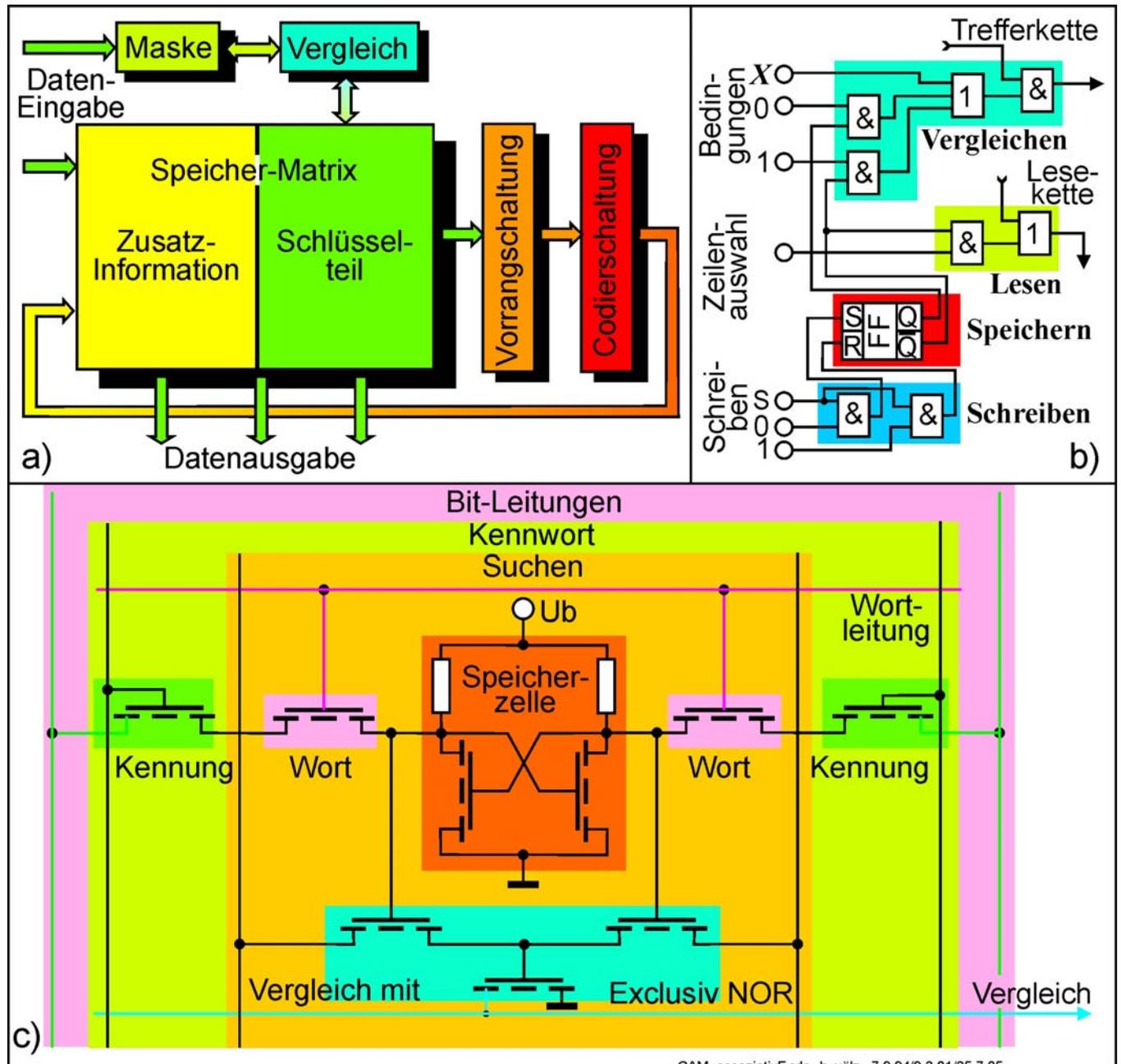
Sie verlangt etwa 10 Transistoren je Bit und viele Pin nach außen.

So konnte sie sich trotz einzelner Anwendungen bisher leider nicht durchsetzen.

a) Struktur des *assoziativen Speichers*.

b) Verschaltung der einzelnen Speicherzellen.

c) Aufbau einer Speicherzelle.



CAM\_assoziativF.cdr h. vözl 7.9.94/9.2.01/25.7.05

# Zur geschichtlichen Entwicklung der Speichertechnik

Die technische Speicherung beginnt mit der *Menschwerdung*.

Indem *Werkzeug aufgehoben* wurde, war in ihm doch seine Funktion festgelegt (ROM).

Es kamen dann nacheinander *Bilder, Schrift, Buckdruck, Fotografie, Film und Schallplatte* hinzu.

Die elektronische Speicherung begann nach in den 20er mit der elektrischen *Schallaufzeichnung* und dem *Rundfunk*.

Den entscheidenden Beginn forderte ab den 40er Jahren die *Rechentechnik* und zwar gleich *digital*.

Von da an kamen dann immer neue Verfahren, wie *Tonband, Ferritkerne, Trommel- und Plattenspeicher* hinzu.

Eine neue Qualität brachten der *Transistor* und die *integrierte Schaltung* und ab 1982 die *CD*.

Das Spektrum wurde dabei immer vielfältiger und insbesondere nahmen die *Speicherdichte* und *Kapazität* steil zu.

Lange Zeit war die *Speicherkapazität* dennoch ein wesentlicher *Engpass*.

Es wurde trotz allem immer *mehr verlangt als bereitgestellt* werden konnte.

Das wurde ab etwa *1995 durchbrochen*: Es stand mehr zur Verfügung als in den meisten Fällen gebraucht wurde.

Mit den folgenden Bildern ist an einigen typischen *Kenndaten* die *Entwicklung* im Detail aufgezeichnet.

Z. Z. besteht eine große Unsicherheit darüber, welchen neuen Entwicklungen sich dennoch durchsetzen könnten.

Einige Ansätze und erfolg-versprechende Lösungen folgen nach den Daten.

Es gibt dazu viel mehr Daten, z. B.: [r-h-voelz.de/pdf/HU/SpeichernFundament.pdf](http://r-h-voelz.de/pdf/HU/SpeichernFundament.pdf) und [r-h-voelz.de/PDF/TU/DatenSpeichern.pdf](http://r-h-voelz.de/PDF/TU/DatenSpeichern.pdf)

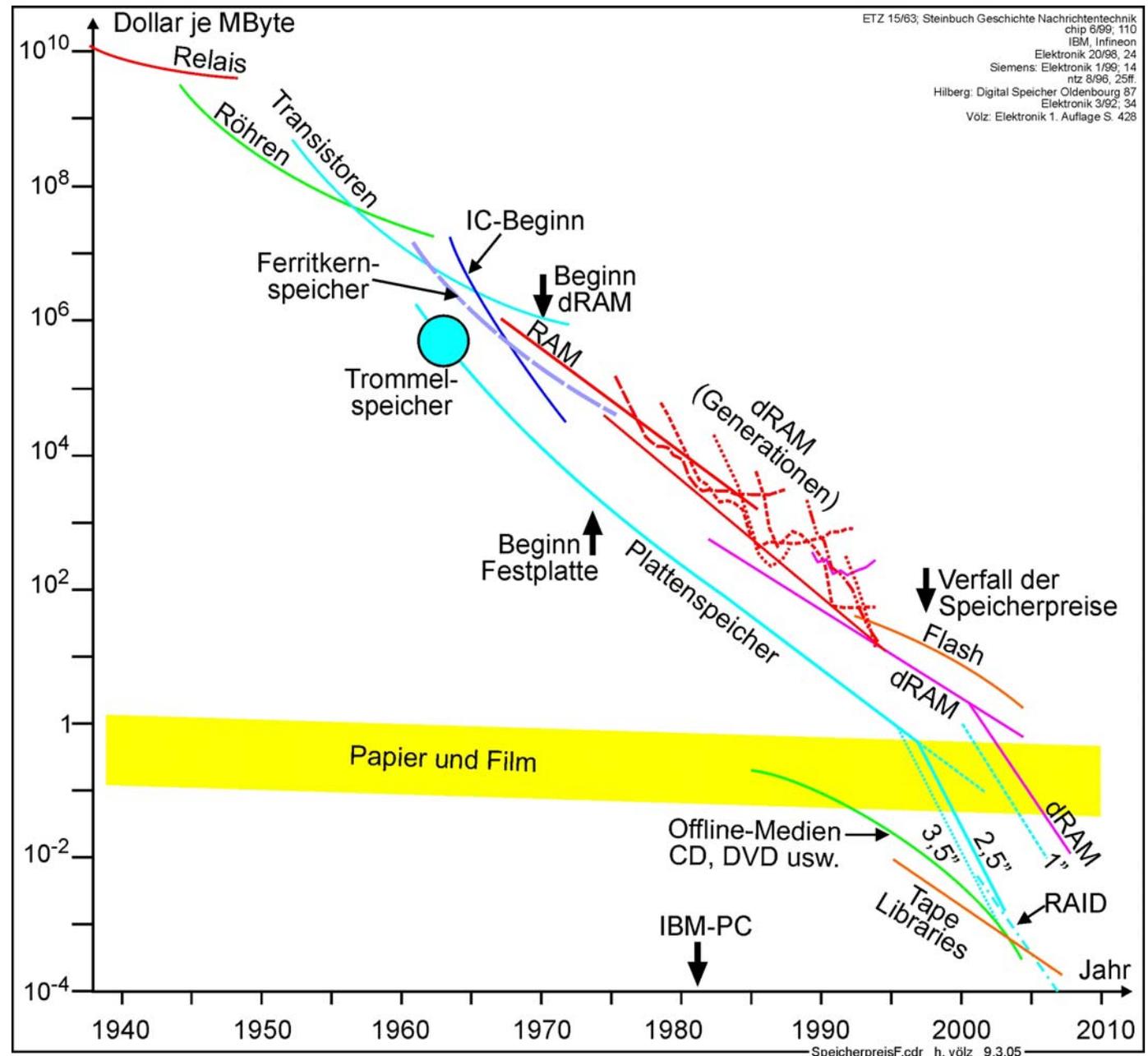
Der **Bitpreis** für Speicher ist im Laufe von über 60 Jahren so steil abgefallen, wie wohl kein anderes technisches Produkt.

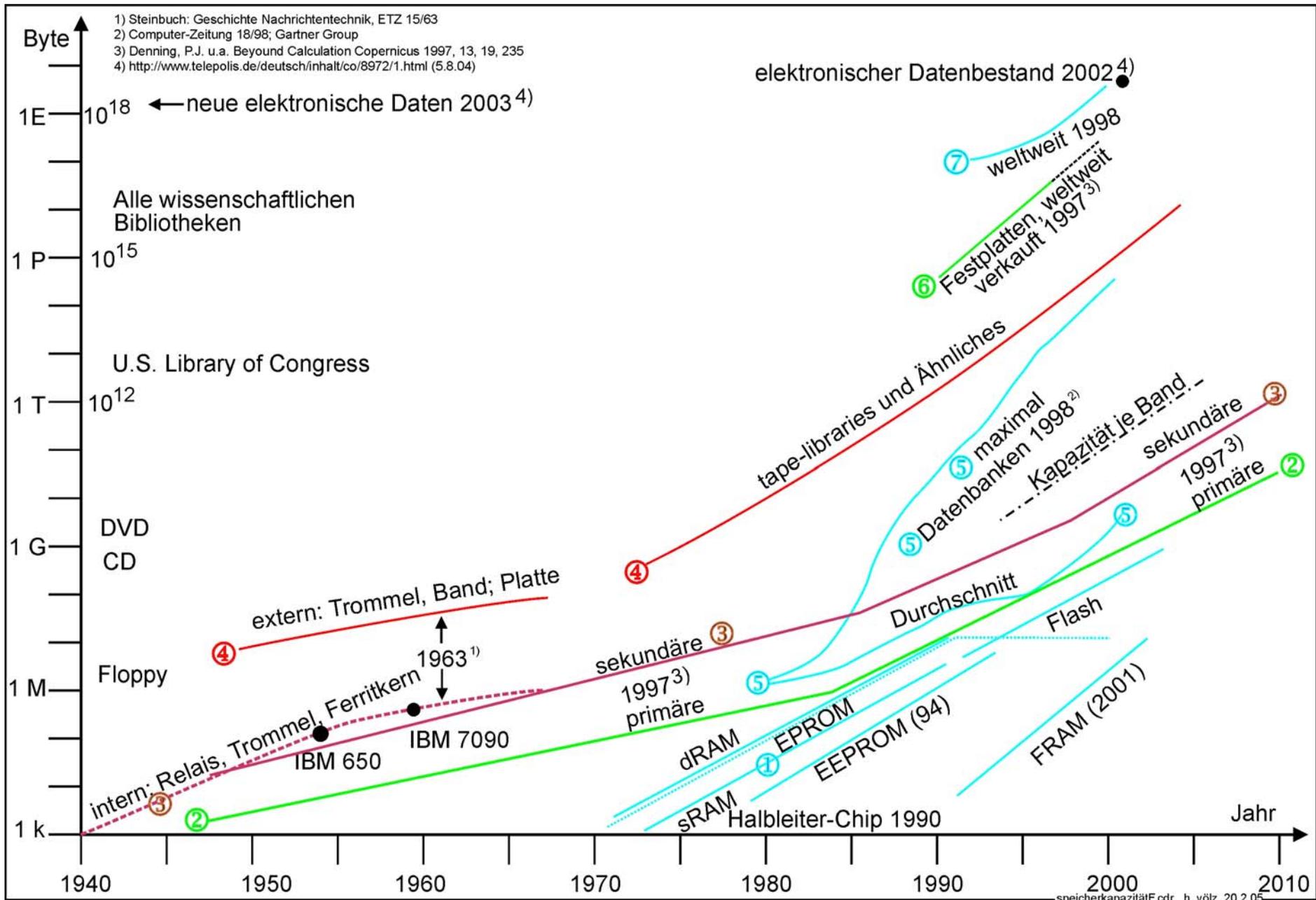
Natürlich sind dabei die Umrechnungen unsicher, da sich auch der Geldwert geändert hat. Doch bei seiner Berücksichtigung wäre der Abfall noch steiler.

In dieser langfristigen Darstellung gilt verläuft die Preissenkung exponentiell und alle Speicherarten liegen etwa gleich auf.

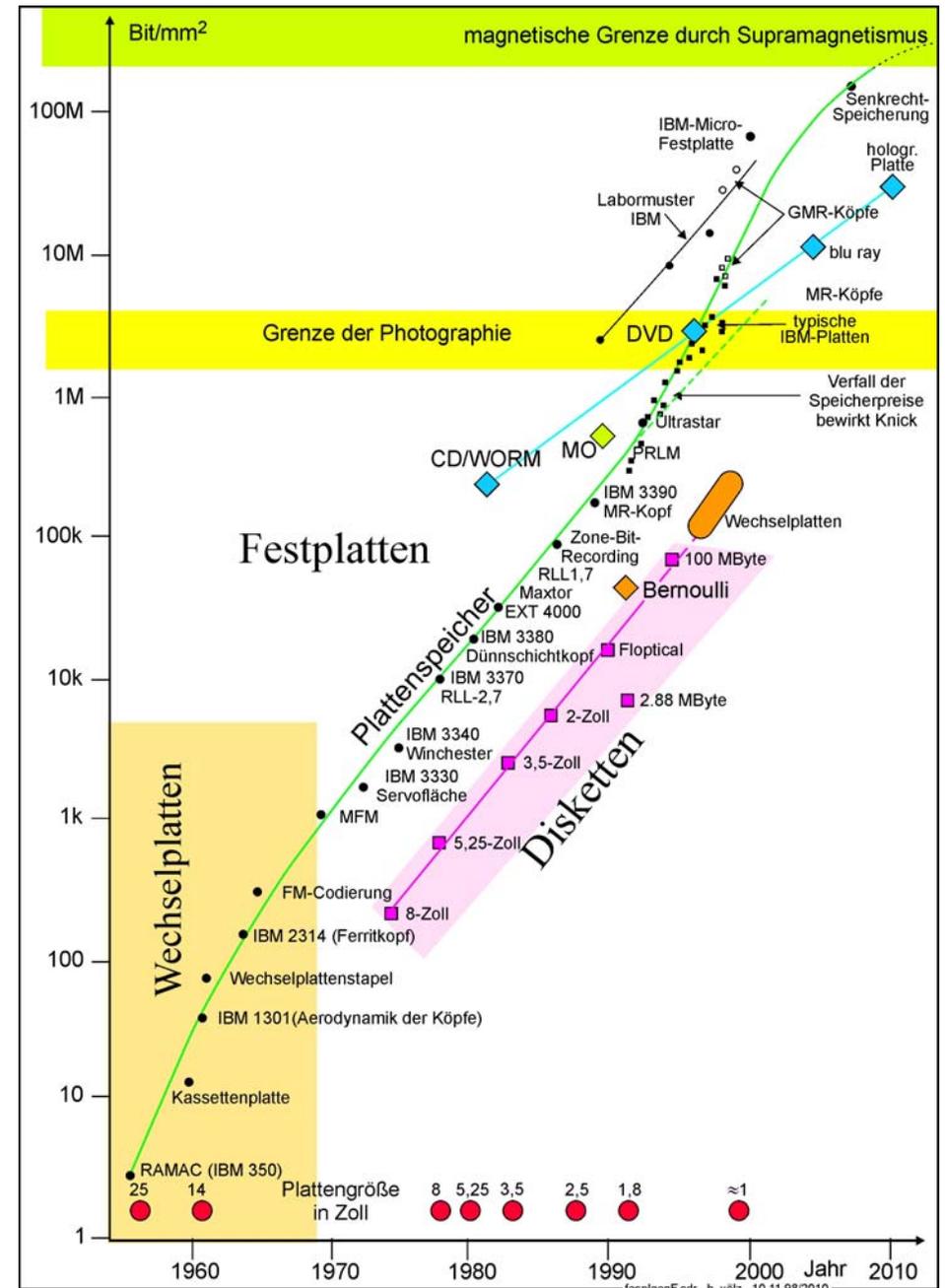
Auffällig ist weiter der zusätzlich versteilerte Abfall ab etwa 1995. Ab diesem Datum waren Speicher kein wesentlicher Engpass mehr. Es wurde immer wieder mehr Speicherkapazität produziert als benötigt und damit verkauft werden konnte.

Das folgende Bild zeigt die jeweils weltweit installierte (benutzte) **Speicherkapazität**.





Ganz wesentlich für die Entwicklung der Speicher ist das Volumen (die Fläche), welche für ein Bit benötigt wird. Auch dies *Speicherdichte* ist in den über 60 Jahren enorm steil geringer worden. Dadurch war es möglich immer mehr Bit in einem Volumen zu archivieren.

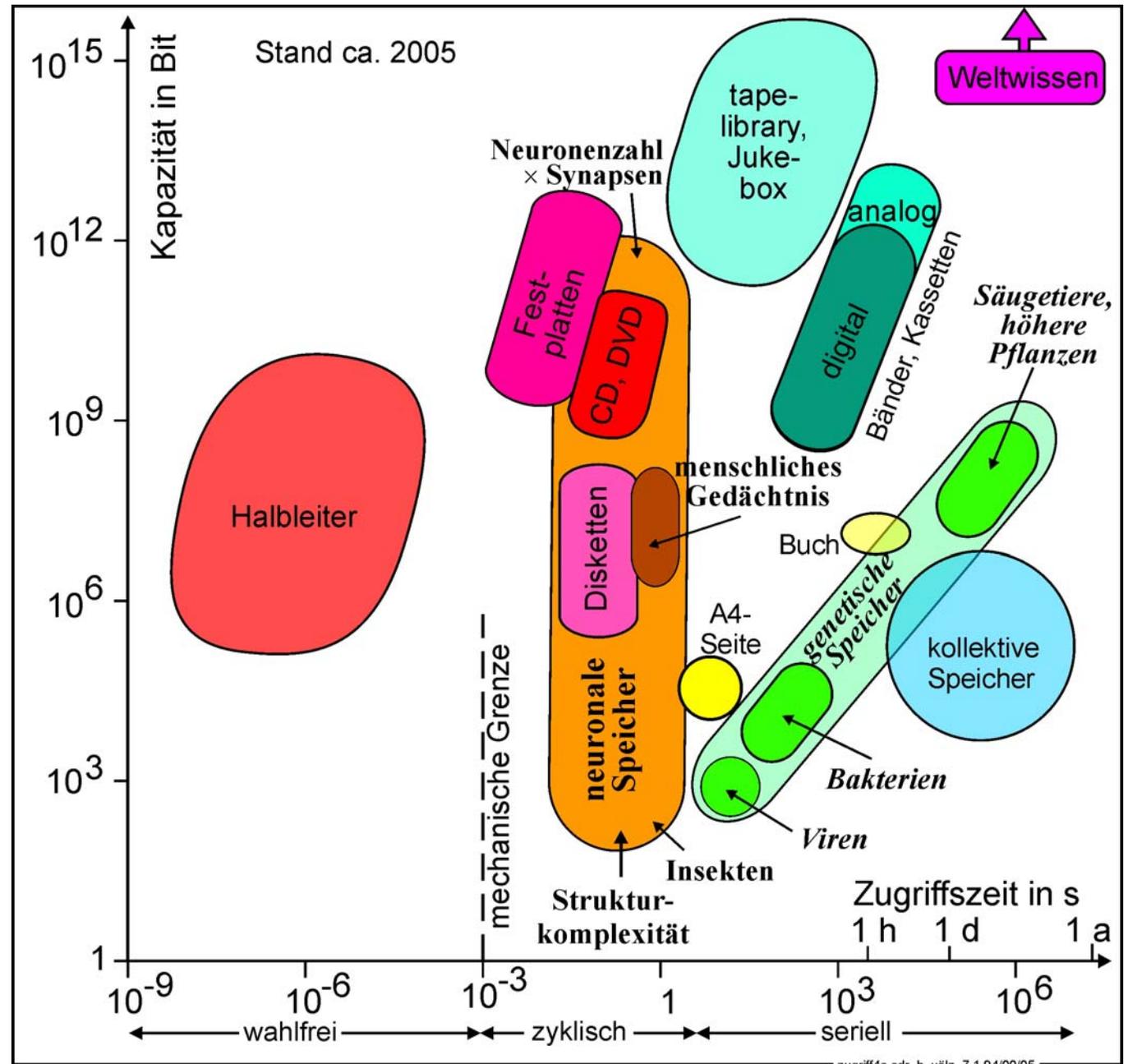


Es ist Auffällig, dass es für jede Speichertechnologie abgegrenzte Bereiche aus **Speicherkapazität und Zugriffszeit** gibt.

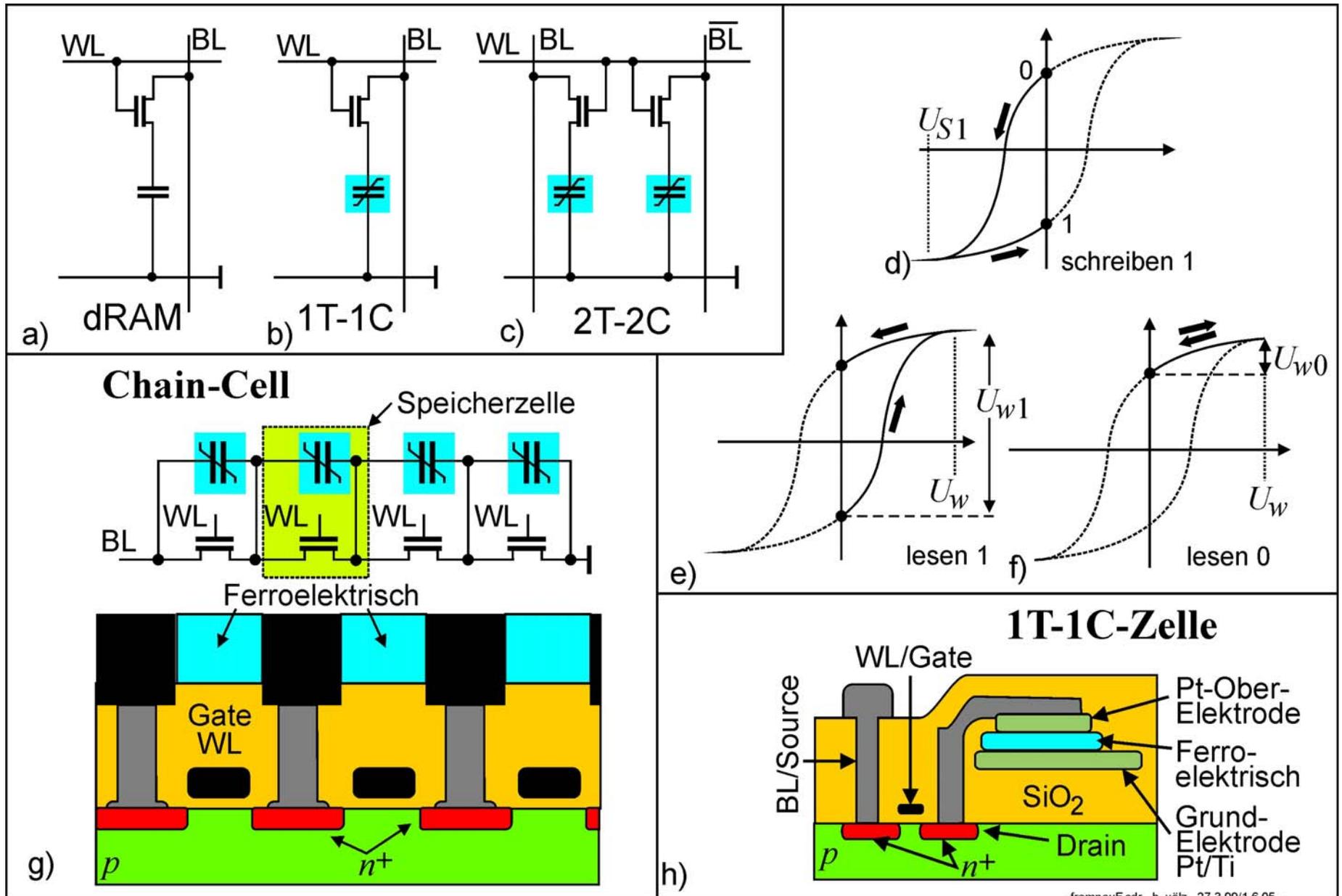
Hierdurch sind immer wieder veränderte Hierarchien aus den verschiedenen Technologien entstanden.

Erst im letzten Jahrzehnt beginnt eine Verringerung der Anzahl der Hierarchiestufen. Zurzeit erfolgt u. a. die Aussonderung der optischen Medien CD, DVD Blu Disk und damit eine Reduzierung der Hierarchie auf Elektronische Speicher, Festplatten und Bänder (nur noch in den Archiven).

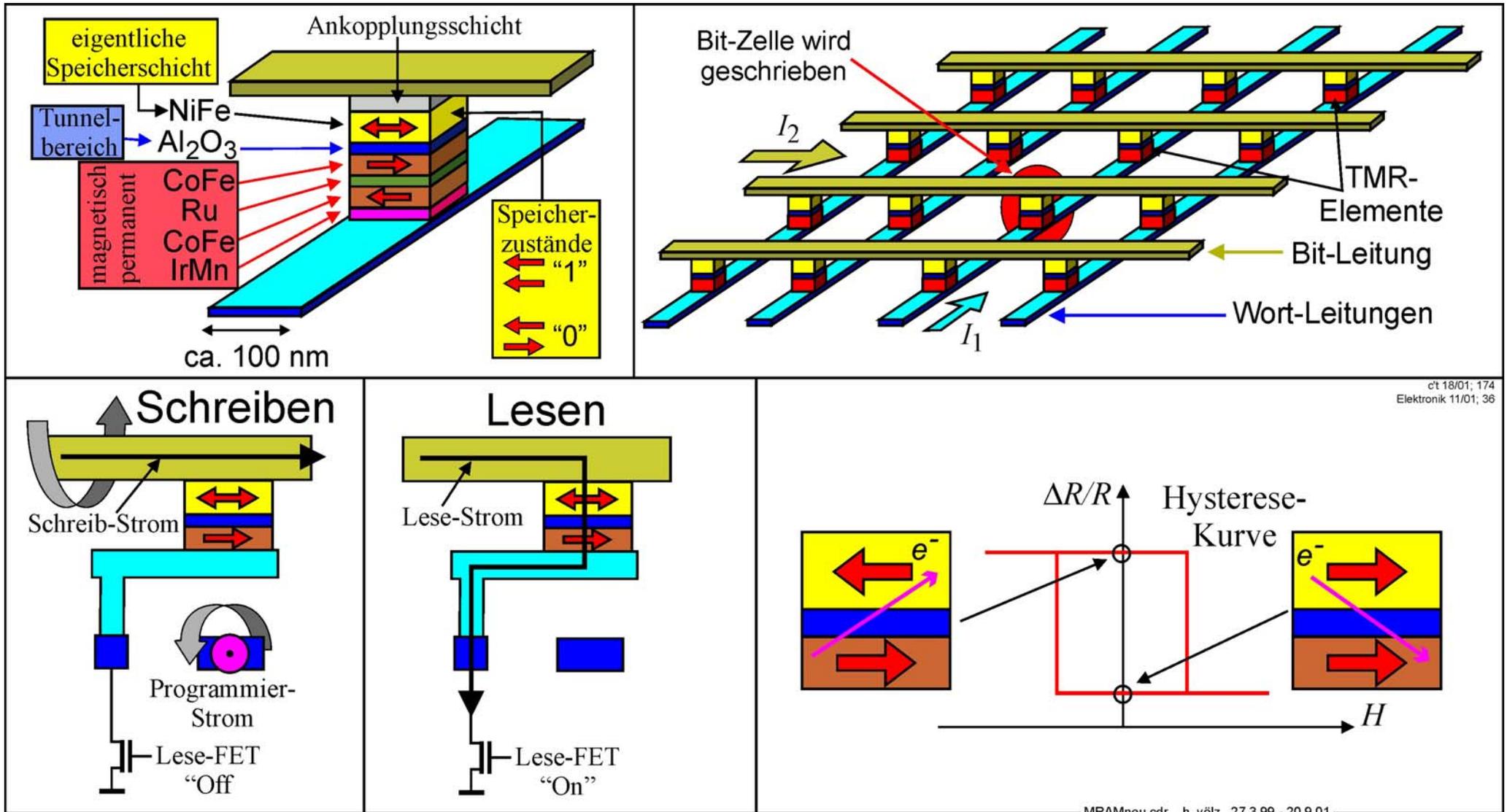
Es ist schon abzusehen, dass in einigen Jahren auch die Festplatten verschwinden werden, s. SSD (solid state disk).



# Der ferroelektrische Speicher FRAM nutzt die Hysterese einiger Materialien



# Der MRAM benötigt komplexes magnetisches Material und besitzt hervorragende Eigenschaften



# Eventuell ermöglicht die Quantenphysik mit dem QBit völlig neue Möglichkeiten

<p>Möglichkeiten zur beidseitigen Kennzeichnung einer runden Scheibe.</p>		<p><b>Modell Traumaskop</b></p> <p>enthält Seiten A und B, können jeweils 1 oder 2 besitzen, auf A-Seite ist Zahl größer + kursiv. Möglich sind die vier Varianten:  <math>AB = \{11, 12, 21, 22\}</math>.</p>
<p><b>Klassische Informationstheorie</b>          verlangt zur Entscheidung Hinlegen + Umdrehen          = 2 Bit</p> <p>für <b>11</b></p>	<p><b>Korrespondenzmodell</b>          nach N. Bohr:          Traumaskop rotiert sehr schnell, beide Seiten sind zugleich sichtbar.  <math>\Delta t \rightarrow \infty</math>.          Zur Entscheidung genügt          = 1 Blick</p> <p>z.B.: <b>11</b> oder <b>12</b></p>	<p><b>Quantentheorie</b>          Traumaskop rotiert (Spin) mit sehr kleiner Masse bei sehr tiefer Temperatur. Es sind immer beide Zustände zugleich vorhanden (verschränkt).  <math>\Psi = c_1 A\rangle + c_2 B\rangle</math>          Messung erfolgt mit Lichtblitz  <math>\Delta t \ll 1/\omega</math>          Foto zeigt nur Zustand <math>c_{1m}, c_{2m}</math>. Lichtblitz zerstört den Zustand <math>\Psi</math>, exakte Wiederholung unmöglich.          = 1 Qubit</p>

QuBitF.odr h. vözl 20.12.05

# Gliederung

1. Grundlagen
2. Kombinatorische Bausteine
3. Die drei typischen Digitalschaltungen
4. Speicher
- 5. Sequentielle Schaltungen**
6. Automaten und Rechner
7. Ergänzende Systeme
8. Berechenbarkeit

# Einführung

Bei sequentiellen Schaltungen sind Speicher und kombinatorische Schaltungen zu einer Einheit zusammengefügt. Die gespeicherten Daten bestimmen dabei den jeweiligen **Zustand** der Schaltung. Mit jedem von ihnen existiert ein unterschiedliches Verhalten der Schaltung. Die Eigenschaften von sequentiellen Schaltungen sind so „zeitabhängig“ und werden mit drei Kenngrößen beschrieben.

- Eingangswerte  $E_i$
- Ausgangswerte  $A_i$ .
- Innere Zustände  $Z_i$ , welche sich aus denen der Speicher ableiten.

Das **Verhalten** lässt sich wieder mittels kleiner Boxen mit **Tasten** und **Leuchten** veranschaulichen.

Die **Eingangswerte** sind dabei die **Tastenbetätigung**.

Die **Ausgangswerte** das **Leuchten der Lämpchen**.

Die inneren **Zustände** sind dabei **nicht** immer direkt zugänglich, gerade das macht die Schaltungen so „undurchsichtig“

Viele sequentielle Schaltungen sind **Automaten**, die auf die Eingangswerte je nach dem inneren Zustand reagieren. Dabei gibt es **zwei Grundtypen**, die unterschiedliche Zusammenhänge besitzen.

Beim **MEALY-Automaten** sind nur die Eingangs- und Ausgangswerte zugänglich, der Zustand bleibt verborgen

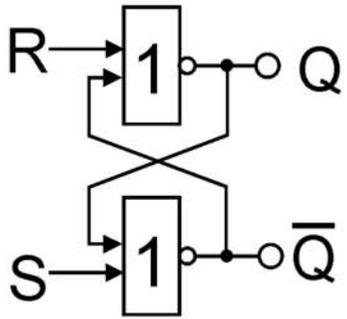
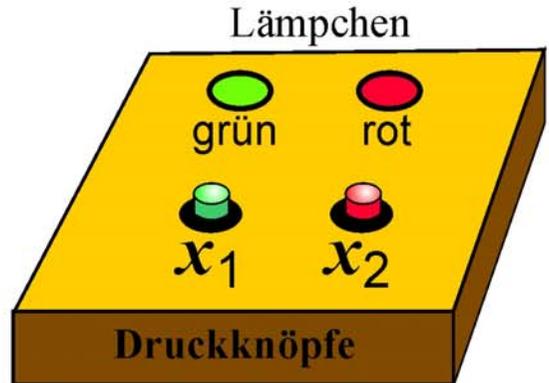
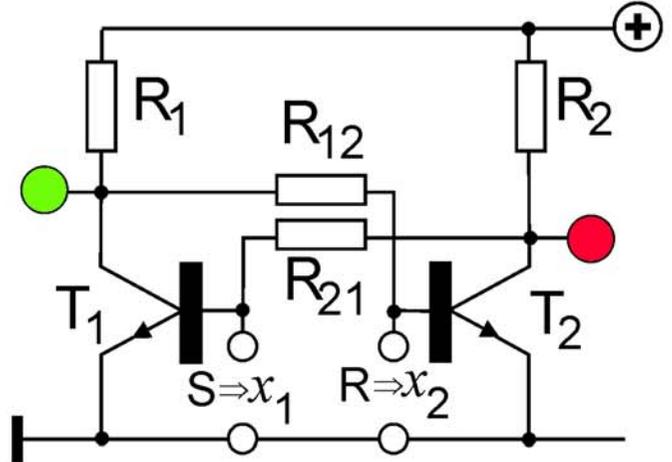
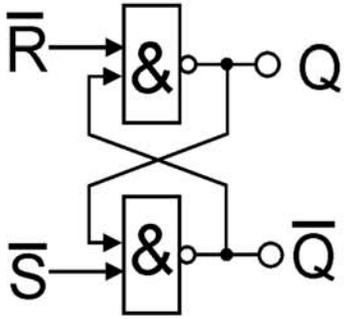
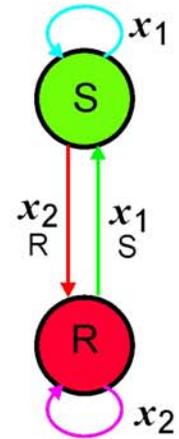
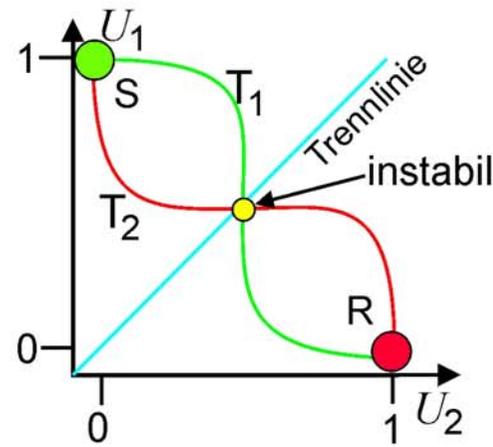
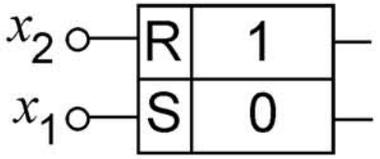
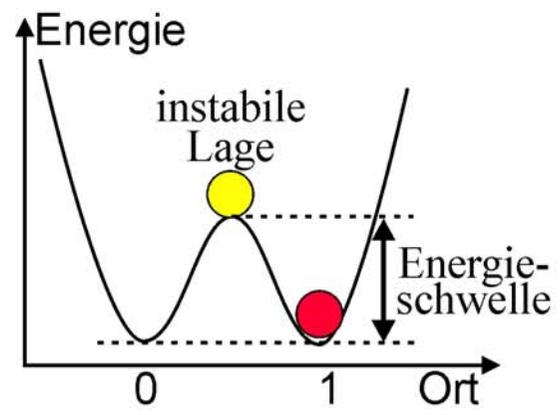
Beim **MOORE-Automaten** sind mit den Ausgangswerten die internen Zustände mittelbar gekoppelt.

Besonders komplexe sequentielle Schaltungen sind **Rechner** und darin vor allem die **CPU** (central processor unit).

# Als noch sehr einfache sequentielle Schaltung kann das Flip-Flop betrachtet werden

## Flip-Flop

(bedingt) irreversibel

	<p>Lämpchen</p>  <p>Druckknöpfe</p>	
		
		<p>Automat3.cdr H. Völz 1.1.12</p>

# Das sind die beiden Automaten-Typen

### Mealy-Automat

		Eingaben			
		$E_1$	$E_2$	$E_y$	$E_m$
Zustände	$Z_1$				
	$Z_2$				
	$Z_x$			$Z_{neu}/$ $Ausgabe$	
	$Z_n$				
	$Z_n$				

### Beispiel-Automat

Lämpchen

Druckknöpfe

### Moore-Automat

		Eingaben			
		$E_1$	$E_2$	$E_y$	$E_m$
Zustände/Ausgaben	$Z_1/A_1$				
	$Z_2/A_2$				
	$Z_x/A_x$			$Z_{neu}$	
	$Z_n/A_n$				
	$Z_n/A_n$				

### Beispiel

	$E_1$	$E_2$	$E_3$
$Z_1$	1/g	0/g	1/r
$Z_2$	1/g	0/g	1/b

Ausgabe beim Übergang

### Beispiel

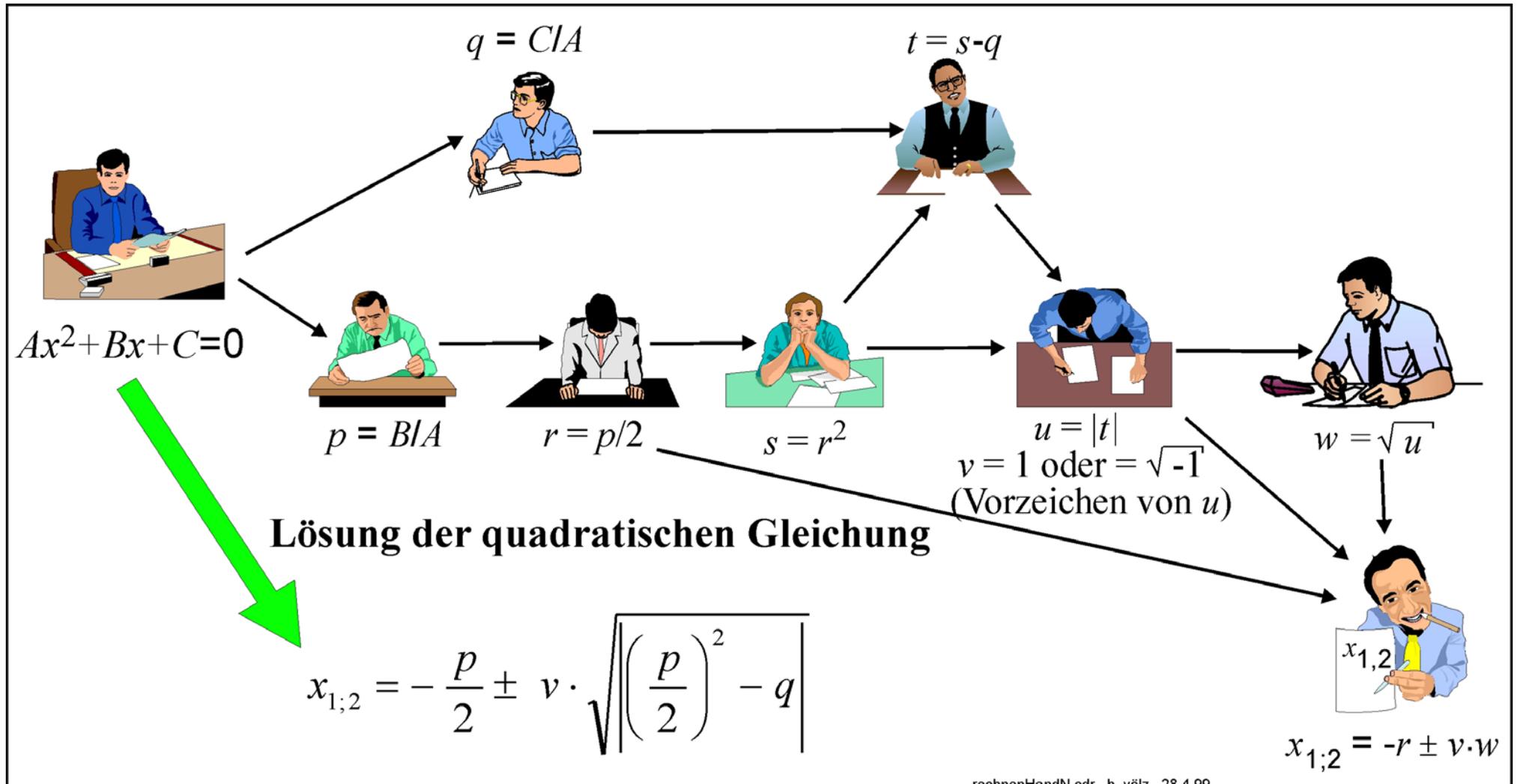
	$E_1$	$E_2$	$E_3$
$Z_1/g$	$Z_1$	$Z_2$	$Z_1$
$Z_2/r$	$Z_1$	$Z_4$	$Z_3$
$Z_3/b$	$Z_3$	$Z_2$	$Z_3$
$Z_4/g$	$Z_4$	$Z_4$	$Z_4$

Ausgabe entspricht aktuellem Zustand

# Gliederung

1. Grundlagen
2. Kombinatorische Bausteine
3. Die drei typischen Digitalschaltungen
4. Speicher
5. Sequentielle Schaltungen
- 6. Automaten und Rechner**
7. Ergänzende Systeme
8. Berechenbarkeit

# Berechnungsweg bei „Rechenknechten“ vor Einführung der elektronischen Rechner



# Der Aufbau von elektronischen Rechnern

Noch mehr als in den historischen Rechenbüros muss bei den elektronischen Rechnern *jede Berechnung* in möglichst *einfache und allgemeingültige Rechenschritte* zerlegt werden.

Dafür gibt es Grundfunktionen, deren Zusammenschaltungen alle erdenklichen Berechnungen ermöglichen.

Hierfür existieren u. a. folgende Grundschaltungen, dabei haben alle Größe eine vorgegebene Wortlänge, z. B. 8 Bit.

- Der *Vergleicher*, welche zwei Eingangswerte  $A$  und  $B$  auf  $>$ ,  $=$  oder  $<$  prüft und dafür je ein Bit ausgibt.
- Der *Halbadder*, der aus  $A$  und  $B$  die Summe  $S$  und den Übertrag  $\ddot{U}$  bildet.
- Der *Volladder*, der zusätzliche einen vorangegangenen Übertrag  $\ddot{U}_{-1}$  als Eingang berücksichtigt.

Diese und weitere Funktionen werden in einer *ALU* (arithmetic and logic unit) vereint.

Sie ist das Fundament aller Rechner und ist wiederum ein Teil der *CPU* (central processor unit)

Zu ihr gehören z. B. noch mehrere Register und Flagspeicher für Zwischenergebnissen sowie Ein-/Ausgabeeinheiten.

Alle Einheiten sind durch verschiedene interne *Bus-Systeme* verbunden.

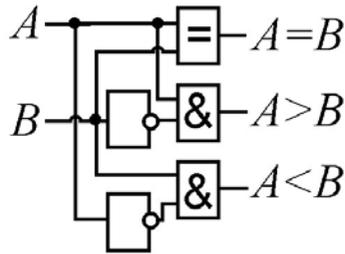
Eine von Takt betriebene *Ablaufsteuerung* regelt das richtige Zusammenwirken aller Einheiten.

*Steuer-, Daten- und Adressbus* stellen die Verbindung zu den anderen (externen) Einheiten des Rechners her.

Hierzu gehören Daten- und Programmspeicher, Ein- und Ausgabe-Geräte, wie Tastatur, Display und Drucker.

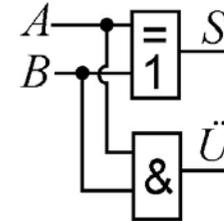
### Vergleicher

A	B	A>B	A=B	A<B
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

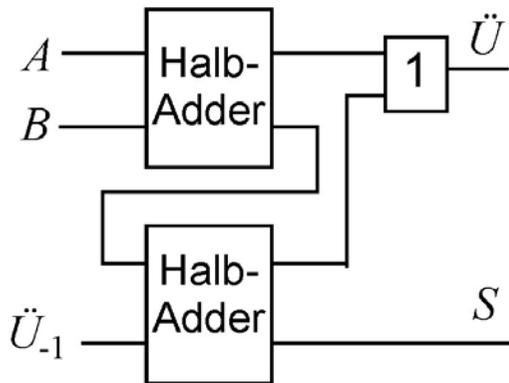


### Halb-Adder

A	B	S	Ü
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



### Voll-Adder

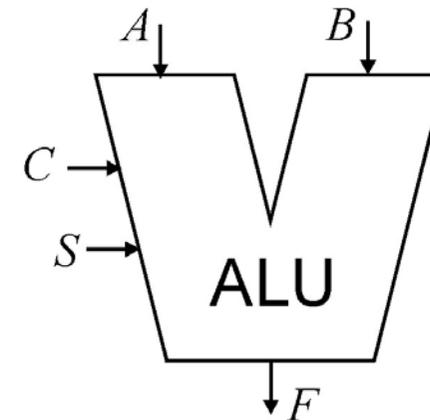


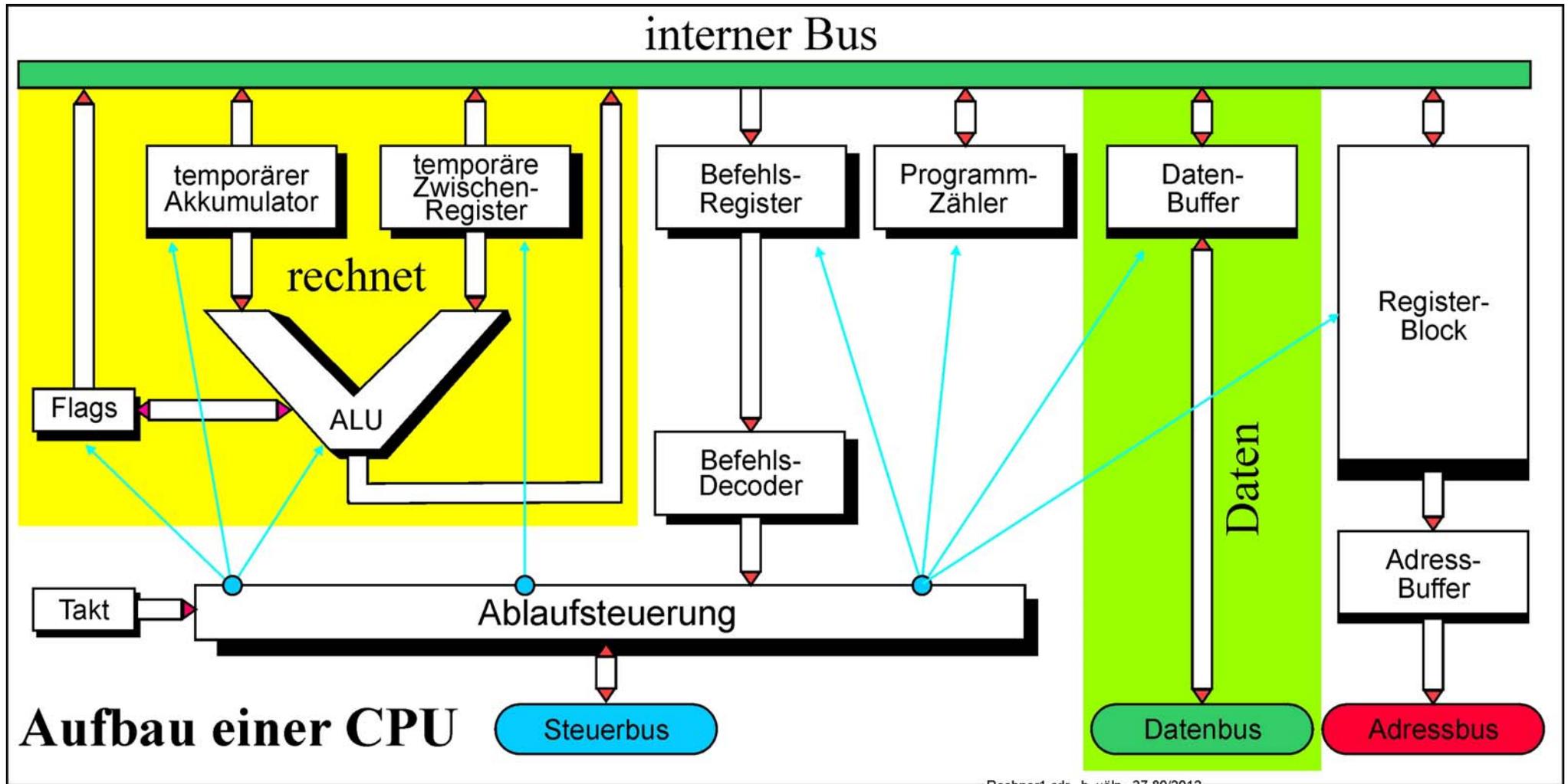
Ü <sub>-1</sub>	A	B	S	Ü
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

S <sub>0</sub>	S <sub>1</sub>	F	Bedeutung
0	0	A + B	Transfer: Inkrement
0	1	A + B + C	Addition
1	0	A + $\overline{B}$ + C	Subtraktion
1	1	A - 1 + C	Transfer: Dekrement

ALU vereinfacht

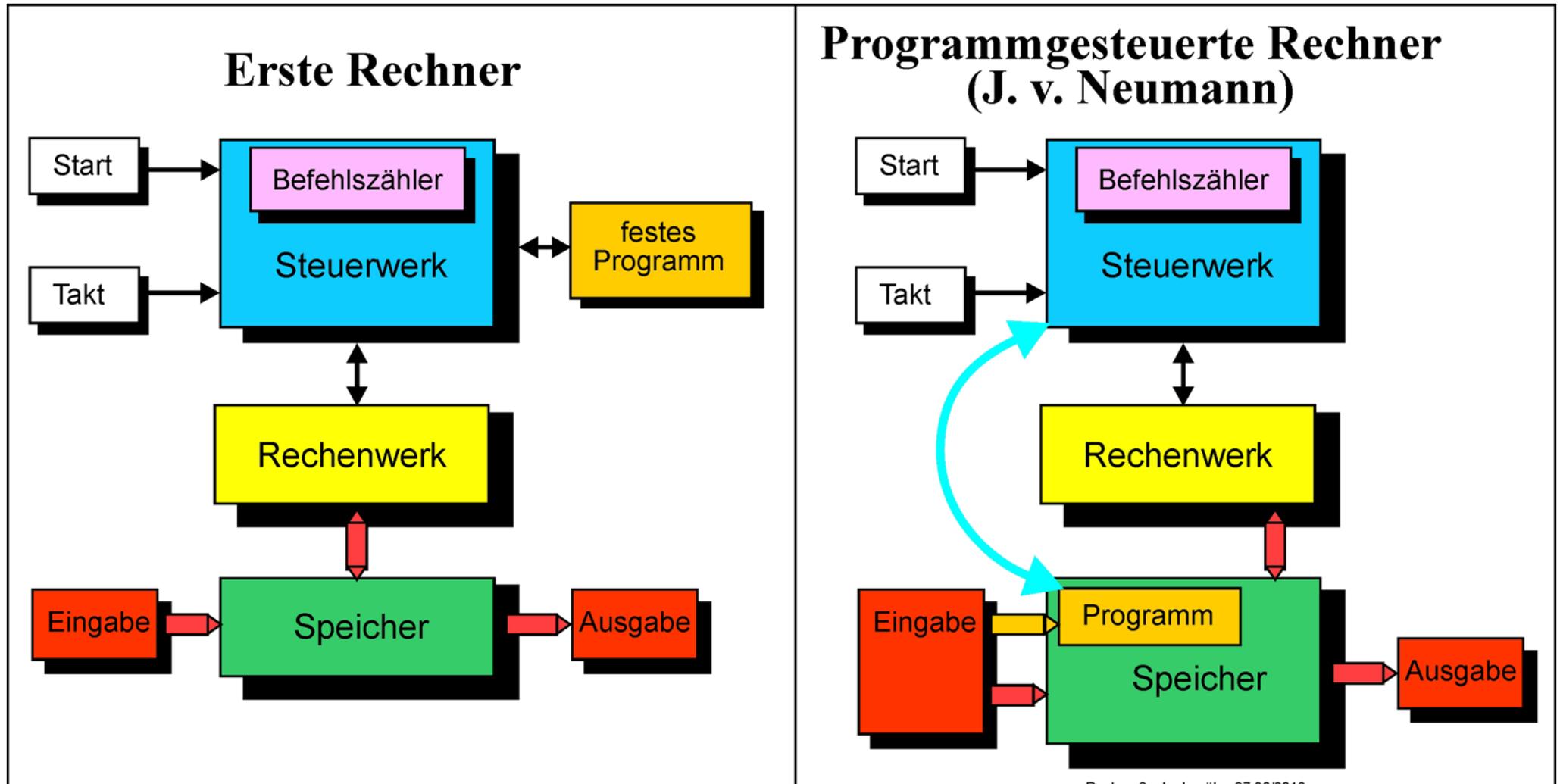
C = Compliment





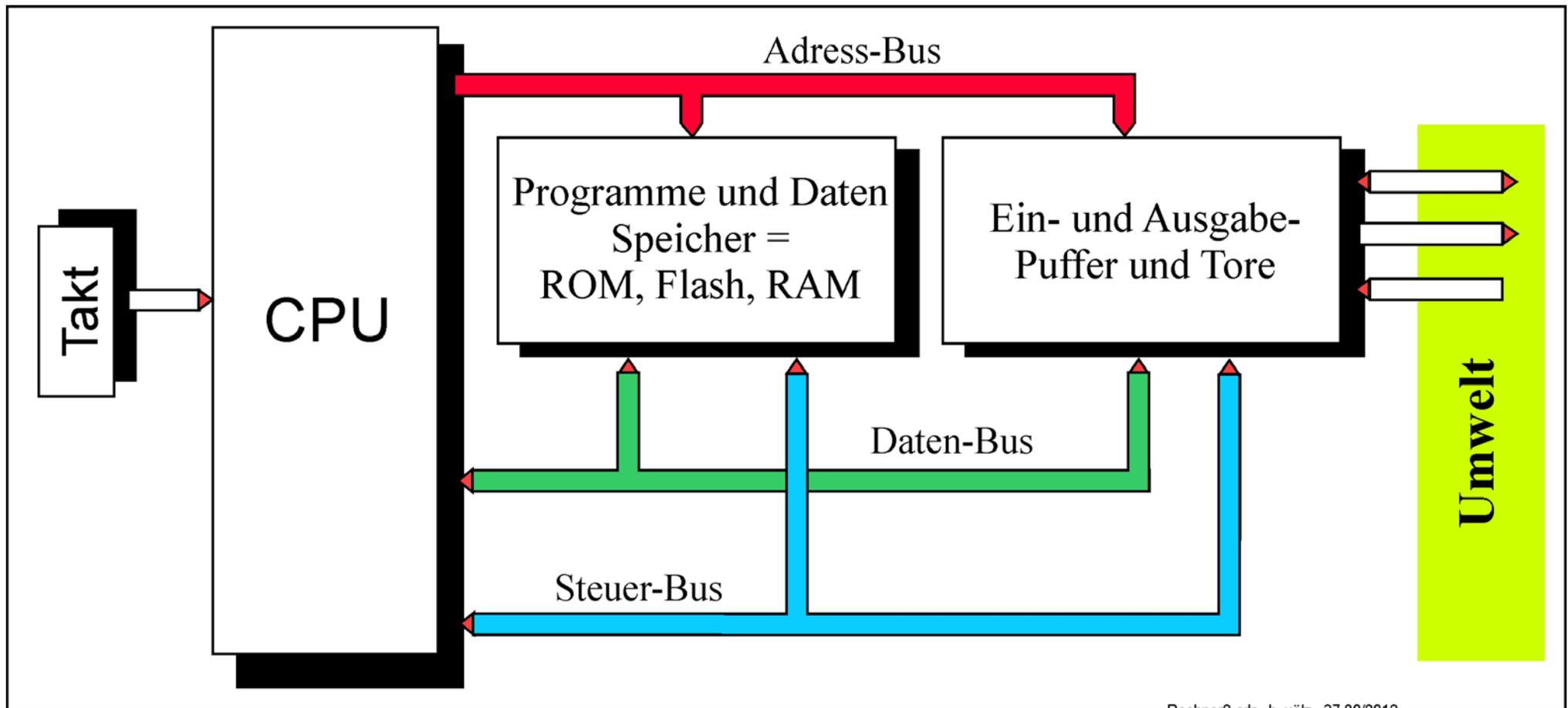
Rechner1.cdr h. vözl 37.80/2012

Die ersten Rechner besaßen noch ein festes Programm, das manuell geändert werden musste.  
 JOHN VON NEUMANN (1903 - 1957) schuf den Universalrechner, indem er das Programm ebenfalls im Speicher ablegte.  
 So konnte es ebenfalls leicht und schnell durch die „übliche“ Eingabe geändert werden.



# Heute übliche Rechner-Struktur.

In der CPU ist heute eine *Arithmetische Einheit* ALU (arithmetic and logic unit) integriert. Sie ist auf vielfältige arithmetische und logische Operationen ausgelegt. Dazu gehören AND, OR, Links-rechts schieben, +, -, \*, / und auch  $\exp(x)$ ,  $\log(x)$ ,  $\sin(x)$  und  $\cos(x)$ .



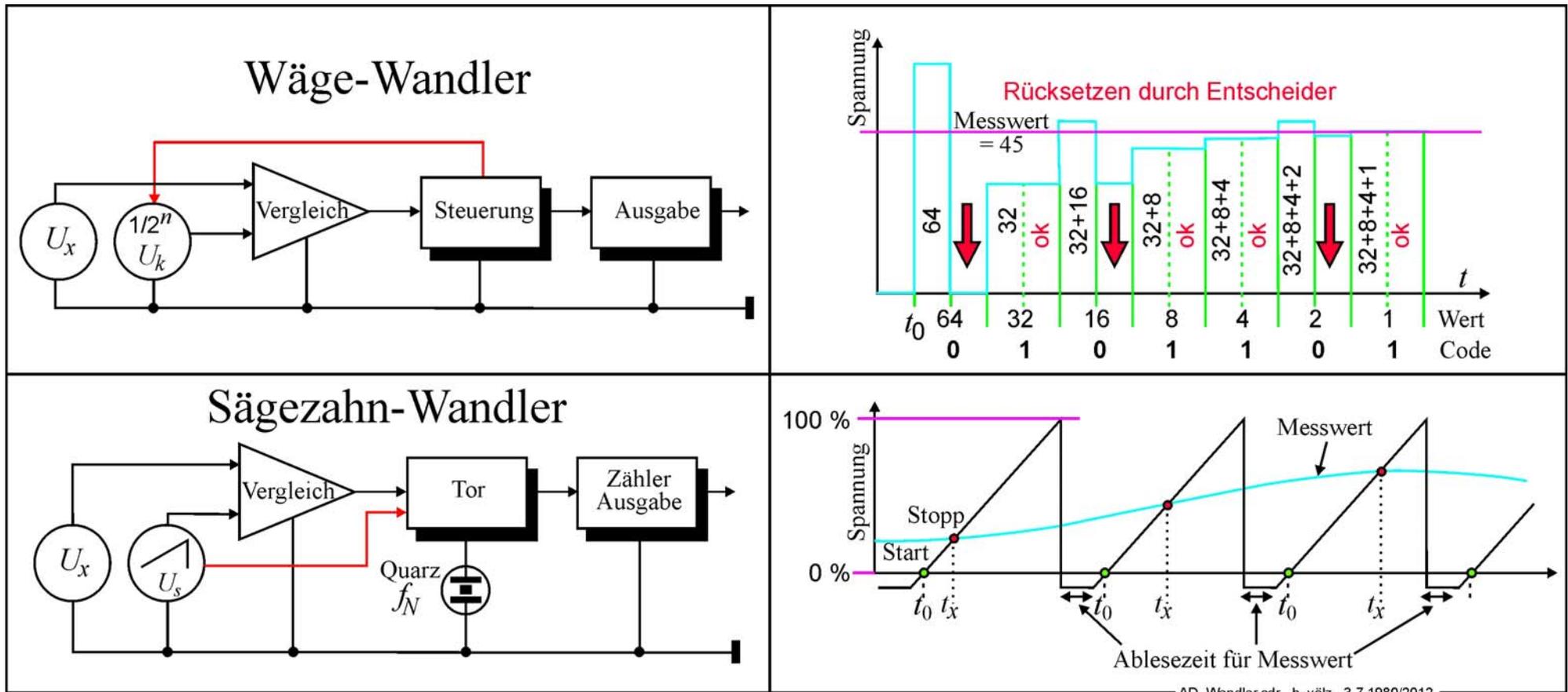
Rechner2.odr h. vözl 37.80/2012

# Gliederung

1. Grundlagen
2. Kombinatorische Bausteine
3. Die drei typischen Digitalerschaltungen
4. Speicher
5. Sequentielle Schaltungen
6. Automaten und Rechner
- 7. Ergänzende Systeme**
8. Berechenbarkeit

# Von der großen Vielzahl ergänzender Systeme werden hier nur einige erwähnt.

Wegen der primär kontinuierlichen Signale sind **AD-Wandler** (analog  $\Rightarrow$  digital) wichtig. Von ihnen gibt es mehrere Schaltungsvarianten. Hier sind nur zwei ausgewählt.



AD\_Wandler.cdr h. vözl 3.7.1980/2012

# Möglichkeiten bei der digitalen Signal-Übertragung

Digitale Signale haben meist eine *feste Taktrate* und eine bestimmte *Bit-Tiefe*.

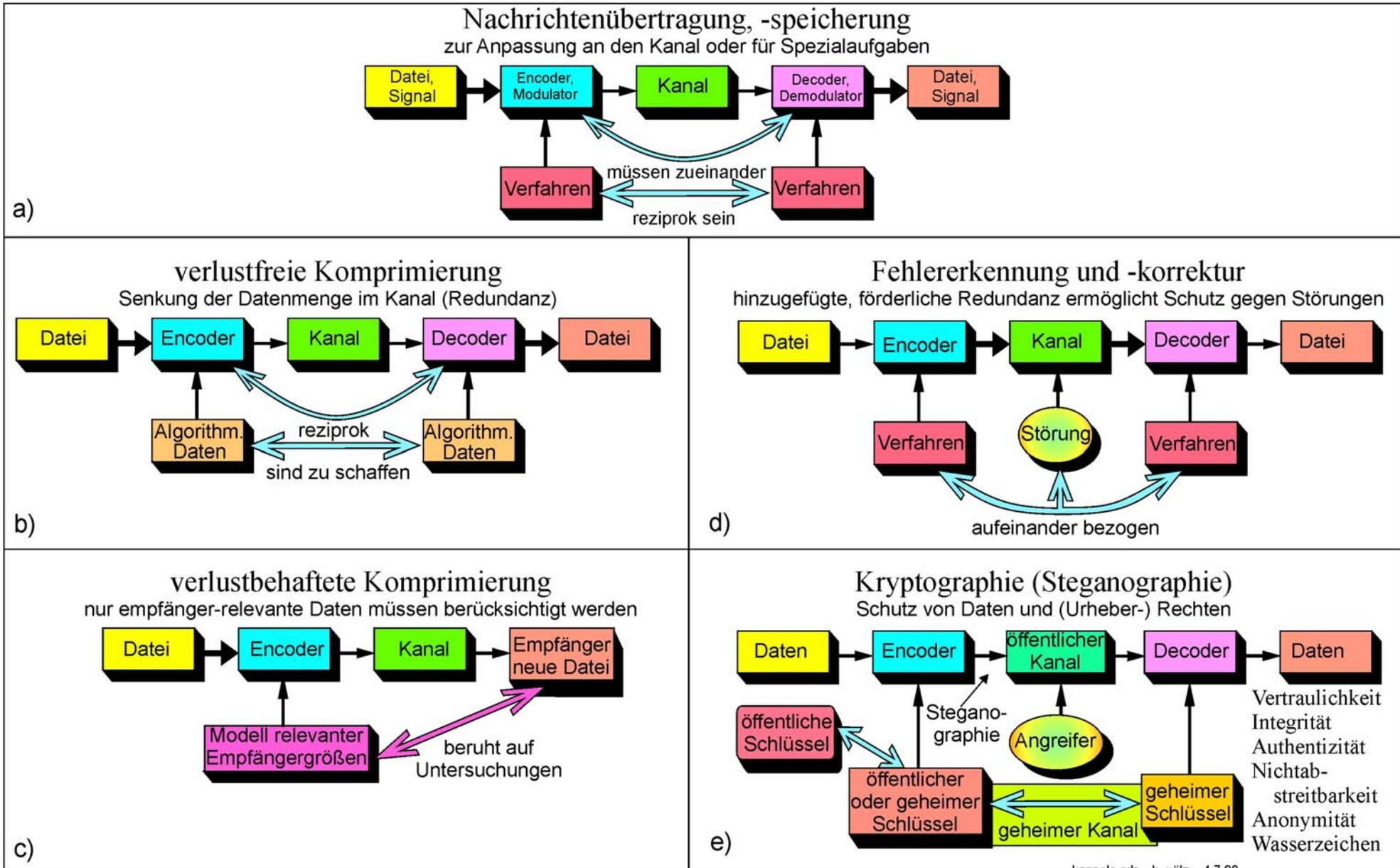
Ein Kanal hat meist Eigenschaften, die hierzu nicht geeignet sind.

Außerdem verändert er teilweise auch einzelne Bit. Das wird durch eine *Fehlerrate* angegeben.

Diese und weitere Einflüsse verlangen eine Anpassung (Encoder und Decoder) zwischen Signal und Kanal.

Dabei sind drei Hauptvarianten zu unterscheiden:

- ***Fehlerreduzierung***: Der Encoder fügt dabei dem Signal eine genau definierte Redundanz hinzu. im Decoder kann dadurch eine Fehler-Erkennung und -Korrektur erfolgen:  
**EDC** (error detection code) und **ECC** (error correction code).  
Der Anteil beider kann dabei unterschiedlich festgelegt werden.  
Nur hierzu folgen noch einige weitere Angaben.
- ***Komprimierung***: Die meisten Signale besitzen eine „*natürliche*“ *Redundanz*.  
Sie kann über Algorithmen erkannt werden und dann beim Encoder entfernt werden.  
Bei Decoder wird sie wieder zurück gewonnen.  
Dazu müssen auf der Sender- und Empfängerseite *Daten und Algorithmen gespeichert* sein.  
Mit großem Rechen- und Daten-Aufwand sind so *nahezu beliebig große Kompressions-Faktoren* zu erreichen.  
Es gibt auch absichtlich *verlustbehaftete Kompressionen*.  
Sie sondern jene Datenanteile aus, die der Empfänger nicht benötigt oder wahrnehmen kann.
- ***Kryptographie***: Sie ermöglicht es, Daten so zu übertragen, dass kein Fremder an sie gelangen kann.



# Fehler-Reduzierung

Für **3-Bit-Signale** kann sie anschaulich recht gut behandelt werden.

Die Signalmöglichkeiten befinden sich dann an den 8 Ecken eines Würfels als x, y und z.

**Mögliche Signale** sind dann u. a.: 000, 001, 010 usw. bis 111.

Als **gültige Signale** für die Übertragung werden davon nur 4 ausgewählt: 000, 011, 101 und 110.

Treten nach dem Kanal andere auf, so liegt ein **Übertragungsfehler** vor.

Fehlerkorrektur ist aber erst dann möglich, wenn nur zwei Signale als gültig erklärt werden: 000 und 111.

Dann ist bei fehlerhaften Signalen, z. B. 001 das nächstliegende nämlich 000 wahrscheinlich.

Wird die Korrektur durchgeführt, so ist keine zusätzliche Fehlererkennung möglich.

In sehr seltenen Fällen kann die Korrektur sogar falsch sein. **Fehlerkorrektur kann also immer einmal falsch sein.**

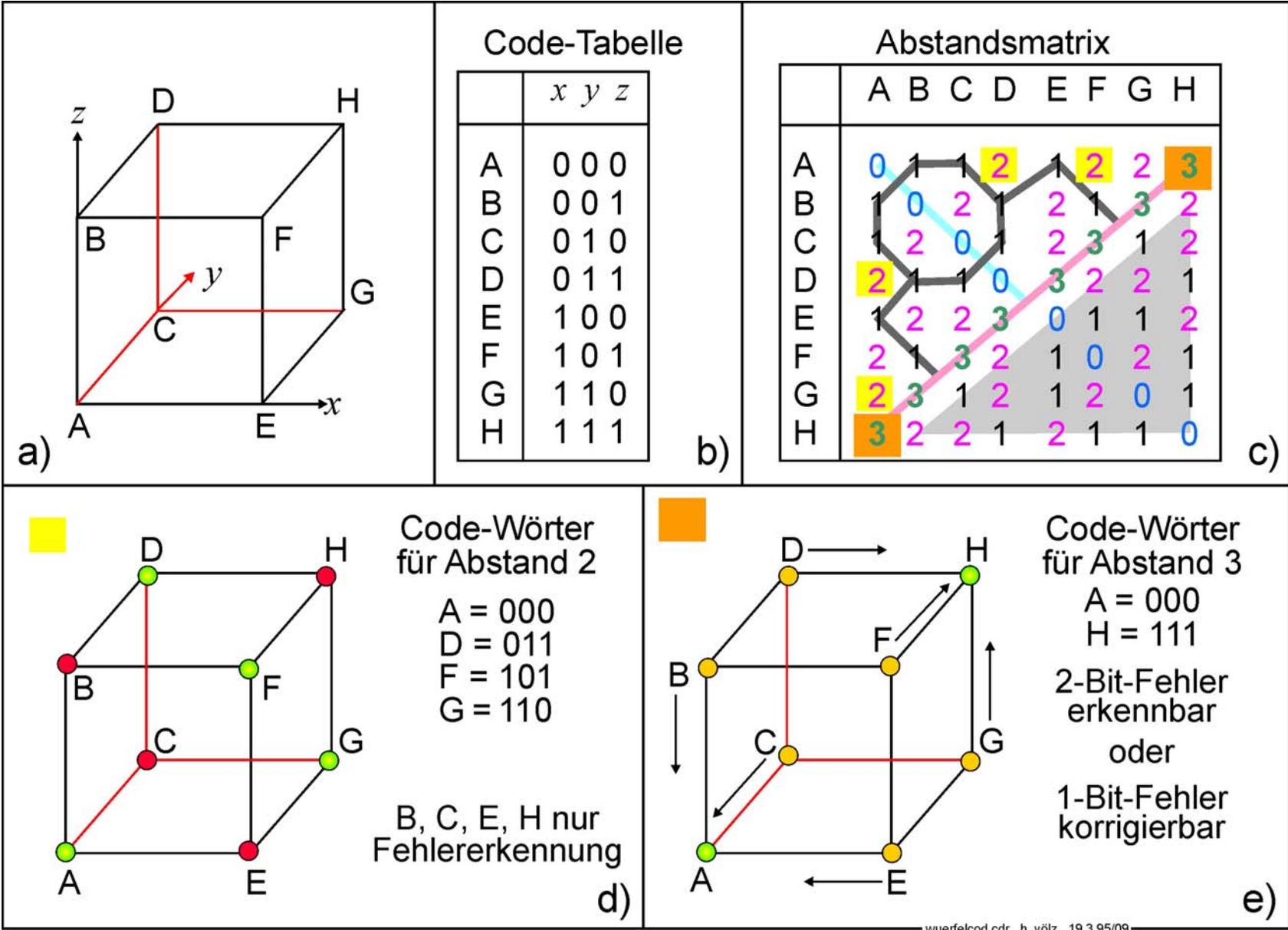
Dennoch wird durch sie die Fehlerrate deutlich gesenkt.

Für die **angewendeten Fehlerkorrekturen** sind Betrachtungen im  $n$ -dimensionalen Raum notwendig.

Eine brauchbare Veranschaulichung durch Projektion auf die Ebene ermöglicht das folgende Bild.

Die **Mathematik** der Fehler-Reduzierung ist **sehr schwierig**.

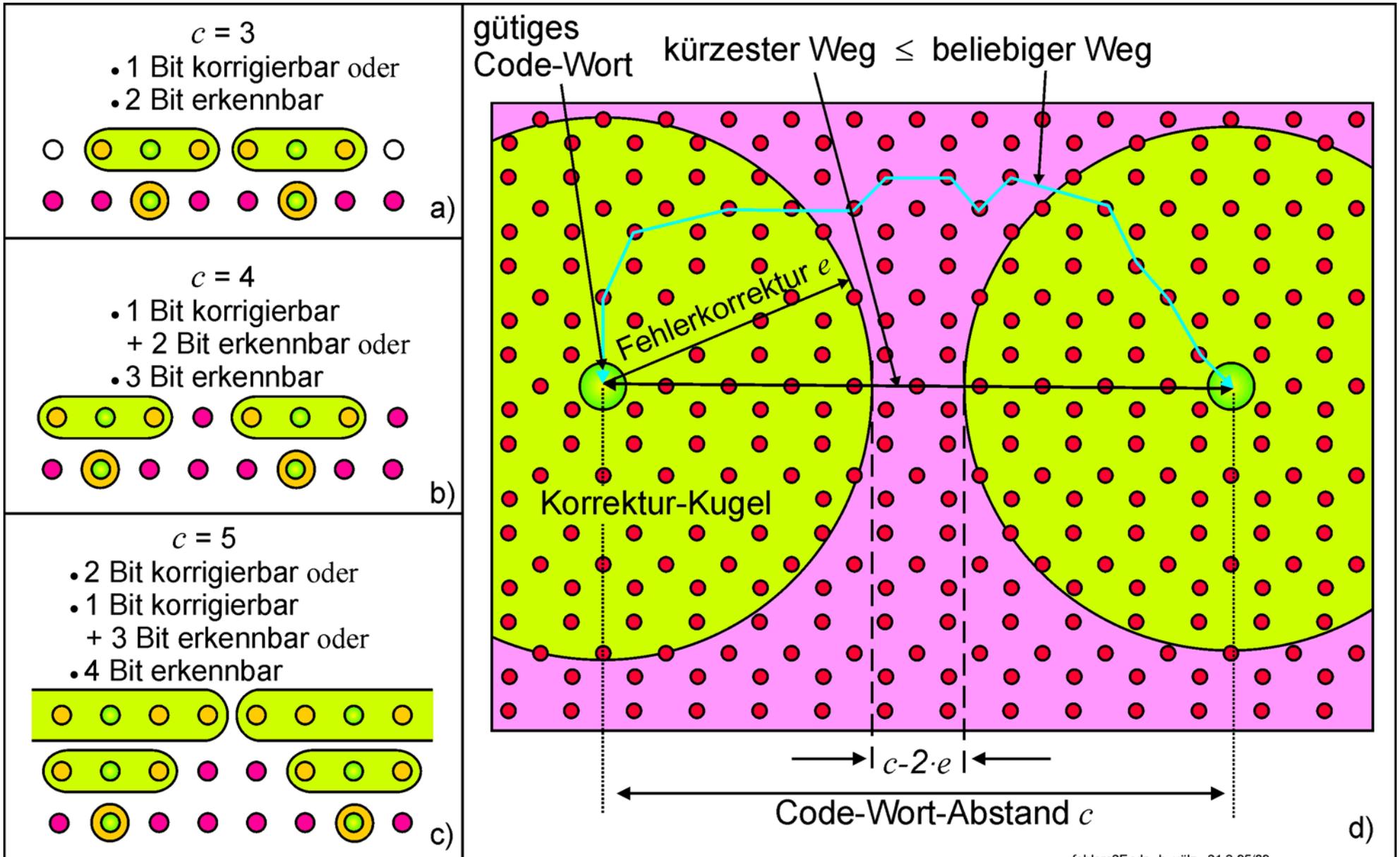
Die eingeführten Verfahren erhalten daher die **Namen der Erfinder**.



## Dies ist eine stark vereinfachte Erklärung mit Wörtern

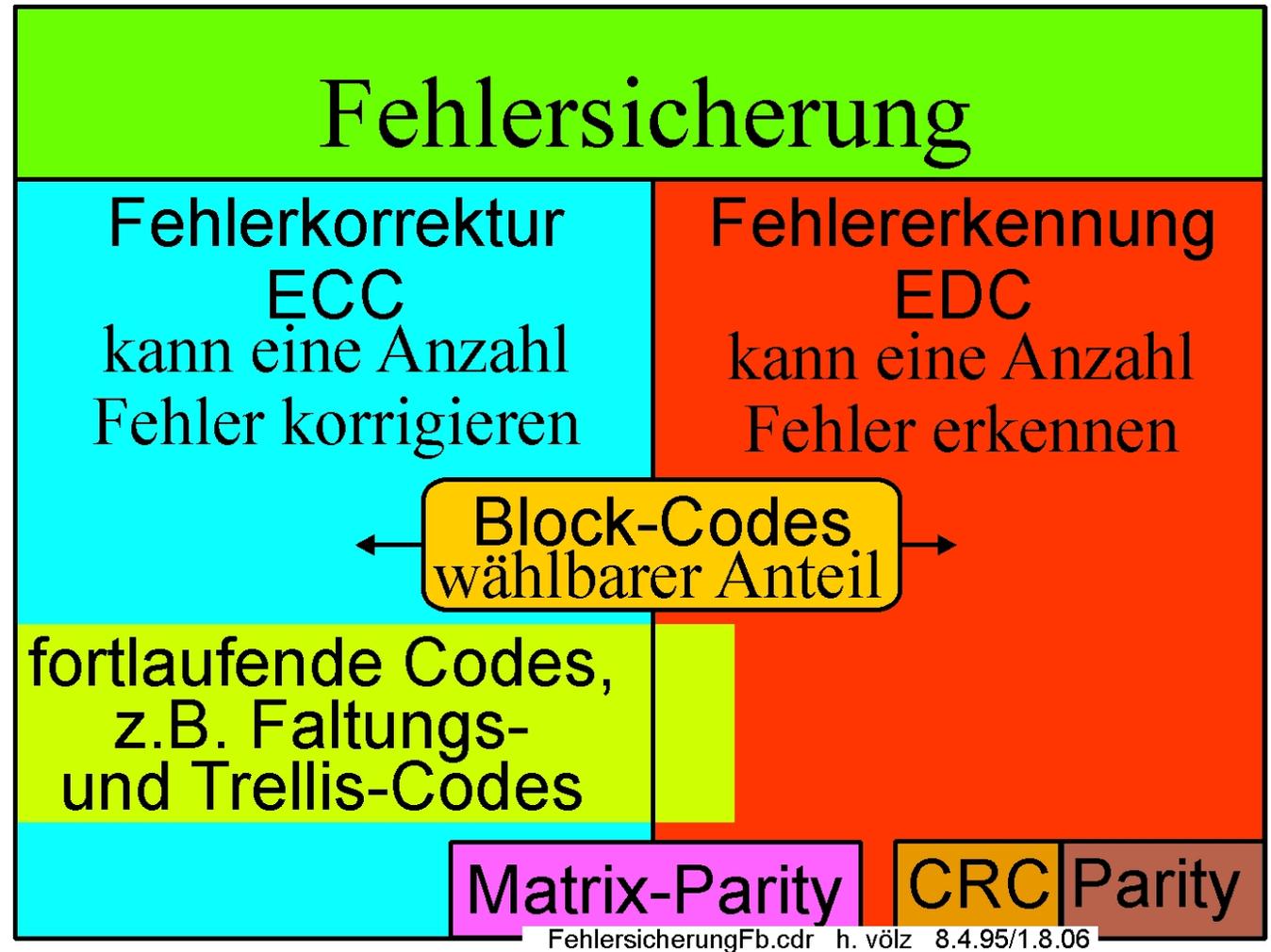
Zugelassene Zeichen (statt 0/1) A, E, G, I, L	Möglichkeiten einer Fehlerkorrektur bei auftretenden ungültigen Wörtern
<b>Gültige Wörter</b> ALGE AGIL EGAL GEIL IGEL LAGE LEGE	ALGO 1 Bit korrigierbar zu ALGE LIGE 1 Bit <b>nicht</b> korrigierbar: LAGE oder LEGE AGEI 2 Bit korrigierbar zu AGIL ELIL 2 Bit <b>nicht</b> korrigierbar AGIL oder EGAL oder GEIL GILI 3 Bit korrigierbar zu GEIL LILA 3 Bit <b>nicht</b> korrigierbar LAGE oder LEGE

ECC\_Wörter.cdr h. völz 15.1.13



Es gibt eine große *Vielfalt der Verfahren* zur Fehler-Erkennung und -Korrektur.

Einen gewissen Überblick gibt das nebenstehende Bild.



Weitere Details siehe [r-h-voelz.de/PDF TU/Fehlerkorrekturneu.pdf](http://r-h-voelz.de/PDF_TU/Fehlerkorrekturneu.pdf)

# Gliederung

1. Grundlagen
2. Kombinatorische Bausteine
3. Die drei typischen Digitalschaltungen
4. Speicher
5. Sequentielle Schaltungen
6. Automaten und Rechner
7. Ergänzende Systeme
- 8. Berechenbarkeit**

# Widersprüche

Für die Pegel der *Digitaltechnik* gibt es *notwendig* immer eine ungültige Zone don't care (s. o.). In der Logik ist das Dritte zwischen *ja* und *nein* bzw. *falsch* und *wahr* jedoch ausgeschlossen. Ohne diese Festlegung würde der *Wissenschaft eine ganz wesentliche Grundlage* fehlen. Dennoch ist eine *Vielzahl* daraus folgender *Widersprüche* seit dem Altertum bekannt. Dafür stehen Namen wie: *Antinomie*, *Aporie* (KANT), *Dilemma*, *Paradoxie* und *Trilemma*. [Falletta] unterscheidet (hier etwas vereinfacht) inhaltlich:

- *Zeit-, Folge*-Antinomien: Körner  $\Rightarrow$  Haufen; Kaulquappe  $\Rightarrow$  Frosch; All-Aussagen, Zenon-Antinomien.
- *Unendlichkeits*-Paradoxien: Hilbert Hotel, Monsterkurven, Fraktale, einige ESCHER-Bilder.
- *Statistische* Antinomien: U-Bahn-Warten, Gefangenen-Dilemma, statistische Erwartungen.
- *Bildliche* Antinomien: Umschlagbilder, mangelhafte Perspektiven.

Die Fülle der Möglichkeiten kann hier nicht einmal angedeutet werden. Eine Kurzfassung, die dennoch sehr umfangreich ist, enthält [Völz 2001]. Hier seien nur angeführt „*Ich lüge!*“ und

**Die eingerahmte Aussage auf dieser Seite ist falsch.**

*griechisch anti*: dagegen, entgegen und *nemein* (nomie): ordnungsgemäß aus-, ver-, zuteilen, lenken, leiten, verwalten

*griechische aporia*: Ausweglosigkeit, ausweglose Lage und Schwierigkeit

*griechische di*: zweifach und *lemma*: Annahme, Hilfssatz, Prämisse drei (griechisch *tri*)

*griechischen paradoxon*, was aus *para*: daneben, dabei, von her, gegen, wider und aus *doxa*: Ansicht, Meinung

Paradoxa sind auch Effekte und Geschehen der Naturwissenschaft, die dem „gesunden“ Menschenverstand widersprechen (Aerodynamisches Paradoxon).

# Axiomatik

*Griechisch **axioma*** Würdigung; Würde, Ansehen

Um in der Mathematik widerspruchsfrei zu arbeiten, wurde ab der Antike die **Axiomatische Methode** entwickelt. Es besteht aus mehreren Axiomen, die *nicht abgeleitet* oder *bewiesen* werden können. Sie müssen *unmittelbar einsichtig* sein.

Daher ist auch das Axiomen-System nicht zu beweisen.

Es muss jedoch folgende **Bedingungen** erfüllen:

- **Korrektheit:** Aus den Axiomen kann nichts Falsches bewiesen werden.
- **Vollständigkeit:** Aus den Axiomen kann alles Gültige bewiesen werden.
- **Unabhängigkeit:** Alle Axiome sind notwendig, Es gibt keine Redundanz.
- **Minimalität:** Von verschiedenen möglichen Axiomen-Systemen für die gleiche Theorie wird immer das kleinste ausgewählt.

Bei der Ableitung aus dem Axiomen-System ist der Satz vom **ausgeschlossenen Dritten** entscheidend.

Die **dazugehörenden Regeln** lassen nur die logischen Ableitungen mit Entweder-Oder bzw. Ja oder Nein zu.

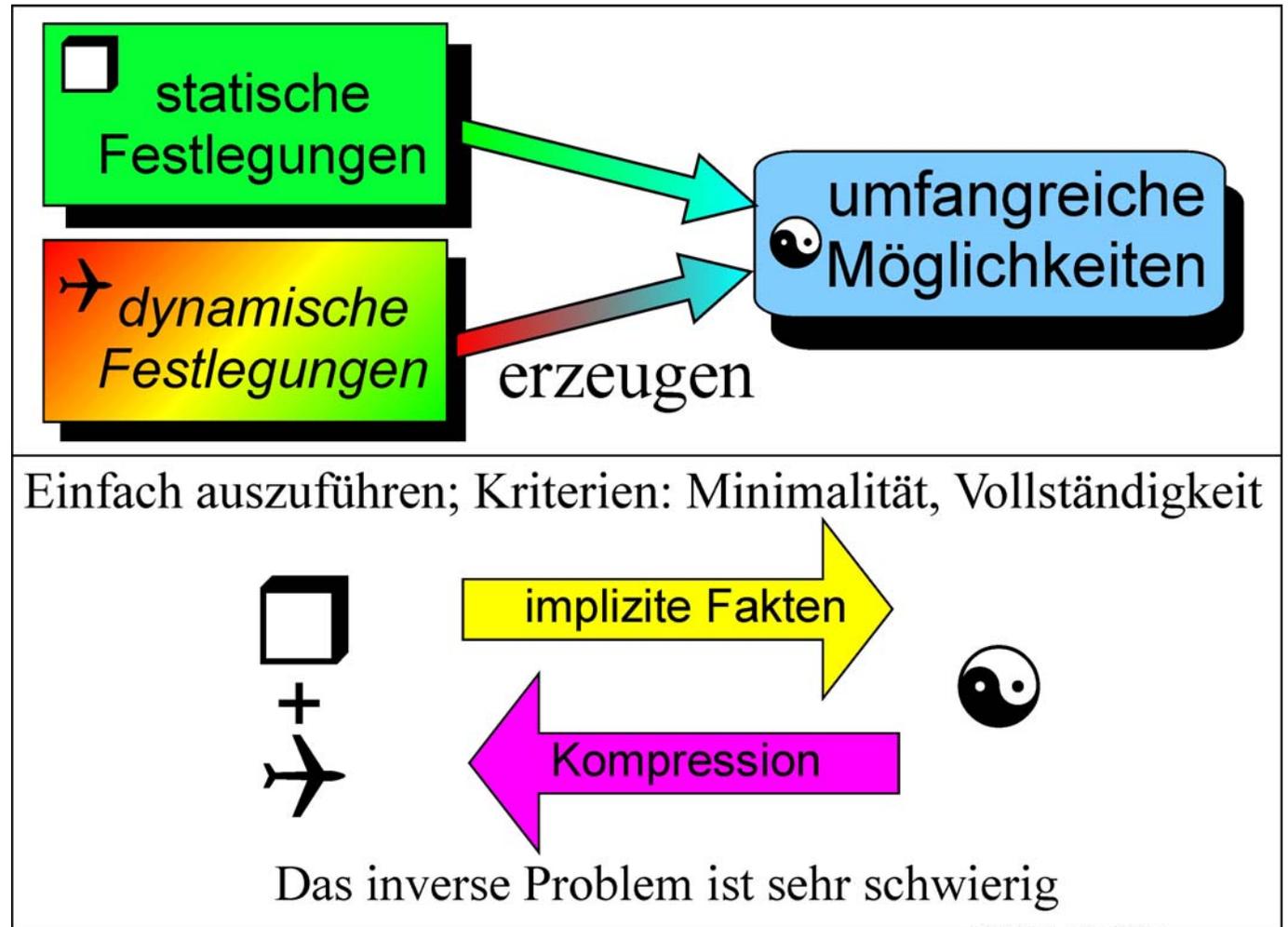
# Verallgemeinertes Prinzip von Axiomen-Systemen.

Die *statischen Festlegungen* müssen unmittelbar einsichtig sein und dürfen keine Hinterfragung herausfordern

Die *Regeln* müssen dem Gesetz des ausgeschlossenen Dritten genügen.

Auf diese Weise wird mit wenigen Aussagen eine große Vielfalt von Folgerungen gewonnen. Das entspricht *einer Kompression* von Daten bzw. den Gesetzen der Natur.

Die Umkehrung von vielfältigen Einzel-Erkenntnissen zu Gesetzen und Regeln ist erheblich schwieriger und entspricht – zumindest bei den Naturgesetzen – den Leistungen bedeutender Wissenschaftler und trägt daher meist deren Namen.



# Widerspruch in der Mengentheorie und FREGE

Von den Problemen war besonders stark FRIEDRICH LUDWIG GOTTLOB FREGE (1848 - 1925) betroffen. Er schloss gerade den 2. Band seines Hauptwerkes „Grundgesetze der Arithmetik“ (1893 - 1903) ab. 1901 fragte ihn BERTRAND ARTHUR WILLIAM RUSSEL (1872 - 1970) in einem Brief:

*„Ist die Menge aller Mengen wieder eine Menge?“*

Daraus folgt nämlich:

*„Die Menge  $R$  enthält sich selbst als Element und die Menge  $R$  enthält sich nicht selbst als Element.“*

Er konnte diesen Widerspruch gerade noch im Vorwort wie folgt berücksichtigen:

*„Einem wissenschaftlichen Schriftsteller kann kaum etwas Unerwünschteres begegnen, als dass ihm nach Vollendung seiner Arbeit eine der Grundlagen seines Baues erschüttert werden. In diese Lage wurde ich durch den Brief des Herrn Bertrand Russel versetzt, als der Druck dieses Bandes sich seinem Ende näherte.“*

Später wurden deshalb in der Mengentheorie meist *drei Objektarten* eingeführt:

- **Urelemente** treten als Elemente von Gesamtheiten auf, enthalten aber selbst keine Elemente.
- **Mengen** sind sowohl Gesamtheiten als auch Elemente von anderen Gesamtheiten.
- **Unmengen** sind Gesamtheiten, die aber nicht Elemente anderer Gesamtheiten sind.

Das ermöglicht die folgende Klassifikation:

	<b>Elemente</b>	Unmengen
Ur-Elemente	Nullmenge, Mengen	
	<b>Klassen</b>	

# HILBERT und GOEDEL

Auf dem Mathematik-Kongress von 1900 forderte DAVID HILBERT (1862 - 1943) die *vollständige Axiomatisierung* aller mathematischen Theorien.

Noch einschneidender als bei FREGE beweist 1930 KURT GOEDEL (1906 - 1978) in seiner Dissertation die Unmöglichkeit. Sein konkretes Beispiel ist die noch recht einfache ganzzahlige Arithmetik [Goedel 1931]. Verallgemeinert kann das so gewonnene Ergebnis unterschiedlich ausgedrückt werden, z. B.:

*Ein hinreichend komplexes System enthält immer Aussagen, die sowohl falsch als auch wahr sind.*

Einen leider nicht konkret nutzbaren Ausweg zeigte 1933 ALFRED TARSKI (1901 - 1983) mit dem Begriff **Metasprache**. Danach dürfen *Aussagen in einer Sprache* nicht mit *Aussagen über* diese „Objekt“-Sprache vermischt werden. Doch letztlich steht uns aber nur die *Umgangssprache* zur Verfügung, über die wir auch Aussagen machen müssen.

Als Folge des Geschehens werden heute *drei Gruppen mathematischer Theorien*, jene.

1. die *voll axiomatisierbar* sind,
2. die *Widersprüche enthalten* und
3. bei denen *noch nicht entschieden* ist, ob sie zu 1. oder 2. gehören.

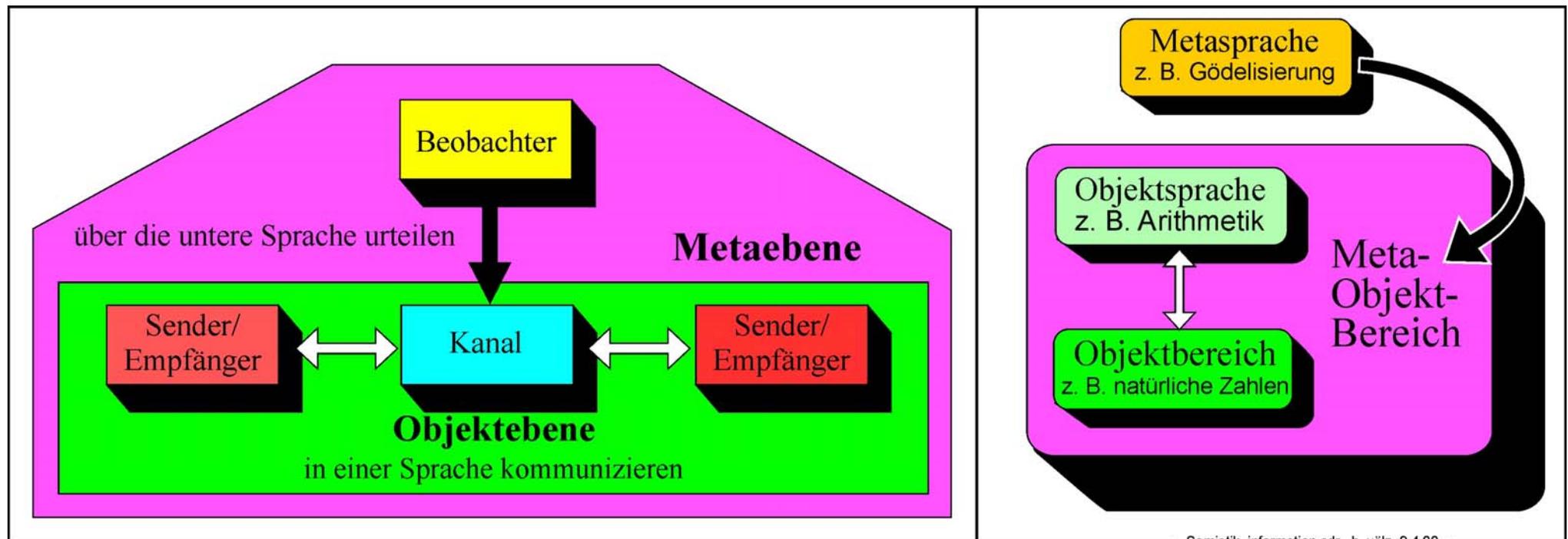
## Widerspruchsfreiheit ist unter folgenden Bedingungen zu erreichen:

Zu einem Objektbereich mit einer Objektsprache müssen existieren:

Ein übergeordneter *Meta-Objektbereich* mit einer *unabhängigen Meta-Sprache*.

Dann ist eine widerspruchsfreie GOEDELisierung möglich.

Leider gibt es zu unserer *Umgangssprache* keine unabhängige übergeordnete Sprache. So entstehen die Widersprüche. Sie müsste informationstheoretisch dem *Beobachter* eines Kanals *ohne Wechselwirkung* entsprechen.



# Der TURING-Automat

ALAN MATHISON TURING (1912 - 1954)

GOEDEL zeigte auf, was nicht möglich ist.

Entgegengesetzt verfuhr TURING: Er schuf 1936 ein *konstruktives Gedanken-Modell* für allgemeine Berechnungen.

Es ist eigentlich nur ein *Denkschema* und besteht aus *drei Teilen* [Turing1937]:

- Ein *endloses Speicherband*, das bewegt, beschrieben und gelesen werden kann.
- Einen veränderbaren *Zustandsspeicher*.
- Mehrere *TURING-Tabellen*, die einfachen, klassischen *Automaten* entsprechen (s. o.) und so Eingaben je nach dem aktuellen Zuständen in Ausgaben umsetzen und dabei auch Entscheidungen treffen.

Zu *Beginn* werden die Daten für die Berechnung auf das Speicherband geschrieben.

Nach dem *Start* laufen die Prozesse gemäß dem jeweils implementierten *Algorithmus* (s. u.) getaktet ab.

Wenn der Automat *endet*, liegt das gewünschte *Ergebnis auf dem Band* vor.

Bis auf den *endlosen Speicher* ist so ein Automat prinzipiell *technisch* herstellbar, das geschah aber wohl nie.

Den Ablauf einer einfachen Rechnung nach diesem Prinzip zeigt [Völz 1991] ab S. 250 im allen Details auf.

TURING beweist aber gleich mit seinem Denkschema: Es kann keinen Algorithmus geben, der entscheidet, ob der Automat jemals anhält = *Halte-Problem*.

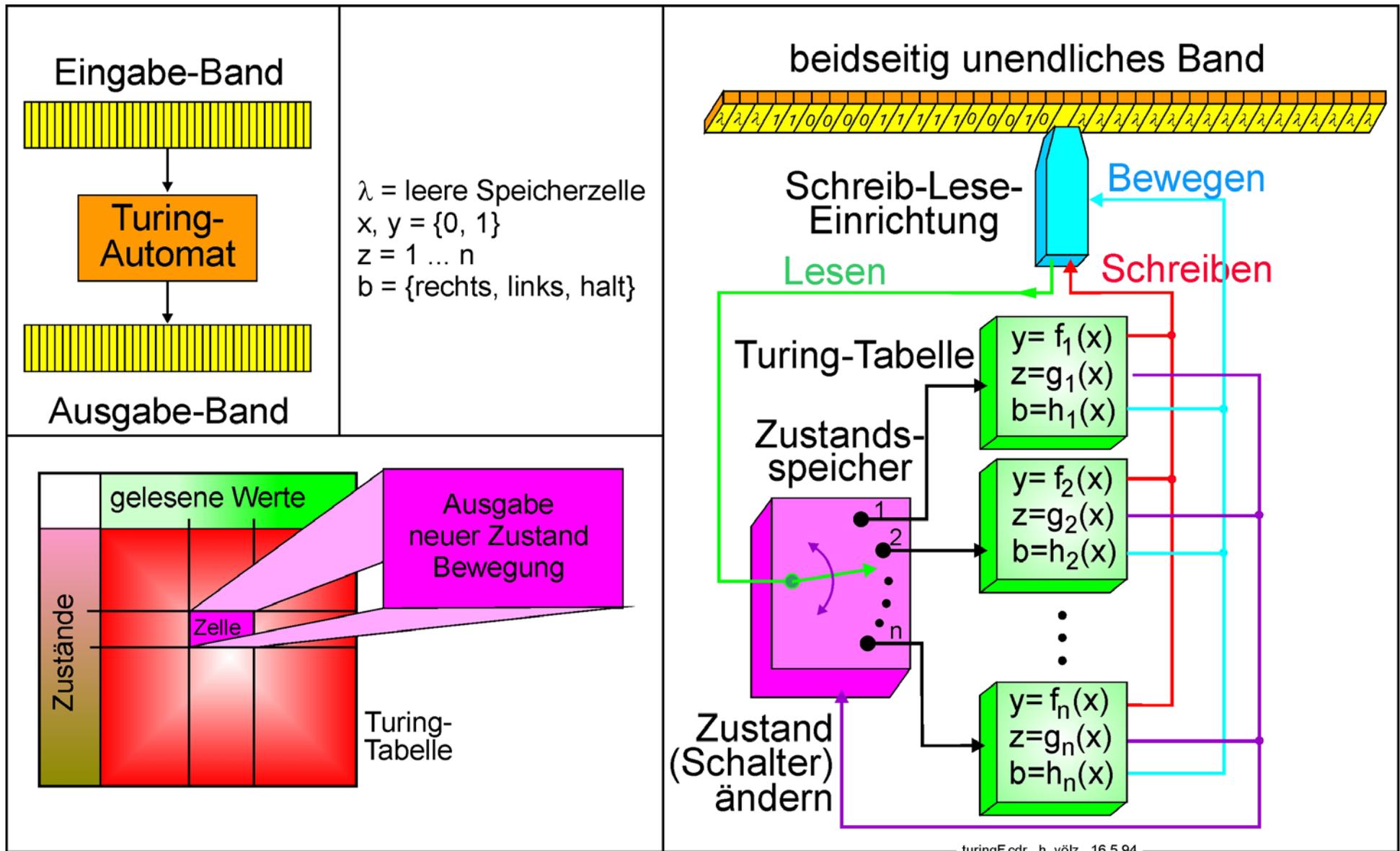
Es könnte also Probleme geben, die in endlicher Zeit nicht lösbar sind!

Weiter konnte er zeigen: Für alle mit *speziellen TURING-Automaten* berechenbaren Funktionen lässt sich ein einziger *universeller TURING-Automat* konstruieren.

1950 Computing Machinery and Intelligence. (Turing-Test für künstlichen Intelligenz, KI; bzgl. Bewusstsein)

1952 The Chemical Basis of Morphogenesis (Grundstein für die Entwicklungsbiologie).

# Ins Technische „übersetzter“ Turing-Automat



# Weiterentwicklung durch CHURCH

Nach TURING entstanden etwa zehn weitere, deutliche andere Methoden zur Berechnung.

Eine genaue Untersuchung zeigte, dass sie alle die gleiche Menge mathematischer Funktionen berechnen.

Daher stellte 1940 ALONZO CHURCH (1903 - 1995) folgende *nicht beweisbare These* (CHURCH-These) auf.

Alle diese und jede vernünftige jemals aufgestellte Algorithmus ist gleichwertig.

Alle bestimmen dabei die Grenze der Berechenbarkeit.

Ihr wesentliches Kennzeichen ist die *Rekursivität* (*lateinisch recursio* das Zurücklaufen) und so folgt:

*Alle berechenbaren Funktionen sind rekursiv*

Achtung! Der Begriff *rekursiv* kann aber auch neben Algorithmus in unterschiedlichem Bezug verwendet werden, Z. B. für Verfahren, Prozeduren, Datenstrukturen, Mengen, Aufzählbarkeit und Theorien.

# Algorithmus

Das Wort eine seltsam *Abwandlung des Namens* von MUHAMMAD IBN MUSA AL-CHWARIZMI ( $\approx 783 - \approx 850$ )  
Er ist *Autor des Buches* Hisab al-dschabr wa-l-muqabala (825, Regeln zur Wiederherstellung und Reduktion).  
Mit dem Buch wurde die *Algebra im Westen* verbreitet.

Die lateinische Fassung *beginnt mit* „Dixit Algoritmi...“ (Algorithmus sprach...), womit der Autor gemeint war.  
Das Wort kommt auch im *Titel des Buches* vor: al-Jabr – „Einrenkung“

*Ursprünglich* wurde „Algorism“ nur für die Regeln zur *Arithmetik mit arabischen Ziffern* benutzt.

*Heute* umfasst es *alle Prozeduren*, mit denen Probleme (Aufgaben) aller Art gelöst werden können.

Seine wichtigsten *Eigenschaften* sind:

- Für gleiche *Eingabewerte* muss er unter gleichen *Bedingungen* stets die *gleiche Ausgabe* liefern.
- Er muss die Eigenschaften: *Determiniertheit*, *Allgemeinheit* und *Endlichkeit* besitzen

Zuweilen werden auch *nicht deterministische* (z. B. stochastische) Algorithmen benutzt,  
die bei gleichen Startbedingungen unterschiedliche Ergebnisse liefern können.

Algorithmen können unterschiedlich *dargestellt werden*, z. B.:

- *Verbale* Beschreibung.
- *PAP* = Programmablaufplan.
- *Struktogramm*.
- In einer *Programmiersprache*.

Bei den Programmiersprachen sind immer einige *Grundbefehle* vorhanden, z. B.:

Load, Write, Goto, For-To-Step-Next, While-Do, For-Do, Repeat until, Return usw.

# Kurze Zusammenfassung

Für den heutigen *Stand der Informatik* sind vier große Erkenntnisse entscheidend:

1900 HILBERT *fordert* die *Axiomatisierung* der gesamten Mathematik.

1930 GOEDEL *beweist*, das ist selbst für die ganzzahlige Arithmetik *unmöglich*.

1936 TURING *bestimmt* mit seinem TURING-Automaten Wege zur *Berechenbarkeit* [Turing 1937].

1940 CHURCH *behauptet* alle berechenbaren Funktionen sind *rekursiv*.

# Zeit- und Raum-Komplexität

Jedes rechnerische Problem besitzt eine *Anzahl*  $n$  von *Parameter*:

Bei *Primzahl-Zerlegung* ist es z. B. ein beachtlicher Unterschied, ob 15 oder 103 398 947 untersucht wird.

Für 15 ist sofort  $3 \cdot 5$  zu finden. Dagegen dauert  $17 \cdot 367 \cdot 16573$  bei der zweiten Zahl erheblich länger.

Weitere Beispiele für  $n$  sind:

- Bei arithmetischen Operationen ist es die *Anzahl der Stellen* und/oder der *Variablen*.
- Bei Such- oder Sortieralgorithmus betrifft es die *Anzahl der Wörter* oder *Begriffe*.
- Bei Programm geht die *Anzahl der Befehle* ein.

Von der Parameterzahl  $n$  hängen deutlich die *Rechenzeit*  $T$  und die *benötigte Speicherkapazität*  $R$  (Raum!) ab.

Für *kleine*  $n$  lassen sich oft besonders schnelle und speicherarme Algorithmen finden.

Für immer *größer werdende Parameter* ( $n \rightarrow \infty$ ) erfolgt eine *Näherung an Funktionen*  $T = \Omega(n)$  und  $R = \Phi(n)$ .

Sie hängen natürlich von der Hardware und vom benutzten Algorithmus ab.

Dennoch lassen sich *zwei typische Fälle* deutlich unterscheiden:

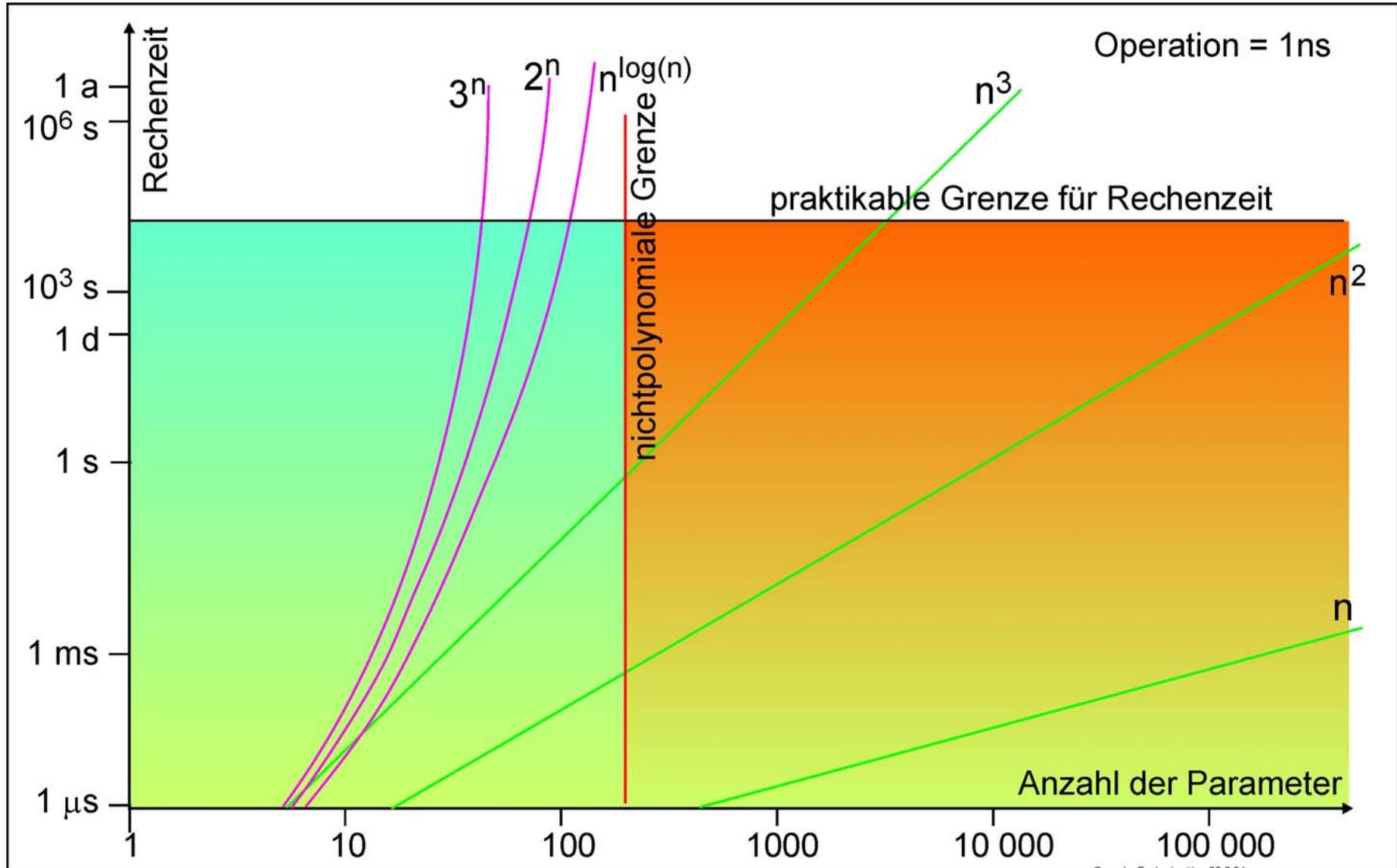
Polynomiales Wachstum =  $\Omega(n) \sim n^x$ ; exponentielles =  $\Omega(n) \sim x^n$ .

Diese einfache Schreibweise entsteht, weil bei großen  $n$  die jeweils höchste Potenz genügt.

Neben dem exponentiellen gibt weitere *nichtpolynomiale* Funktionen, wie etwa  $n^{\log(n)}$ .

Diese Kurven gelten für 1 ns je Operation.

Der Unterschied zwischen *polynomialer* und *nichtpolynomialer* Rechenzeit bleibt aber unabhängig davon bestehen.



# $P \Leftrightarrow NP$

Der große *Unterschied* von *polynomial* und *nichtpolynomial* betrifft die folgte *Begrenzung* der verfügbaren Zeit. Alle nichtpolynomialen (= *überpolynomialen*) Algorithmen ermöglichen nur eine *stark begrenzte Parameterzahl*. Deshalb werden zwei typische Begriffe der Berechenbarkeit unterschieden:

- **P**-Probleme sind mit *polynomialem* Aufwand lösbar und heißen *feasible* (durchführbar).
- **NP**-Probleme betreffen Algorithmen mit steilerem (nichtpolynomialem) Verlauf und heißen – wegen der stark begrenzten Parameterzahl – *infeasible* (nicht durchführbar).

Für **NP**-Probleme könnte aber auch eine *nichtdeterministische* Lösung in polynomialer Zeit möglich sein. Dazu wird u. a. ein „Orakel“ angenommen, das den Weg zur Lösung kennt.

Details verlangen spezielle Fachliteratur. Dort wird auch die noch unbeantwortete Frage  $P \neq NP$  gestellt.

Weiter existiert der Begriff **NP vollständig**. Ein Beispiel ist das Problem des Handelsreisenden.

Wird für sie ein Algorithmus mit einer polynomiale Lösung gefunden, dann ist er auf alle NP-Probleme übertragbar.

Hier noch einige *Ergänzungen*:

- Die *polynomiale Grenze* ist weitgehend *unabhängig* vom gewählten Rechner- bzw. Automatentyp. Selbst bei sehr unterschiedlichen TURING-Modellen ändert sich maximal der Grad des Polynoms.
- Alle *wichtigsten Operationen*, wie Addition, Multiplikation und die gebräuchlichen arithmetischen Funktionen besitzen eine polynomiale Zeitschranke mit vom Grad  $\leq 3$ .
- Da eine Summe, ein Produkt sowie die Kombination von Polynomen immer wieder zu einem Polynom führen, bleibt auch die Kombination solcher Algorithmen beim polynomialen Aufwand.

**NP-vollständig**  
mit Transformationen der  
Klasse P sind so alle  
NP-Probleme lösbar

**Klasse P**  
durchführbar,  
feasible

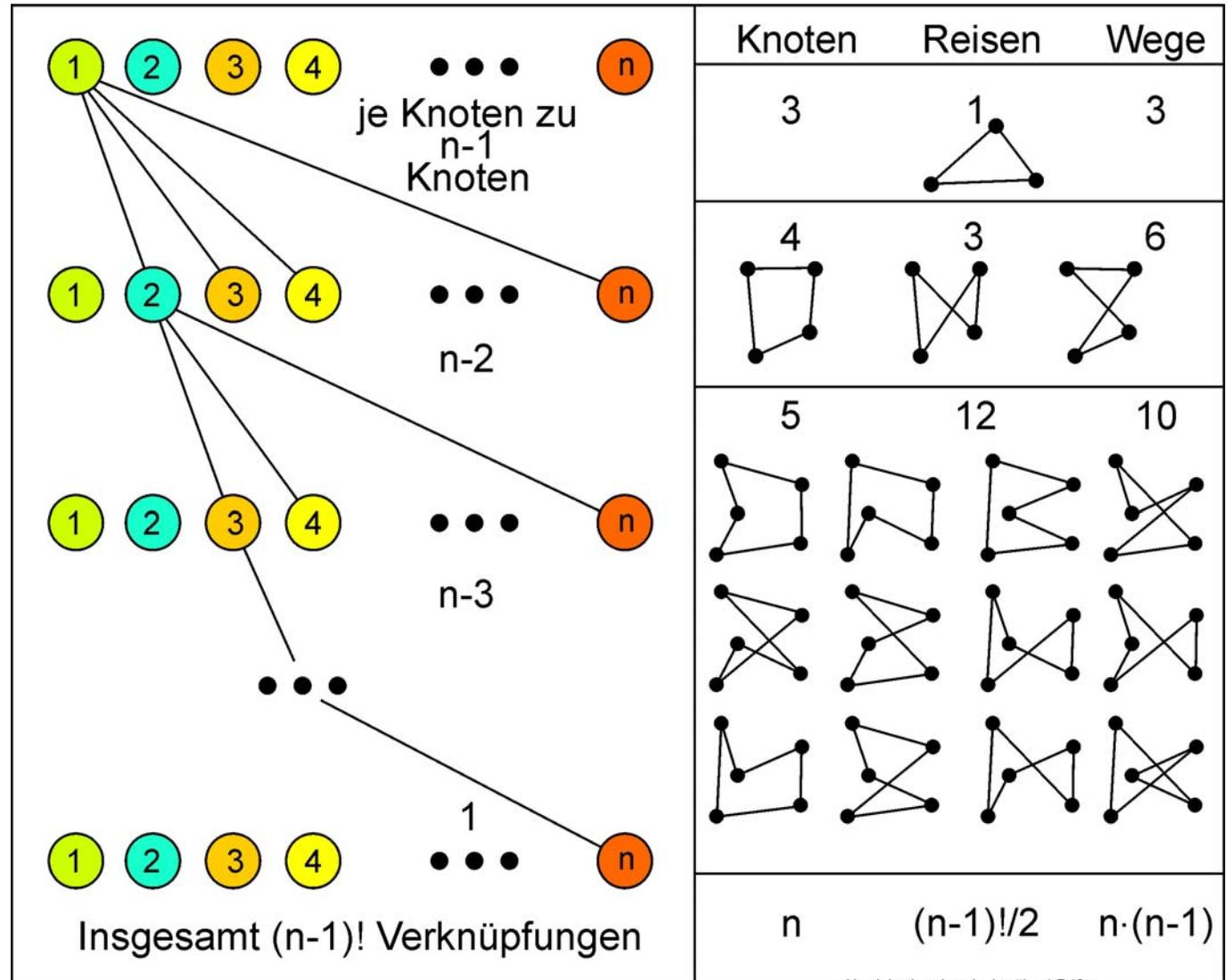
**Klasse NP, berechenbar, computable**

**Alle Probleme, auch nicht berechenbare**

# Problem des Handelsreisenden

Es sollen  $n$  Orte genau einmal auf dem kürzesten Weg besucht werden.

Die Anzahl der möglichen Wege nimmt mit  $n$  so schnell zu, dass ab grob 30 Orten keine Berechnung in sinnvoller Zeit mehr möglich ist.



# Zusammenfassung

Der Übergang von kontinuierlich nach digital erzwingt einen verbotenen Signalbereich zwischen 0 und 1. Es wird gefragt, ob er dem verbotenen Dritten der Logik zugeordnet werden kann. Nach Einführung der möglichen Signalkennlinie werden weitere Eigenschaften von kontinuierlich und digital verglichen. Es folgt die Einführung der kombinatorischen Schaltungen, ihrer Grenzen, Basis-Systeme und technischen Varianten. Als komplexe Bausteine werden u. a. Coder, Decoder, Multiplexer und Demultiplexer sowie PLA, PROM und PAL vorgestellt. Die Einteilung der digitalen Bausteine führt zur Dreiteilung in kombinatorische, sequentielle Bausteine und Speicher. Die große Vielfalt der Speicher erzwingt selbst bei starker Einschränkung einen ziemlich umfangreichen Teil. Es folgen wichtige sequentielle Schaltungen einschließlich der verschiedenen Automaten- und Rechner-Strukturen. Als ergänzende Systeme werden nur kurz AD-Wandler, Kompression, Fehlerbehandlung und Kryptografie behandelt. Abschließend erfolgt eine sehr kurze Behandlung der allgemeinen Grundlagen. Sie beginnt mit den Widersprüchen der Logik, geht über Axiomatik, GOEDEL-Theorem, TURING-Automat und CHURCH-These und endet bei N- und NP-Problemen.

# Eine Auswahl von ergänzender Literatur

Churchland: „Ist die denkende Maschine möglich?“; Spektrum der Wissenschaft, März 1990 S. 40 - 54.

Falletta, N.: Paradoxien. Fischer - Logo, Frankfurt/M. 1988/1990

Gödel, K.: „Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme“. Monatshefte für Mathematik und Physik 38(1931), 173 - 198

Searle, J. R. u. a.: „Ist der menschliche Geist ein Computerprogramm?“ Spektrum der Wissenschaft, März 1990, S. 40 - 54

Turing, A. M.: „On computable numbers with applications to the Entscheidungsproblem“. Proc. London Math. Soc. 42(1937), 230 - 266

Völz, H.: Grundlagen der Information. Akademie - Verlag, Berlin 1991

Völz, H.: Wissen - Erkennen - Information. Allgemeine Grundlagen für Naturwissenschaft, Technik und Medizin. Shaker Verlag, Aachen 2001