

# **41TD1**

## ***Technik der DV-Anlagen - Teil 1***

### **1. Klassische Digitalschaltungen an Beispielen**

- Digitale Zeitstufen
- Fernsteuerung nach dem Impulszählverfahren
- Schieberegister-Interfaces
- Serialisierung/Deserialisierung
- Digitale Modulationsverfahren
- Taktgeneratoren, Rücksetzschaltungen und analoge Zeitstufen

### **2. Bussysteme**

- Grundlagen
- Alternativen zum Bus
- Auslegung von Bussystemen

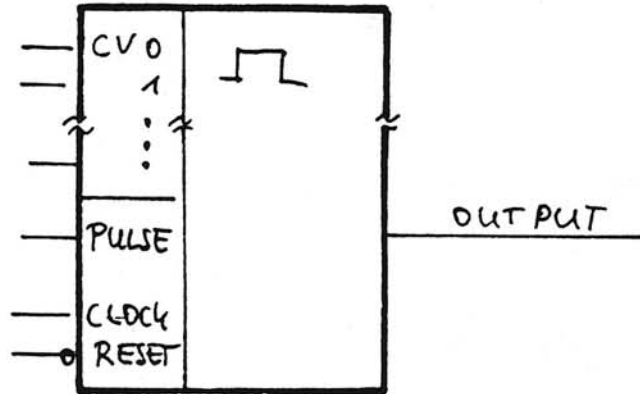
### **3. Speichersubsysteme**

- Auslegung von Speichersubsystemen
- Prinzipien der Speicherverwaltung
- Speicherverwaltungshardware
- Caches

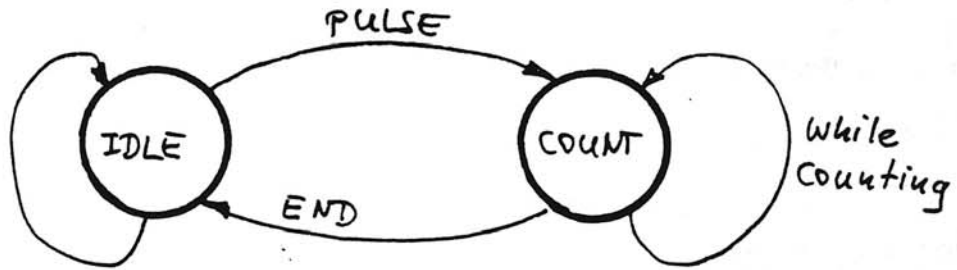
### **4. Leistungsgrenzen von Universalrechnern**

# Digitale Zeitgeber

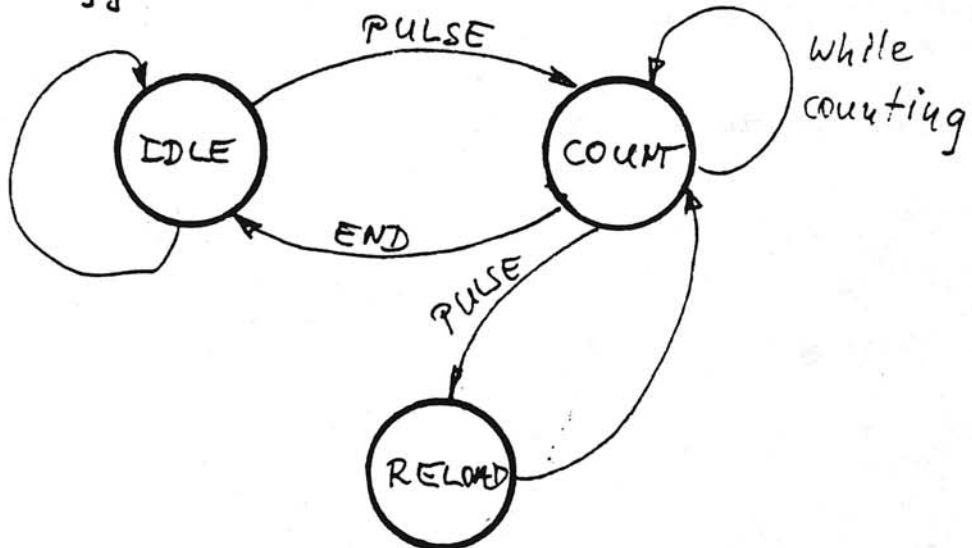
a) Symbol

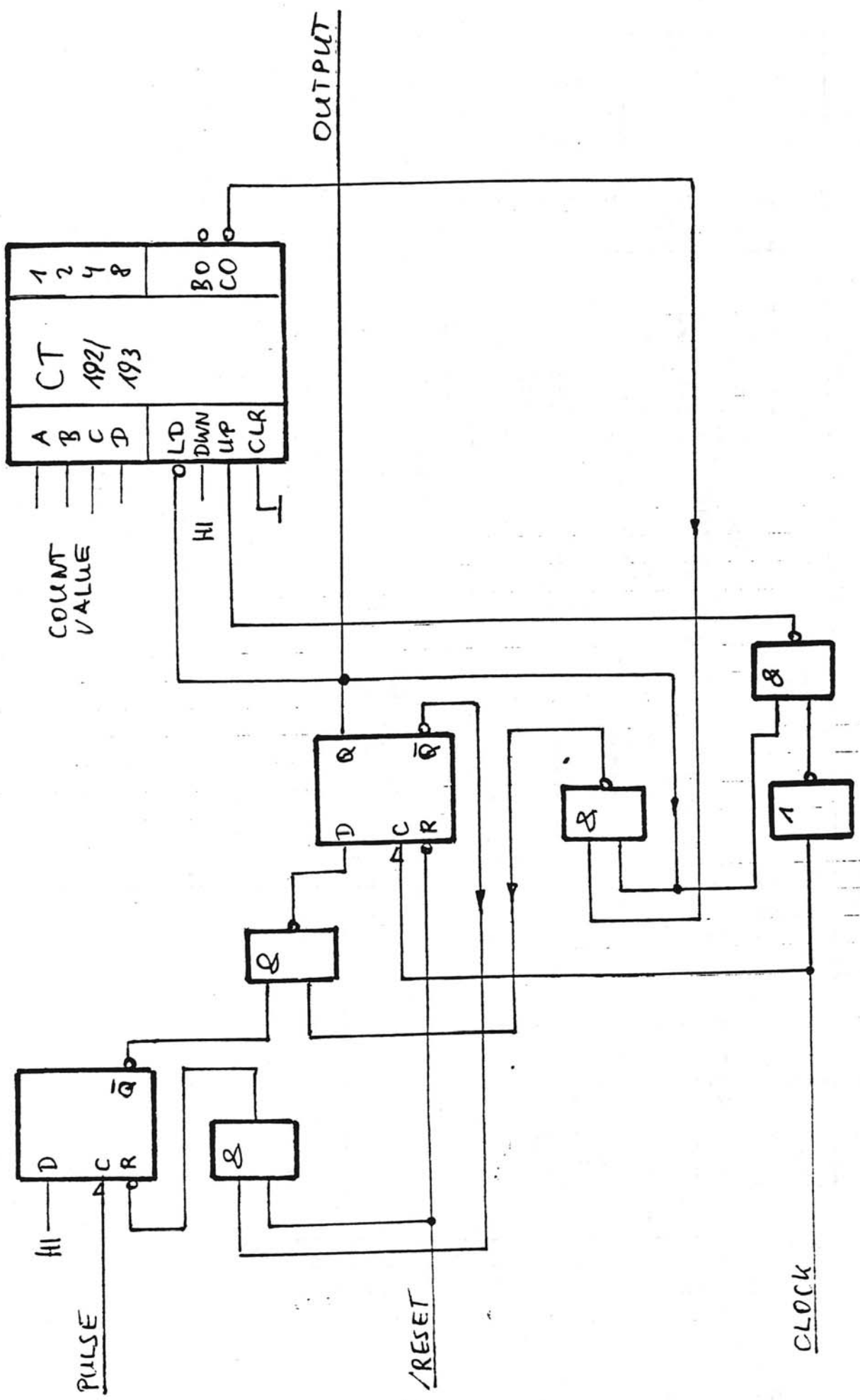


b) nicht retrIGGERbar



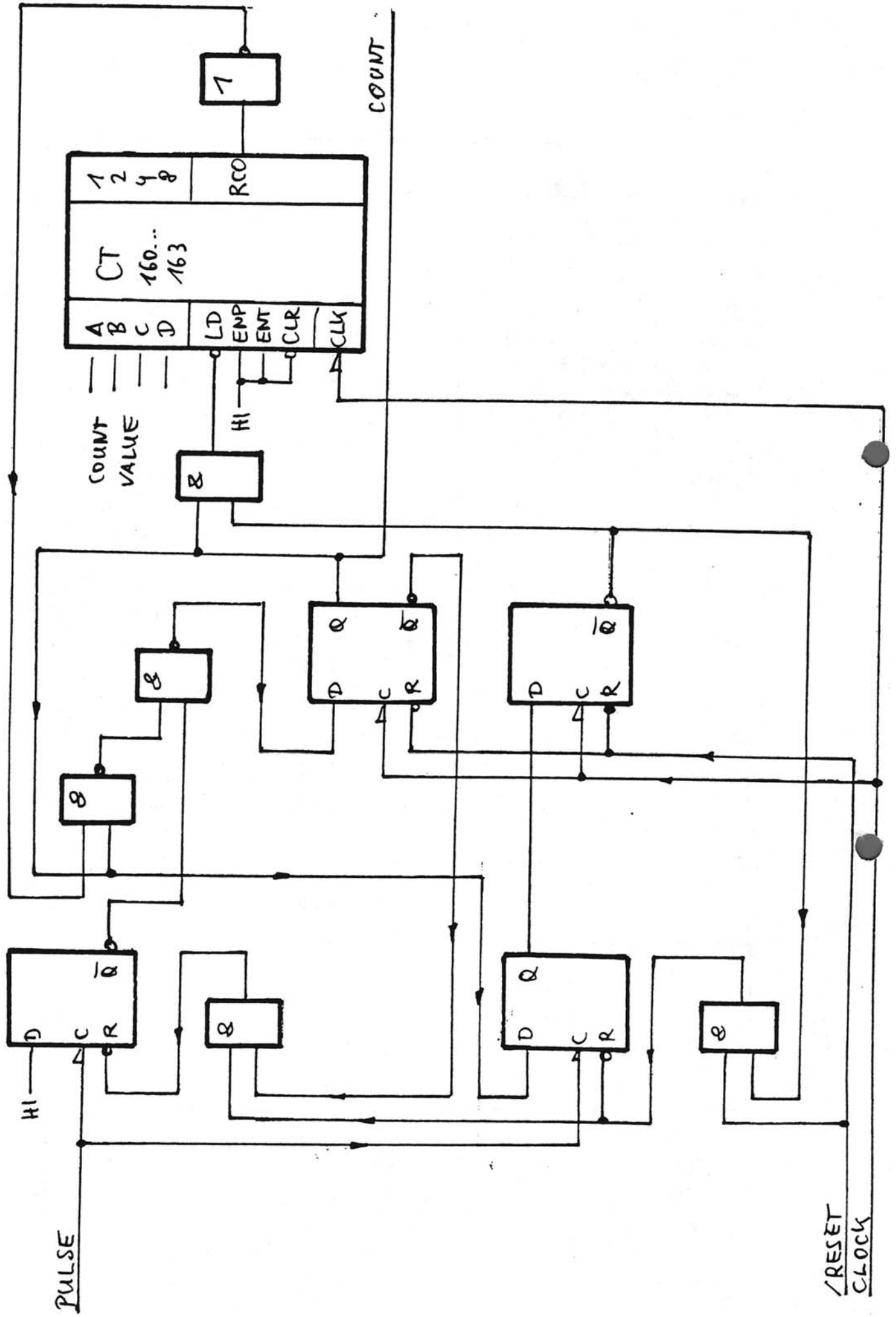
c) retrIGGERbar



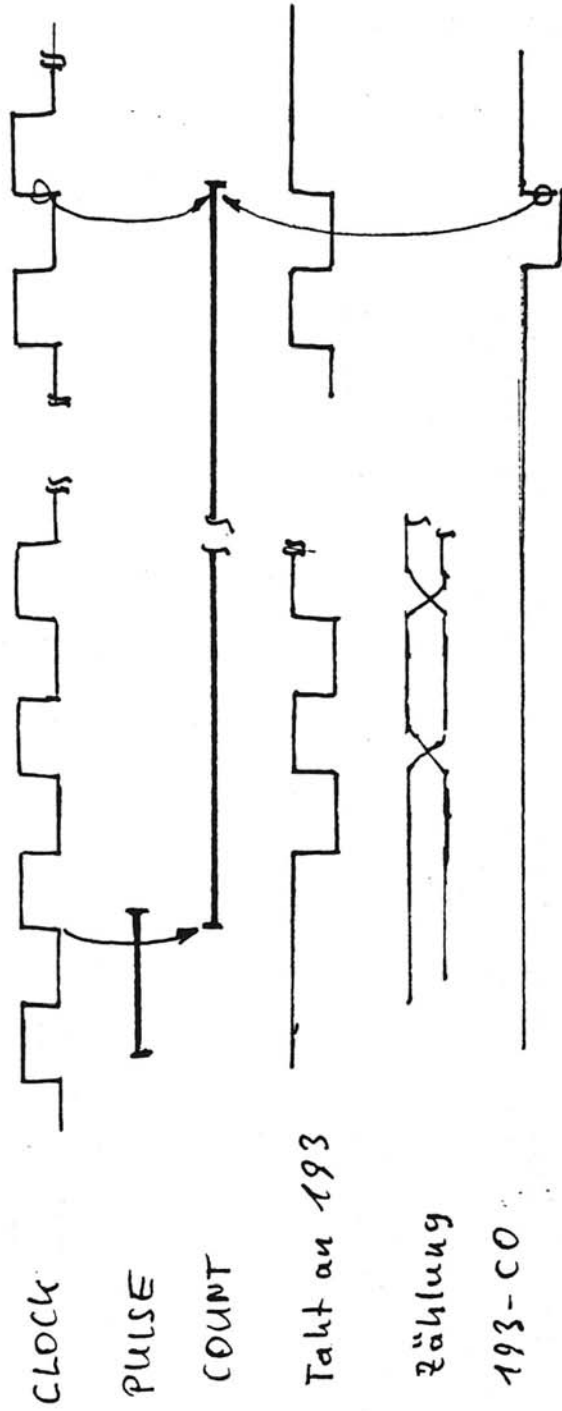


Digitaler Zeitgeber (I): nicht retriggebar

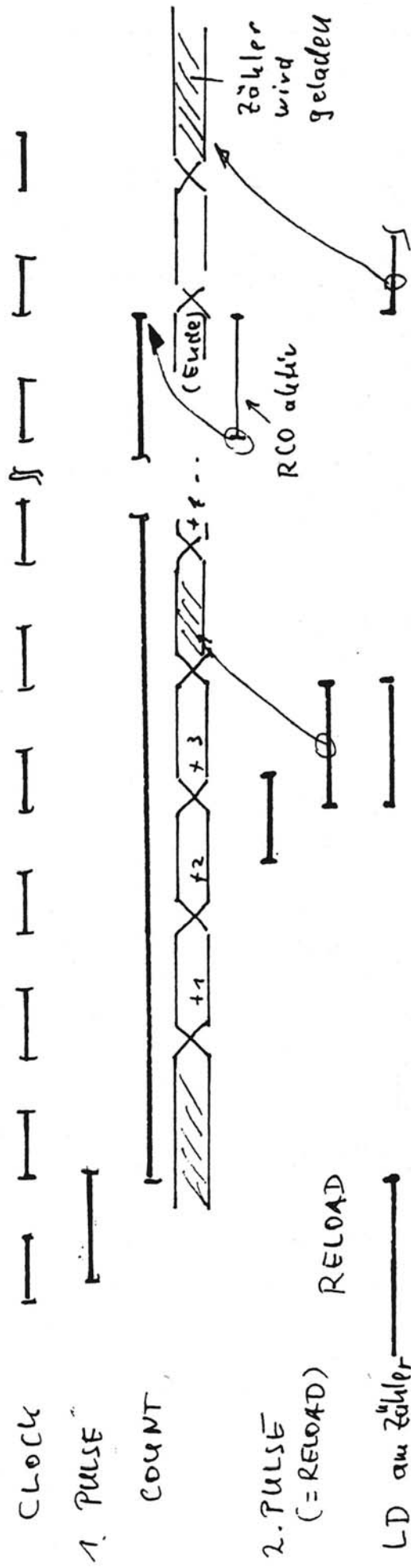
# Digitaler Zeitgeber (II): retriggerbar



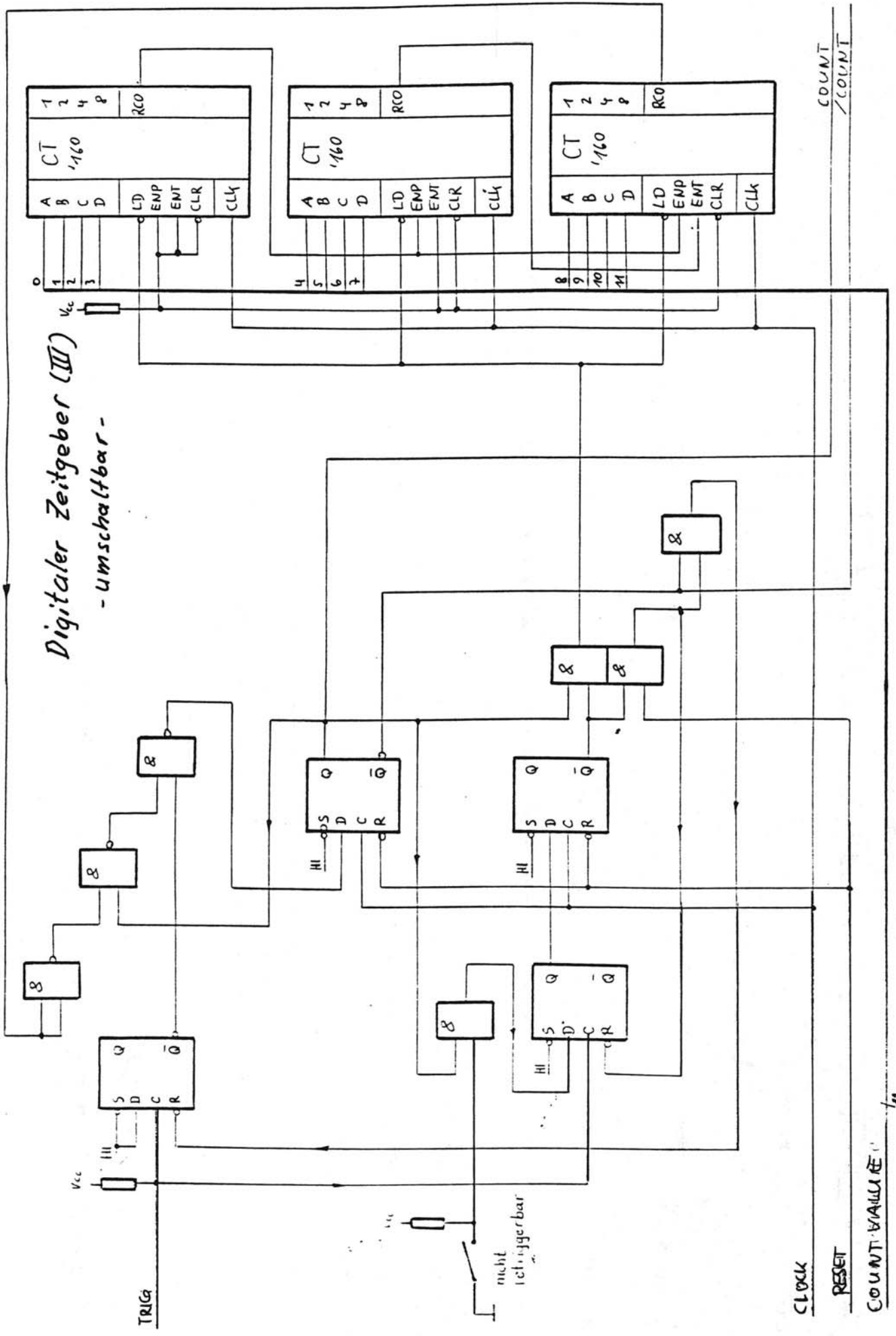
(I): nicht retriggerbar

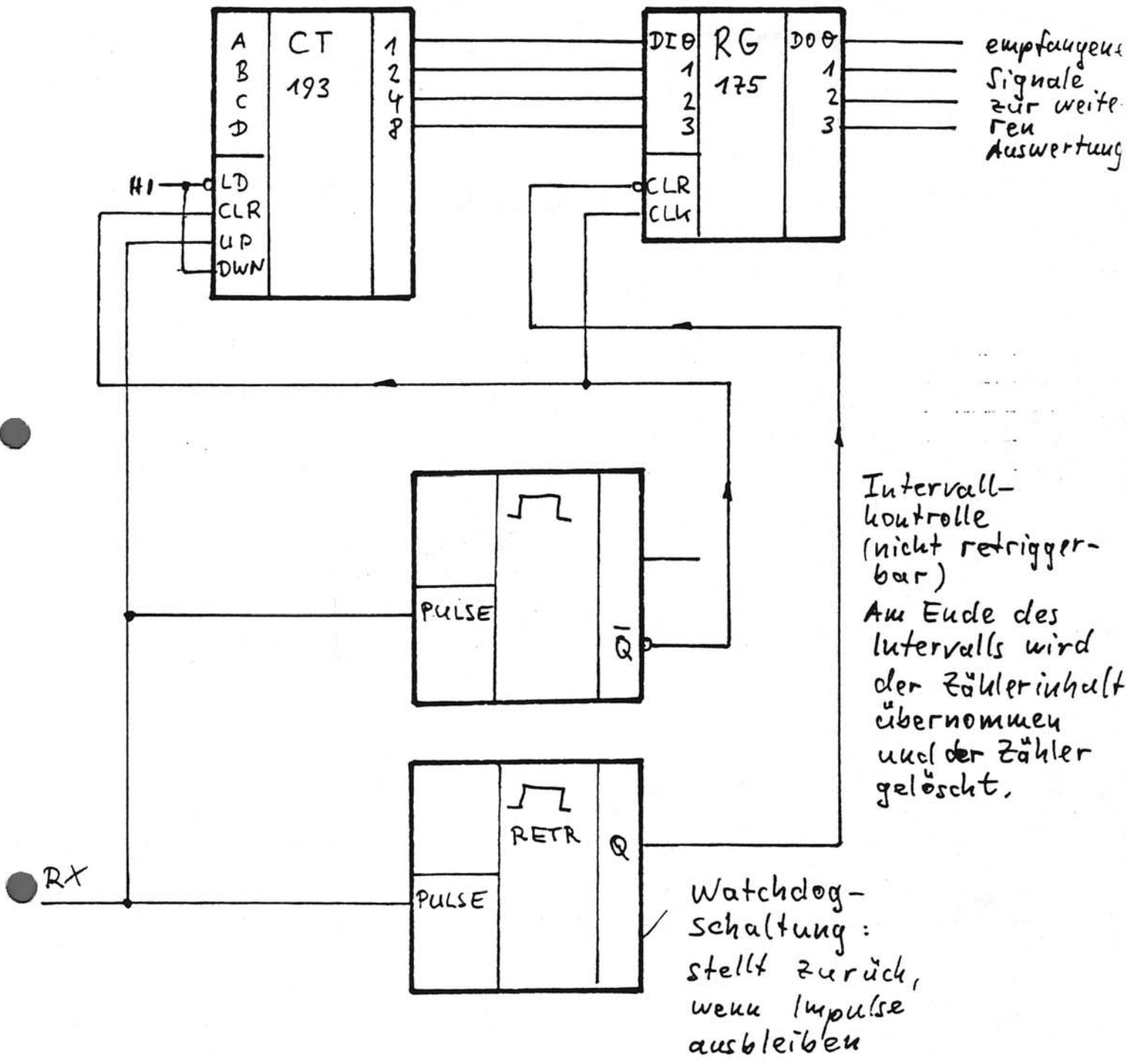


(II): retriggerbar



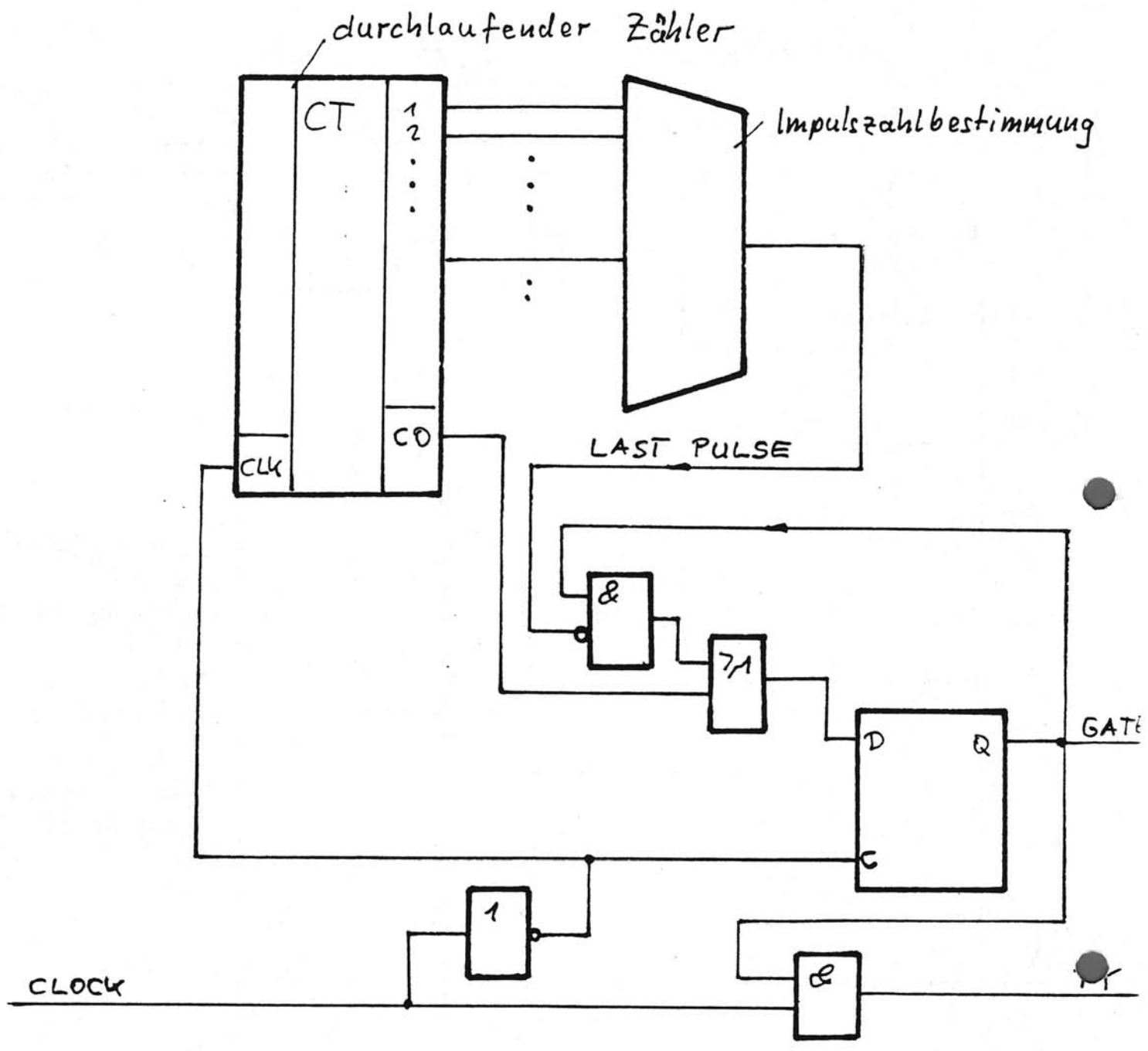
# Digitaler Zeitgeber (III) - Umschaltbar -





Fernsteuerung nach dem Impulszählverfahren

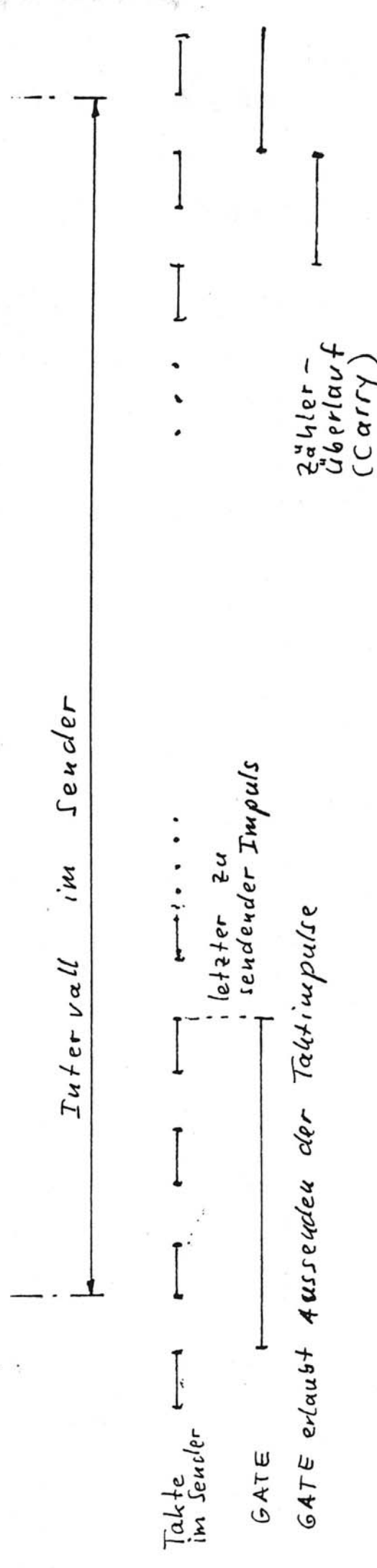
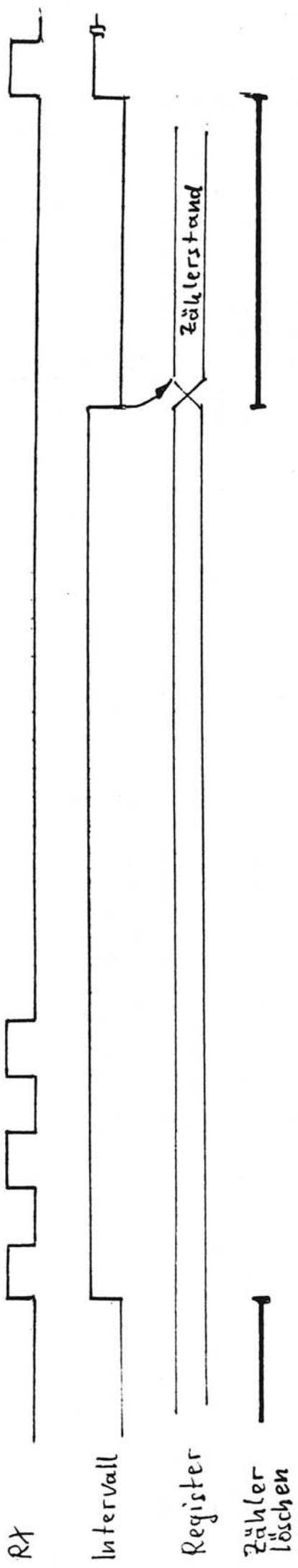
Empfänger-Grundschialtung



Sender- Grundschaltung



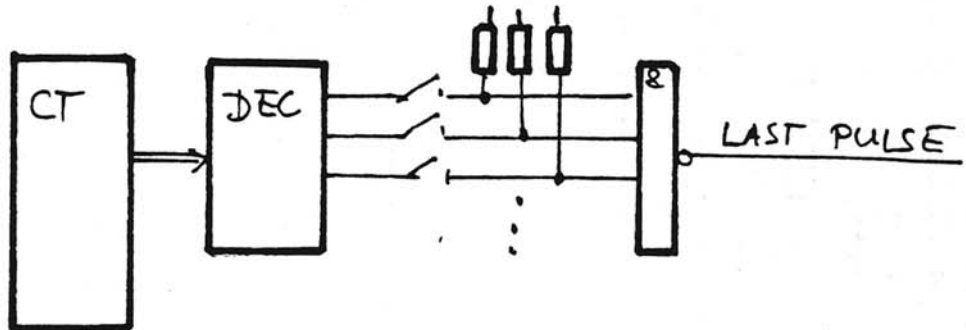
# Fernsteuerung nach dem Impulszählverfahren: Prinzip-Ablauf



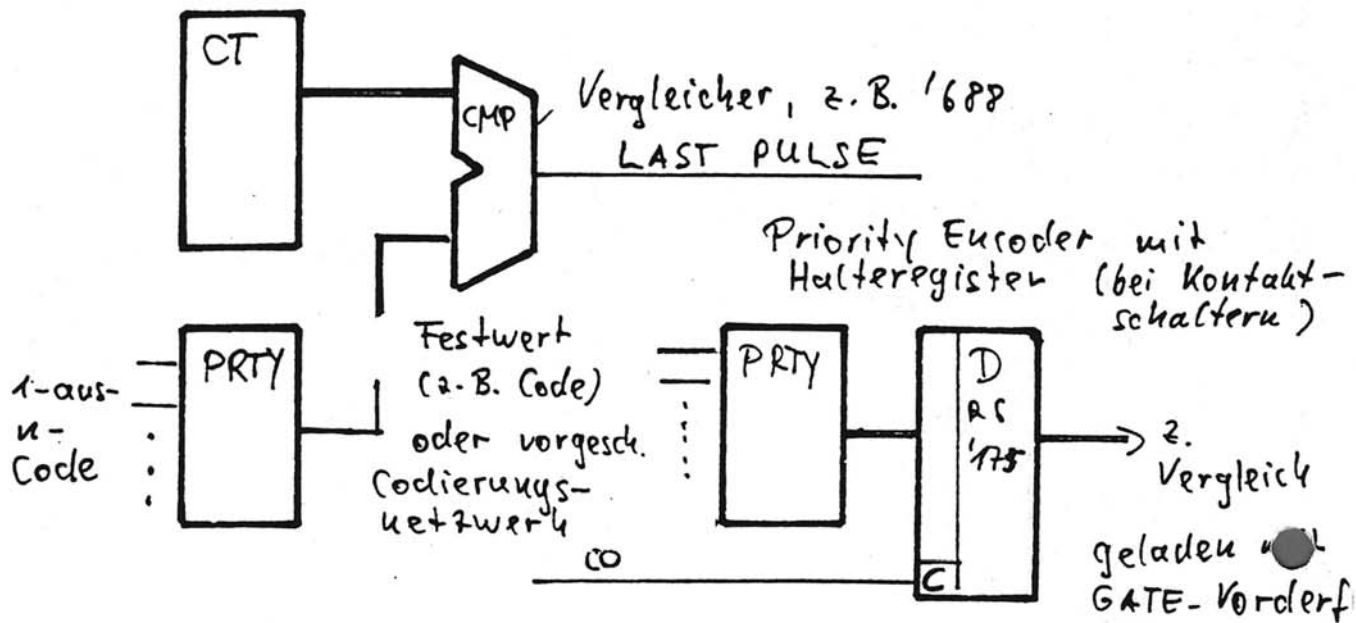
Zeitbedingungen: Es sind maximal  $n$  Impulse zu senden. Taktdauer im Sender:  $t_c$   
 Intervall im Empfänger  $I_R > n t_c$  (Toleranzen beachten; worst case!)  
 Intervall im Sender  $I_T > I_R$  ;  $I_T - I_R$  muß im Empfänger für die Auswertung / die gewünschten Wirkungen  $\infty$

# Impulszahlbestimmung

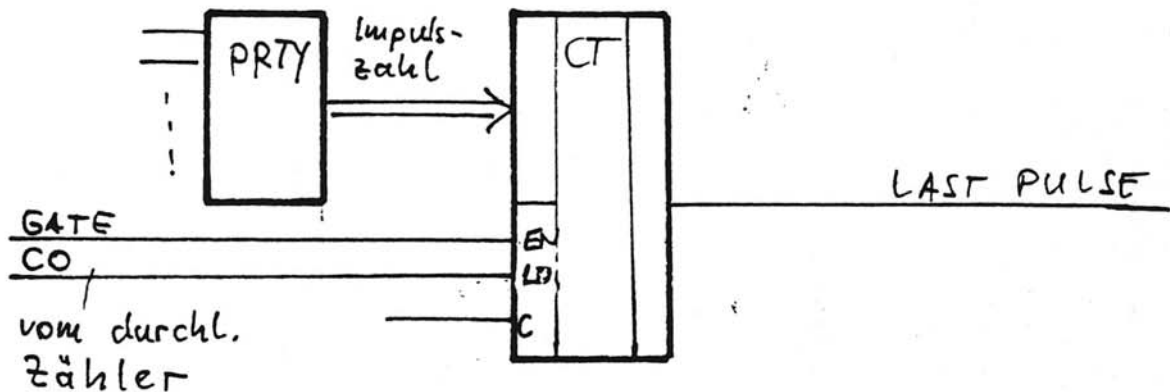
a) durch Decodierung



b) durch Vergleich



c) durch ladbaren Hilfszähler



Grundsätzlich: "Keine Erregung" erfordert Aussenden von 2<sup>n</sup> Impulsen.

# Simultanbetrieb

- Mehrere Ein-Aus-Funktionen sollen gleichzeitig und unabhängig wirksam werden können.-

a) Keine Codierung / Decodierung.

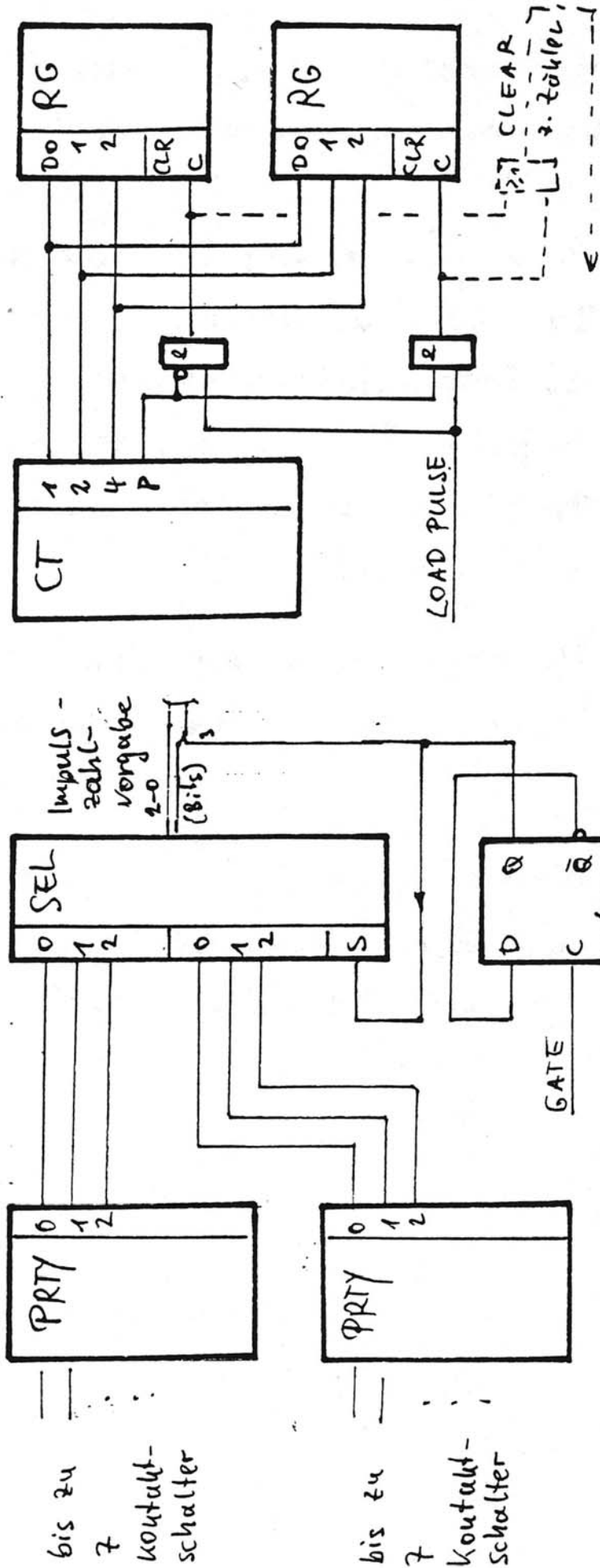
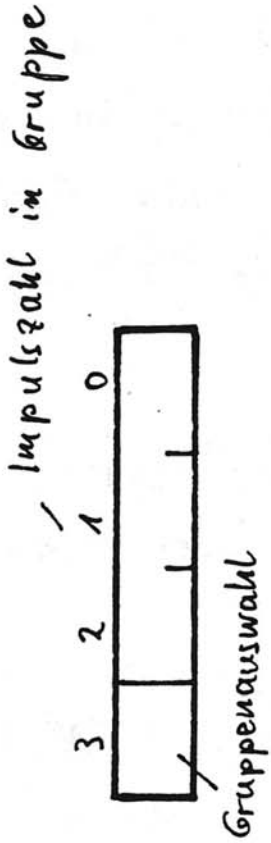
Das Ein-Aus-Bitmuster wird direkt als Binärzahl interpretiert. Es werden so viele Impulse übertragen, wie diese Zahl angibt.  
Problem: Bandbreite bzw. Zeit.  $2^n$  Impulse für  $n$  Bits!

b) Mehrere Gruppen jeweils abhängiger Funktionen; zeitmultiplexe Übertragung

c) Unabhängige, separat stellbare Halteflipflops auf der Empfängerseite. Es wird bei jeder Änderung die betr. Bitadresse übertragen

# Gruppenweise zeitmultiplexe Simultanübertragung (Beispiel)

Interpretation der binären Impulszahlvorgabe:



Sendeseite

Empfangsseite

# Schieberegister-Interfaces

## Anwendungen

- low-cost-Interfaces (Alternative zum Feldbus)
- Diagnose von Digitalschaltungen (Scan Design)
- Hochleistungs-Bussysteme

## Vorteile

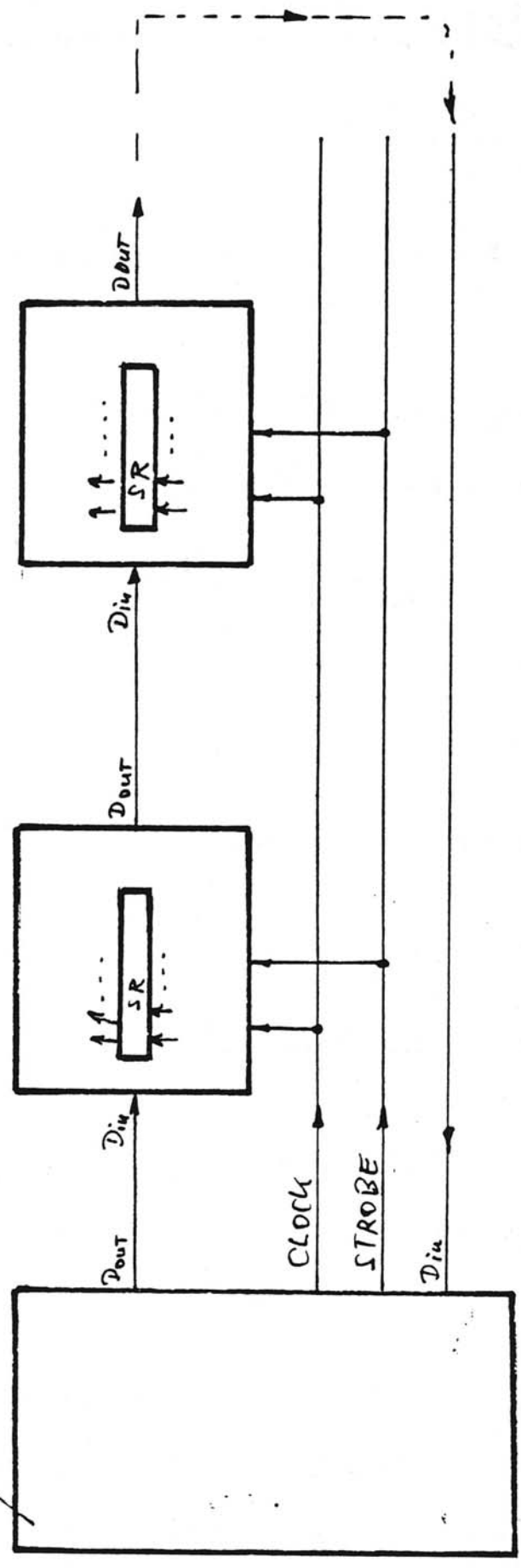
- Einfachste Schaltungstechnik
- Geringer Aufwand
- Nutzbar von "Low Cost" bis "High Performance"
- Bewegen vieler Bits mit einfachsten Mitteln
- Gut prüfbar

## Nachteile

- Kein "echter" Bus; Datenweg nicht durchgehend; Praxisprobleme beim Entfernen / Hinzufügen von Funktionseinheiten; Gesamter Ausfall bei Unterbrechung
- Takttoleranzprobleme

Zentrale Steuerung

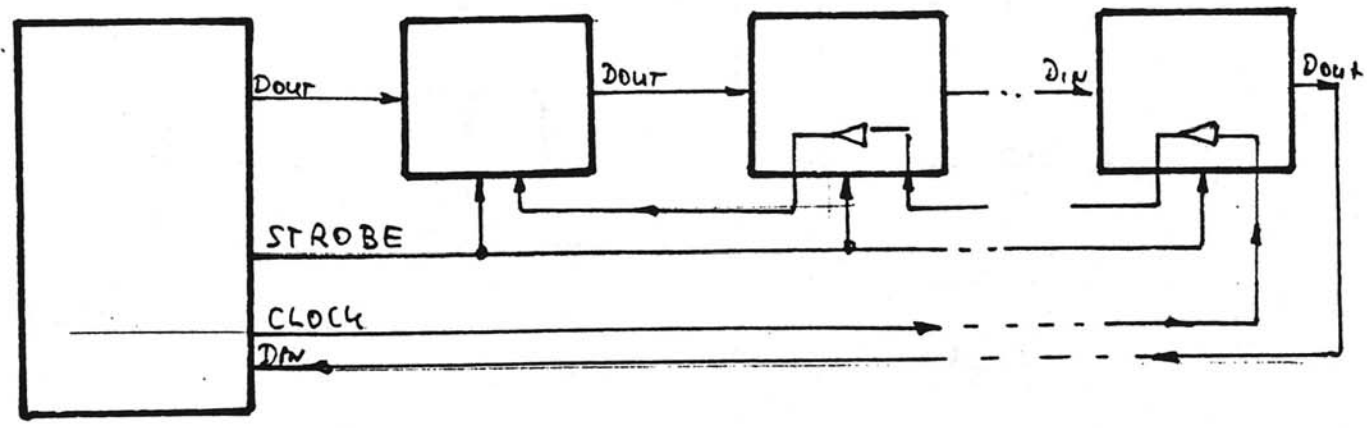
Angeschlossene Einrichtungen



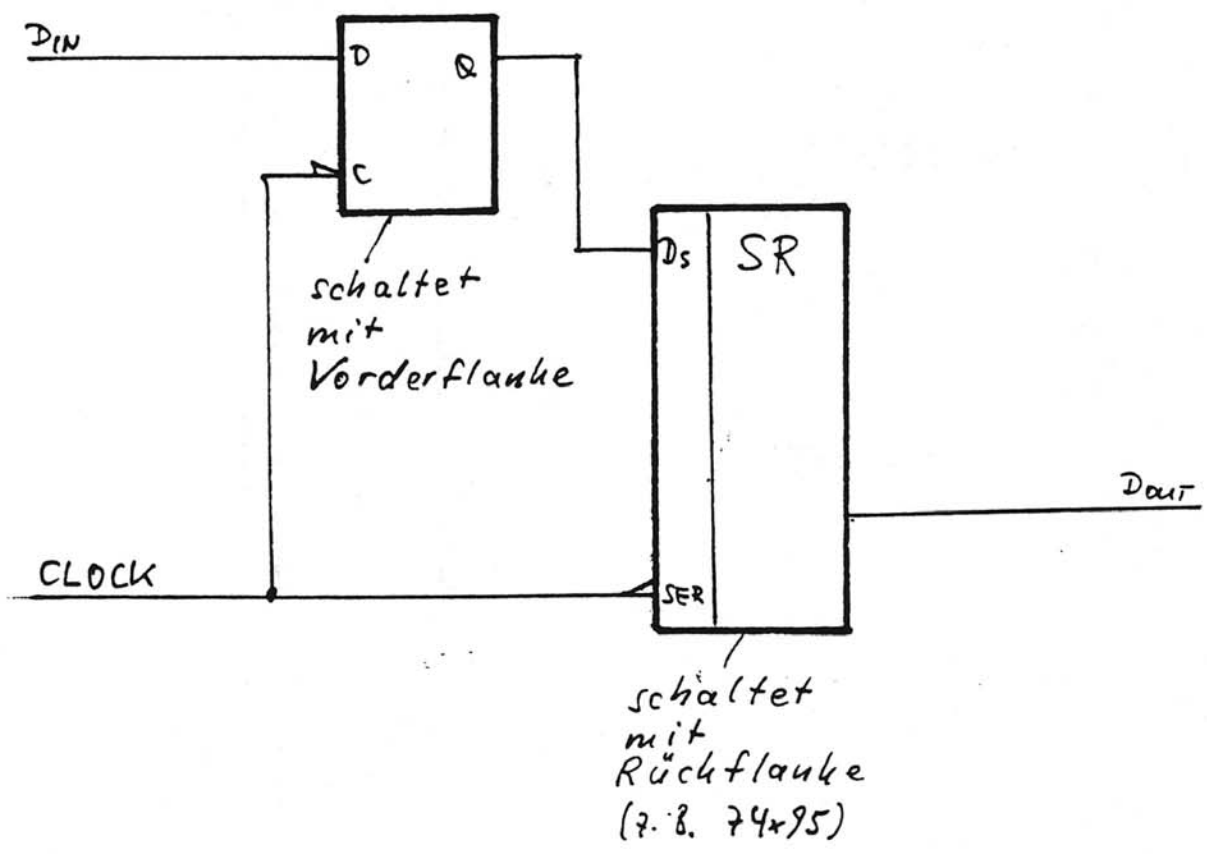
Elementares Schieberegister-Interface

# Eliminierung des Takttoleranzproblems

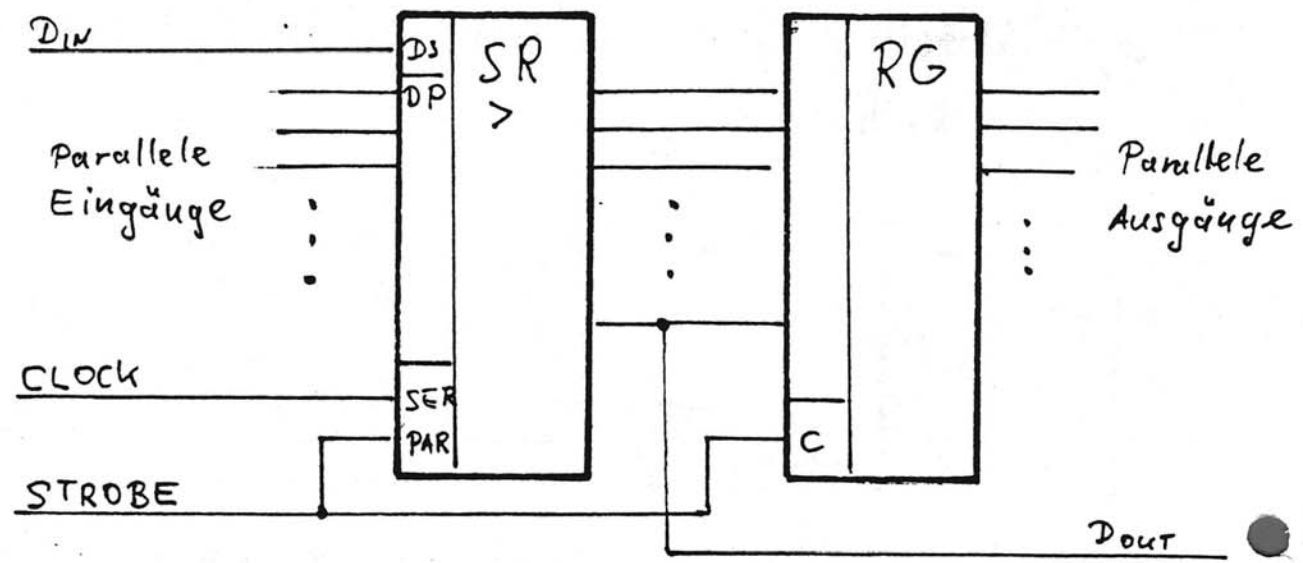
a) Takteinspeisung im Gegenstrom, Takttreiber in Reihe



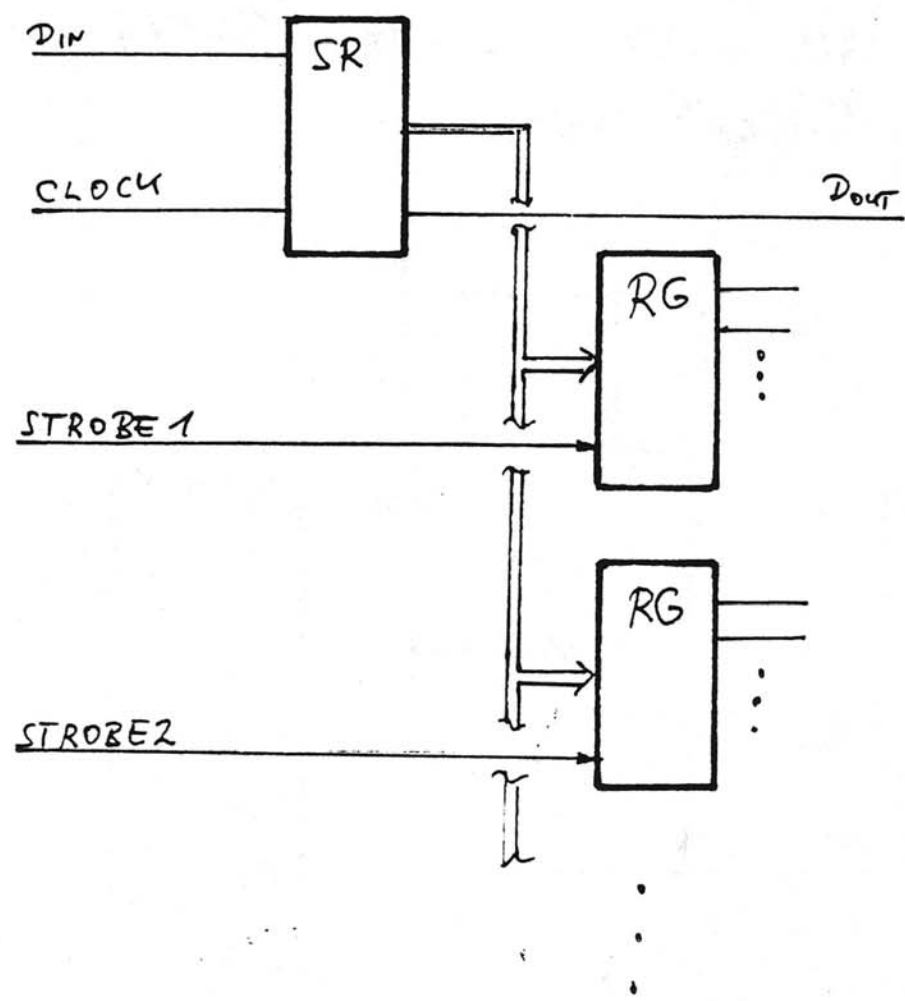
b) Ausnutzung beider Taktflanken mittels Hilfs-Flipflop



# Prinzip der Informationsübernahme



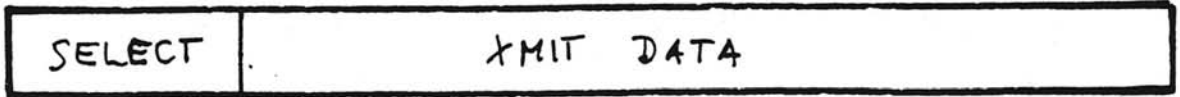
Mehrere Bestimmungen über mehrere Strobes



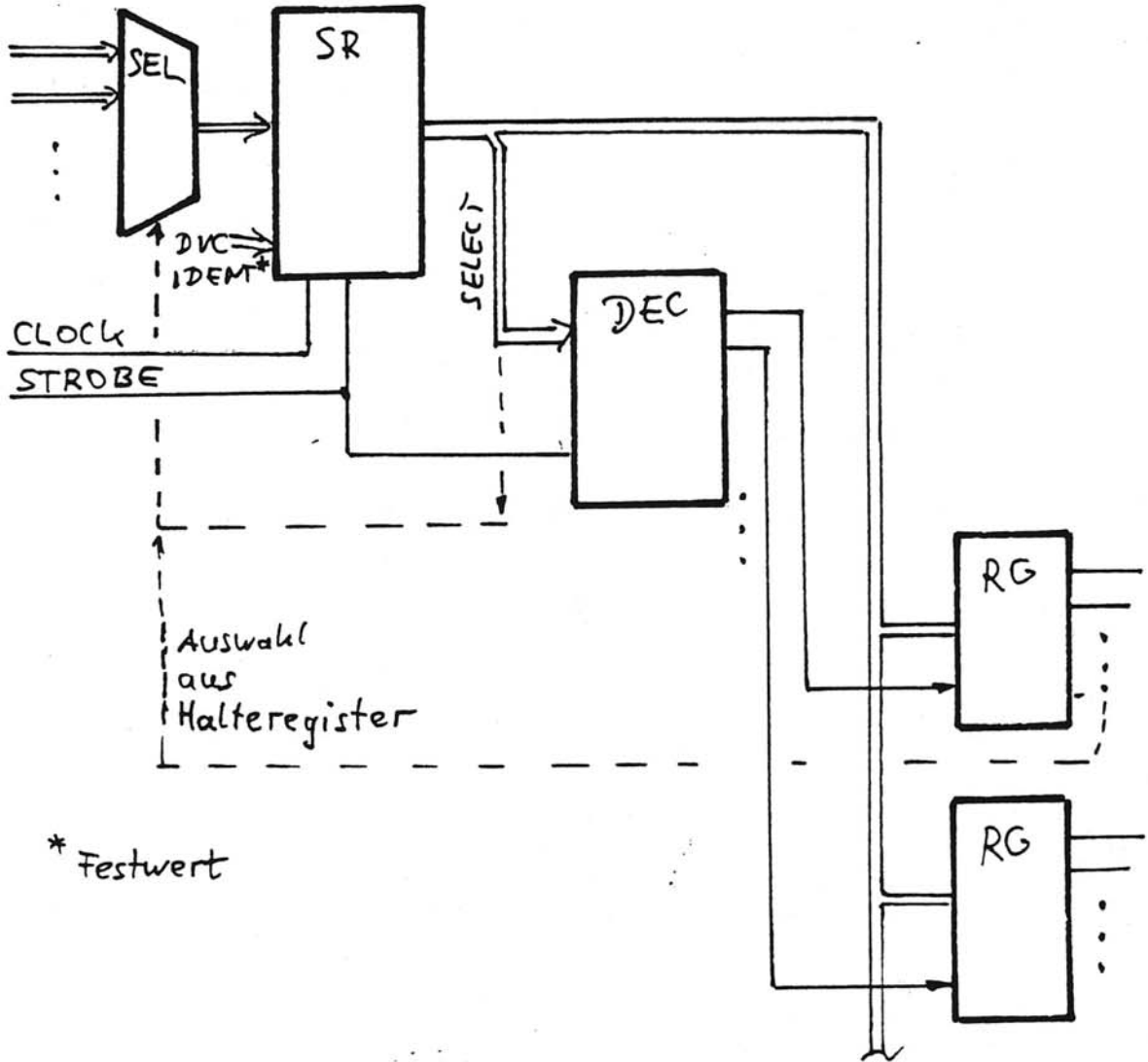
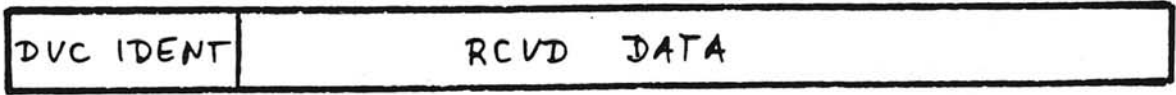


# Mehrere Bestimmungen / Quellen durch Auswahl

Gesendete Daten (für eine Einrichtung)

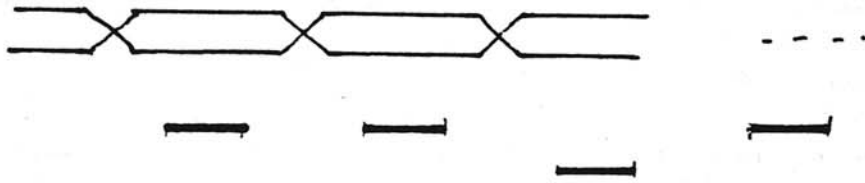


Empfangene Daten

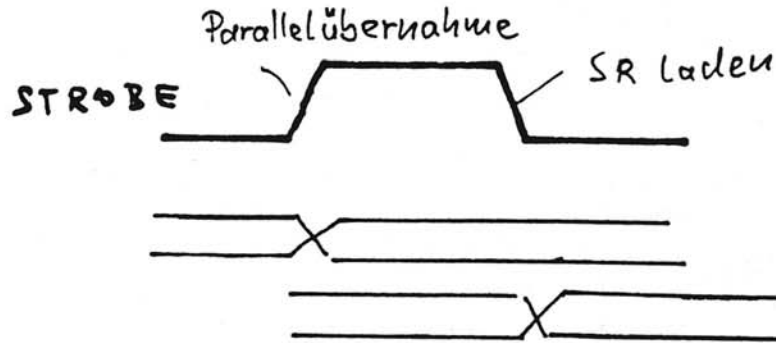


# Alternativen für das STROBE-Timing

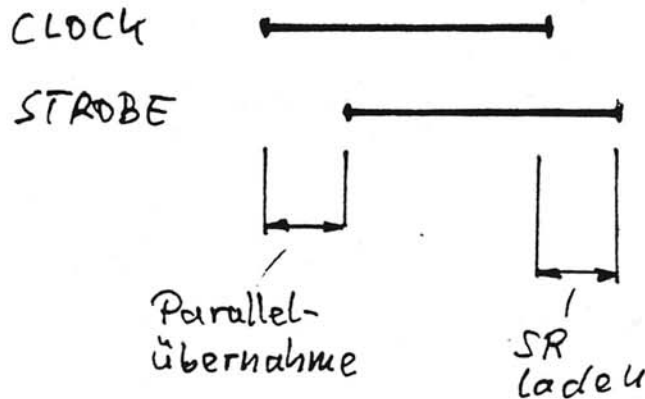
a) STROBE ist Taktsignal für "Übernahme" (1 Flanke wirkt)



b) Wirtschaftlichkeit beider Flanken



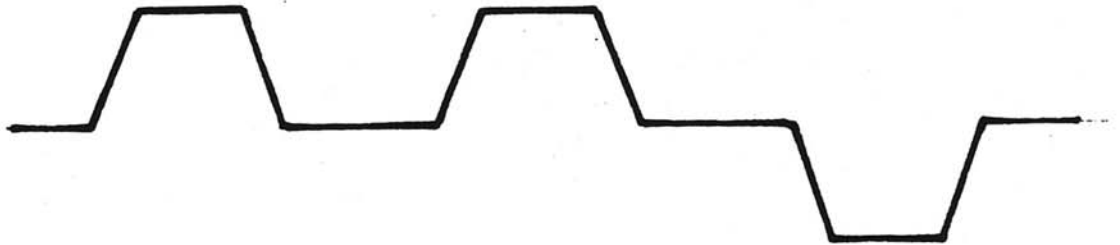
c) Überlappung von CLOCK und STROBE



d) Interpretation der STROBE-Leitung als Modussteuerung (mittels Schaltwerk). Beispiel: Boundary Scan.

# Einsparung der STROBE-Leitung

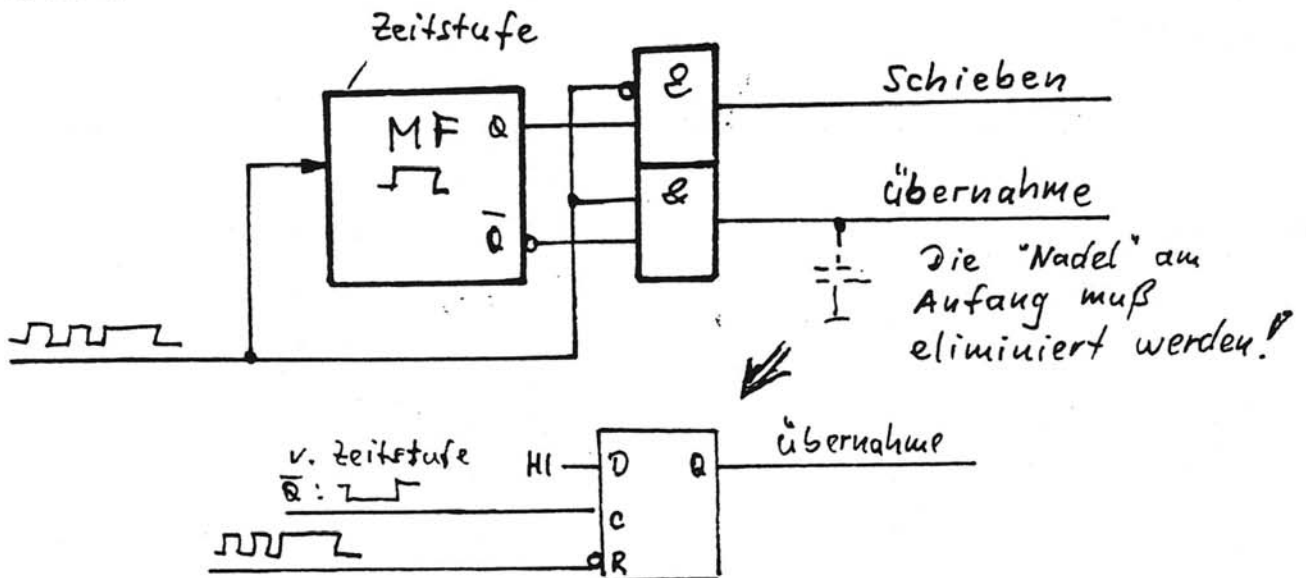
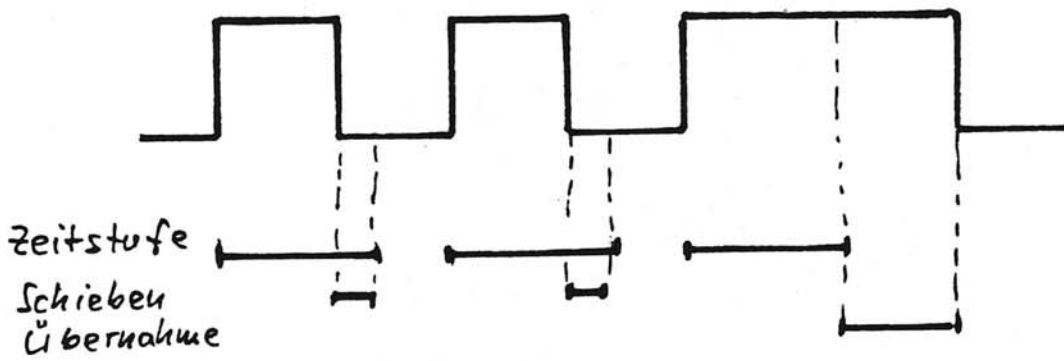
a) Positive und negative CLOCK-Impulse



Positive Impulse: Schieben (CLOCK)

Negative Impulse: Übernahme (STROBE)

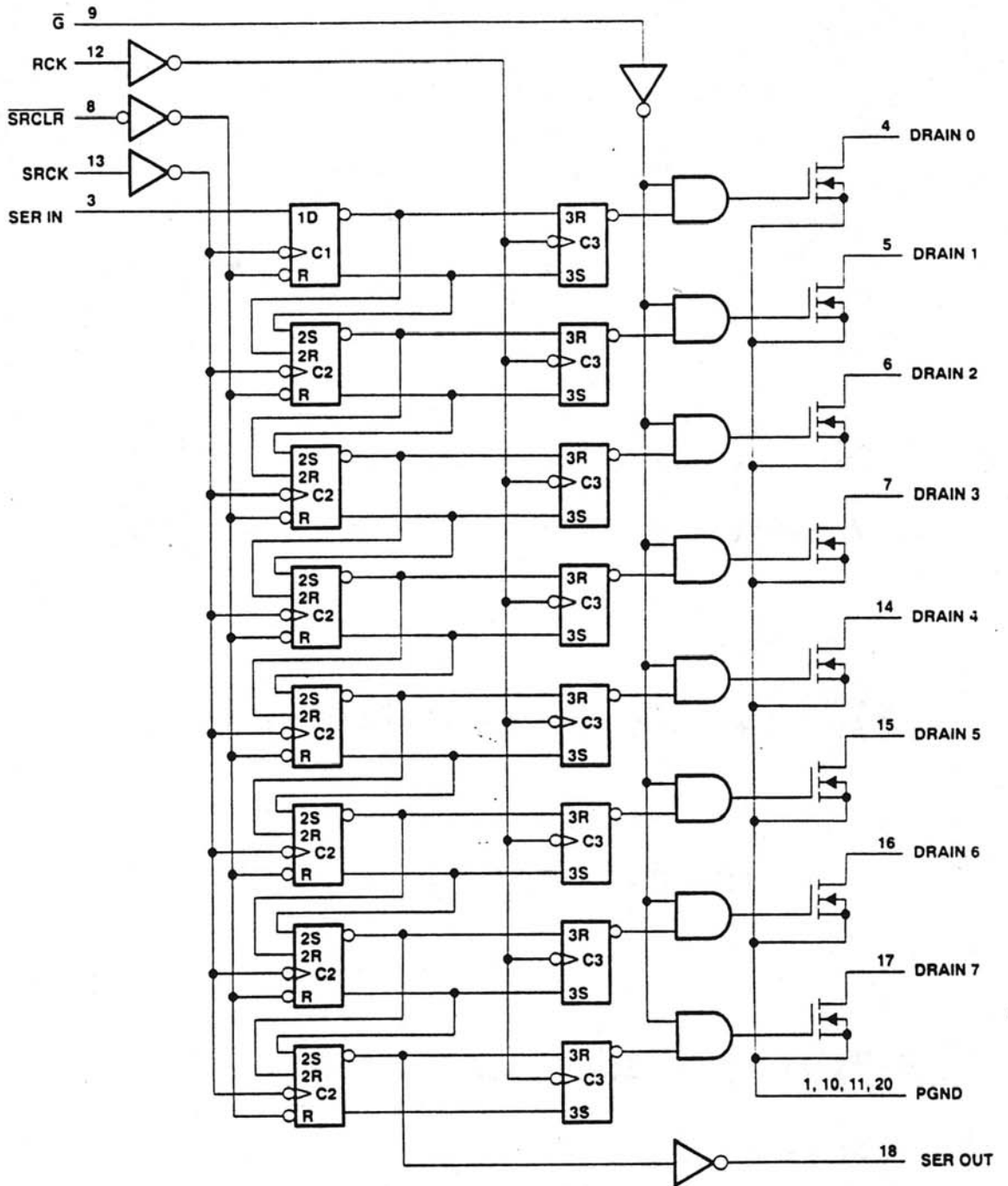
b) Impulsbreitenmodulation



# TPIC6595 POWER LOGIC 8-BIT SHIFT REGISTER

SLPS029-D4009, APRIL 1992

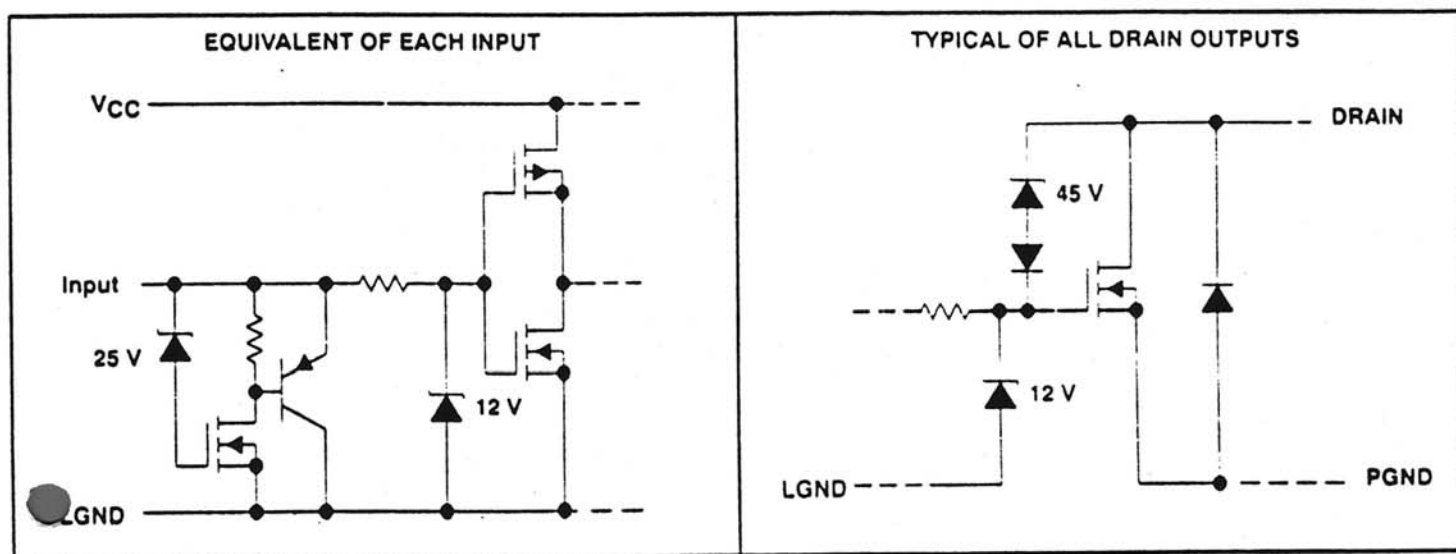
logic diagram (positive logic)



# TPIC6595 POWER LOGIC 8-BIT SHIFT REGISTER

SLPS029-D4009, APRIL 1992

## schematic of inputs and outputs



absolute maximum ratings over recommended operating case temperature range (unless otherwise noted)<sup>†</sup>

Logic supply voltage, $V_{CC}$ (see Note 1)	7 V
Logic Input voltage range, $V_I$	-0.3 V to 7 V
Power DMOS drain-to-source voltage, $V_{DS}$ (see Note 2)	45 V
Continuous source-drain diode anode current	1 A
Pulsed source-drain diode anode current	2 A
Pulsed drain current, each output, all outputs on, $I_{DN}$ , $T_A = 25^\circ\text{C}$ (see Note 3)	750 mA
Continuous drain current, each output, all outputs on, $I_{DN}$ , $T_A = 25^\circ\text{C}$	250 mA
Peak drain current single output, $I_{DM}$ , $T_A = 25^\circ\text{C}$ (see Note 3)	2 A
Single-pulse avalanche energy, $E_{AS}$ (see Figure 4)	75 mJ
Avalanche current, $I_{AS}$ (see Note 4)	1 A
Continuous total dissipation at (or below) $T_A = 25^\circ\text{C}$ (see Note 5)	1.39 W
Operating virtual junction temperature range, $T_J$	-40°C to 150°C
Storage temperature range	-65°C to 150°C
Lead temperature	260°C

<sup>†</sup> Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

- NOTES
- All voltage values are with respect to LGND and PGND.
  - Each power DMOS source is internally connected to PGND.
  - Pulse duration  $\leq 100 \mu\text{s}$ , duty cycle  $\leq 2\%$ .
  - $V_{DD} = 15 \text{ V}$ , starting junction temperature,  $(T_{JS}) = 25^\circ\text{C}$ ,  $L = 100 \text{ mH}$ ,  $I_{AS} = 1 \text{ A}$ . See Figure 4.
  - For operation above 25°C free-air temperature, derate linearly at the rate of 11.1 mW/°C. To avoid exceeding the design maximum junction temperature, this rating should not be exceeded.

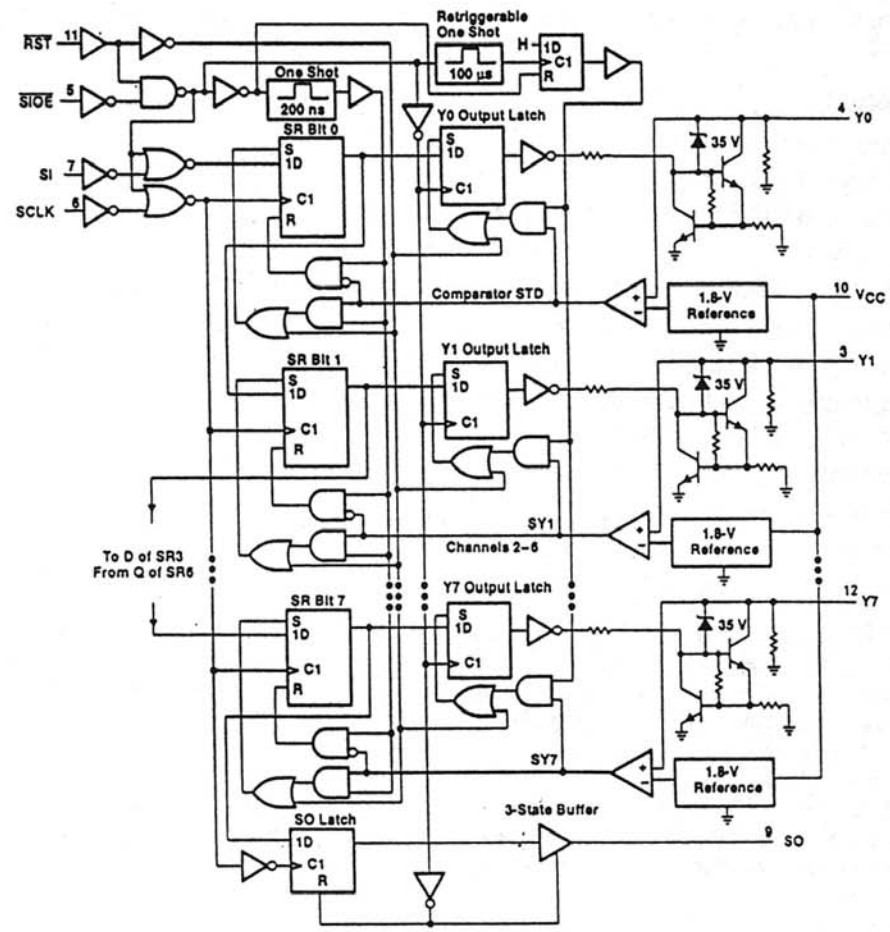
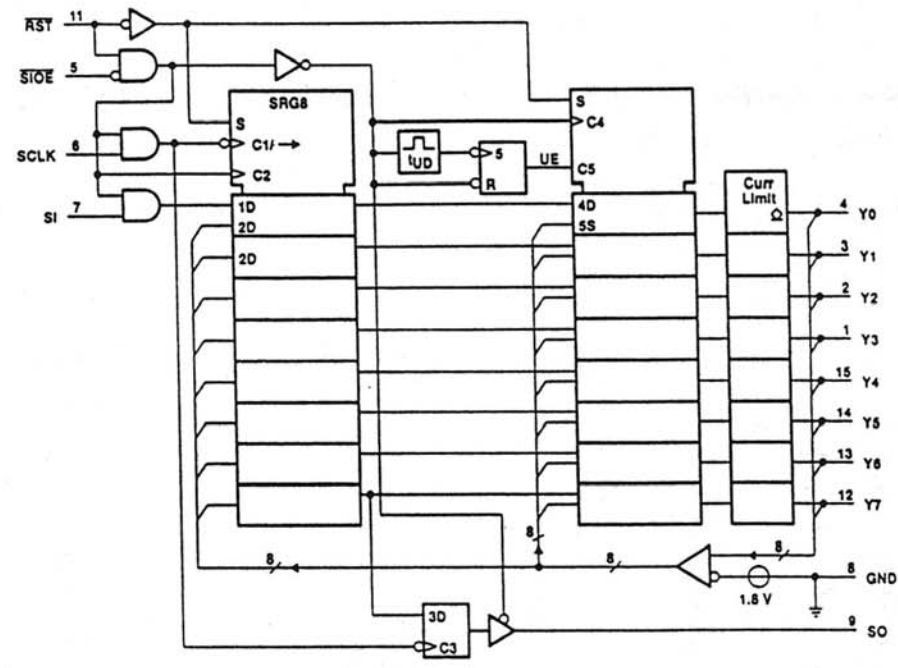


Abbildung B4-2.39

8-bit-Leistungsschalter mit seriellem Eingang (Octal Intelligent Power Switch) TPIC2801/2802 (Quelle: Texas Instruments)

absolute maximum ratings over operating temperature range (unless otherwise noted)

Supply voltage range, $V_{CC}$ (see Note 1)	-0.3 V to 7 V
Input voltage, $V_I$	7 V
Output voltage range at SO, $V_O$	-0.3 V to 7 V
Input current, $I_I$	-15 mA
Peak output sink current at Y, $I_O$ repetitive, $t_w = 10$ ms, duty cycle = 50%, (see Notes 2 and 3)	internally limited
Continuous output current at Y, $I_O$ (see Note 3)	1 A
Peak current through GND: Nonrepetitive $t_w = 0.2$ ms	-8 A
Repetitive $t_w = 10$ ms, duty cycle = 50%	-6 A
Output clamp energy, $E_{OK}$ (after turning off $I_O$ (on) = 0.5 A)	40 mJ
Continuous current through GND	-4.5 A
Continuous dissipation at (or below) $T_A = 25^\circ\text{C}$ (see Note 4)	3.575 W
Continuous dissipation at (or below) $T_C = 75^\circ\text{C}$ (see Note 4)	25 W
Operating case or virtual junction temperature range	-55°C to 150°C
Storage temperature range	-65°C to 150°C
Lead temperature 1, 6 mm (1/16 inch) from case for 10 seconds	260°C

- NOTES: 1. All voltage values are with respect to network GND.  
 2. Each Y output is individually current limited with a typical overcurrent limit of about 1.4 A.  
 3. Multiple Y outputs of this device can conduct rated current simultaneously; however, power dissipation (average) over a short time interval must fall within the continuous dissipation range and GND current range.  
 4. For operation above 25°C free-air temperature, derate linearly at the rate of 28.6 mW/°C. For operation above 75°C case temperature, derate linearly at the rate of 333 mW/°C. To avoid exceeding the maximum virtual junction temperature, these ratings must not be exceeded.

recommended operating conditions

	MIN	NOM	MAX	UNIT
Supply voltage, $V_{CC}$	4.75	5	5.25	V
High-level input voltage, $V_{IH}$	0.75 $V_{CC}$		5.25	V
Low-level input voltage, $V_{IL}$	-0.3		0.2 $V_{CC}$	V
Output voltage, $V_{O(off)}$			30	V
Continuous output current, $I_{O(on)}$			1	A
Operating case temperature, $T_C$	-40	25	105	°C

electrical characteristics over recommended ranges of supply voltage and operating case temperature (unless otherwise noted)

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
$I_{CC}$ Supply current	All outputs on, $I_O = 0.5$ A at all outputs	$T_J = 105^\circ\text{C}$		150	mA
		$T_J = 25^\circ\text{C}$		200	
		$T_J = -40^\circ\text{C}$		250	
	All outputs off	$T_J = 25^\circ\text{C}$	4	10	

Timing Chart

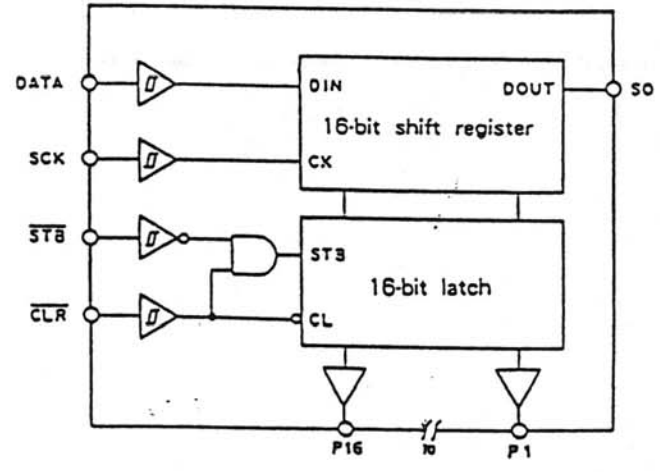
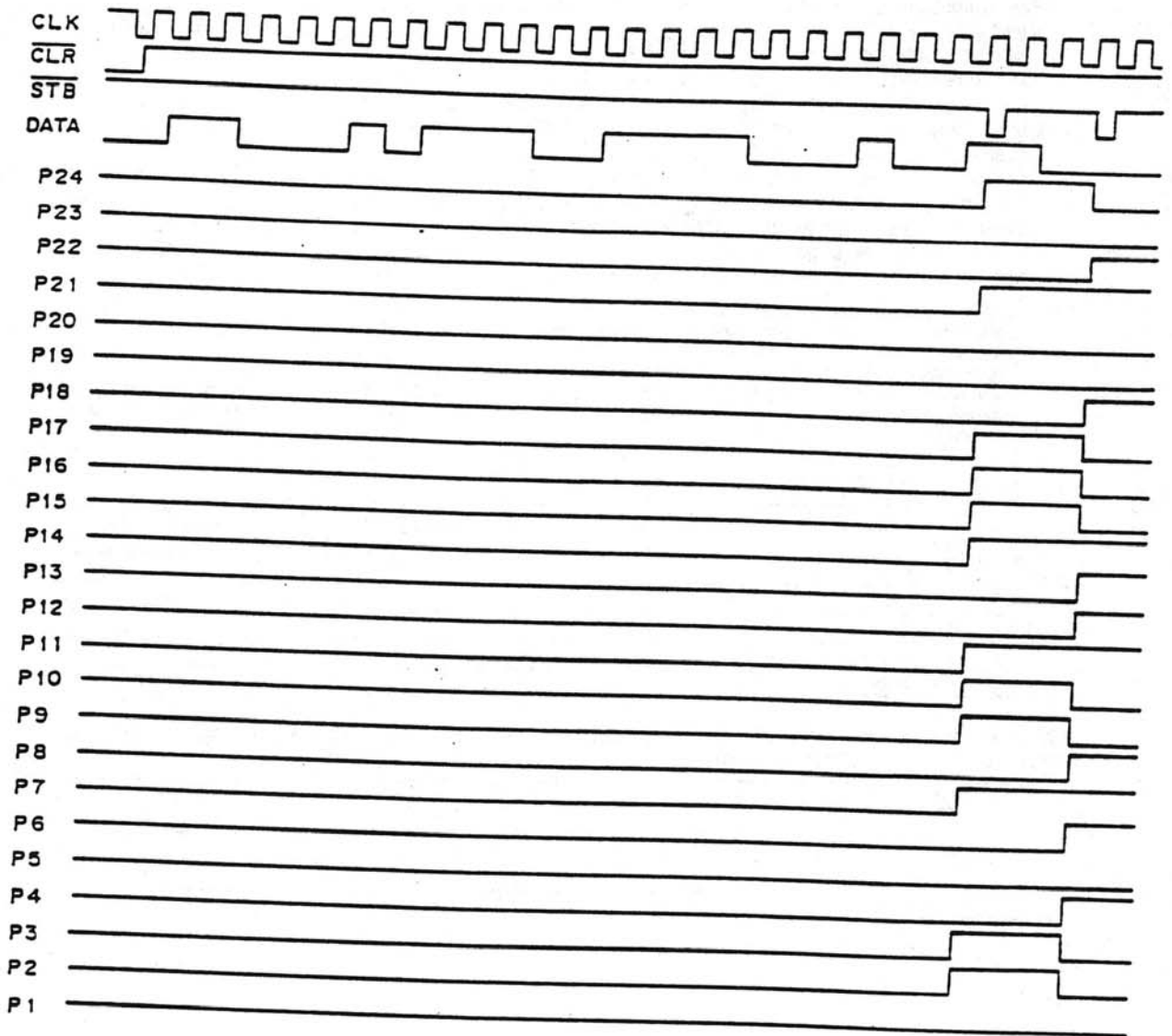
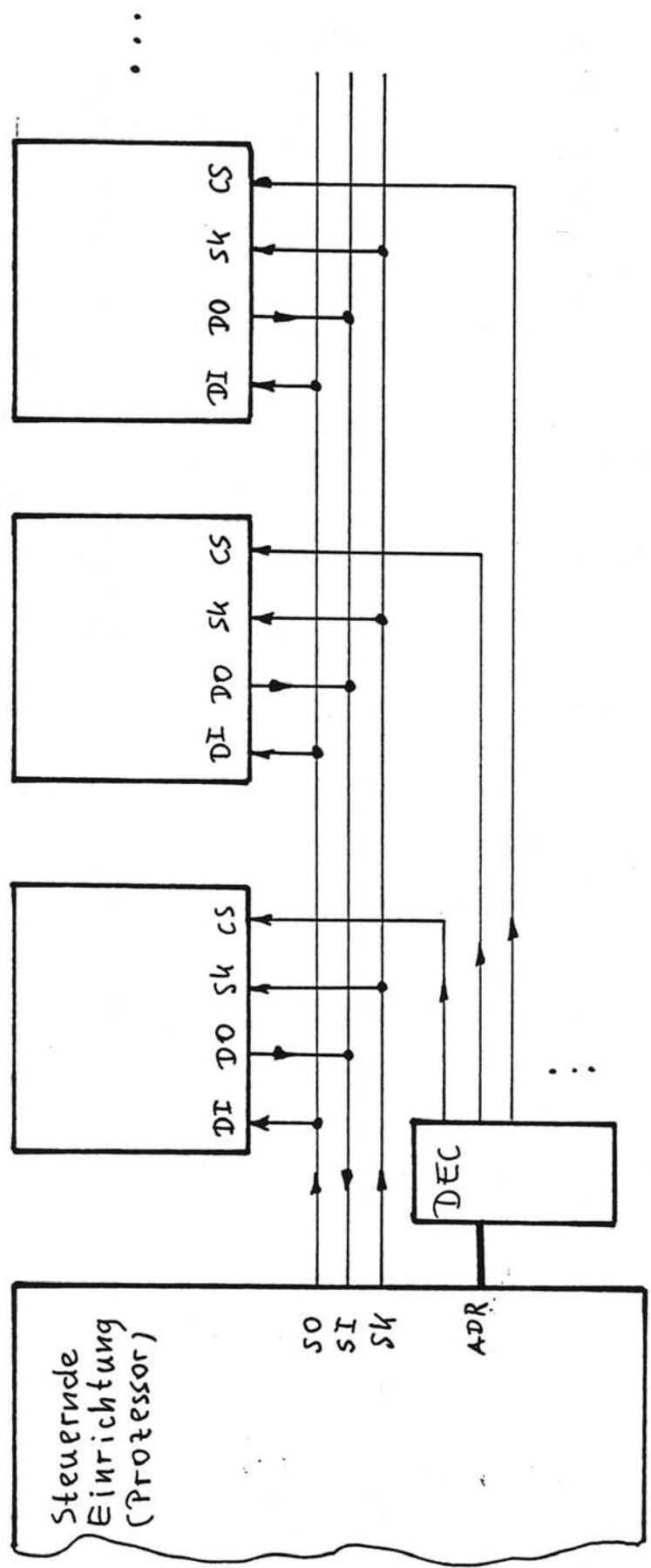


Abbildung B4-2.37  
 Serien-Parallelwandler-Schaltkreis CXP2003  
 (Quelle: Sony)

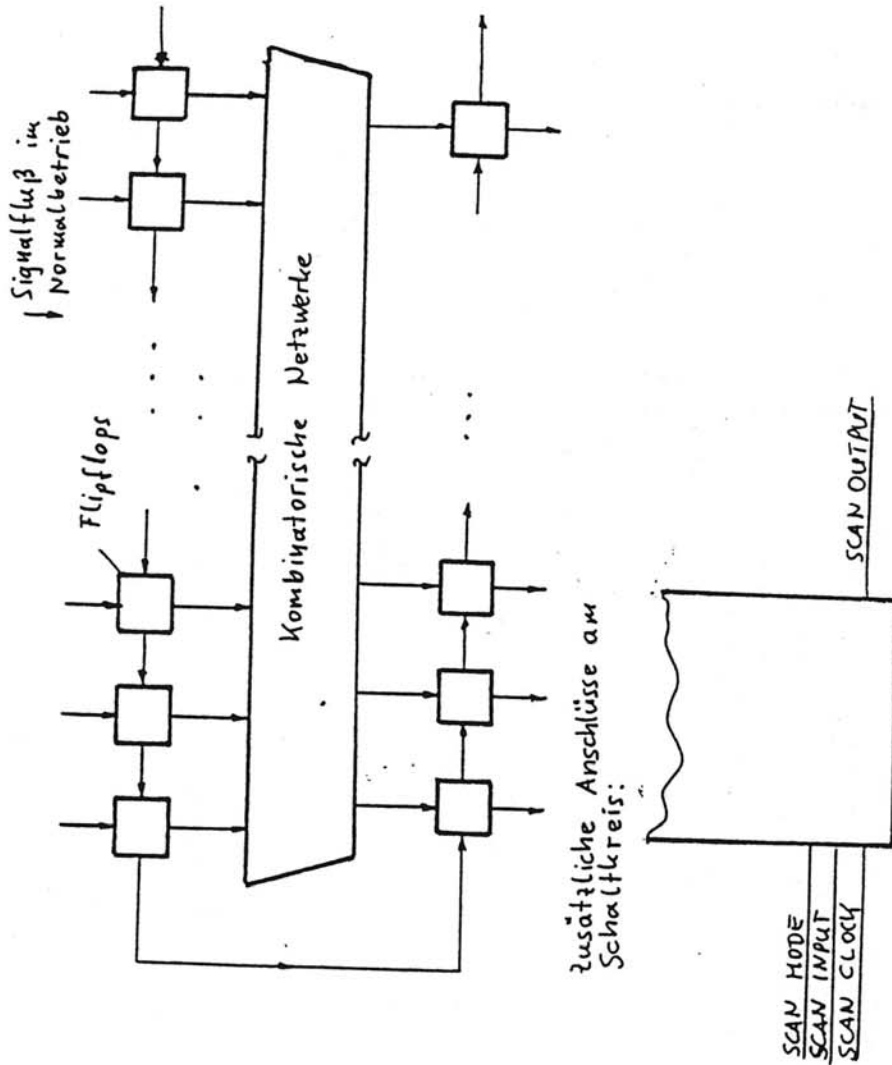


angeschlossene Einrichtungen

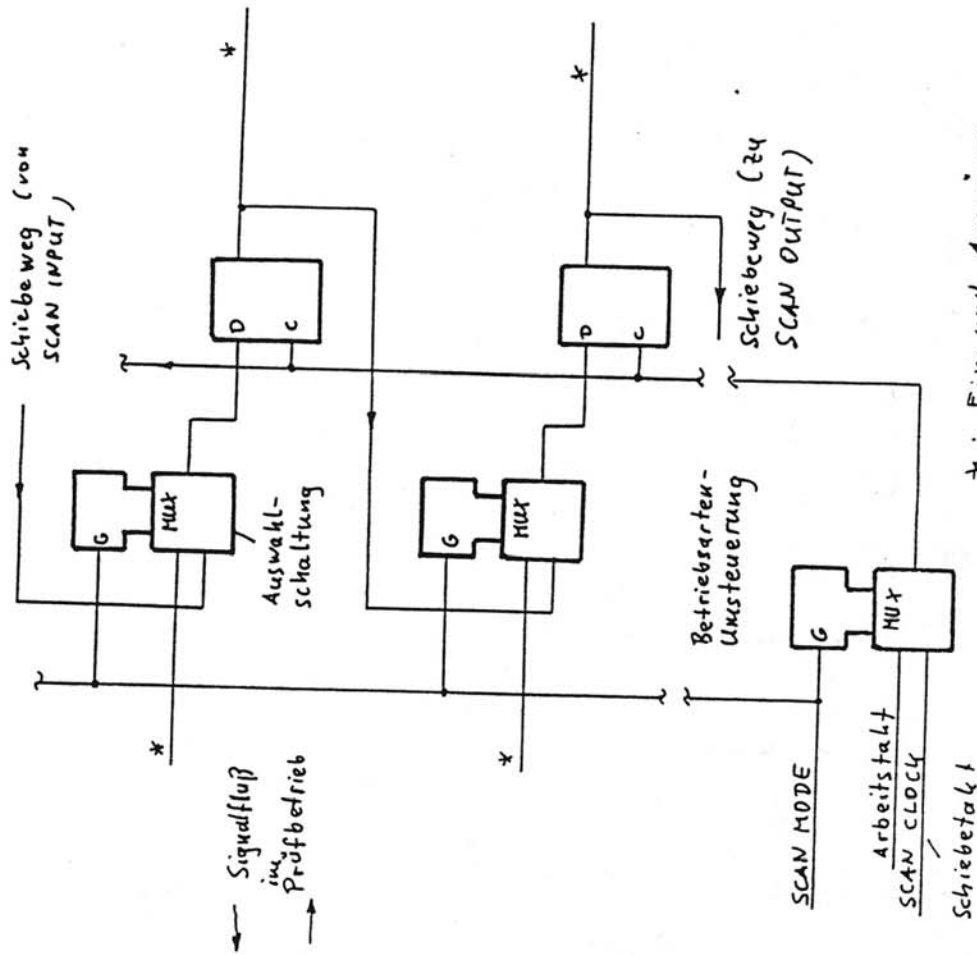


Microwire™ Bus

Prinzipieller Aufbau einer Digitalschaltung:



Einzelheit: Flipflops

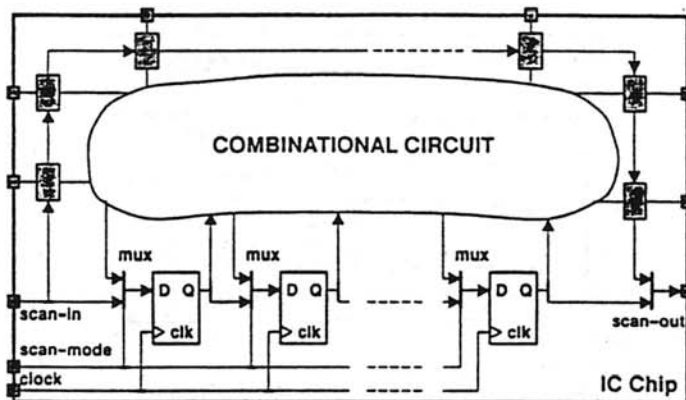


\* : Ein- und Ausgänge  
mit kombinatori-  
schen Netzwerken (Bzw.  
RAMs usw.) verbunden  
(Verschaltung für Normal-  
betrieb)

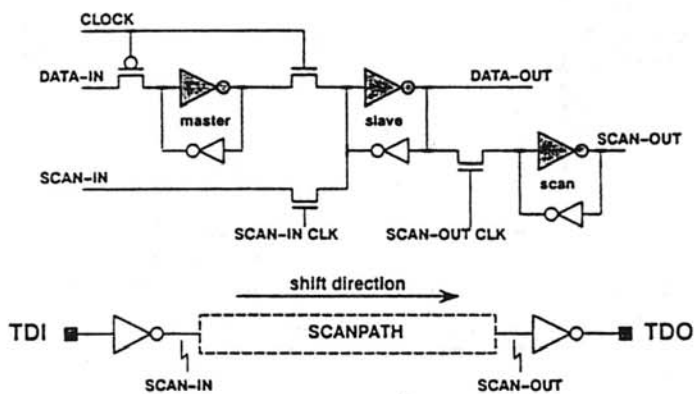
Abbildung 4-6.22

Scan-Prinzip

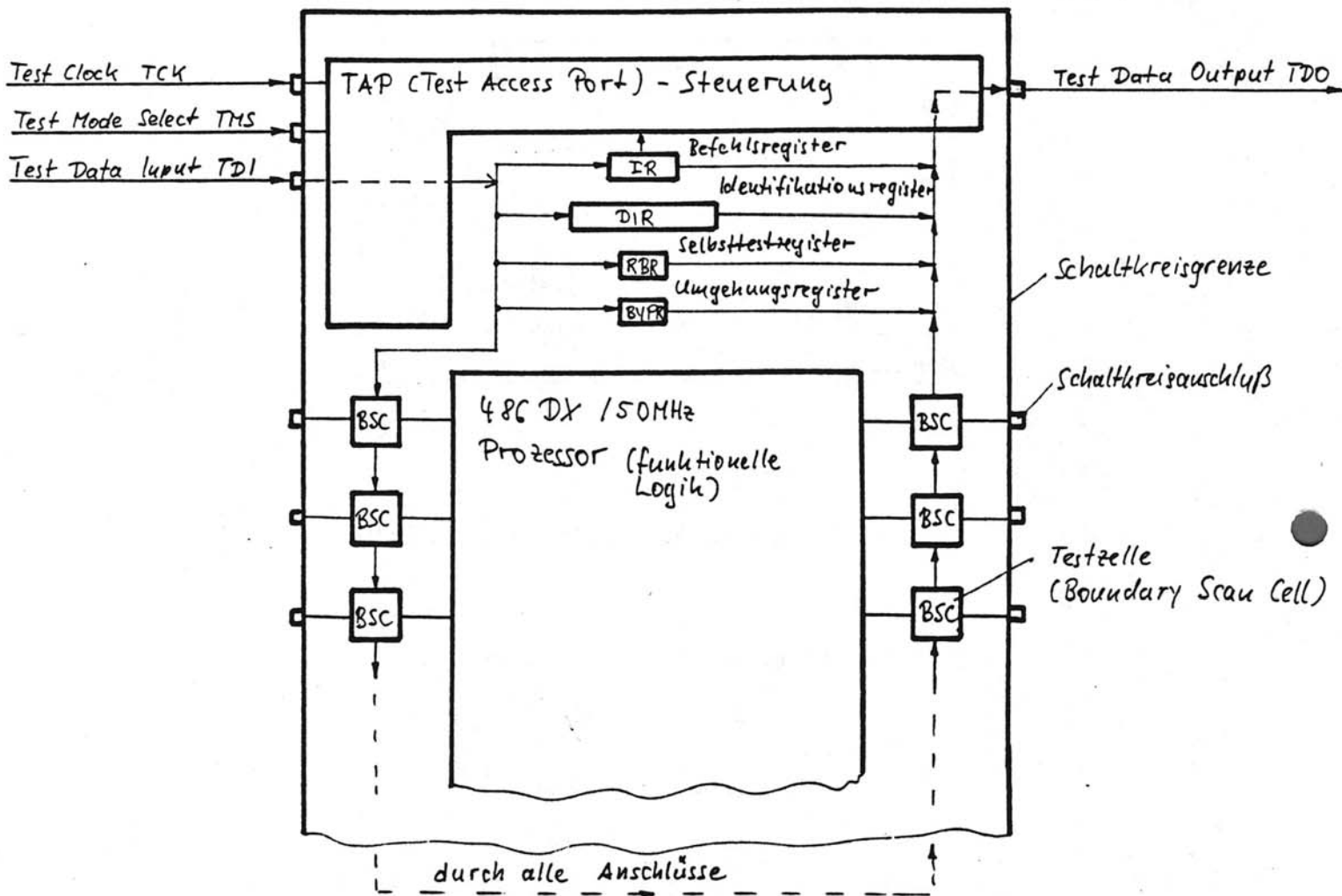
### Basic Scanpath Methodology



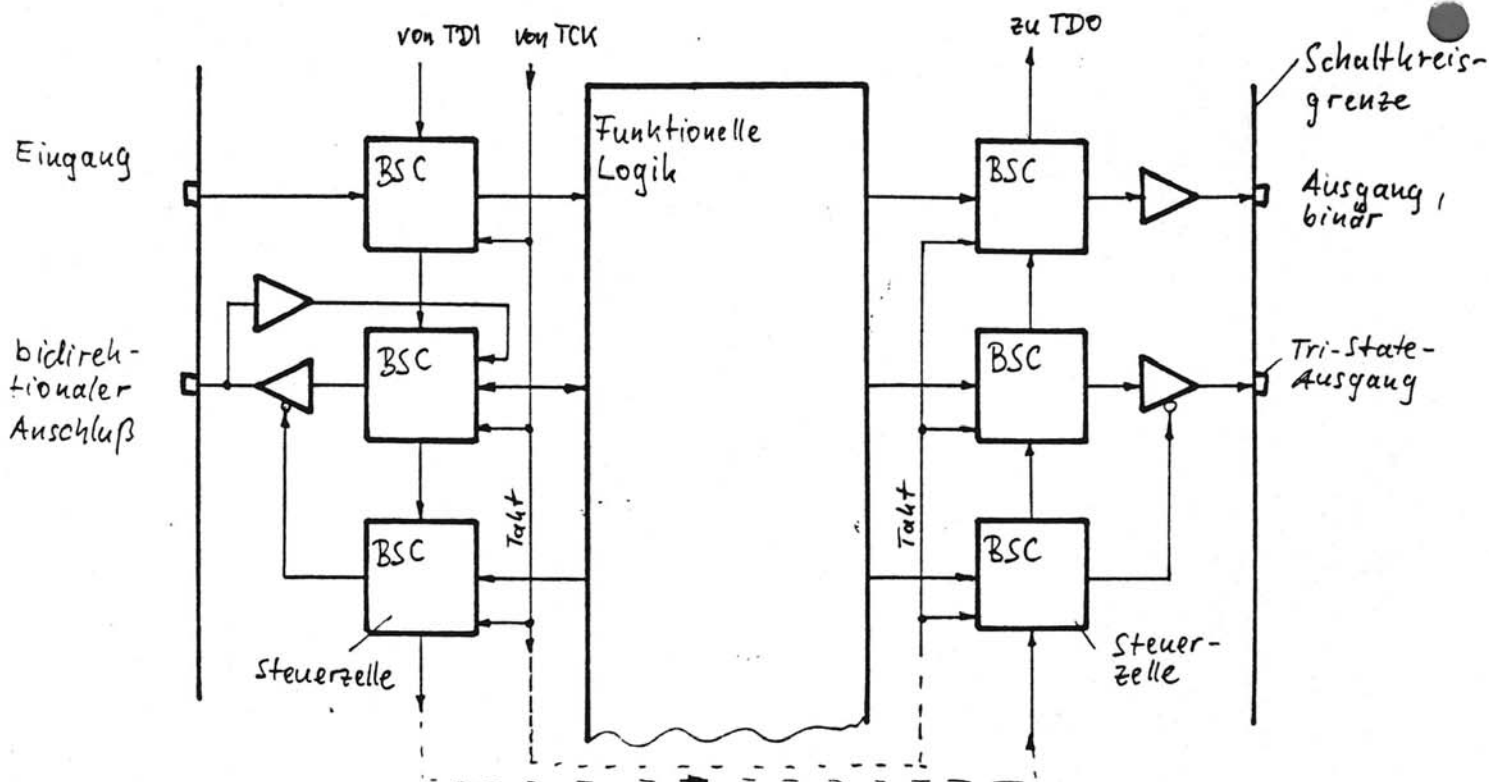
### Scannable Flip-Flop Design Example



# JTAG/IEEE 1149.1 - Testinterface (Prinzip)



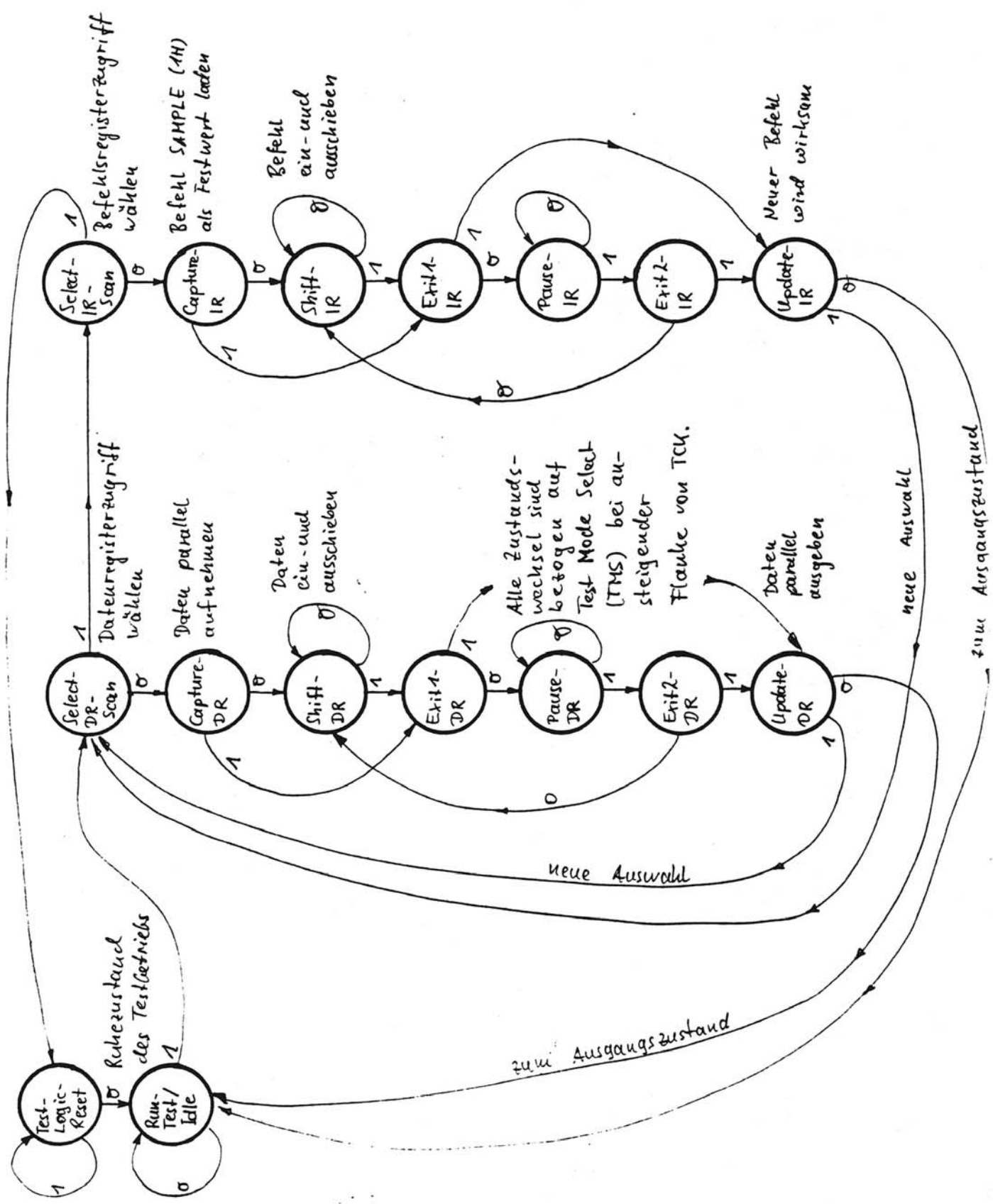
## Einzelheiten: Testzellen



Befehlsregisterzugriffe

Datenregisterzugriffe

Ausgangszustand



Auswahlzustand

parallele Informationsaufnahme

Schiebezustand

Übergangszustand

Pausezustand

Zwischenzustand

Endzustand

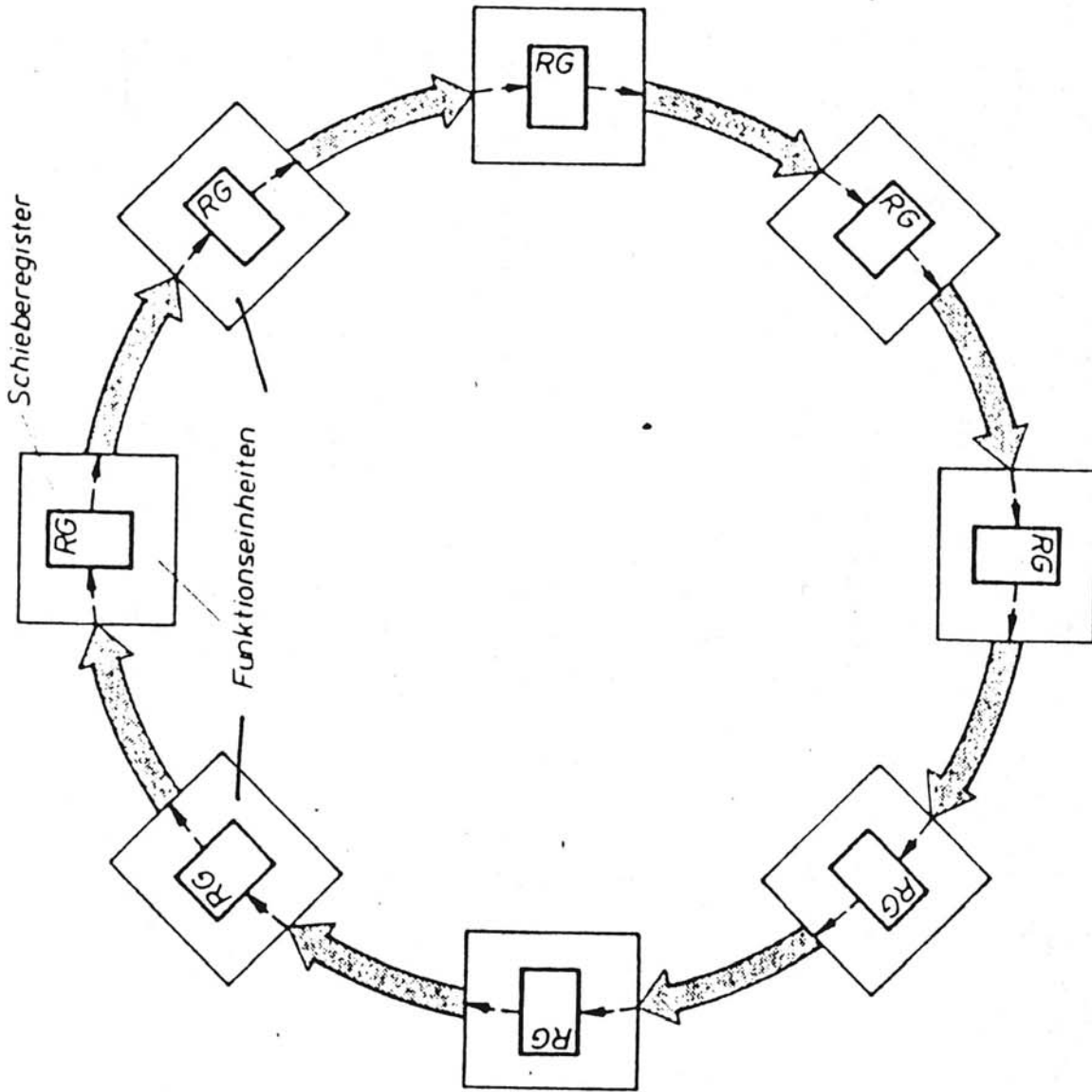


Abbildung 4-5.9 Schieberegister-Ringbus

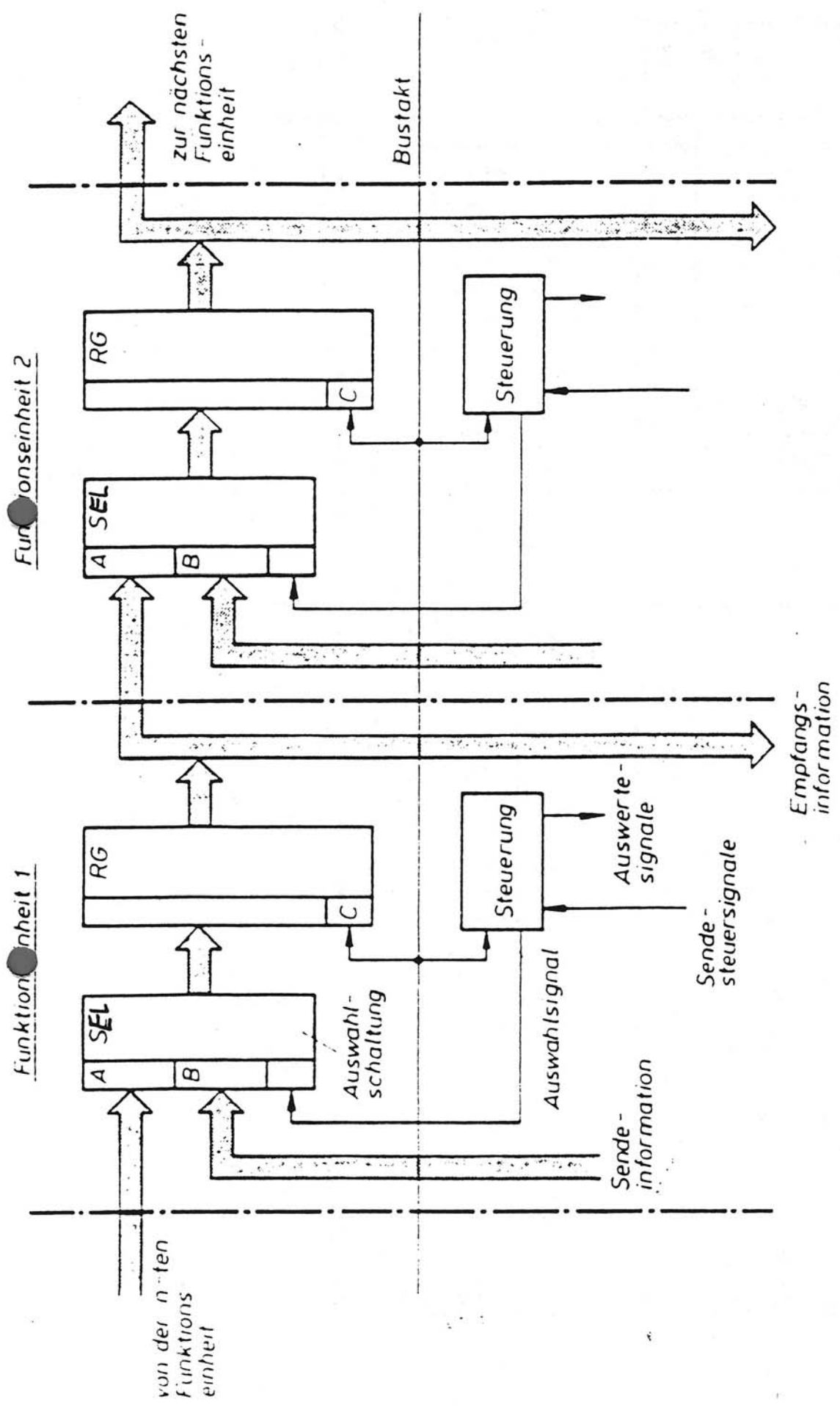
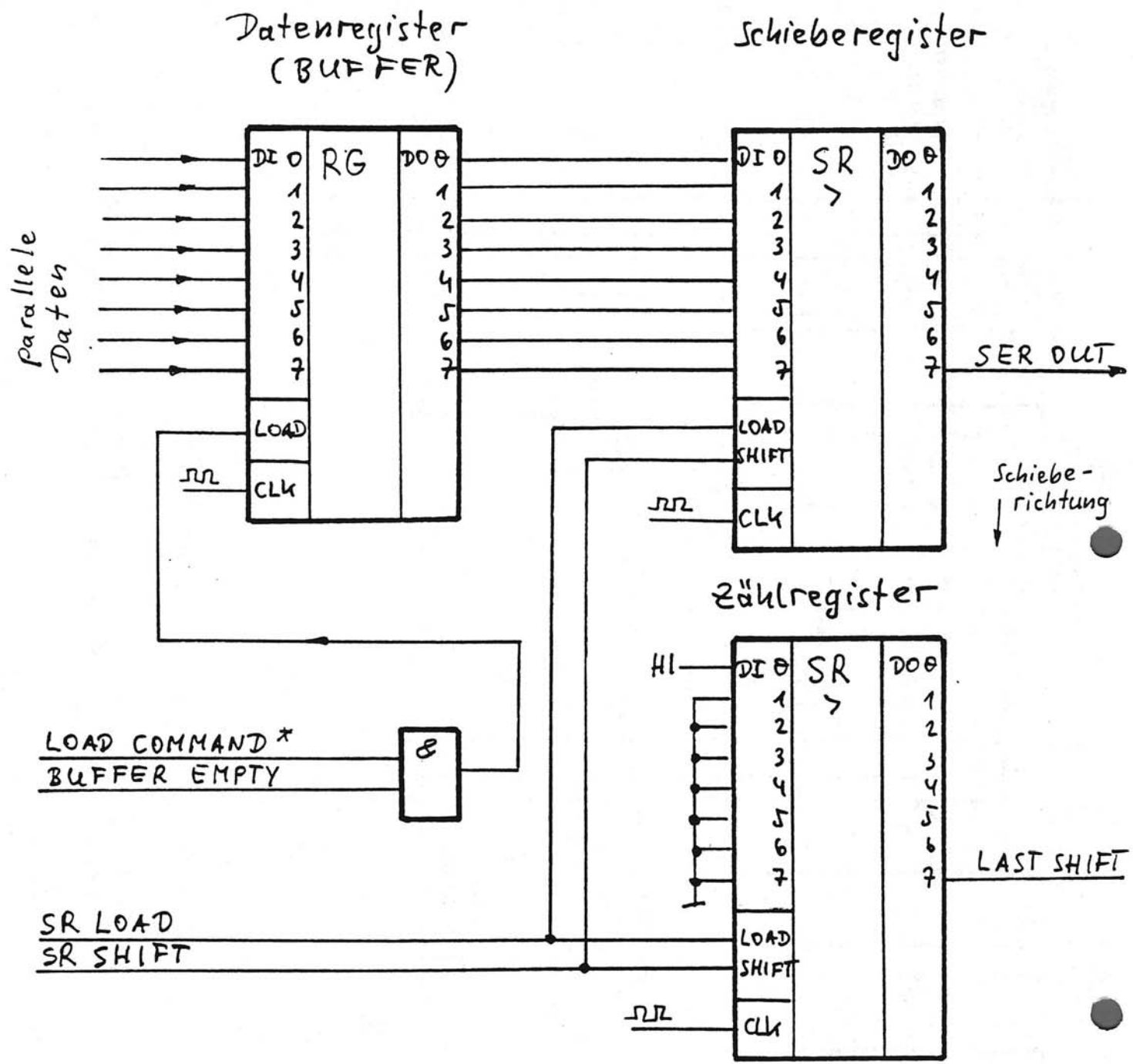


Abbildung 4-5.10 Anschaltung einer Funktionseinheit



\* z. B. vom Prozessor (OUT-Befehl)

# Serializier I: Datenteil



Bibelle & Co. ziehen

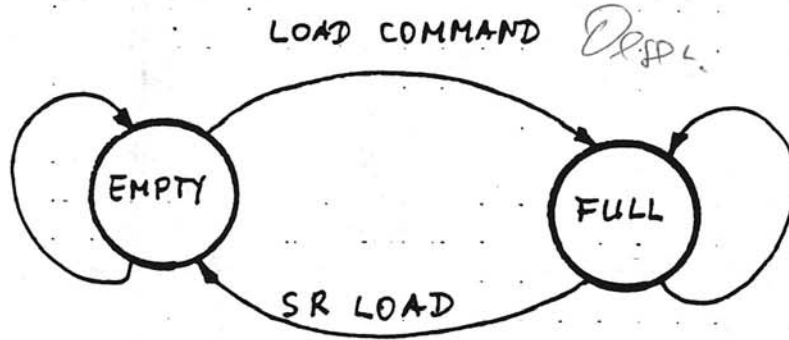
### a) Pufferregister

Serialis. d. d. Schreibe

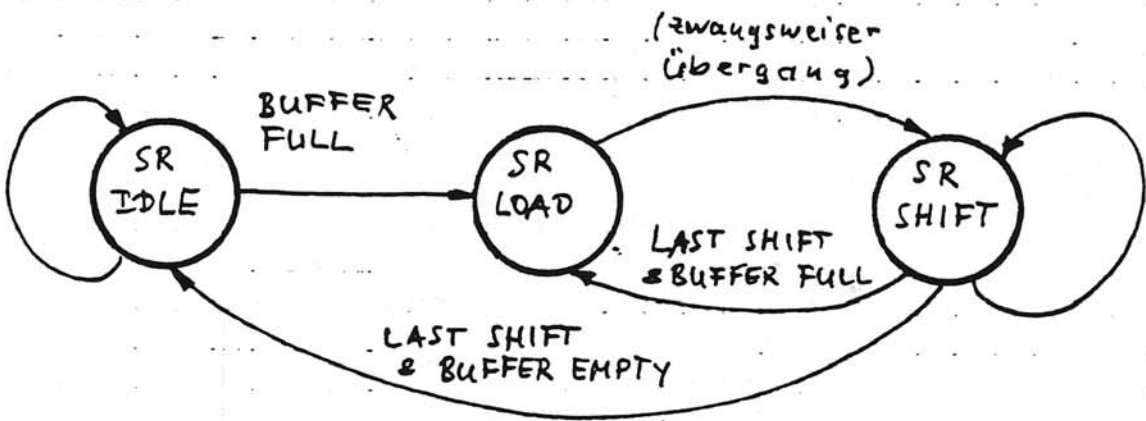
blöcke

Schleifen

Verketten

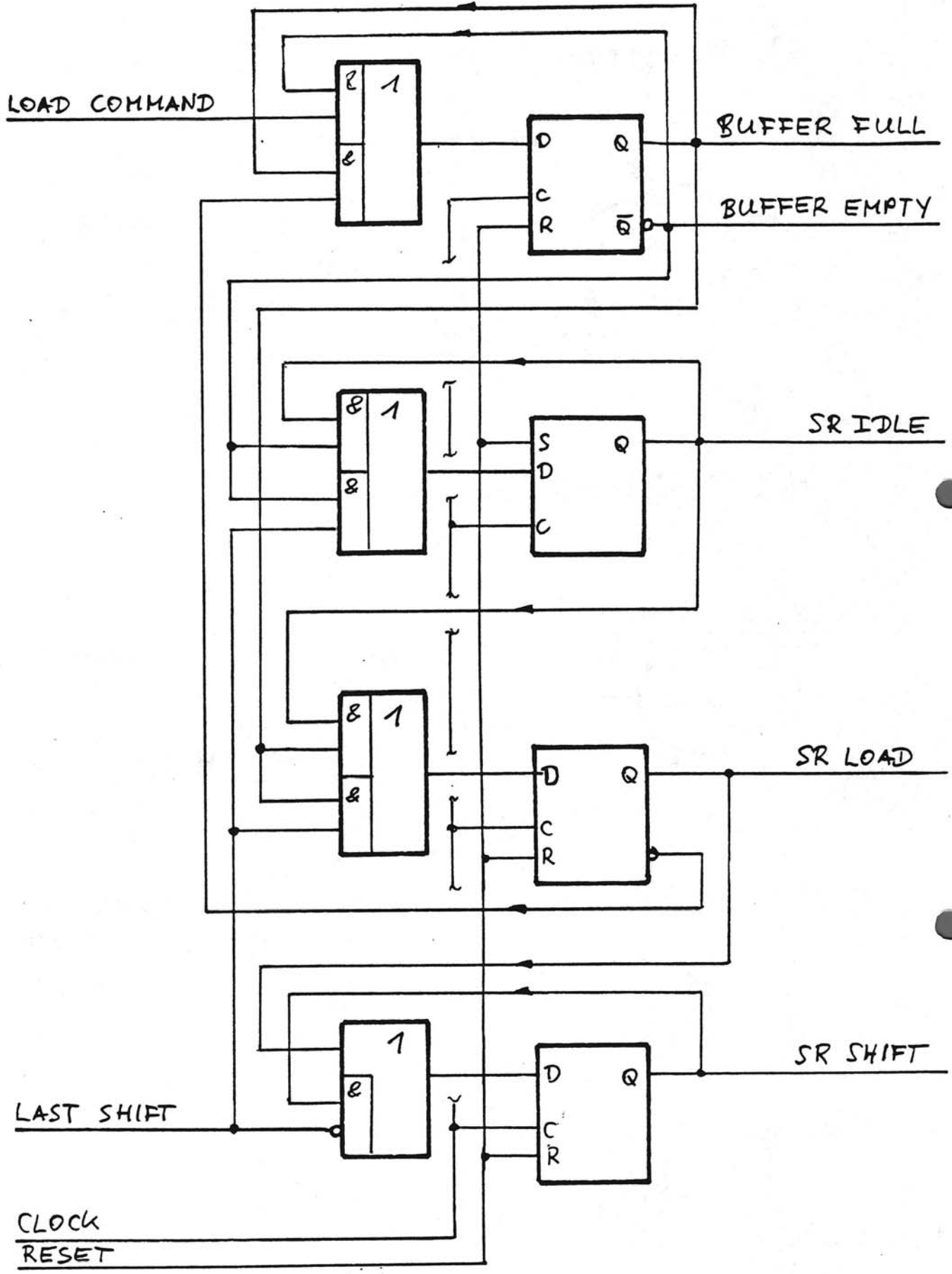


### b) Serialisierung



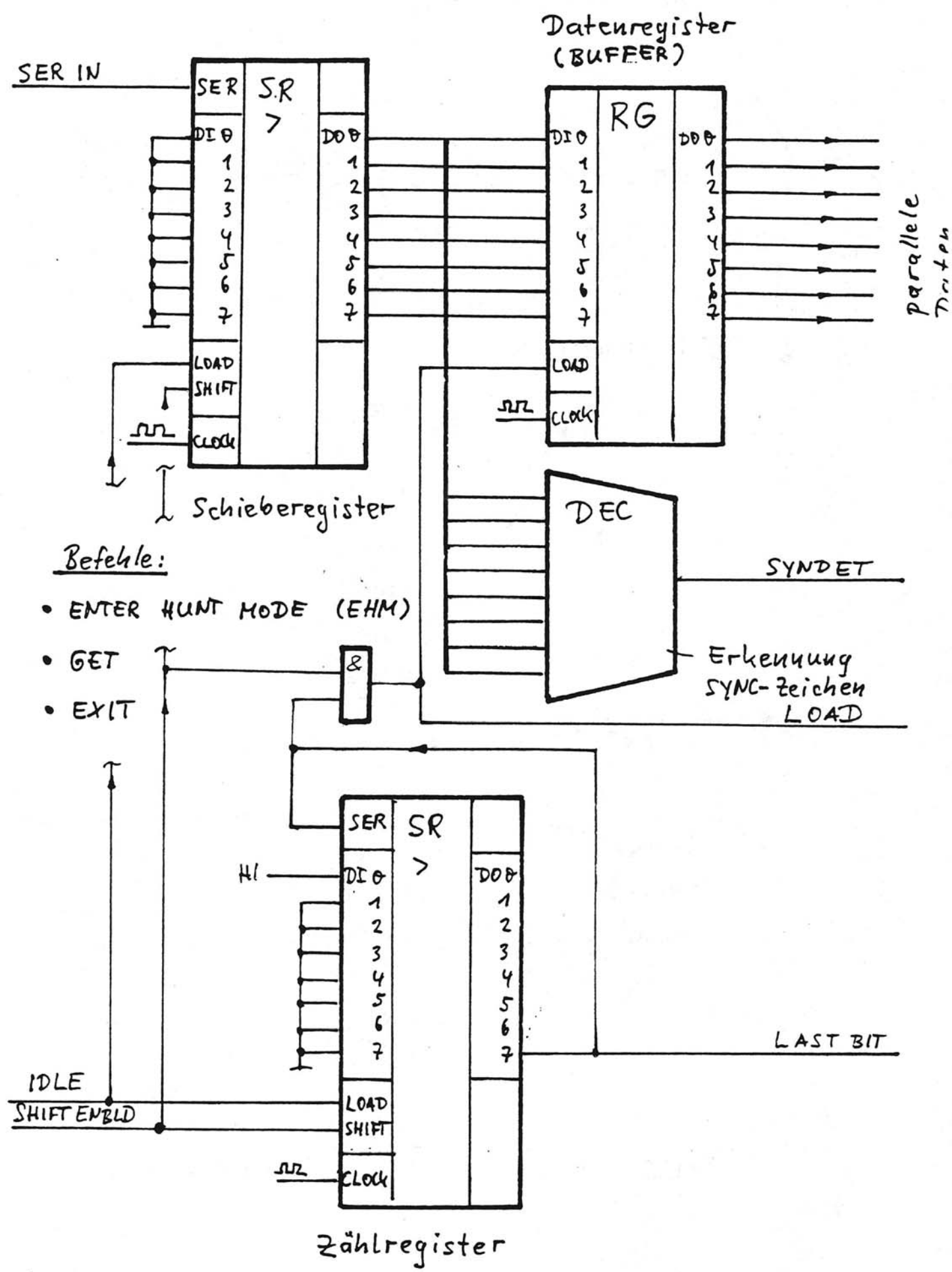
anstelle der Rückkehr zu IDLE ggf. Fehlerzustand (Overrun, Data Lost)

## Serializier II: Zustände

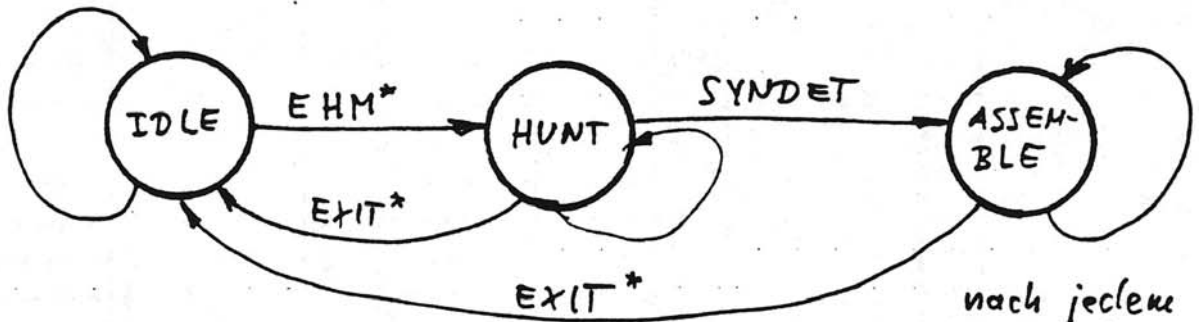


Serializer III: Steuerung

# Deserialiser I, Datenteil



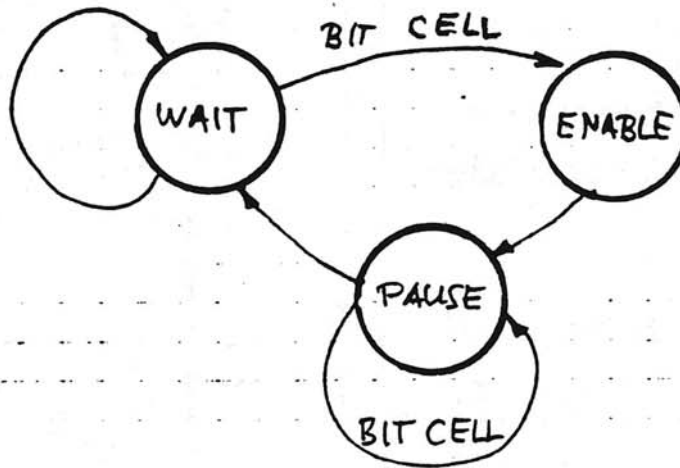
### a) Schiebeablauf



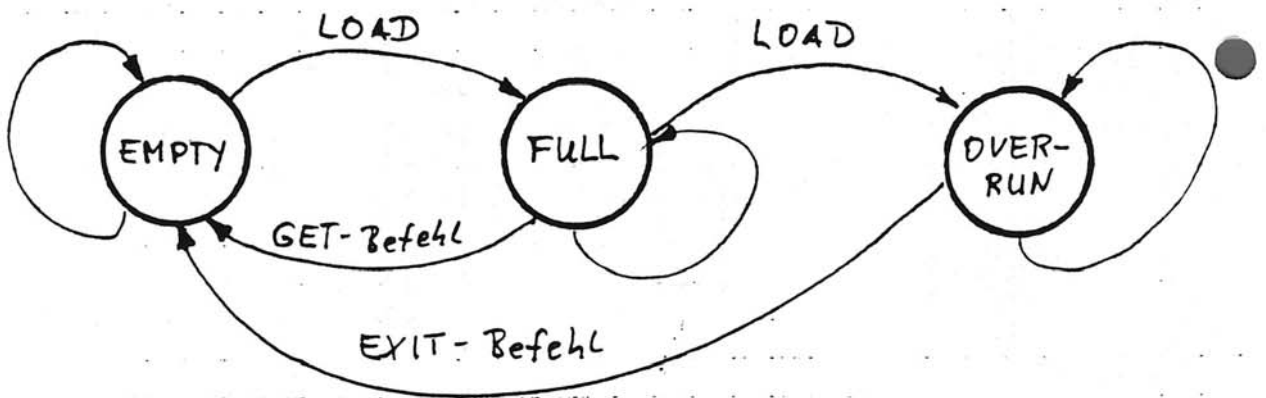
nach jedem P. Bit: LOAD

\*: Befehle

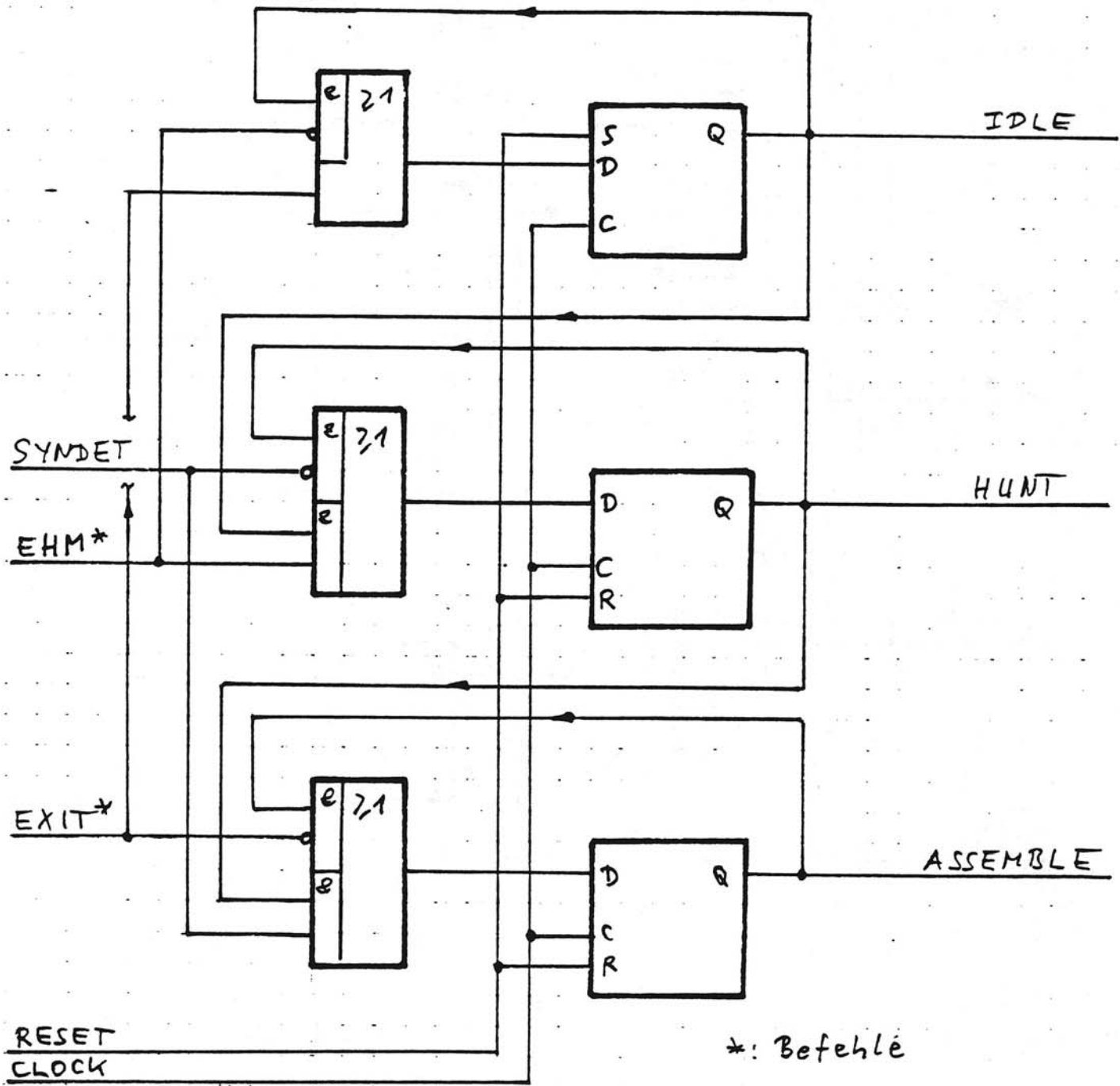
### b) Schieberlaubnis



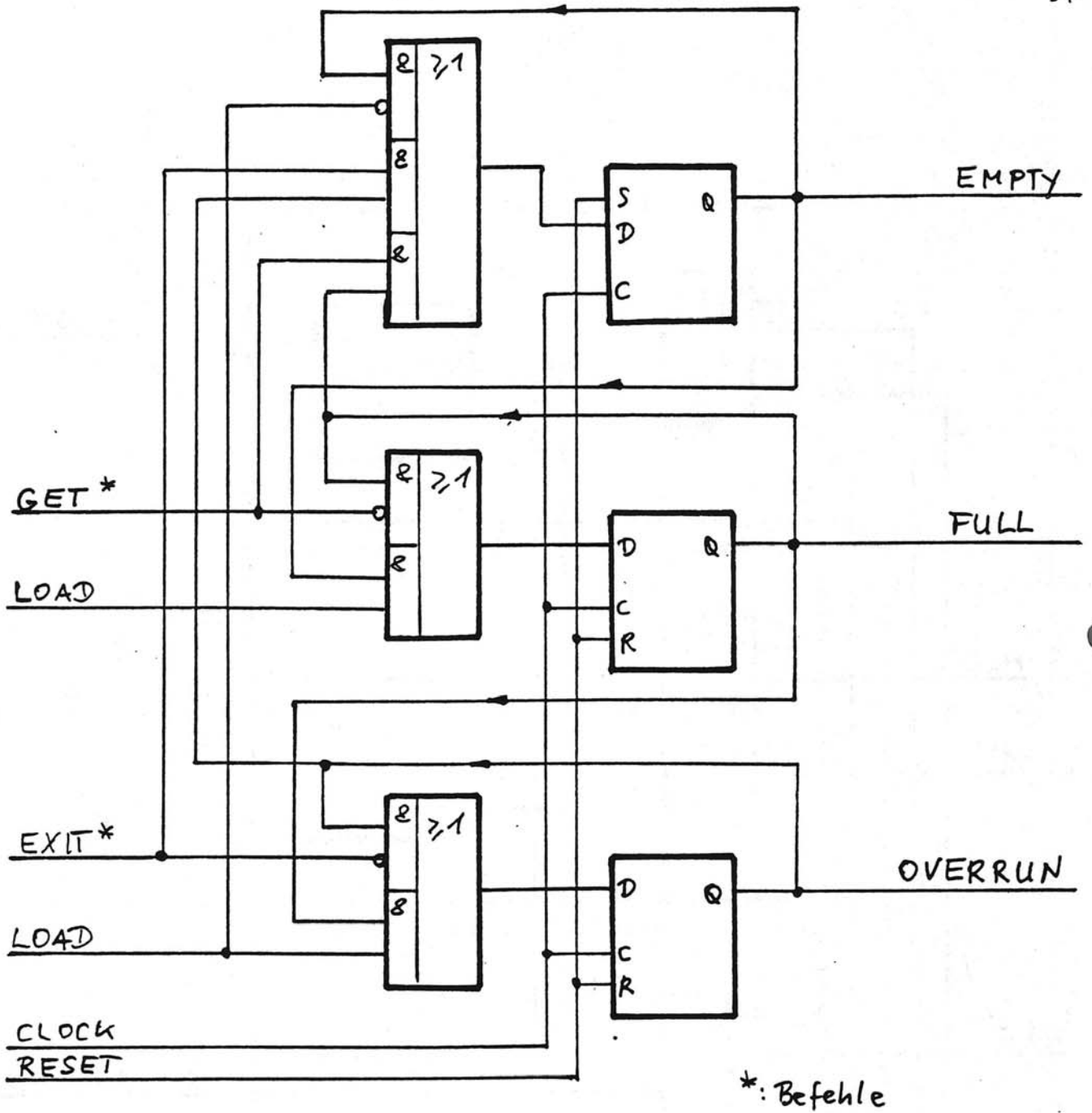
### c) Puffersteuerung



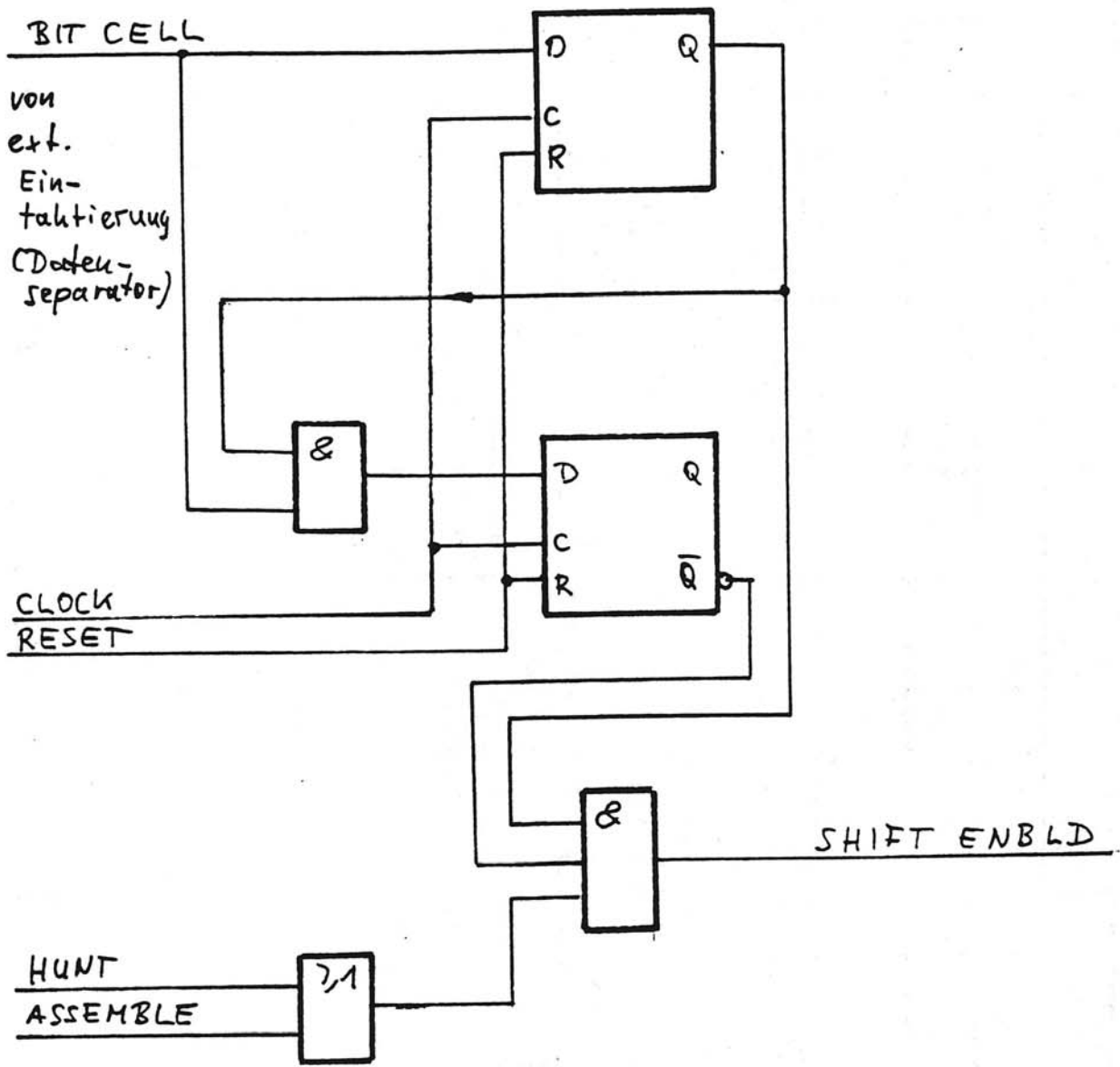
## Deserializer II: Zustände



Deserializer III: Steuerung Schiebe-  
ablauf



Deserializier IV: Steuerung Pufferregister



Deserializier  $\bar{V}$ : Bit Cell Single Shot  
(Schiebe/Lade-Erlaubnis)

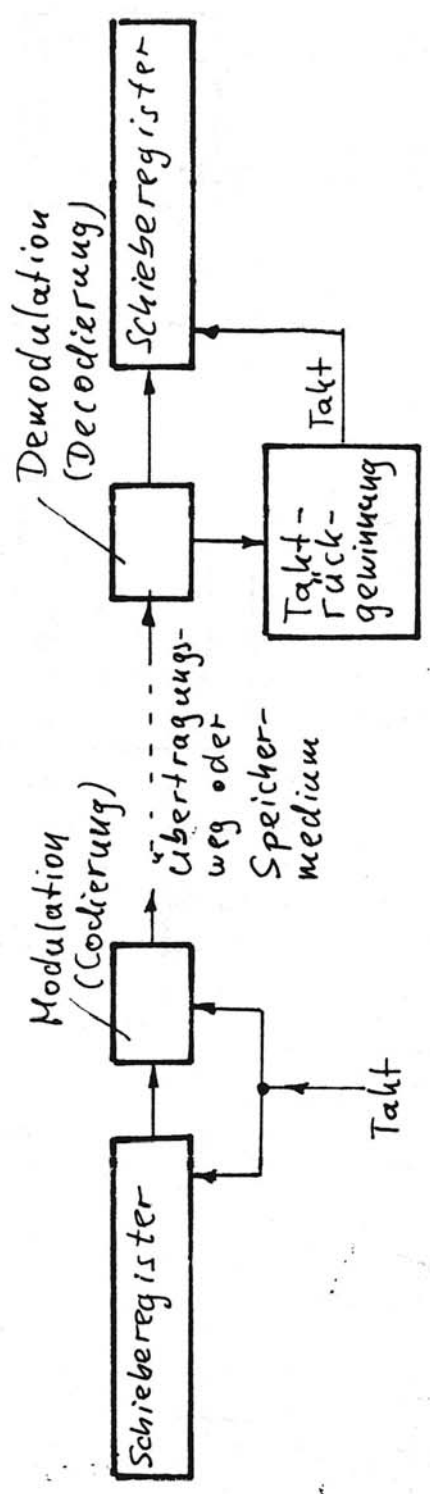
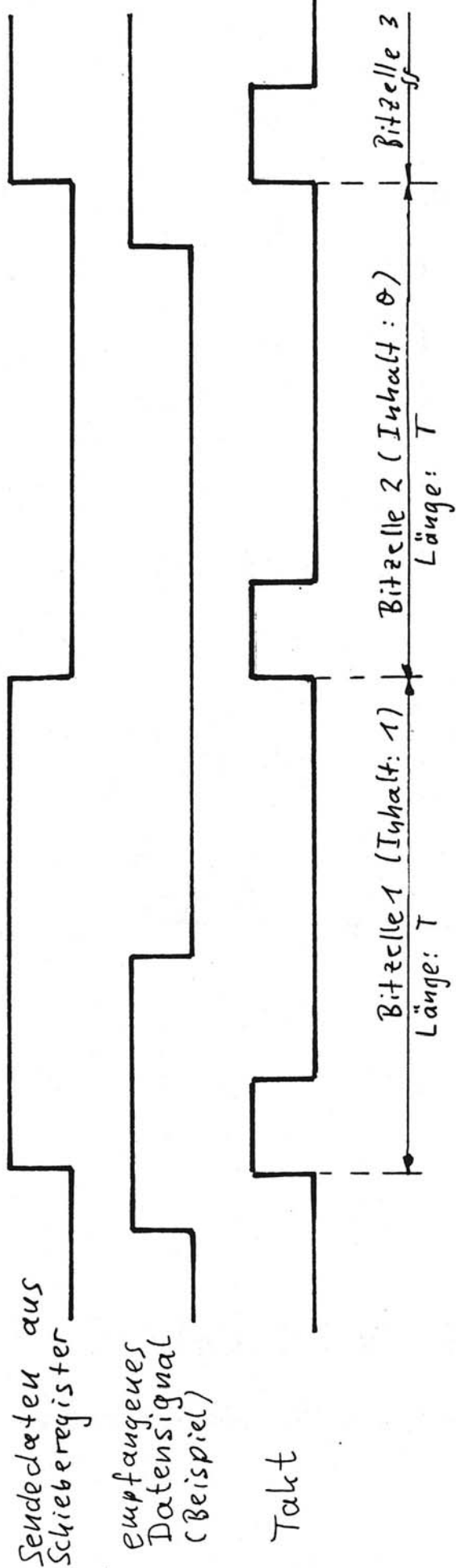
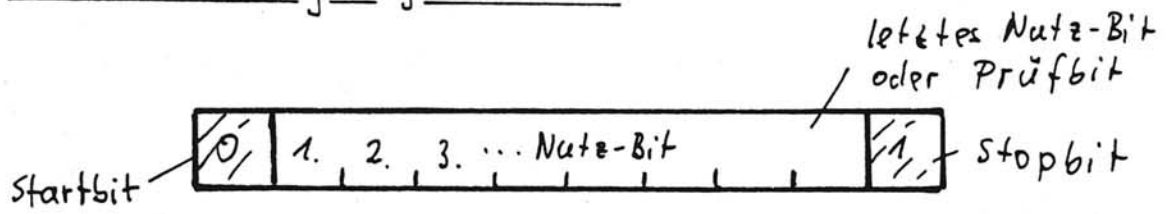


Abbildung B4-6.1 Die Bitzelle beim Senden und Empfangen

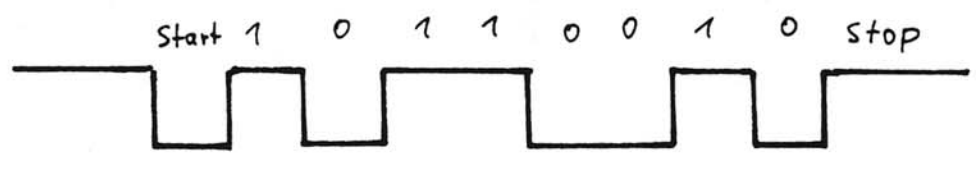


# a) Übertragungsformat

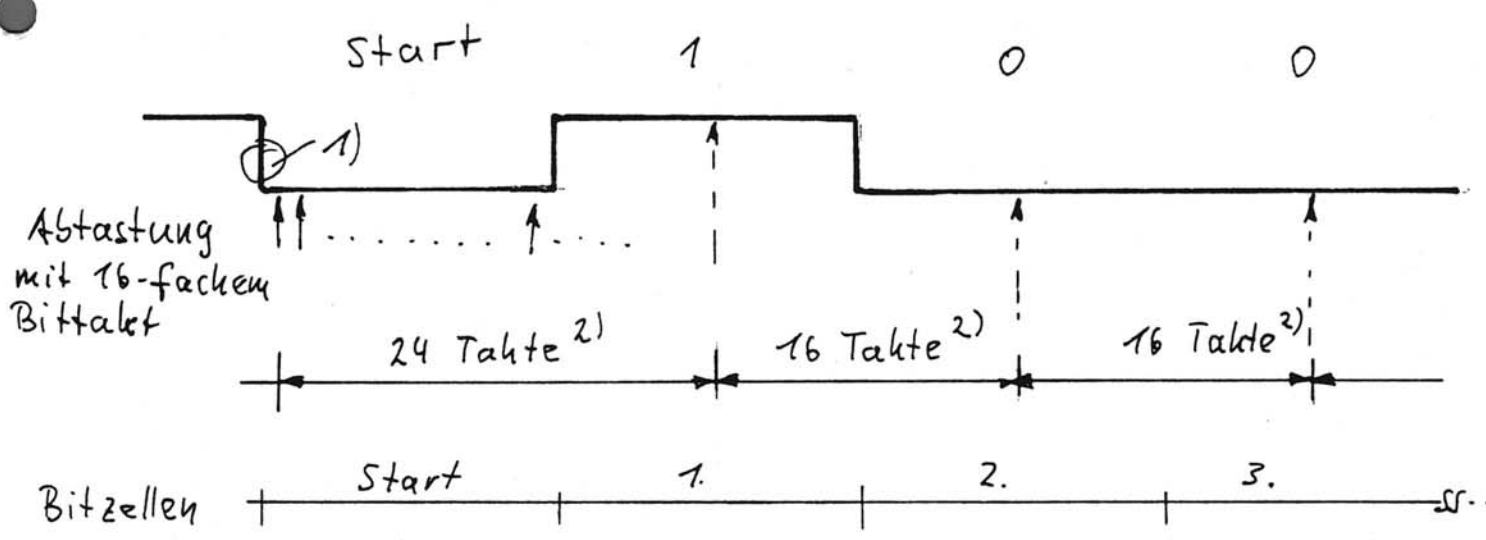


 : feste Wert

# b) Beispiel für Signalverlauf



# c) Einzelheit: Abtastung des empfangenen Signals



- 1): Die 1-0-Flanke eröffnet die Taktzählung
- 2): Es werden jeweils soviele Abtast-Takte abgezählt; dann wird der Signal-Wert übernommen (etwa in der Mitte der Bitzelle)

# d) Zustandsdiagramm (Empfänger)

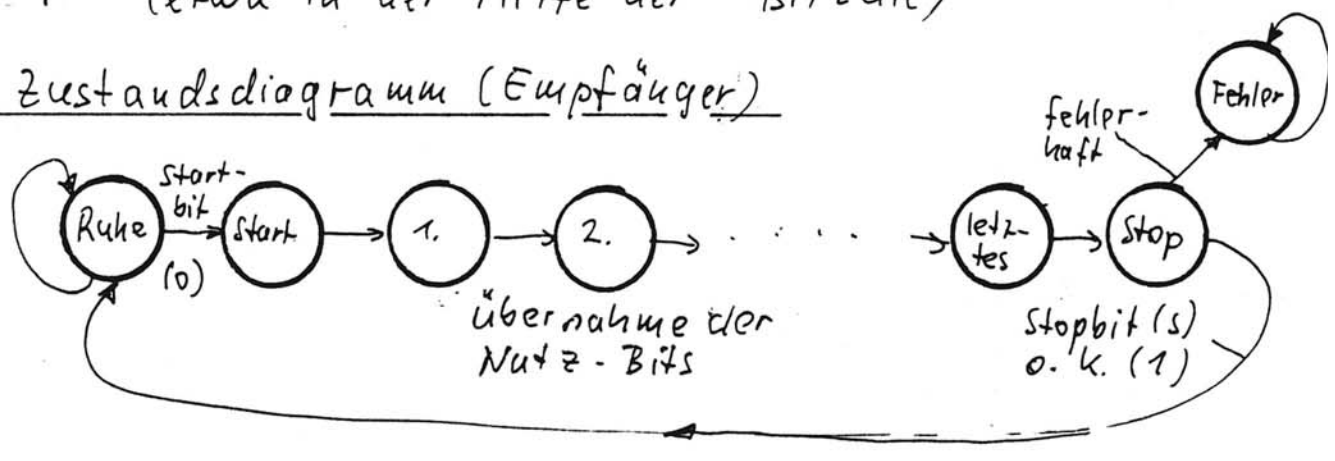
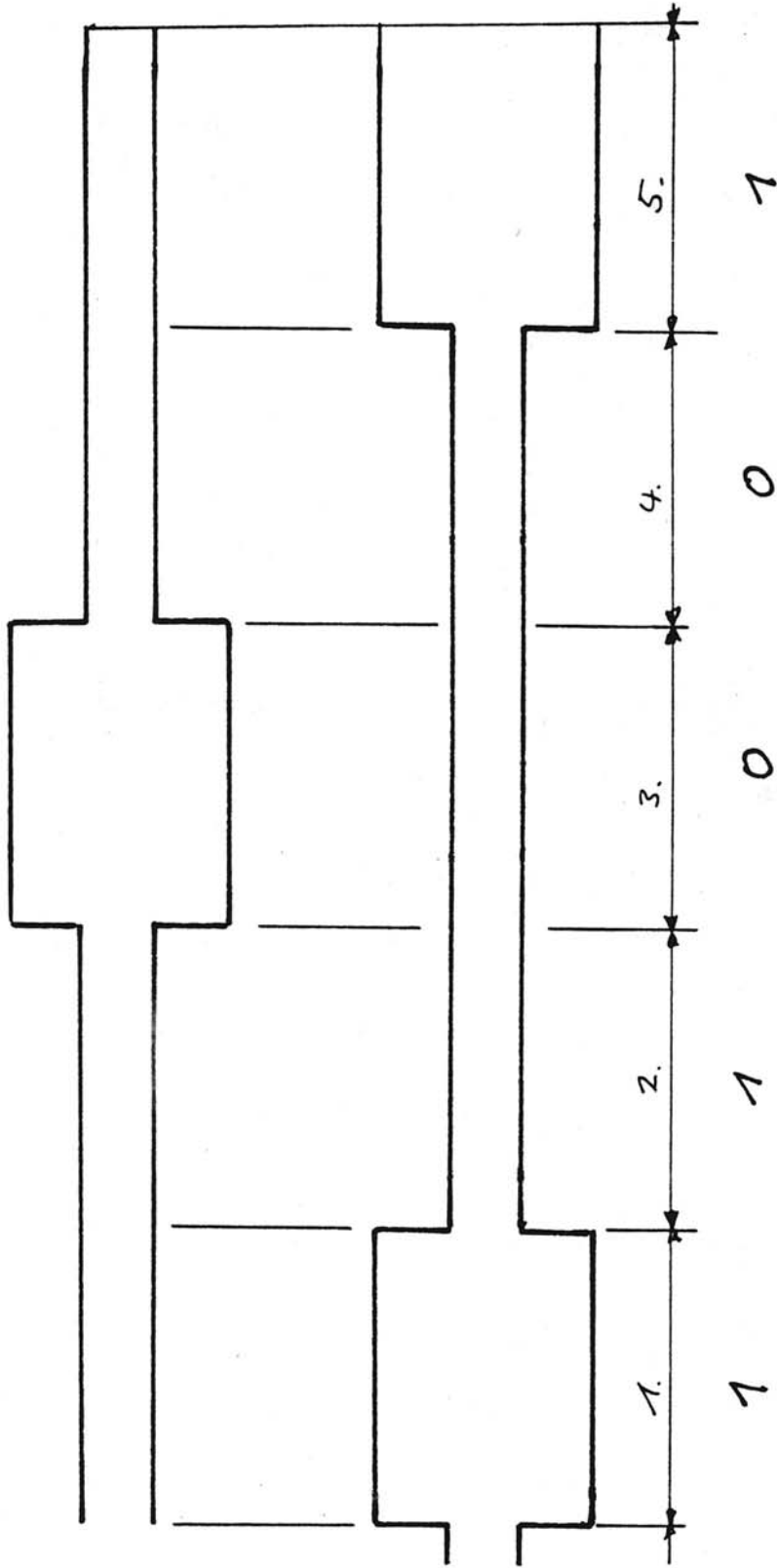


Abbildung B4-6.4

Signale  
Invert on Zero

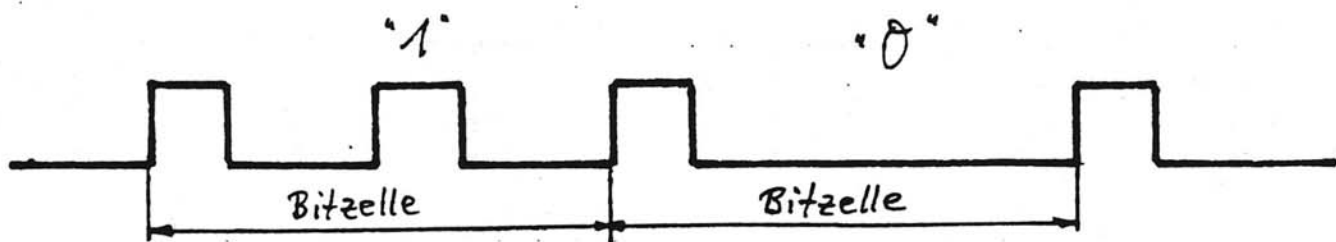


Signale  
Invert on One

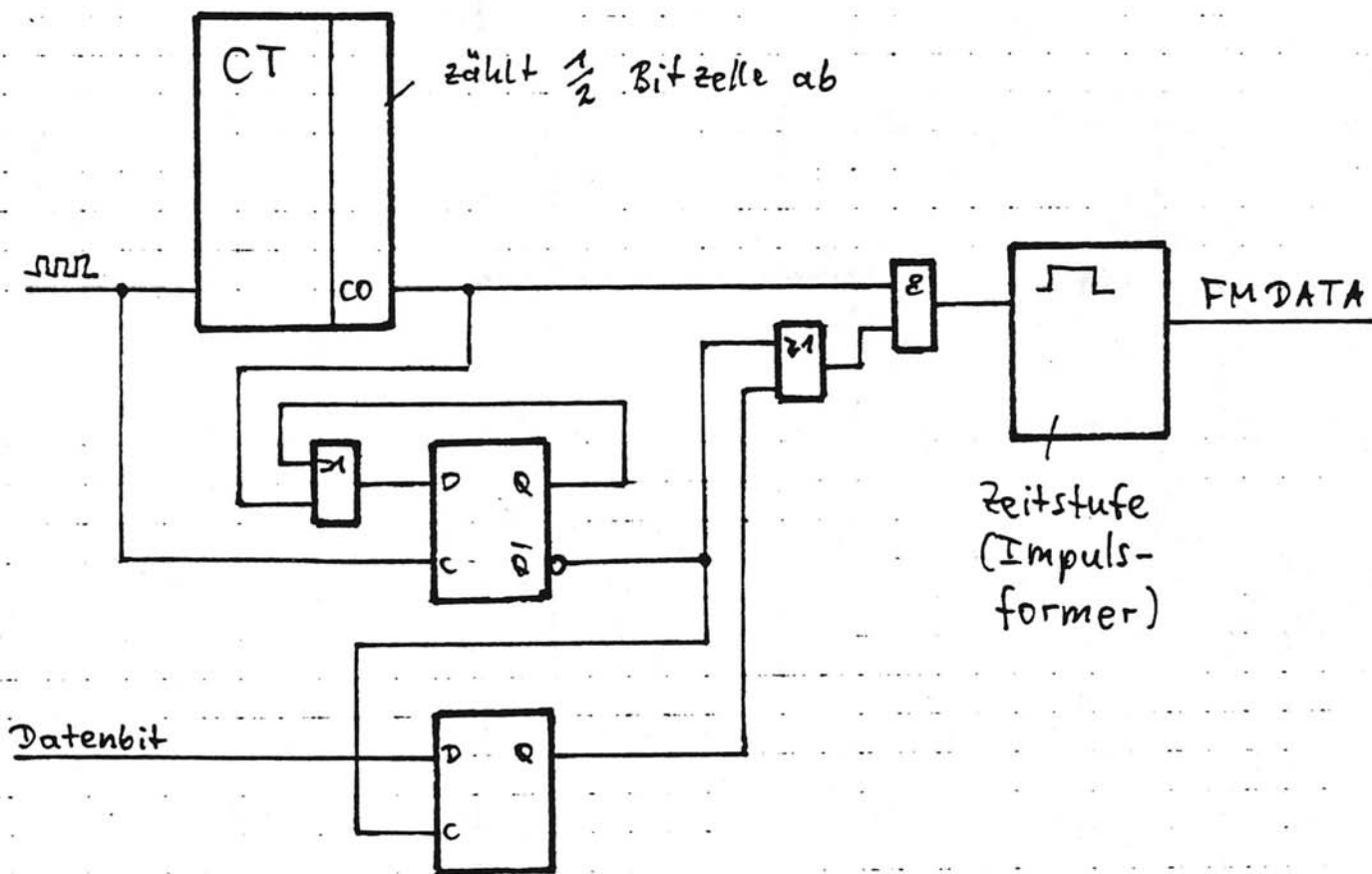
Bitzellen

Abbildung B4-6.5 Signaldarstellungen Non Return to Zero  
and Inversion (NRZI)

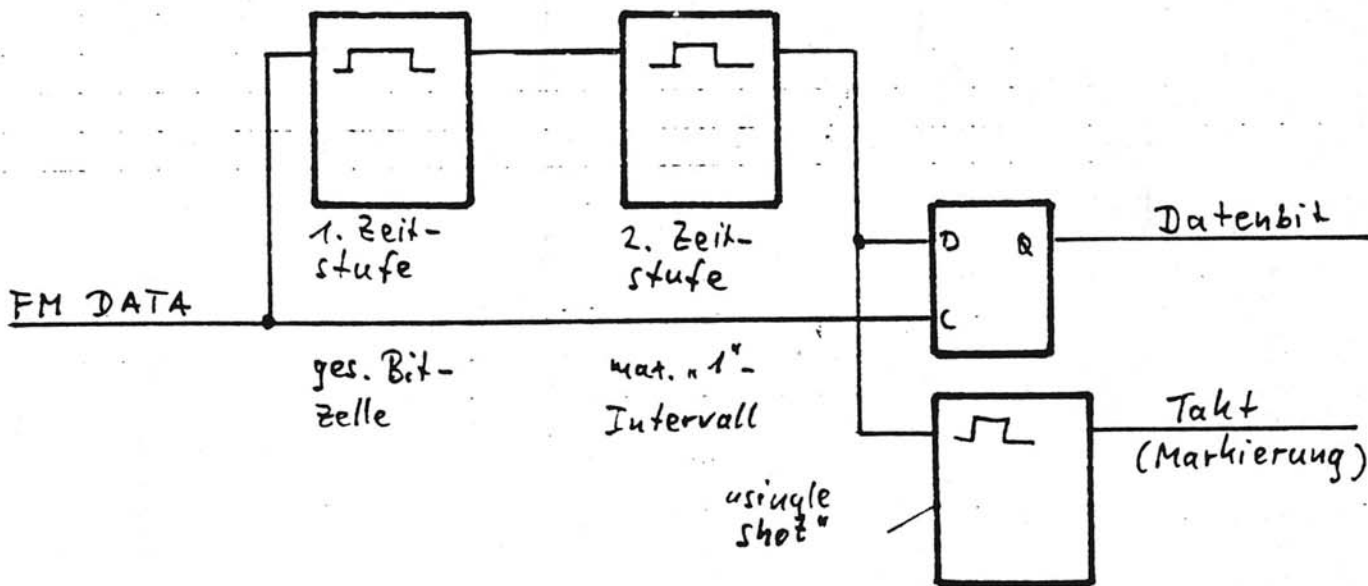
# FM: Prinzip



## FM: Modulator (Beispiel)



## FM: Demodulator (Prinzip)



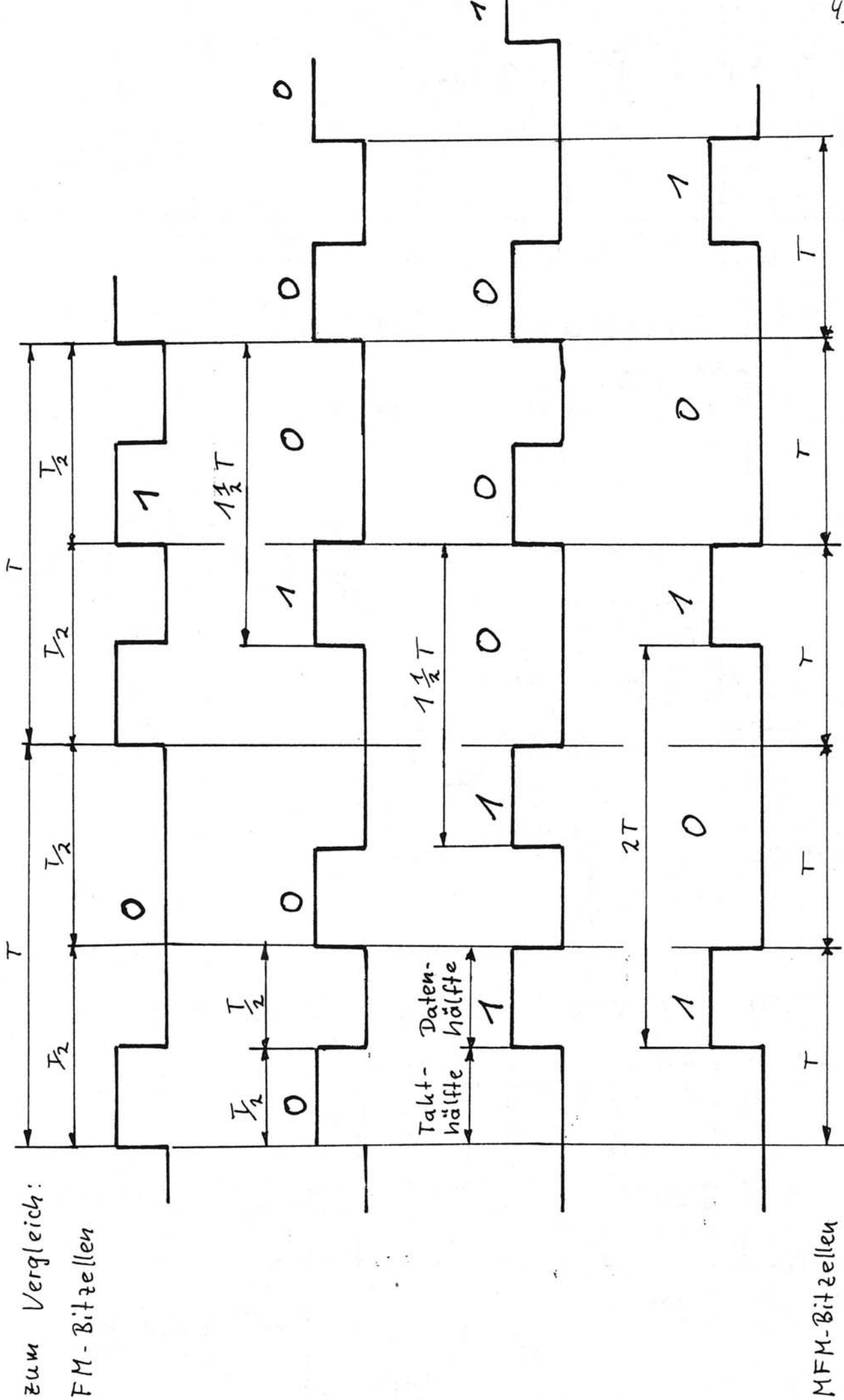
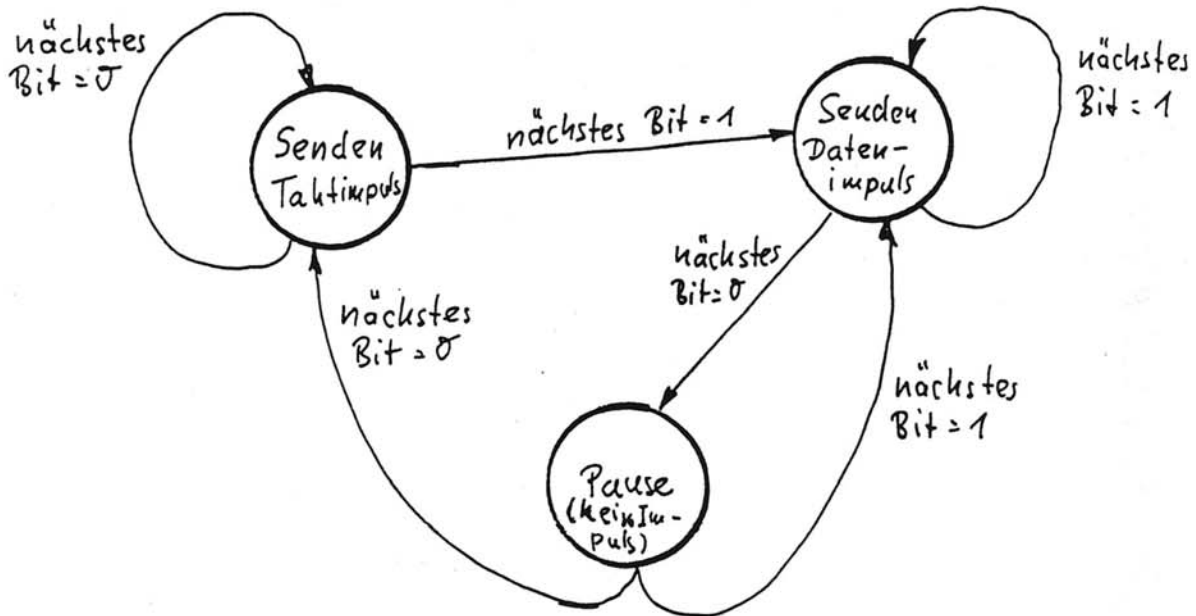


Abbildung B4-6.8 Modifizierte Frequenzmodulation (MFM)



(Zustandswechsel: am Ende der Bitzelle.)

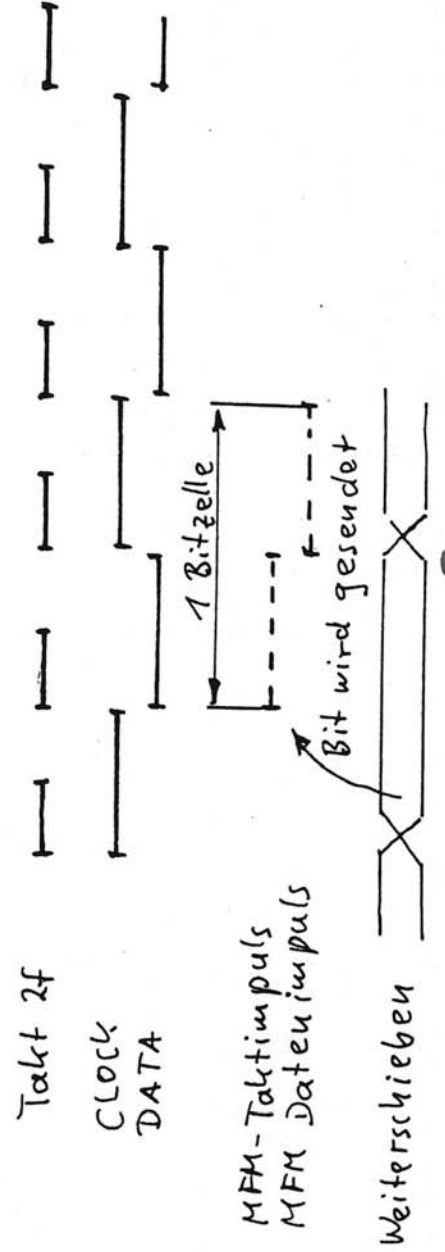
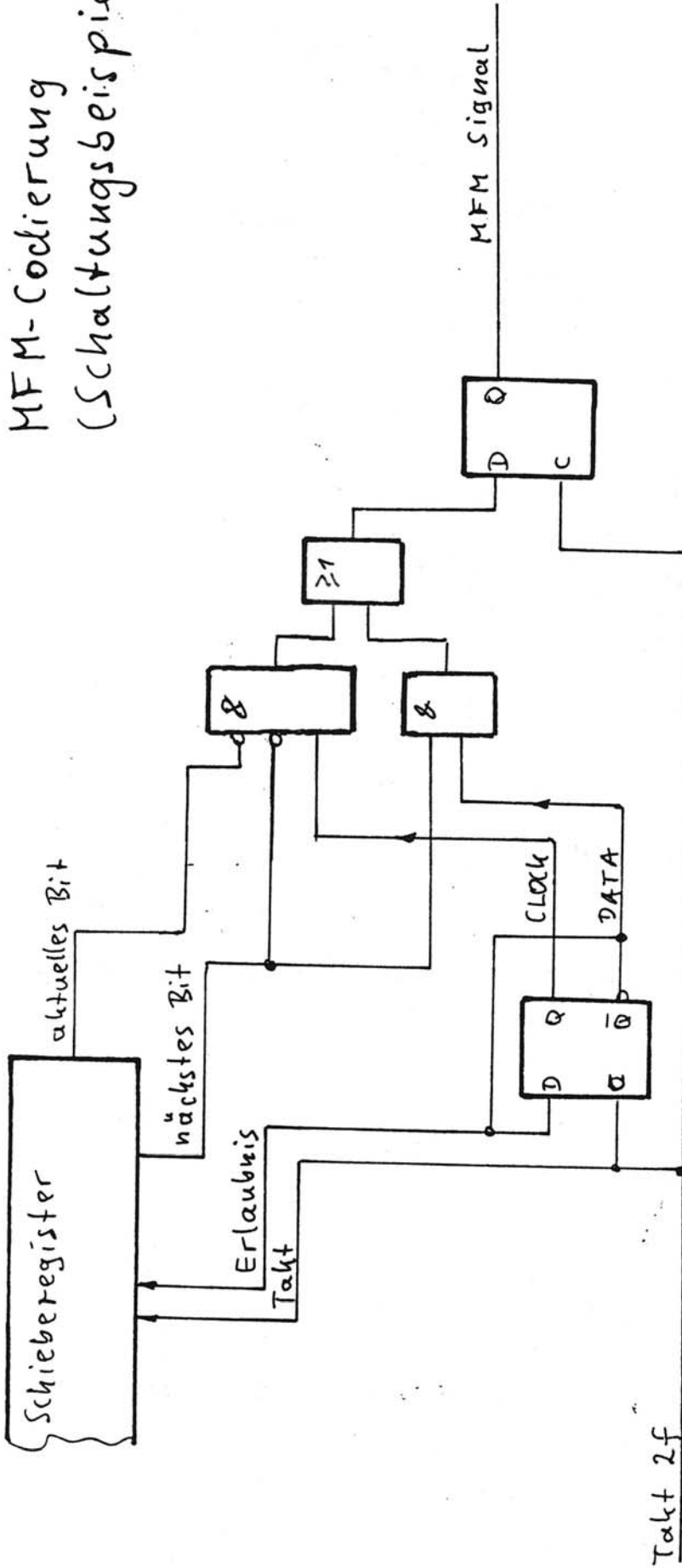
aktuelles Bit	nächstes Bit	zu senden
0	0	Taktimpuls
0	1	Datenimpuls
1	0	- (Pause)
1	1	Datenimpuls

- Es ist jeweils dargestellt, was in der nächsten Bitzelle zu senden ist.

Abbildung B4-6.9 MFM-Codierung als Zustandsgraph und Automatentabelle

# Abbildung BY-6.10

## MFM-Codierung (Schaltungsbeispiel)



### Codierung RLL 2,7

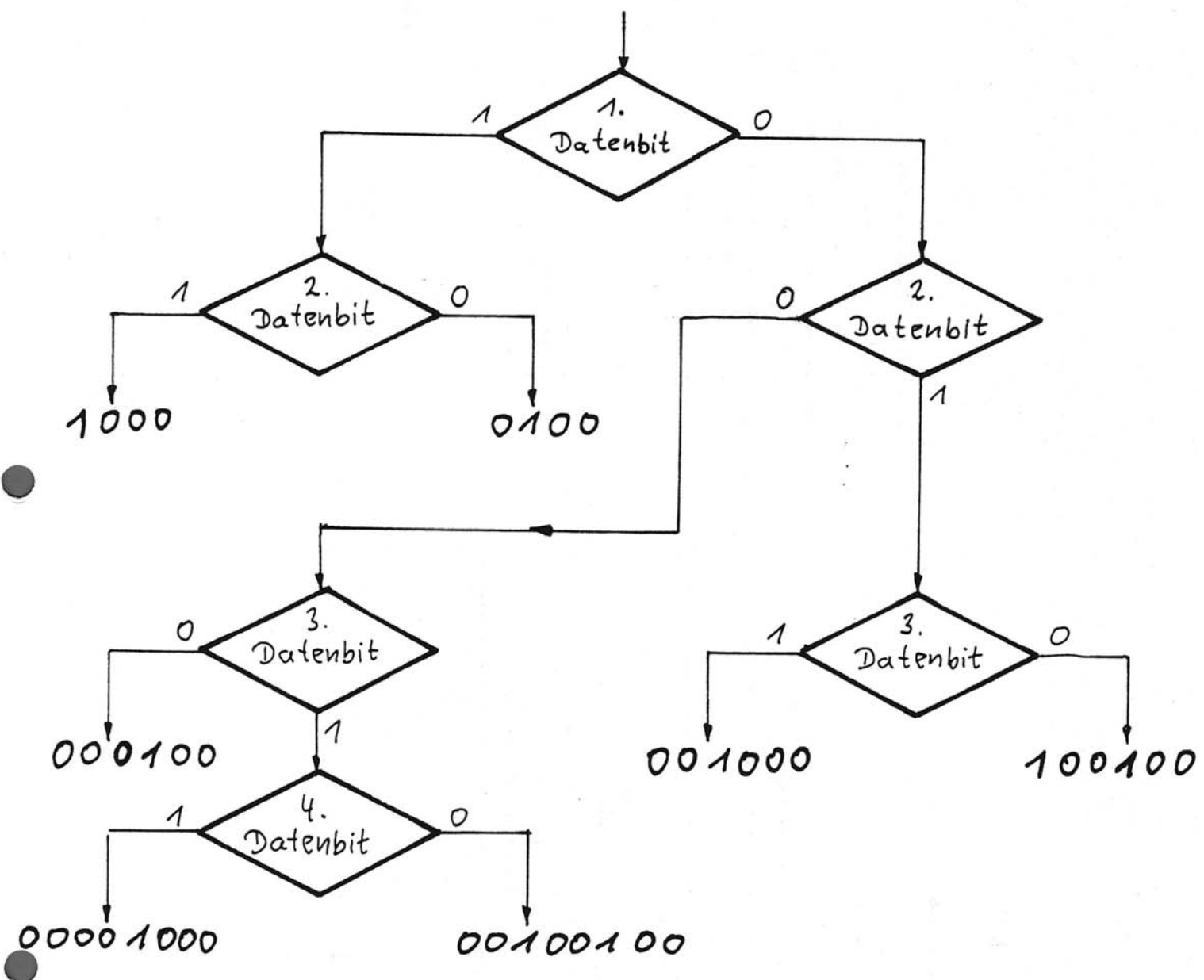


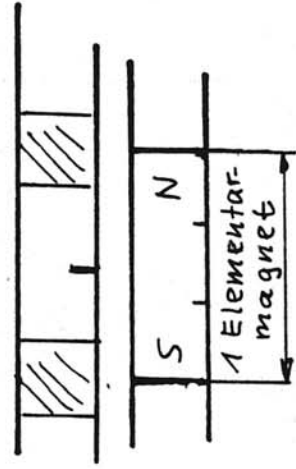
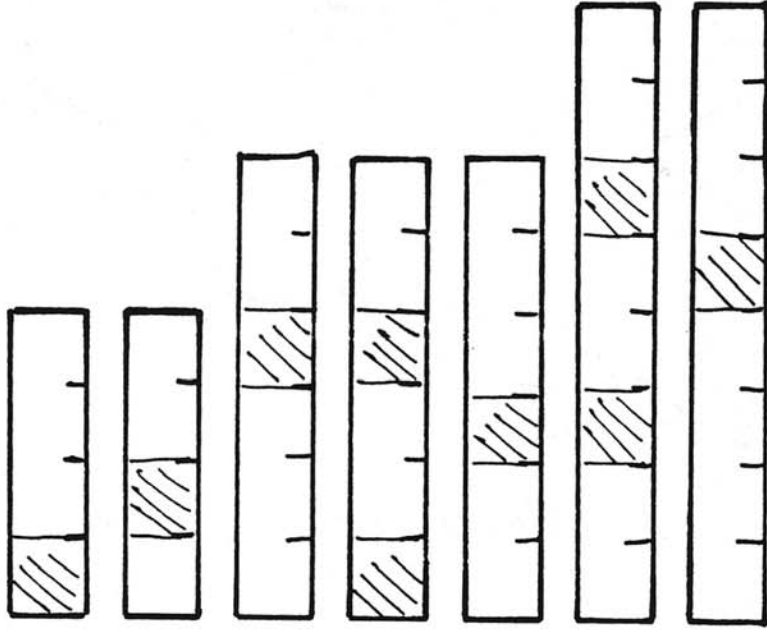
Abbildung B4-6.11

Codierung gemäß RLL 2,7

Datenbits:

1.	2.	3.	4.
1	0		
1	1		
0	0	0	
0	1	0	
0	1	1	
0	0	1	0
0	0	1	1

Bitzellen:



Flußwechsel in Bitzelle



"leere" Bitzelle (kein Flußwechsel)

Abbildung B4-6.12 Flußwechsel bei RLL2,7



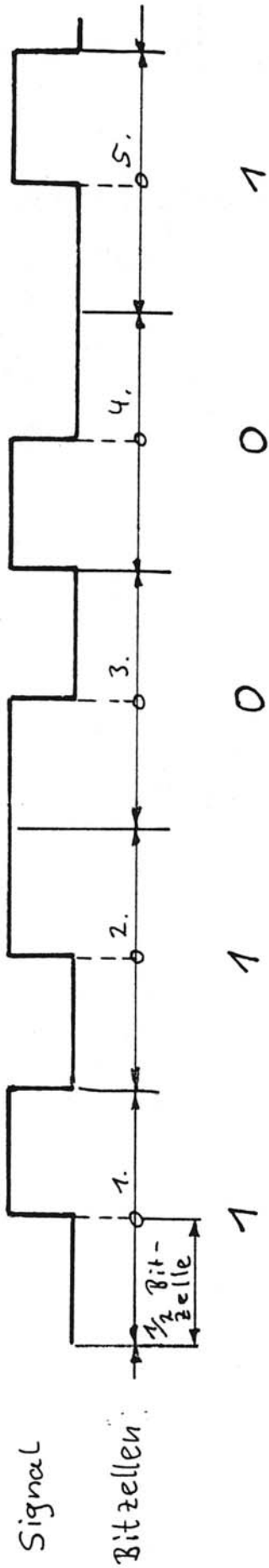
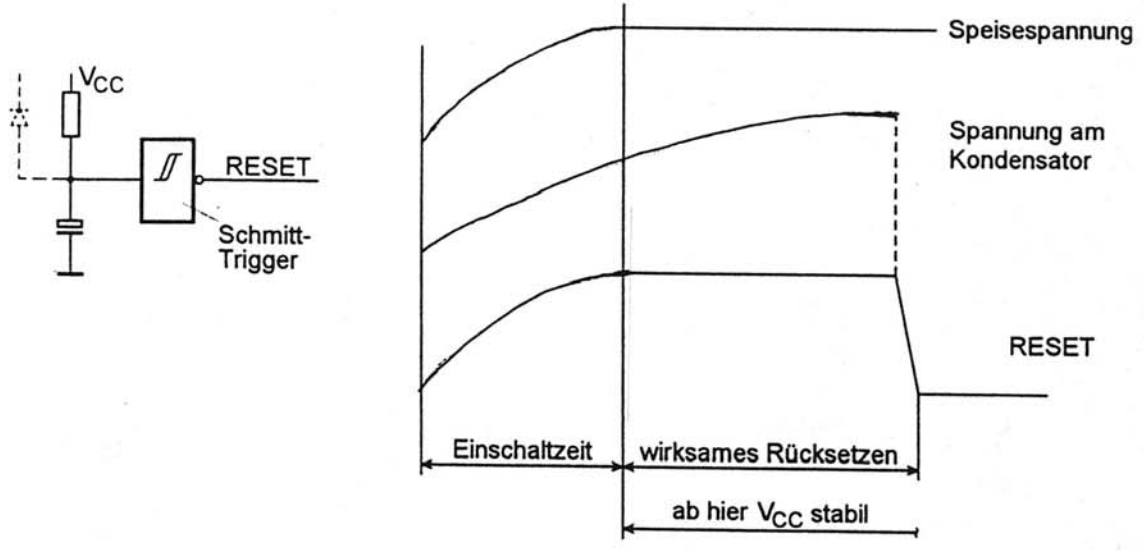


Abbildung B4-6.13 Manchester-Codierung



Rücksetz-Prüfschaltung

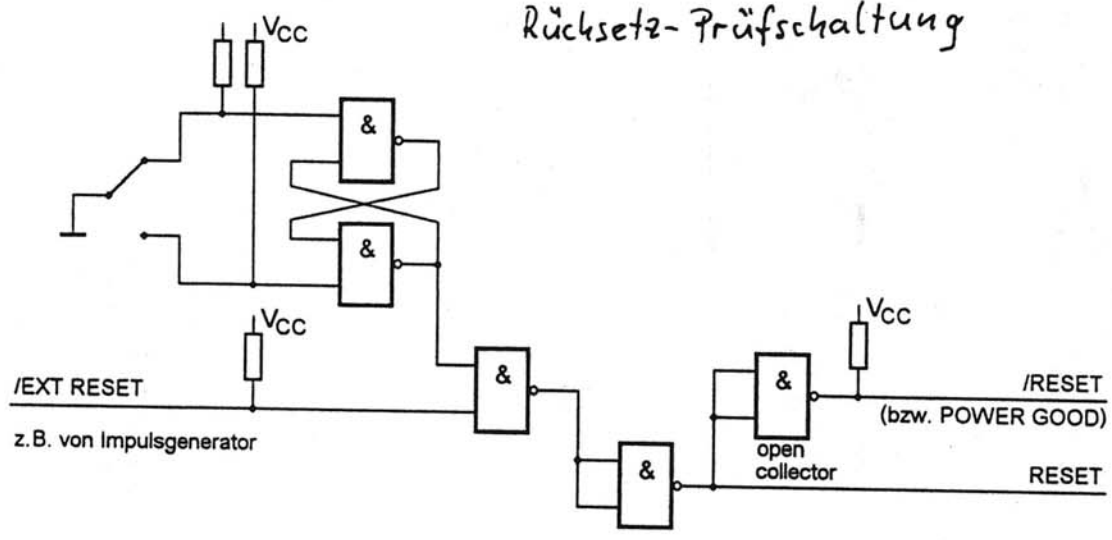
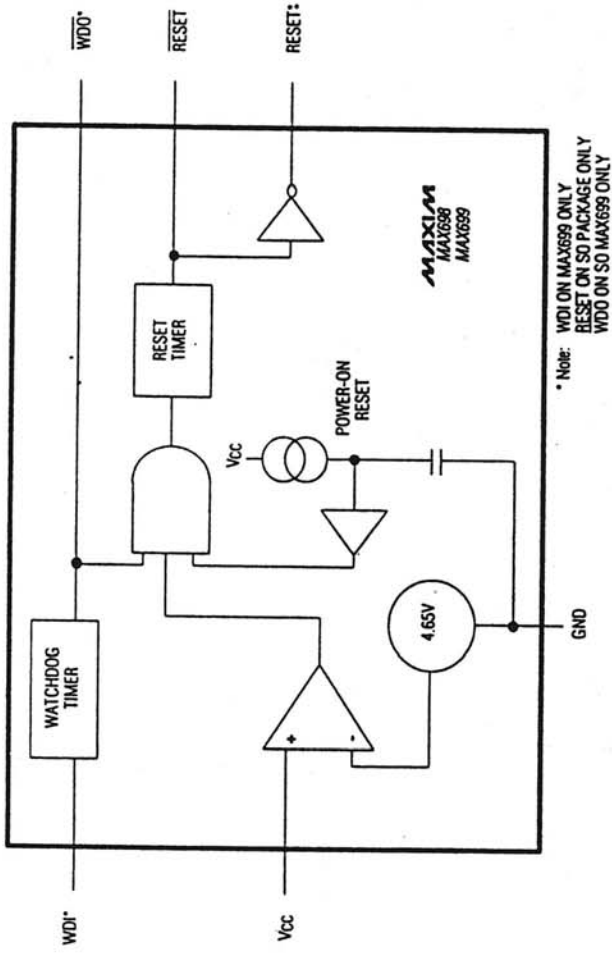


Abb 35-12.8



\* Note: WDI ON MAX698 ONLY  
 RESET ON SO PACKAGE ONLY  
 WDO ON SO MAX699 ONLY

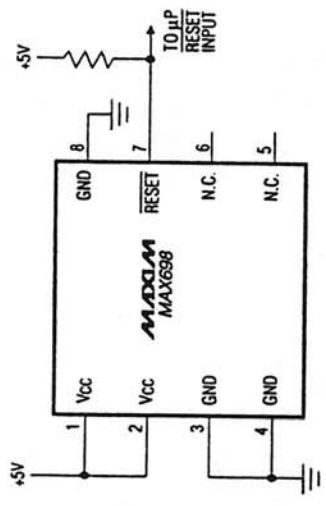
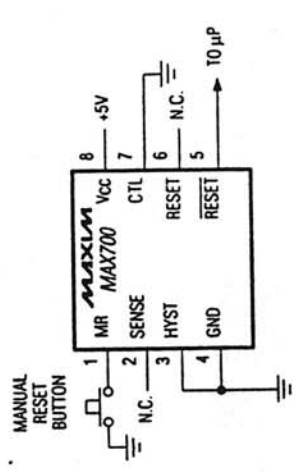
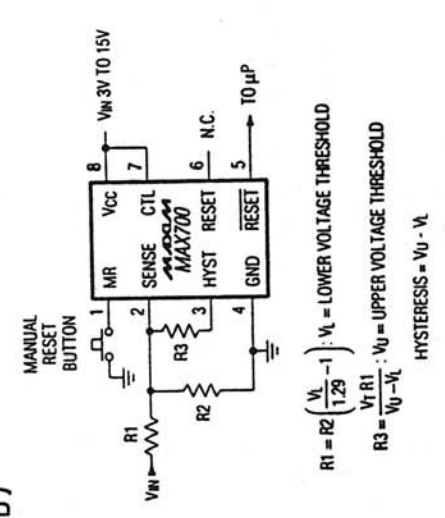


Abbildung B5-12.2

c)



b)



a)

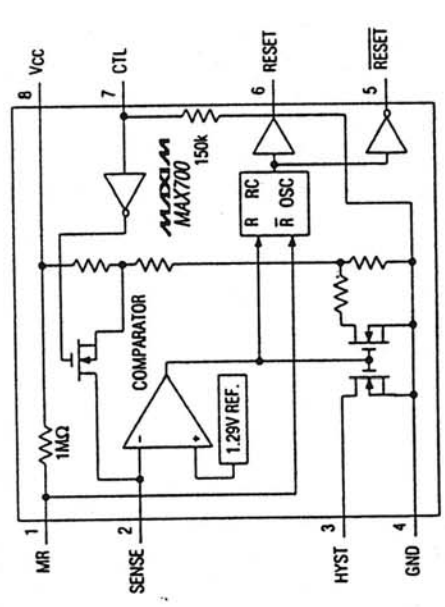
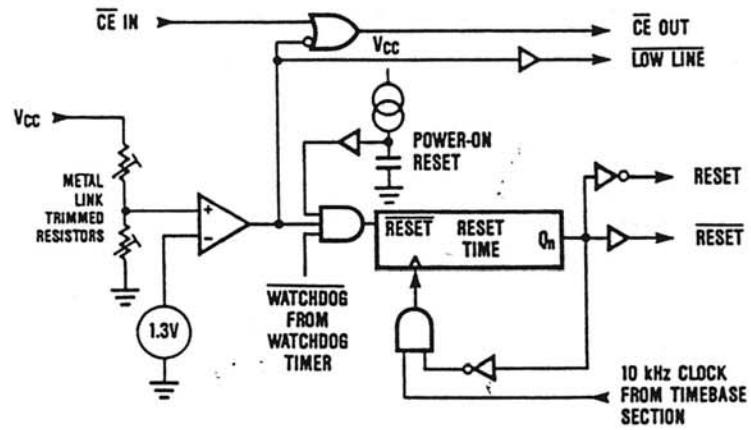
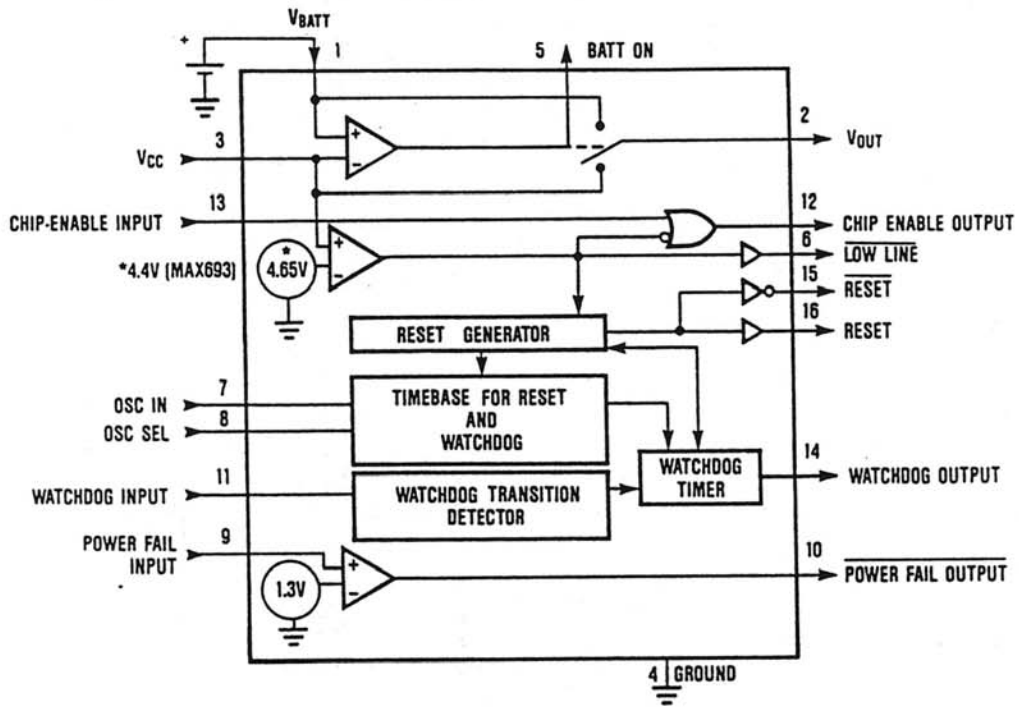
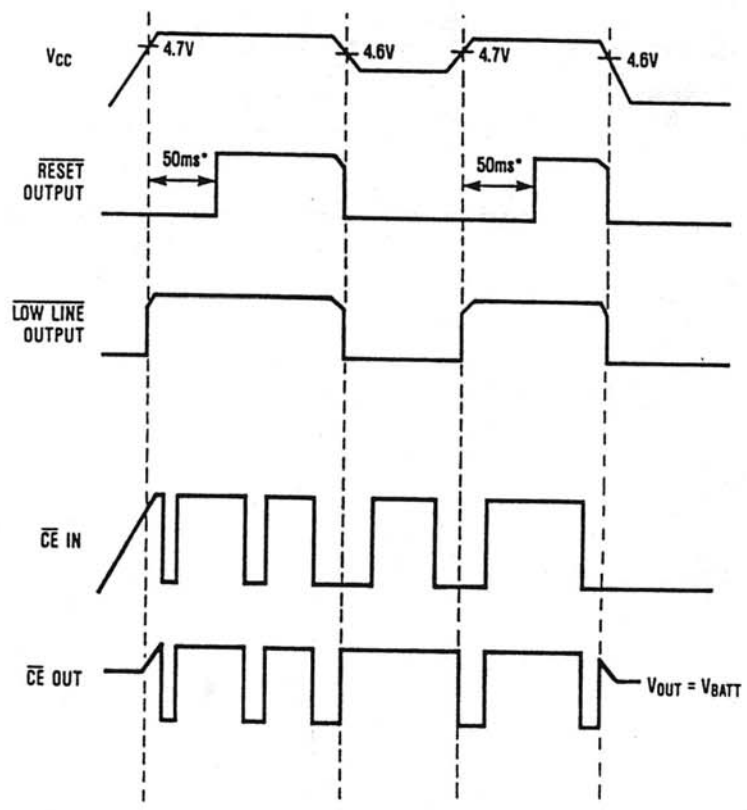


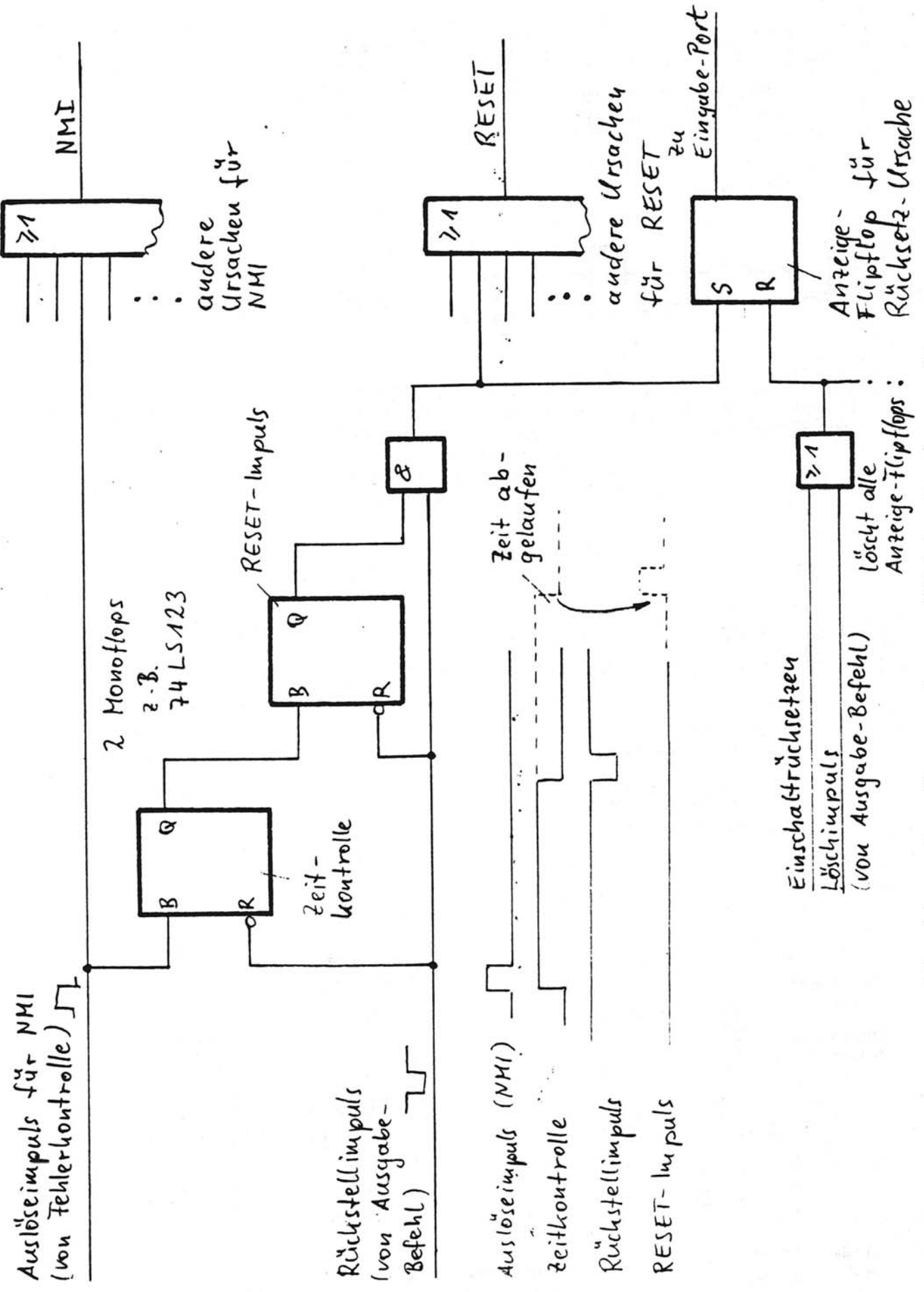
Abbildung BS-12.3





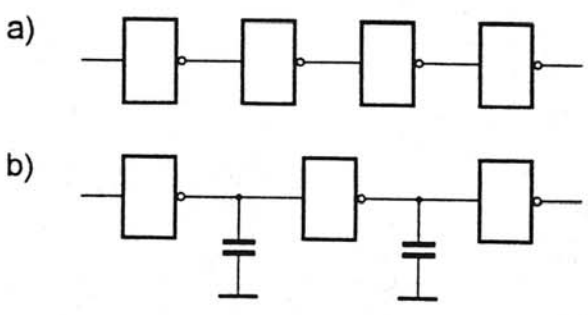
\*200ms for MAX694 and MAX695

Abbildung B5-12.6

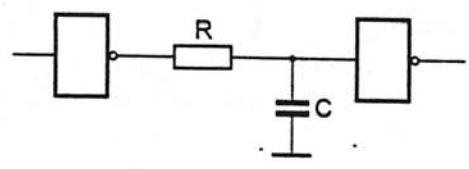


# Zeitkontrolle der Fehlerbehandlung

Abb. B5-12.9



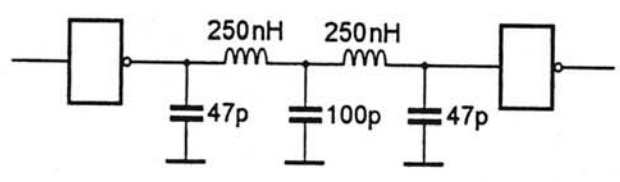
Je Kondensatorstufe  
 470 pF : 50 ns  
 1 μF : 80 ns



Impulsbreite  $\approx R \cdot C$   
 Über R sollte bei  $I_{max}$   
 nicht mehr als 0,3-0,4 V  
 abfallen.  
 ( $R < 220 \dots 670 \Omega$ )

- Bei größeren Zeitkonstanten (geringeren Flankensteilheiten) Schmitt-Trigger verwenden.
- Zeitkonstanten so wählen, daß im Rahmen der max. Impulsfrequenz alle Umladevorgänge abgeschlossen sind (eingeschwungener Zustand), bevor der nächste Impuls eintrifft.

Verzögerung  
von Impulsen

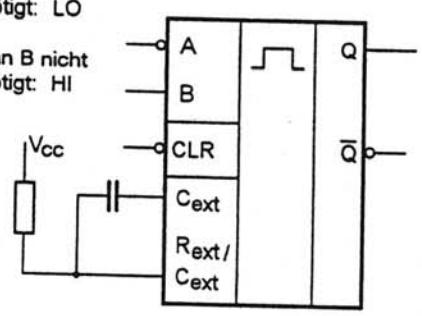


10 ns - Laufzeitkette  
 250 nH  $\hat{=}$  17 cm Leiterzug.  
 Spule läßt sich als gedruckte Spirale ausführen.

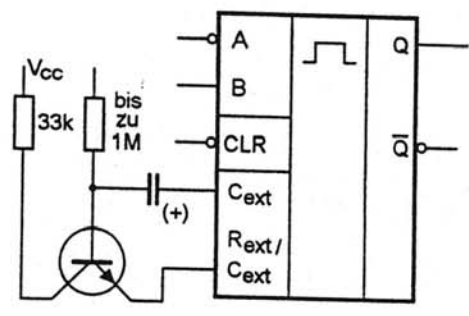
Abb. B5-12.11

Monoflops

Wenn A nicht benötigt: LO  
 Wenn B nicht benötigt: HI



Außenbeschaltung für extreme Zeitkonstanten:



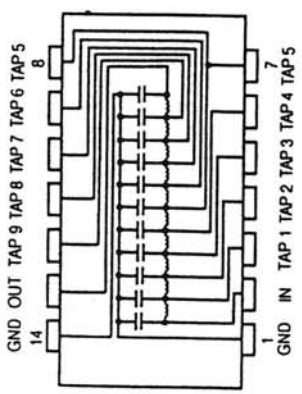
Achtung  $\text{---} \text{---} \text{---}$  '221  
 bei Elko!  $\text{---} \text{---} \text{---}$  '123

Für '221:  $t_p \approx 0,7 \cdot C_{ext} R_{ext}$

Toleranz typisch  $\pm 0,5\%$

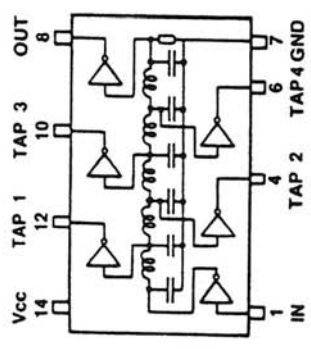
$R_{ext}$  2k ... 100k (LS 221),  $C_{ext}$  10pF ... 10μF  
 ohne  $C_{ext}$ :  $t_p \approx 30ns$





14-poliges DIL-Gehäuse

Typ	Gesamt- verzögerung	Verzögerung zwischen Anzapfungen	Toleranz Eingang- Anzapfung	Ausgangs- Anstiegszeit
10012E	10 ± 2	1 ± 1	± 2	3
10112E	100 ± 5%	10 ± 10%	± 5%	17
20112E	200 ± 5%	20 ± 10%	± 5%	34
25112E	250 ± 5%	25 ± 10%	± 5%	42

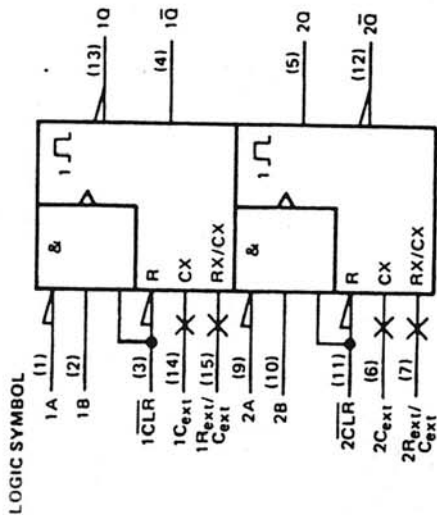


5 Laufzeitverzögerungsabgriffe, Schottky TTL gepuffert. 14 Pin Plastikgehäuse.

Typ	Gesamt- verzögerung	Verzögerung zw. zwei Abgriffen	Toleranz Eingang/ Abgriff	Anstiegszeit Ausgang
42A5250	25 ± 2	5 ± 2	± 2	3
42A5101	100 ± 5%	20 ± 2	± 2	3
42A5161	150 ± 5%	30 ± 2	± 3	4

Abbildung B5-12-12

b)



a)

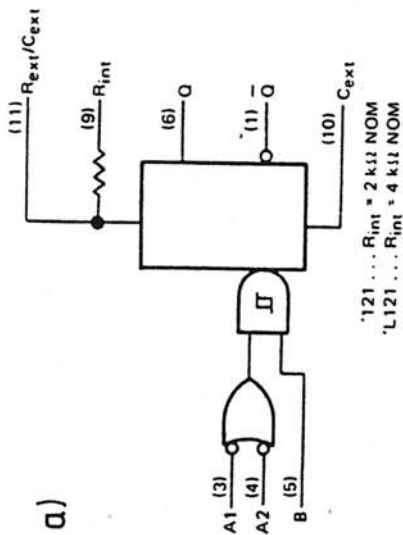
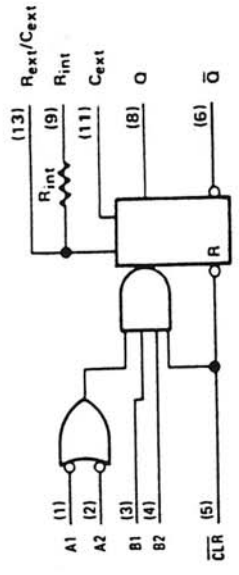
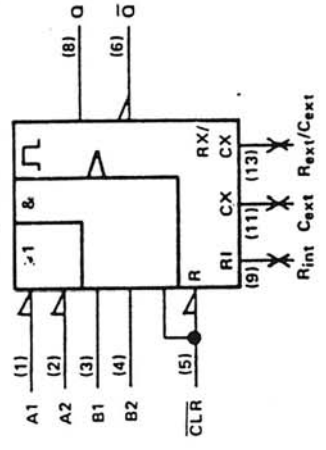


Abbildung B5-12.13

'122, 'LS122



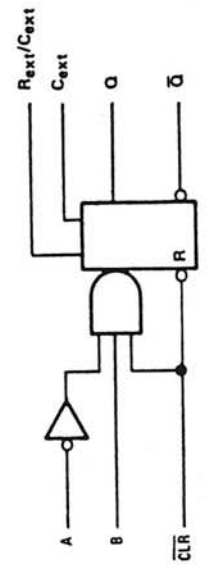
'122, 'LS122



$R_{int}$  is nominally 10 k ohms for '122, 'LS122.

logic diagram (each multivibrator)

'123, LS123



logic symbol

'123, 'LS123

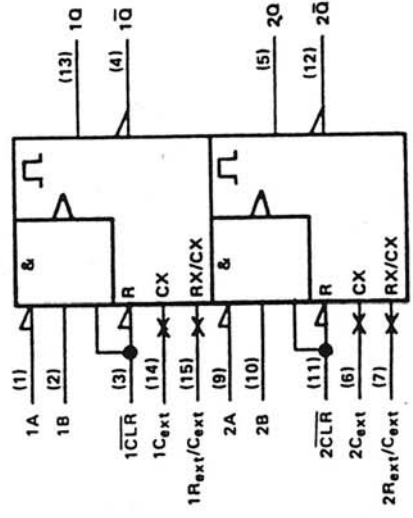


Abbildung B5 - 12.14

# Zeitgeber '555

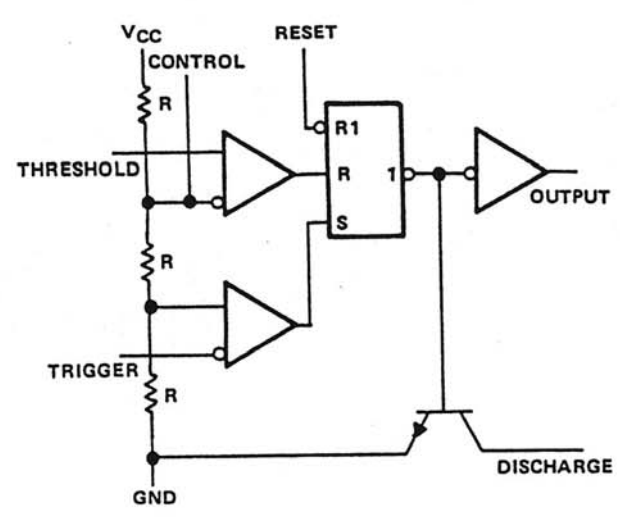


Abbildung BS-12.16

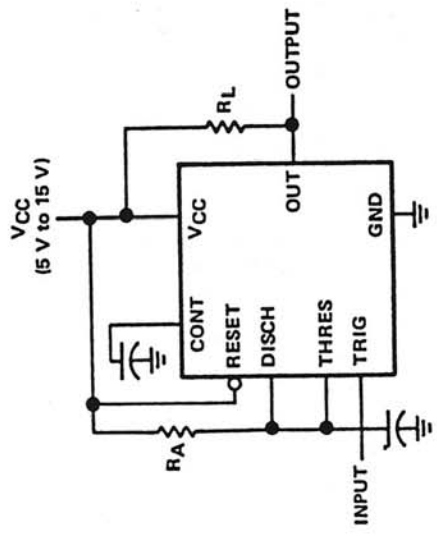
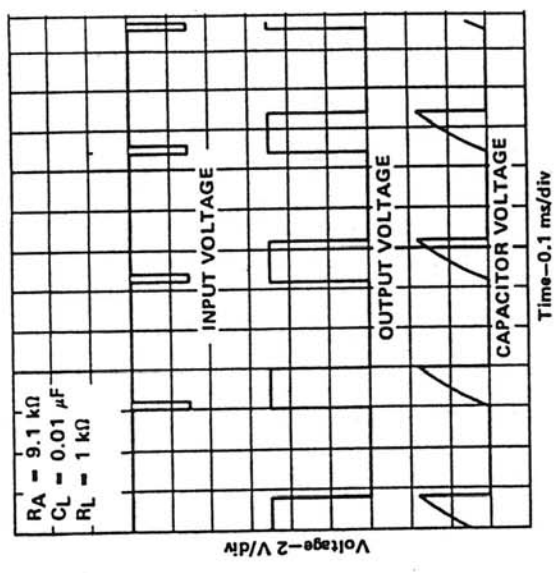


Abbildung BS-12.17

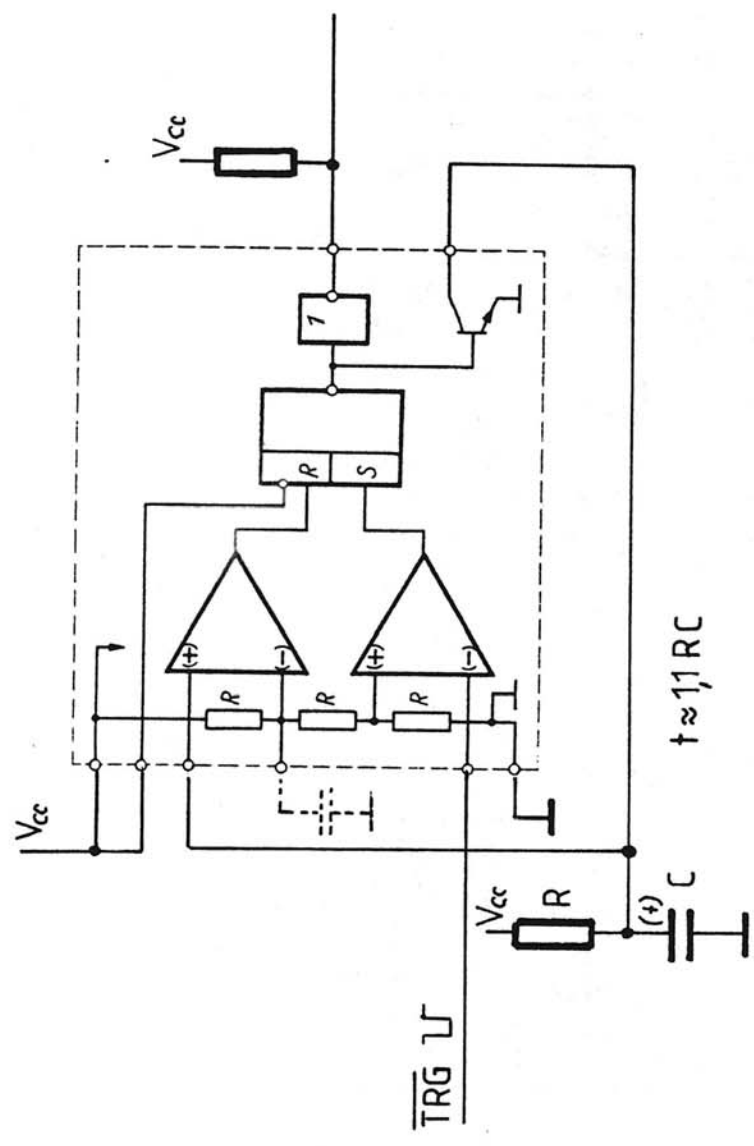
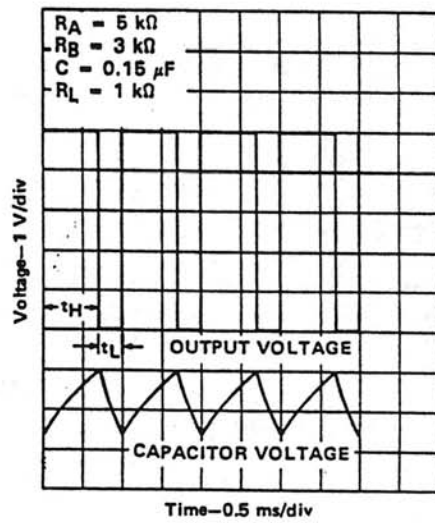
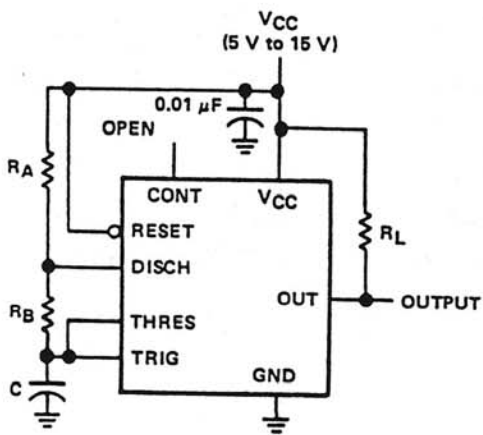
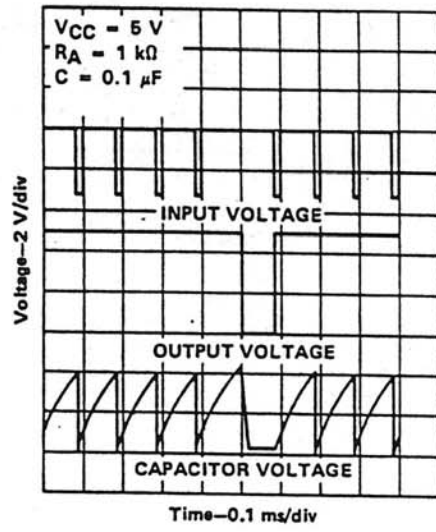
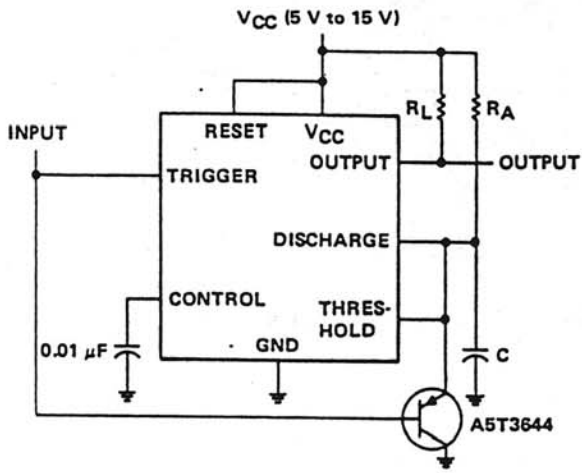
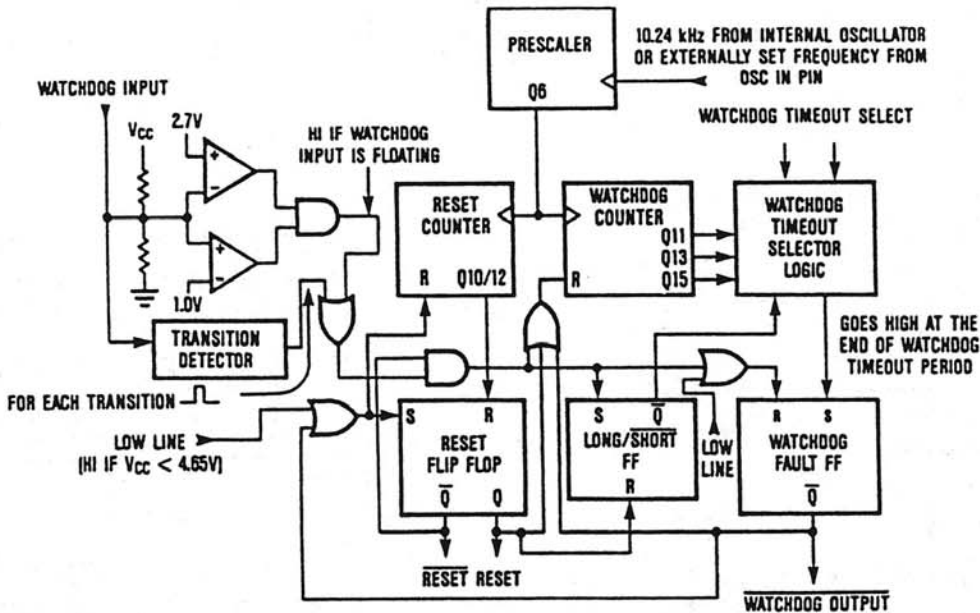


Abbildung B5-12.18





"Digitalisierter" Zeitgeber

OSC SEL	OSC IN	Watchdog Timeout Period		Reset Timeout Period	
		Normal	Immediately After Reset	MAX691/93	MAX695
Low	External Clock Input	1024 clks	4096 clks	512 clks	2048 clks
Low	External Capacitor	$\frac{400\text{ms}}{47\text{pF}} \times C$	$\frac{1.6 \text{ sec}}{47\text{pF}} \times C$	$\frac{200\text{ms}}{47\text{pF}} \times C$	$\frac{800\text{ms}}{47\text{pF}} \times C$
Floating	Low	100ms	1.6 sec	50ms	200ms
Floating	Floating	1.6 sec	1.6 sec	50ms	200ms

Note 1: The MAX690/692/694 watchdog timeout period is fixed at 1.6 seconds nominal, the MAX690/692 Reset pulse width is fixed at 50ms nominal and the MAX694 is 200ms nominal.

Note 2: When the MAX691 OSC SEL pin is low, OSC IN can be driven by an external clock signal, or an external capacitor can be connected between OSC IN and GND. The nominal internal oscillator frequency is 10.24kHz. The nominal oscillator frequency with external capacitor is

$$F_{osc}(\text{Hz}) = \frac{184,000}{C(\text{pF})}$$

Note 3: See Electrical Characteristics Table for minimum and maximum timing values.



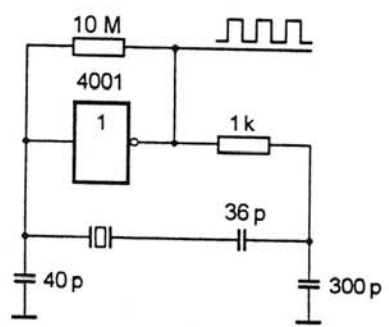
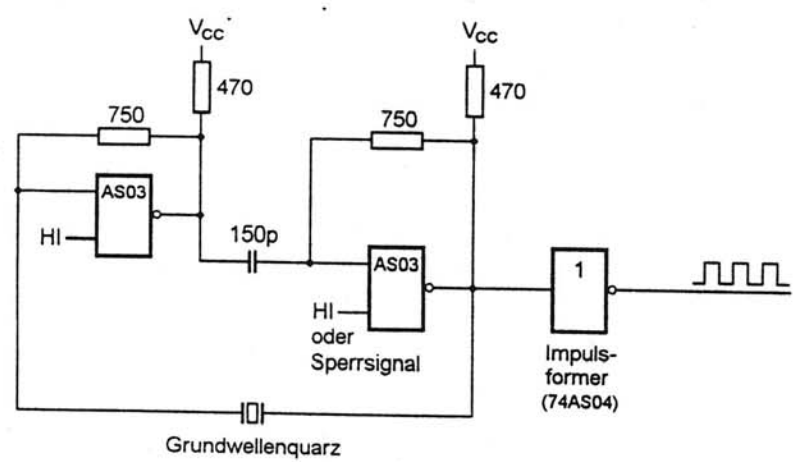
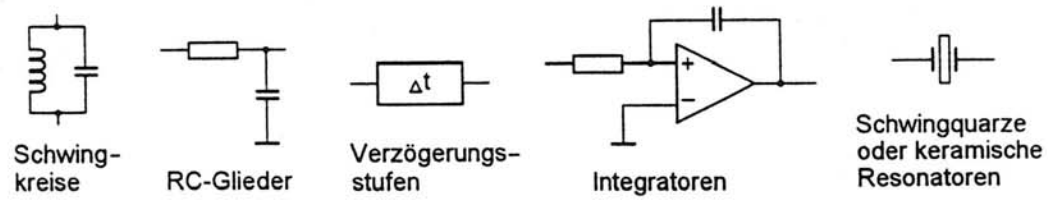
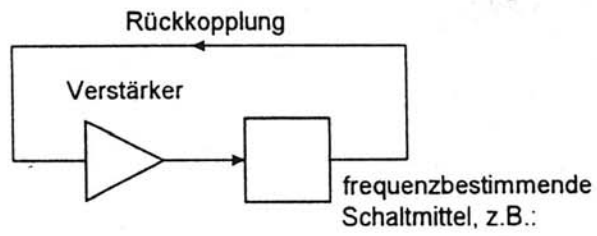
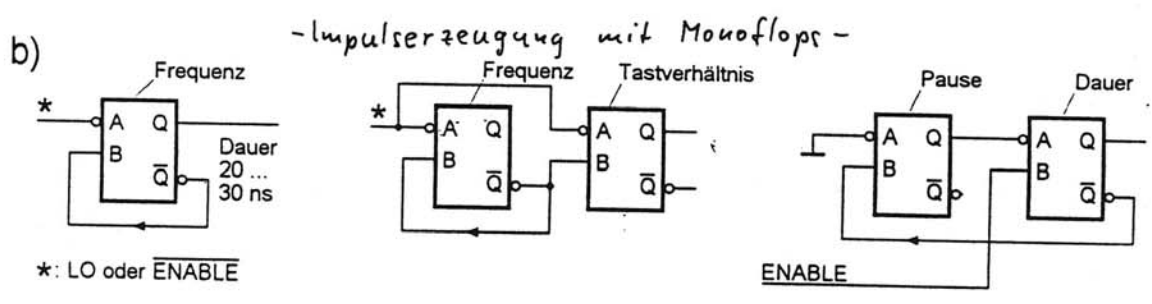
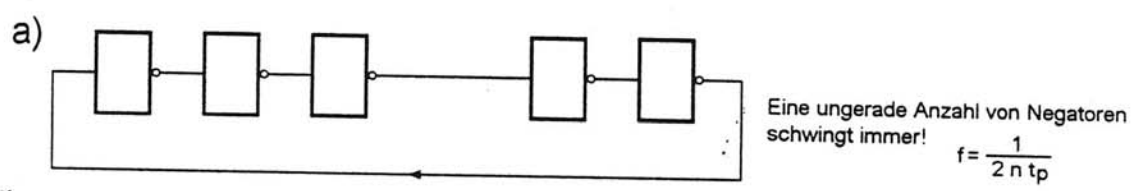


Abb B5-13



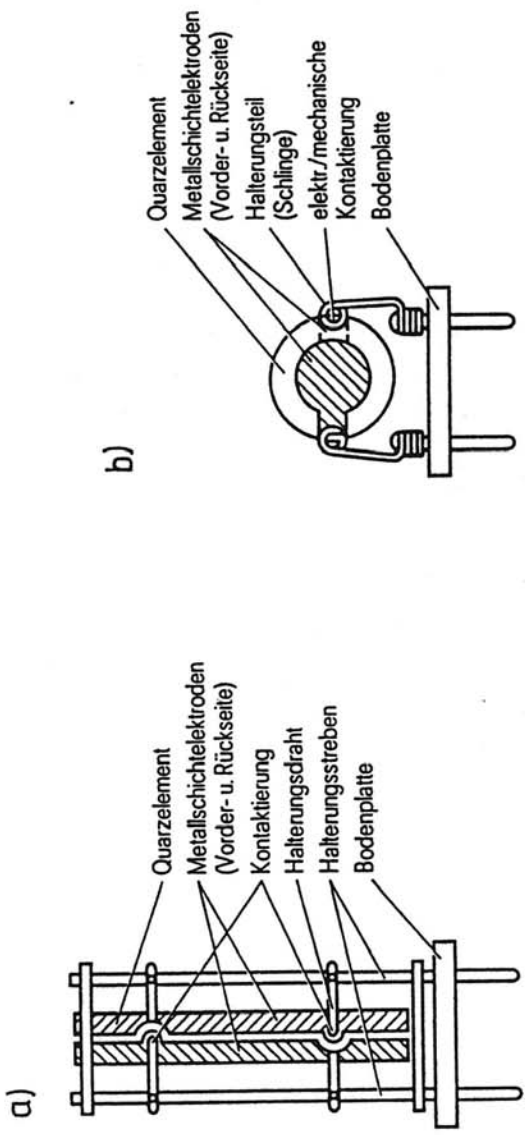
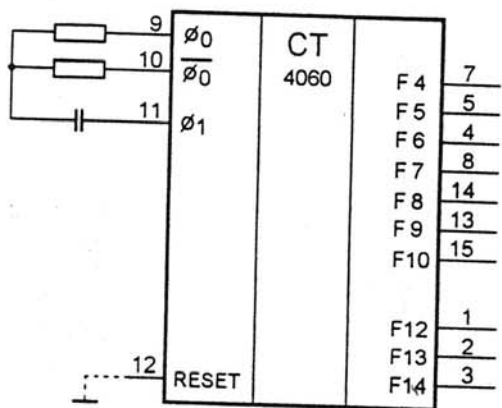
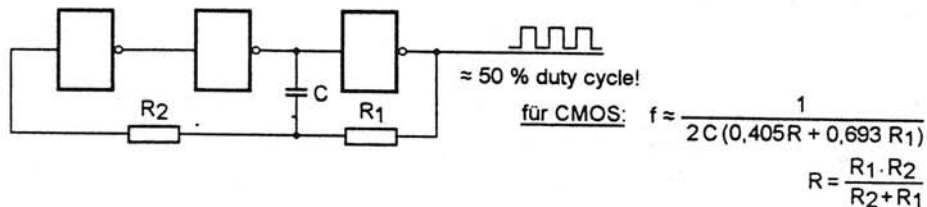
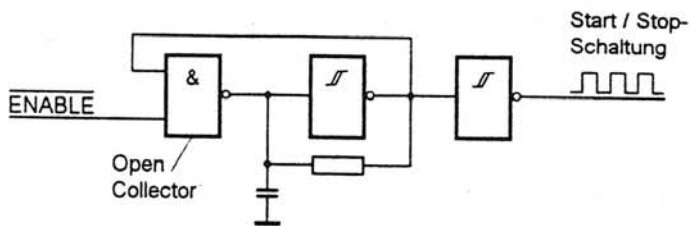
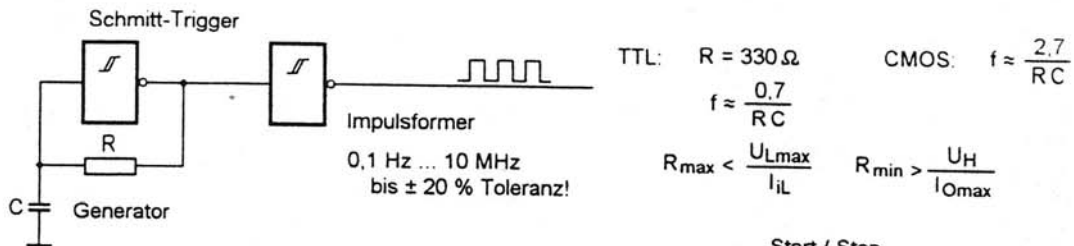


Abbildung B5-13-2



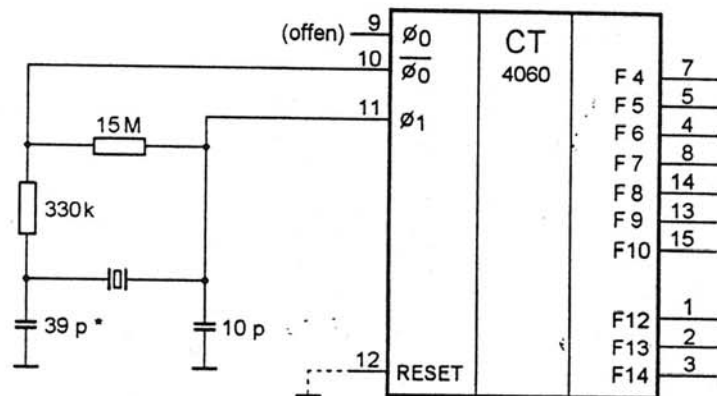
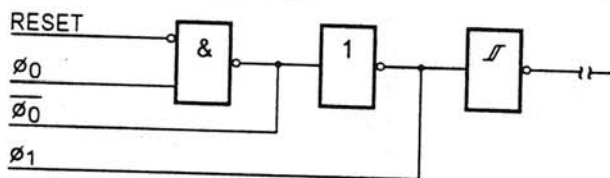


$F_n$  entspricht Teilung durch  $2^n$

max. Taktfrequenz bei

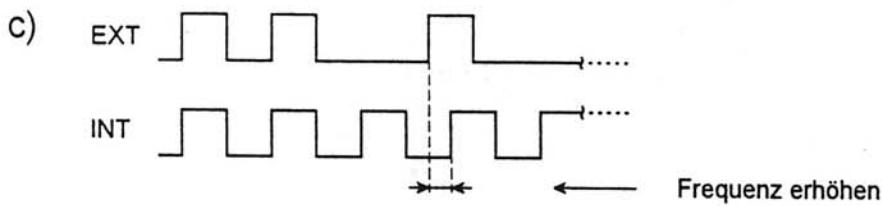
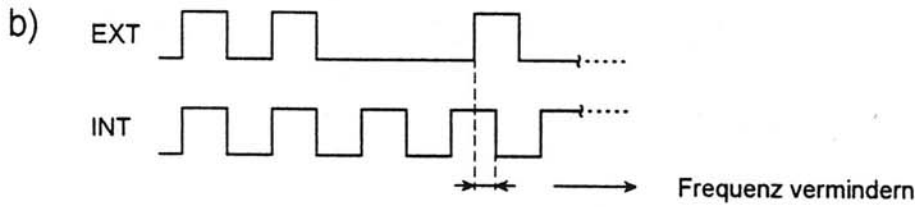
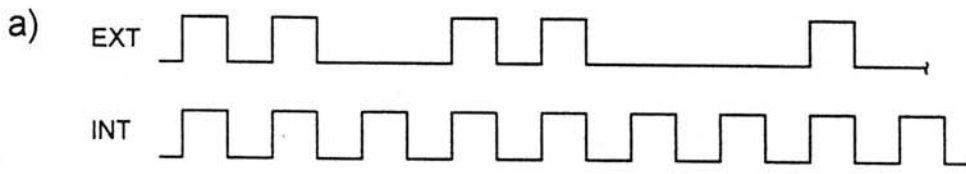
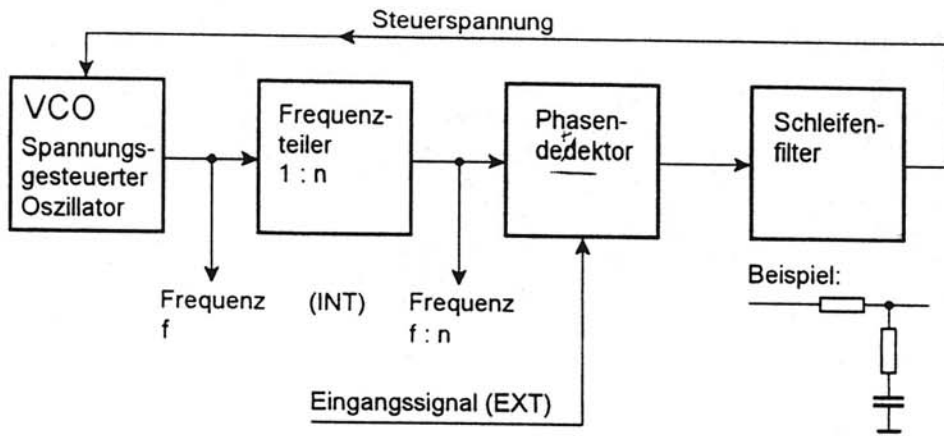
- 5 V 1 MHz
- 10 V 3 MHz
- 15 V 4 MHz

Innenschaltung Eingänge:

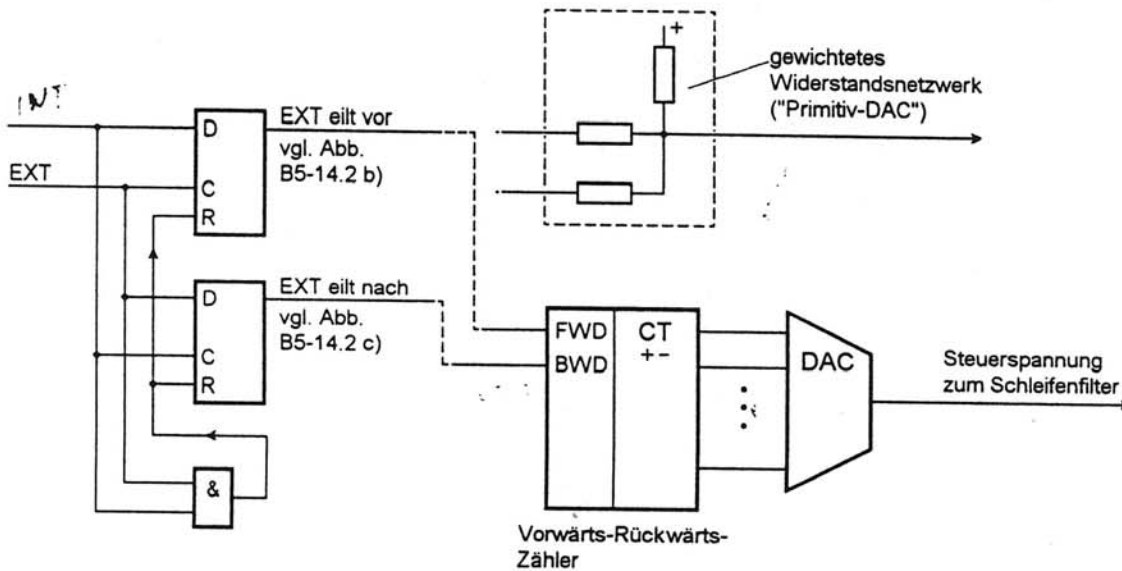


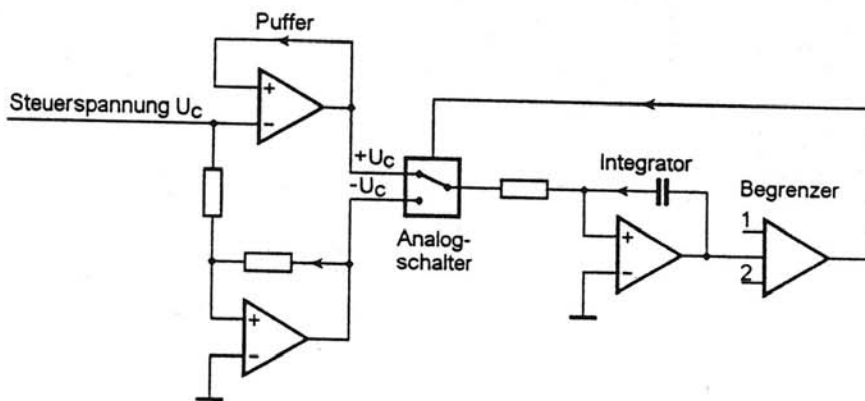
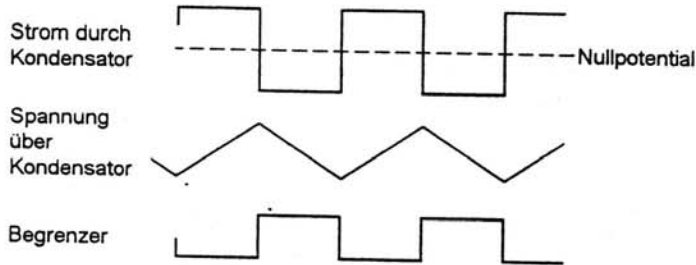
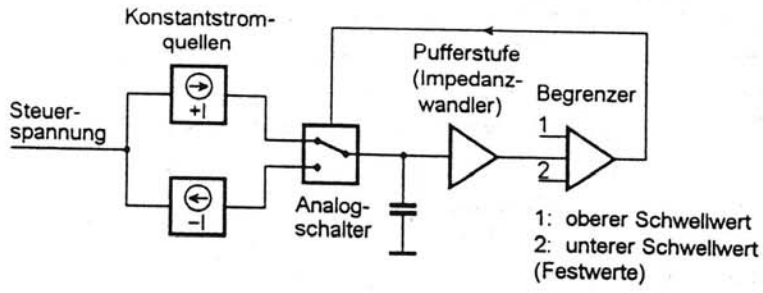
\* wenn erforderlich, als Trimmer

# PLL - Grundschialtung

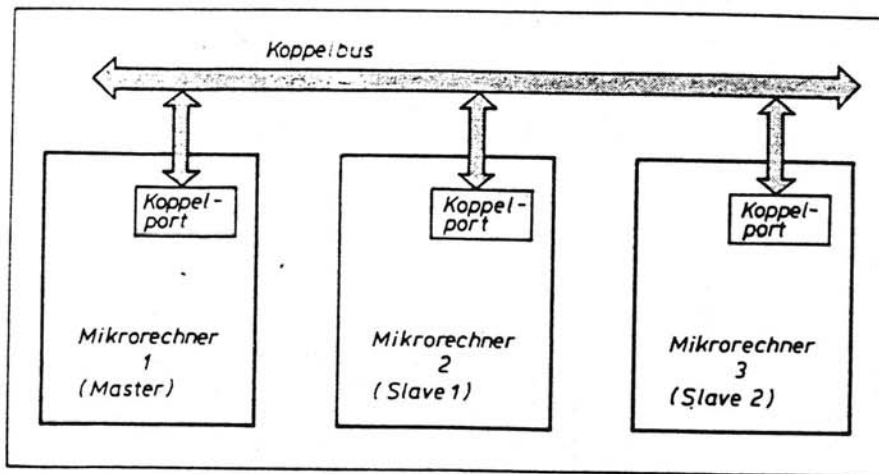


## Alternativen zur Gewinnung der Steuerspannung (Beispiele):

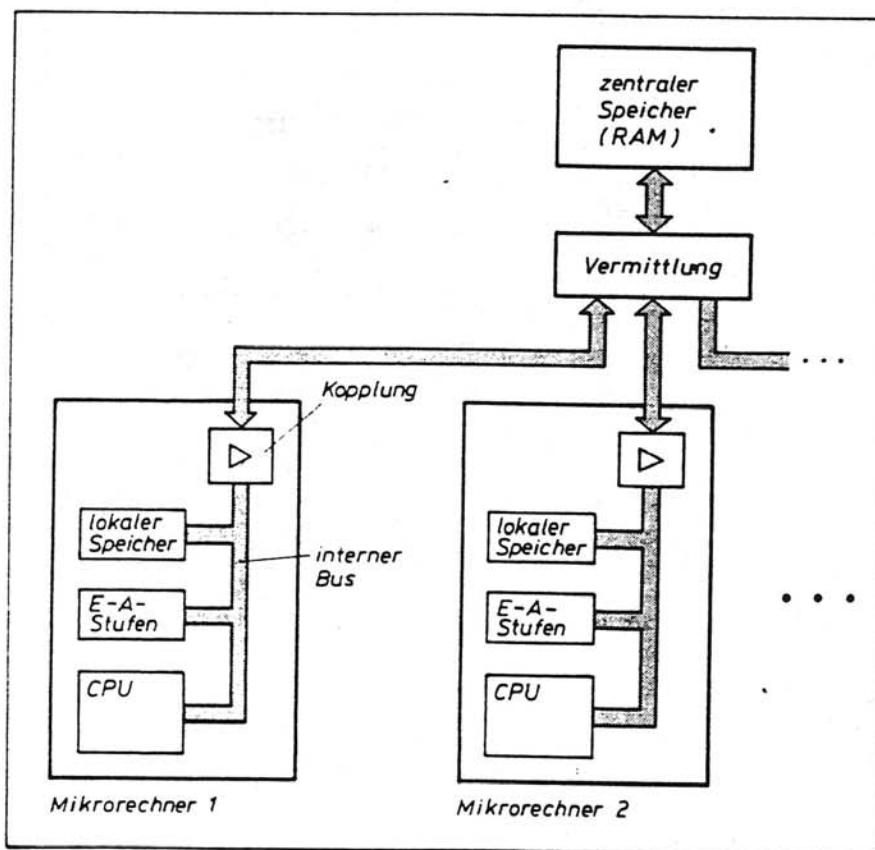




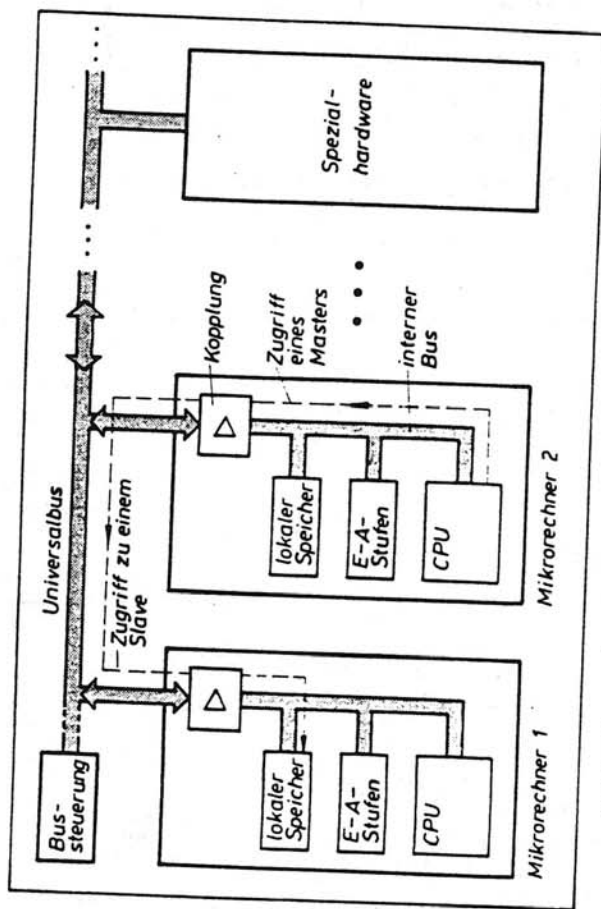
Üb 35-13



Multimikrorechnerkopplung durch E-A-Ports



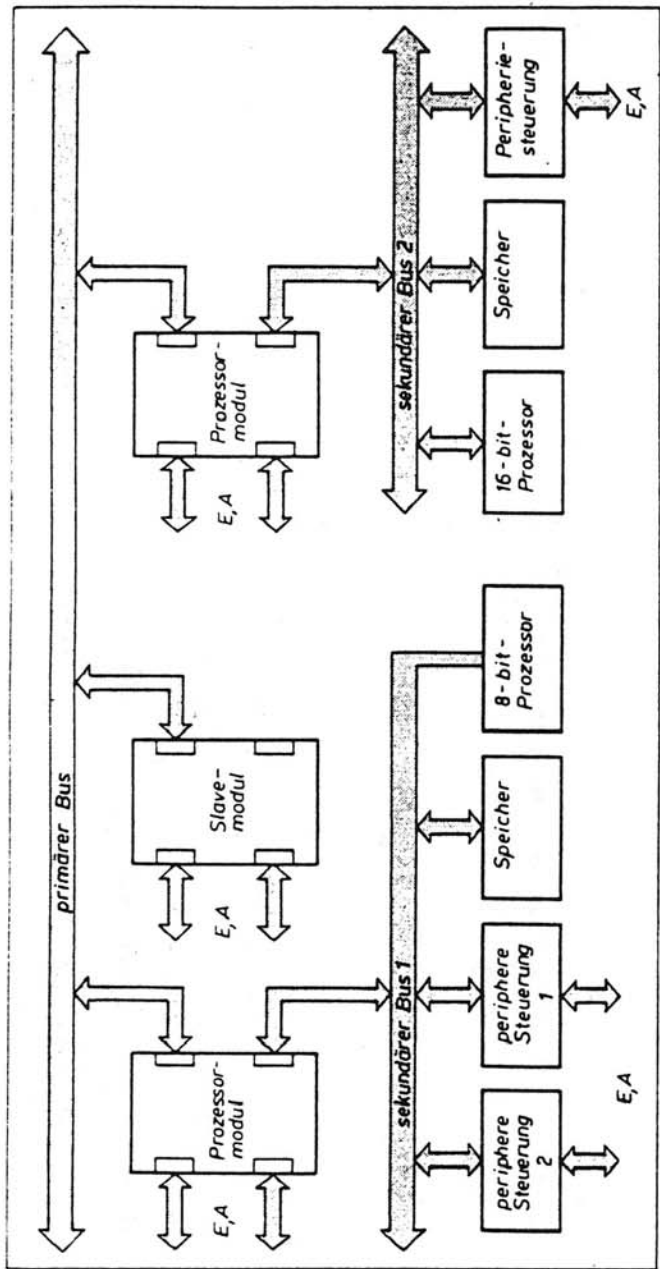
Multimikrorechnerkopplung durch zentralen Speicher

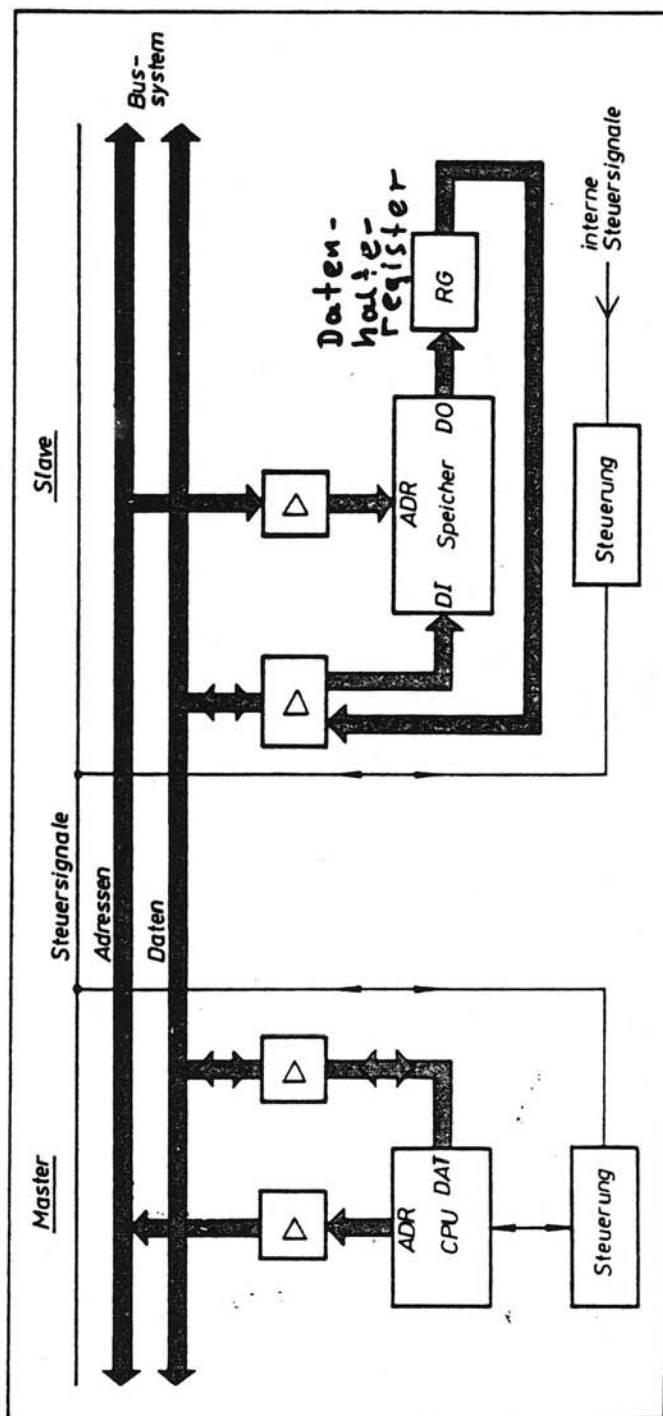


Multimikrorechnerkopplung durch gemeinsamen Bus



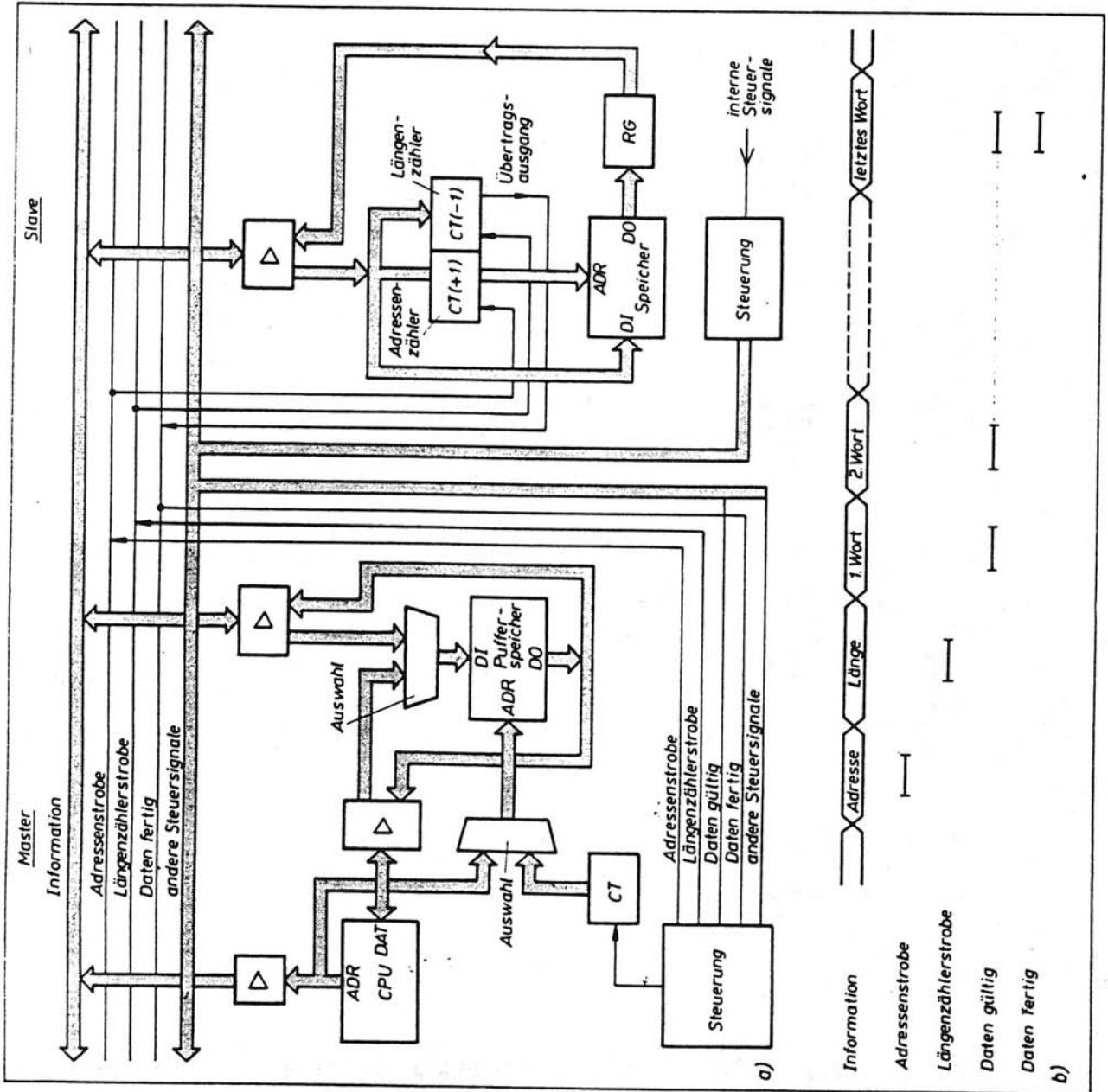
Komplexes Multimikrorechnersystem  
mit unterschiedlichen Bussystemen



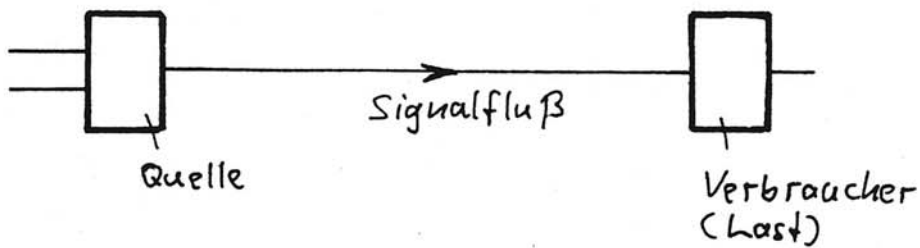


Parallele Übertragung von Daten und Adressen

Paketweise Übertragung von Datenblöcken über zeitmultiplex organisiertes Bussystem. a) Prinzipschaltung; b) Zeitdiagramm

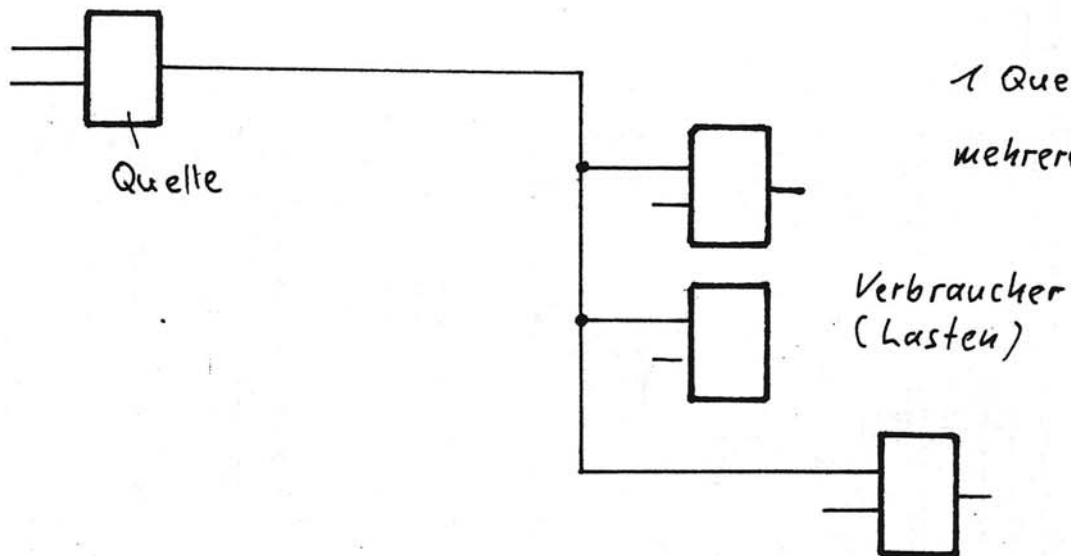


## a) Punkt-zu-Punkt-Verbindung



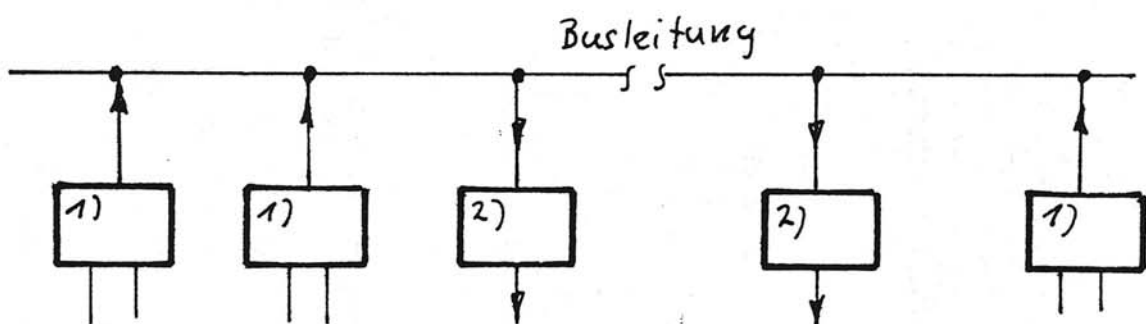
1 Quelle,  
1 Verbraucher

## b) Mehrpunktverbindung



1 Quelle,  
mehrere Verbraucher

## c) Busleitung



1) : Aufschaltung (Treiber)  
2) : Verbraucher (Empfänger)

Abbildung 4-5.1 Einzweckverbindungen  
und Busleitungen

Einrichtungen am Bus (Steckkarten)

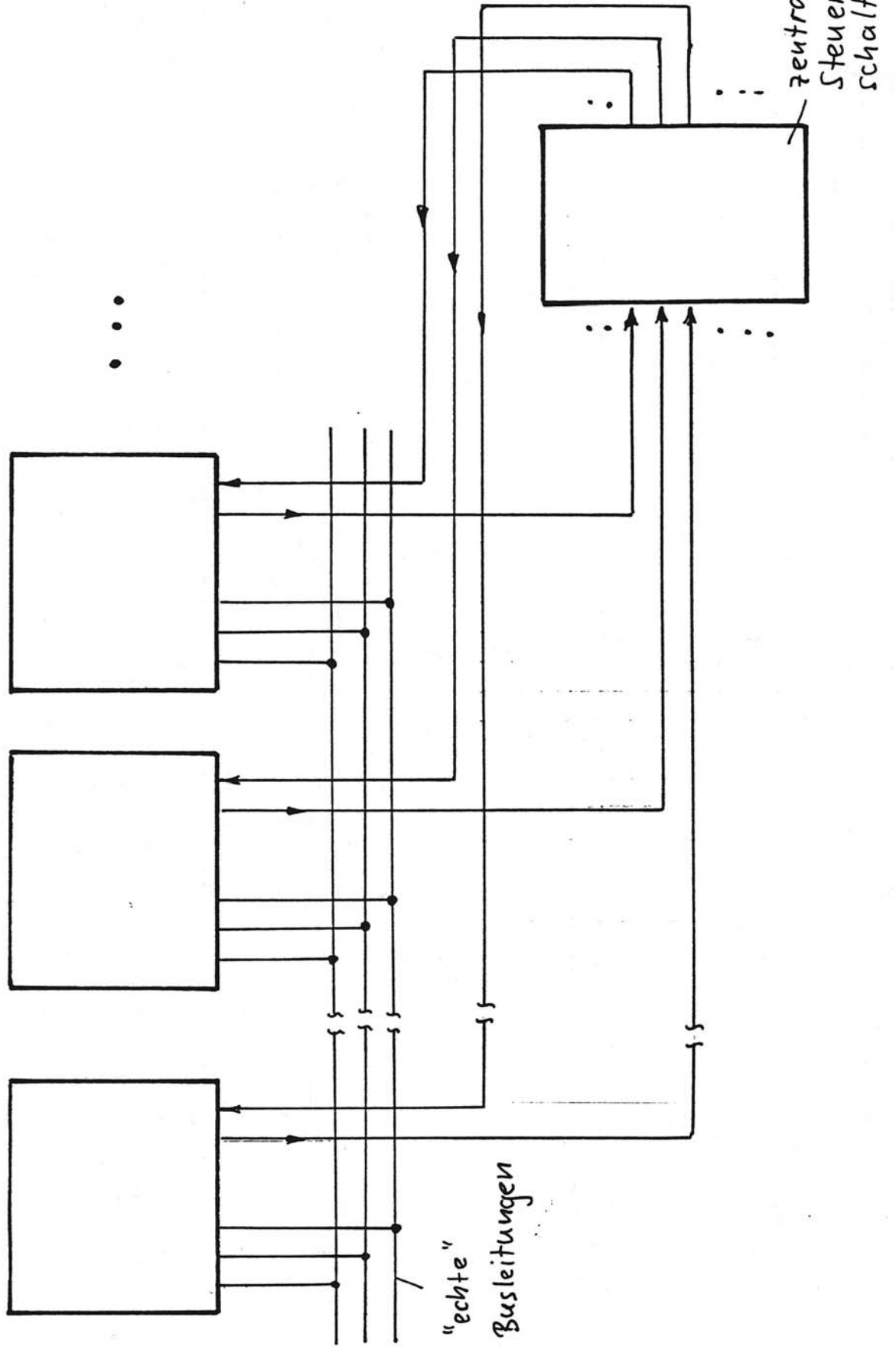


Abbildung 4-5.14 Steckpositionsbezogene Leitungen

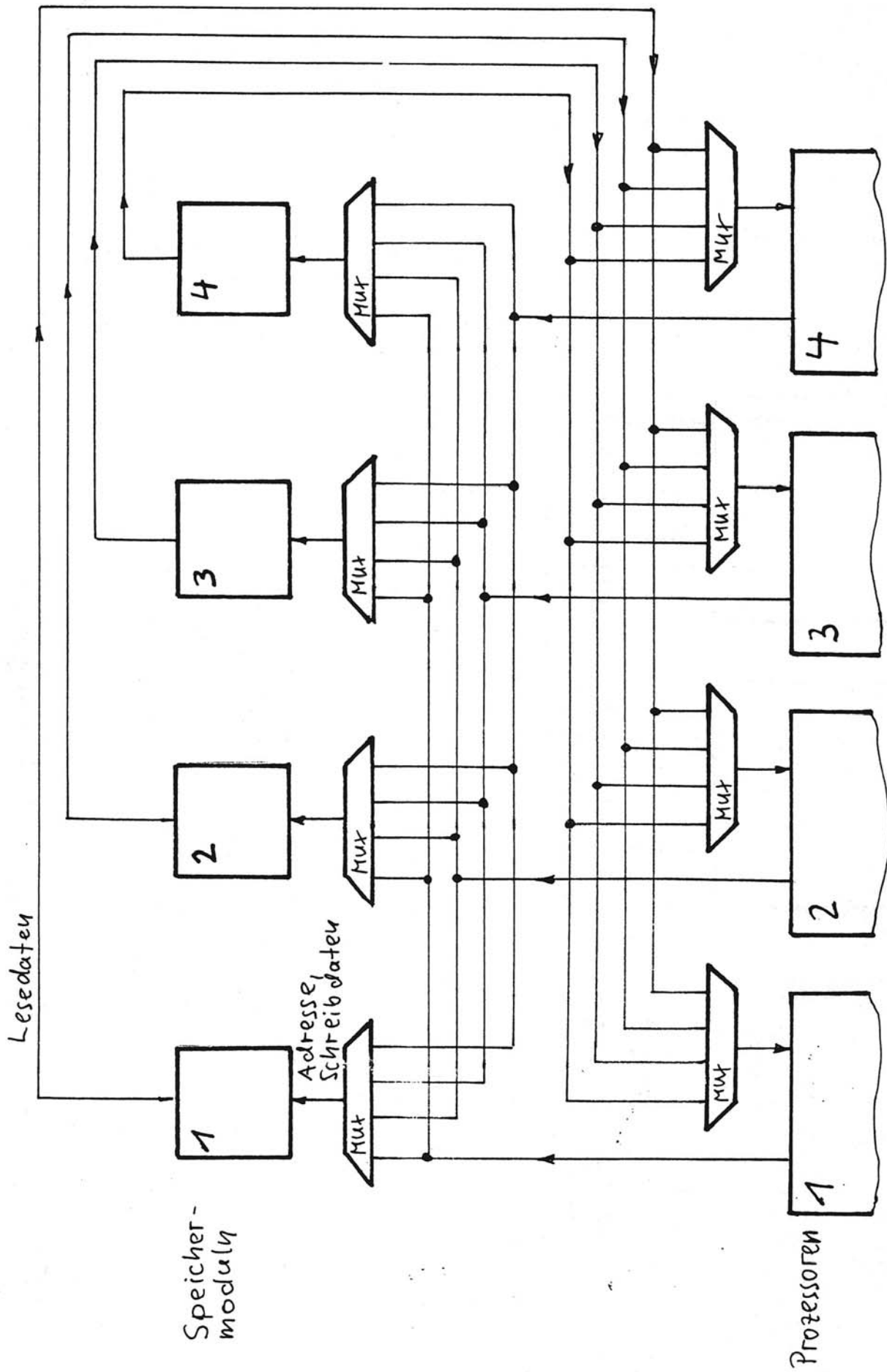
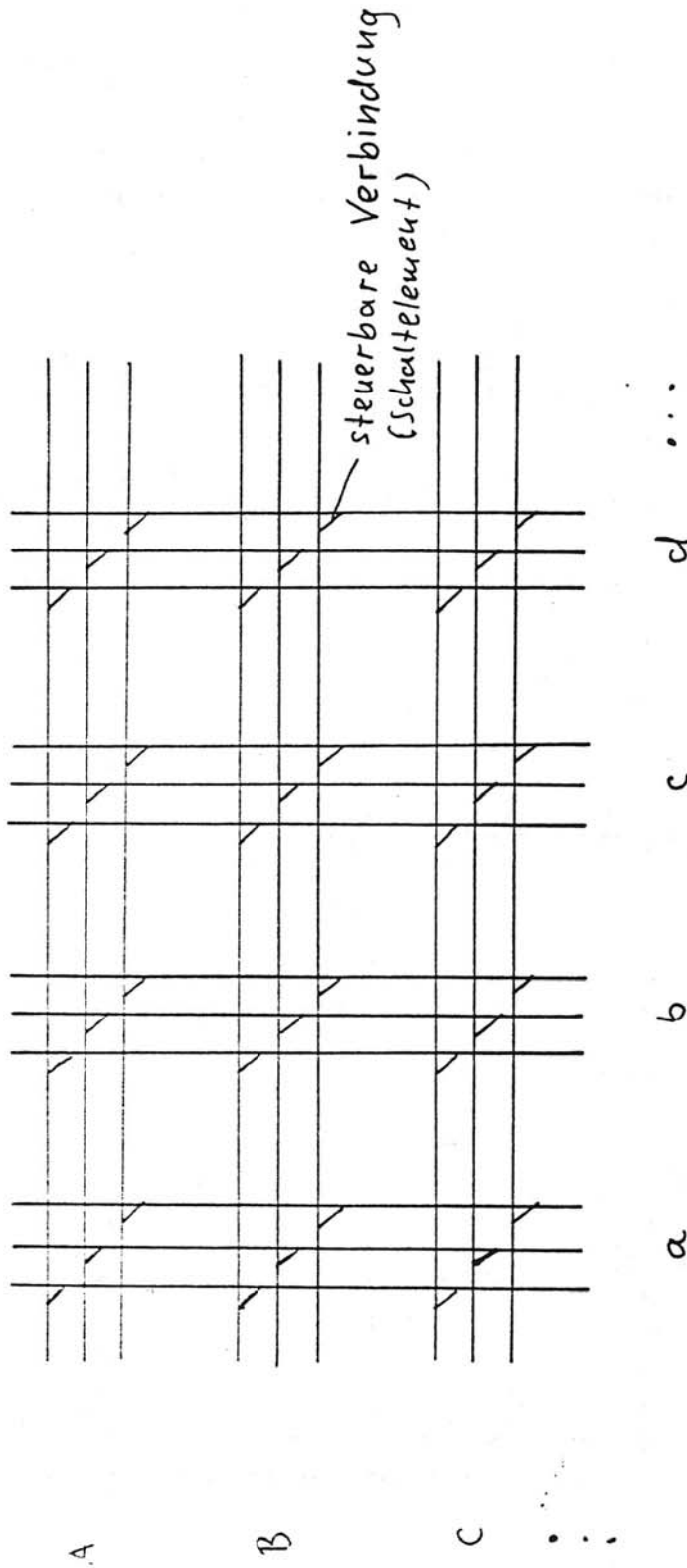


Abbildung 4-5.11 Crossbar-Verbindung mit Multiplexern

Eingänge

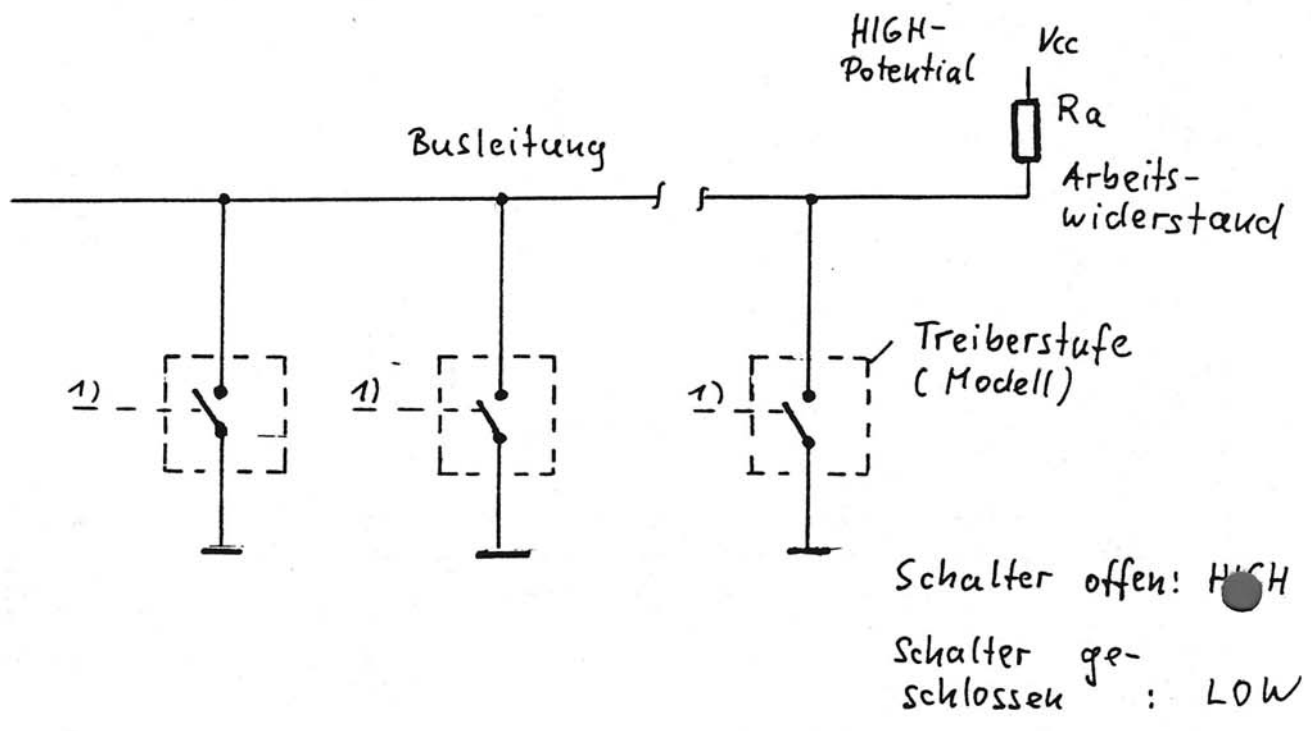


Ausgänge

Die zu verbindenden Einrichtungen sind  $A, B, C, \dots$ ,  $a, b, c, d, \dots$ . Es sind jeweils 3 zu verbindende Signalleitungen dargestellt.

Abbildung 4-5.12 Crossbar-Struktur

### a) Open-Collector-Prinzip



### b) Tri-State-Prinzip

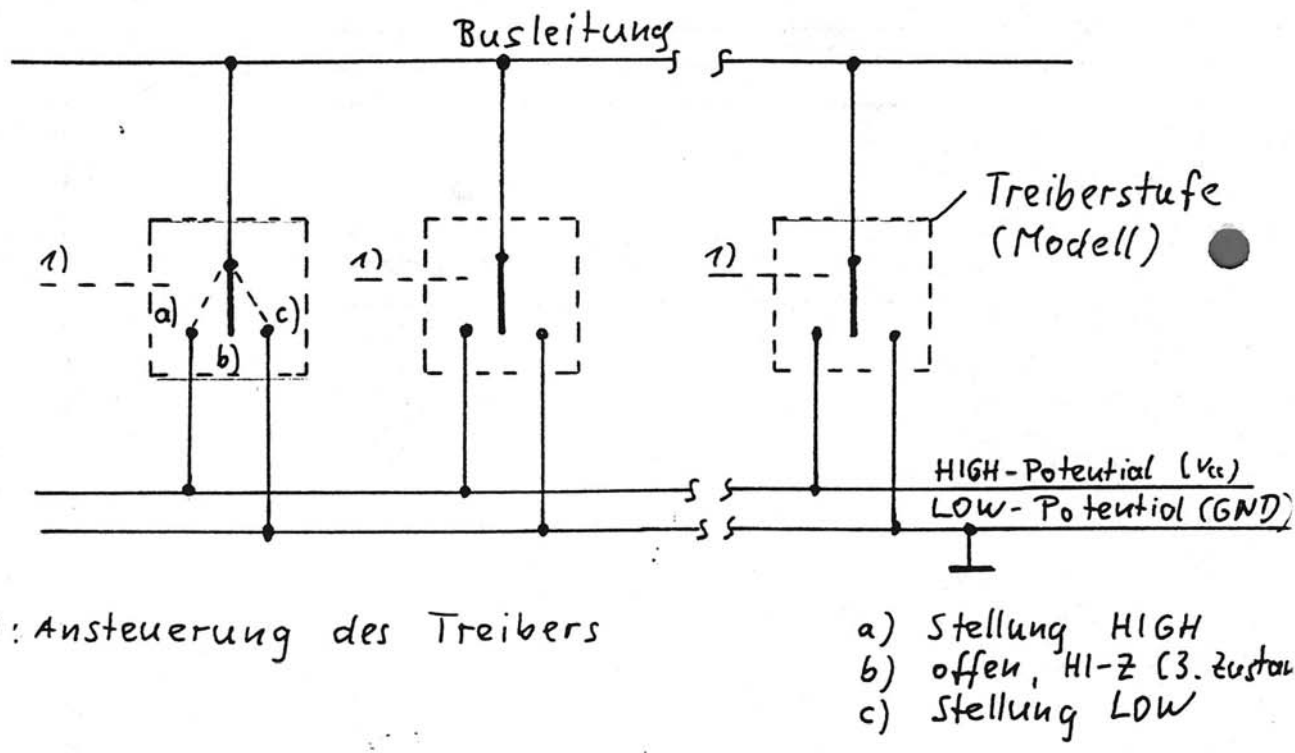


Abbildung 4-5.2 Prinzipien der Busanschaltung (Modellvorstellungen)



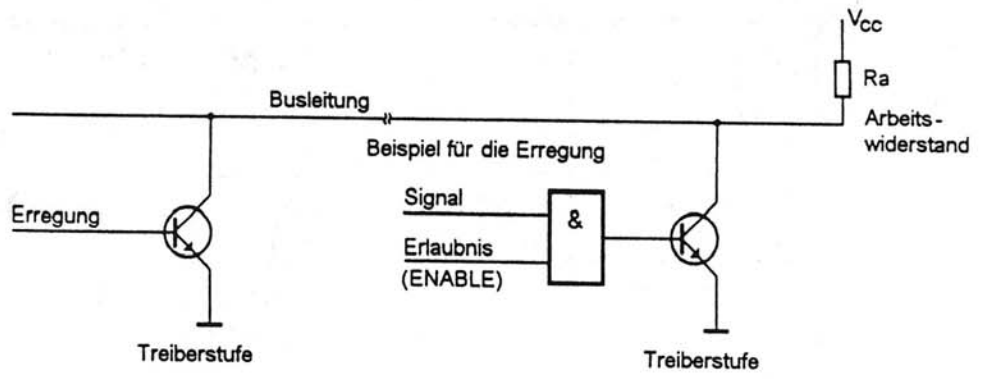


Abbildung 4-5.5  
Open-Collector-Prinzip

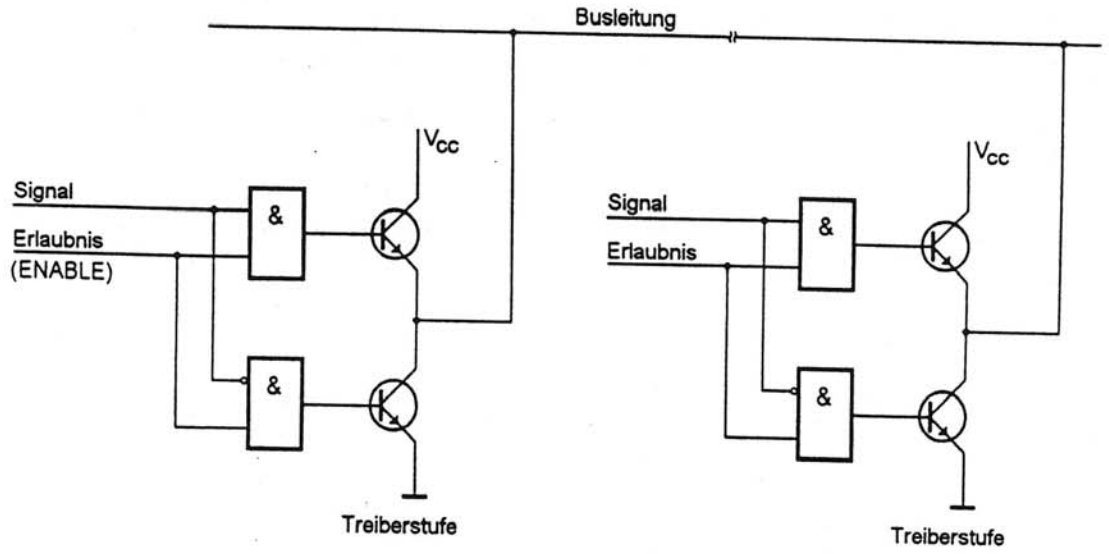
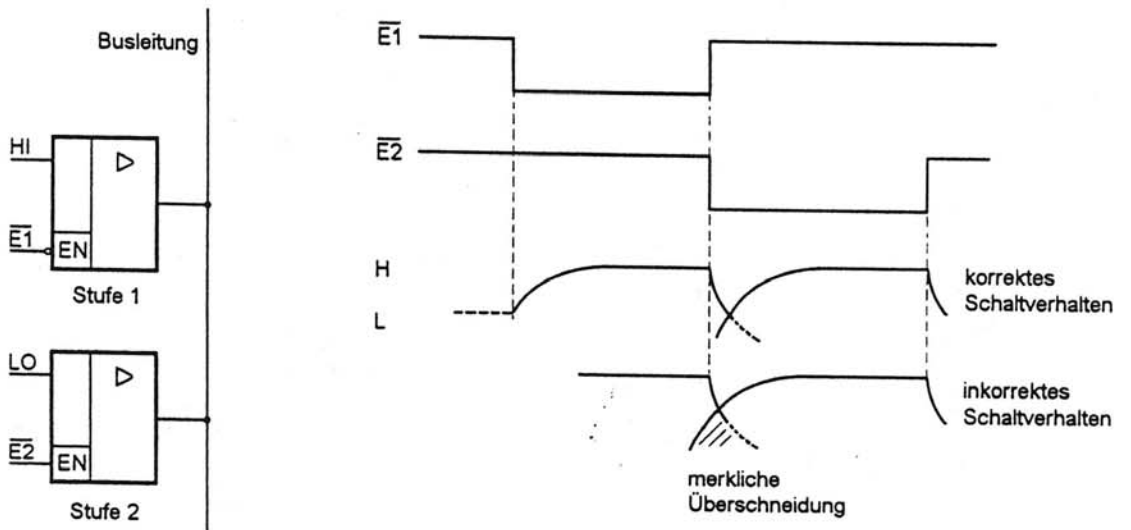


Abbildung 4-5.6  
Tri-State-Prinzip



Korrektes Schaltverhalten:

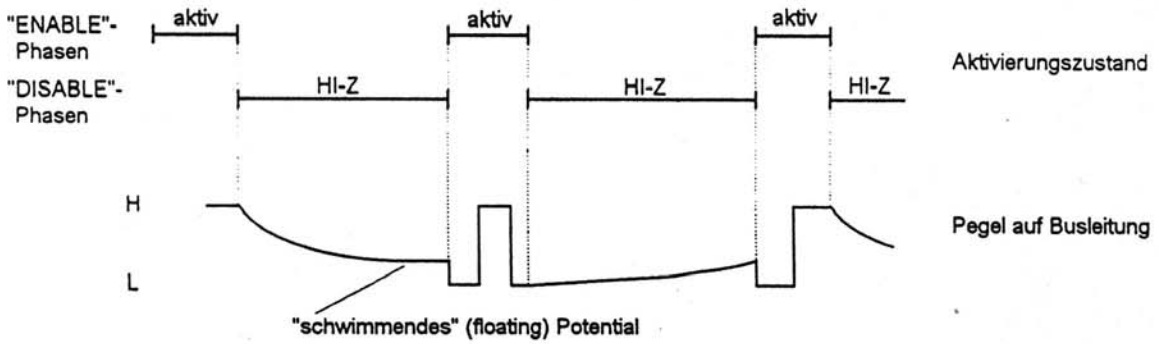
Abschalten (Disable; active to HI-Z) schneller als

Zuschalten (Enable; HI-Z to active)

Abbildung 4-5.7  
Überschneidung beim Umschalten zwischen zwei Busbelegungen

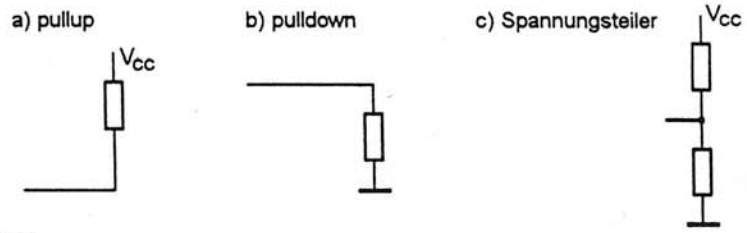
	Open Collector	Tri State
Externe Beschaltung	1 Widerstand je Leitung	nicht erforderlich
Strombedarf	groß (niederohmige Dimensionierung für hohe Geschwindigkeit)	vergleichsweise geringer
Geschwindigkeit	nur HI-LO-Flanke an sich schnell; LO-HI-Flanke durch Widerstand bestimmt	beide Flanken gleich schnell
Umschaltung, Überlappung (Buskonflikte)	Überlappung unproblematisch; zwischen zwei Stufen kann daher überlappend umgeschaltet werden	Überlappung (Contention) ist unbedingt zu vermeiden; Umschaltung zwischen zwei Stufen daher langsamer
Prüfprobleme	inkorrekte Mehrfachaktivierung, Ausgänge (z.B. Steckkontakte) ohne Arbeitswiderstand	inkorrekte Mehrfachaktivierung, schwimmendes Potential

Tabelle 4-5.1  
Open Collector und Tri State in der Gegenüberstellung



**Auswege:**

1. Widerstände



2. Sonderaufschaltung

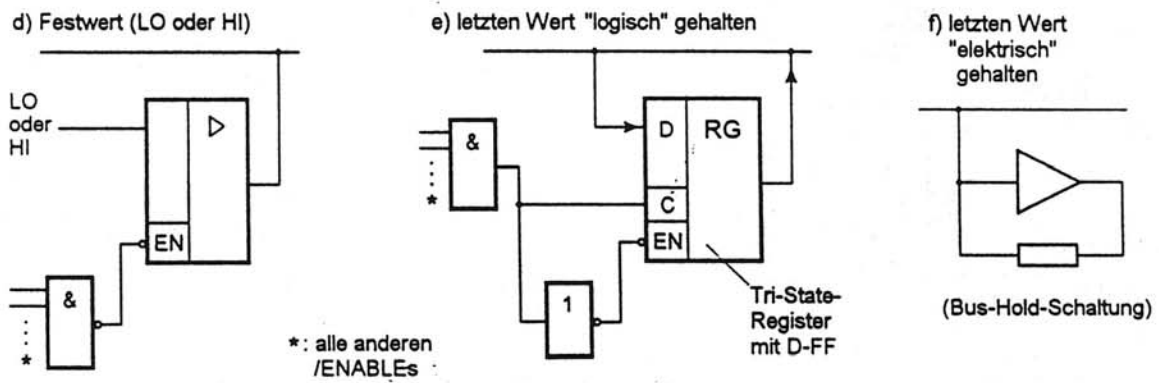


Abbildung 4-5.8  
Der Tri-State-Bus in Ruhe: Problem und Lösungen

# Buskoppelstufen: Einteilung nach

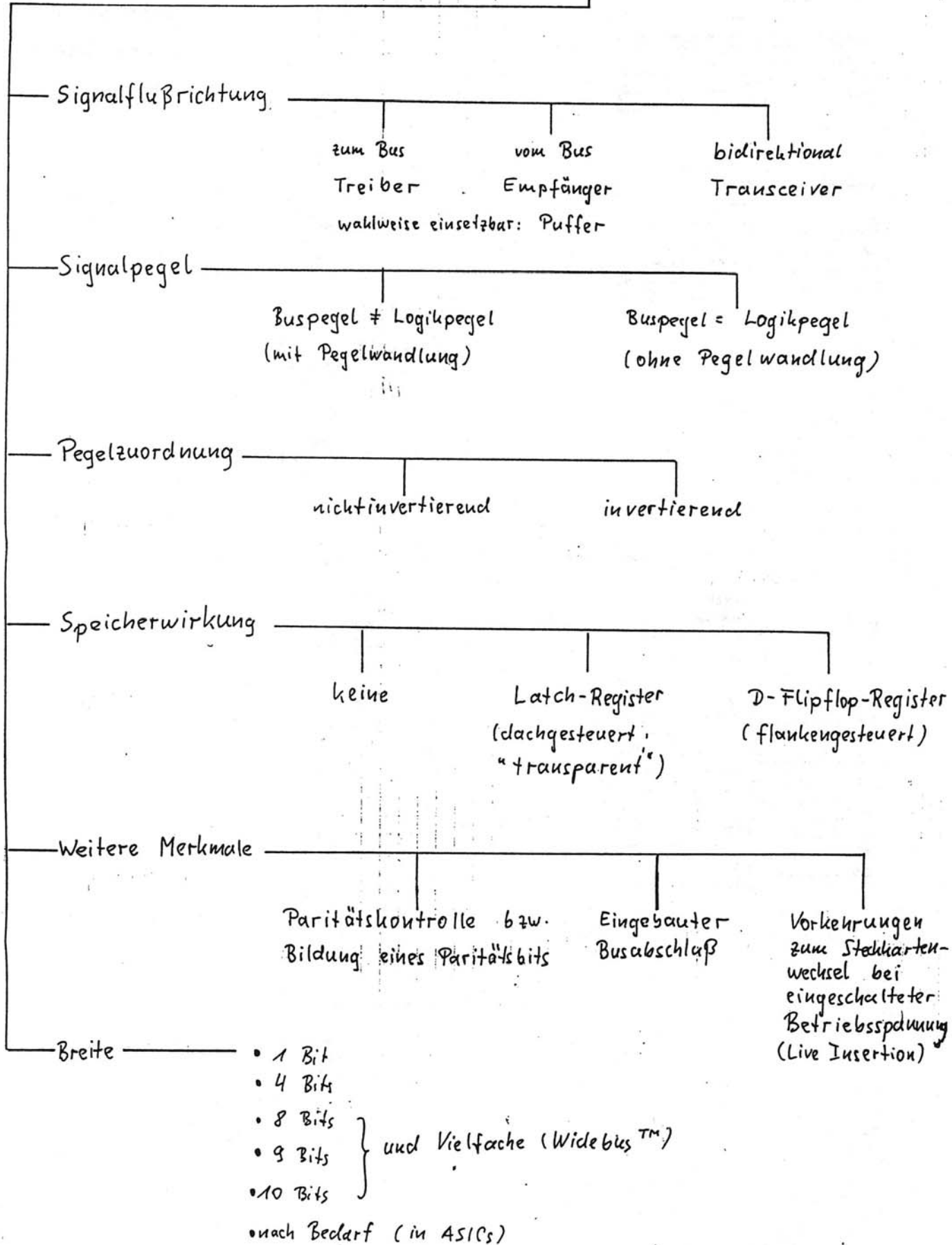
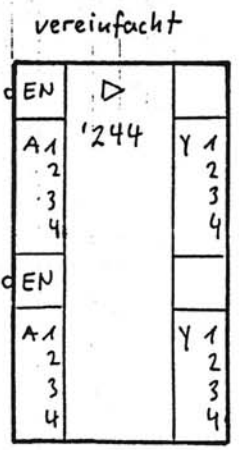
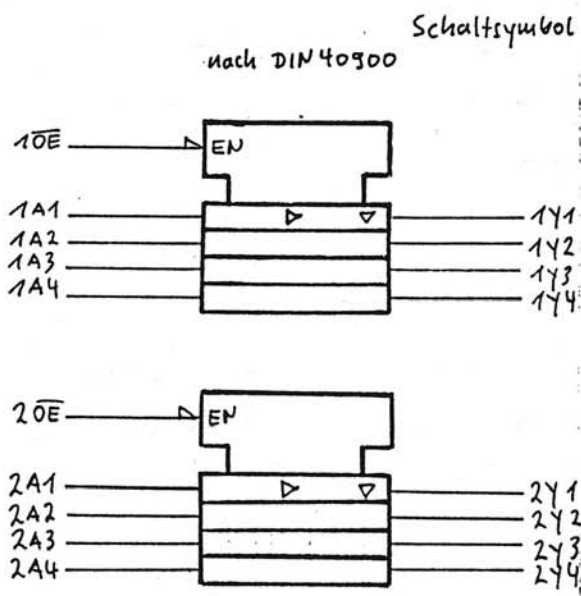
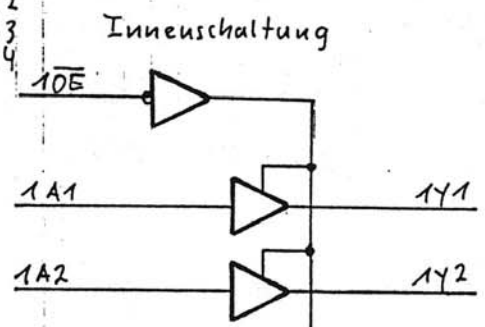


Abbildung 11.5? Dual... ein... (Übersicht)

# Bustreiber (Puffer) '244



Der Schaltkreis enthält zwei Blöcke zu vier Tri-State-Stufen mit gemeinsamem Erlaubnissignal (Enable).



# Transceiver '245

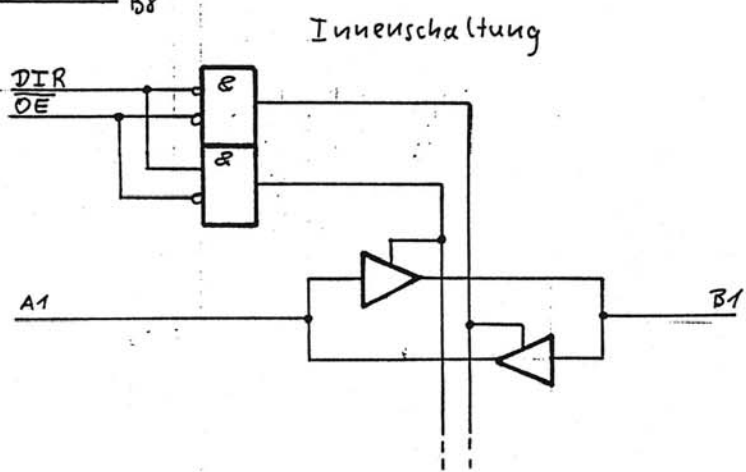
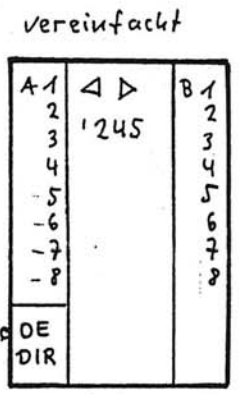
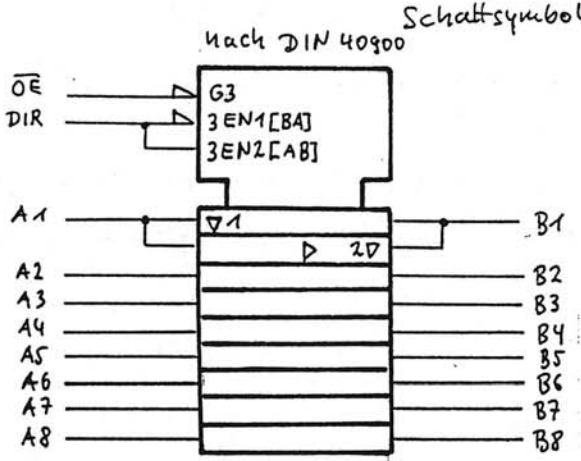
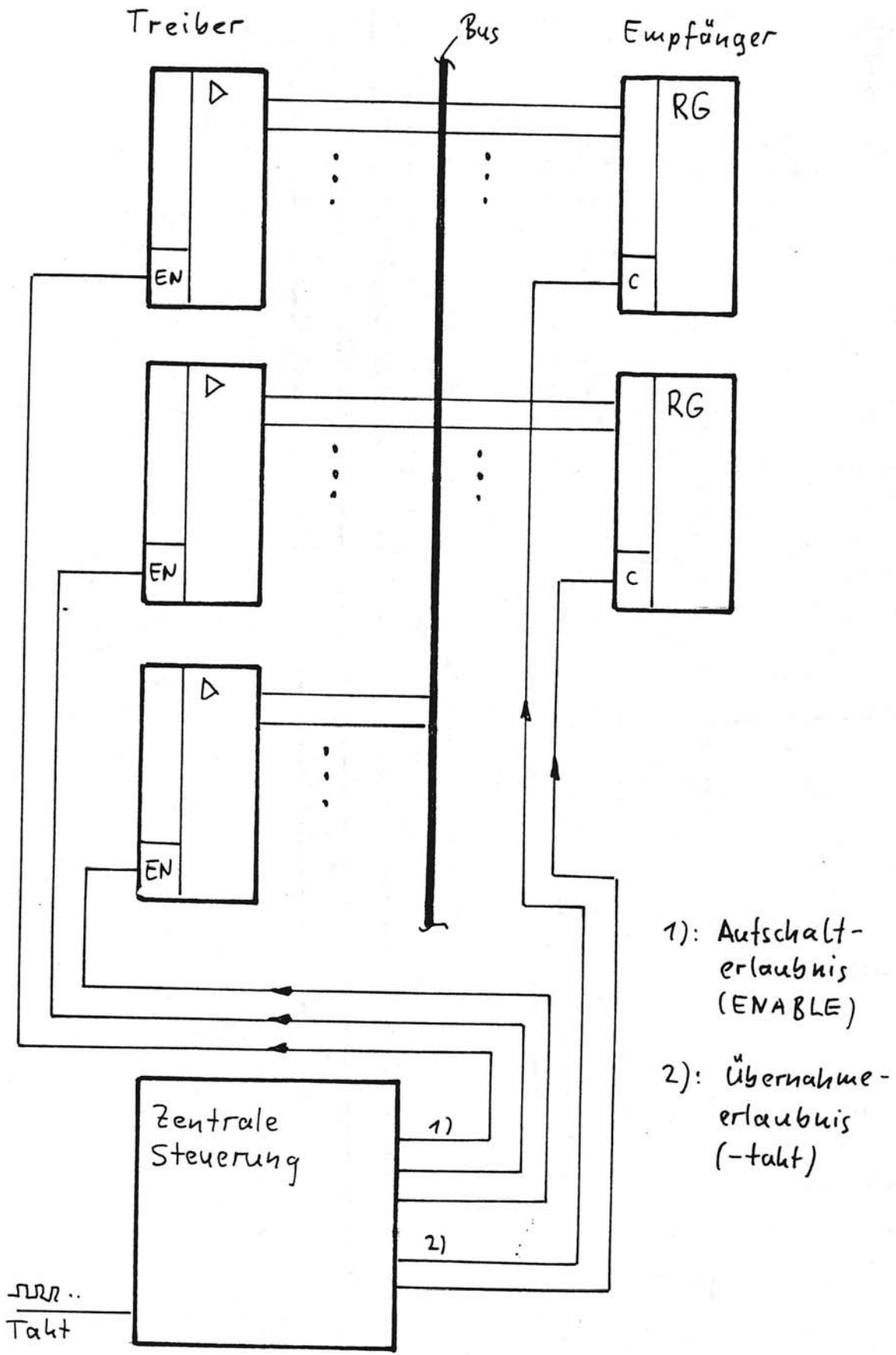


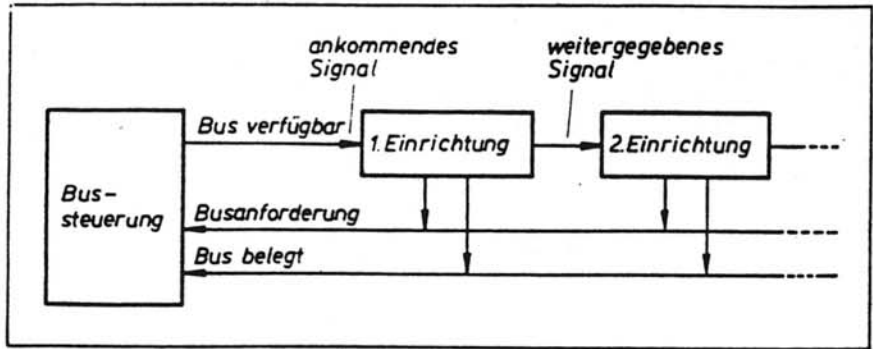
Abbildung 4-5.4 Buskoppelschaltkreise (Beispiele)



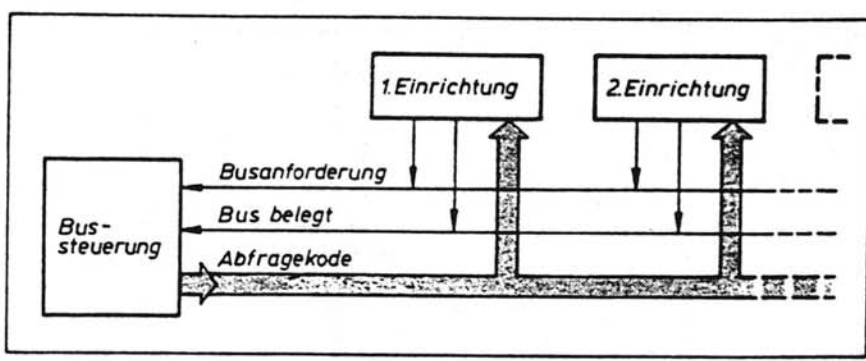
- 1): Aufschalter-  
erlaubnis  
(ENABLE)
- 2): Übernahme-  
erlaubnis  
(-takt)

Abbildung 4-5.15 Zentralgesteuertes Bussystem

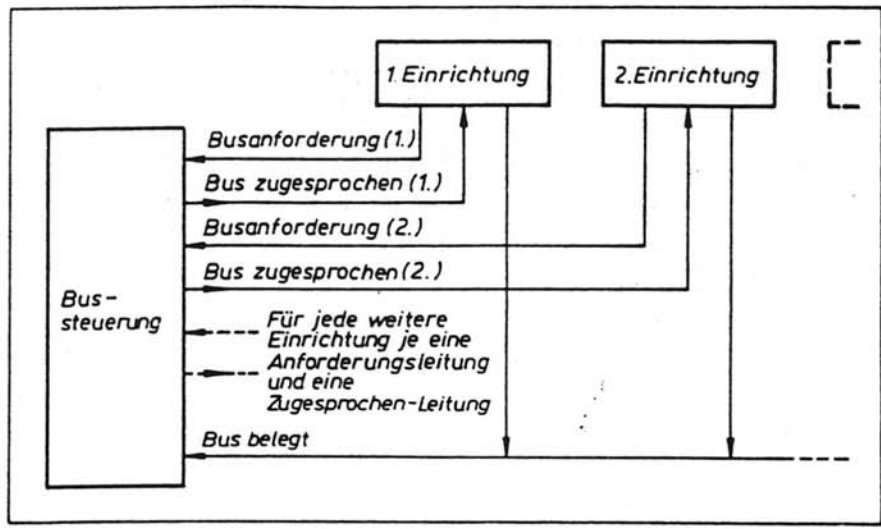




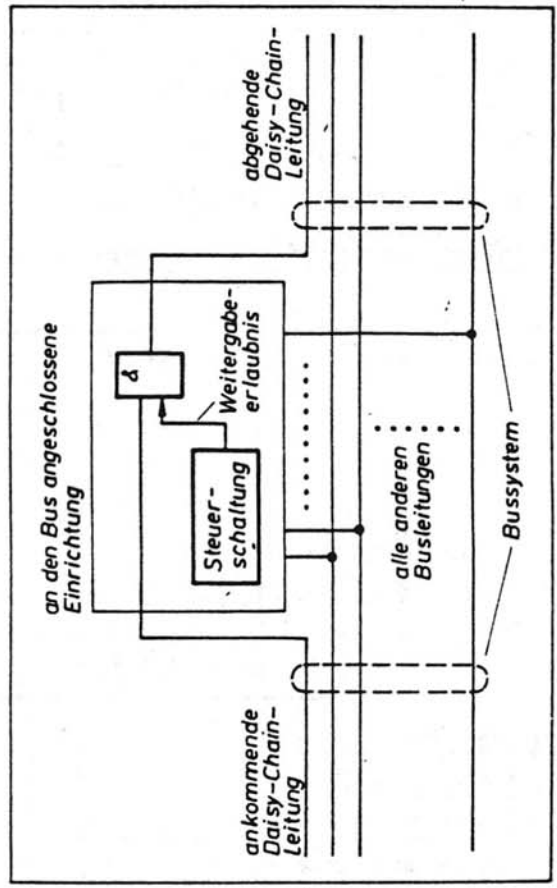
Busvermittlung nach dem Daisy-Chain-Prinzip



Busvermittlung nach dem Abfrageprinzip

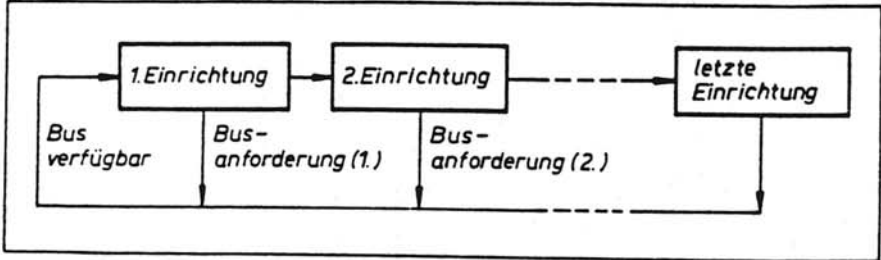


Busvermittlung nach dem Prinzip der unabhängigen Anforderungen

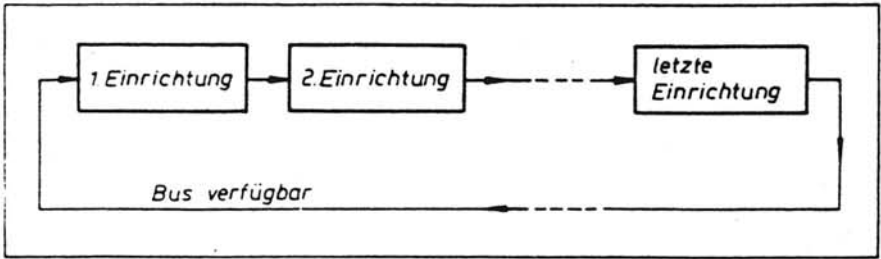


Anschluß von Busleitungen an eine Funktionseinheit

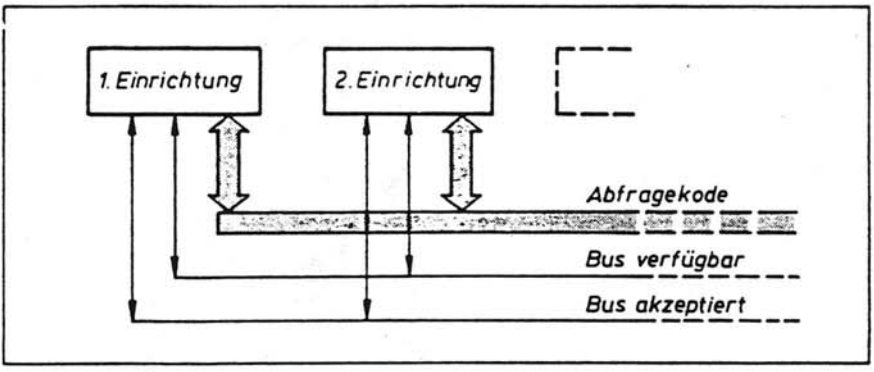




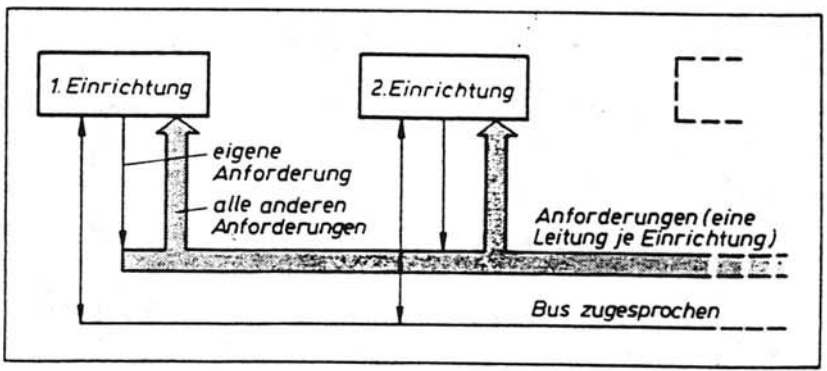
Dezentrale Daisy-Chain-Vermittlung (Pegel des Vermittlungssignals werden ausgewertet)



Dezentrale Daisy-Chain-Vermittlung (Änderungen des Vermittlungssignals werden ausgewertet)



Abfrageprinzip bei dezentraler Bussteuerung (eine der Einrichtungen ist als abfragende Einrichtung konfiguriert)



Prinzip der unabhängigen Anforderungen bei dezentraler Bussteuerung

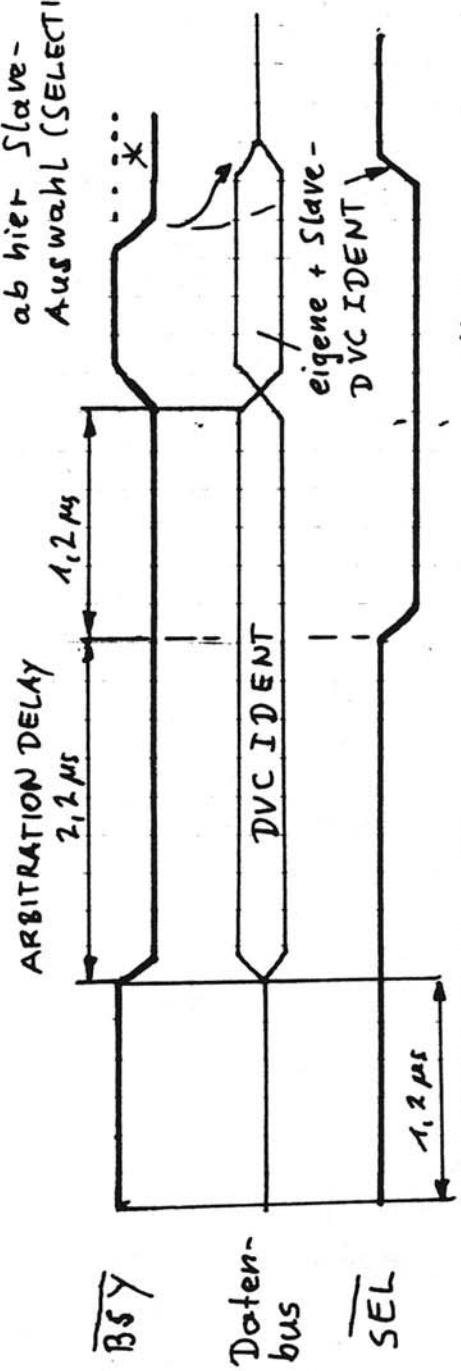
# SCSI

## Bus

### Arbitration

#### - Prinzip -

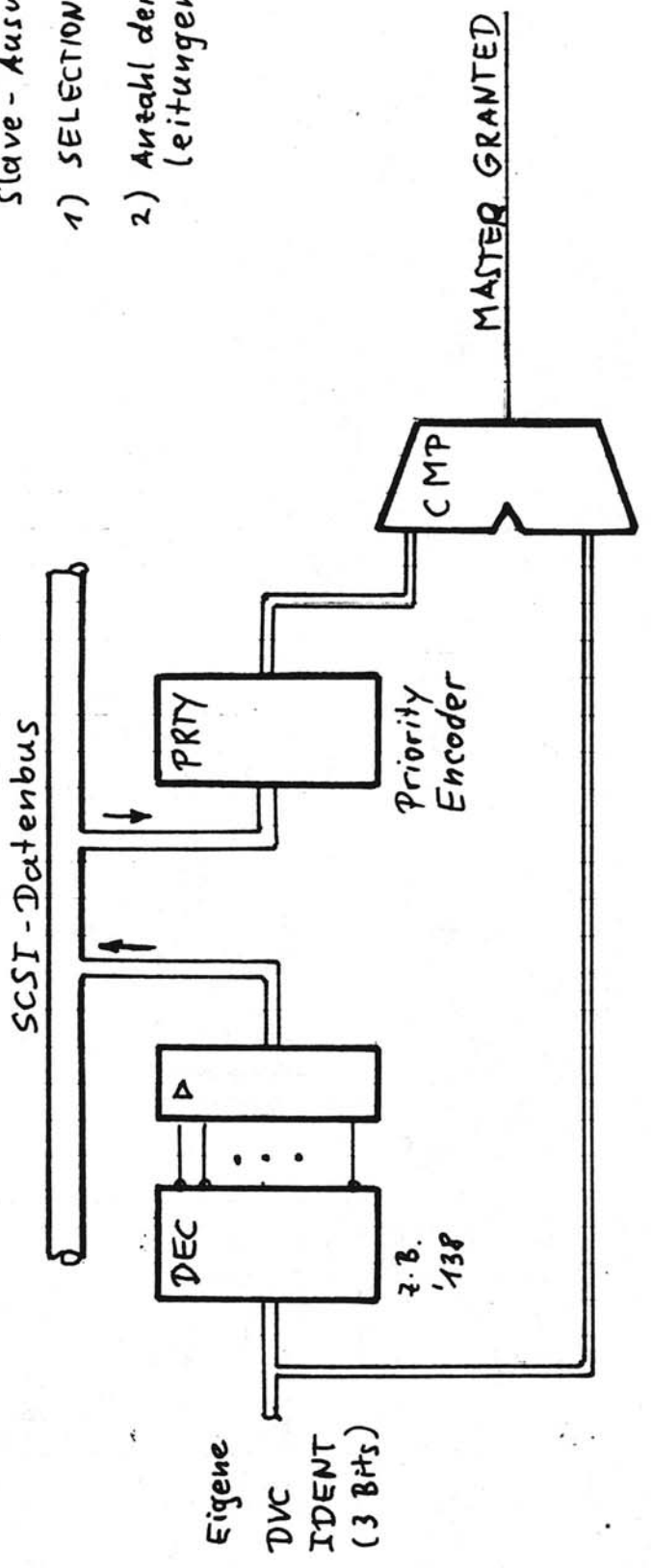
ab hier Slave-Auswahl (SELECTION)



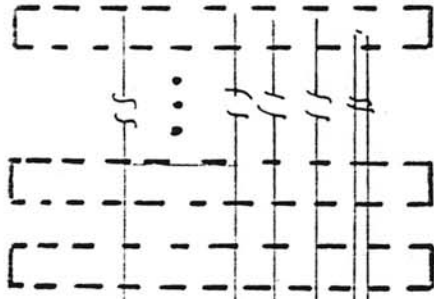
\* Slave antwortet durch Aktivieren von BSY

Fehlerbedingungen bei Slave-Auswahl.

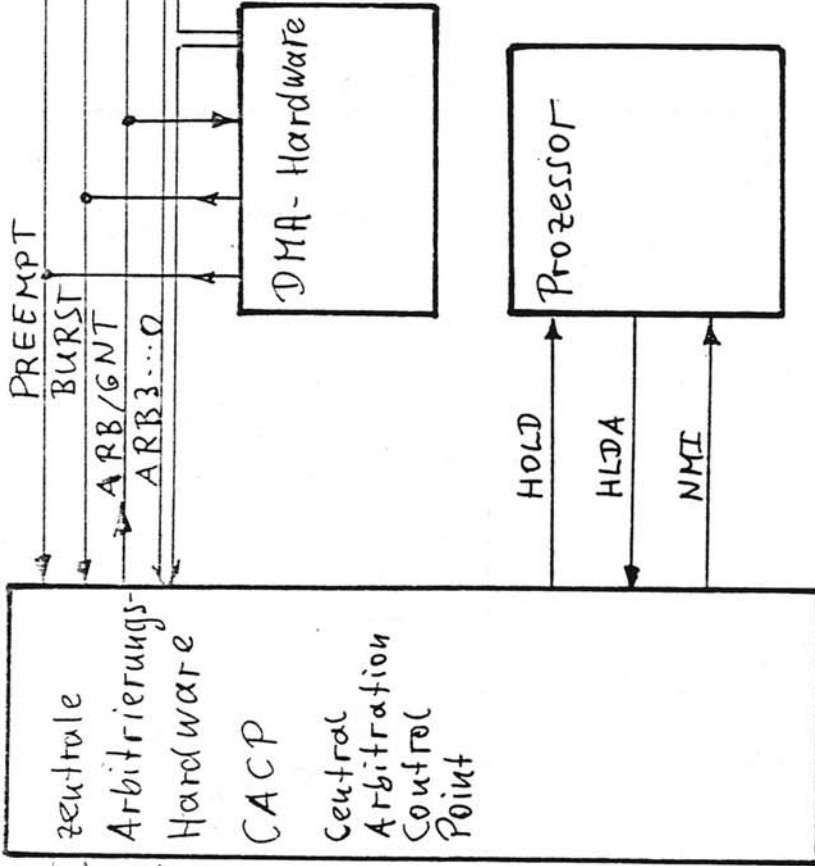
- 1) SELECTION TIMEOUT (250µs)
- 2) Anzahl der aktiven Datenleitungen ≠ 2



MCA Slots



REFRESH



PREEMPT:  
Busanforderung

BURST: keinzeit-  
net Stoßbetrieb  
(s. Abschnitt 5.3.6.)

ARB/GNT: keinzeit-  
net Vermittlungs-  
zyklus

ARB3-0: Prioritätsco-

Abbildung 4-5.25 Die zentrale Arbitrierungshardware des Microchannel

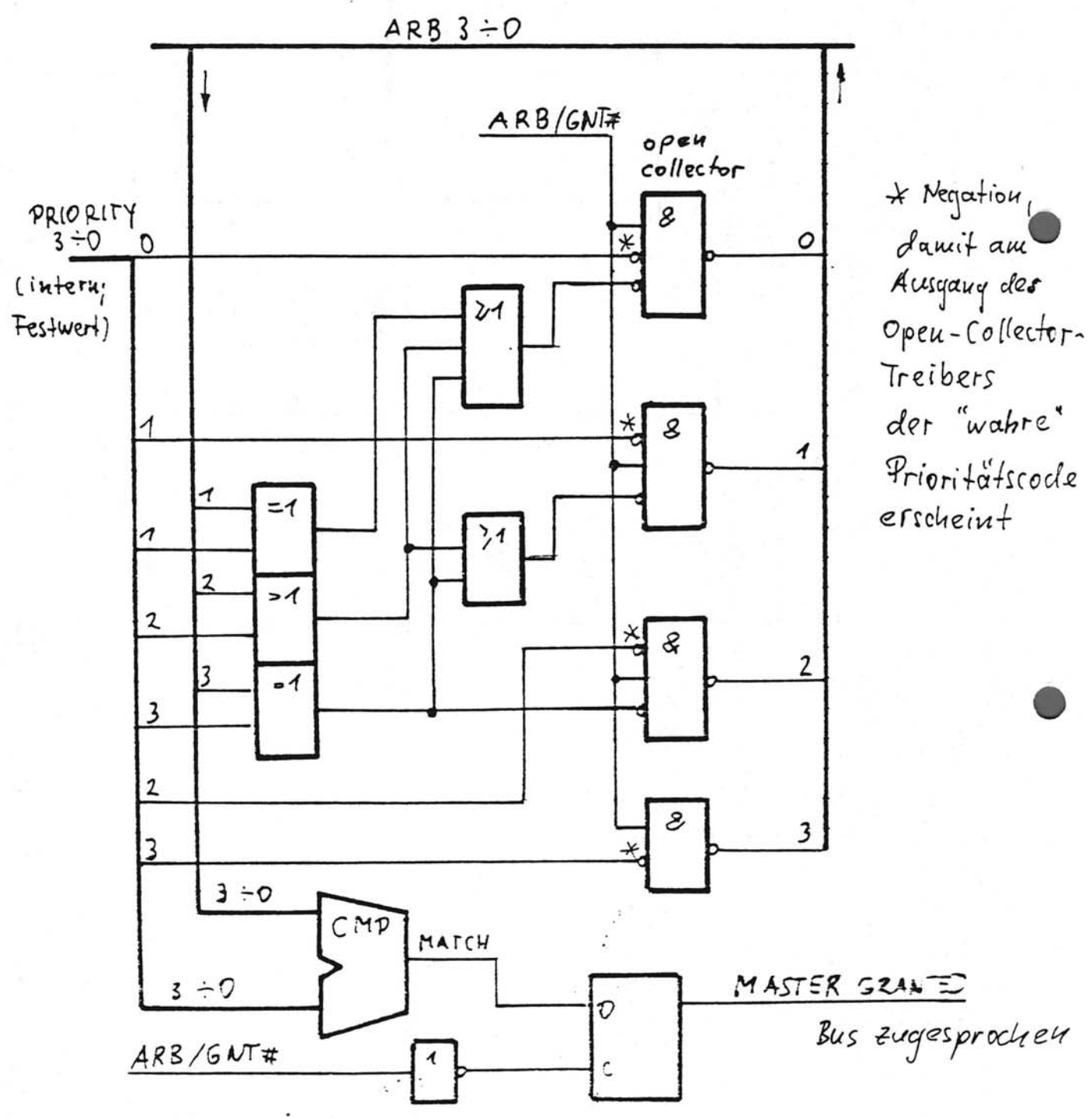
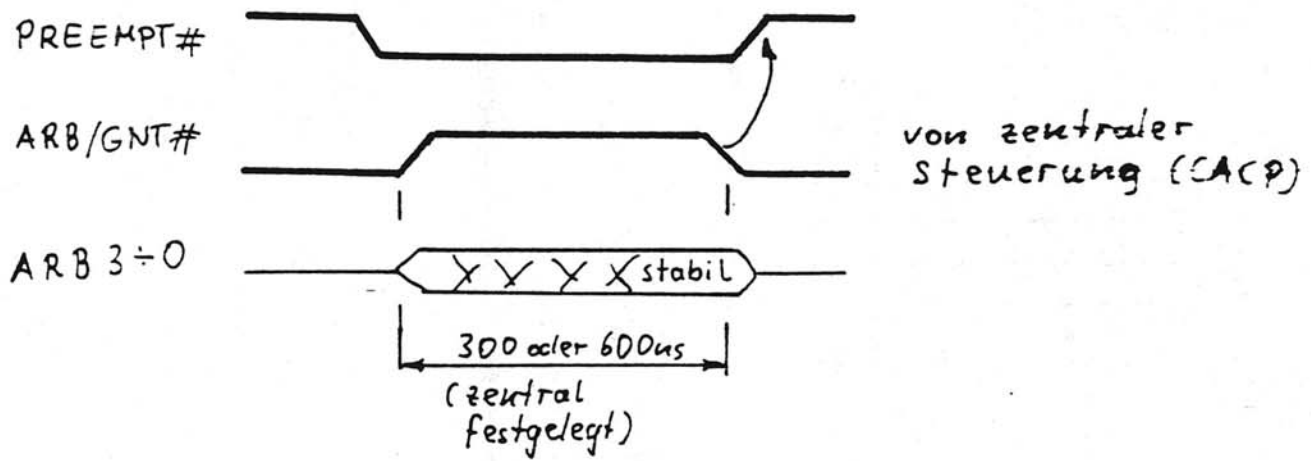


Abbildung 4-5.25 Arbitrierungshardware auf Hochbausteinen (Beispiel) 35

Verfahren ist eine LIN Beispiel für das Einstellen des Prioritätscodes

Einrichtung A

- Priorität 5 -
- treibt 0101
- vergleicht 0001

• Unterschied in Bit 2

- treibt 0111

- vergleicht 0111

• Unterschied in Bit 1; Gleichheit in Bit 2

- treibt 0101

• findet Gleichheit:

Bus zugesprochen

Leitungen ARB3-0

resultierende Mischung  
0001

0111

0101

Einrichtung B

- Priorität 9 -
- treibt 1001
- vergleicht 0001

• Unterschied in Bit 3

- treibt 1111

- vergleicht 0111

• Unterschied in Bit 3

- treibt weiterhin 1111

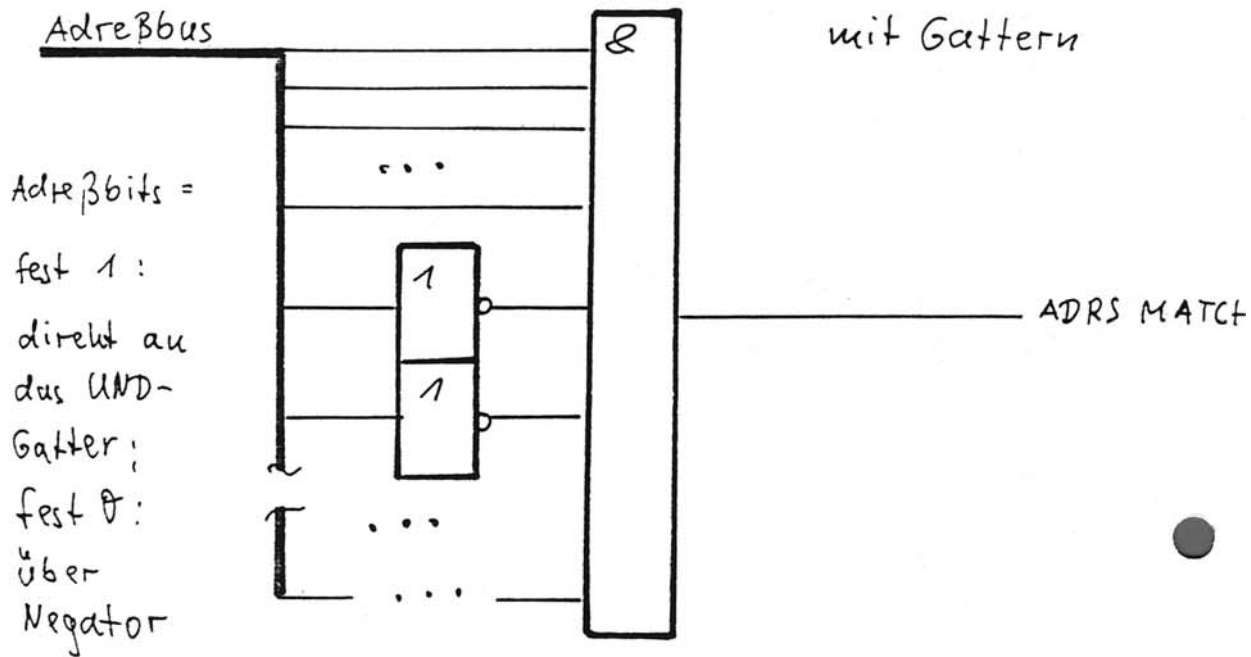
• findet Ungleichheit:

Bus nicht zugesprochen

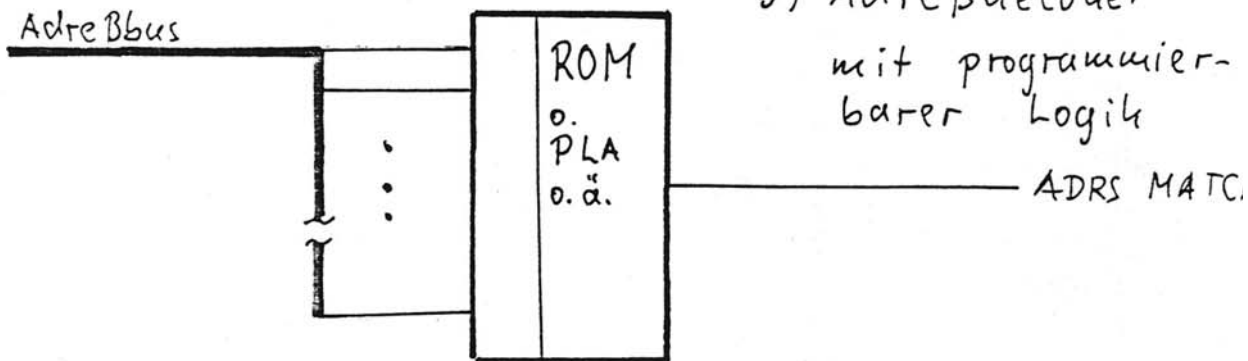
Zeit (300 oder 600 ns) →

# Abbildung 4-5.28 Erkennung der Slave-Adresse<sup>9</sup>

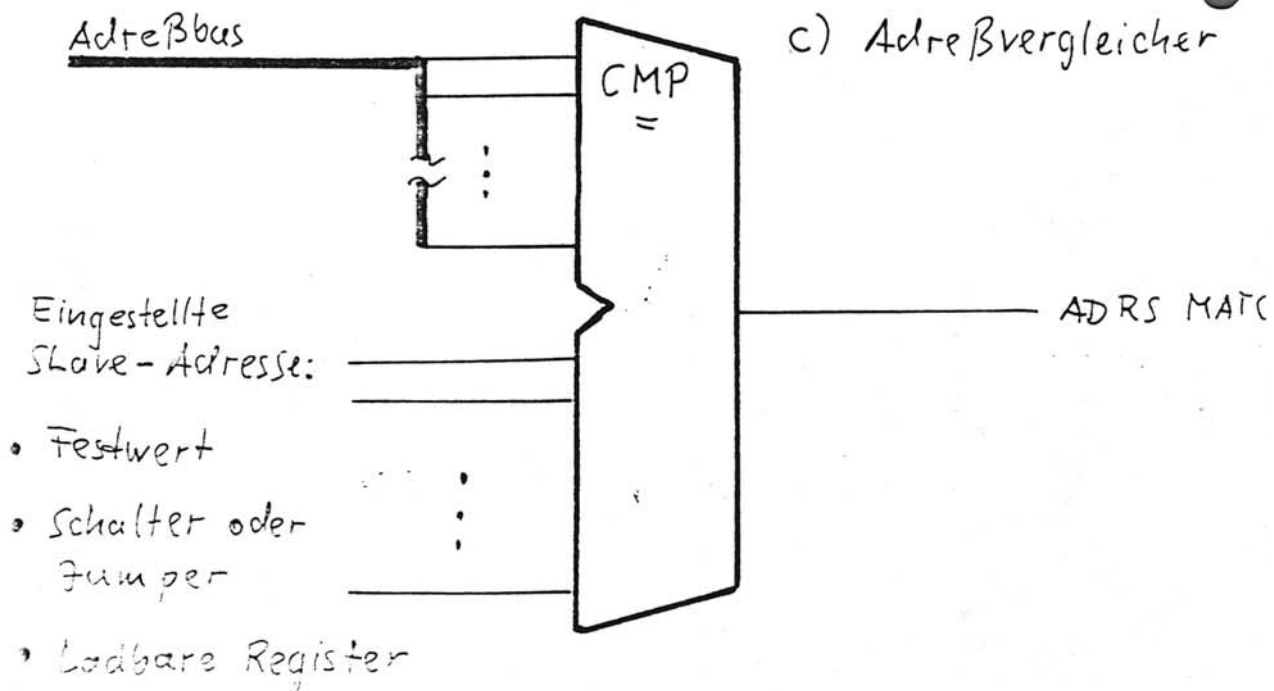
a) Adreßdecoder mit Gattern

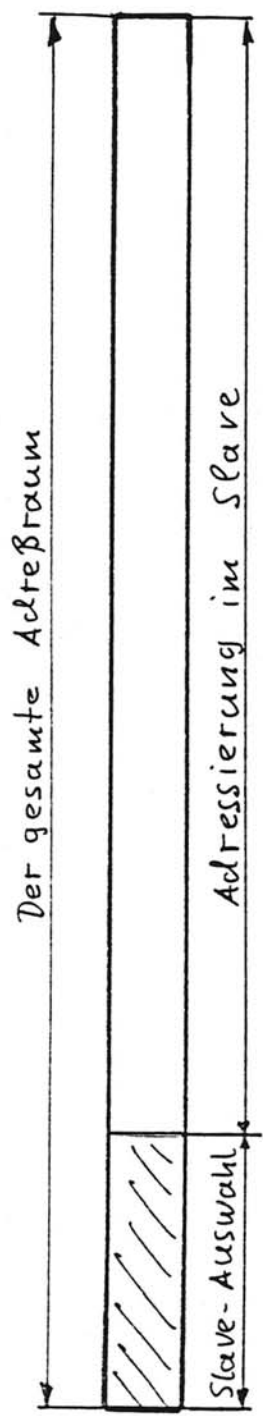


b) Adreßdecoder mit programmierbarer Logik



c) Adreßvergleich

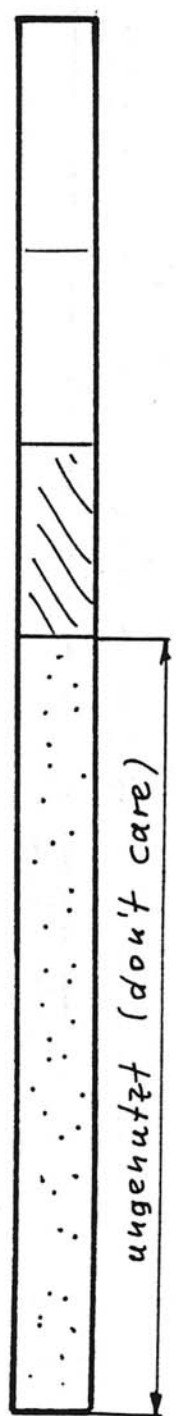




a)



b)



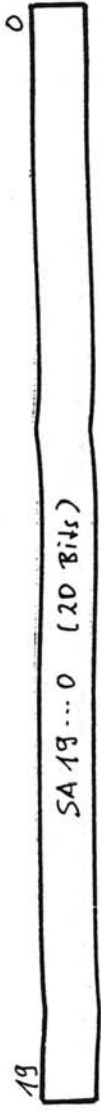
c)



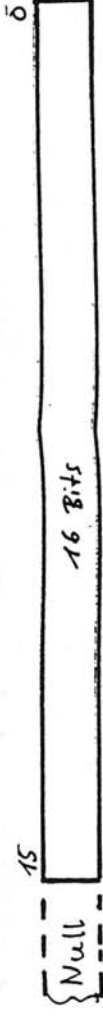
d)

Erläuterungen s. Tabelle 4-5.3

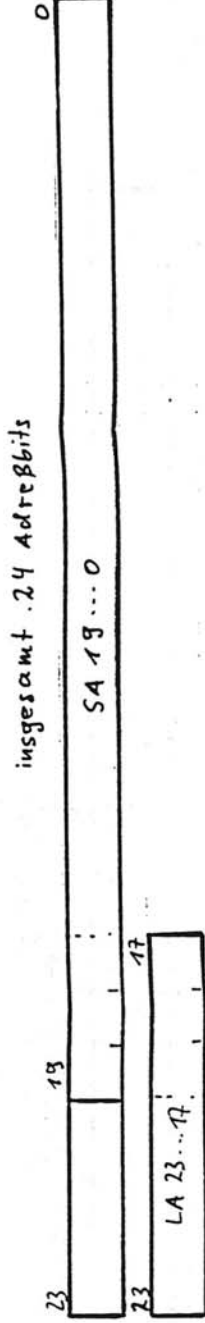
Abbildung 4-5.30 Adreßaufteilungen



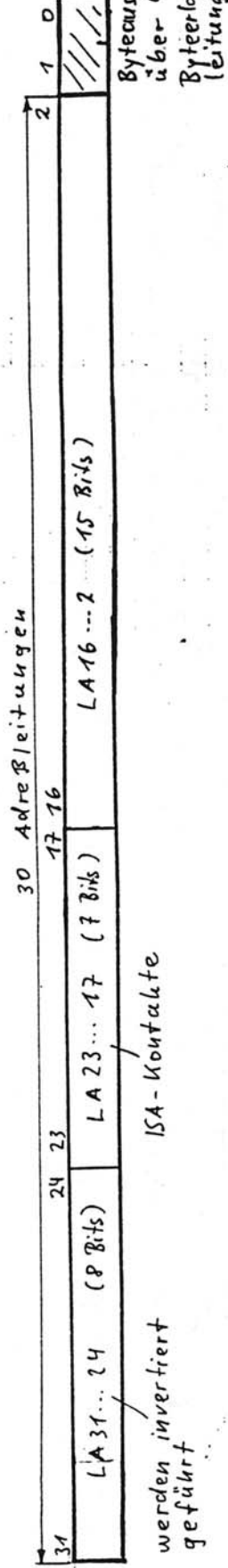
XT-Bus



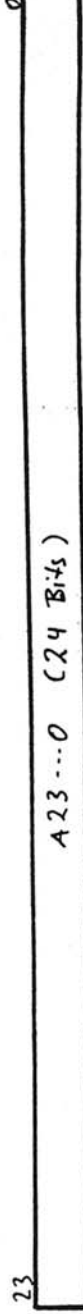
E-A-Adresse



AT-Bus (ISA)

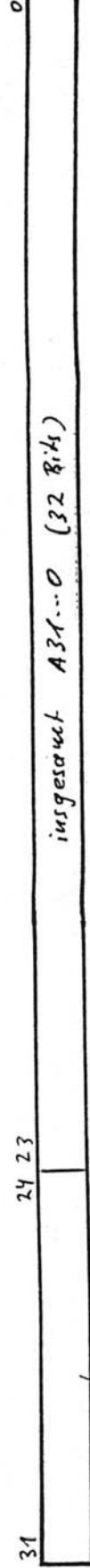


EISA



16-bit-Slots:

MCA



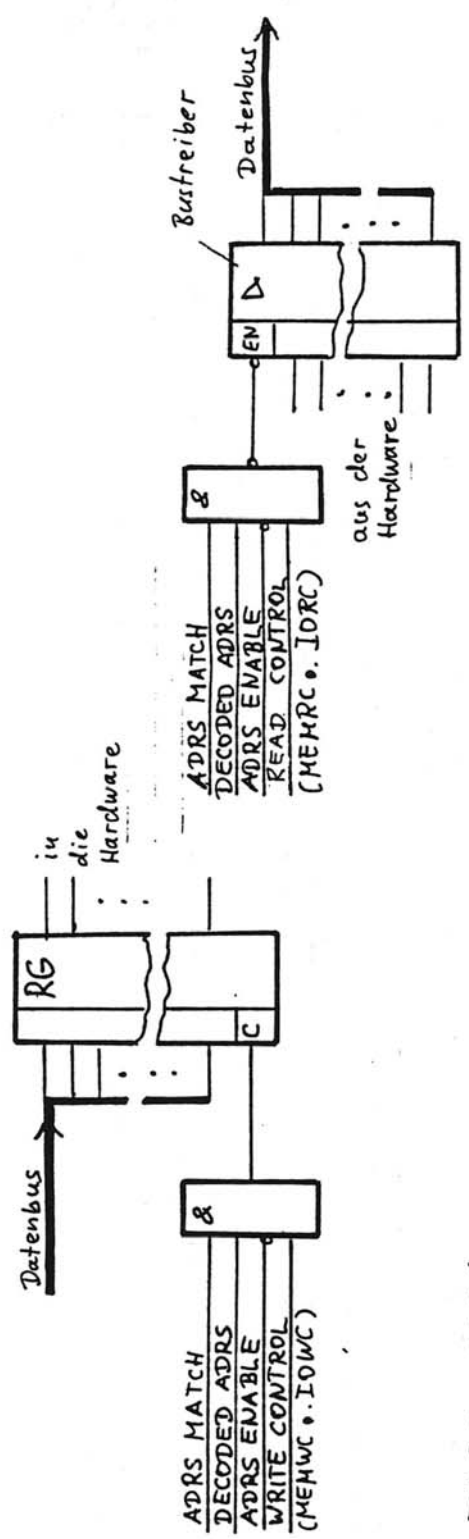
32-bit-Slots:

in erweitertem Stechverbinder

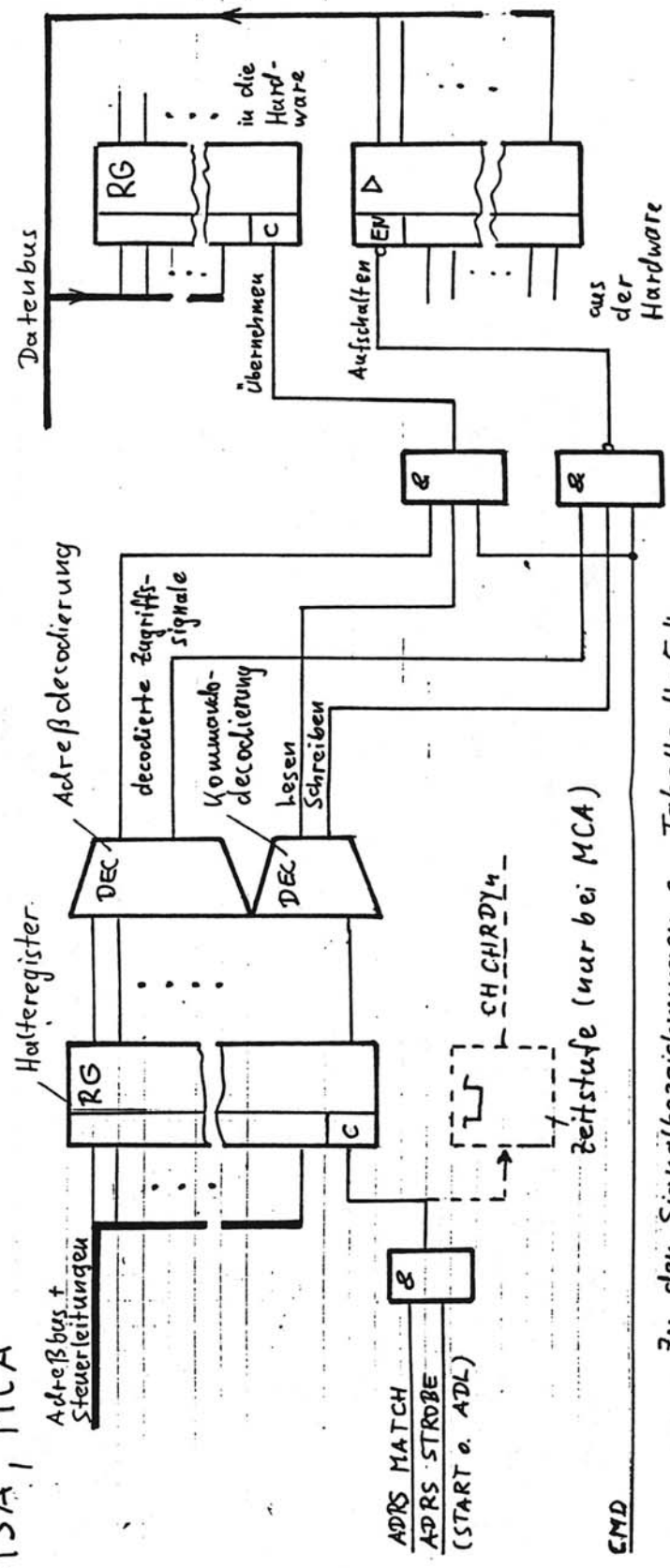
Abbildung 4-5.29 Die Adreßbusbelegungen der PC-Bussysteme



a) ISA

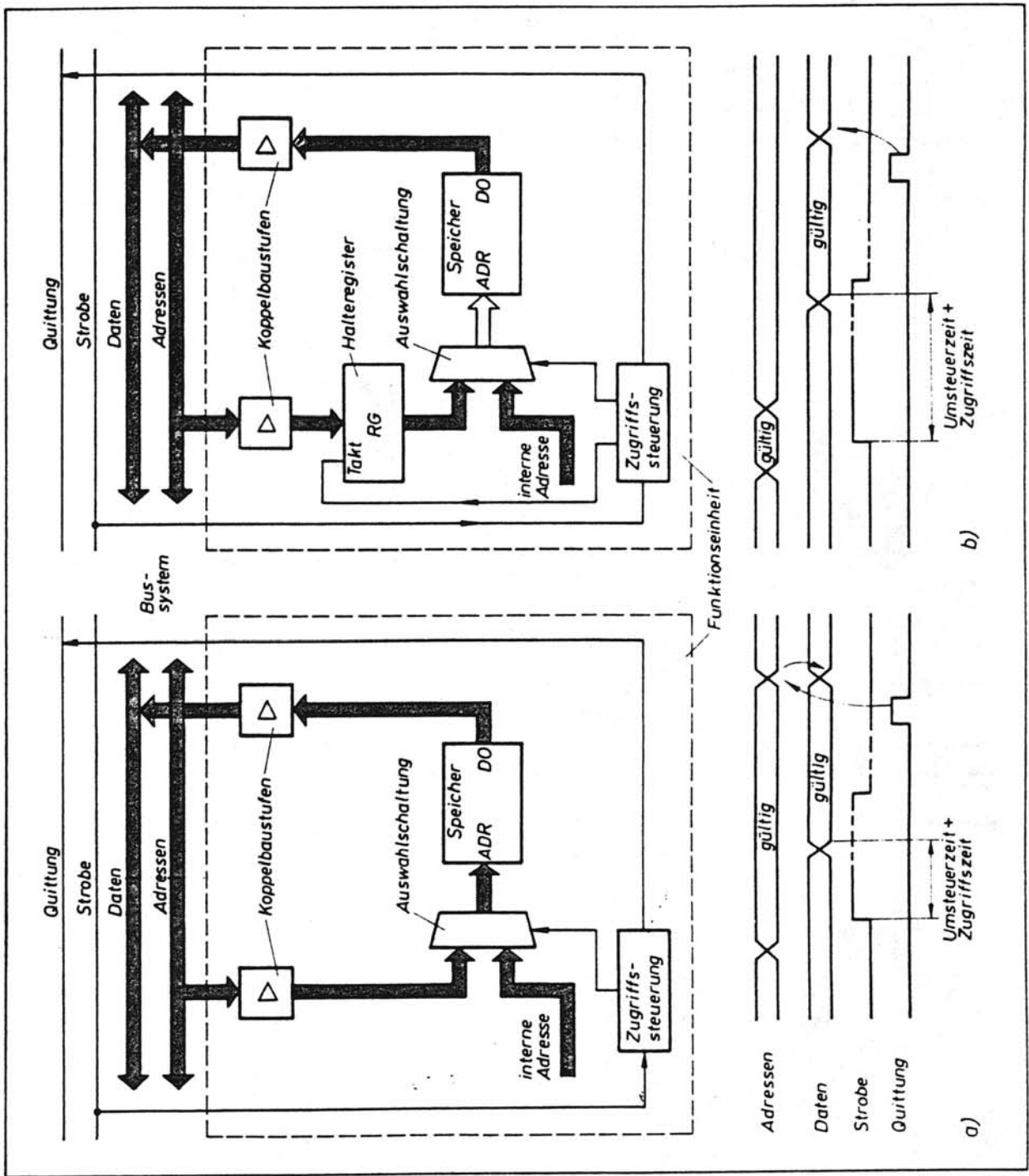


b) EISA, MCA



Zu den Signalbezeichnungen s. Tabelle 4-5.4

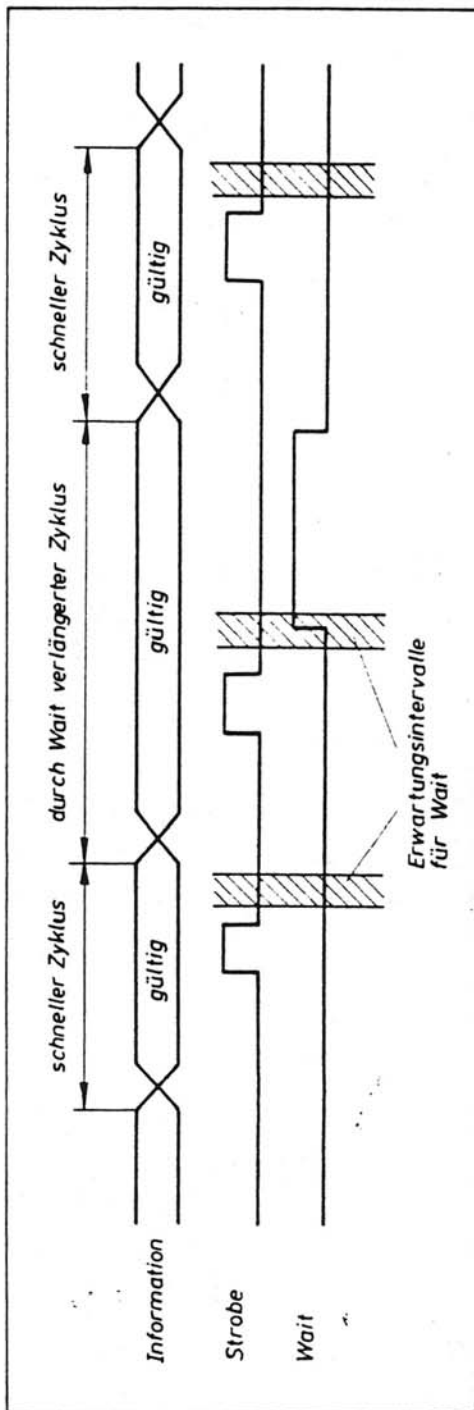
Abbildung 4-5.36 Einfache Slave-Schnittstellen

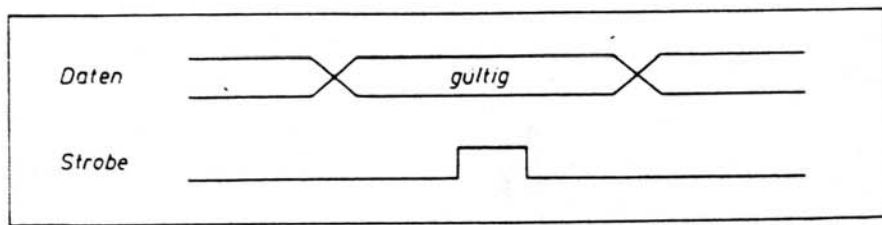


Definition der zeitlichen Verhältnisse für die Informationsübertragung in Abhängigkeit von der Hardware

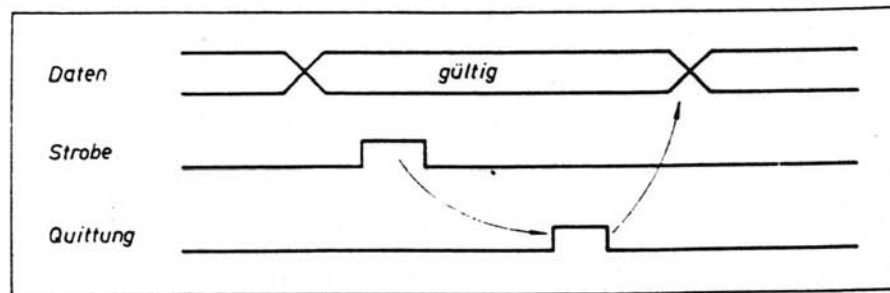
a) Adressen für Speicherzugriff bis zum Eintreffen des Quittungsimpulses gültig; b) Adressen nur im Bereich der Vorderflanke des Strobeimpulses gültig

Informationsübertragung,  
deren Dauer durch ein Waitsignal  
über das nominelle Maß hinaus  
verlängert werden kann

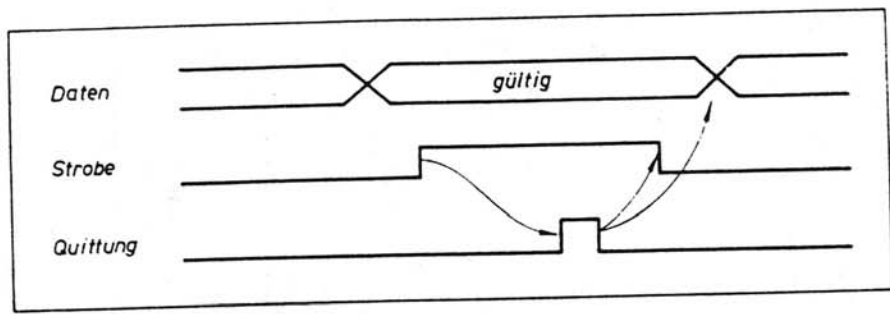




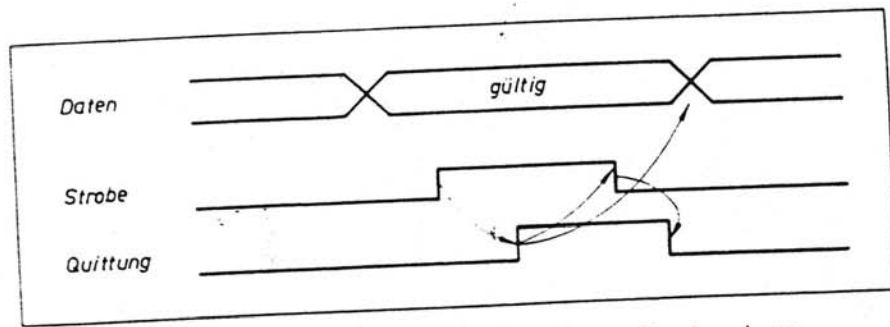
Informationsübertragung mit Strobeimpuls von der sendenden Einrichtung



Informationsübertragung mit Strobeimpuls und Quittungsimpuls



Informationsübertragung mit teilweiser Verriegelung



Informationsübertragung mit vollständiger Verriegelung

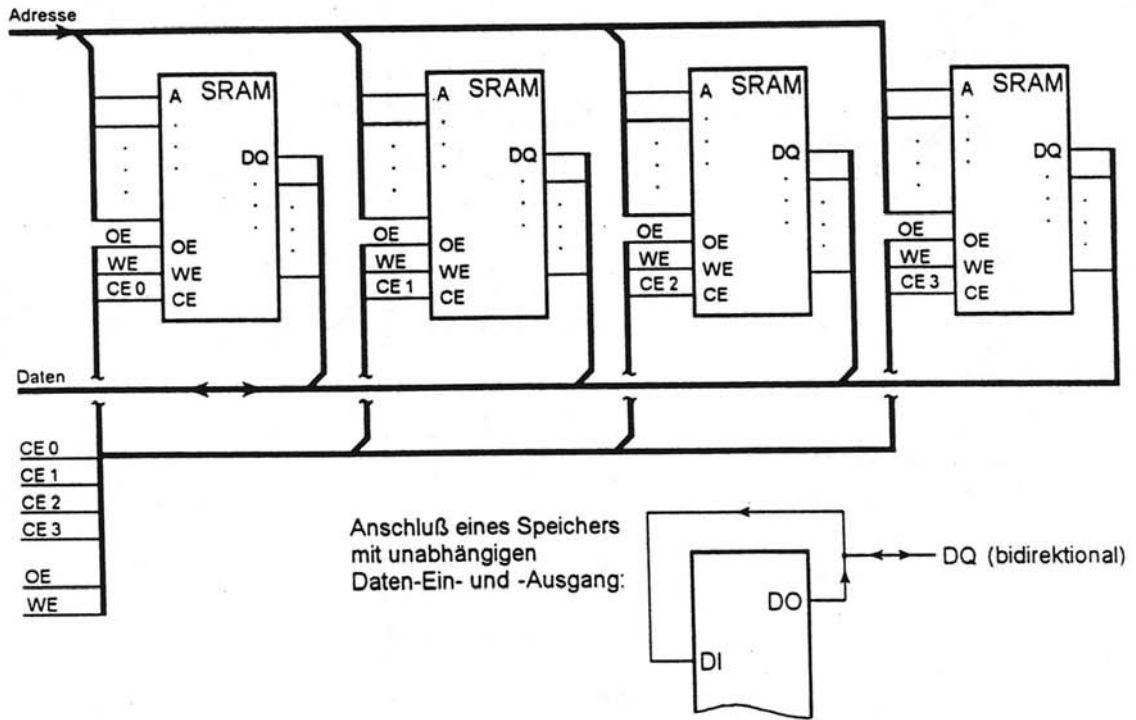


Abbildung 4-4.15  
Einfaches Speichersubsystem mit bidirektionalem Datenbus

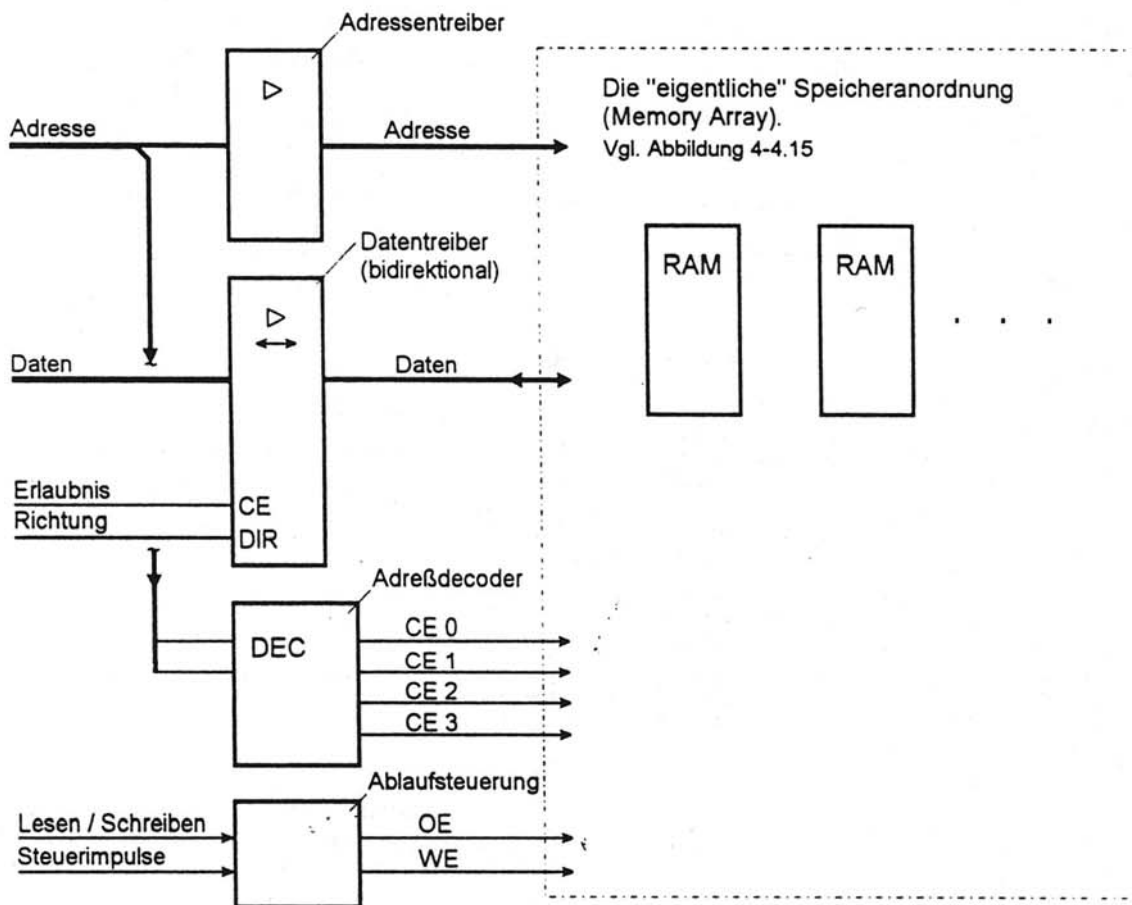


Abbildung 4-4.16  
Ein typisches Speichersubsystem

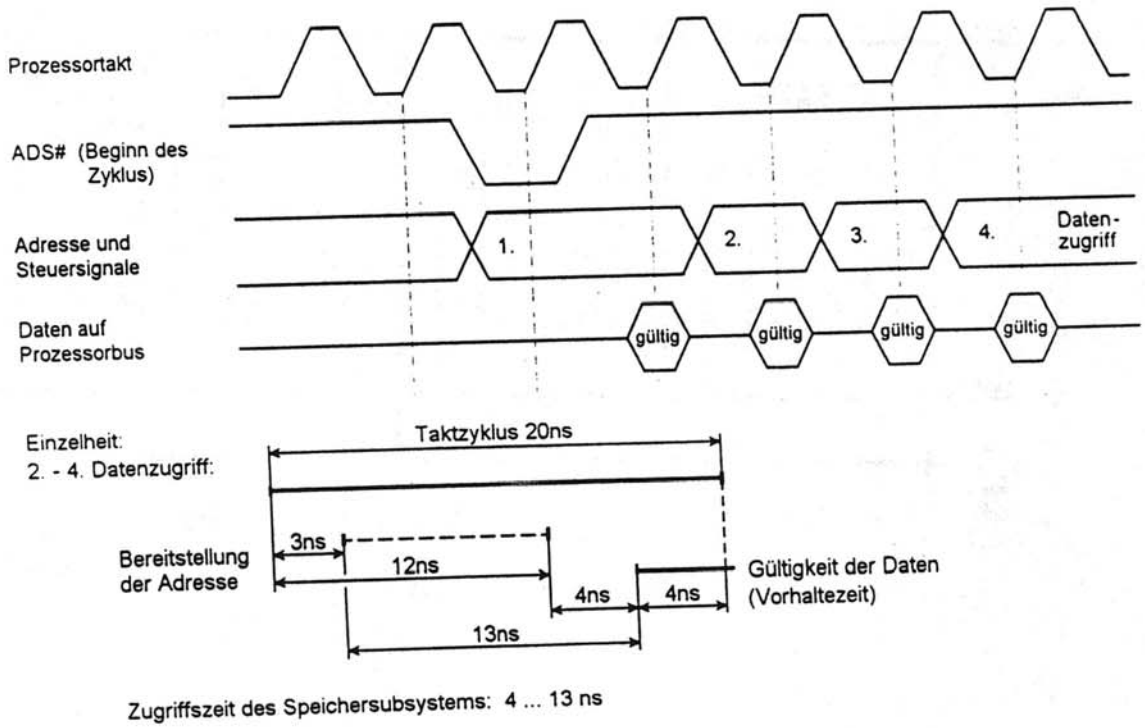
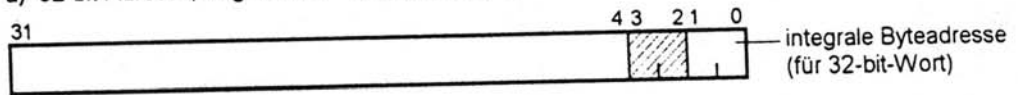


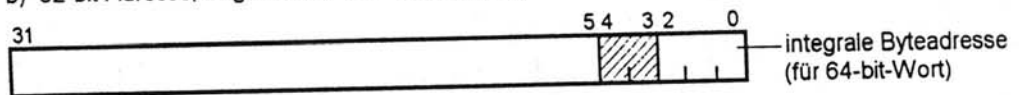
Abbildung 4-4.17  
Datenzugriff im Burst-Mode des i486 DX (50 MHz)

### Adreßaufteilungen

a) 32-bit-Adresse; Zugriff auf 4 \* 32 Bits (i486)



b) 32-bit-Adresse; Zugriff auf 4 \* 64 Bits (Pentium)



Diese Adreßbits werden beim Zählen verändert.

### Zählweisen (i486/Pentium)

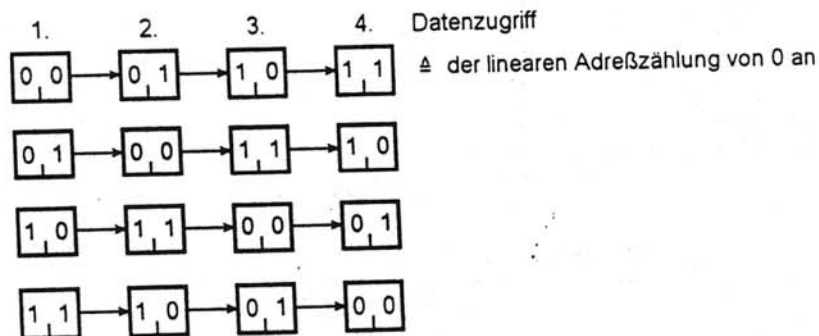


Abbildung 4-4.28  
Adreßzählweisen bei Burst-Übertragungen (Quelle: Intel)

## Die Auslegung der Datenwege

Wir nehmen die Organisationsform der Speicherschaltkreise als gegeben an und betrachten die Strukturen der Datenwege im Speichersubsystem, die sich bei den verschiedenen Kombinationen der Interface-Auslegungen ergeben. Mit "U" bezeichnen wir unabhängige Datenwege für Lese- und Schreibzugriffe, mit "B" bidirektionale Datenwege. Es ergeben sich somit 4 Kombinationen (im folgenden kennzeichnet der erste Buchstabe jeweils das System-Interface, der zweite das Interface der Speicherschaltkreise).

**Hinweis:** Die Abbildungen 4-4.18...4.21 zeigen die jeweils grundsätzliche Struktur am Beispiel eines Datenbits. Denken Sie bitte daran, daß (1) die Datenausgänge mehrerer RAM-Schaltkreise zusammenschaltet werden können, falls eine höhere Speicherkapazität vorzusehen ist, und daß (2) über einen einzelnen Treiberschaltkreis üblicherweise mehrere Datenbits geführt werden (8, 9, 16, 18 usw.).

### 1. Der Fall U-U

Besondere Wandlungen sind nicht notwendig; bei Bedarf (zu treibende Kapazitäten) sind lediglich Treiberstufen vorzusehen (Abbildung 4-4.18).

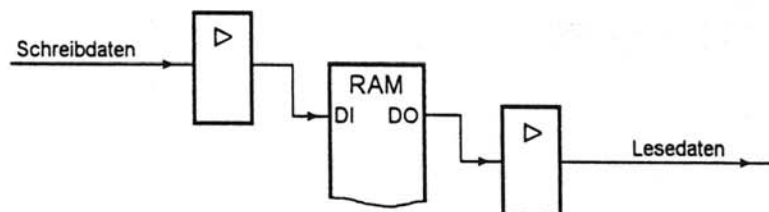


Abbildung 4-4.18  
Datenwege im Speichersubsystem: der Fall U-U

## 2. Der Fall B-B

Wir brauchen bidirektionale Treiberstufen (Abbildung 4-4.19), die erforderlichenfalls passend anzusteuern sind, um Bus Contention (Mehrfachaufschaltung) zu vermeiden. Prinzip: "von außen" darf erst aufgeschaltet werden, wenn sich die Datenausgänge der Speicherschaltkreise im HI-Z-Zustand befinden.

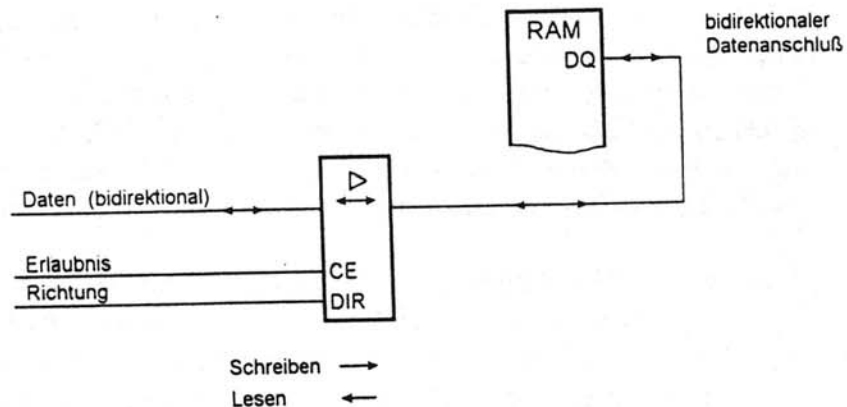


Abbildung 4-4.19  
Datenwege im Speichersubsystem: der Fall B-B

## 3. Der Fall B-U

Viele Speicherschaltkreise lassen sich direkt anschließen; bei Bedarf sind Treiberstufen vorzusehen (Abbildung 4-4.20). Es liegt nahe, DI und DO so miteinander zu verbinden, daß auch speicherseitig ein bidirektionaler Datenweg entsteht (s. Abbildung 4-4.14). Damit haben wir das Problem auf Fall 2. zurückgeführt. Aber Achtung: die Datenausgänge der Speicher müssen nun größere Lastkapazitäten treiben (nämlich auch ihre eigenen Dateneingänge).

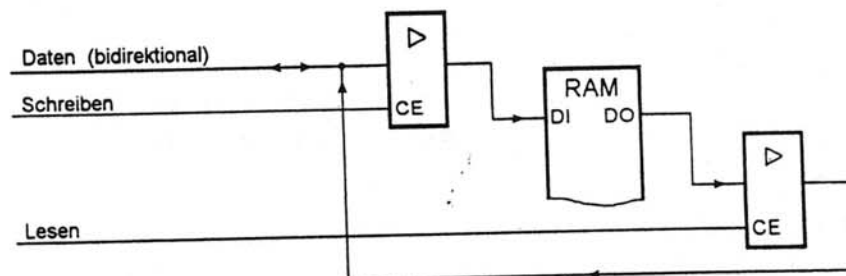


Abbildung 4-4.20  
Datenwege im Speichersubsystem: der Fall B-U



#### 4. Der Fall U-B

Man braucht zwei Treiberstufen, um beide Wege beschalten zu können (Abbildung 4-4.21). Ist eine solche Lösung - beim heutigen Schaltkreissortiment (vgl. Tabelle 4-4.2) - noch zeitgemäß? - Ein Szenarium für die Wahl: es ist ein Speichersubsystem für 16 Datenbits aufzubauen. Wählen Sie 16 Speicherschaltkreise in "\* 1"-Organisation (die womöglich auch noch mit Treibern zu beschalten sind) oder einen in "\* 16"-Organisation und dazu zwei passende Treiberstufen, also insgesamt 3 Gehäuse, verglichen mit 16 und mehr?

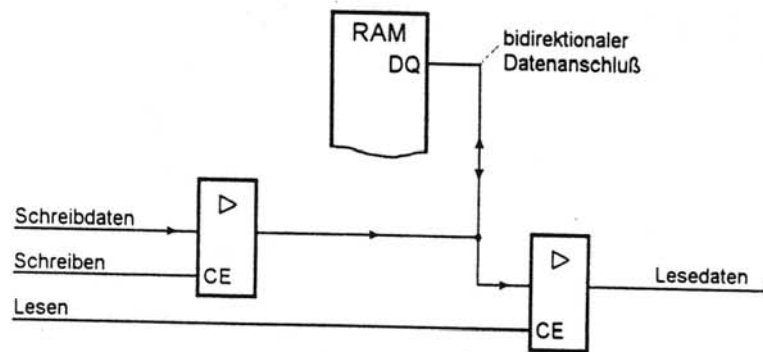


Abbildung 4-4.21  
Datenwege im Speichersubsystem: der Fall U-B

### Die Adressierung

Ist die gesamte Speicherkapazität nicht in einem Schaltkreis oder in mehreren gleichzeitig angesteuerten Schaltkreisen untergebracht, ist die Adresse aufzuteilen: ein Teil ist auf die Adreßeingänge der Schaltkreise zu führen (erforderlichenfalls unter Zwischenschaltung von Treiberstufen), ein anderer Teil ist zu **decodieren**, um jeweils einen Speicherschaltkreis oder eine Gruppe zusammengehörender Schaltkreise auszuwählen.

#### Die "naive" Adreßaufteilung

Gemäß Abbildung 4-4.22 werden die niederwertigen Adreßbits den Speicherschaltkreisen zugeführt, die höherwertigen werden decodiert. Der Decoder steuert dann die CE-Eingänge an. Die Vorteile: (1) Einfachheit, (2) Stromersparnis (namentlich bei modernen Schaltkreisen, deren Leistungsaufnahme sich deutlich vermindert, wenn CE inaktiv ist; Stichwort: Standby Mode).

Beispiel: 1 MBytes; verwirklicht mit 8 SRAMs  $128\text{ k} \cdot 8$   
 Adreßaufteilung:

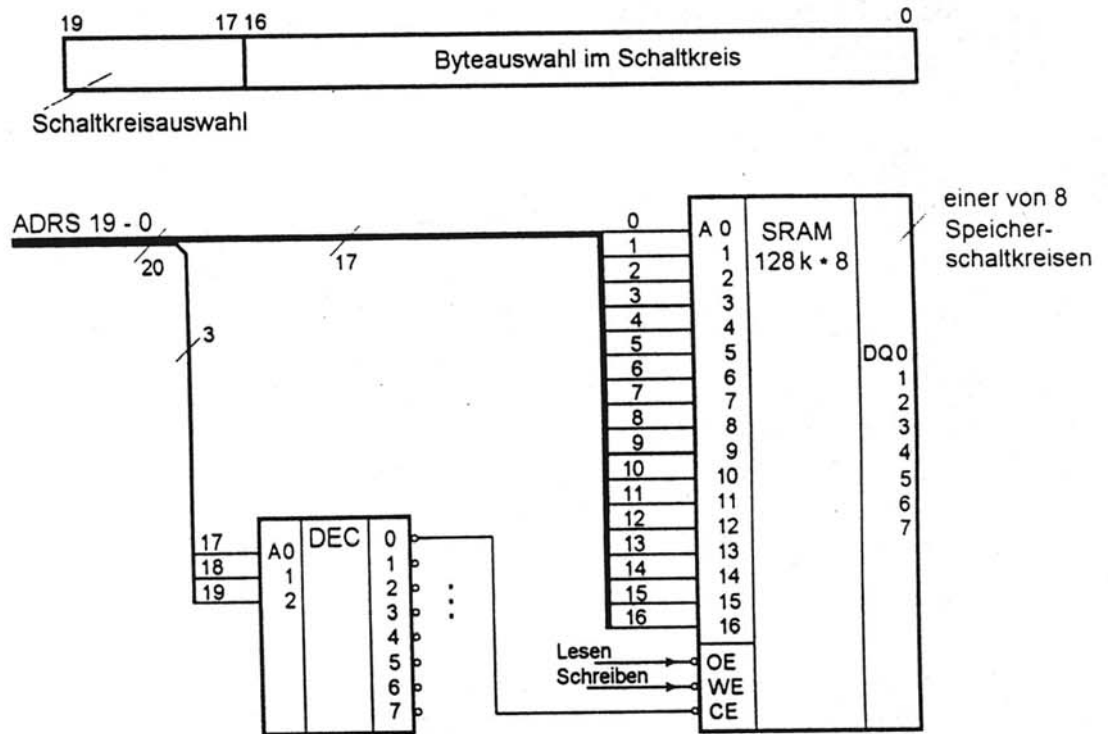


Abbildung 4-4.22  
 "Naive" Adreßaufteilung und -decodierung

### "Schnelle" Adreßaufteilungen

Bei den meisten Speicherschaltkreisen, die eine zusätzliche Aufschalterlaubnis (OE) haben, ist die auf OE bezogene Zugriffszeit kürzer als die auf CE bezogene. Man kann also einige ns mehr "herauskitzeln", indem man beim Lesen die CE-Eingänge aller Schaltkreise fest ansteuert und den Adreßdecoder auf die OE-Eingänge wirken läßt (Abbildung 4-4.23). Beim Schreiben muß man dann allerdings die CE-Eingänge auch vom Decoder aus aktivieren lassen, oder man muß die WE-Eingänge entsprechend selektiv ansteuern (dies ist zumeist die "schnellere" Lösung).

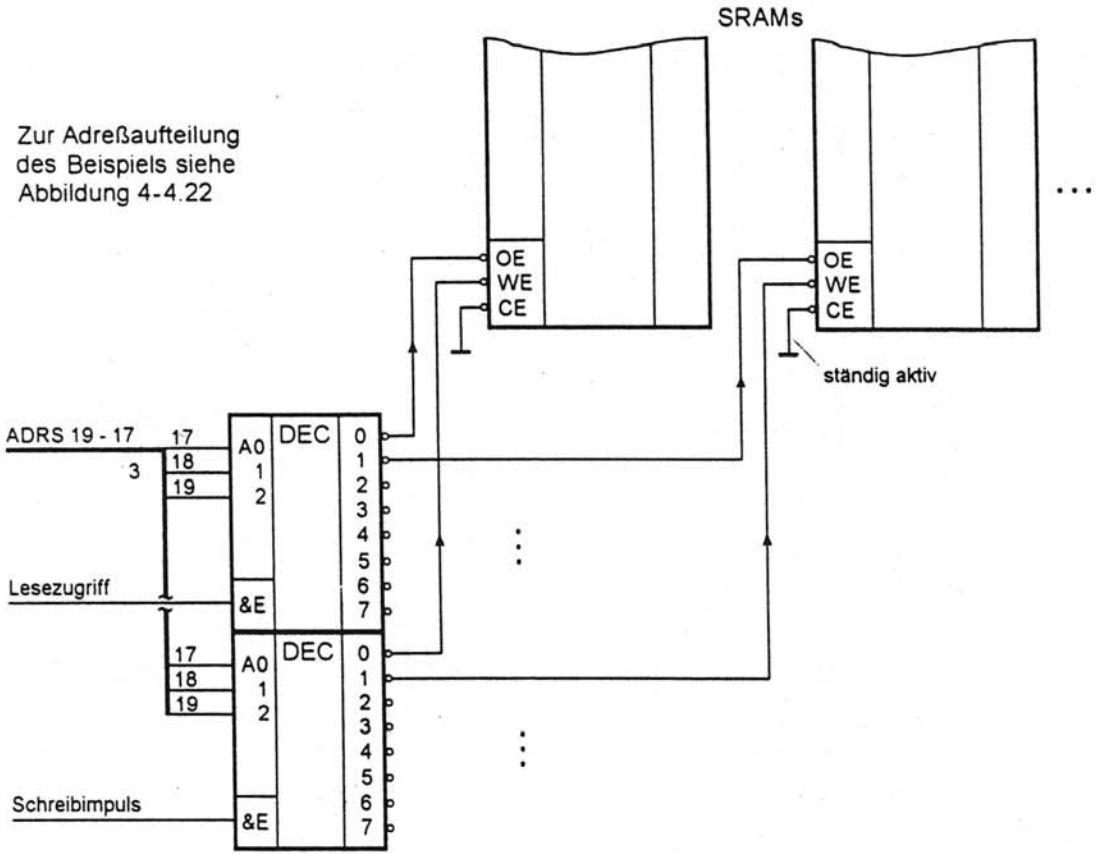


Abbildung 4-4.23  
"Schnelle" Adreßdecodierung

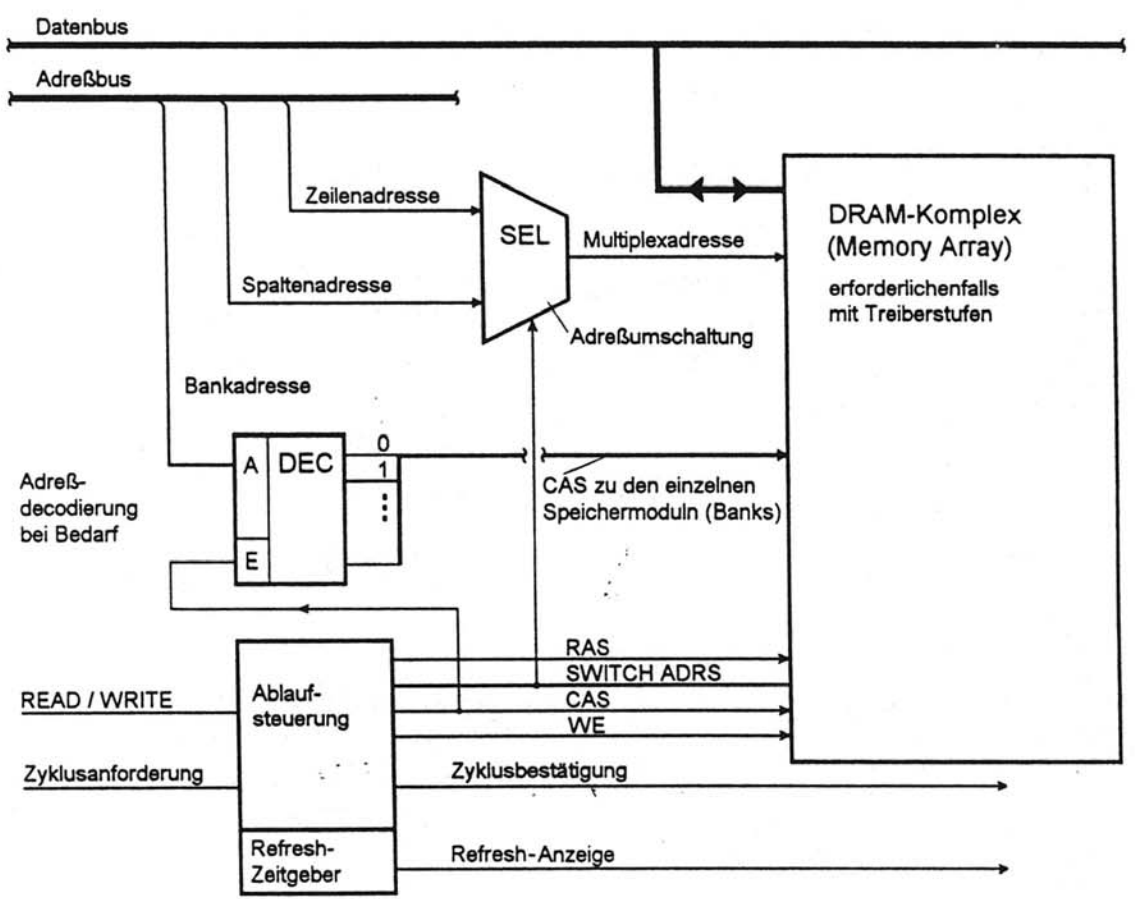


Abbildung 4-4.62  
Grundsätzliche Struktur eines DRAM-Speichersubsystems

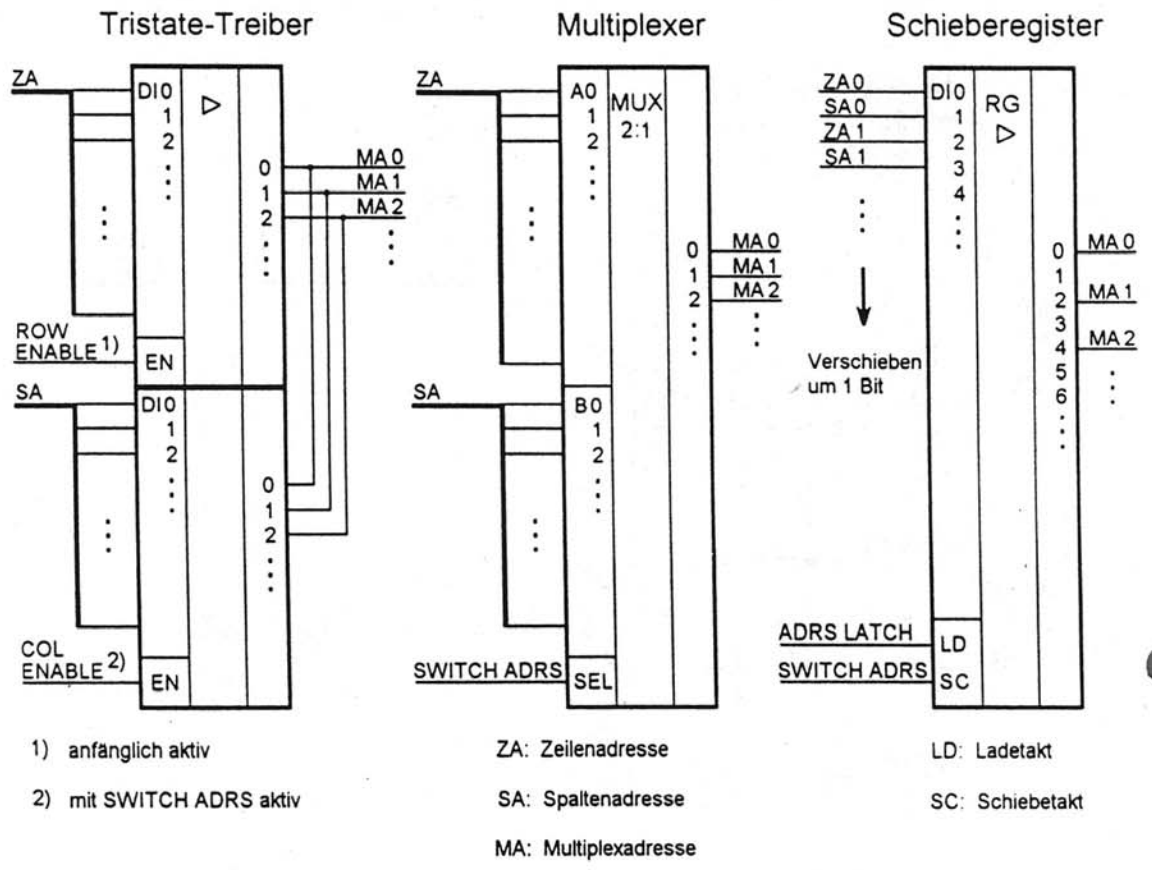


Abbildung 4-4.63  
Prinzipien der Adreßumschaltung

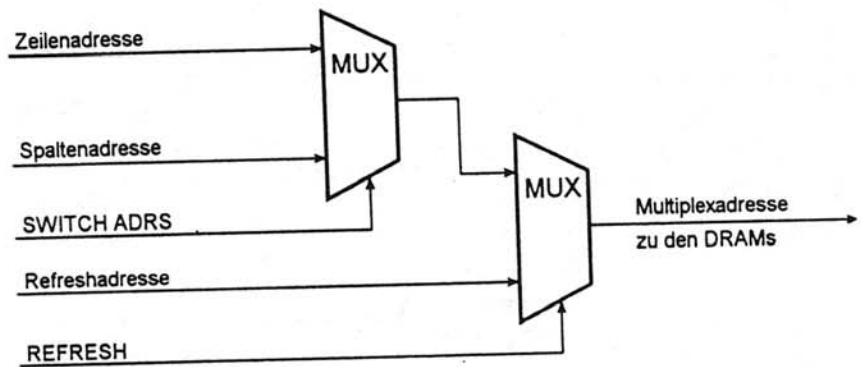


Abbildung 4-4.65  
Zuführung der Refreshadresse

**Hinweis:** Solche Schaltungen sind beispielsweise notwendig, wenn man DRAMs auf Steckkarten anordnet und sowohl eine größere Zugriffsbreite (16 Bits bei ISA; 32 Bits bei EISA oder MCA) als auch die über den Bus gelieferte Refreshadresse nutzen will.

## Adreßbitzuordnung

Abbildung 4-4.64 zeigt verschiedene Formen der Adreßbitzuordnung.

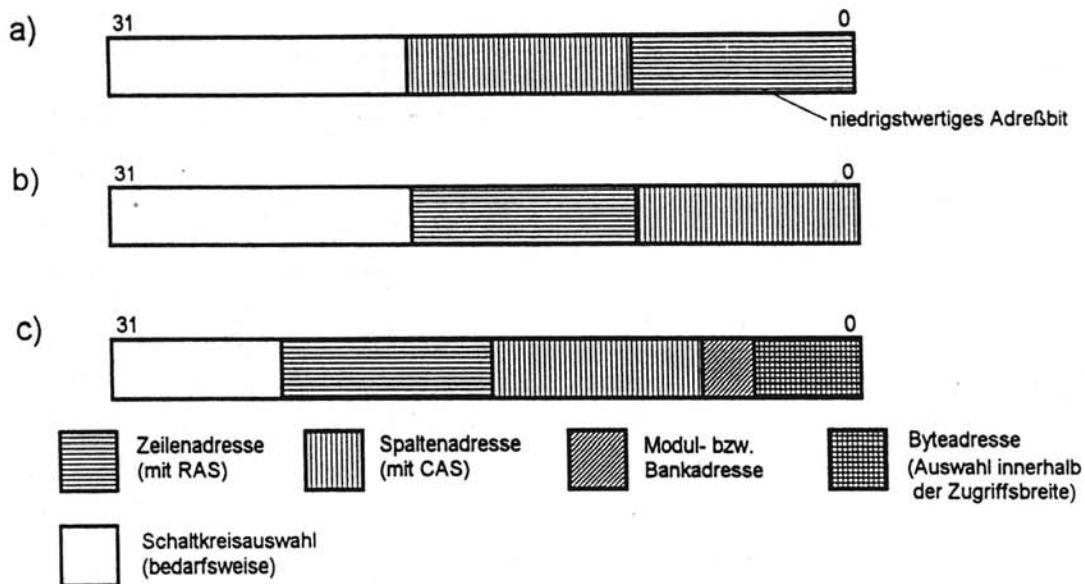


Abbildung 4-4.64  
Adreßbitzuordnungen für DRAM-Speichersubsysteme

Die Wahl hängt wesentlich von folgenden Gesichtspunkten ab:

1. Wird eine Refresh-Adresse "von außen" benötigt, so ist diese im Rahmen der Zeilenadresse (Row Address) zuzuführen (von Bit 0 an). Will man den Speicherinhalt mit Normalzugriffen auffrischen, so muß die Zeilenadresse von den niedrigstwertigen Adreßbits gebildet werden, um zu sichern, daß in der einzelnen Refresh-Periode jeweils der gesamte Bereich "überstrichen" wird (Abbildung 4-4.64 a)).
2. Will man die jeweilige "schnelle" Zugriffsweise ausnutzen, muß die Zeilenadresse von höherwertigen Adreßbits gebildet werden, da diese sich bei aufeinanderfolgenden Zugriffen wesentlich weniger häufig ändern als die niedrigstwertigen (Abbildung 4-4.64 b)).
3. Speichersubsysteme mit größerer Zugriffsbreite oder solche aus mehreren, im Interleaving betriebenen Speichermoduln :  
erfordern, daß die niedrigstwertigen Adreßbits zur Auswahl von Worten bzw. Bytes oder zur Modul- (Bank-) -Auswahl verwendet werden (Abbildung 4-4.64 c)).

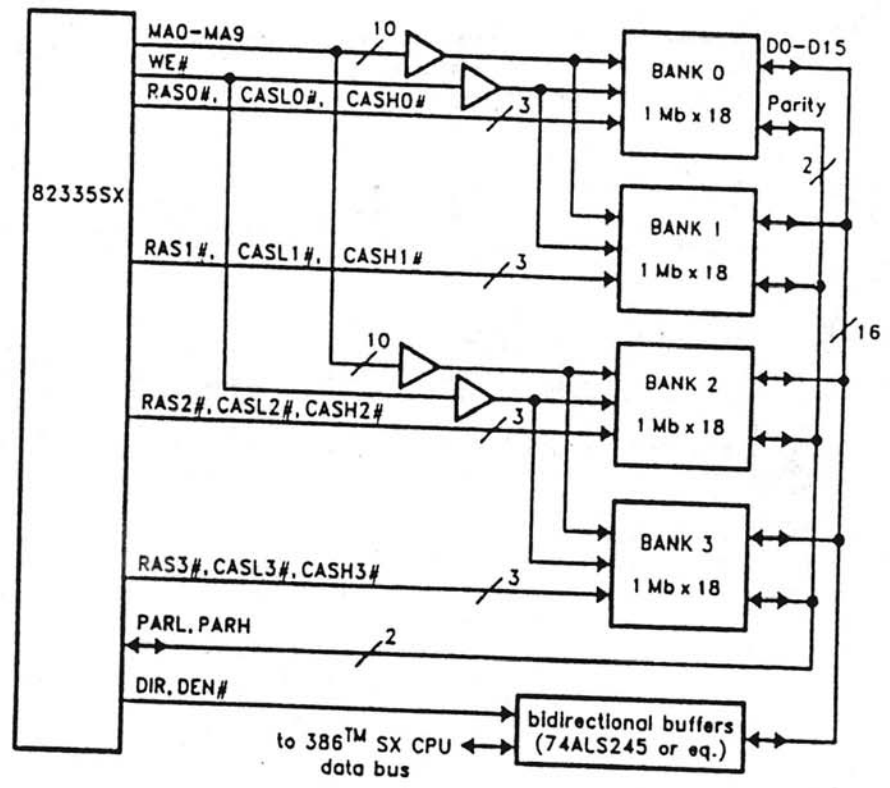


Abbildung 4-4.66  
DRAM-Interface auf einem 16-bit-Motherboard (Quelle: Intel)

Abbildung 4-4.66 zeigt eine DRAM-Anordnung, wie sie auf Motherboards mit 386-SX-Prozessoren üblich sind. Gemäß der Zugriffsbreite dieser Prozessoren sind 4 16-bit-Moduln bzw. -Banks vorgesehen (wobei jedes Byte durch ein Paritätsbit erweitert wird). Es können bis zu  $4 * 2 = 8$  MBytes installiert werden. Die Abbildung zeigt auch, an welchen Stellen in der DRAM-Ansteuerung (Adressen, WE, Daten) zusätzliche Treiberstufen eingefügt sind.

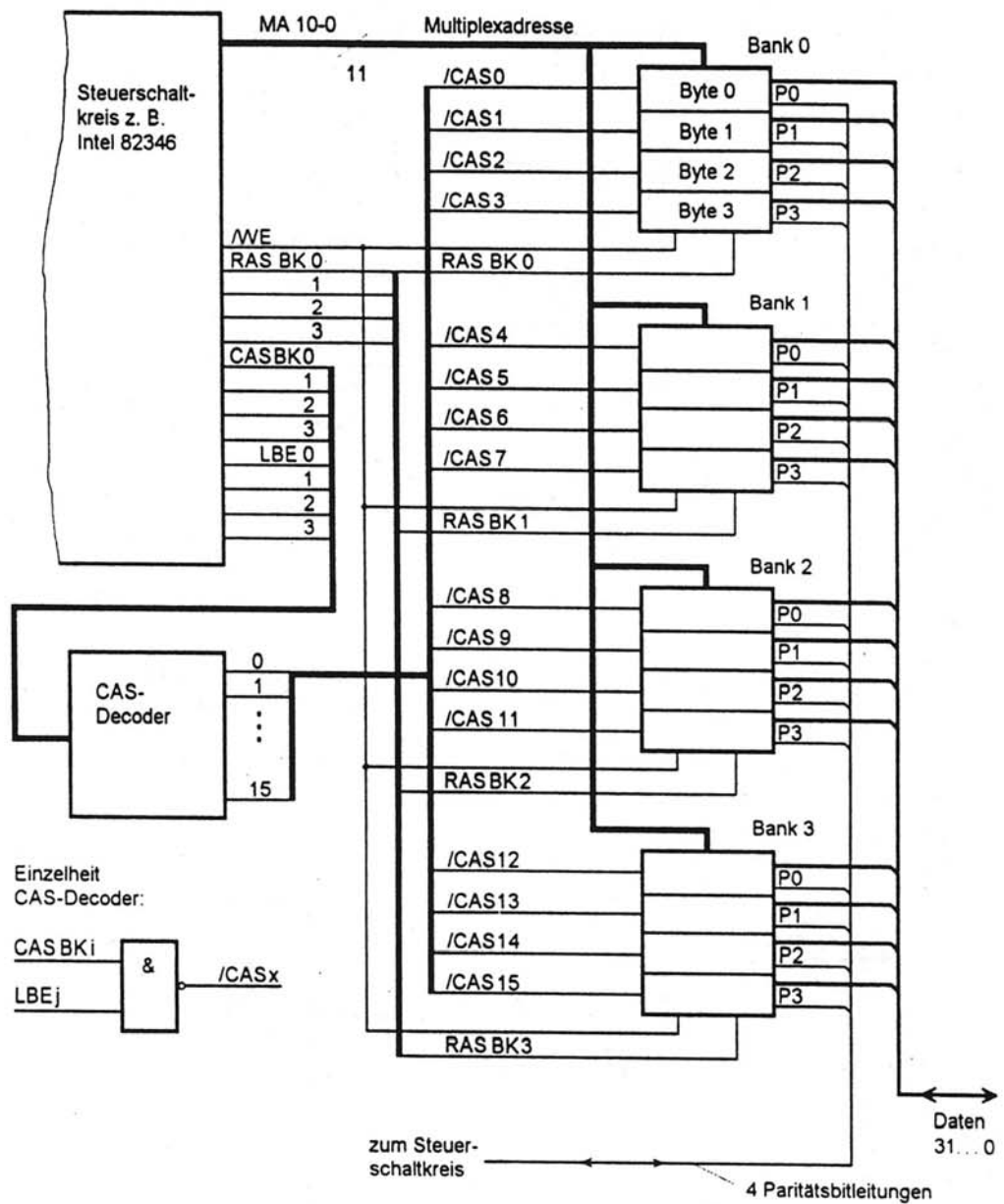


Abbildung 4-4.67  
 DRAM-Interface auf einem 32-bit-Motherboard (Quelle: Intel)

DRAM-Anordnungen entsprechend Abbildung 4-4.67 sind auf Motherboards mit "echten" 32-bit-Prozessoren zu finden. Die 4 Banks haben eine Zugriffsbreite von 32 Bits (ergänzt durch 4 Paritätsbits). Auch hier werden unabhängige RAS-Signale (RASBK3...0) für die einzelnen Banks sowie insgesamt 16 CAS-Signale zur Ansteuerung der einzelnen Bytepositionen gebildet.

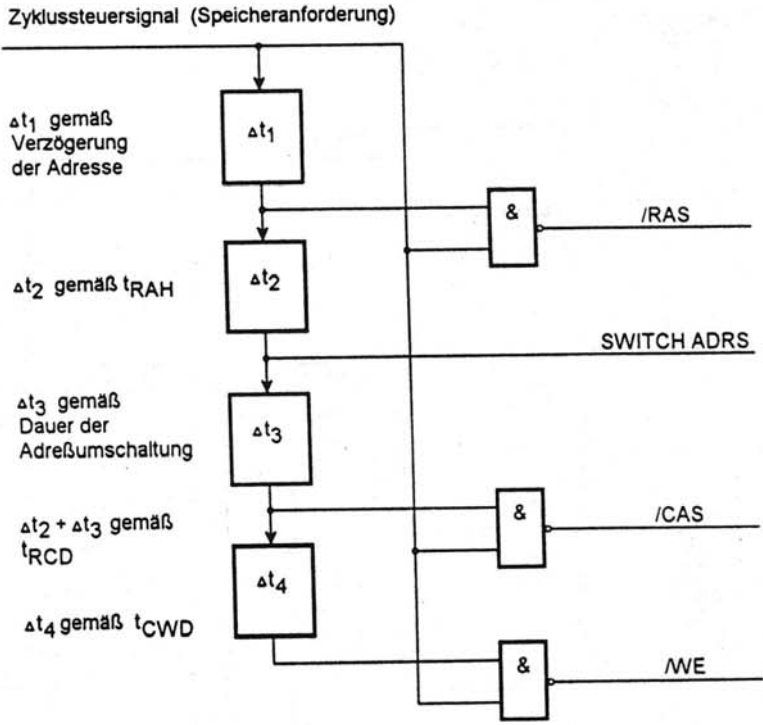


Abbildung 4-4.68  
Ablaufsteuerung mit Verzögerungsgliedern (Laufzeitketten)

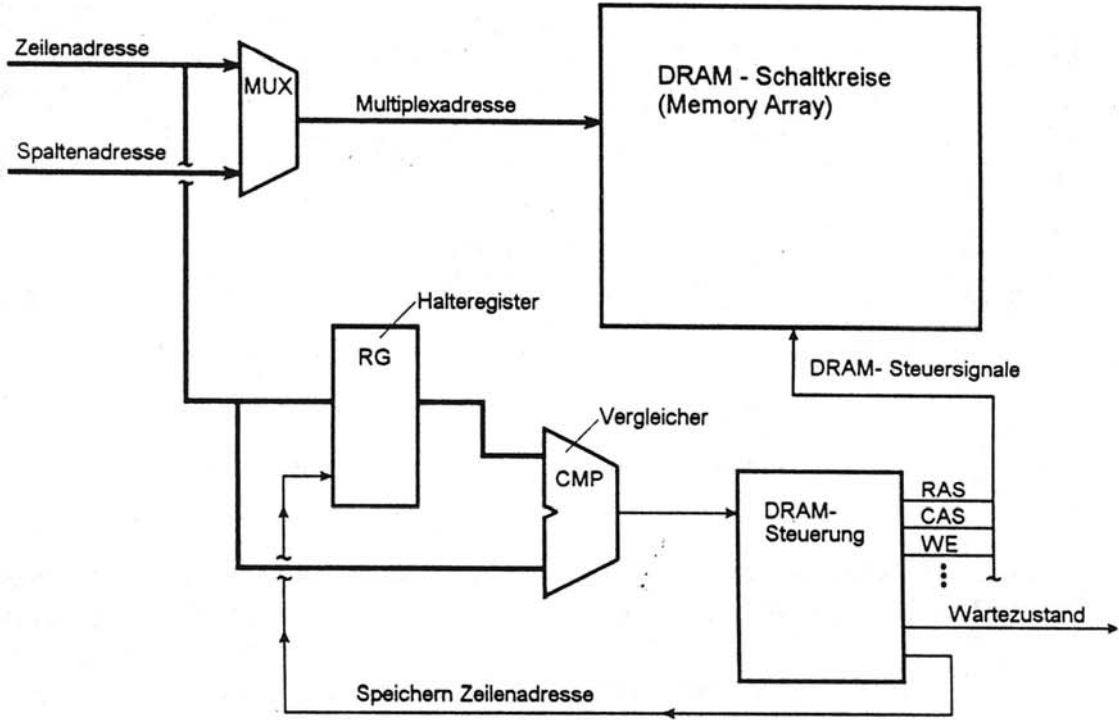
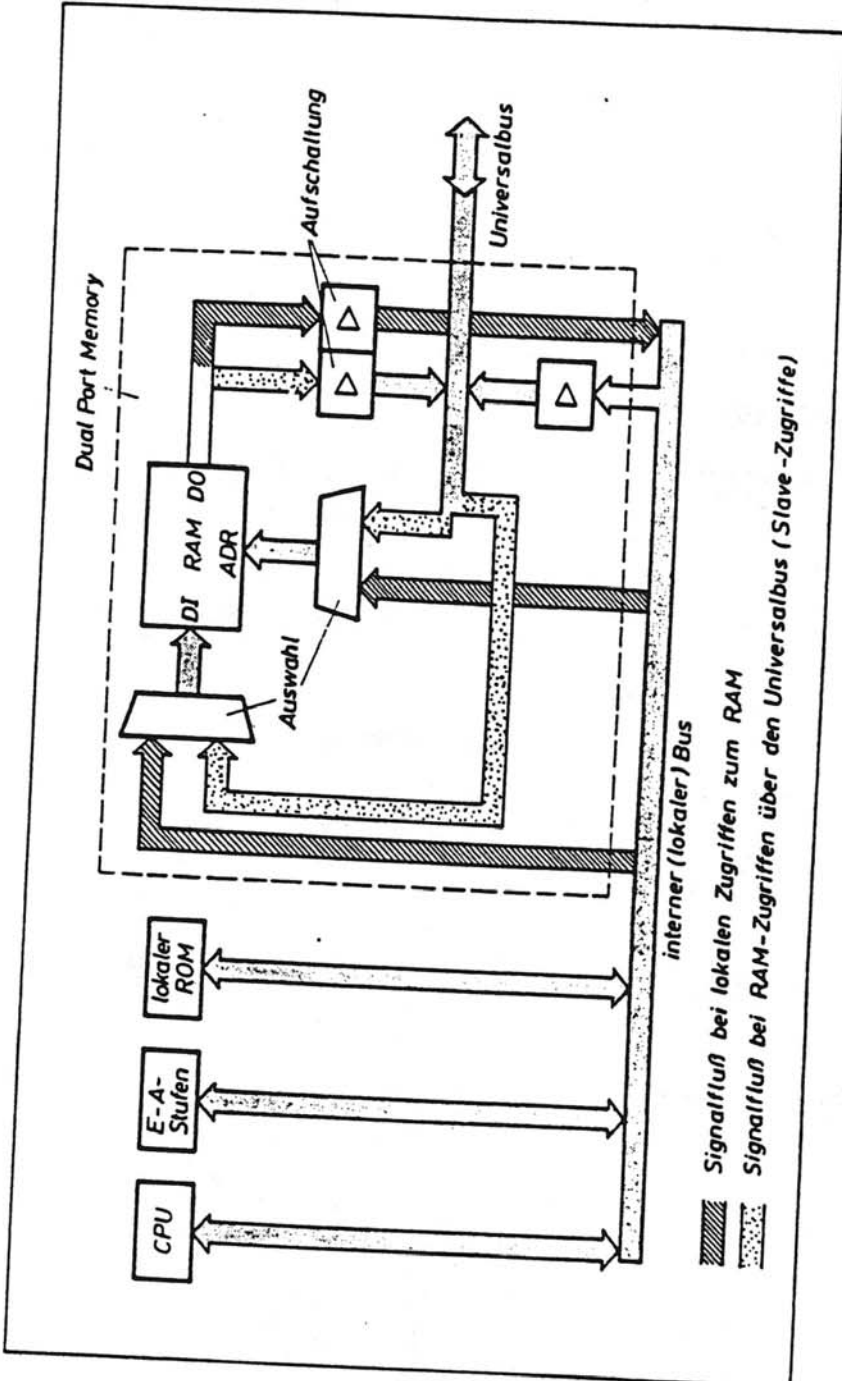
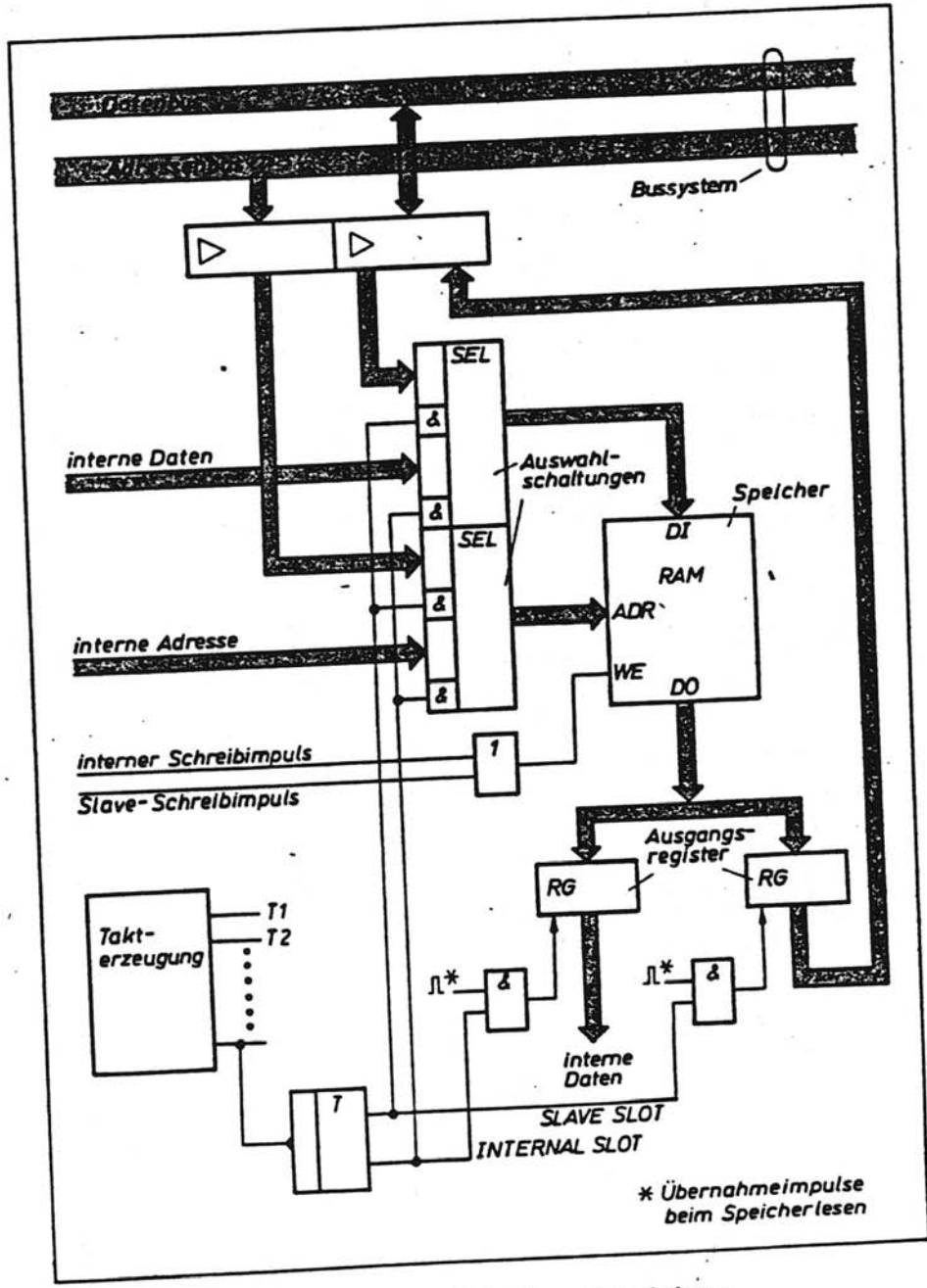


Abbildung 4-4.69  
Speichersubsystem für Nutzung des Schnellzugriffsverfahrens  
(z. B. Page Mode)



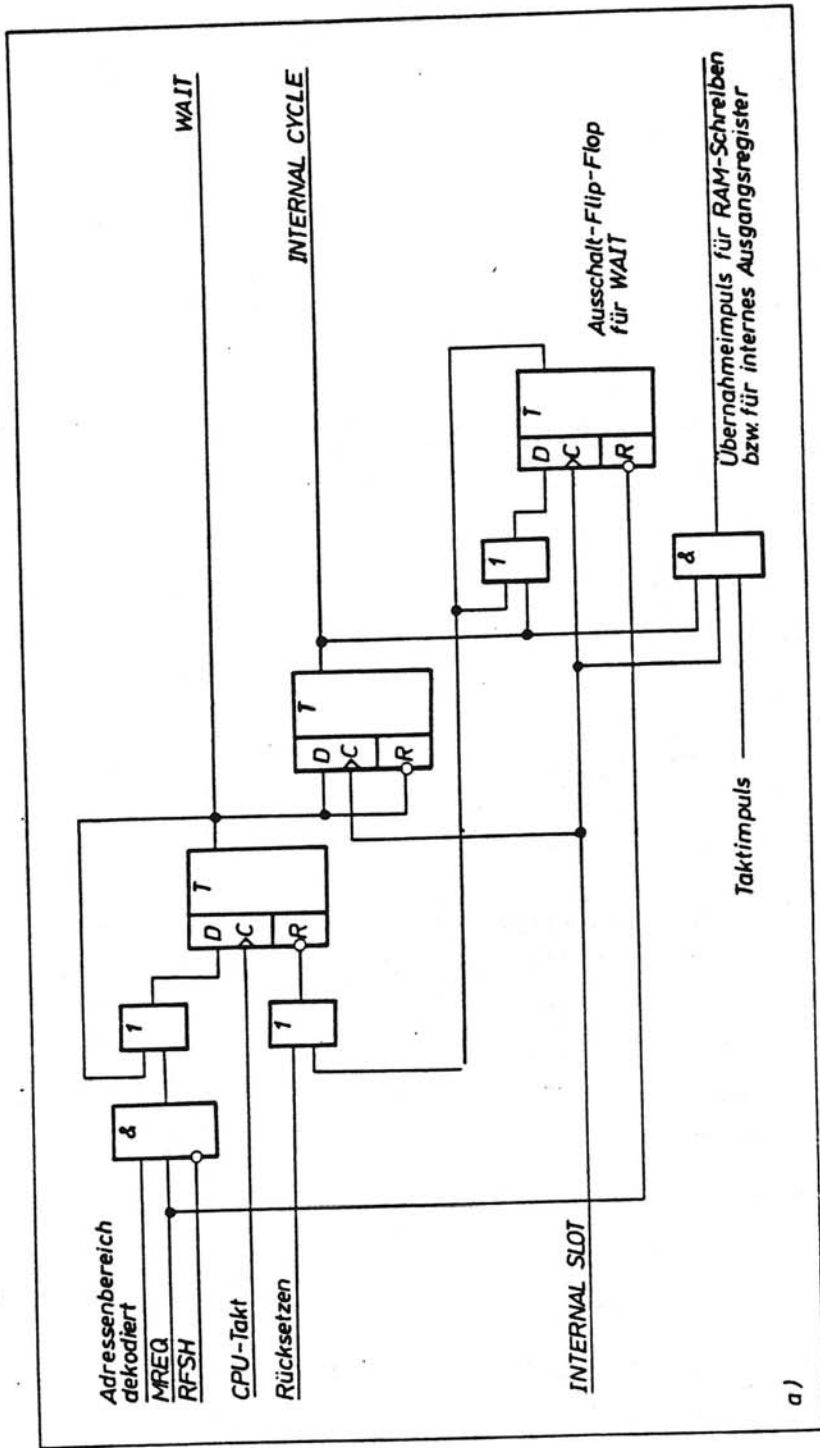


Prinzip eines Mikrorechners mit  
 "Dual Port Memory"

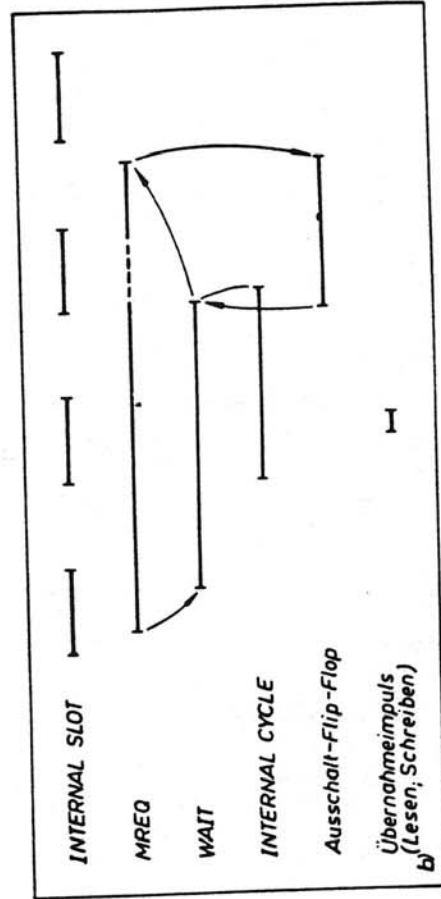


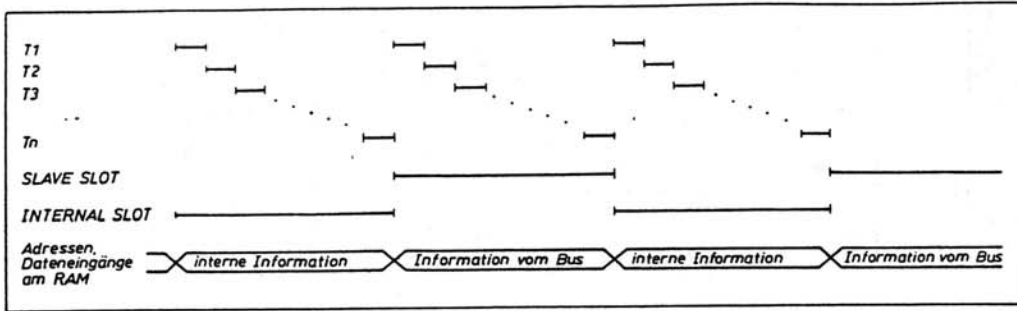
Speicheranordnung mit Zeiteilungsvermittlung

Synchronisationsschaltung  
(Waitsteuerung) für den Anschluß  
eines Mikroprozessors an eine Zeit-  
teilungsvermittlung. a) Prinzipschal-  
tung; b) Zeitdiagramm

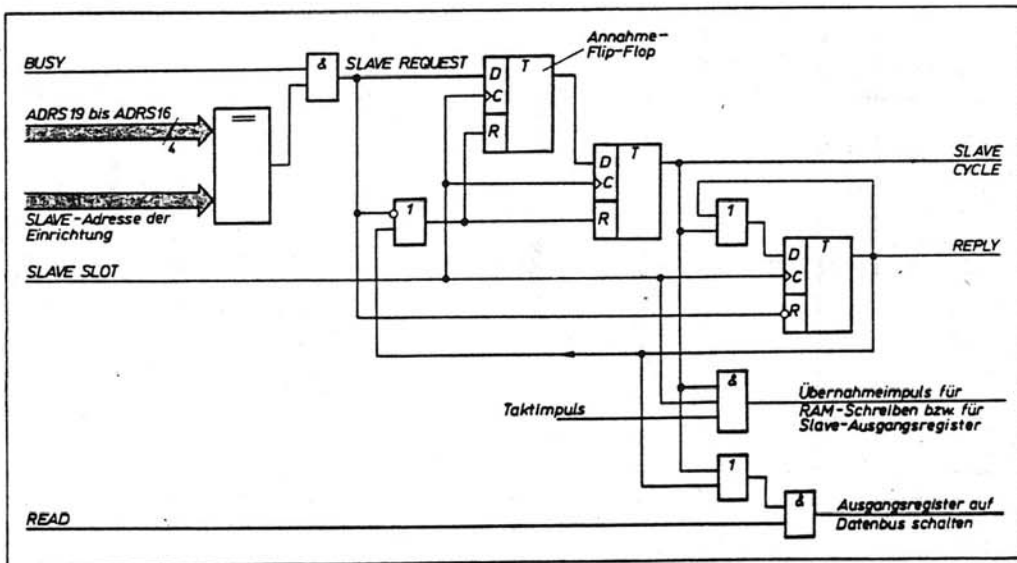


a)

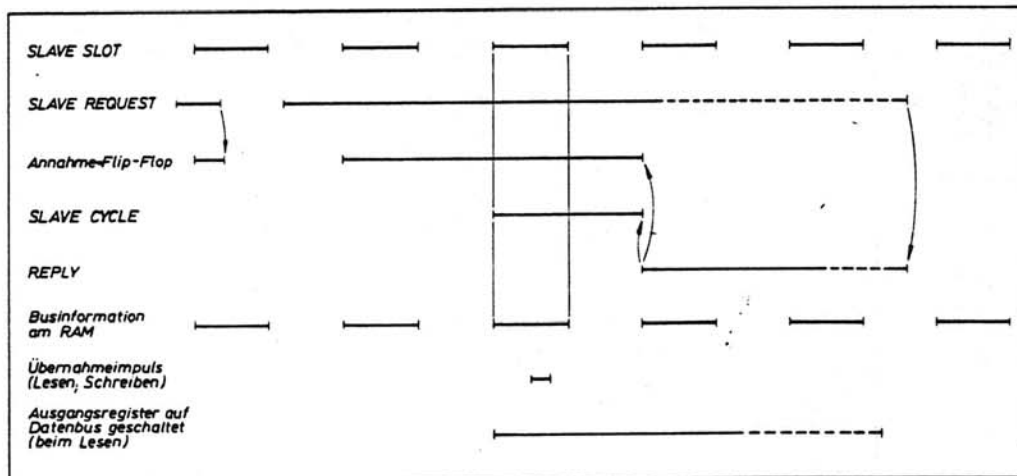




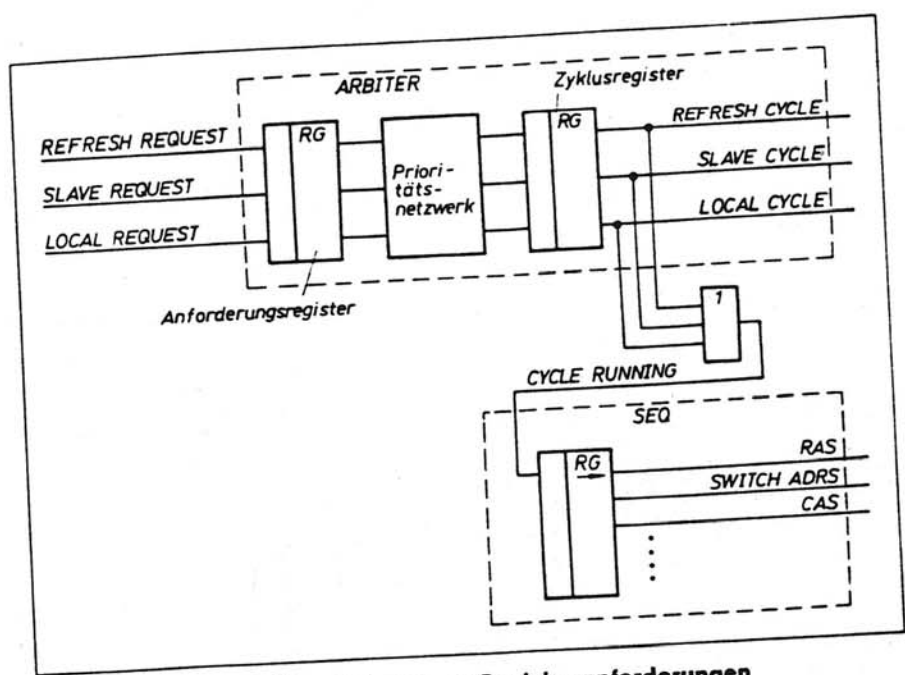
Zeitdiagramm einer einfachen Zeitteilungsvermittlung



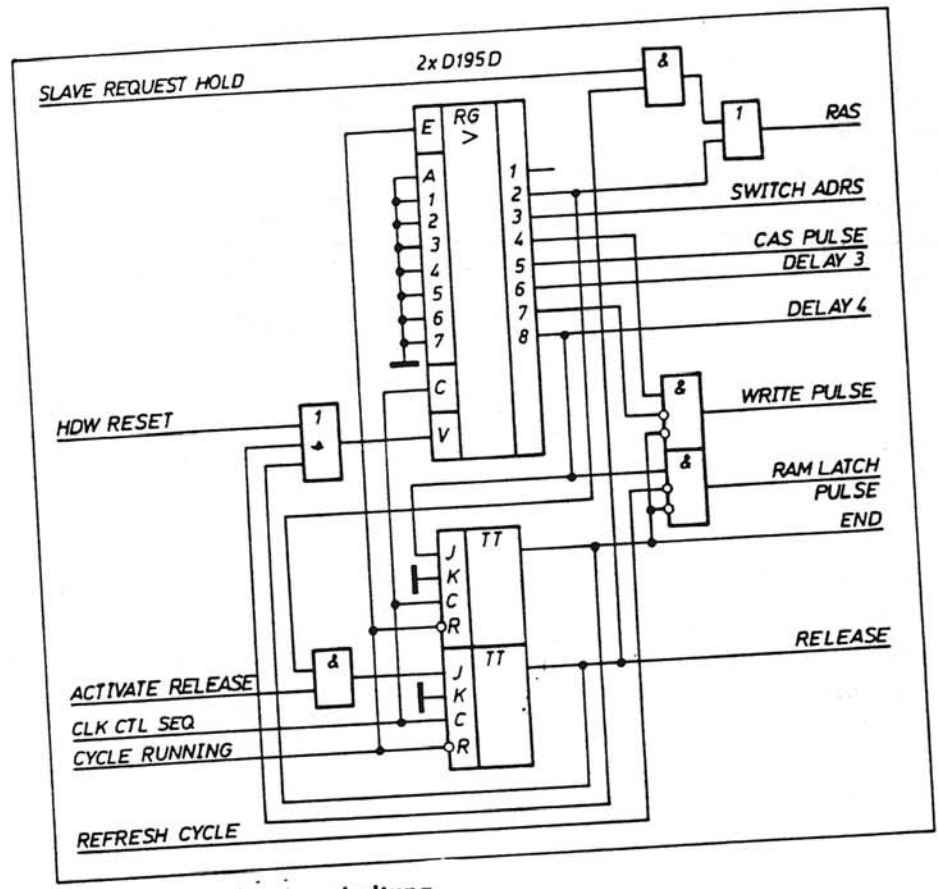
Steuerschaltung für Slavezugriffe zu einem Speicher mit Zeitteilungsvermittlung



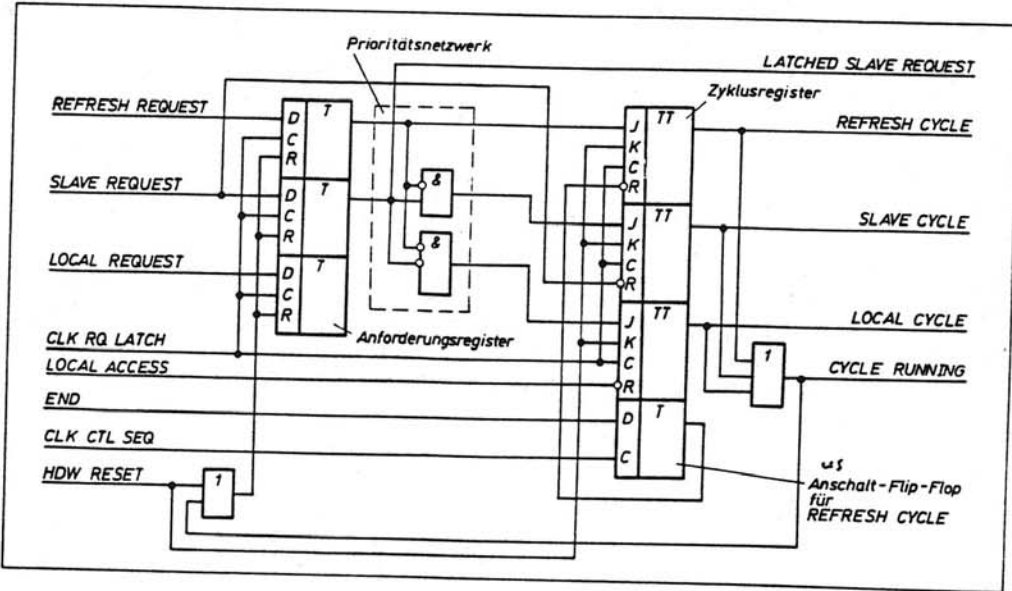
Zeitdiagramm zur Steuerung



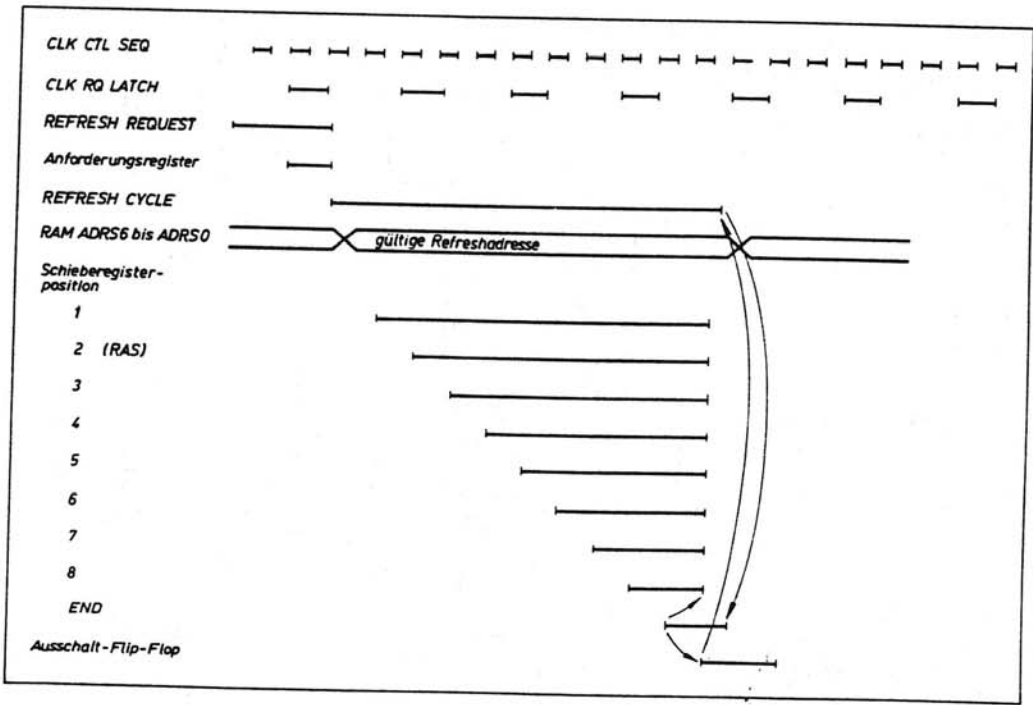
Prinzip der Vermittlung der Speicheranforderungen



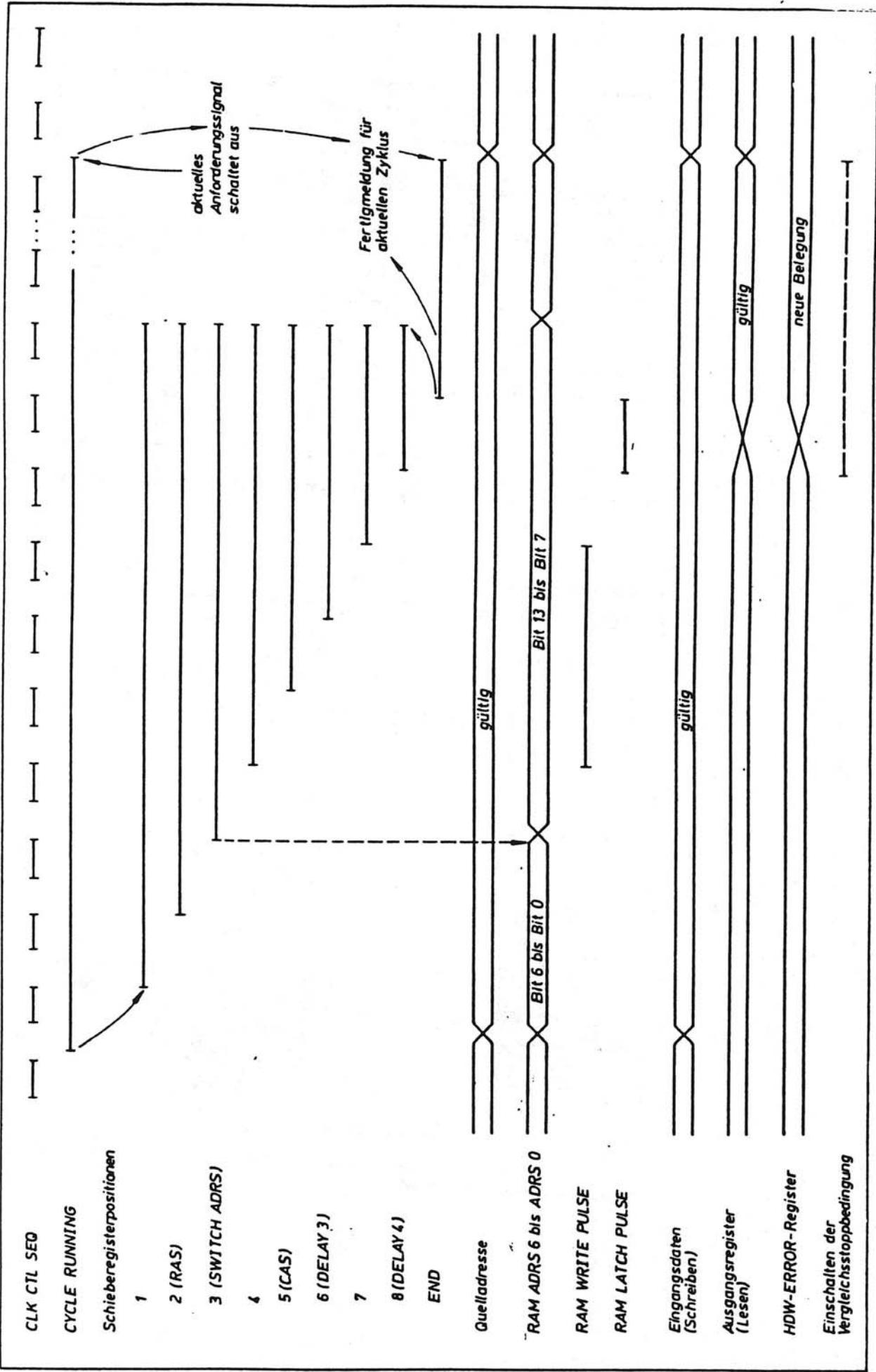
Ablaufsteuerschaltung



Vermittlungsschaltung



Ablauf eines Refreshzyklus



Allgemeiner Ablauf eines Speicherzugriffs

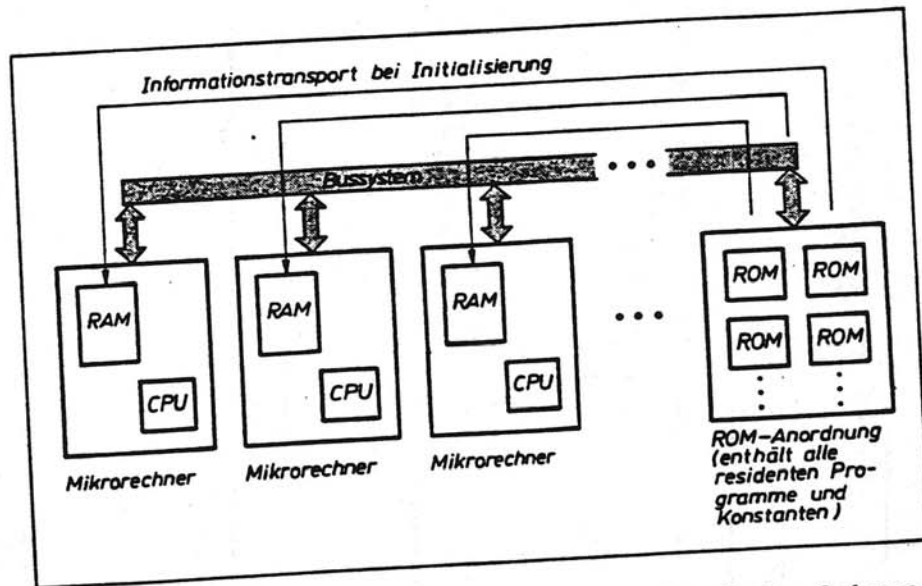
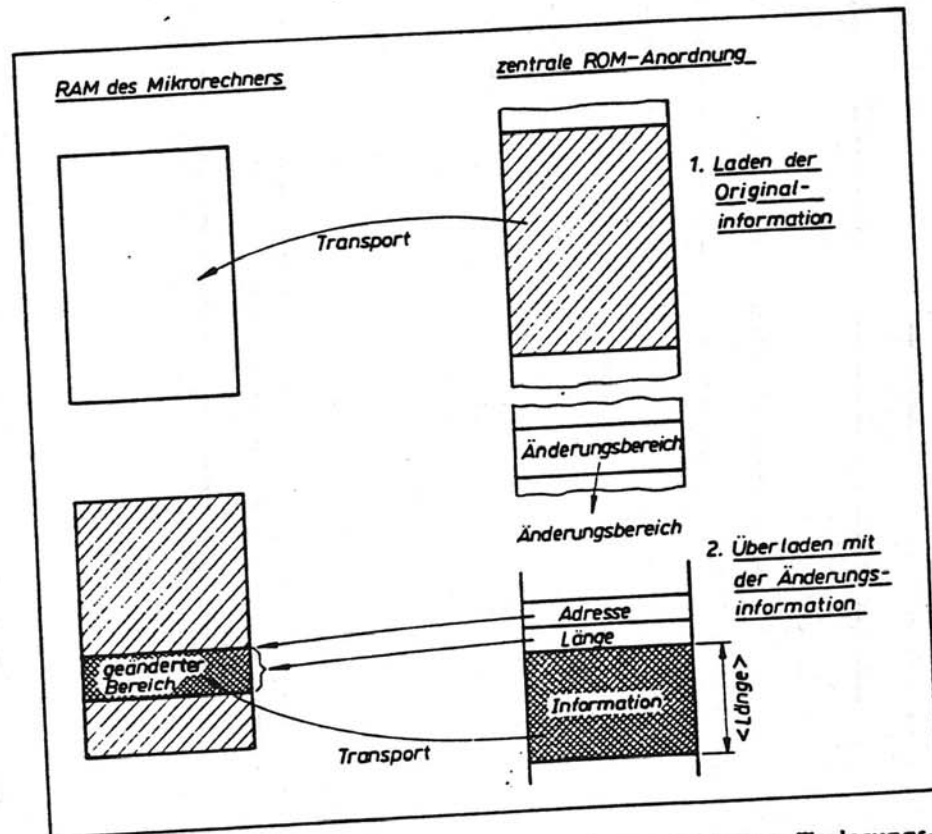


Bild 70: Multimikrorechnersystem mit zentralem ROM für das Anfangsprogramm laden



Einbringen von Änderungen aus einem speziellen Änderungsbereich des zentralen ROM



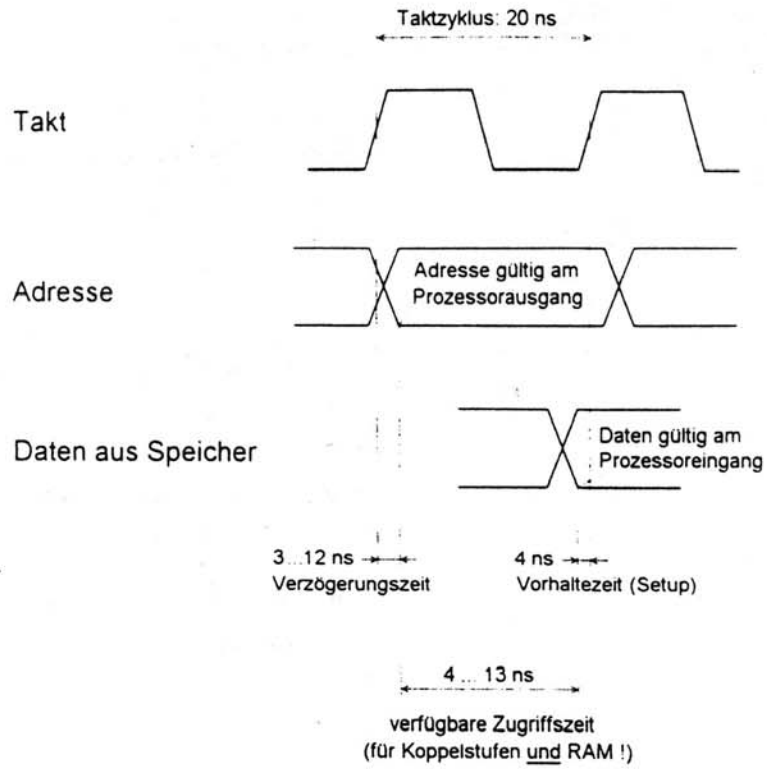


Abbildung 3-4.2  
Zeitverhältnisse beim Speicherzugriff (Lesen) am Beispiel des i486 DX, 50 MHz  
(schnellste Zugriffsweise)

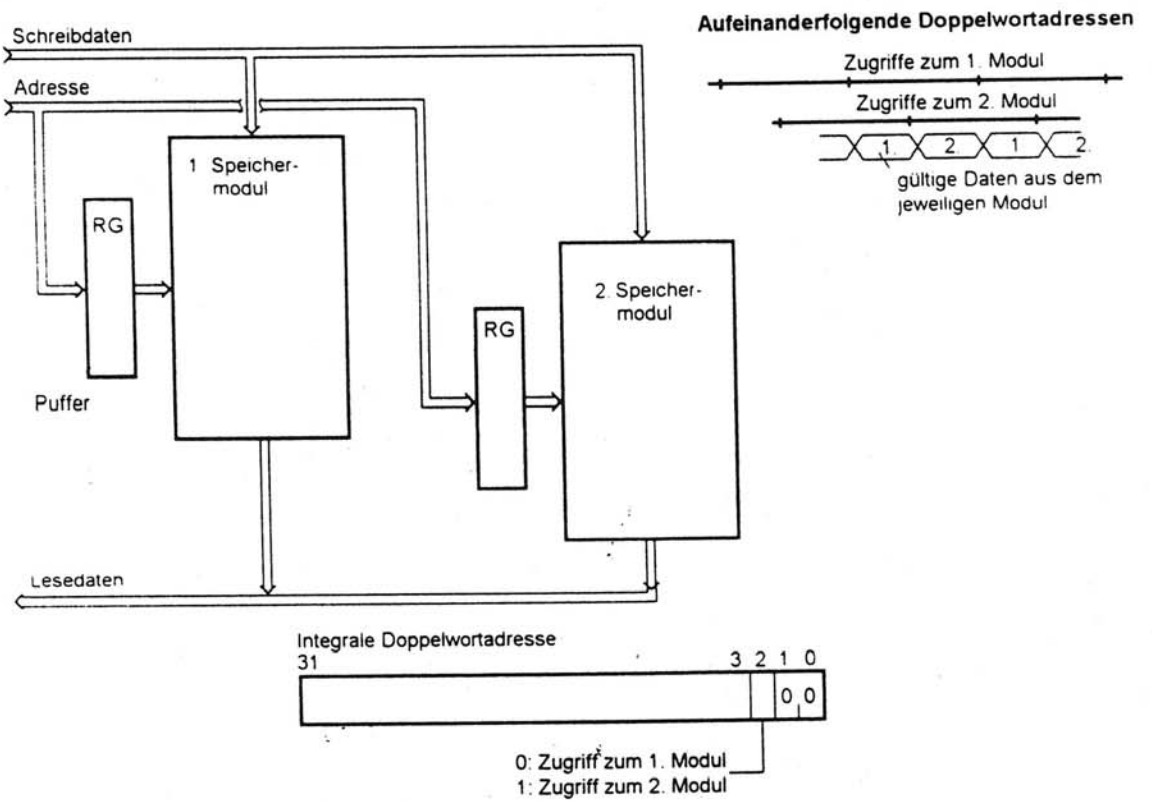


Abbildung 3-4.3  
Interleaving-Organisation

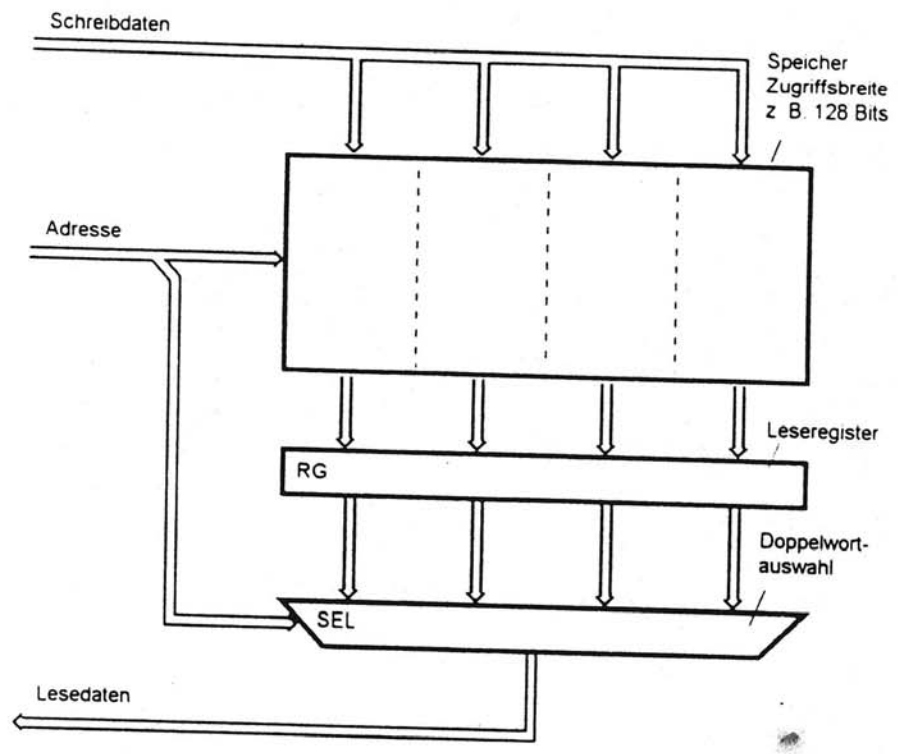


Abbildung 3-4.4 Speicher mit mehrfacher Zugriffsbreite

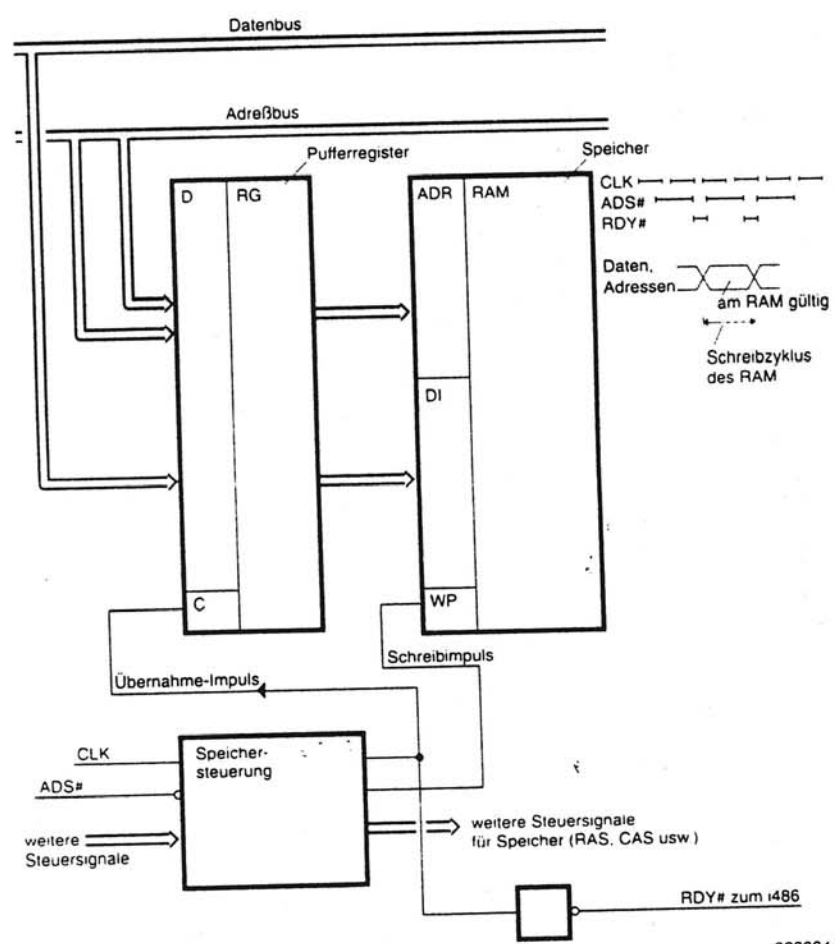
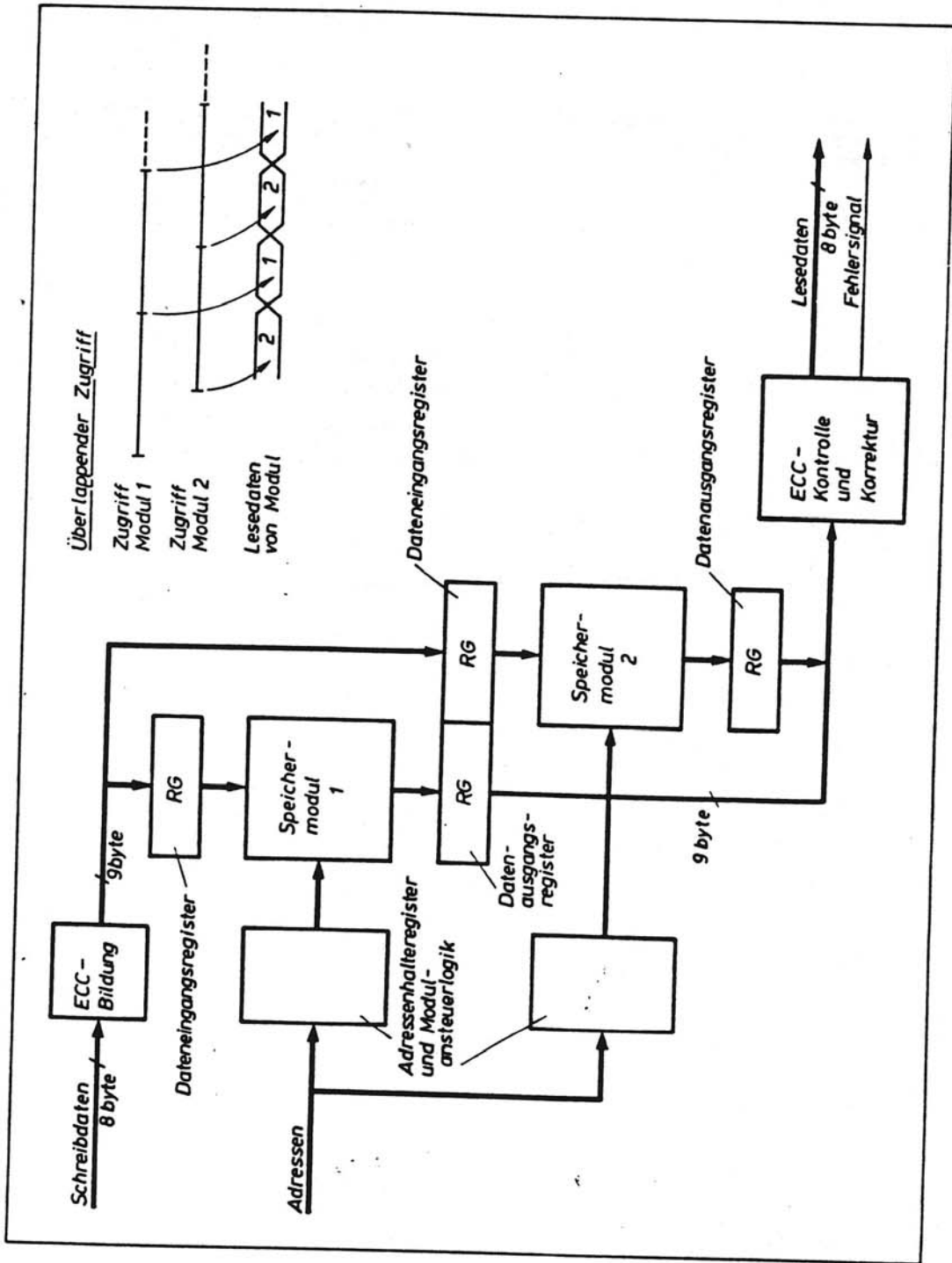
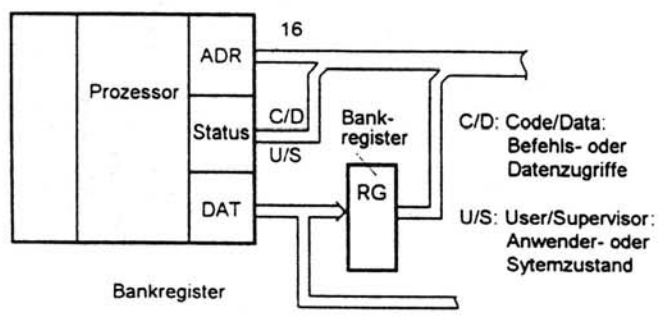


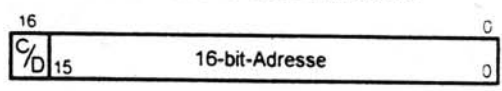
Abbildung 22.29 Nacheilende Schreibzugriffe (Prinzip-Schaltung)



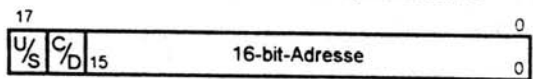
Struktur eines zentralisier-  
ten Speichers, der aus zwei über-  
lappend arbeitenden Modulen  
aufgebaut ist



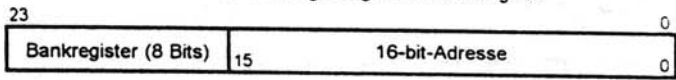
a) Verlängerung durch Unterscheidung zwischen Befehls- und Datenzugriffen



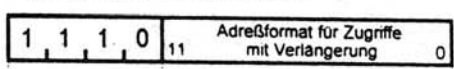
b) Weitere Verlängerung durch Unterscheidung zwischen Anwender- und Systemzustand



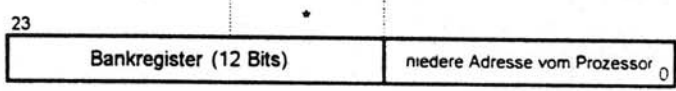
c) Verlängerung durch Bankregister



d) Verlängerung eines Ausschnitts aus dem Adreßraum durch Bankregister



\* Dieser Abschnitt der Prozessor-Adresse wird beim Zugriff aus dem Bankregister ersetzt



e) Mehrere Bankregister für verschiedene Zugriffe



Abbildung B3-1.51  
Adreßverlängerung durch Konkatenation

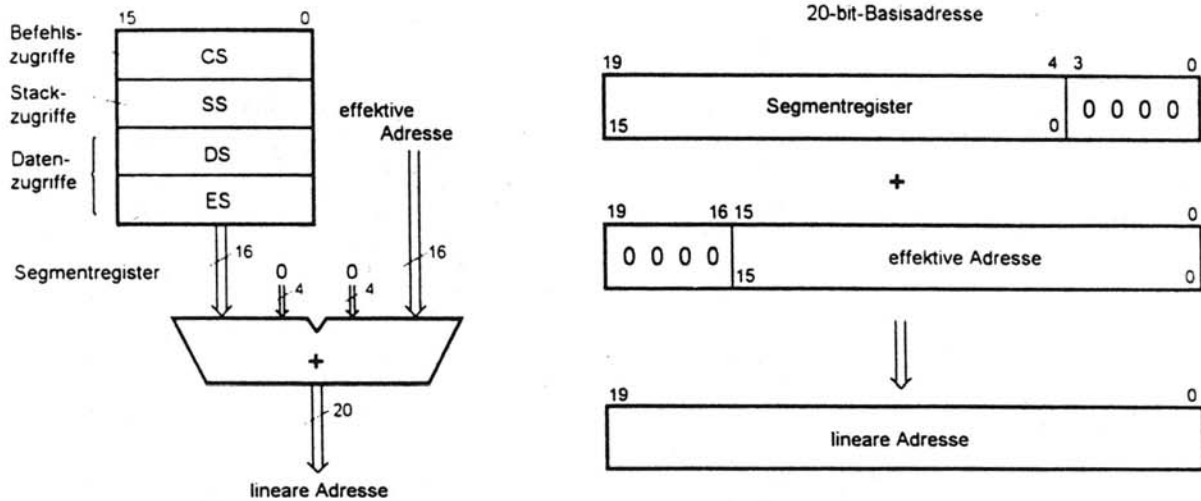


Abbildung B3-1.52  
Adreßverlängerung durch Segmentierung beim 8086

## Umsetzung

Man kann eine beliebige binär codierte Angabe a in eine beliebige andere binäre Angabe b umsetzen, wenn man a als Adresse für einen Direktzugriffsspeicher verwendet, der für jeden Wert von a den zugeordneten Wert von b enthält (Abbildung B3-1.53).

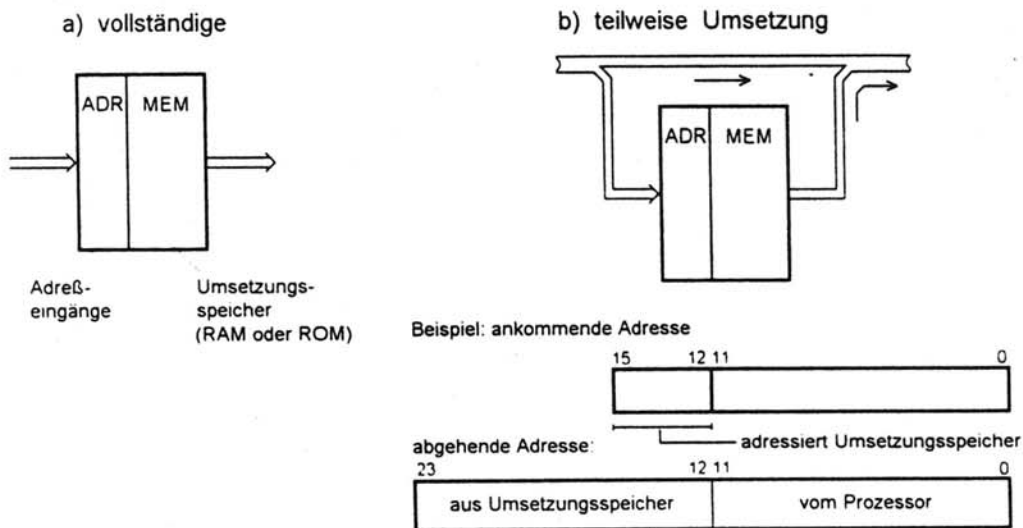
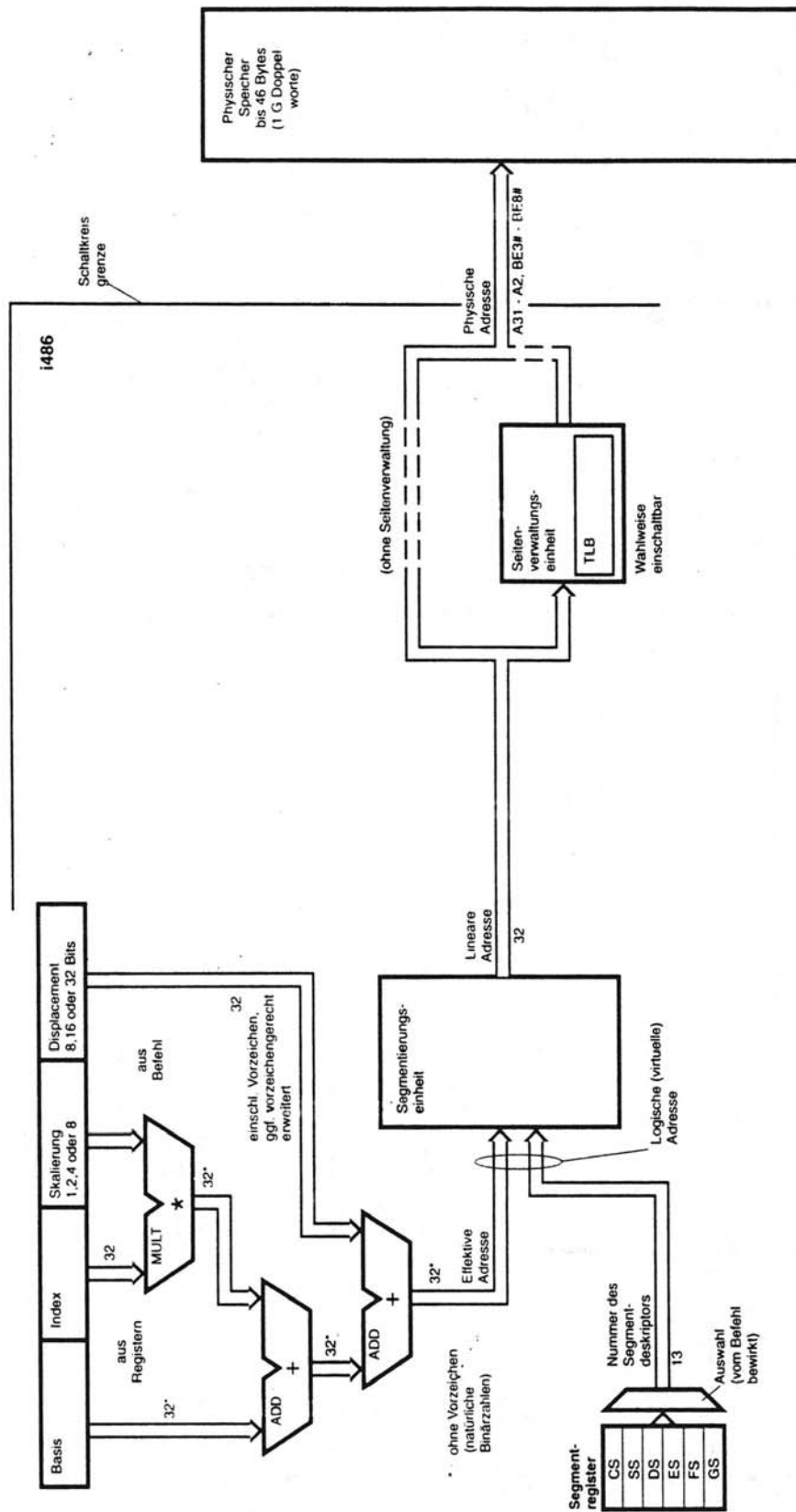


Abbildung B3-1.53  
Adressenumsetzung über Umsetzungsspeicher

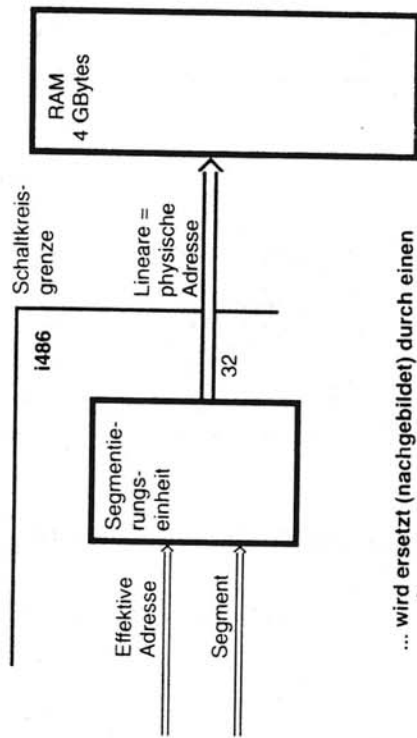




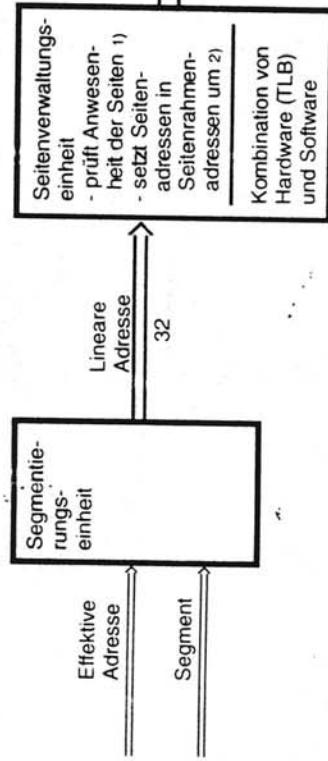
929004-1.9

Abbildung 2.8 Adreßrechnung: Vom Befehl zur physischen Speicheradresse

Diese (sehr teure) Anordnung ...

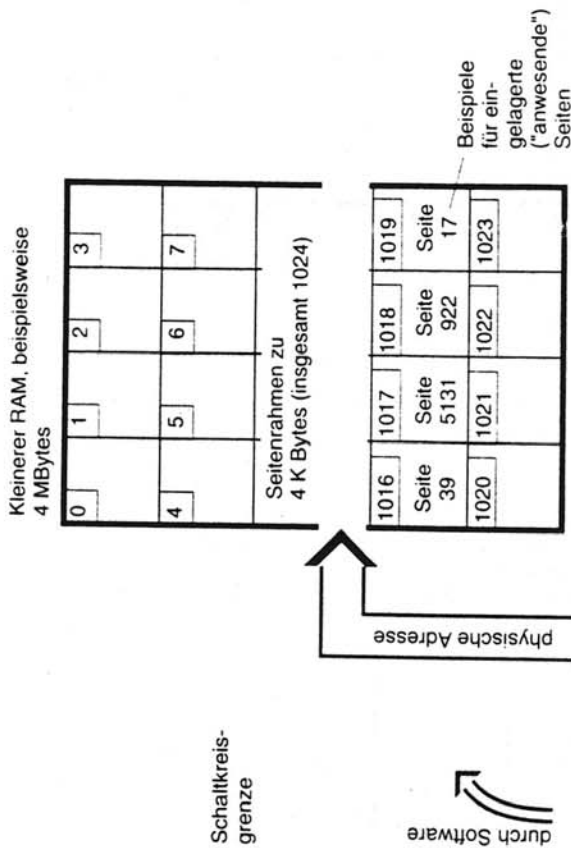


... wird ersetzt (nachgebildet) durch einen seitenorientierten virtuellen Speicher.

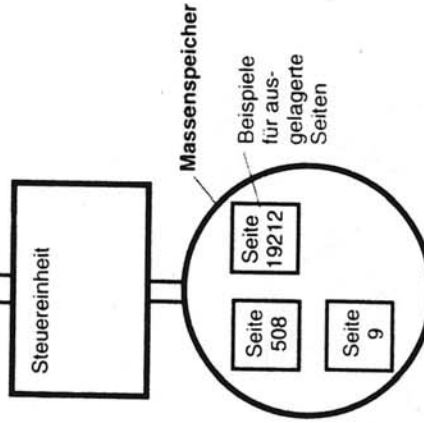


1) Beispiele: Seite 39 ist anwesend  
Seite 508 ist nicht anwesend

2) Beispiele: Seite 39  
Seitenrahmen 1016



Systembus  
Insgesamt sind 1 048 567 (1M) Seiten zu verwalten



Schaltkreisgrenze

durch Software

Seitenaustausch

Abbildung 2.10 Prinzip des seitenorientierten virtuellen Speichers

929004-1.11



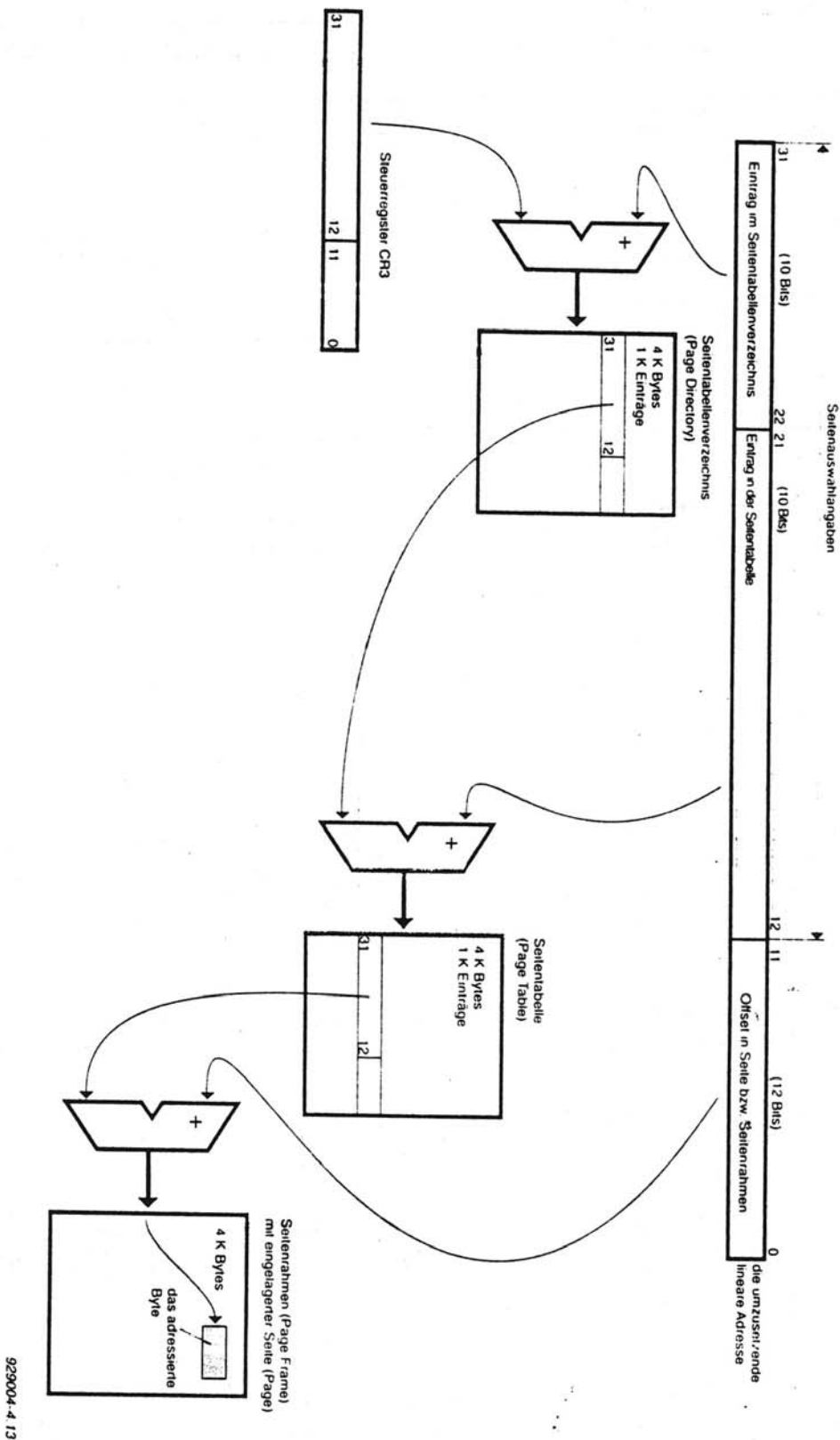


Abbildung 6.4 Prinzip der Adreßumsetzung

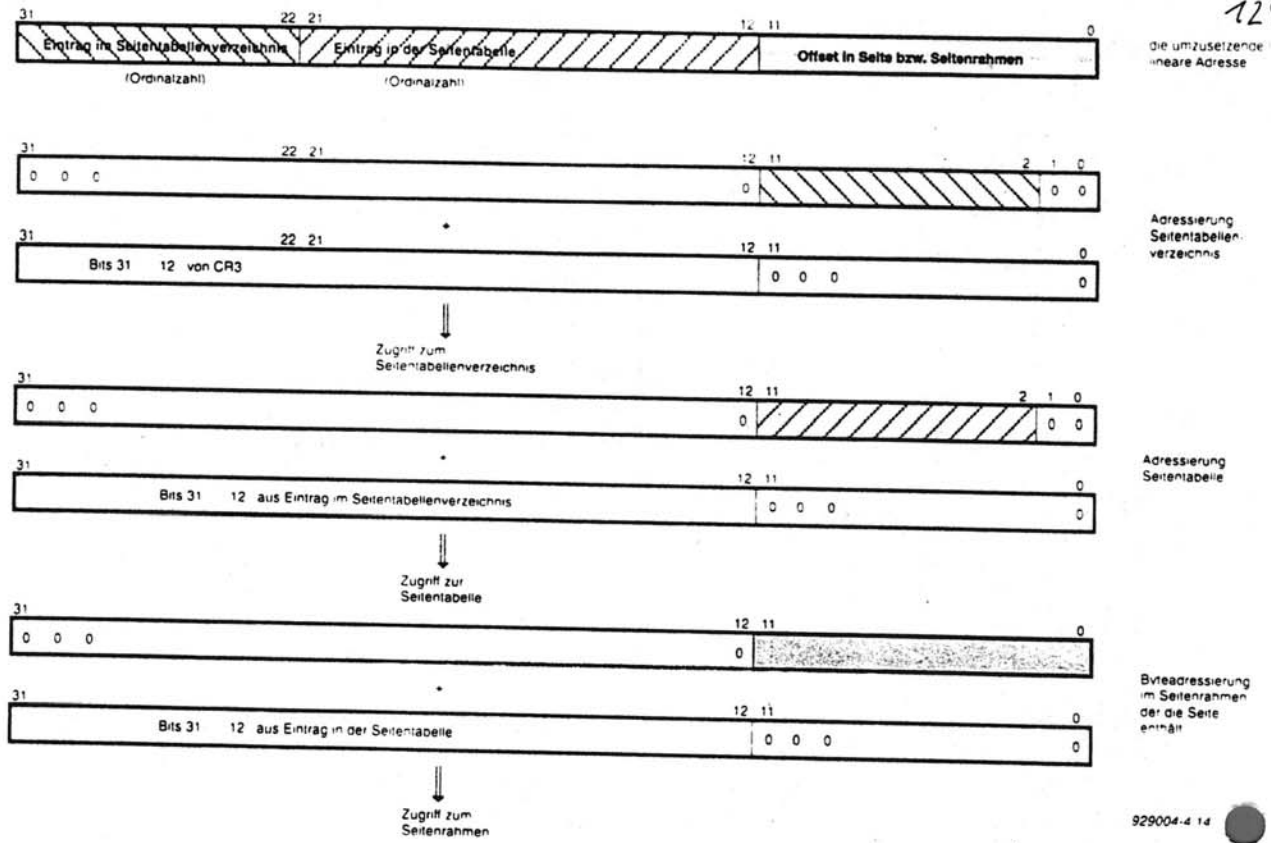


Abbildung 6.5 Einzelheiten der Adressierung

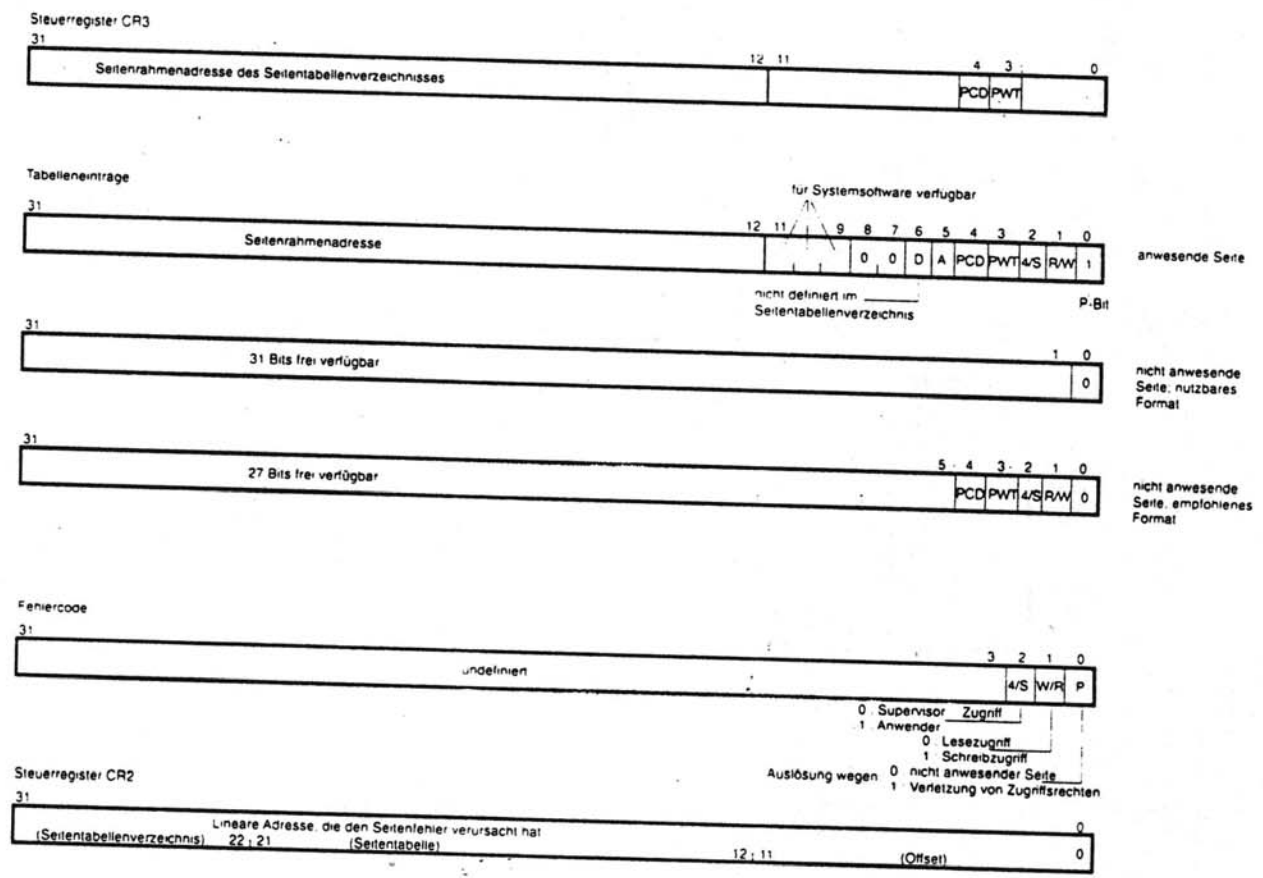


Abbildung 6.6 Informationsstrukturen der Seitenverwaltung

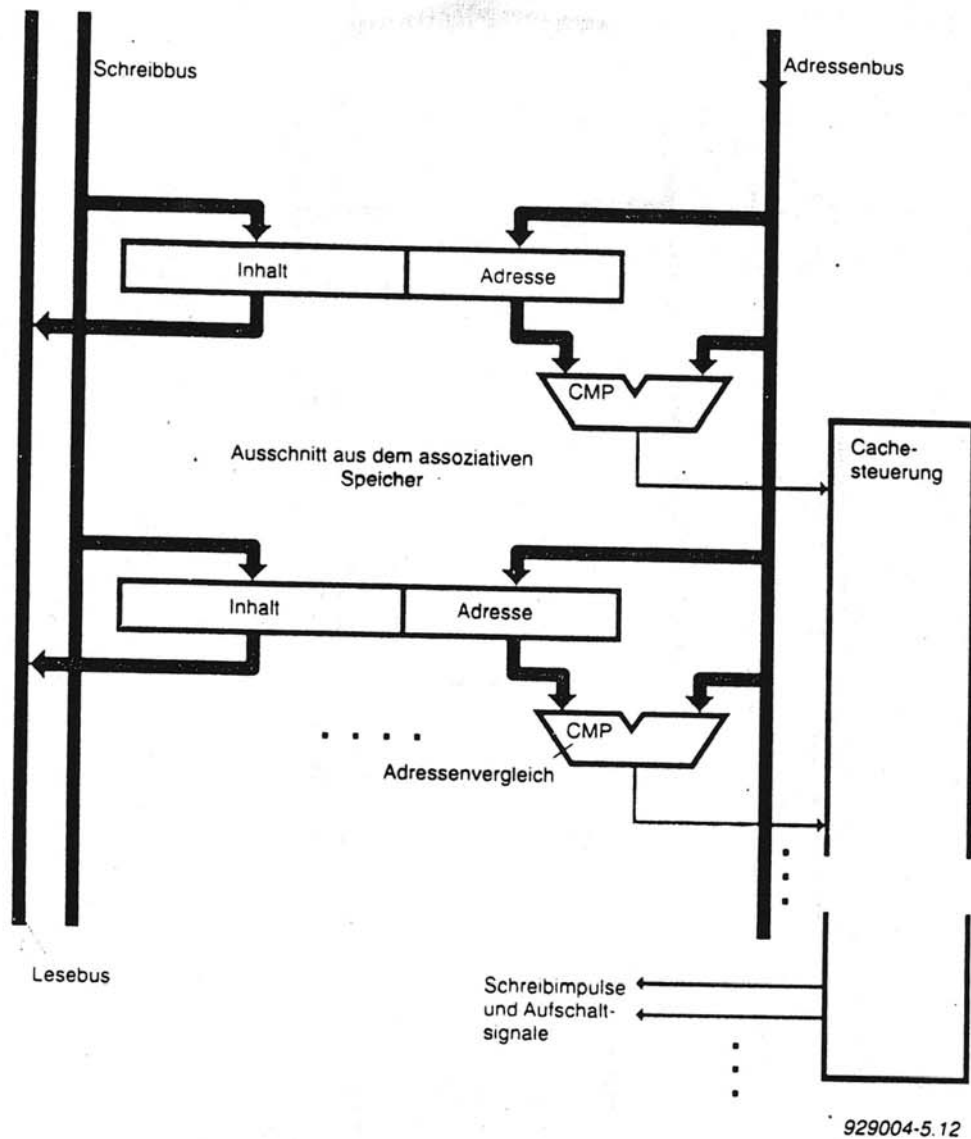


Abbildung 12.1 Cache-Organisation (1): assoziativ

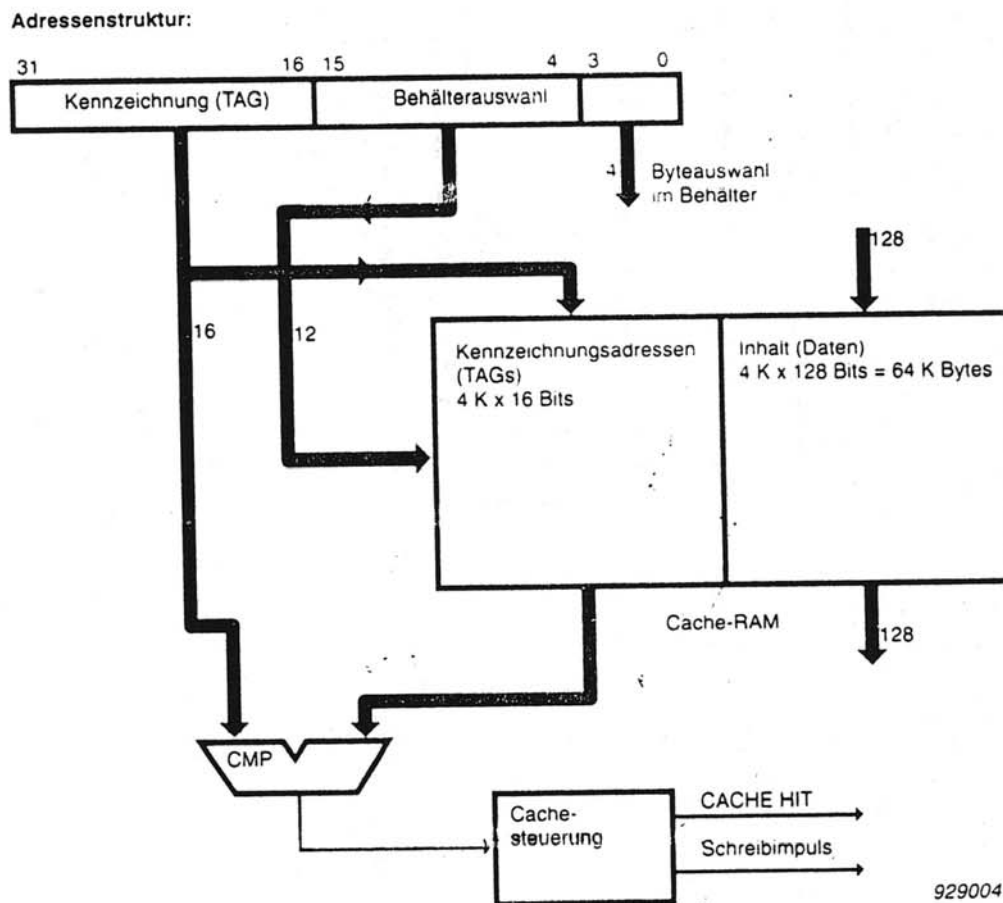
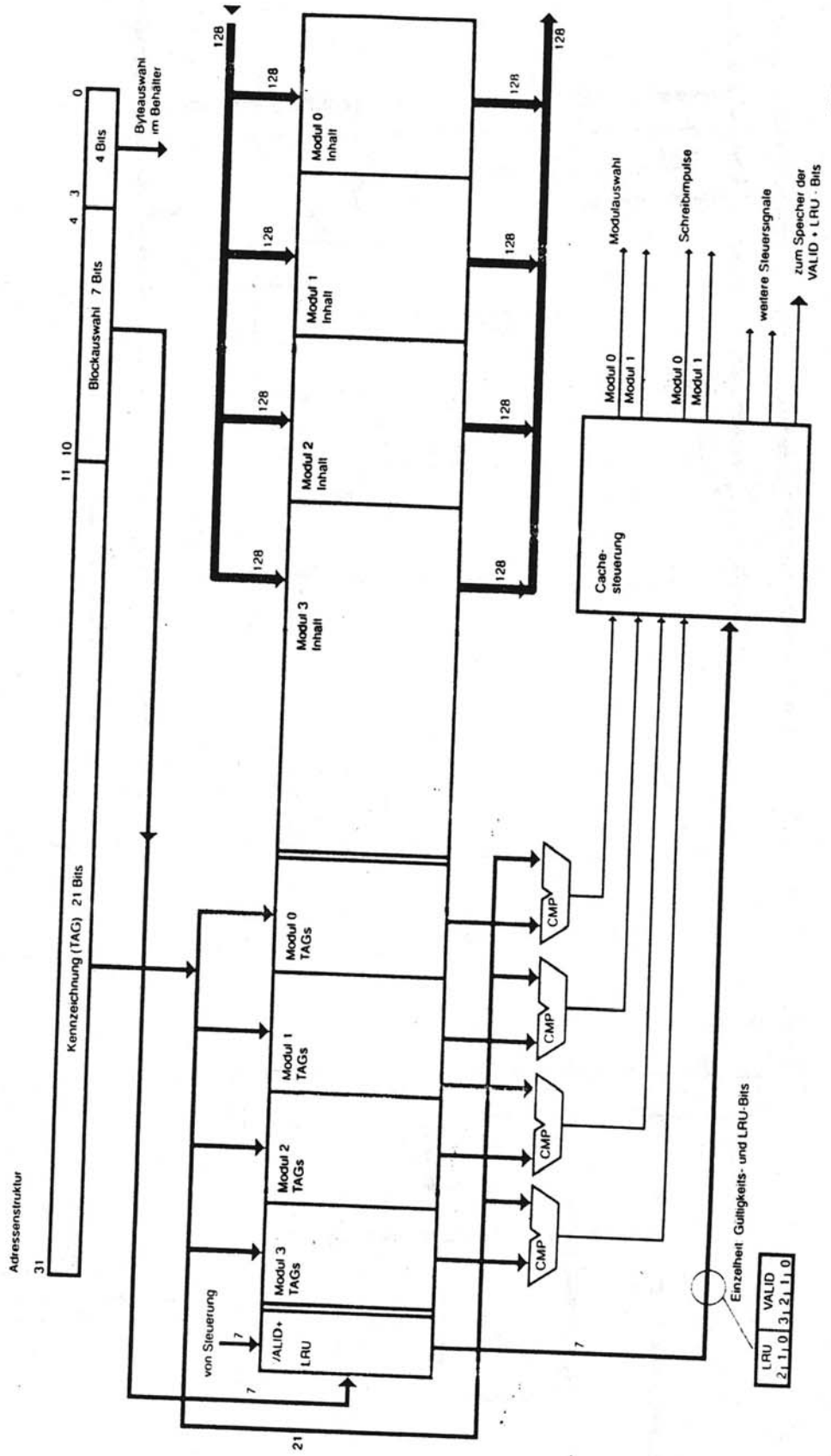
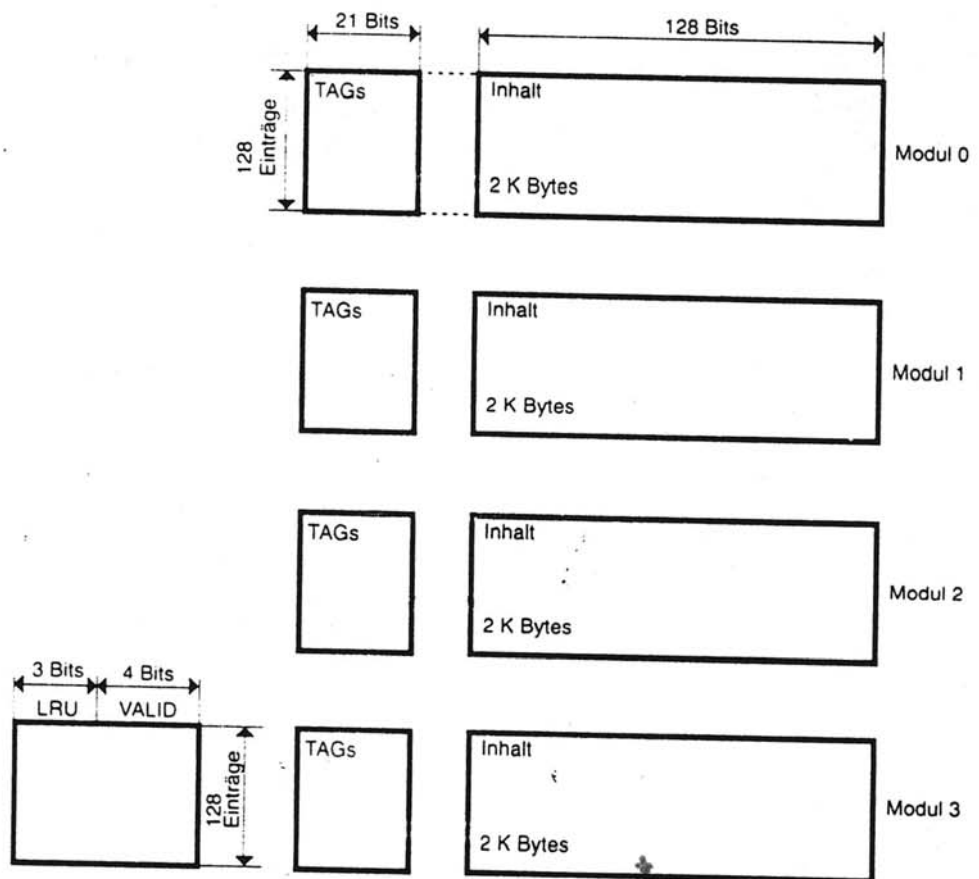


Abbildung 12.2 Cache-Organisation (2): Direktabbildung



929004 5 14

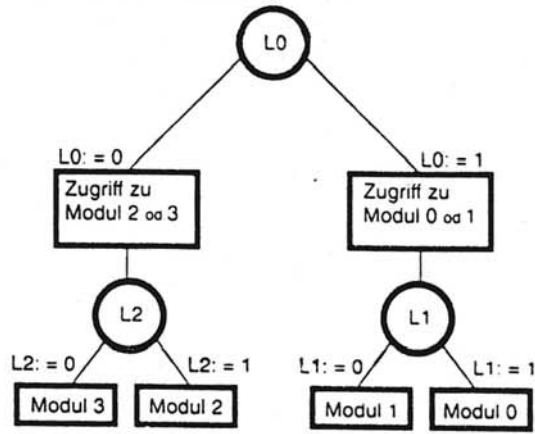
Abbildung 12.3 Cache-Organisation (3): blockassoziativ (der interne Cache des i486)



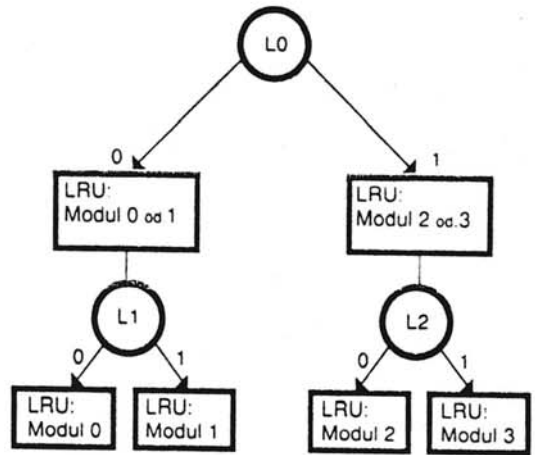
929004-5.15

Abbildung 12.4 Die physische Organisation des internen Cache

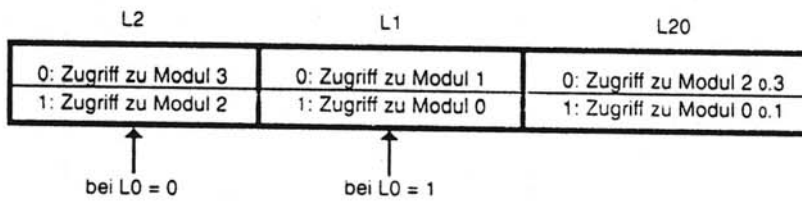
Stellen beim Eintragen bzw. Cache Hit



Abfragen zwecks Modulauswahl



LRU: Least Recently Used (Der betreffende Behälter wird neu belegt.)



929004-5.16

Abbildung 12.5 Nutzung der LRU-Bits

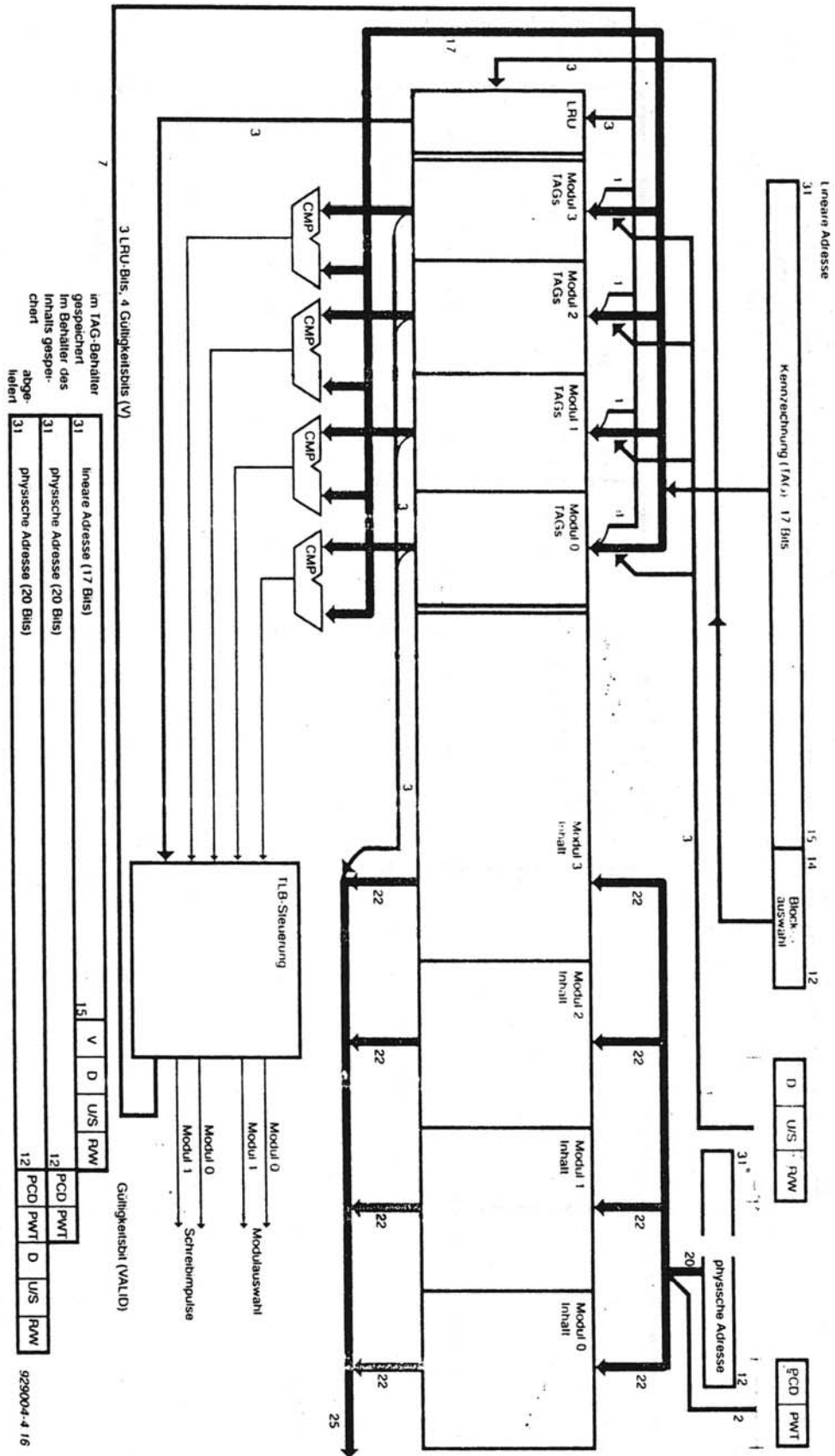
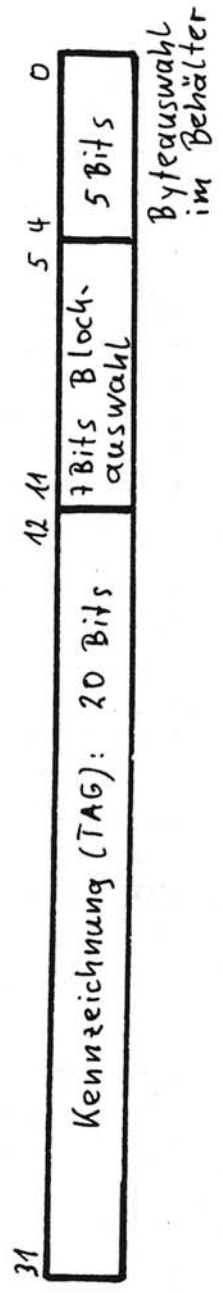
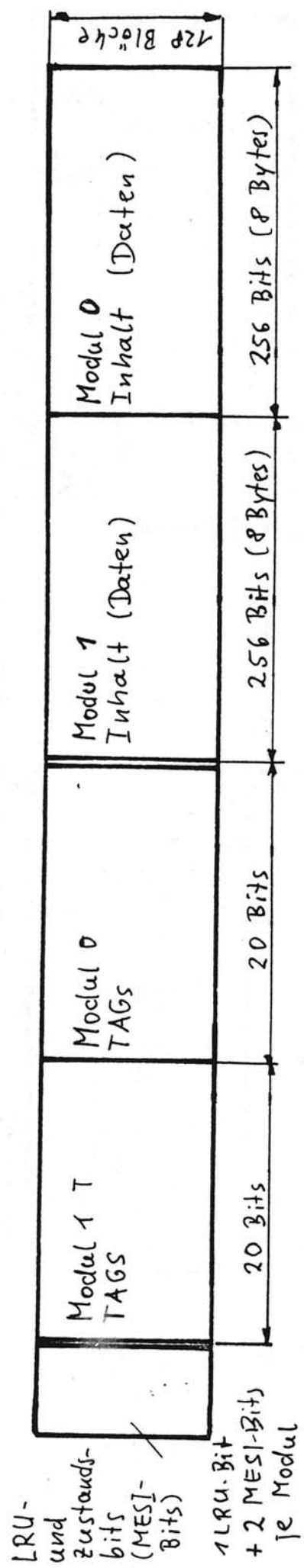


Abbildung 6.7 Der TLB des i486

Adressenstruktur:



### Datencache



### Befehls-cache

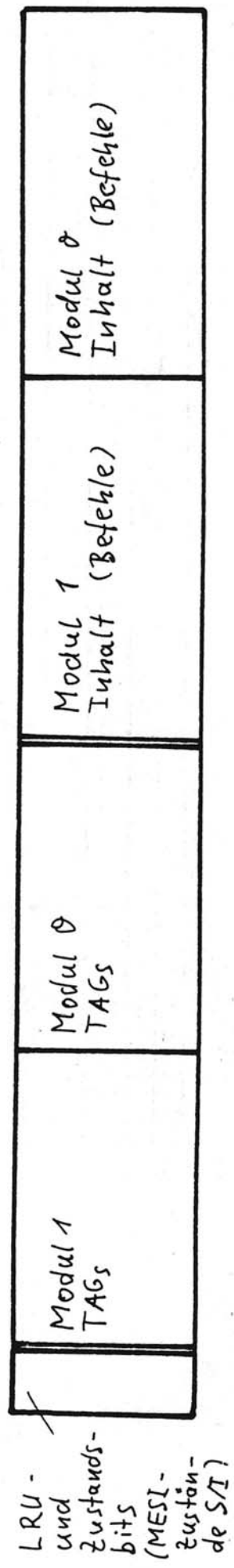


Abbildung B3-4.3 Die internen Caches des Pentium



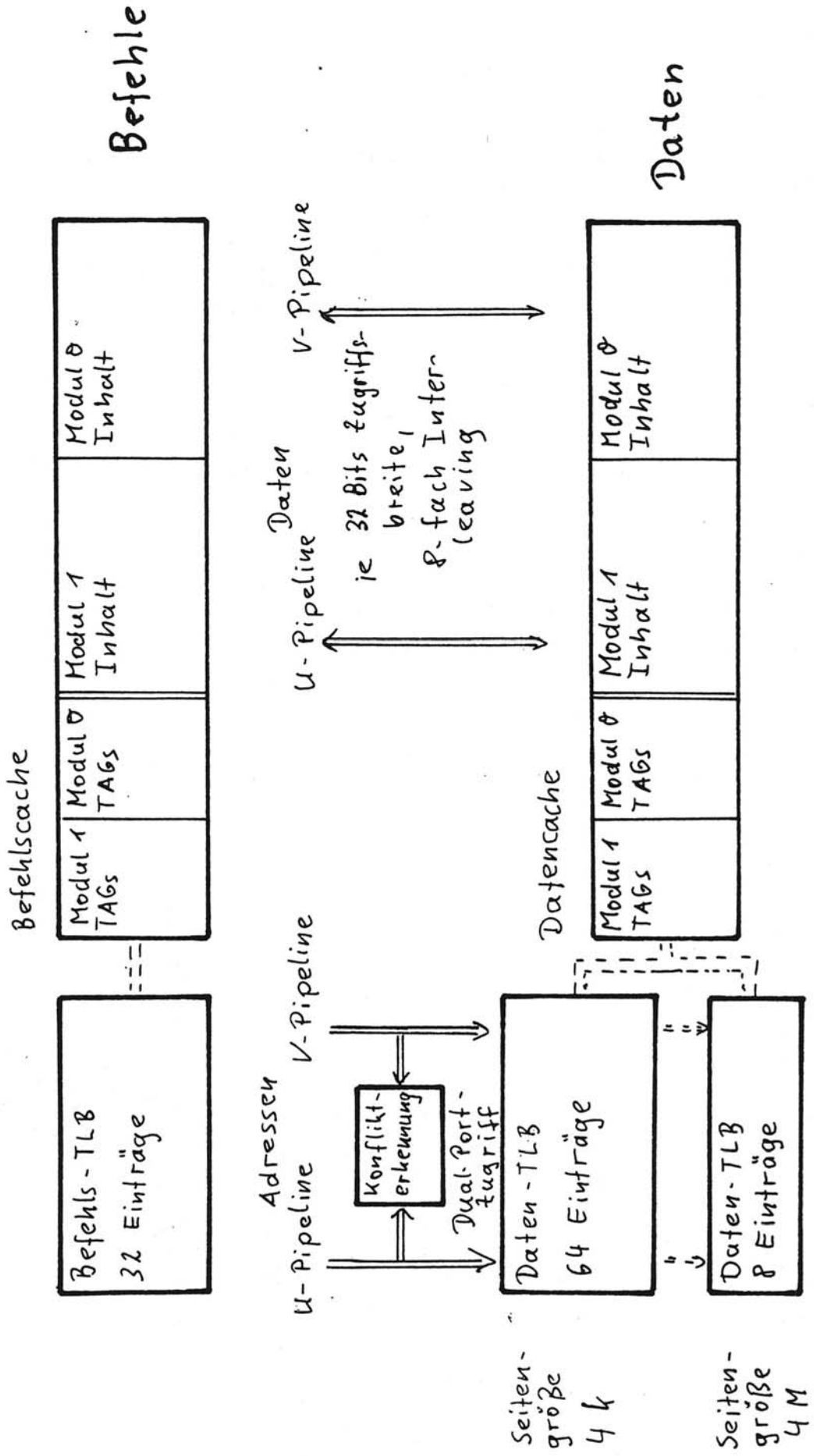


Abbildung B3-4.9 TLBs und Caches im Pentium

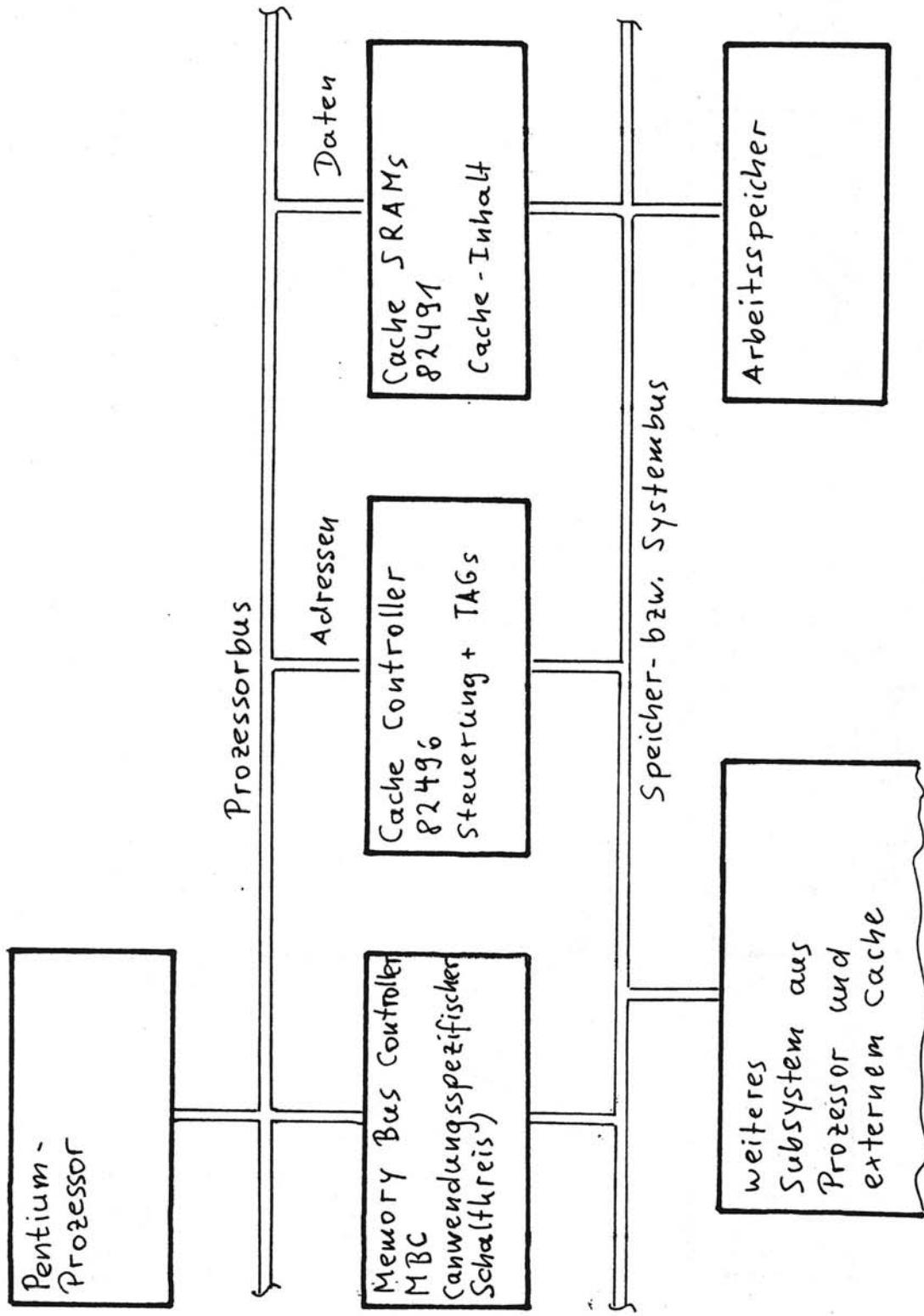


Abbildung B3-4.4 Pentium-Cache-Subsystem (nach Intel)

Table 6-1. First-Level Cache Hit Rates

Cache Configuration			Hit Rate
Size	Associativity	Line Size	
1K	direct	4 bytes	41%
8K	direct	4 bytes	73%
16K	direct	4 bytes	81%
32K	direct	4 bytes	86%
32K	2-way	4 bytes	87%
32K	direct	8 bytes	91%
64K	direct	4 bytes	88%
64K	2-way	4 bytes	89%
64K	4-way	4 bytes	89%
64K	direct	8 bytes	92%
64K	2-way	8 bytes	93%
128K	direct	4 bytes	89%
128K	2-way	4 bytes	89%
128K	direct	8 bytes	93%

rate of a particular configuration is software dependent. However, the table allows a meaningful comparison of the various cache configurations. It also indicates the degree of hardware complexity needed to arrive at a particular cache efficiency. Table 6-1 presents direct-mapped, 2-way, and 4-way set associative caches which are all discussed in the next section.

Program behavior is another important factor in determining cache efficiency. If a program uses a piece of data only once, then the cache may spend all its time thrashing or replacing itself with new data from memory. This is common in vector processing. The processor receives no added efficiency from the cache as main memory is being requested frequently. In such instances, the user can consider mapping the data entries as noncacheable.

Cache system performance can be calculated based on the main memory access time, the cache access time, the miss rate, and the write cycle time.

$C_s$  is defined as the ratio of the cache system access time to the main memory access time.  $C_s$  is a dimensionless number but provides a useful measure of the cache performance.

$$C_a = (1-M)T_c + MT_m$$

$$C_s = C_a/T_m = (1-M)(T_c/T_m) + M = (1-M)C_m - M$$

where:

- $C_a$  = average cache system cycle time averaged over reads and writes
- $T_c$  = cache cycle time
- $T_m$  = main memory cycle time
- $M$  = miss rate = 1-hit rate
- $C_s$  = cache system access time as a fraction of main memory access time
- $C_m$  = cache memory access time as compared to main memory cycle time

386 DX

i486

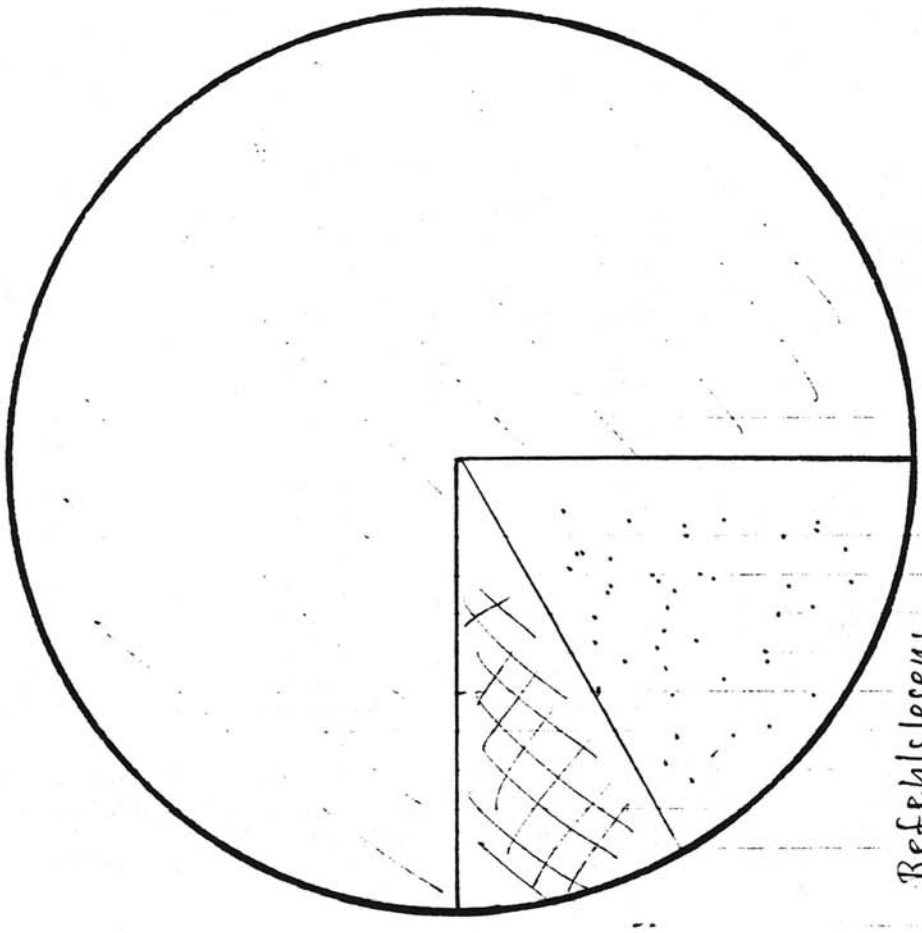
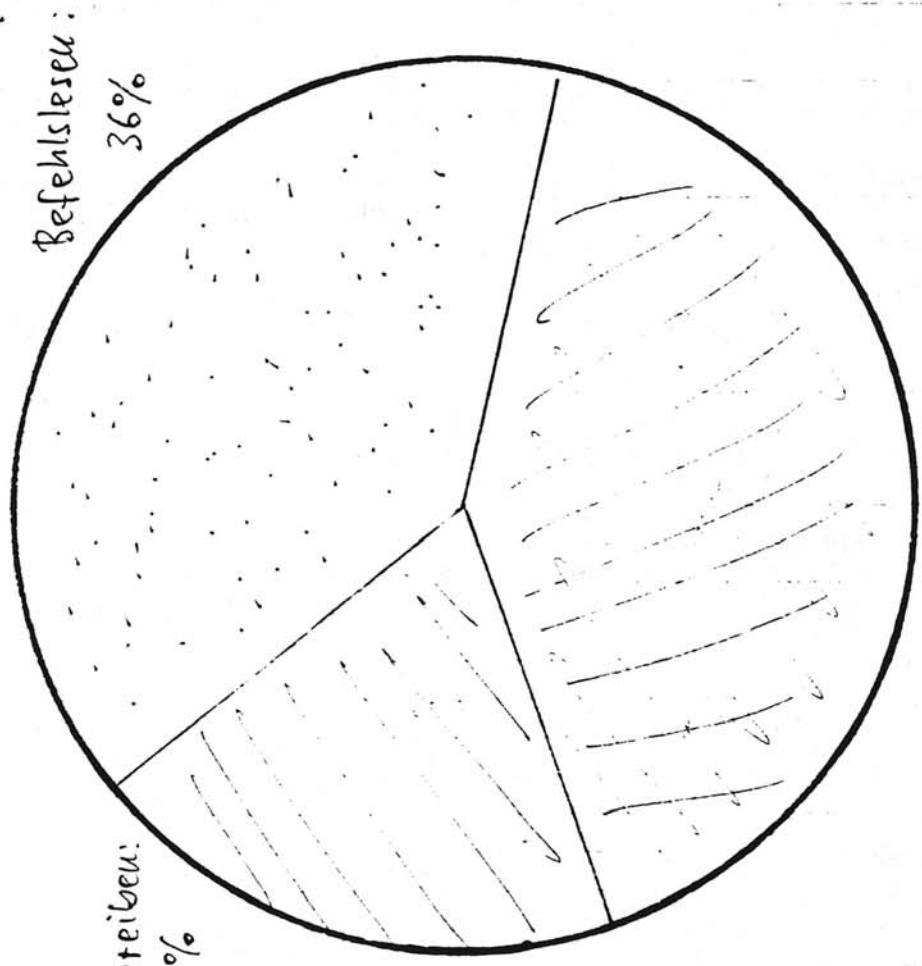
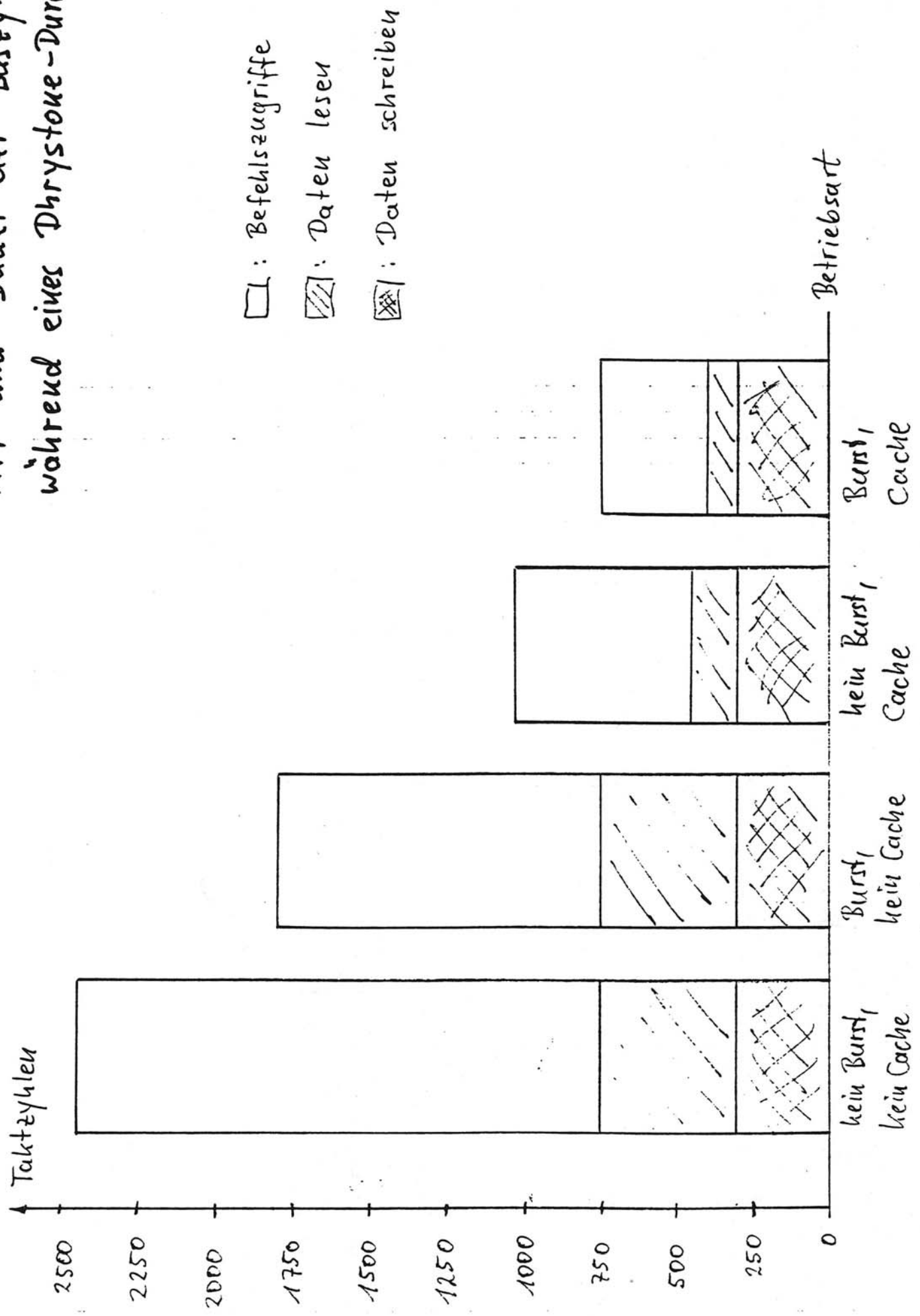
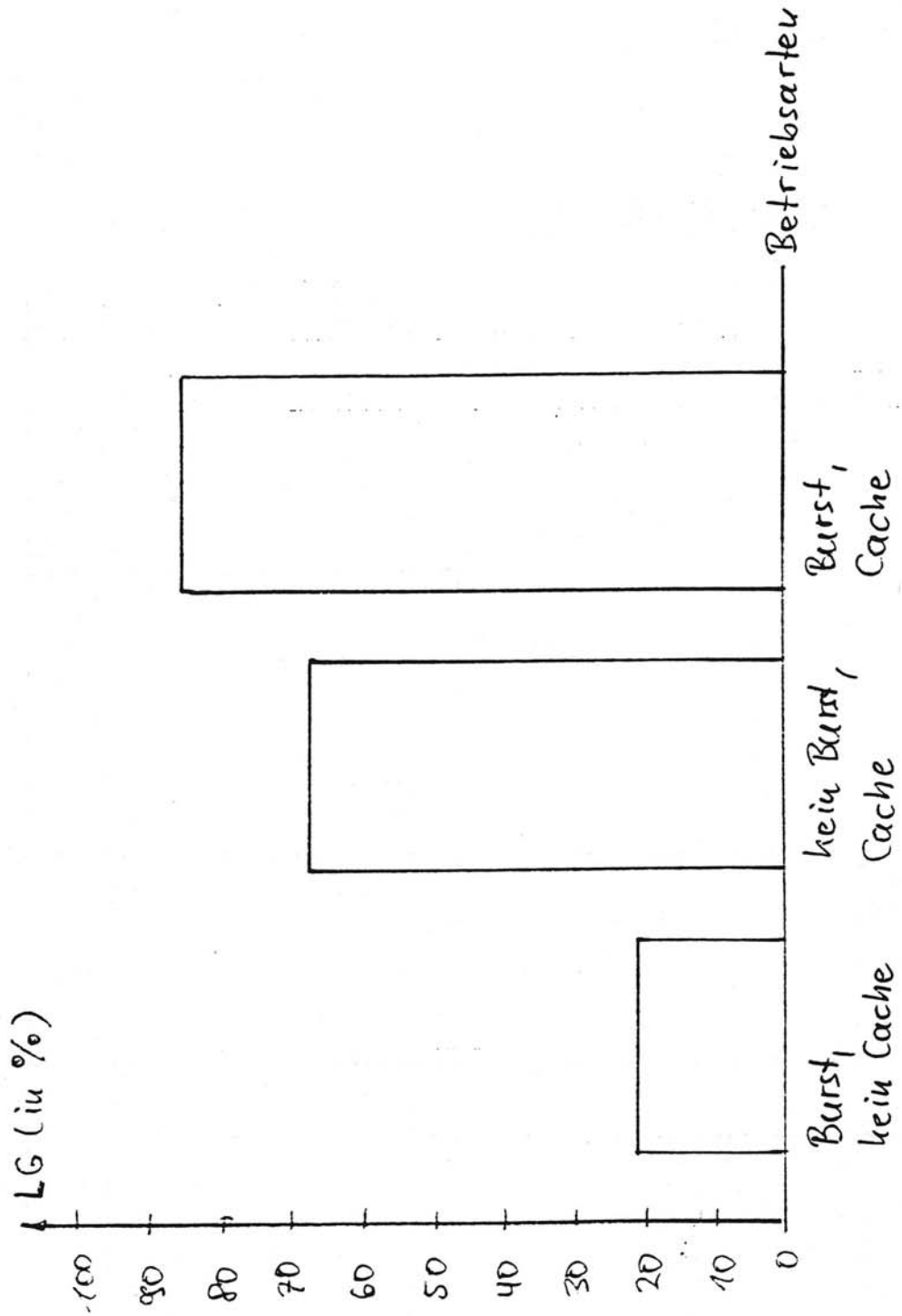


Abbildung 3-3.10 Datenbusausnutzung im Vergleich

(Quelle: Intel)

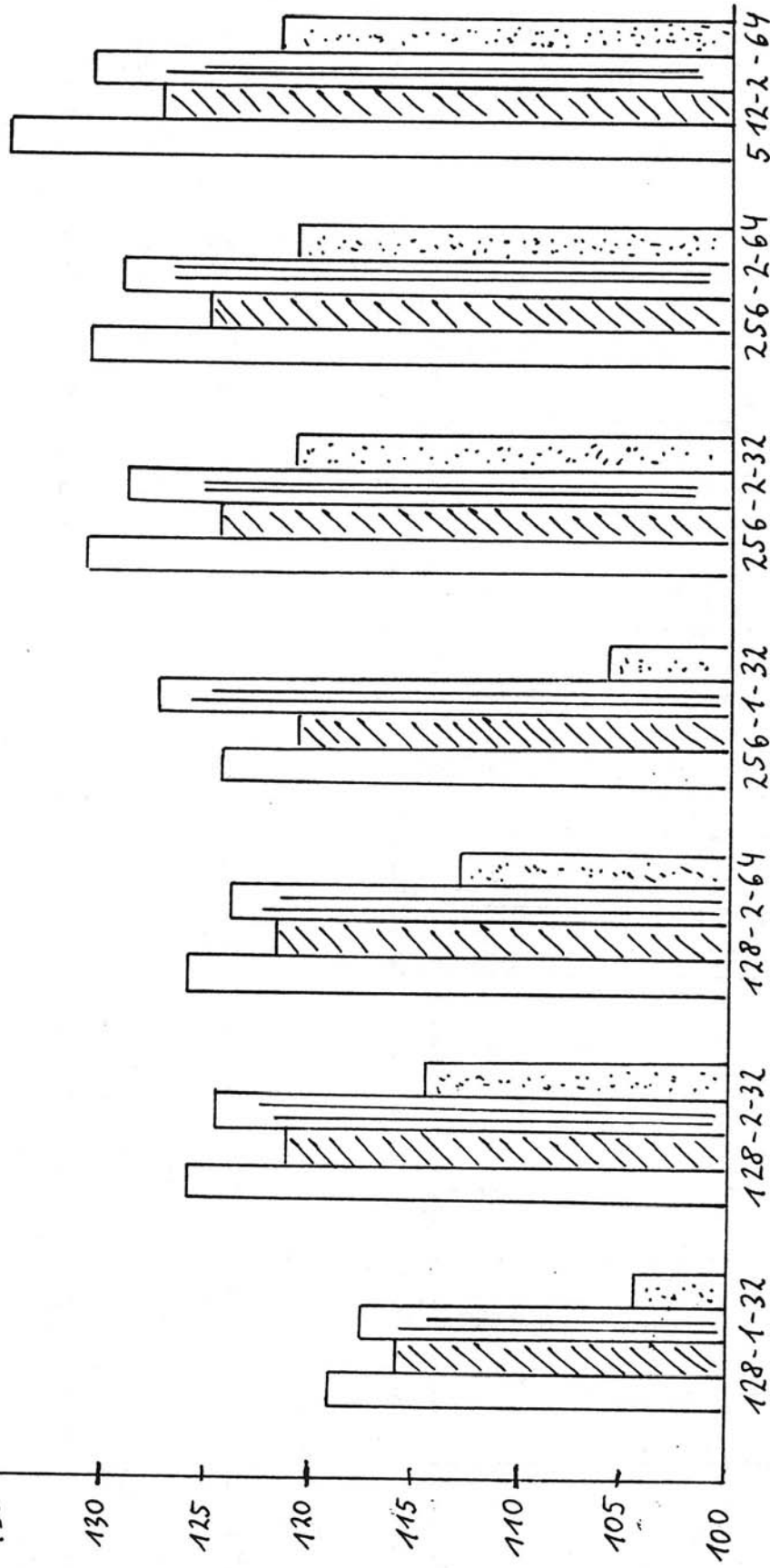
Art und Dauer der Bustypen während eines Thrustoke-Durchlauf





Leistungsgewinne in verschiedenen Betriebsarten

Leistungsgewinn in % gegenüber System ohne externen Cache (= 100%)



Die Angaben: a-b-c bedeuten:

a - Größe des Cache in kBytes

b - Assoziativität (1: Direktabbildung, 2: 2-fach blockassoziativ)

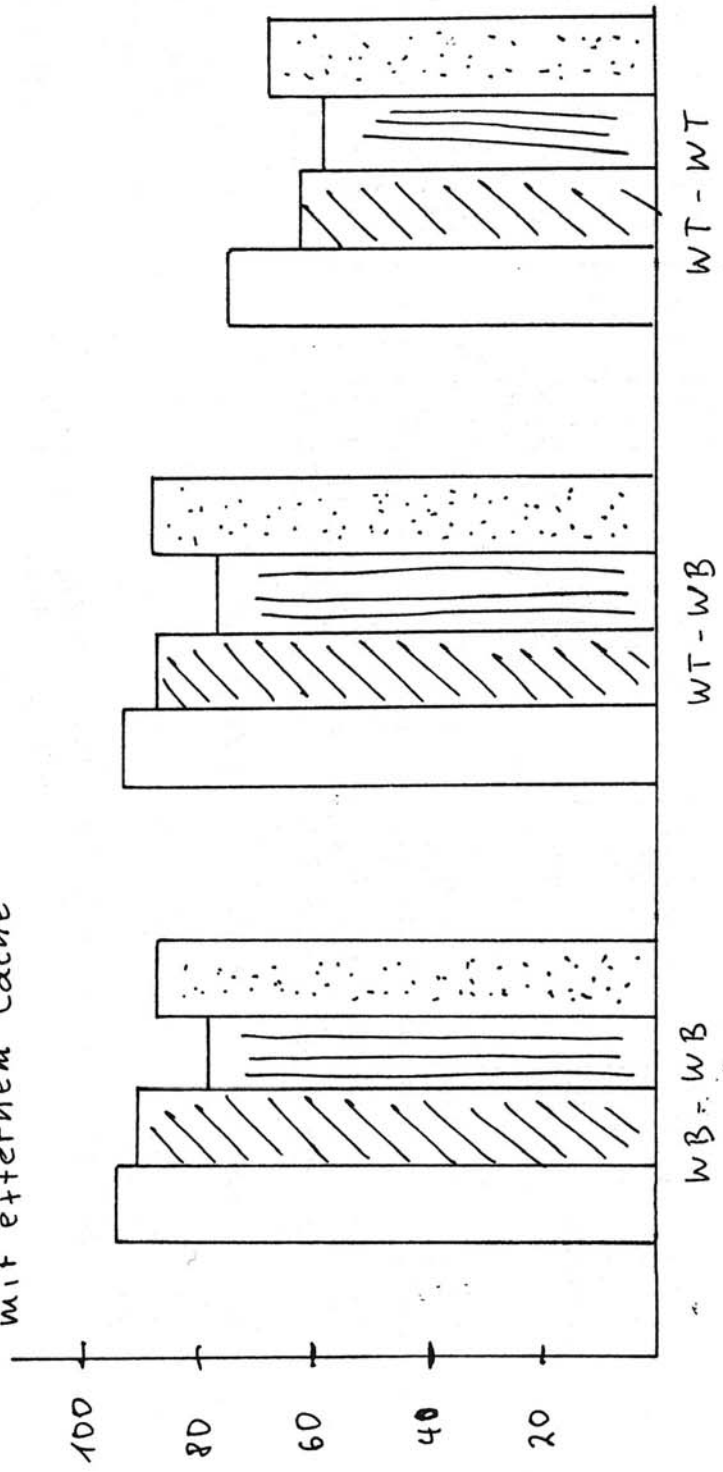
c - Länge eines Cache-Eintrags in Bytes

- Programm Gcc1 (UNIT)
- ▨ Programm Spect (UNIT)
- ▤ WinExcel (Windows)
- ▧ WinWord (Windows)

Abbildung B3-4.5

Auswirkungen verschiedener Cache-Größen und -auslegungen (Quelle: Intel)

Relative Leistung [in %] gegenüber einem "idealen" Pentium-System mit externer Cache



WB: Write Back

WT: Write Thru

Die erste Angabe steht jeweils für den internen, die zweite für den externen Cache.

- Programm Gcc1 (UNIX)
- Programm Spec1 (UNIX)
- WinExcel (Windows)
- WinWord (Windows)

Abbildung B3-4.6 Auswirkungen des gewählten Schreibprinzips (Quelle: Intel)



# Grundsätzliche Leistungsgrenzen des Einzelprozessors

Heutzutage bietet ein moderner Personalcomputer mehr Verarbeitungsleistung als eine mittlere EDV-Anlage vor 10 Jahren. Hardware, die mehrere Millionen Befehle je Sekunde ausführt, ist mittlerweile für vierstellige DM-Beträge in jedem Kaufhaus zu haben. Können wir auch in Zukunft damit rechnen, daß die Verarbeitungsleistung immer weiter erhöht werden wird?

Der klassische Einzelprozessor führt zu einer Zeit nur eine Operation aus. Die Hardware-Grundlage dafür bilden die Verknüpfungsschaltungen in der CPU (die **Operationswerke**). Der Einzelprozessor hat ein einziges Operationswerk, dessen Verknüpfungsschaltungen jeweils so angesteuert werden, daß jeder Operationsbefehl die in seinem Operationscode angegebene Wirkung erbringt (Grundsatz: ein Befehl - eine Operation).

Die Struktur des Operationswerkes bestimmt somit die Leistung, wenn wir annehmen, daß alle "Nebenfunktionen" (Adreßrechnungen, Verzweigungen usw.) nicht zur Verarbeitungszeit beitragen, daß also Operationen lückenlos aufeinander folgen.

Abbildung 3-3.13 zeigt, wie die grundsätzliche Struktur eines Operationswerkes aussieht.

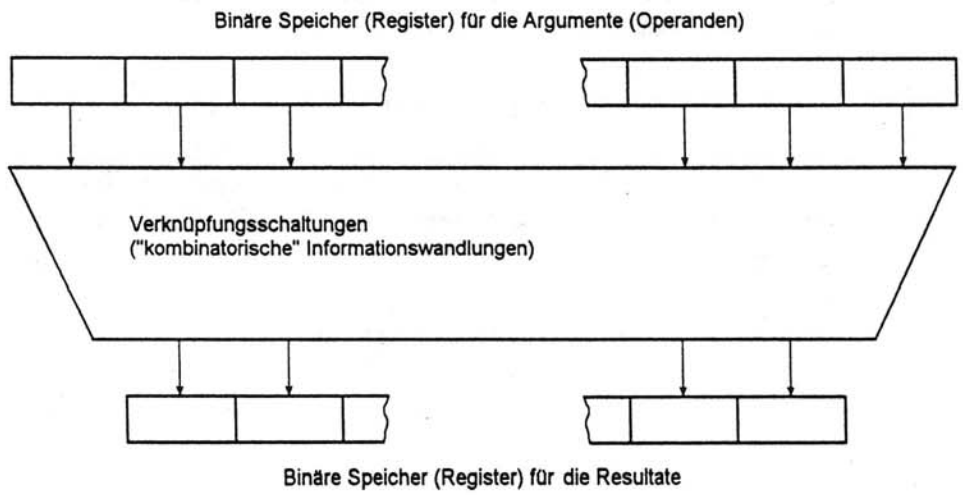


Abbildung 3-3.13  
Grundsätzliche Struktur eines Operationswerkes

An Register, die aus binären Speicherelementen (Flipflops) aufgebaut sind, sind Verknüpfungsschaltungen angeschlossen, die die jeweiligen Operationen ausführen. Die Verknüpfungsschaltungen sind aus Gattern aufgebaut, das heißt aus Schaltmitteln, die aussagenlogische Verknüpfungen verwirklichen (wie UND, ODER, NICHT usw.). Ihnen sind wiederum Register nachgeschaltet, um die Verarbeitungsergebnisse aufzunehmen. Wir nehmen den günstigsten Fall an: die Daten werden in die erste Register-Ebene eingeschrieben, dann verknüpft, und die Resultate werden in die zweite Register-Ebene übernommen. Solche Abläufe (Maschinenzyklen) folgen lückenlos aufeinander. Die Dauer eines solchen Maschinenzyklus ergibt sich somit zu:

$$t_c = s \cdot t_p + t_{acc} + t_{SETUP} + t_L + t_{TOL}$$

Bedeutung der Symbole:

- $t_c$ : Zykluszeit.
- $t_p$ : Verzögerungszeit des einzelnen Gatters.
- $s$ : Schaltungstiefe ( $s = 1, 2, \dots, n$ ); der Wert drückt aus, wieviele Gatter im ungünstigsten Fall nacheinander durchlaufen werden müssen
- $t_{acc}$ : Zugriffszeit zu den gespeicherten Bits in der ersten Register-Ebene (gleichbedeutend: Verzögerungszeit der ersten Register-Ebene, bezogen auf das Ende des vorhergehenden Zyklus).
- $t_{SETUP}$ : Vorhaltezeit für die Resultatübernahme in die zweite Register-Ebene.
- $t_L$ : Summe der Laufzeiten durch alle Verbindungsleitungen (für jedes Resultatbit einzeln zu bestimmen; als  $t_L$  wird der ungünstigste Einzelwert verwendet)
- $t_{TOL}$ : technisch bedingte Zugabe (Toleranz-Ausgleich).

Parameter	Größenordnung (Stand der Technik)
$t_p$ : Gatterverzögerungszeit	CMOS: 400ps .. 2ns, ECL: 350ps, GaAs: 150ps
$t_A$ : Zugriffszeit	Richtwert: $4 t_p$
$t_{SETUP}$ : Vorhaltezeit	Richtwert: $4 t_p$
$t_L$ : Laufzeit durch die Verbindungen	Näherungswert: $1 ns$ je 15 cm
$t_{TOL}$ : Toleranz-Zuschlag	Richtwert: wenige $t_p$ , technologie-abhängig. Beispiel: 10% von $s \cdot t_p$

Tabelle 3-3.2  
 Näherungswerte für leistungsbestimmende technische Parameter

Die maximal mögliche Leistung  $P_{max}$  (in "Befehlen je Zeiteinheit") ergibt sich

folgendermaßen: 
$$P_{max} = \frac{1}{t_c}$$

Dieser Wert ist aber nur unter idealen Betriebsbedingungen zu erreichen, das heißt, vollkommen gleichzeitig mit (parallel zu) den lückenlos aufeinanderfolgenden Operationen müssen zusätzliche Schaltmittel die steuernden Befehle holen und entschlüsseln, die Operanden heranschaffen und die Resultate abtransportieren (Pipelining).

Welche Möglichkeiten gibt es, die Parameter zwecks Leistungssteigerung zu beeinflussen?

$t_p$ : Die Schaltungstechnologie hat nach wie vor beträchtliche Auswirkungen auf die Hardwarekosten: je "schneller" die Technologie, um so bedeutsamer die Zusatzaufwendungen (Kühlung, Stromversorgung usw.), um so geringer der fertigungstechnisch beherrschbare Integrationsgrad.  $t_p$  kann aus physikalischen Gründen nicht beliebig vermindert werden. Einige hundert Picosekunden sind Stand der Technik. Für elektronische Technologien werden bis zu 30 ps angestrebt. Mit optischen Prinzipien hofft man, 1 ps noch auf dem Wege normaler Ingenieurarbeit erreichen zu können; 100 fs werden als äußerste Grenze angesehen.

$t_{acc}$ . hängen ebenfalls direkt mit der Technologie zusammen.

$t_{SETUP}$ .

$t_{TOL}$ : Die Verminderung von  $t_{TOL}$  erfordert hohe Aufwendungen in der Fertigung.

$t_L$ : Die Schaltungstechnologie bestimmt in erster Linie die physischen Abmessungen. Wenn das Schema gemäß Abbildung 3-3.13 auf einen Schaltkreis paßt, ist  $t_L < 1$  ns und - wenigstens bei heutigen Technologien - üblicherweise vernachlässigbar. Kritisch sind die Übergänge zwischen den Schaltkreisen. Die notwendigen Koppelstufen verlangsamen den Informationstransport, und zwar um so mehr, je leistungsfähiger die Basistechnologie ist (bei GaAs hat das Laufzeitverhältnis zwischen internen Verbindungen und solchen, die Schaltkreisgrenzen überschreiten, die Größenordnung 1:10). Optische Verbindungen sind schneller, wobei die Laufzeit nicht durch induktive, kapazitive und ohmsche Belastung verlängert wird.

s: Die Schaltungstiefe wird vom Gatter-Sortiment bestimmt. Entscheidende Parameter sind:

1. Eingangszahl: übliche Werte liegen zwischen 2 und 8..12; bei großer Zahl an Eingängen wird die Schaltungsstruktur größer und daher langsamer. Die Grundstruktur hochintegrierter Schaltkreise ist oft das Gatter (NAND oder NOR) mit zwei Eingängen. Daraus werden alle anderen Digital-schaltungen aufgebaut, allerdings - wie oben angedeutet - mit "Tricks" und Optimierungen auf elektrischer Ebene.
2. Funktionsvielfalt: Die Verknüpfungen werden - auf elektrischer Ebene - mit Transistorstrukturen verwirklicht. Komplizierte Verknüpfungen, die mehr leisten als elementare "Logik"-Gatter, bedingen ausgedehntere Anordnungen mit entsprechend längeren Verzögerungszeiten. Deshalb gibt es in den meisten Technologien nur einen Typ elementarer Verknüpfungen (NAND bzw. NOR).
3. Anzahl der nachschaltbaren Eingänge (fan out). Wenn man einen Ausgang nur mit beispielsweise 4 nachfolgenden Eingängen verbinden darf, das betreffende Signal aber auf 20 Eingänge geführt werden muß, sind Koppelstufen nachzuschalten, wodurch sich die Schaltungstiefe erhöht.

Alle Verknüpfungen werden letztlich durch Boolesche Gleichungen beschrieben, die für jedes Resultatbit die Abhängigkeit von den jeweiligen Argumentbits angeben. Die Schaltungstiefe  $s$  bestimmt sich aus diesen Gleichungen und aus dem jeweiligen Gattersortiment (der **Realisierungsbasis**). Man muß die Gleichungen mit dem Gattersortiment technisch verwirklichen und kann dann die Schaltungstiefe einfach auszählen. Beim Streben nach höchstem Leistungsvermögen hat man somit nur zwei Alternativen:

1. Beschränkung von  $s$  im Hinblick auf einen bestimmten (kleinen) Wert von  $t_c$ , indem nach jeweils  $s$  Gatter-Stufen Speichermittel (Flipflops, Register) eingefügt werden. Kompliziertere Operationen werden so durch eine Kaskaden-Anordnung von Schaltungen gemäß Abbildung 3-3.13 verwirklicht. Das ist das Pipeline-Prinzip
2. Anordnung kombinatorischer Schaltmittel für die gewünschten komplizierten Verknüpfungen, wobei  $t_c$  nach dem jeweiligen  $s$  gewählt wird; das Ziel besteht darin, eine Verarbeitungsaufgabe statt in  $n$  Schritten mit  $t_{c1}$  in einem Schritt mit  $t_{c2}$  auszuführen (wobei  $t_{c2} < n \cdot t_{c1}$  gelten muß, um leistungsmäßig überlegen zu sein).

Beim Pipelining wird das einzelne Resultat nicht schneller gebildet als bei rein kombinatorischer Verknüpfung. Es wird sogar mehr Zeit benötigt: jede eingefügte Speicherstufe fügt zur "rein kombinatorischen" (von  $s$  abhängigen) Durchlaufzeit noch die Verzögerungszeit des Speichergliedes (des Flipflops), dessen Vorhaltezeit sowie einen Toleranz-Zuschlag hinzu. Der Pipeline-Takt muß gemäß diesen Zeitverhältnissen festgelegt werden.

Der Vorteil besteht vielmehr darin, daß bei  $n$  Speicherstufen  $(n-1)$  Operationen parallel mit einem Zeitversatz von jeweils  $t_c$  ausgeführt werden können. Das Prinzip lohnt sich also nur, wenn eine Vielzahl gleichartiger Operationen auszuführen ist (**Vektorrechner**), oder wenn es gelingt, die Pipeline mit mehreren verschiedenartigen Operationen beschäftigt zu halten (**Superpipelining**).

Der alternative Ansatz läuft darauf hinaus, sehr komplexe Verknüpfungen vorzusehen. Diese sind aber nicht immer universell nutzbar (**Spezialrechner**).

Schließlich verbleibt noch die **Parallelverarbeitung**, die Anordnung mehrerer bzw. vieler Verarbeitungseinrichtungen oder kompletter Rechner. Ein entsprechender Zuwachs an nutzbarer Verarbeitungsleistung ist aber nur dann zu erzielen, wenn es gelingt, das jeweilige Anwendungsproblem in gleichzeitig bearbeitbare Teilaufgaben zu zerlegen (**Parallelisierung**).

Von diesen Überlegungen ausgehend wollen wir einige absolute Leistungsgrenzen abschätzen. Wir stützen uns dabei auf folgende Annahmen:

1. Auf elektronischer Grundlage wird eine Gatterverzögerungszeit  $t_p$  von 30 ps erreicht; auf optischer Grundlage von 1 ps.
2. Der Maschinenzyklus wird für eine Schaltungstiefe  $s = 50$  festgelegt; das ist ein Wert "auf der sicheren Seite", in dem Zuschläge für Toleranzen usw., also die Zeiten  $t_{acc}$ ,  $t_{SETUP}$ ,  $t_L$ ,  $t_{TOL}$  bereits berücksichtigt sind.
3. Die einzelne Verarbeitungseinrichtung (Prozessor) wird in sich auf maximale Leistung ausgebildet. Es werden die üblichen Gleitkommaoperationen (Floating Point Operations; FLOP) als leistungsbestimmend angesehen (Leistungsangabe deshalb in MFLOP, GFLOP bzw. TFLOP; 1 MFLOP =  $10^6$  FLOP; 1 GFLOP =  $10^9$  FLOP; 1 TFLOP =  $10^{12}$  FLOP).
4. Für die Vektorverarbeitung werden 16 gleichzeitige Operationen angenommen (2 verkettete Werke zu 8 Pipeline-Stufen; das ist eine Struktur, die für wichtige Anwendungs-Algorithmen sinnvoll genutzt werden kann). Die Anzahl der Operationen je Sekunde ergibt sich somit zu

$$P_{\max_{\text{vect}}} = \frac{16}{t_c}$$

5. Für die Skalarverarbeitung wird angenommen, daß elementare Befehle (wie Addition, Verzweigung usw.) in einem Zyklus ausgeführt werden. Da komplexere Operationen (z. B. die Multiplikation) mehr Zyklen brauchen, werden 2 Zyklen als mittlerer Wert angesetzt. Das ist ein Erfahrungswert aus üblichen Mix-Leistungsbewertungen; die komplexeren Befehle werden weitaus seltener genutzt als die elementaren (Mittelwert beim i486: 1,95 Zyklen).  
Damit ergibt sich die Anzahl der Operationen je Sekunde zunächst zu

$$P_{\max_{\text{scal}}} = \frac{1}{2t_c}$$

- 6. Es wird angenommen, daß (wie beim Pentium) durchschnittlich zwei (skalare) Verarbeitungsbefehle gleichzeitig ausführbar sind. Dies ist ein recht sicherer Wert für den nutzbaren innewohnenden Parallelismus, der durch viele Untersuchungen bestätigt wird. Damit erhalten wir

$$P_{\max_{\text{supscal}}} = \frac{1}{t_c}$$

(Bei Nutzung dieses Grades an innewohnendem Parallelismus führt die Maschine in jedem Zyklus eine Operation aus.)

Die entsprechenden Leistungs-Angaben sind in Tabelle 3-3.3 zusammengefaßt.

Wirkprinzip	elektronisch	optisch
Gatterverzögerungszeit $t_p$	30 ps (0,03 ns)	1 ps (0,001 ns)
Zykluszeit $t_c$ für $s = 50$	1,5 ns	50 ps
Taktfrequenz	≈ 667 MHz	20 GHz
Skalarleistung, 1 Operationswerk	≈ 333 MFLOP	≈ 10 GFLOP
Skalarleistung, 2 Operationswerke	≈ 667 MFLOP	20 GFLOP
Vektorleistung (16-stufige-Pipeline)	≈ 10 GFLOP	320 GFLOP

Tabelle 3-3.3  
Absolute Leistungsgrenzen von Einzelprozessoren (Abschätzung)

Um solche Werte zu erreichen, muß eine Vielzahl komplizierter Probleme erfolgreich gelöst werden. Wichtige Aufgabenkreise sind:

- Beherrschung der Basistechnologien (mehrere Millionen Transistoren auf einem Schaltkreis, Taktfrequenzen um 600 MHz),
- Miniaturisierung des gesamten Hochleistungsrechners,
- Speichermittel hinreichend großer Speicherkapazität (Caches mit hinreichenden Trefferraten) für die angesetzten Zykluszeiten (1,5 ns bzw. 50 ps),
- softwareseitige Ausnutzbarkeit des Systems (Programm-Optimierung, Vektorisierung, Erkennung und Nutzung des innewohnenden Parallelismus).

*Zum Vergleich:* Intel spricht beim Projekt "Micro 2000" von 250 MHz Taktfrequenz ( $t_c = 4 \text{ ns}$ ). Der Einzelprozessor würde dann maximal 250 Millionen Operationen je Sekunde ausführen, ein Schaltkreis mit vier solchen Prozessoren einer Milliarde Operationen.

Die Angaben zum optischen Wirkprinzip beruhen auf Aussagen der Grundlagenforschung zu einzelnen Gattern bzw. elementaren Schaltungen. Fragen der preiswerten - konkurrenzfähigen - Massenproduktion sind noch völlig ungeklärt.