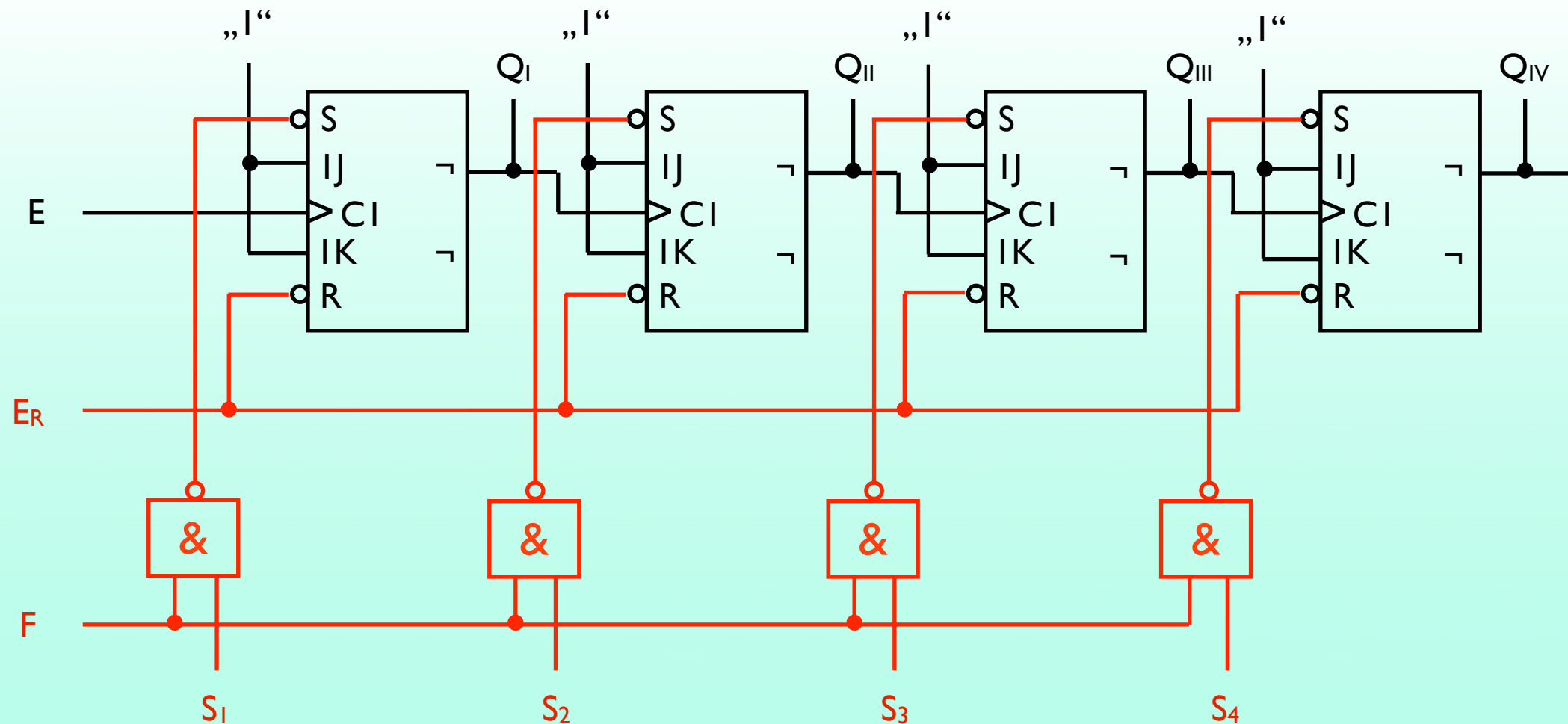


Digitaltechnik 2



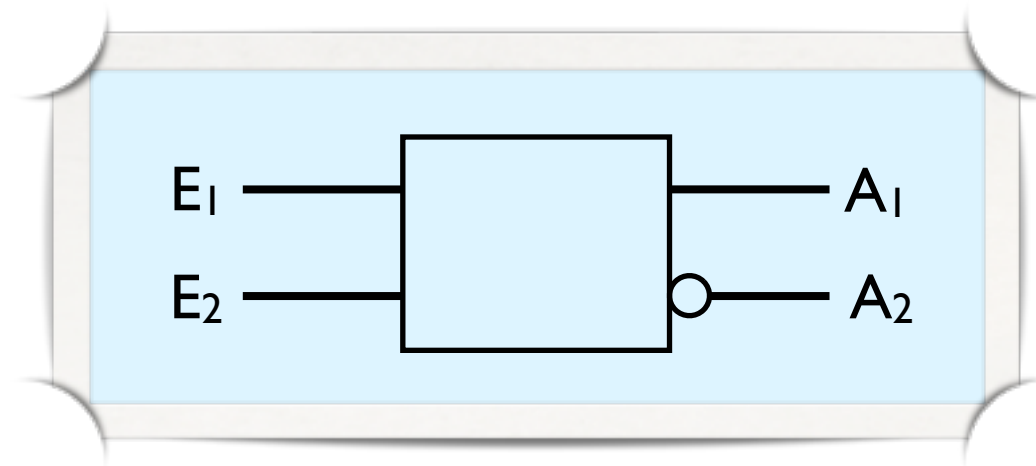
Inhaltsverzeichnis

- 10. Zeitabhängige binäre Schaltungen
- 11. Zähler und Frequenzteiler
- 12. Digitale Auswahl- und Verbindungsschaltungen
- 13. Register- und Speicherschaltungen
- 14. Digital-Analog-Umsetzer, Analog-Digital-Umsetzer
- 15. Rechenschaltungen
- 16. Programmierbare Logik

10. Zeitabhängige binäre Schaltungen

I. Einführung (1/4)

- Flipflop: bistabile Kippstufe mit Speicherwirkung (Herstellung als integrierte Schaltungen)
- Schaltzeichen der Flipflops:

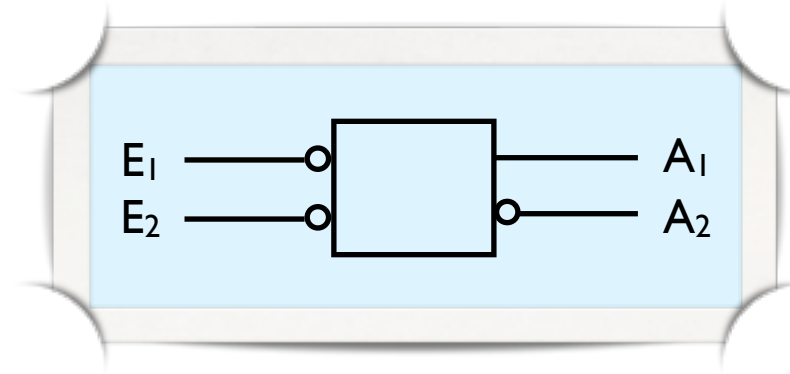


- Konventionen:
 - keine Anschlüsse für U_V einzeichnen
 - entgegengesetzte Ausgangszustände A_1 und A_2
 - Beschreibung der Arbeitsweise mit logischen Zuständen
 - Setzvorgang: $E_1 = 1 \rightarrow A_1 = 1$
 - Rücksetzvorgang: $E_2 = 1 \rightarrow A_2 = 1$
 - keine steuernde Wirkung durch Zustand 0
 - Speicherzustand des Flipflops: Zustand A_1

10. Zeitabhängige binäre Schaltungen

I. Einführung (2/4)

- Steuerung durch 0-Zustände



- Grundstellungen: Bezeichnung im Schaltsymbol

- Zustand $A_1 = 0$ nach Anlegen von U_V

$I = 0$

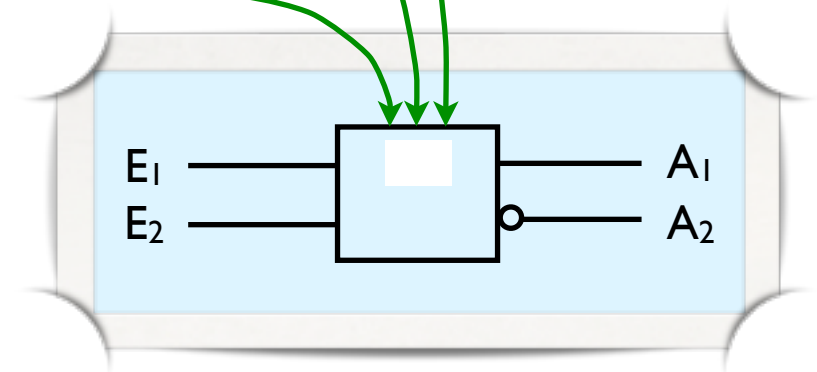
- Zustand $A_1 = 1$ nach Anlegen von U_V

$I = 1$

- nach Anlegen von U_V gleicher Zustand wie vor Abschalten

NV

- bei eindeutiger Grundstellung entfällt Bezeichnung

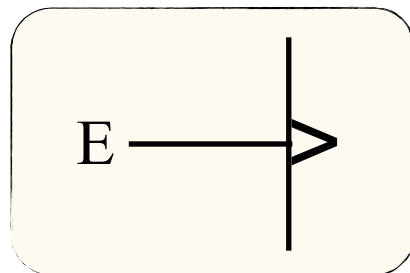
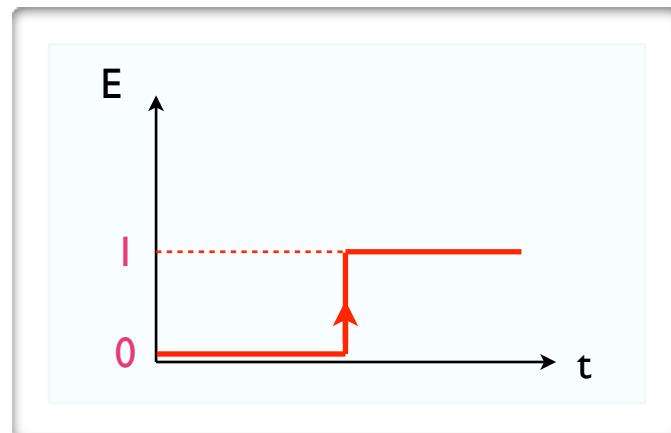


10. Zeitabhängige binäre Schaltungen

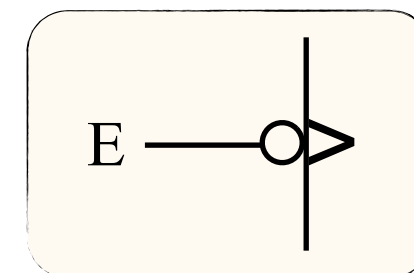
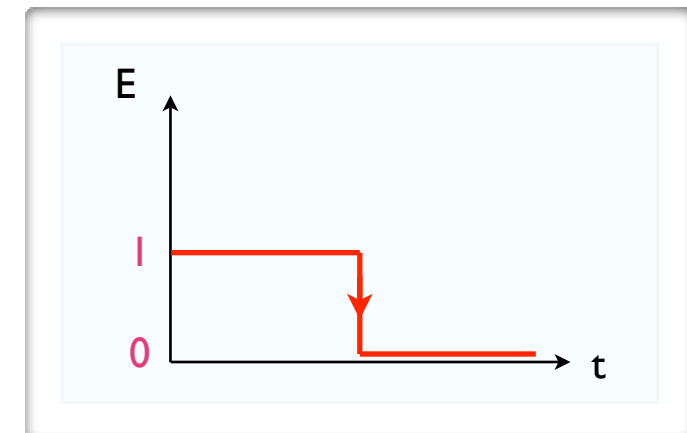
I. Einführung (3/4)

- Ansprechverhalten der Flipflops
 - statische Eingänge (Ansprechen auf Eingangszustand)
 - dynamische Eingänge (Ansprechen auf Eingangszustandsänderung)

ansteigende Flanke



abfallende Flanke



Schaltsymbol

10. Zeitabhängige binäre Schaltungen

I. Einführung (4/4)

- Verknüpfung von Eingängen: Notationen

- $G \rightarrow$ UND-Abhängigkeit
- $V \rightarrow$ ODER-Abhängigkeit
- $C \rightarrow$ Steuer-Abhängigkeit
- $S \rightarrow$ Setz-Abhängigkeit
- $R \rightarrow$ Rücksetz-Abhängigkeit

- Unterscheidung zwischen ...

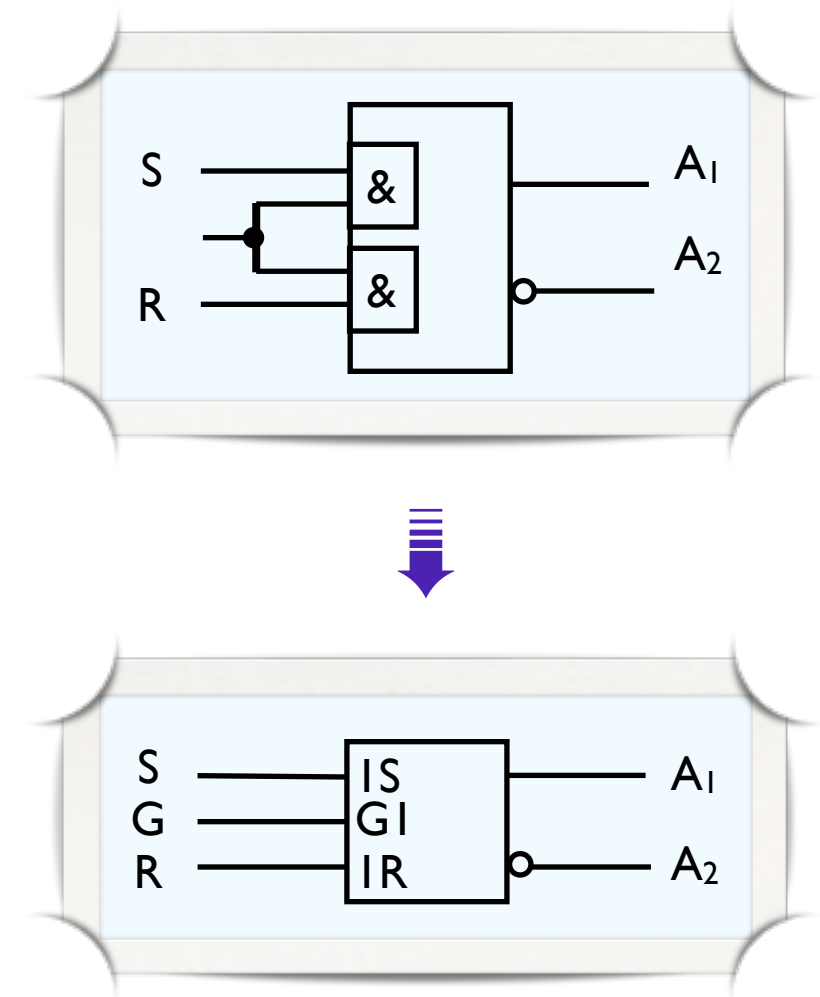
- ... steuernden Eingängen (Buchstabe + Kennzahl)
- ... gesteuerten Eingängen (Kennzahl + Buchstabe)

- Dominierende Eingänge (für gleiche Eingangszustände)

- dominierender S-Eingang

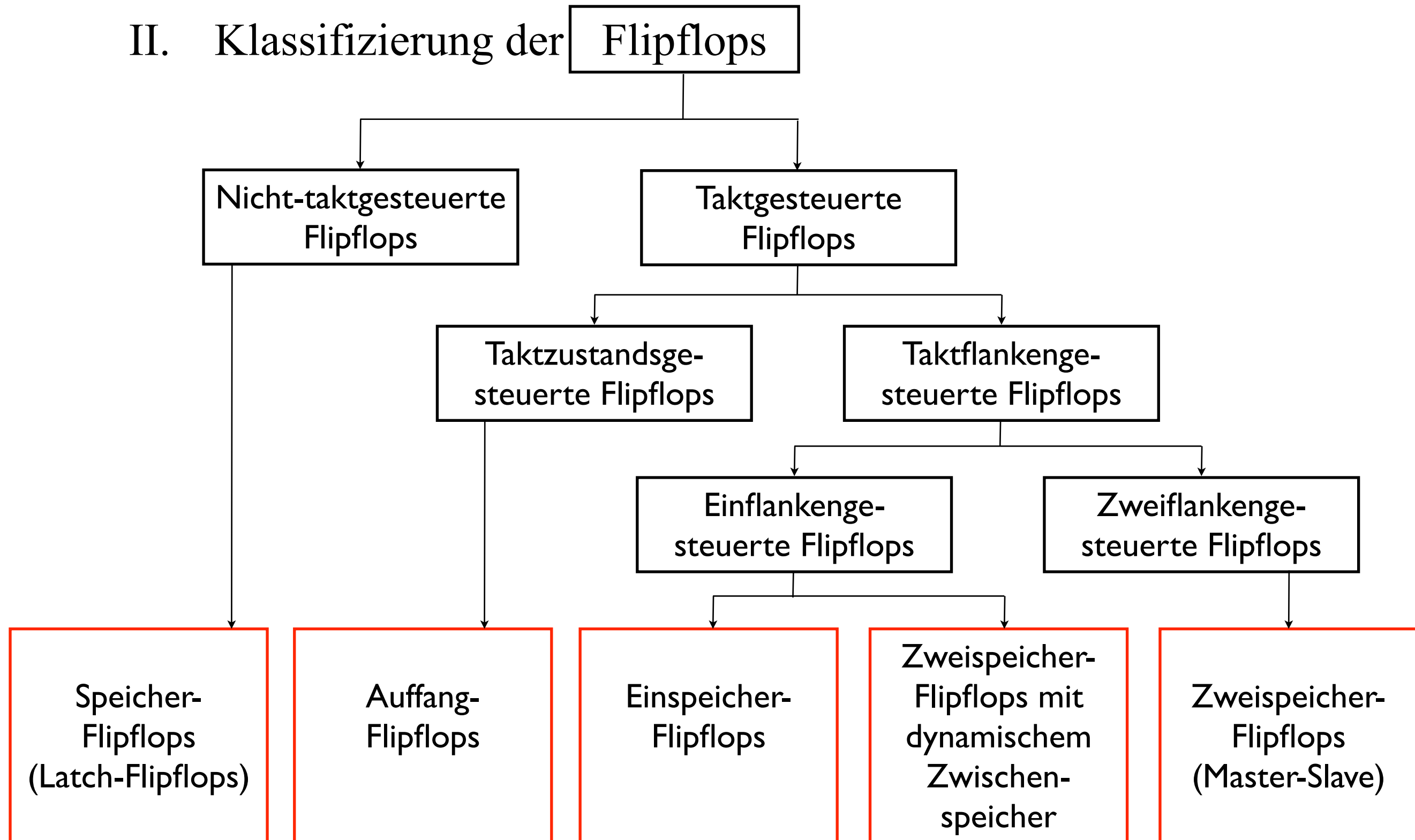


- dominierender R-Eingang



10. Zeitabhängige binäre Schaltungen

II. Klassifizierung der Flipflops

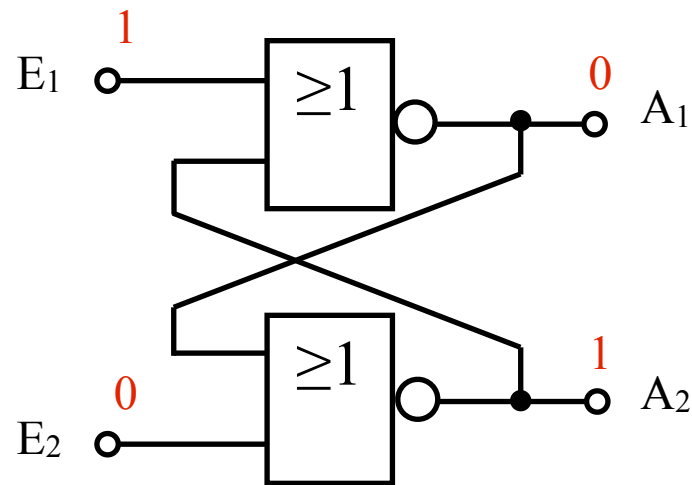


aus: Digitaltechnik; K. Beuth

10. Zeitabhängige binäre Schaltungen

III. Nicht-taktgesteuertes Flipflop (1/2)

- NOR-Flipflop (NOR-Latch)



E₂	E₁	A₁	A₂	
0	0	X	X	Speicherfall
0	1	0	1	
1	0	1	0	
1	1	0	0	irregulär

- Umbenennung der Ausgänge:

- $A_2 \rightarrow Q_1$
- $A_1 \rightarrow Q_2$

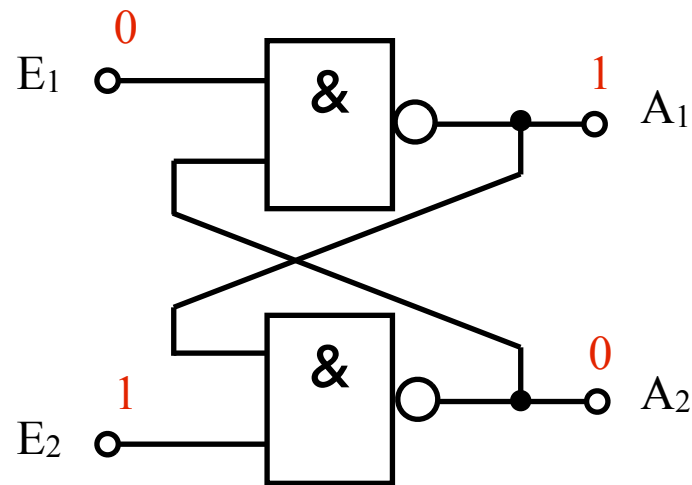
- Bezeichnung:

- SR-Speicher-Flipflop
- RS-Speicher-Flipflop

10. Zeitabhängige binäre Schaltungen

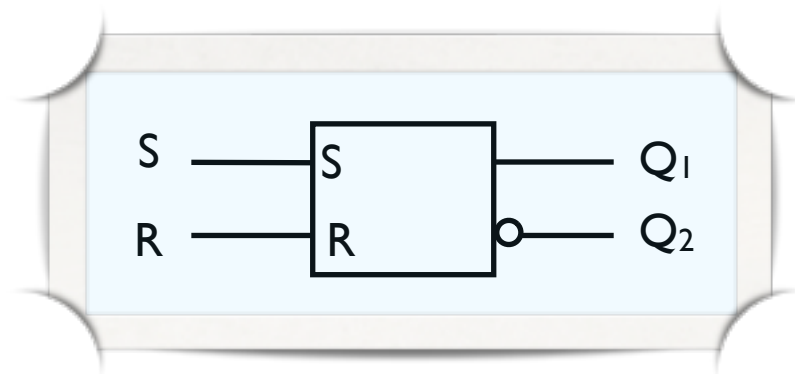
III. Nicht-taktgesteuertes Flipflop (2/2)

- NAND-Flipflop (NAND-Latch)



E ₂	E ₁	A ₁	A ₂	
0	0	1	1	irregulär
0	1	0	1	
1	0	1	0	
1	1	X	X	Speicherfall

- Zusammenfassung: SR-Speicher-Flipflop

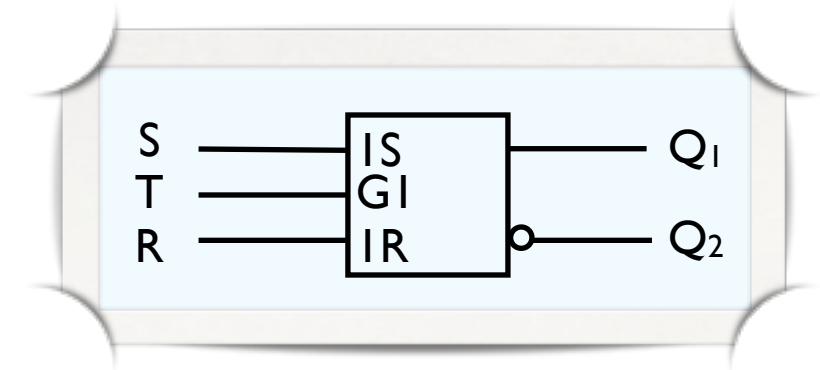


R	S	Q ₁	Q ₂	
0	0	Q _{1(m-1)}	Q _{2(m-1)}	Speichern
0	1	1	0	Setzen
1	0	0	1	Rücksetzen
1	1	1	1	verbotener Fall

10. Zeitabhängige binäre Schaltungen

IV. Taktzustandgesteuerte Flipflops

- SR-Flipflop
 - › AND-Glied vorschalten
 - › Zustand 1 an E₁ bereitet Setzen nur vor
 - › Setzen erfolgt beim Anlegen des Steuereingangs T

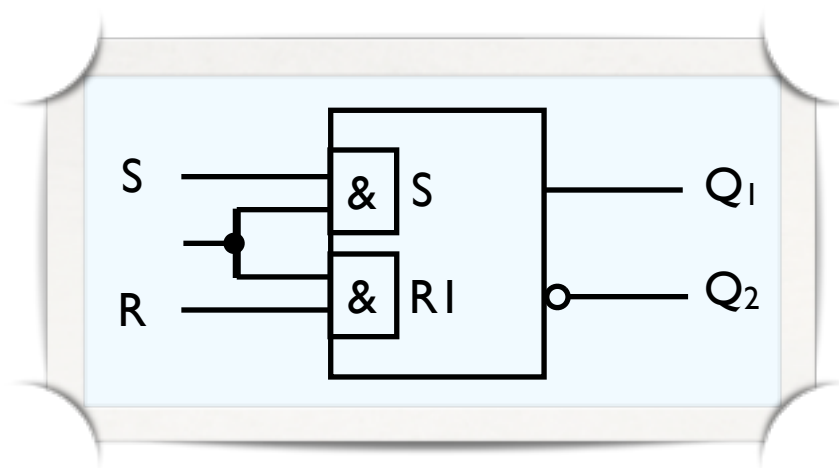
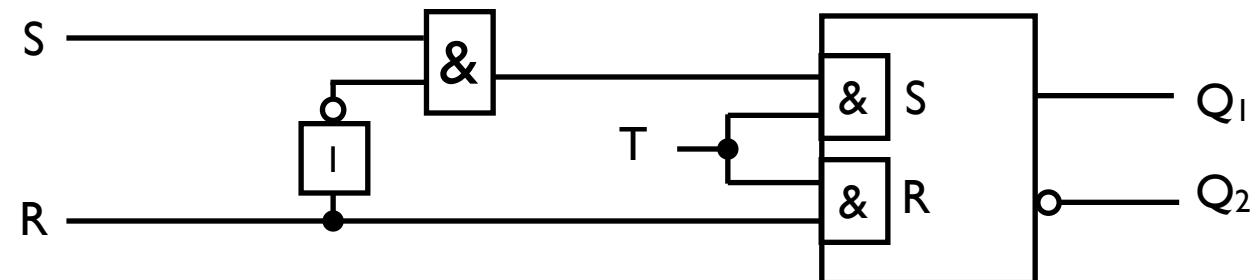


T	R	S	Q ₁	Q ₂	
0	0	0			keine Signal- änderung, Speicher- fälle
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1	1	0	Setzen
1	1	0	0	1	Rücksetzen
1	1	1	—	—	verbotener Fall

10. Zeitabhängige binäre Schaltungen

IV. Taktzustandgesteuerte Flipflops

- SR-Flipflop mit dominierendem R-Eingang
 - Forderung: $Q_1=0$ bei $S=R=1$ (Handling der verbotenen Fälle)
 - besondere Beschaltung des SR-Flipflops nötig



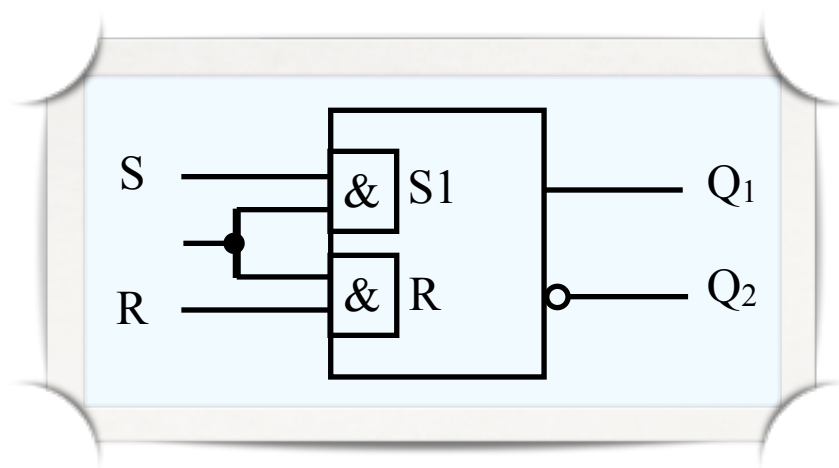
t_n		t_{n+1}
R	S	Q_1
0	0	Q_{1n}
0	1	1
1	0	0
1	1	0

10. Zeitabhängige binäre Schaltungen

IV. Taktzustandgesteuerte Flipflops

- SR-Flipflop mit dominierendem S-Eingang
 - Forderung: $Q_1=1$ bei $S=R=1$ (Handling der verbotenen Fälle)
 - besondere Beschaltung des SR-Flipflops nötig

Übung: Erstellen der Beschaltung

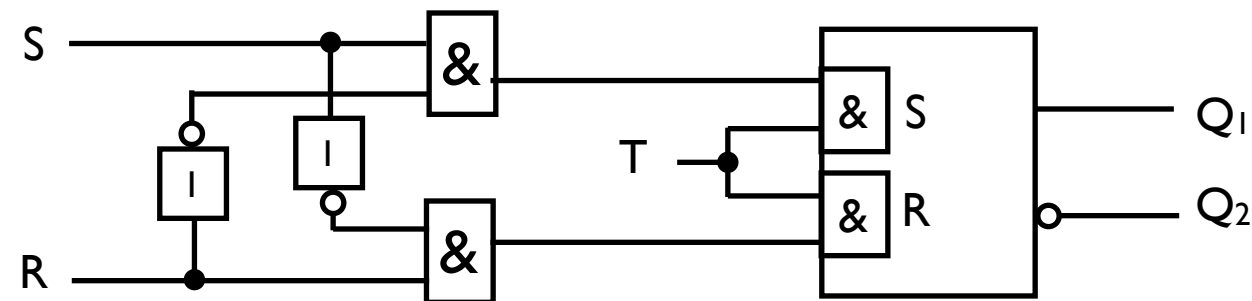


t_n		t_{n+1}
R	S	Q_1
0	0	Q_{1n}
0	1	1
1	0	0
1	1	1

10. Zeitabhängige binäre Schaltungen

IV. Taktzustandgesteuerte Flipflops

- E-Flipflop
 - nicht häufig verwendet
 - Speicherfall für $S=R=1$

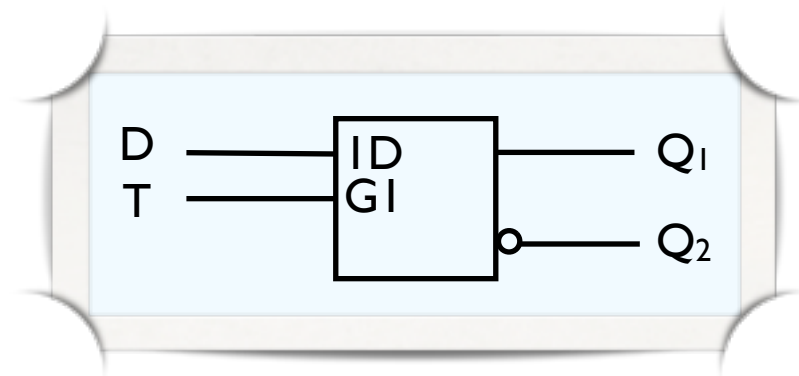
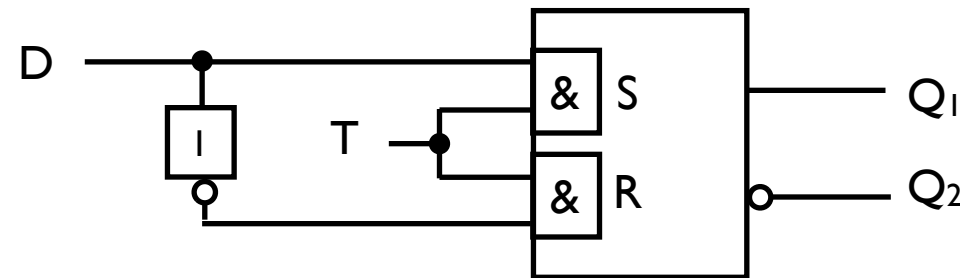


t_n		t_{n+1}
R	S	Q_1
0	0	Q_{1n}
0	1	1
1	0	0
1	1	Q_{1n}

10. Zeitabhängige binäre Schaltungen

IV. Taktzustandgesteuerte Flipflops

- D-Flipflop
 - Verzögerungs-Flipflop (Delay-Flipflop; Verzögerung des Eingangssignals bis Taktsignal anliegt)
 - kein R-Eingang

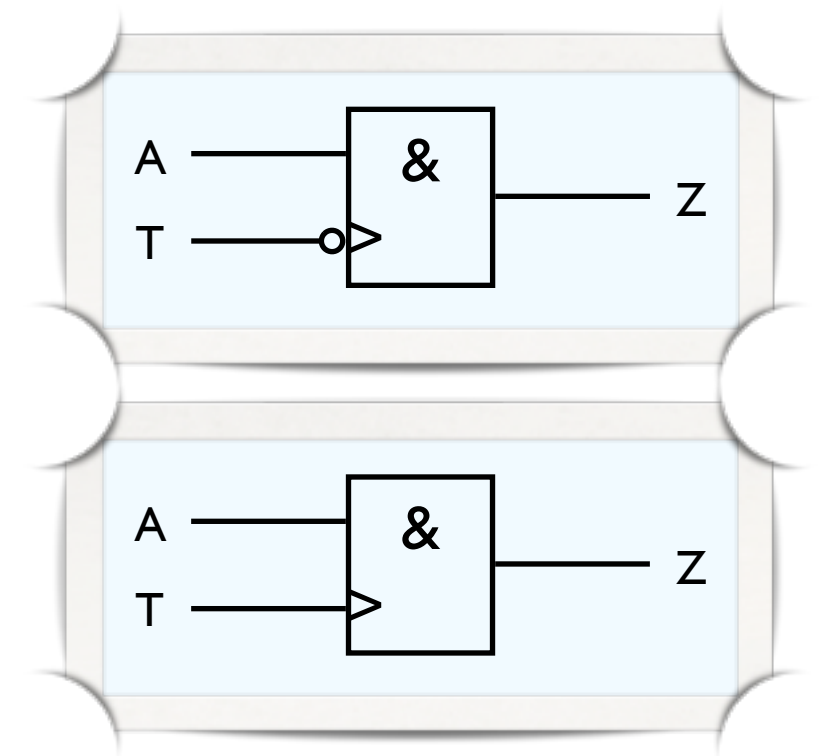


t_n	t_{n+1}
D	Q ₁
0	0
1	1

10. Zeitabhängige binäre Schaltungen

V. Taktflankengesteuerte Flipflops

- Vorteil
 - synchrone Schaltung mehrerer Flipflops möglich
 - größere Störsicherheit
- Impulsglieder ...
 - ... sind notwendig für Taktflankensteuerung
 - ... besitzen statischen sowie dynamischen Eingang
 - ... arbeiten wie UND-Glieder
 - ... liefern einen negativen Ausgangsimpuls Z bei statischem Eingang A=1 und abfallender Flanke des dynamischen Eingangs T
 - ... zweiter Art liefern positive Ausgangsimpulse Z bei statischem Eingang A=1 und ansteigender Flanke des dynamischen Eingangs T

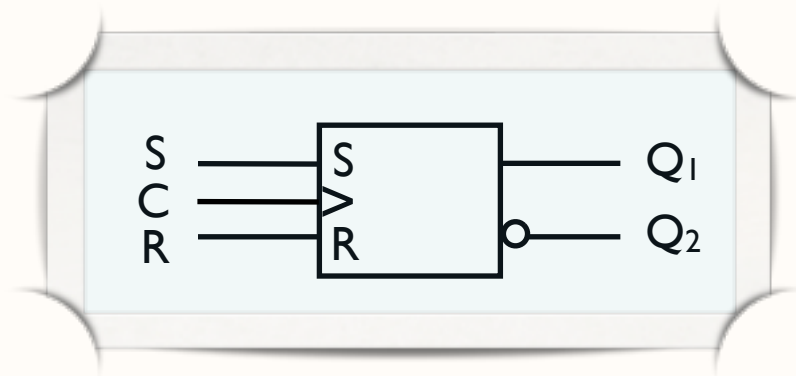
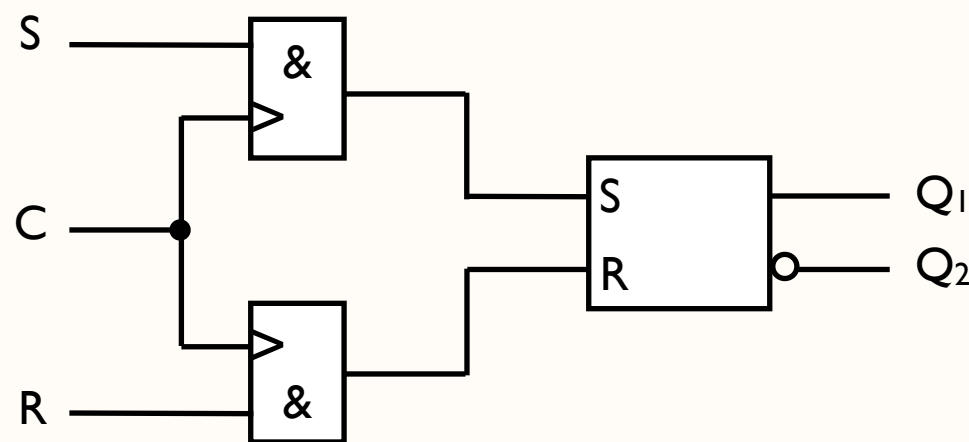


10. Zeitabhängige binäre Schaltungen

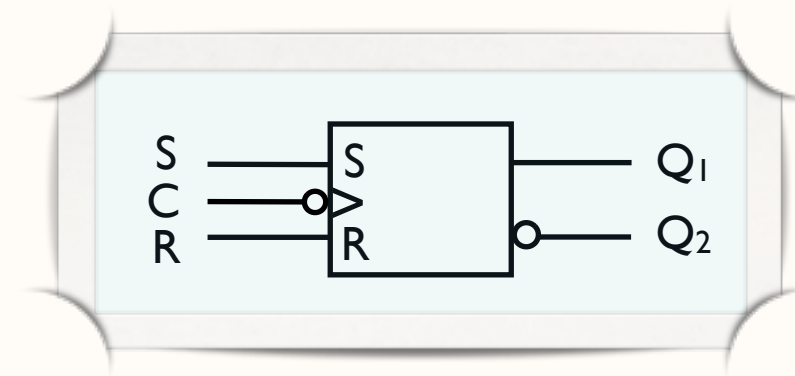
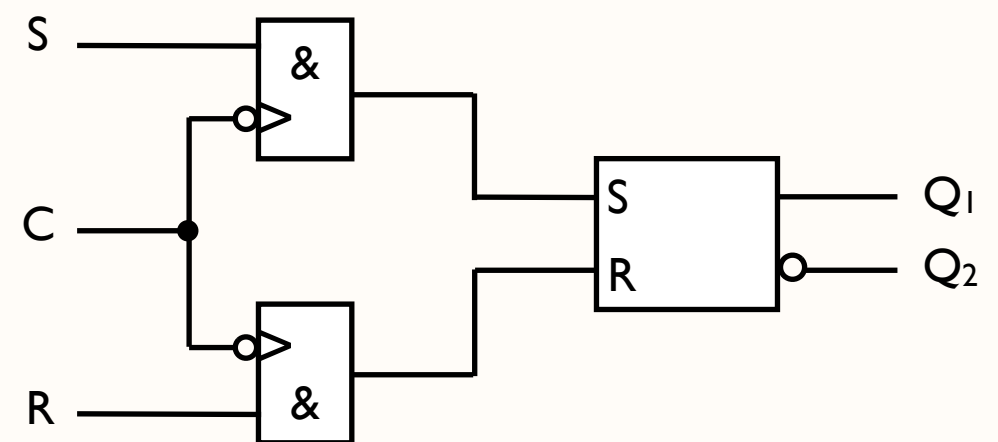
V. Taktflankengesteuerte Flipflops

- Einflankengesteuerte SR-Flipflops
 - Ersetzen der UND-Glieder im taktzustandgesteuerten SR-FF durch Impulsglieder (gleiche Wahrheitstabelle)
 - Bezeichnung des Takteingangs mit C (Clock)

ansteigende Flanke



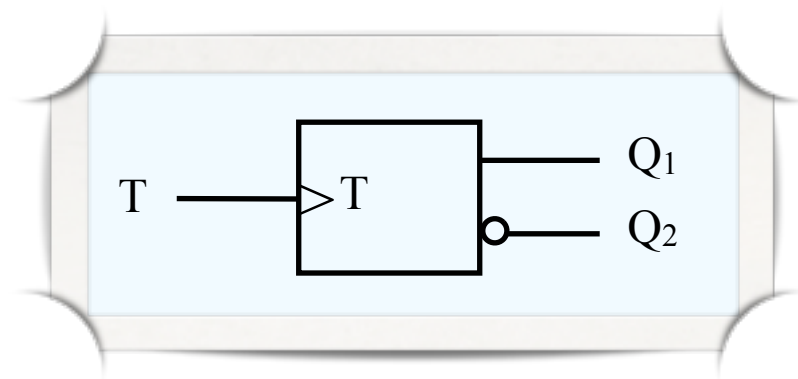
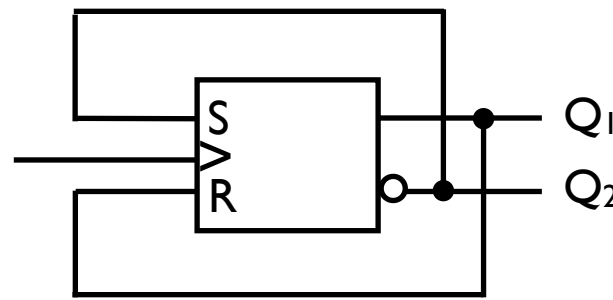
abfallende Flanke



10. Zeitabhängige binäre Schaltungen

V. Taktflankengesteuerte Flipflops

- Einflankengesteuerte T-Flipflops
 - › Änderung des Ausgangszustands bei jeder Taktflanke (ansteigend oder abfallend)
 - › Bezeichnung: Trigger-Flipflop (T-Flipflop)



t_n	t_{n+1}
Q_1	Q_1
0	1
1	0

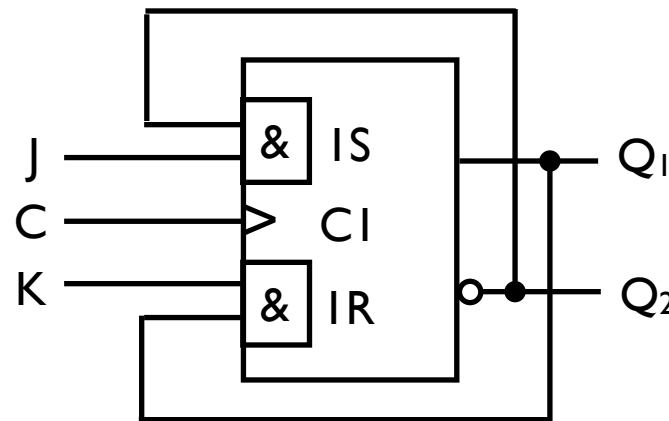
- › Erweiterung: T-Flipflop, das über einen zusätzlichen Eingang gesperrt oder freigegeben werden kann
→ Übung

10. Zeitabhängige binäre Schaltungen

V. Taktflankengesteuerte Flipflops

- Einflankengesteuerte JK-Flipflops

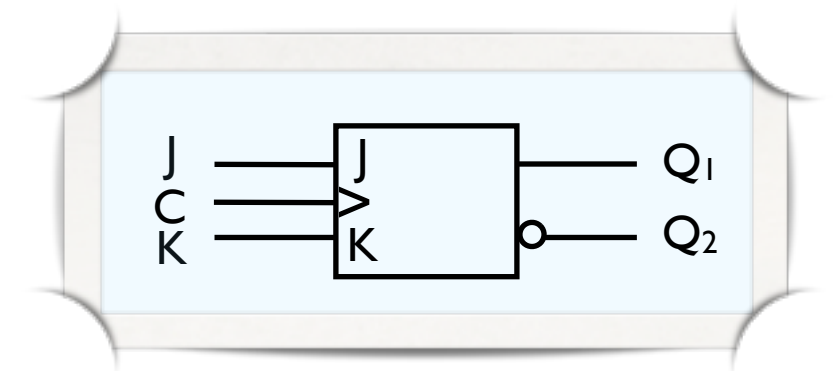
- › Setzen und Rücksetzen wie beim SR-FF
- › verbotener Fall → Kippen des Ausgangs
- › Bezeichnung J-K willkürlich



- › Haltezeit

- ab Schaltzeitpunkt (bei TTL 1,5V) müssen Eingangssignale gewisse Zeit anliegen → Haltezeit
- anschließend sind Änderungen des Eingangssignals wirkungslos
- d.h. Störanfälligkeit besteht nur in der Haltezeit
→ je kürzer die Haltezeit, desto geringer die Störanfälligkeit

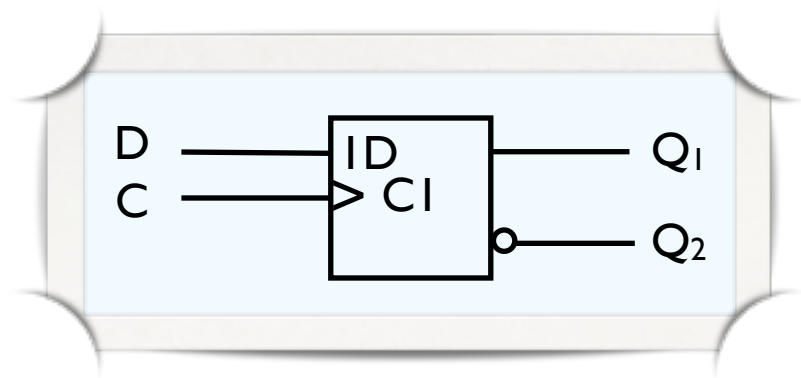
t_n		t_{n+1}
K	J	Q_1
0	0	Q_{1n}
0	1	1
1	0	0
1	1	$\neg Q_{1n}$



10. Zeitabhängige binäre Schaltungen

V. Taktflankengesteuerte Flipflops

- Einflankengesteuerte D-Flipflops
 - gleicher Aufbau wie taktzustandsgesteuertes D-FF
 - Unterschied in der Ansteuerung



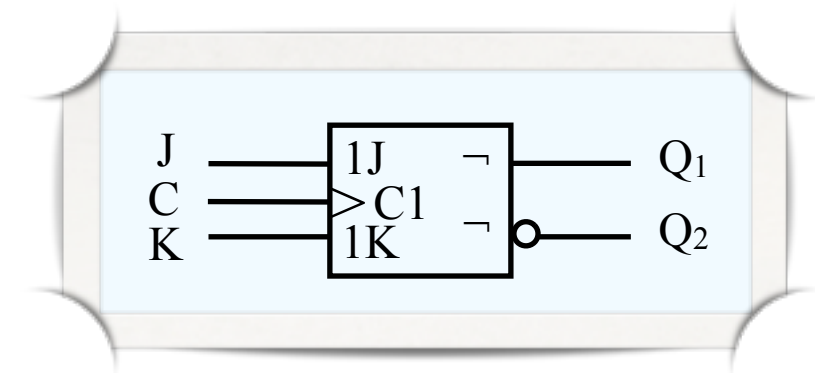
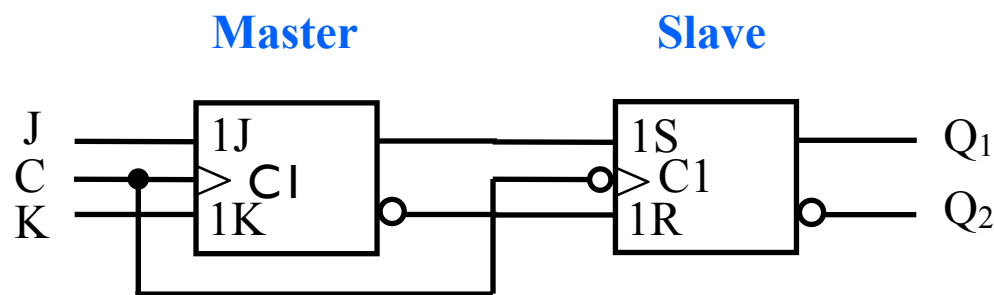
t_n	t_{n+1}
D	Q₁
0	0
1	1

- Verwendung von einflankengesteuerten D-Flipflops in Schieberegistern (→ Kapitel 13)

10. Zeitabhängige binäre Schaltungen

V. Taktflankengesteuerte Flipflops

- Zweiflankengesteuerte JK-Flipflops
 - Master-Slave-Flipflop
 - Kippen des Ausgangs bei $J=K=1$ gefordert \rightarrow Master-Flipflop muss JK-Flipflop sein
 - Slave-Flipflop kann ein SR-Flipflop sein
 - Im Schaltzeichen erfolgt lediglich Angabe der Taktflanke, mit der die Information aufgenommen wird.

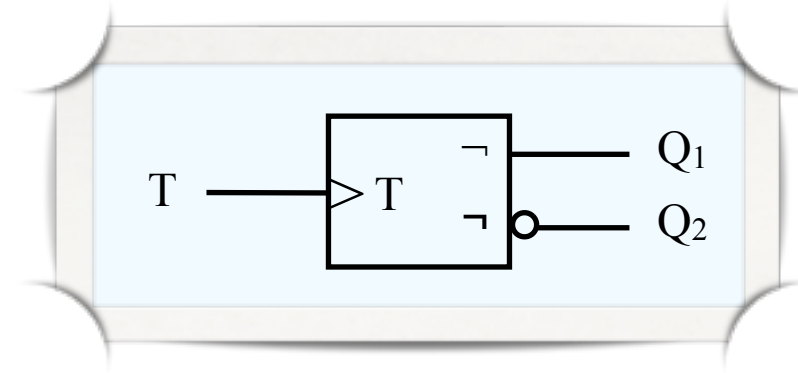
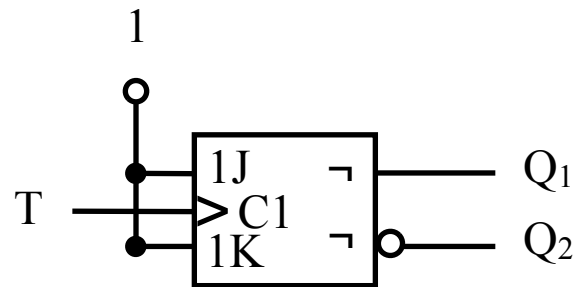


10. Zeitabhängige binäre Schaltungen

V. Taktflankengesteuerte Flipflops

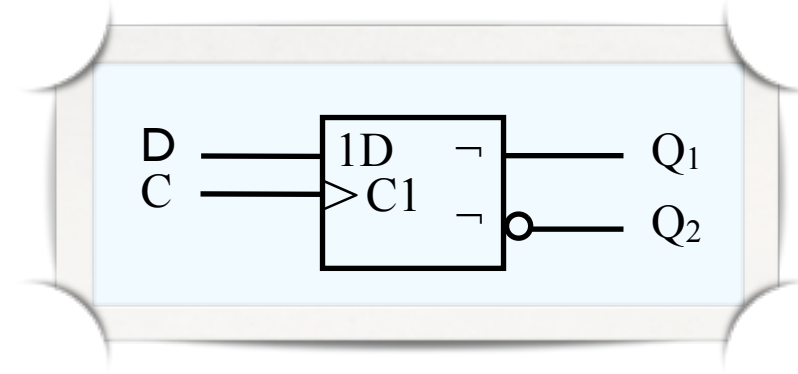
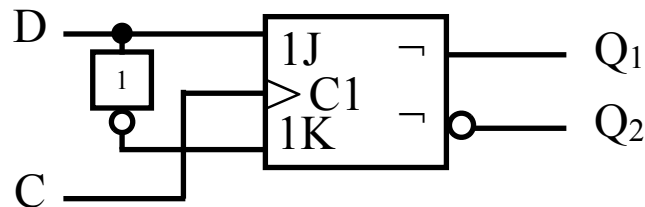
- T-Master-Slave-Flipflop

- Entstehung aus JK-Master-Slave-FF



- D-Master-Slave-Flipflop

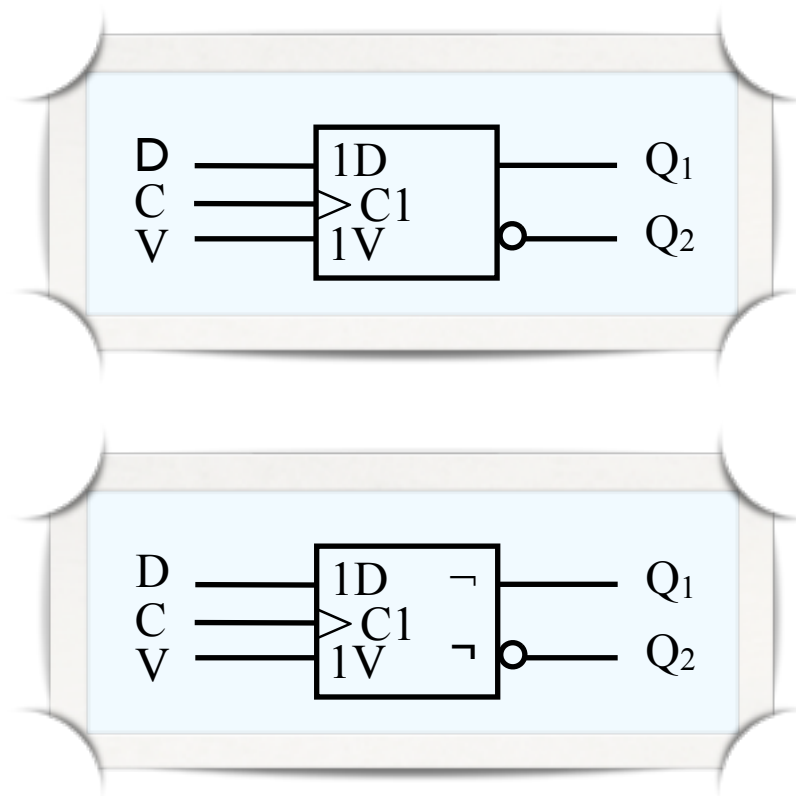
- ebenfalls Entstehung aus JK-Master-Slave-FF



10. Zeitabhängige binäre Schaltungen

V. Taktflankengesteuerte Flipflops

- DV-Flipflop
 - entspricht D-Flipflop, falls Eingang V=1 (Vorbereitungseingang)
 - verfügbar als ein- und zweiflankengesteuertes (Master-Slave-) Flipflop

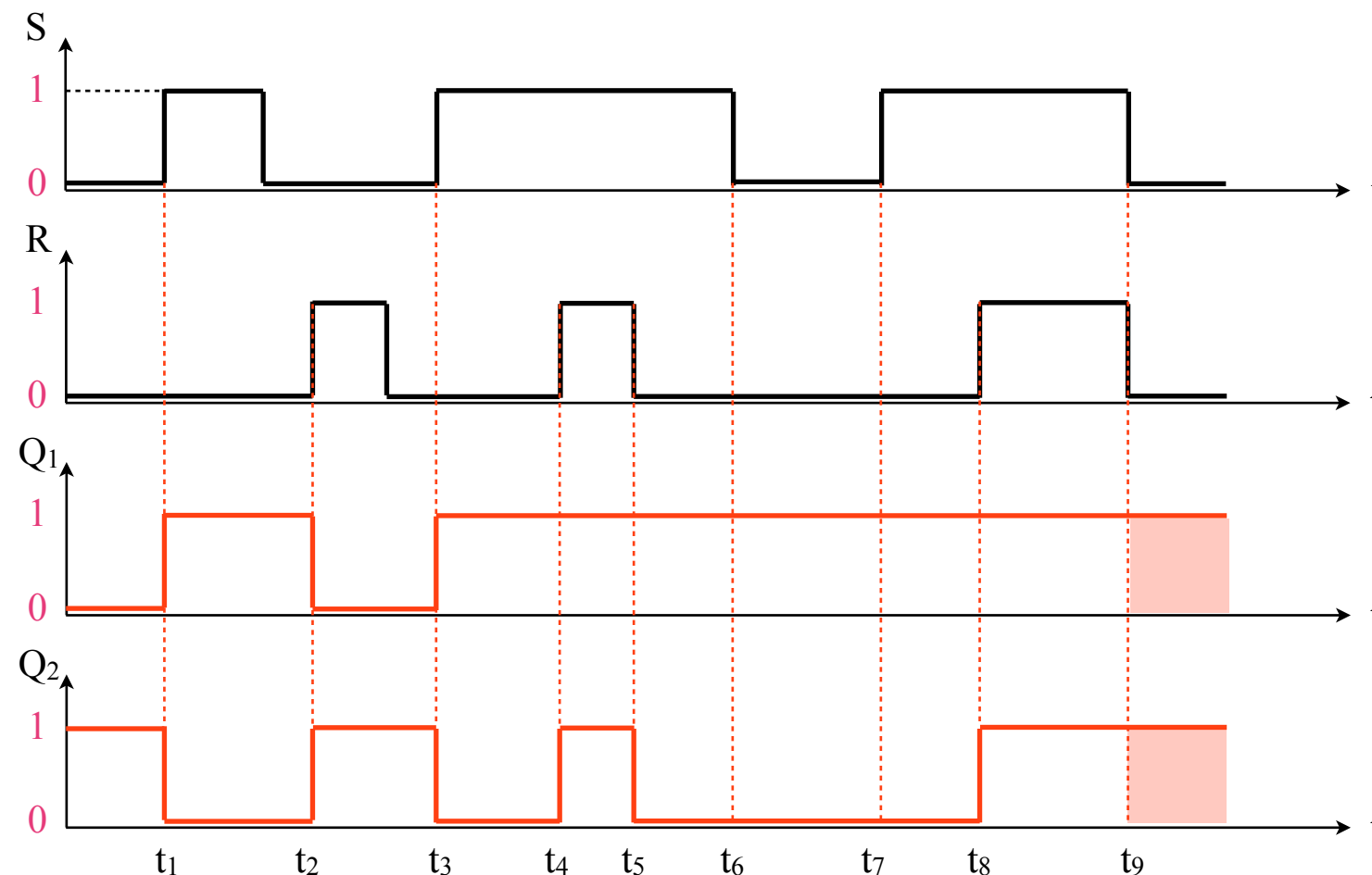


t_n		t_{n+1}
D	V	Q ₁
0	0	Q _{1n}
0	1	0
1	0	Q _{1n}
1	1	1

10. Zeitabhängige binäre Schaltungen

VI. Zeitablaufdiagramme

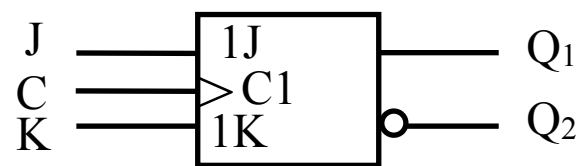
- Ziel: Visualisierung der Funktion einzelner Flipflops mit zeitlichen Darstellungen
- vorgegebene oder beliebig wählbare Eingangssignale
- Darstellung der Ausgangssignale in Abhängigkeit der Eingangssignale
- Beispiel: SR-Speicherflipflop



10. Zeitabhängige binäre Schaltungen

VII. Charakteristische Gleichungen

- algebraische Beschreibung der Arbeitsweise von Flipflops
- Gleichungen enthalten:
 - Einangs- und Ausgangsvariablen
 - Zeitangaben (Zeitpunkt von und nach einem betrachteten Takt)
- Beispiel: taktflankengesteuertes JK-Flipflop
 - Schritt 1: Umformung der Wahrheitstabelle (WT) in eine ausführliche WT (inkl. Angabe der möglichen Ausgangszustände vor dem betrachteten Takt)
 - Schritt 2: Bildung der ODER-Normalform
 - Schritt 3: Vereinfachung der Gleichung (mit Hilfe der Schaltalgebra oder KV-Diagrammen)



t_n		t_{n+1}
K	J	Q_1
0	0	Q_{1n}
0	1	1
1	0	0
1	1	$\neg Q_{1n}$

10. Zeitabhängige binäre Schaltungen

VII. Charakteristische Gleichungen

- Schritt 1: ausführliche Wahrheitstabelle

t_n			t_{n+1}	
K	J	Q_1	Q_1	
0	0	0	0	
0	0	1	1	$\Rightarrow Q_1 \wedge \bar{J} \wedge \bar{K}$
0	1	0	1	$\Rightarrow \bar{Q}_1 \wedge J \wedge \bar{K}$
0	1	1	1	$\Rightarrow Q_1 \wedge J \wedge \bar{K}$
1	0	0	0	
1	0	1	0	
1	1	0	1	$\Rightarrow \bar{Q}_1 \wedge J \wedge K$
1	1	1	0	

- Schritt 2: ODER-Normalform

$$Q_{1(n+1)} = [(Q_1 \wedge \neg J \wedge \neg K) \vee (\neg Q_1 \wedge J \wedge \neg K) \vee (Q_1 \wedge J \wedge \neg K) \vee (\neg Q_1 \wedge J \wedge K)]_n$$

- Schritt 3: Vereinfachen der Gleichung

	Q_1	Q_1	$\neg Q_1$	$\neg Q_1$	
J	1		1	1	$\neg Q_1 \wedge J$
$\neg J$	1				
	$\neg K$	K	K	$\neg K$	

$Q_1 \wedge \neg K$

\Rightarrow

$$Q_{1(n+1)} = [(\neg Q_1 \wedge J) \vee (Q_1 \wedge \neg K)]_n$$

10. Zeitabhängige binäre Schaltungen

VII. Charakteristische Gleichungen

Aufgabe: Bestimmen Sie die charakteristische Gleichung des taktflankengesteuerten SR-Flipflops.

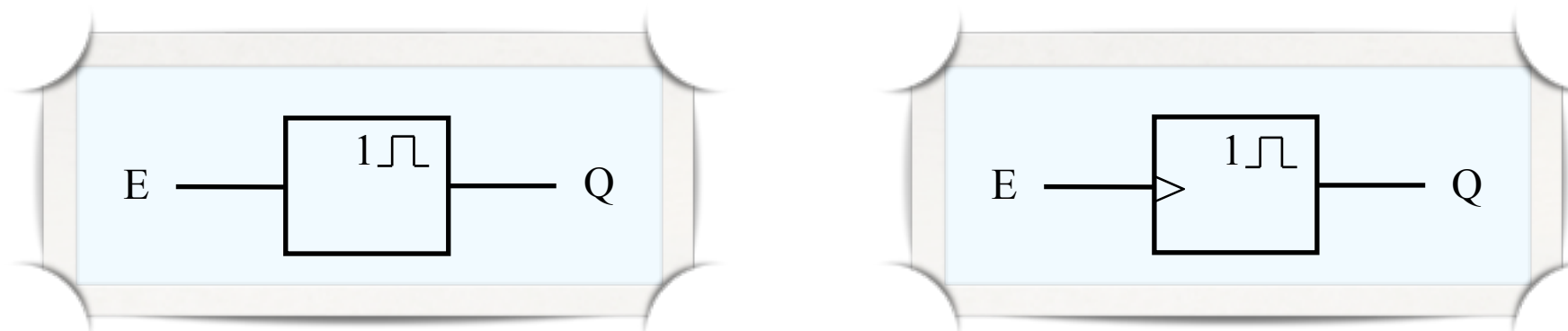


10. Zeitabhängige binäre Schaltungen

VIII. Monostabile Kippstufen

- bisher: Diskussion bistabiler Kippstufen $\rightarrow \forall$ zwei stabile Zustände
- Monostabile Kippstufe
 - \forall ein stabiler Zustand ($Q = 0$)
 - \forall ein instabiler (nichtstabiler) Zustand ($Q = 1$)
- Dauer (Verweilzeit) des instabilen Zustands definiert sich durch das verwendete RC-Glied (meist externe Bauelemente, d.h. zusätzliche Eingänge nötig)

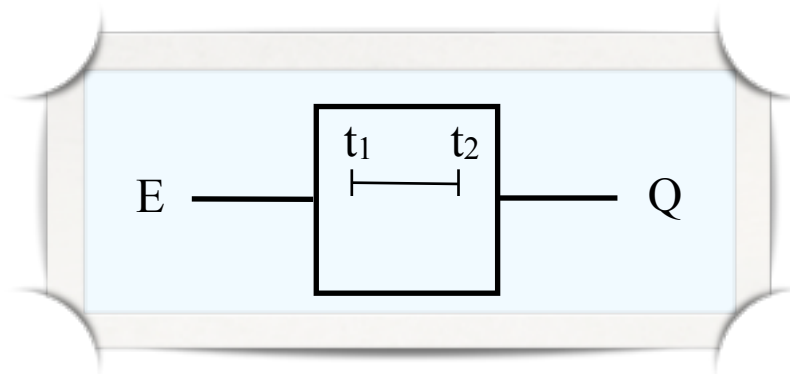
$t_Q = 0,69 \cdot R \cdot C$
- Änderung des Eingangssignals während t_Q
 - normale monostabile Kippstufe \rightarrow kein Einfluss auf Ausgangssignal
 - nachtriggerbare monostabile Kippstufe \rightarrow Verlängerung des nichtstabilen Zustands um t_Q
- Zustands- oder Flankensteuerung möglich



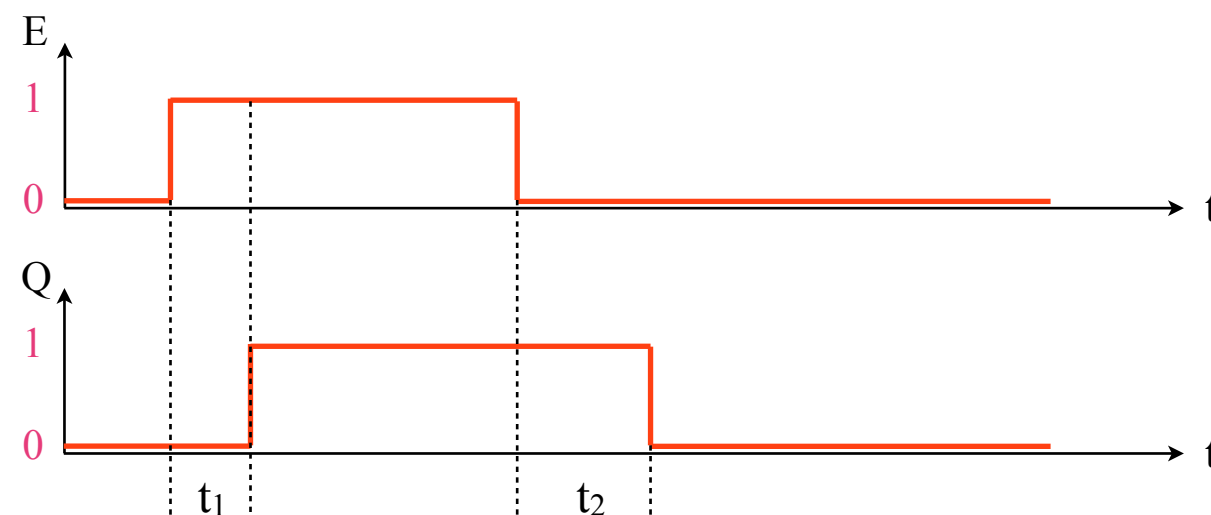
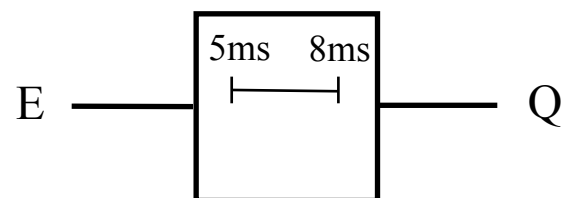
10. Zeitabhängige binäre Schaltungen

IX. Verzögerungsglieder

- Verzögerung ansteigender und abfallender Signalflanken
- Definition
 - t_1 : Verzögerungszeit der ansteigenden Flanke
 - t_2 : Verzögerungszeit der abfallenden Flanke
- Schaltzeichen



- Beispiel:



10. Zeitabhängige binäre Schaltungen

IX. Verzögerungsglieder

Aufgabe: Sie haben für den Aufbau eines Einschalt-Verzögerungsglieds (Verzögerungszeit: 5ms) ein UND-Glied, sowie eine monostabile Kippstufe (normaler + negierter Ausgang) mit einer Verweilzeit von 5ms zur Verfügung.

- a) Skizzieren Sie die Verschaltung und geben Sie das Zeitablaufdiagramm an.
- b) Welches Problem kann am Ausgang auftreten?
- c) Wie können Sie das Problem beheben?



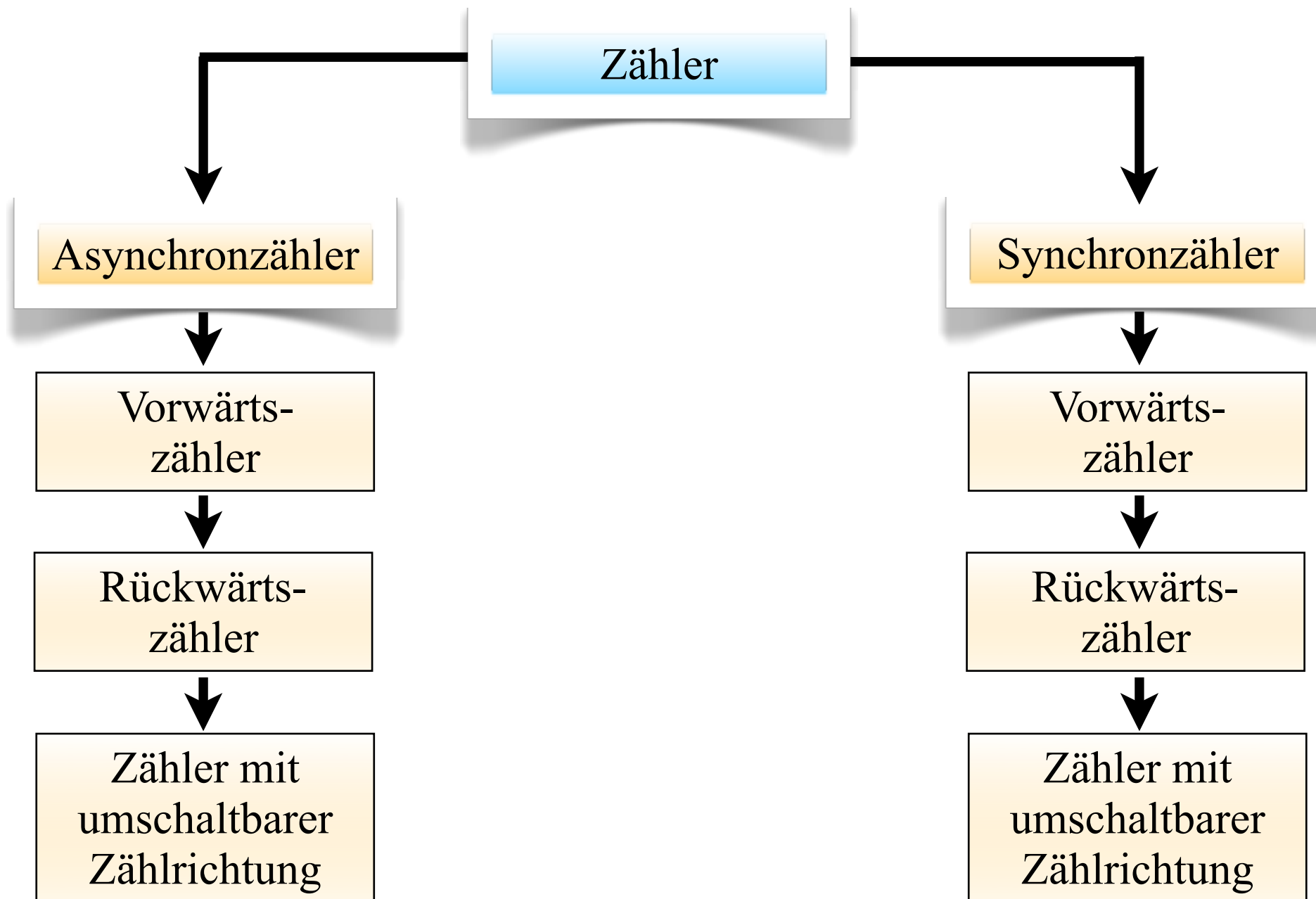
11. Zähler und Frequenzteiler

I. Zählen und Zählerarten

- Vorwärtszählen: fortlaufende Addition mit Eins
 $1+1=2$; $2+1=3$; $3+1=4$; ...
- Rückwärtszählen: forlaufende Subtraktion mit Eins
 $6-1=5$; $5-1=4$; ...
- Zählen ist in allen Codes möglich
(Dual, BCD, 3-Exzess, Aiken, ...)
- bei Binärzählern werden nur „0“- und „1“-Signale verarbeitet
- praktische Bedeutung der Binärzähler → „Zähler“
- Unterscheidung der Zähler nach
 - verwendetem Code
 - Zählrichtung
 - Betriebsart
 - synchron
 - asynchron

11. Zähler und Frequenzteiler

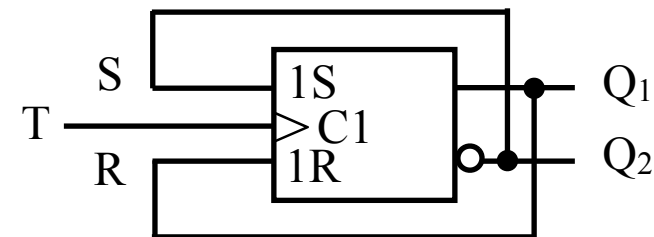
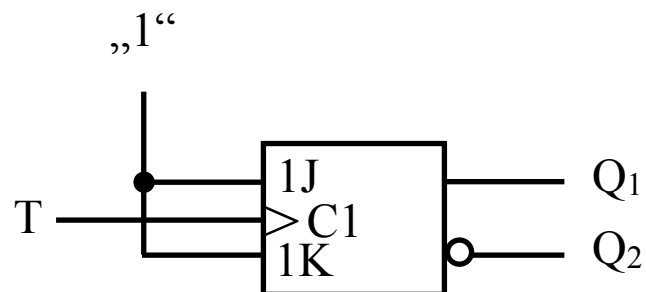
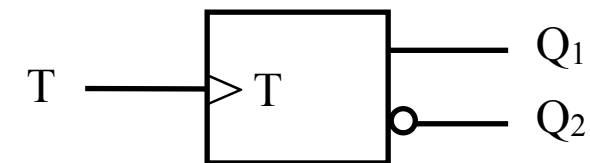
I. Zählen und Zählerarten: Übersicht



11. Zähler und Frequenzteiler

II. Asynchronzähler

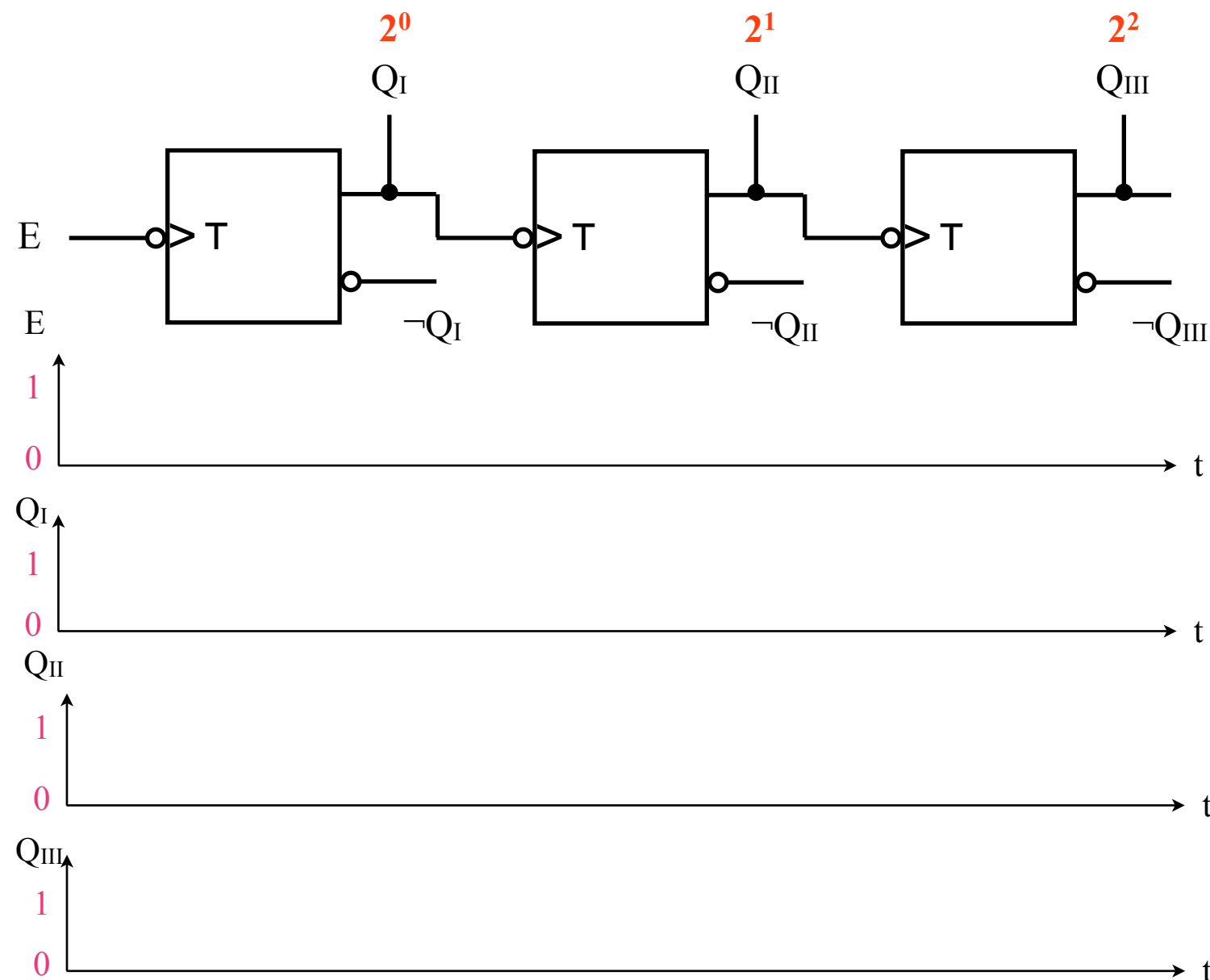
- Funktionsweise:
Die im Funktionsumfang eines Zählers enthaltenen Schaltglieder werden nicht mit einem gemeinsamen Takt parallel, sondern nacheinander angesteuert.
- Aufbau wahlweise mit:
 - T-Flipflops
 - JK-Flipflops (als T-Flipflop verschaltet)
 - SR-Flipflops (als T-Flipflop verschaltet)



11. Zähler und Frequenzteiler

II. Asynchrone Zähler

- Dual-Vorwärtszähler
 - zählt von 0 bis max. Wert → Sprung auf 0 → erneuter Zählvorgang
 - Funktionsweise anhand eines 3-Bit-Zählers: 3-Bit → es werden 3 Ausgänge (Flipflops) benötigt

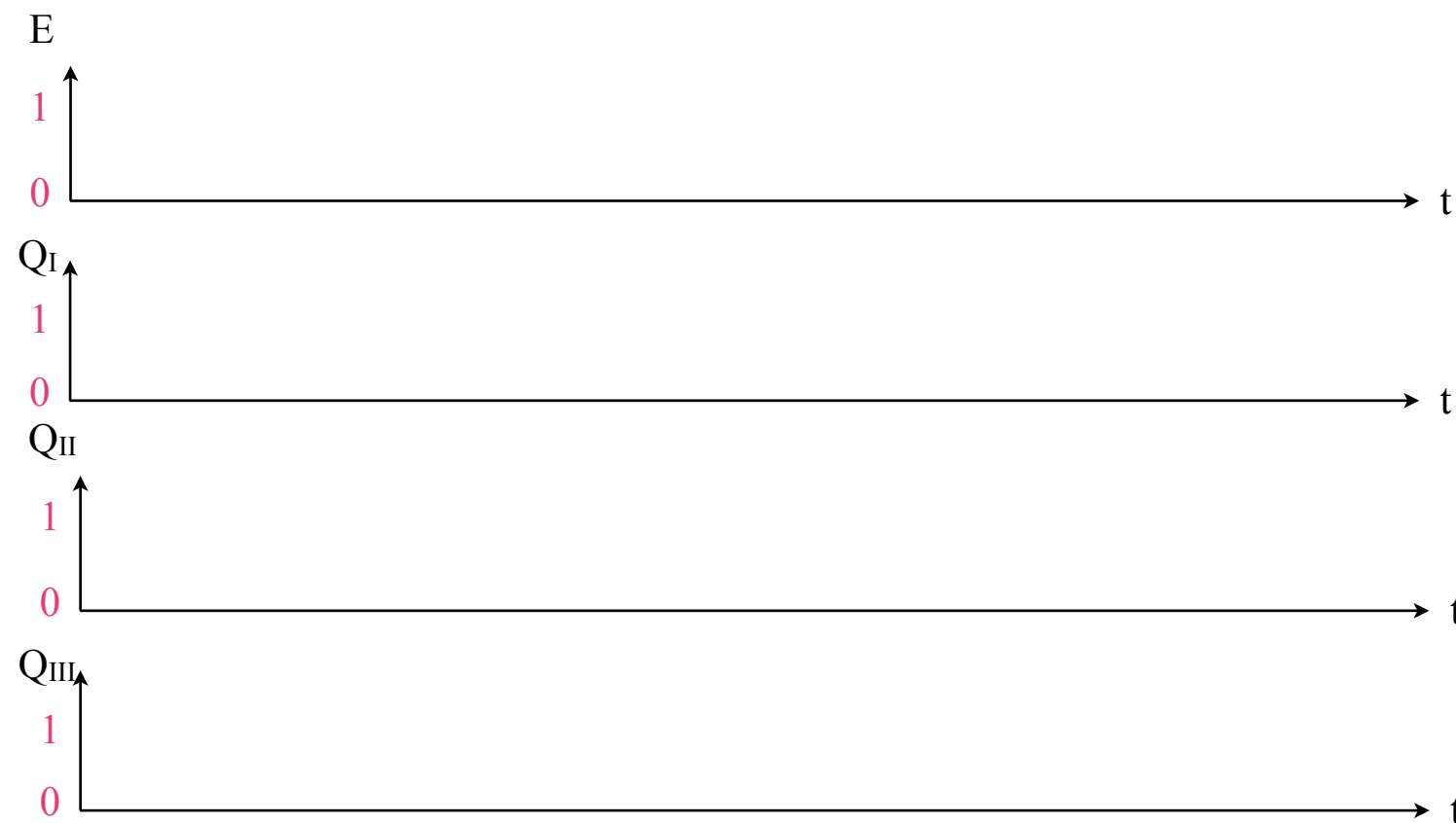


11. Zähler und Frequenzteiler

II. Asynchrone Zähler

- Dual-Vorwärtszähler

- Berücksichtigung der Signallaufzeiten: (FF aus TTL-Familie: $t \approx 30\text{-}50\text{ ns}$)



- Nachteil des Asynchrone Zählers: Verschiebung der Impulsreihen → Verringerung der höchstmöglichen Zählfrequenz

11. Zähler und Frequenzteiler

II. Asynchrone Zähler

- Dual-Vorwärtszähler

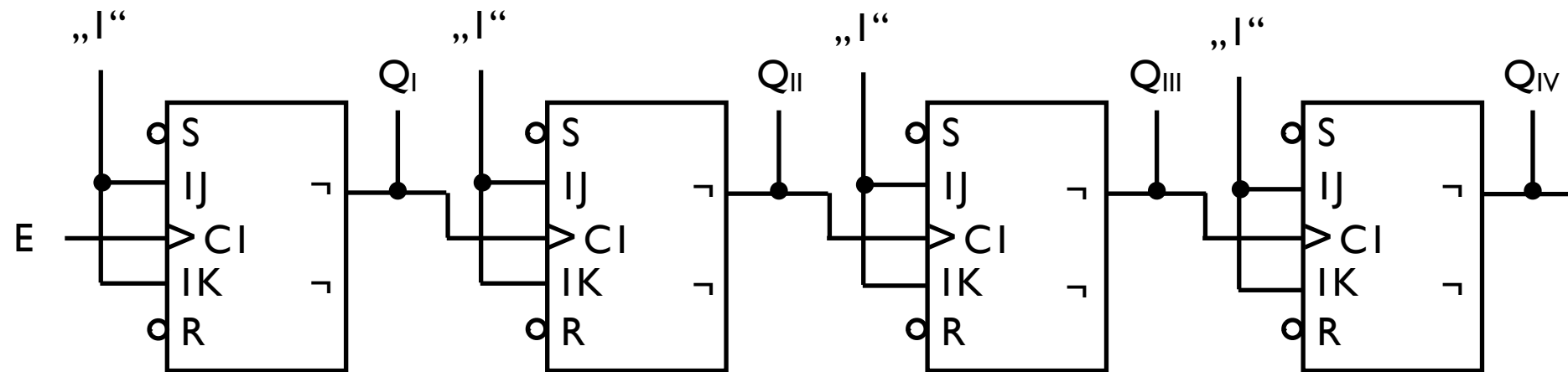
Aufgabe: Bauen Sie mit JK-Master-Slave-Flipflops einen 4-Bit-Dual-Vorwärtszähler auf (Schaltbild).



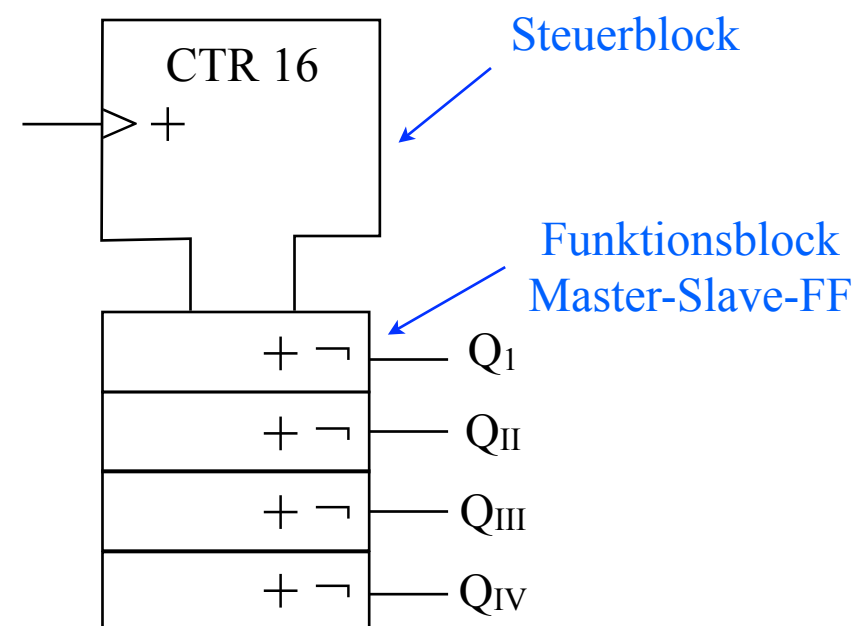
11. Zähler und Frequenzteiler

II. Asynchronzähler

- Dual-Vorwärtszähler: Aufbau mit JK-Master-Slave-FF



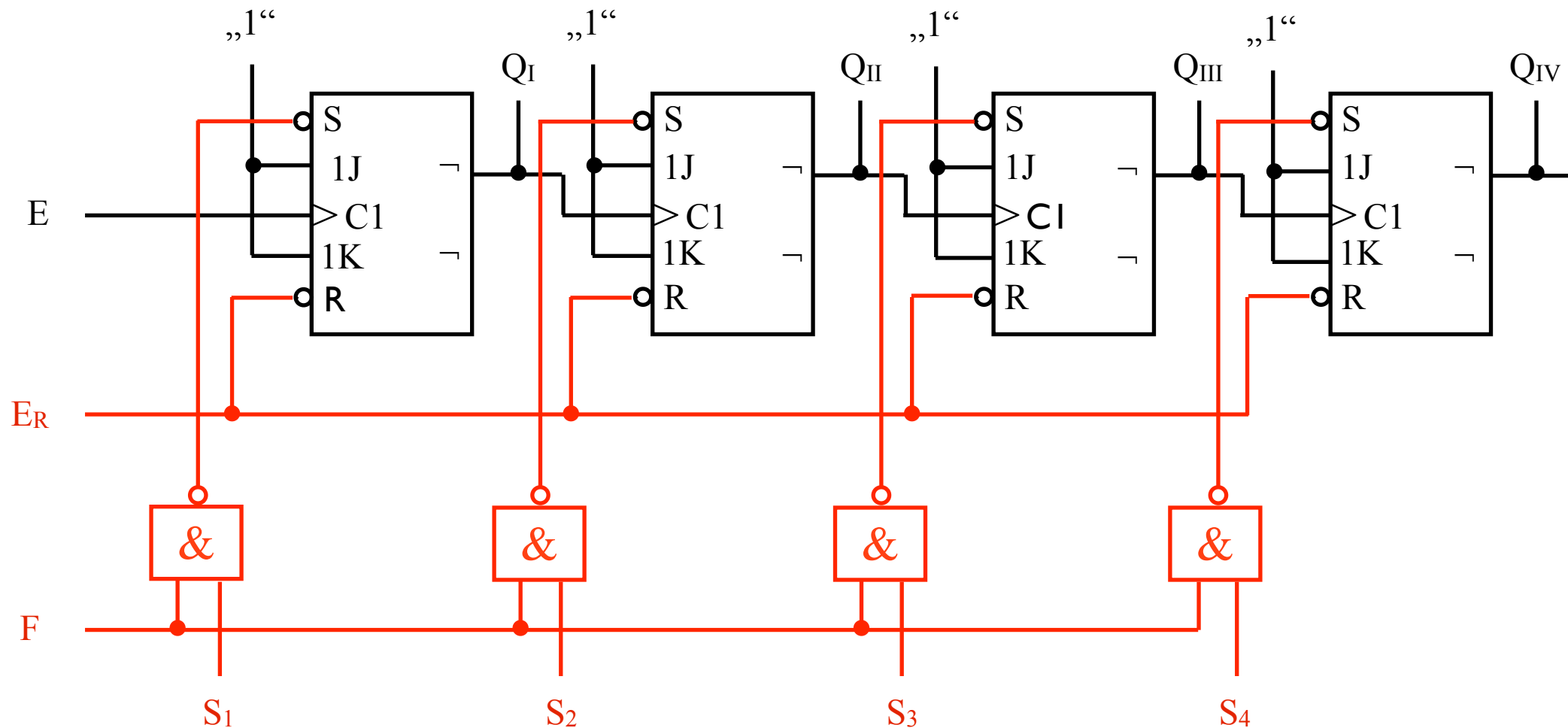
- Schaltzeichen des Dual-Vorwärtszählers



11. Zähler und Frequenzteiler

II. Asynchrone Zähler

- Dual-Vorwärtszähler: mit taktunabhängiger Setz- und Rücksetzmöglichkeit

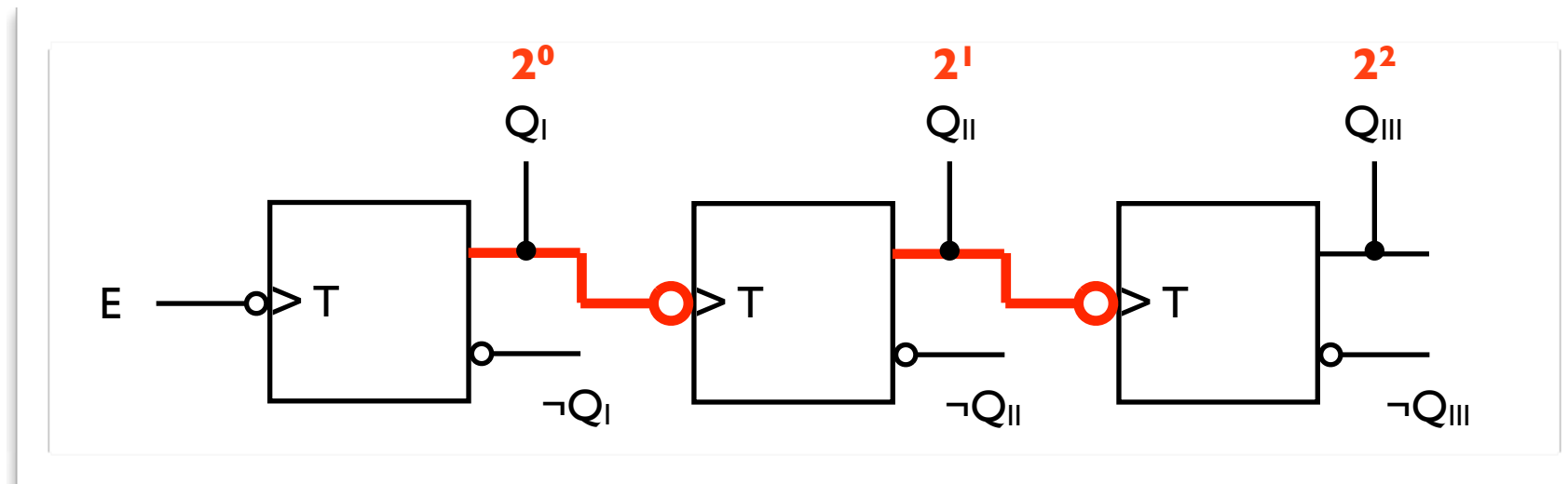


- Setzen auf einen bestimmten Zahlenwert mit Setzeingängen S₁₋₄ möglich

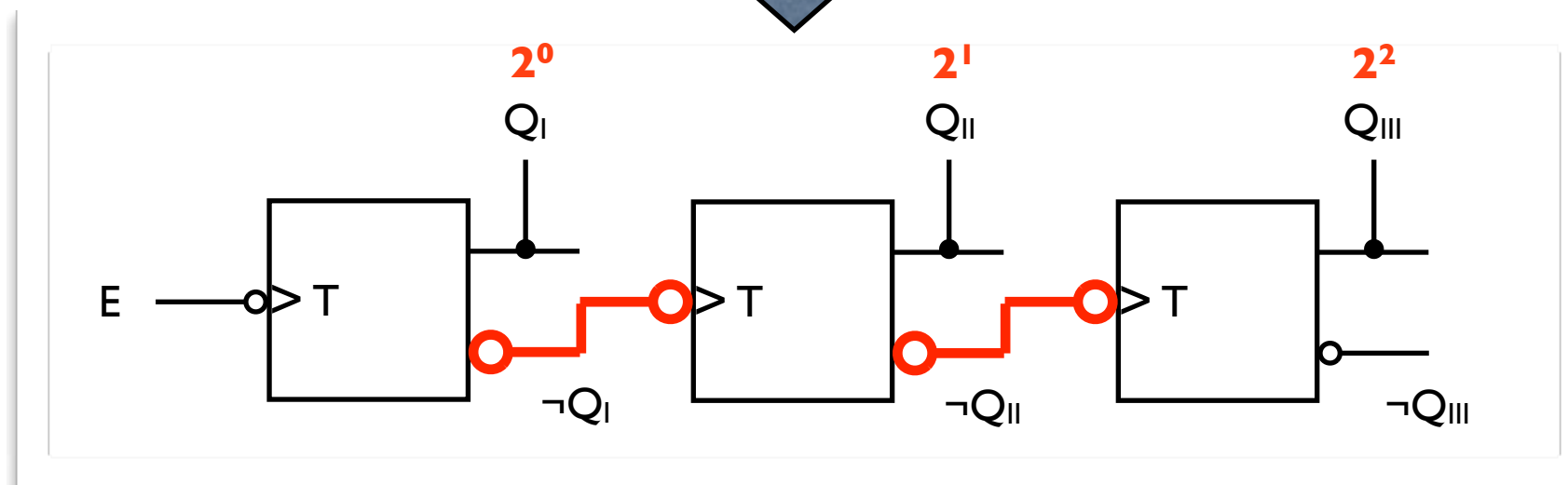
II. Zähler und Frequenzteiler

II. Asynchrone Zähler

- Dual-Vorwärtszähler



- Dual-Rückwärtszähler



II. Zähler und Frequenzteiler

II. Asynchronzähler

- Dual-Rückwärtszähler: Zeitablaufdiagramm



11. Zähler und Frequenzteiler

II. Asynchronzähler

- Dual-Rückwärtszähler

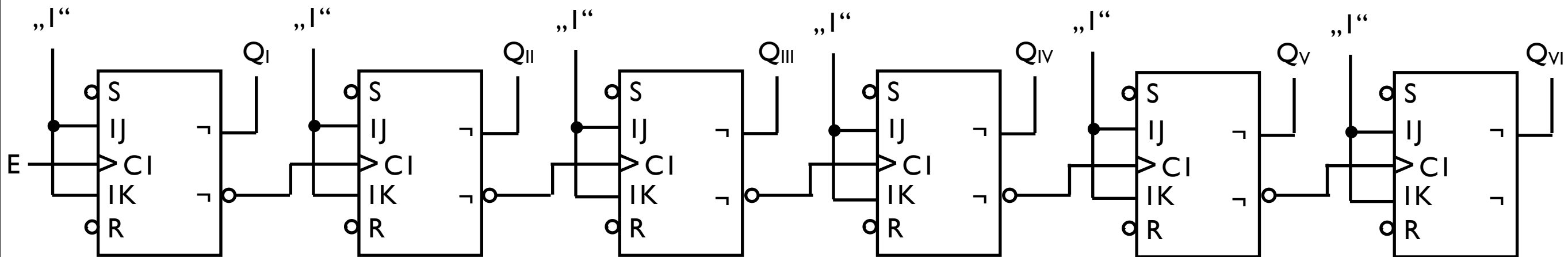
Aufgabe: Bauen Sie mit JK-Master-Slave-Flipflops einen 6-Bit-Dual-Rückwärtszähler auf (Schaltbild).



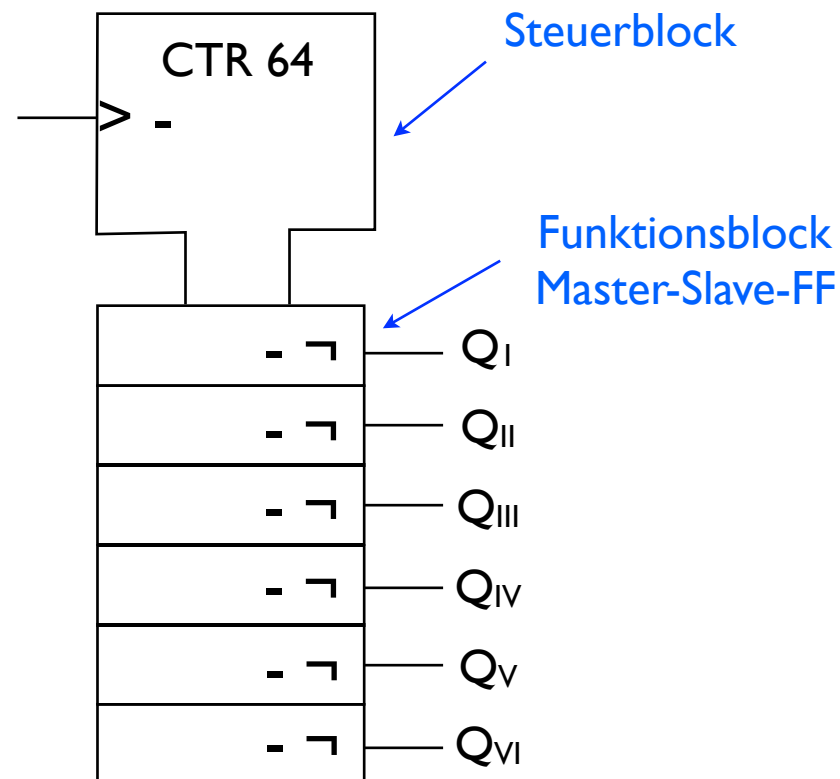
II. Zähler und Frequenzteiler

II. Asynchrone Zähler

- Dual-Rückwärtszähler: Aufbau mit JK-Master-Slave-FF

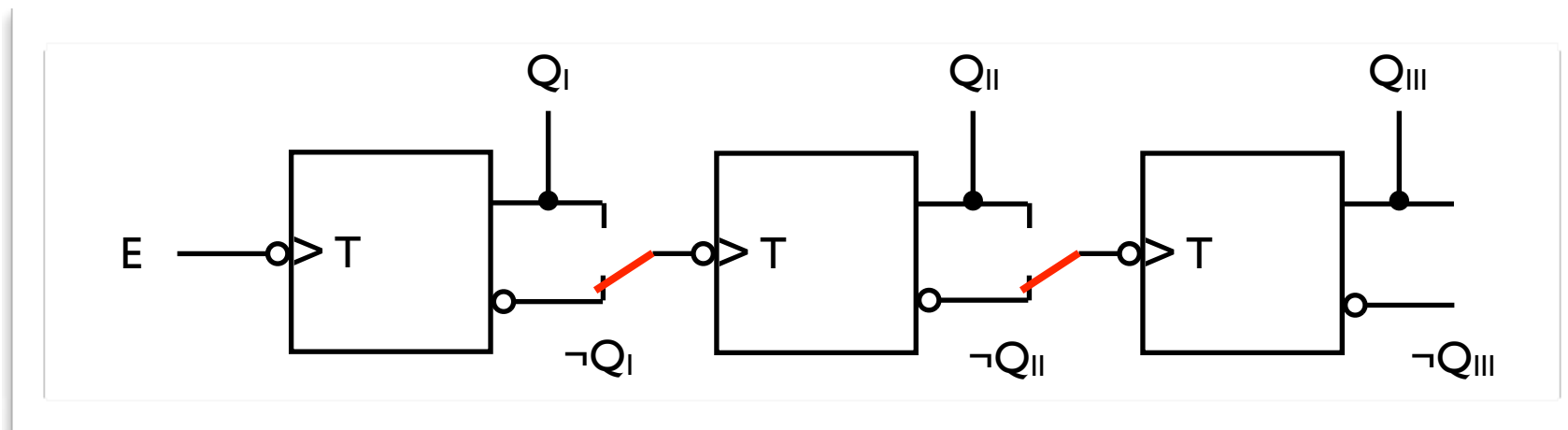


- ▶ Schaltzeichen des Dual-Rückwärtszählers

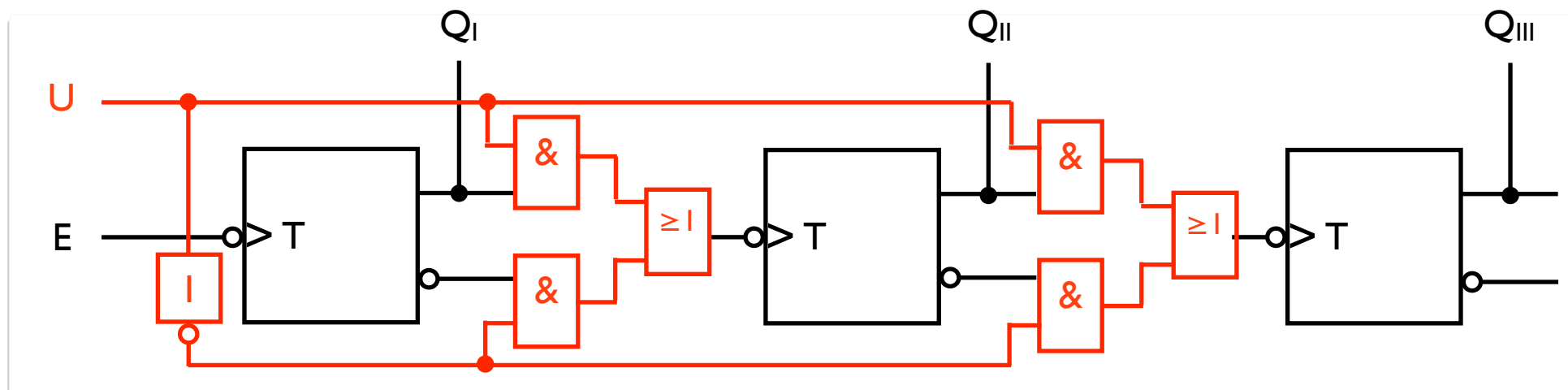


II. Asynchrone Zähler

- Dual-Zähler mit umschaltbarer Zählrichtung
 - Realisierung durch umschaltbare Ansteuersignale Q und $\neg Q$



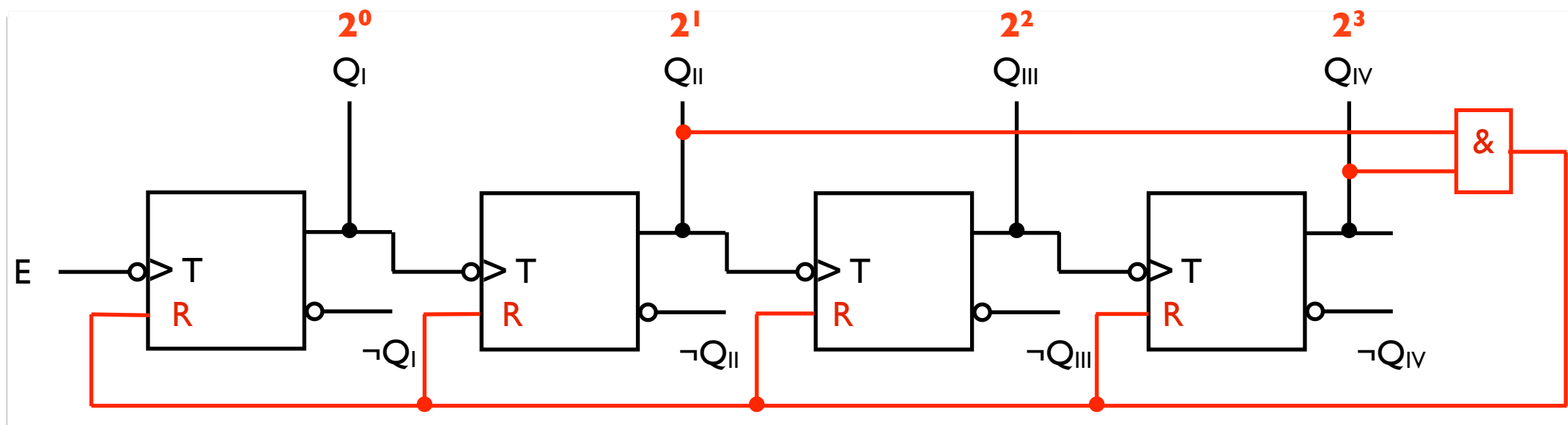
- Umschaltung durch Kontaktschalter → Ersetzen durch Verknüpfungsglieder



II. Zähler und Frequenzteiler

II. Asynchronzähler

- BCD-Vorwärtszähler
 - Ziffern von 0 bis 9 → 4-Bit-Zähler

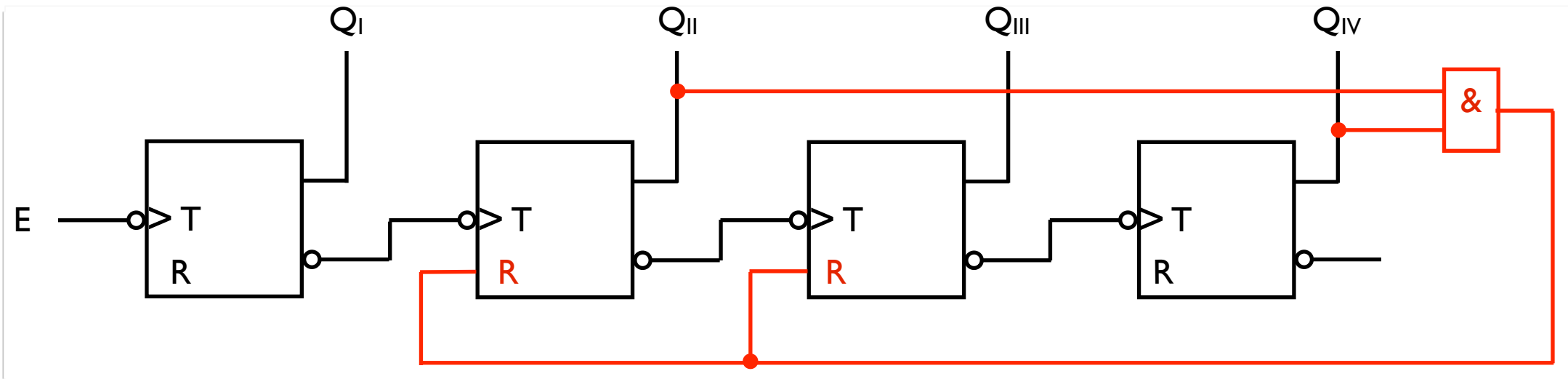


- taktunabhängiger Rückstelleingang nötig
- Zähler muss von 9 auf 0 springen, d.h. von „1001“ auf „1010“ → $Q_{II} = 1 \wedge Q_{IV} = 1$
- Zustand „1010“ liegt ca. 50ns an den Ausgängen (bei TTL-Schaltkreisfamilie)
→ kann evtl. Störungen verursachen

11. Zähler und Frequenzteiler

II. Asynchrone Zähler

- BCD-Rückwärtszähler

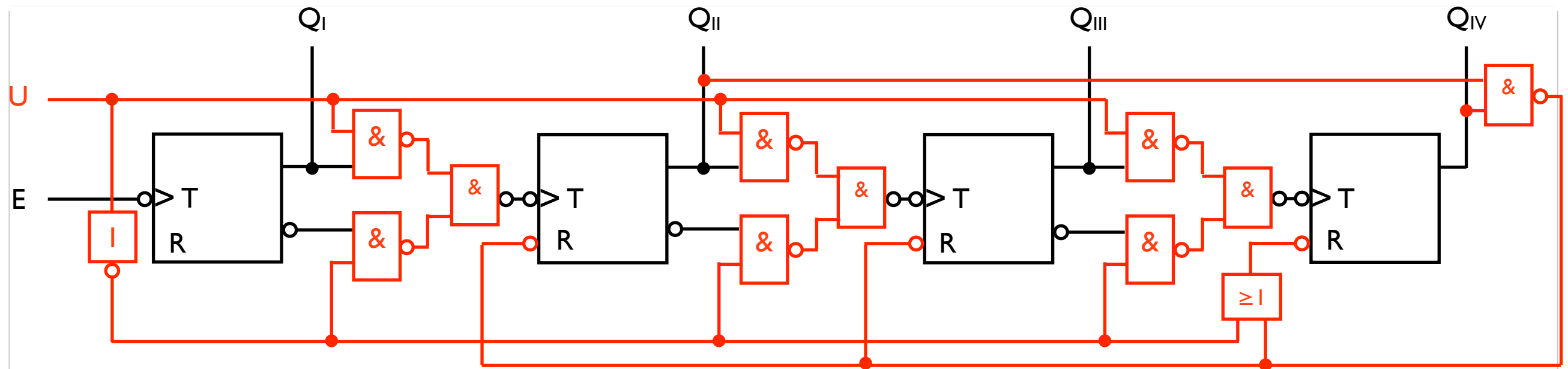


- ▶ benötigter Startwert der Rückwärtszählung: „1001“
- ▶ üblicher Startwert: „1111“ → Rücksetzen von Flipflop II und III nötig (Signallaufzeit ca. 50 ns)

II. Zähler und Frequenzteiler

II. Asynchronzähler

- BCD-Zähler mit umschaltbarer Zählrichtung

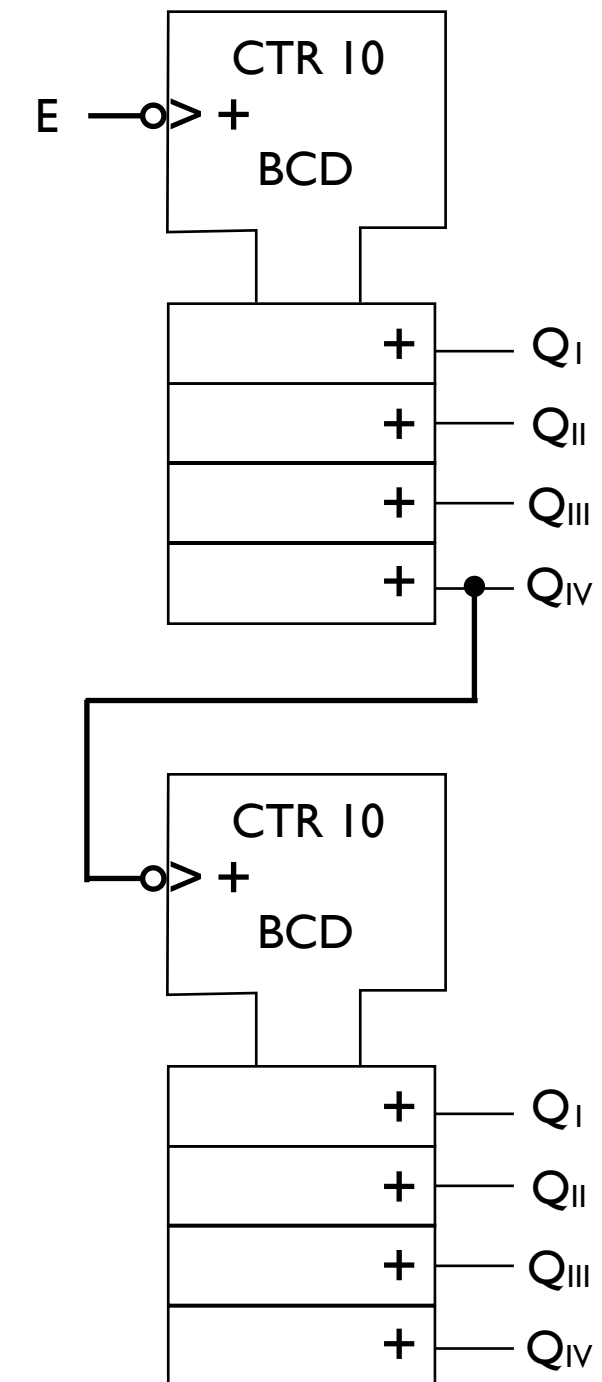


- › Steuerung der Rückstellung durch „0“-Signale
- › Signal $U = 1$: Vorwärtszähler
 - Rücksetzen auf „0000“, wenn $Q_{II} = Q_{IV} = 1$
- › Signal $U = 0$: Rückwärtszähler
 - Umstellung des Rückwärtszählers auf „1001“ vor dem Zählvorgang

11. Zähler und Frequenzteiler

II. Asynchrone Zähler

- BCD-Dekadenzähler
 - Erweiterung des Zählbereichs erfordert weitere BCD-Vorwärtszähler
 - Zwei BCD-Vorwärtszähler → Zählbereich: 0 - 99
 - Drei BCD-Vorwärtszähler → Zählbereich: 0 - 999
 - Vier Ausgänge pro Dezimalziffer (Darstellung mit BCD-7-Segment-Kodeumsetzer)
 - Übergang von 9 auf 10:
 - Ausgang Q_{IV} von Zähler 1 wechselt von „1“ auf „0“ → Eingangssignal in Zähler 2



11. Zähler und Frequenzteiler

II. Asynchronzähler

- Modulo-n-Zähler
 - Anforderung:
 - bis zu einem gewünschten Zahlenwert zählen
 - auf Null zurücksetzen
 - erneut zählen, bzw. stehenbleiben und erneutes Startsignal abwarten
 - BCD-Zähler entspricht Modulo-10-Zähler

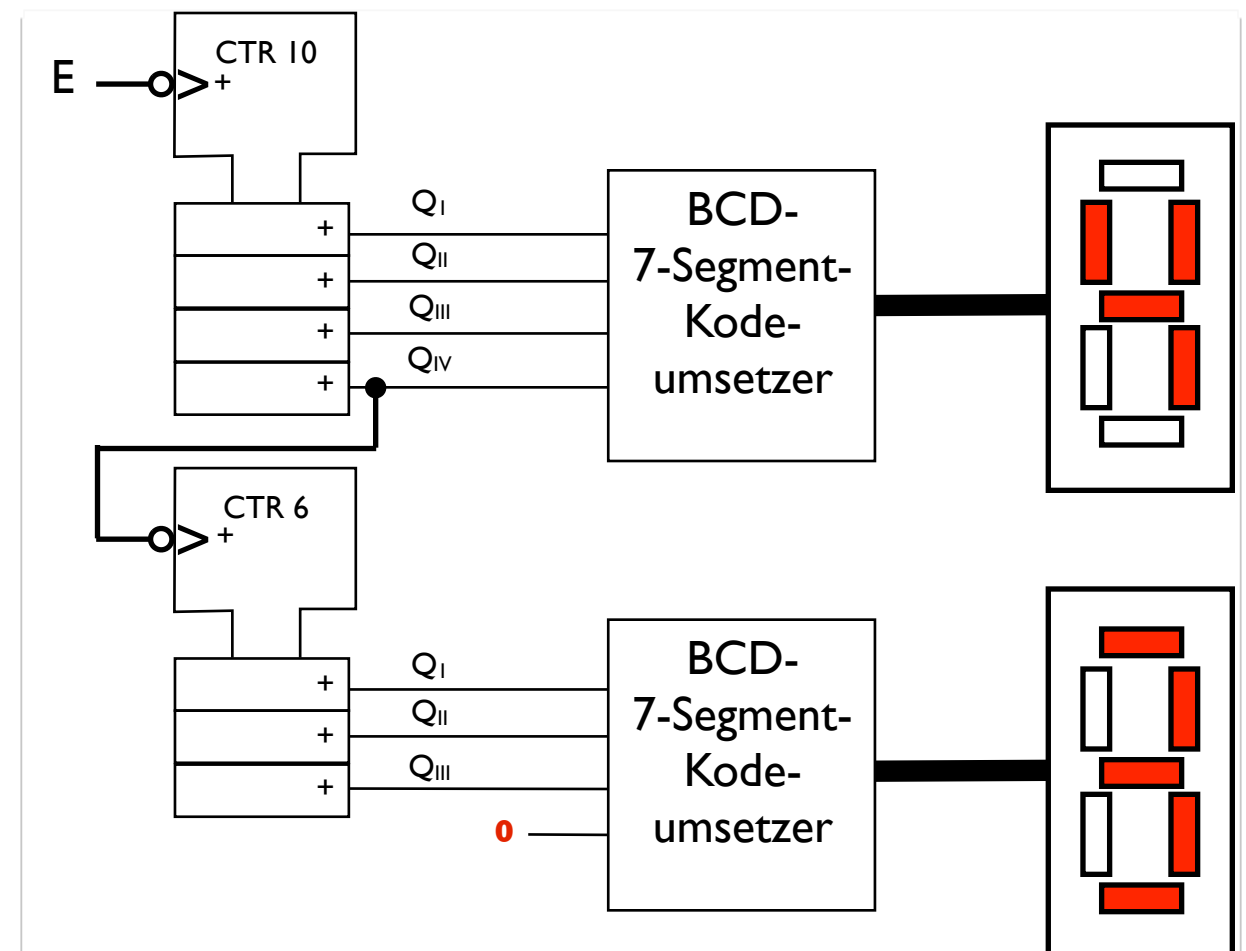
Aufgabe: Bauen Sie einen Modulo-5-Zähler auf.



11. Zähler und Frequenzteiler

II. Asynchronzähler

- Modulo-60-Zähler
 - Anwendung z.B. für elektronische Uhren
 - Variante 1:
 - Zählbereich: 0 bis 59 → 6 Flipflops nötig
 - Nullung bei Wechsel auf 60 („111100“)
 - Variante 2:
 - Zusammenschaltung eines Modulo-10- und eines Modulo-6-Zählers



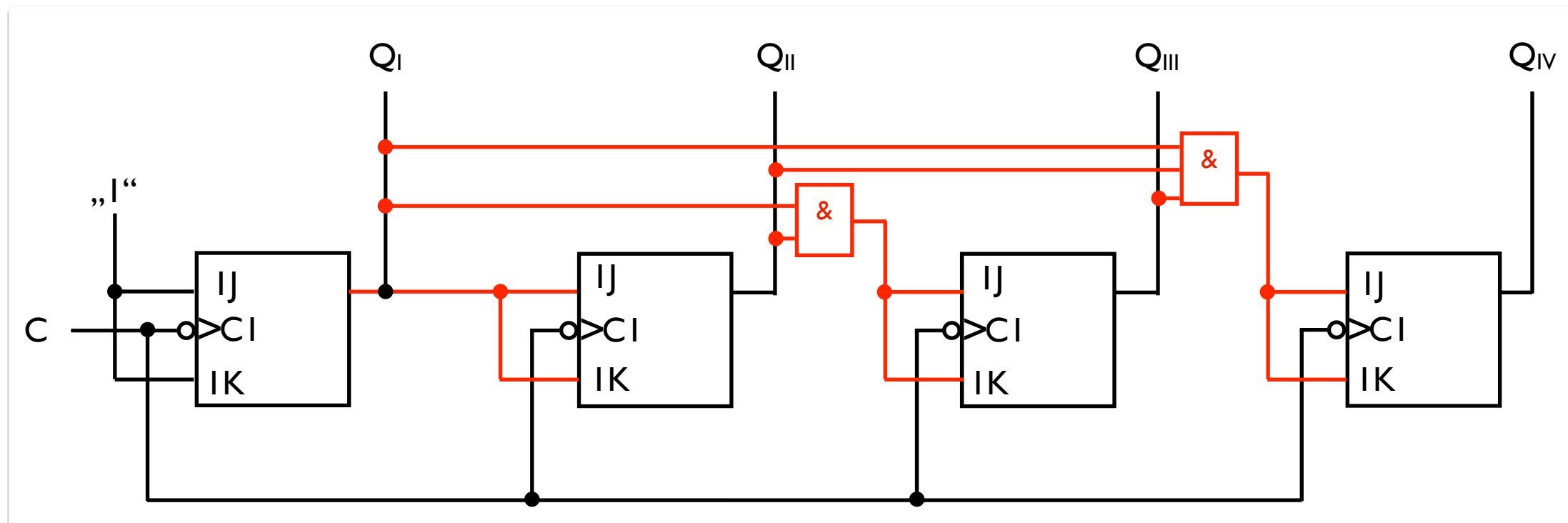
III. Synchronzähler

- Prinzip des Synchronzählers
 - Asynchronprinzip: Ausgang eines Flipflops steuert nächstes Flipflop
 - Nachteil: Schaltverzögerung durch Signal-Laufzeit der FF (30-50ns pro Flipflop)
 - → Fehler bei hohen Zähhfrequenzen und hoher Bit-Zahl
 - Synchronprinzip: gemeinsamer Schaltbefehl steuert gleichzeitig alle Flipflops an
 - vor dem Taktsignal erfolgt Festlegung, ob ein Flipflop zum nächsten Takt geschaltet wird oder nicht
 - weitere Eingänge erforderlich
 - Verwendung von JK-Flipflops für Synchronzähler
 - Vorteil bezüglich der Sicherheit bieten JK-Master-Slave-Flipflops

11. Zähler und Frequenzteiler

III. Synchronzähler

- Dual-Vorwärtszähler
 - Schaltung eines 4-Bit-Zählers

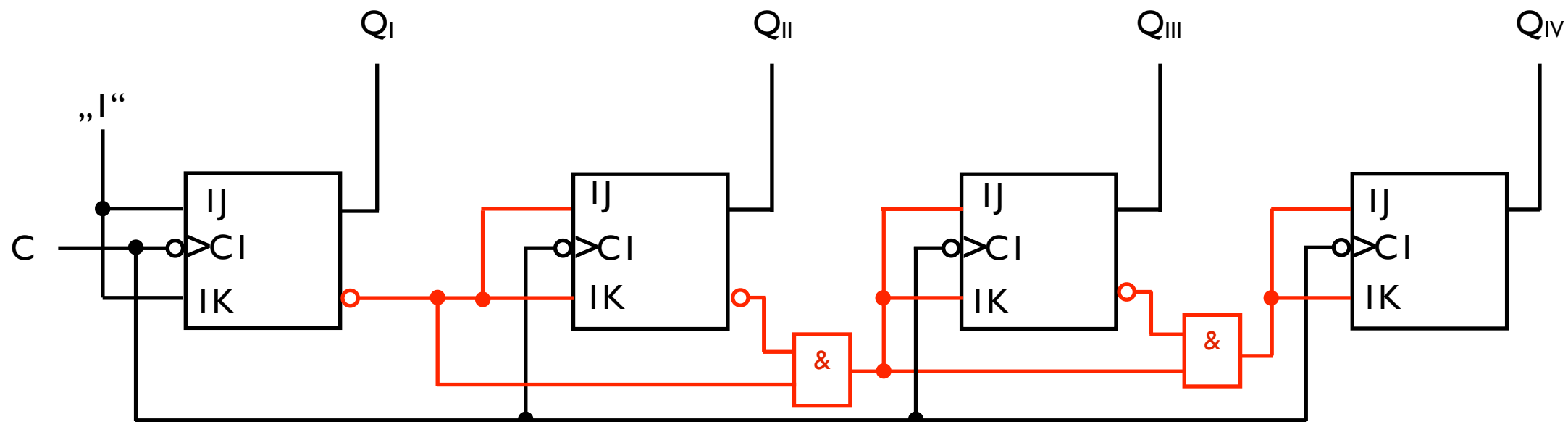


- › Herleitung der Schaltung mit Hilfe des Zeitablaufdiagramms möglich
- › Eingänge J und K werden jeweils verbunden
- › am ersten Flipflop wird „I“ an J und K gelegt
- › alle folgenden Flipflops erhalten als Eingangssignal UND-Verknüpfung der vorherigen Q-Ausgänge

11. Zähler und Frequenzteiler

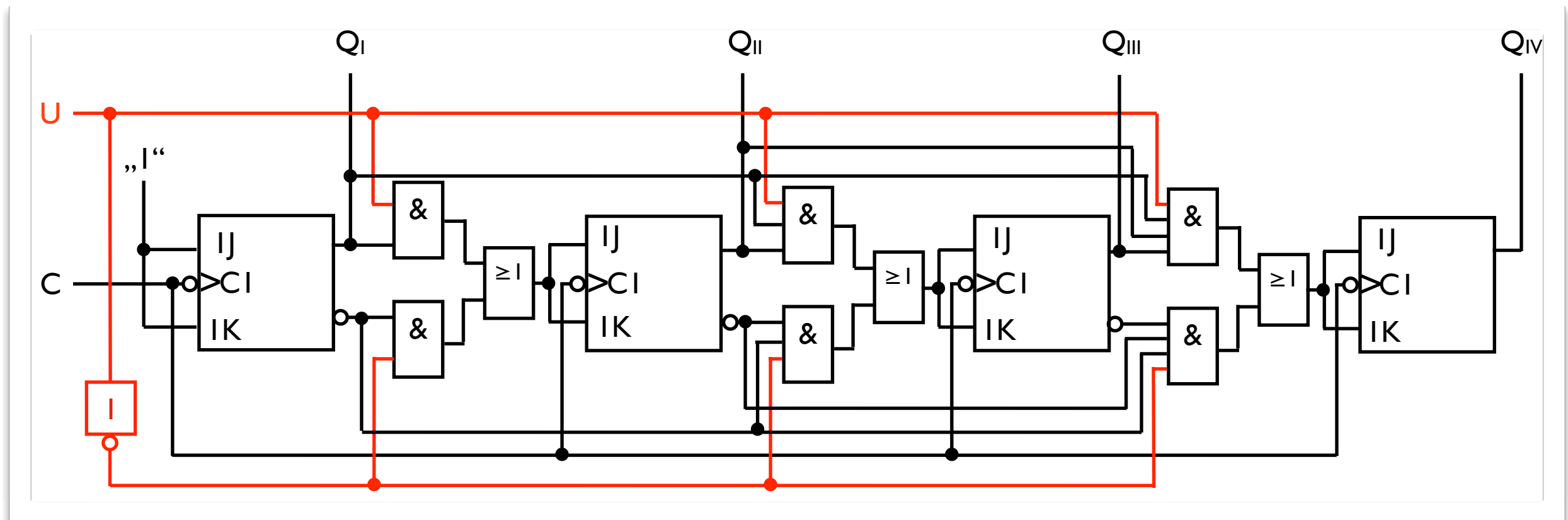
III. Synchronzähler

- Dual-Rückwärtszähler
 - Aufbau analog zu Vorwärtszähler, jedoch Verwendung der negierten Ausgänge



III. Synchronzähler

- Dual-Zähler mit umschaltbarer Zählrichtung
 - › Entwicklung des Zählers aus Dual-Vorwärts- und -Rückwärtszähler



- › U = 1: Vorwärtszähler
- › U = 0: Rückwärtszähler

III. Synchronzähler

- Berechnung von Synchronzählern
 - Schritte des Berechnungsverfahrens

- 1 Wahrheitstabelle aufstellen
- 2 Anwendungsgleichungen aufstellen und vereinfachen
- 3 charakteristische Gleichung der zu verwendenden Flipflops bestimmen
- 4 Verknüpfungsgleichungen durch Koeffizientenvergleich bestimmen
- 5 Schaltbild auf Basis der Verknüpfungsgleichungen zeichnen

- Berechnungsbeispiel:
4-Bit-Synchron-Dual-Vorwärtszähler

II. Zähler und Frequenzteiler

III. Synchronzähler

- Berechnung von Synchronzählern - 4-Bit-Synchron-Dual-Vorwärtszähler
 - Schritt I: Wahrheitstabelle aufstellen

Dezimalwert	t_n				t_{n+1}				Dezimalwert
	Q_D	Q_C	Q_B	Q_A	Q_D	Q_C	Q_B	Q_A	
0	0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	0	1	0	2
2	0	0	1	0	0	0	1	1	3
3	0	0	1	1	0	1	0	0	4
4	0	1	0	0	0	1	0	1	5
5	0	1	0	1	0	1	1	0	6
6	0	1	1	0	0	1	1	1	7
7	0	1	1	1	1	0	0	0	8
8	1	0	0	0	1	0	0	1	9
9	1	0	0	1	1	0	1	0	10
10	1	0	1	0	1	0	1	1	11
11	1	0	1	1	1	1	0	0	12
12	1	1	0	0	1	1	0	1	13
13	1	1	0	1	1	1	1	0	14
14	1	1	1	0	1	1	1	1	15
15	1	1	1	1	0	0	0	0	0

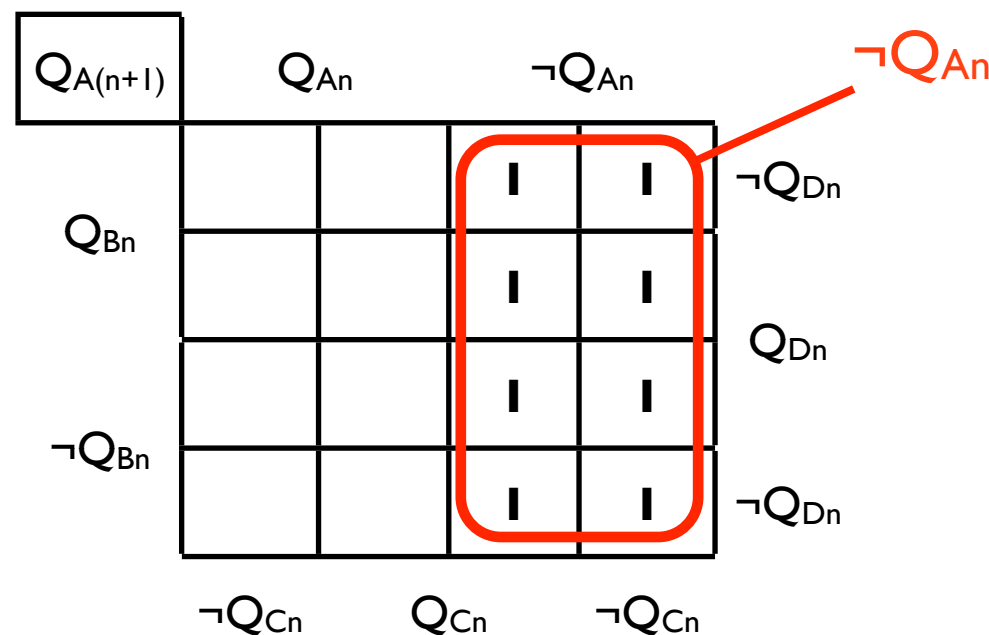
II. Zähler und Frequenzteiler

III. Synchronzähler

- Berechnung von Synchronzählern - 4-Bit-Synchron-Dual-Vorwärtszähler
 - › Schritt 2: Anwendungsgleichungen aufstellen (ODER-Normalform) und vereinfachen

$$Q_{A(n+1)} = [\neg A \neg B \neg C \neg D \vee \neg A B \neg C \neg D \vee \neg A \neg B C \neg D \vee \neg A B C \neg D \vee \neg A \neg B \neg C D \vee \neg A B \neg C D \vee \neg A \neg B C D \vee \neg A B C D]_n$$

Vereinfachung mit KV-Diagramm



$$Q_{A(n+1)} = \neg Q_{An}$$

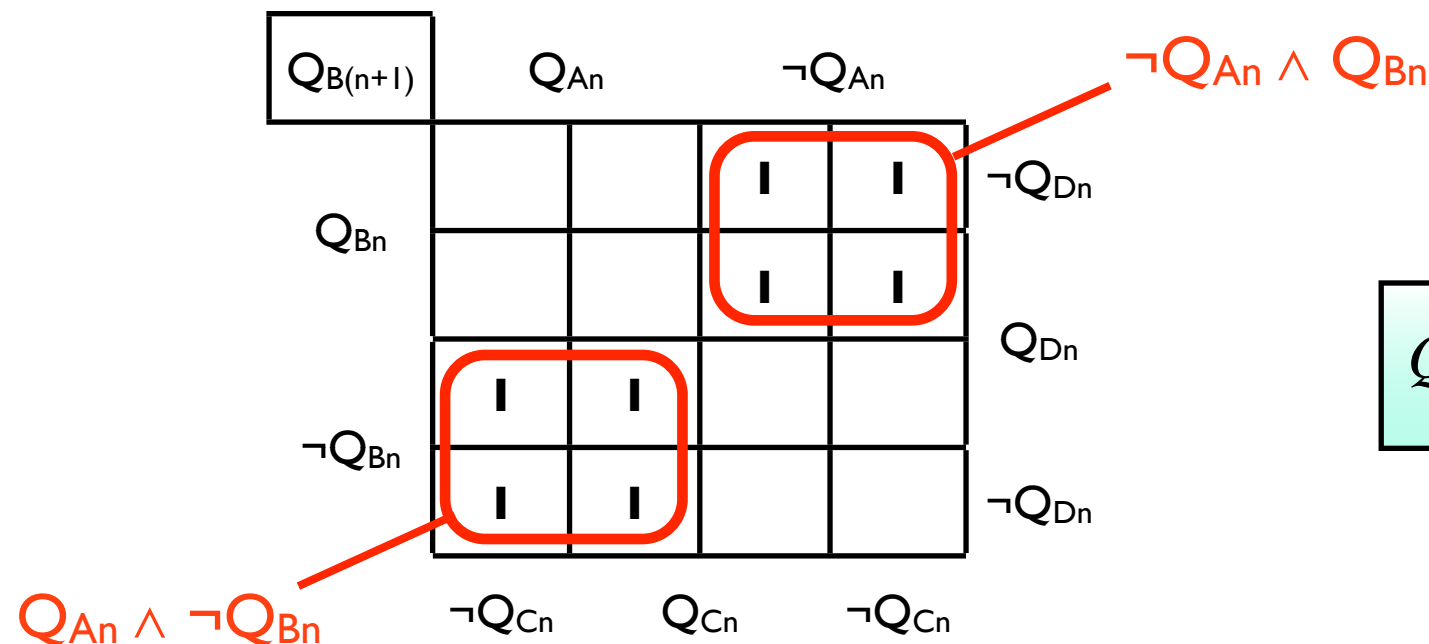
II. Zähler und Frequenzteiler

III. Synchronzähler

- Berechnung von Synchronzählern - 4-Bit-Synchron-Dual-Vorwärtszähler
 - › Schritt 2: Anwendungsgleichungen aufstellen (ODER-Normalform) und vereinfachen

$$Q_{B(n+1)} = [A \neg B \neg C \neg D \vee \neg A B \neg C \neg D \vee A \neg B C \neg D \vee \neg A B C \neg D \vee A \neg B \neg C D \vee \neg A B \neg C D \vee \neg A B C D \vee A B C D]_n$$

Vereinfachung mit KV-Diagramm



$$Q_{A(n+1)} = [Q_{An} \neg Q_{Bn} \vee \neg Q_{An} Q_{Bn}]_n$$

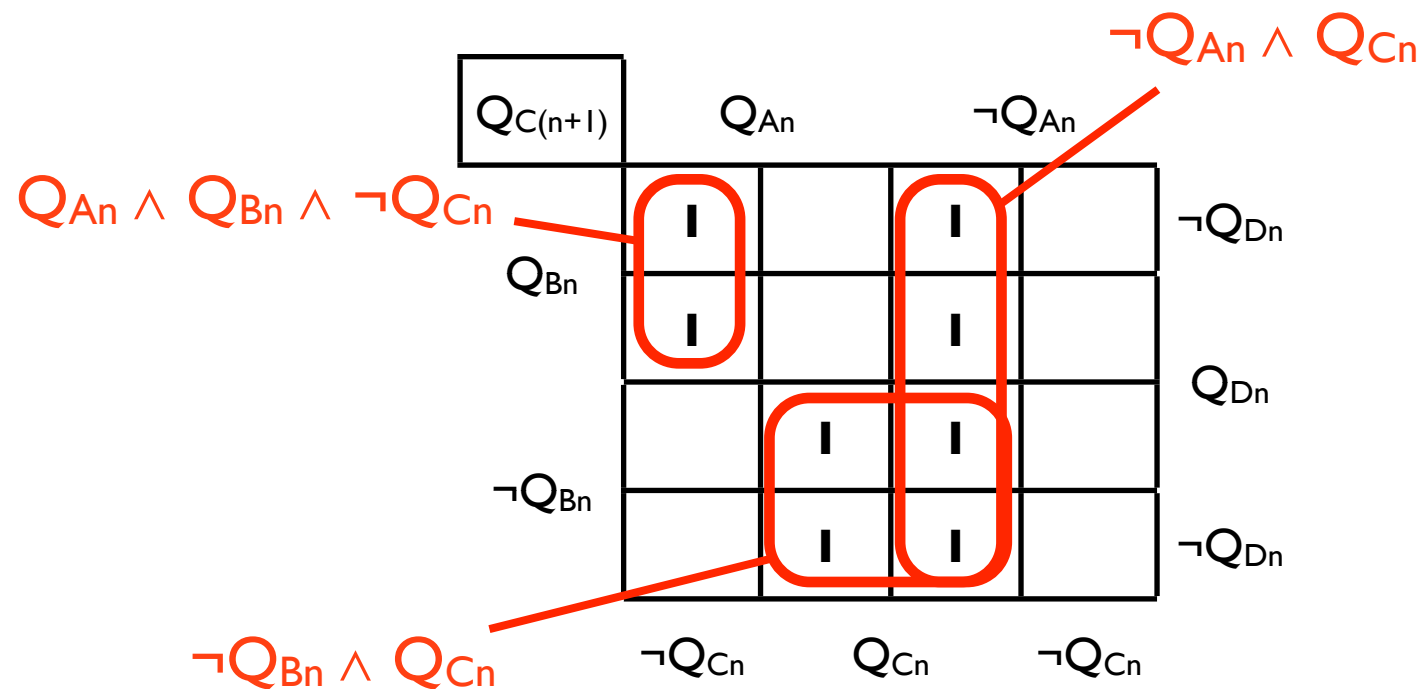
II. Zähler und Frequenzteiler

III. Synchronzähler

- Berechnung von Synchronzählern - 4-Bit-Synchron-Dual-Vorwärtszähler
 - Schritt 2: Anwendungsgleichungen aufstellen (ODER-Normalform) und vereinfachen

$$Q_{C(n+1)} = [AB\bar{C}\bar{D} \vee \bar{A}\bar{B}C\bar{D} \vee A\bar{B}C\bar{D} \vee \bar{A}BC\bar{D} \vee AB\bar{C}D \vee \bar{A}\bar{B}CD \vee A\bar{B}CD \vee \bar{A}BCD]_n$$

Vereinfachung mit KV-Diagramm



$$Q_{C(n+1)} = [Q_A Q_B \bar{Q}_C \vee \bar{Q}_A Q_C \vee \bar{Q}_B Q_C]_n$$

$$= [Q_A Q_B \bar{Q}_C \vee Q_C (\bar{Q}_A \vee \bar{Q}_B)]_n$$

$$Q_{C(n+1)} = [Q_A Q_B \bar{Q}_C \vee \neg(Q_A Q_B) Q_C]_n$$

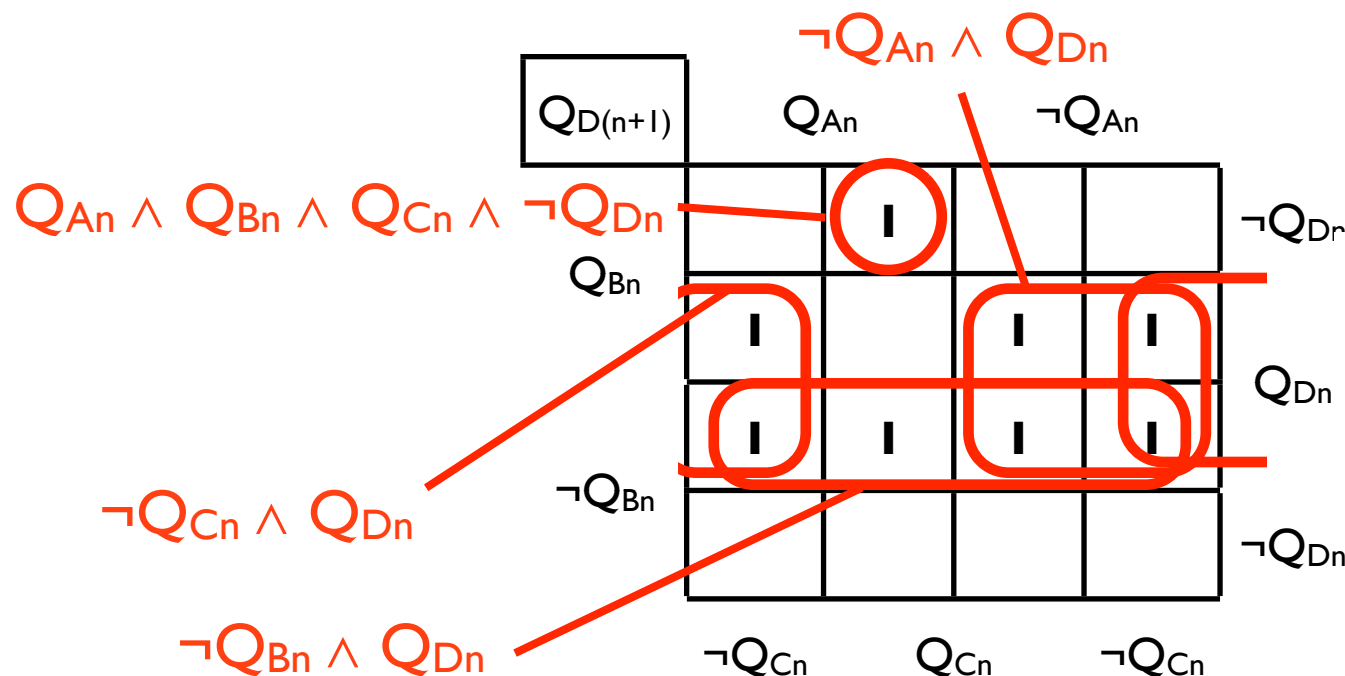
II. Zähler und Frequenzteiler

III. Synchronzähler

- Berechnung von Synchronzählern - 4-Bit-Synchron-Dual-Vorwärtszähler
 - › Schritt 2: Anwendungsgleichungen aufstellen (ODER-Normalform) und vereinfachen

$$Q_{D(n+1)} = [Q_A Q_B Q_C \neg Q_D \vee \neg Q_A \neg Q_B \neg Q_C Q_D \vee Q_A \neg Q_B \neg Q_C Q_D \vee \neg Q_A Q_B \neg Q_C Q_D \vee Q_A Q_B \neg Q_C Q_D \vee \neg Q_A \neg Q_B Q_C Q_D \vee Q_A \neg Q_B Q_C Q_D \vee \neg Q_A Q_B Q_C Q_D]_n$$

Vereinfachung mit KV-Diagramm



$$Q_{D(n+1)} = [Q_A Q_B Q_C \neg Q_D \vee \neg Q_A Q_D \vee \neg Q_B Q_D \vee \neg Q_C Q_D]_n$$
$$= [Q_A Q_B Q_C \neg Q_D \vee Q_D (\neg Q_A \vee \neg Q_B \vee \neg Q_C)]_n$$

$$Q_{D(n+1)} = [Q_A Q_B Q_C \neg Q_D \vee \neg(Q_A Q_B Q_C) Q_D]_n$$

III. Synchronzähler

- Berechnung von Synchronzählern - 4-Bit-Synchron-Dual-Vorwärtszähler

- Schritt 3: charakteristische Gleichung der zu verwendenden Flipflops bestimmen

- allgemein gilt für JK-Flipflops:

$$Q_{(n+1)} = [J \neg Q \vee \neg K Q]_n$$

- 4-Bit-Zähler → 4 charakteristische Gleichungen

$$Q_{A(n+1)} = [J_A \neg Q_A \vee \neg K_A Q_A]_n$$

$$Q_{B(n+1)} = [J_B \neg Q_B \vee \neg K_B Q_B]_n$$

$$Q_{C(n+1)} = [J_C \neg Q_C \vee \neg K_C Q_C]_n$$

$$Q_{D(n+1)} = [J_D \neg Q_D \vee \neg K_D Q_D]_n$$

II. Zähler und Frequenzteiler

III. Synchronzähler

- Berechnung von Synchronzählern - 4-Bit-Synchron-Dual-Vorwärtzähler
 - Schritt 4: Verknüpfungsgleichungen durch Koeffizientenvergleich bestimmen

- erstes JK-Flipflop

$$\begin{array}{ll} Q_{A(n+1)} = [J_A \neg Q_A \vee \neg K_A Q_A]_n & J_A = 1 \\ Q_{A(n+1)} = \neg Q_{An} & \Rightarrow K_A = 1 \end{array}$$

- zweites JK-Flipflop

$$\begin{array}{ll} Q_{B(n+1)} = [J_B \neg Q_B \vee \neg K_B Q_B]_n & J_B = Q_A \\ Q_{B(n+1)} = [Q_A \neg Q_B \vee \neg Q_A Q_B]_n & \Rightarrow K_B = Q_A \end{array}$$

II. Zähler und Frequenzteiler

III. Synchronzähler

- Berechnung von Synchronzählern - 4-Bit-Synchron-Dual-Vorwärtszähler
 - Schritt 4: Verknüpfungsgleichungen durch Koeffizientenvergleich bestimmen

- drittes JK-Flipflop

$$Q_{C(n+1)} = [J_C \neg Q_C \vee \neg K_C Q_C]_n \quad J_C = Q_A Q_B$$

$$Q_{C(n+1)} = [Q_A Q_B \neg Q_C \vee \neg (Q_A Q_B) Q_C]_n \quad K_C = Q_A Q_B$$

- viertes JK-Flipflop

$$Q_{D(n+1)} = [J_D \neg Q_D \vee \neg K_D Q_D]_n \quad J_D = Q_A Q_B Q_C$$

$$Q_{D(n+1)} = [Q_A Q_B Q_C \neg Q_D \vee \neg (Q_A Q_B Q_C) Q_D]_n \quad K_D = Q_A Q_B Q_C$$

II. Zähler und Frequenzteiler

III. Synchronzähler

- Berechnung von Synchronzählern - 4-Bit-Synchron-Dual-Vorwärtszähler
 - › Schritt 5: Schaltbild auf Basis der Verknüpfungsgleichungen zeichnen

$$J_A = 1$$

$$J_B = Q_A$$

$$J_C = Q_A Q_B$$

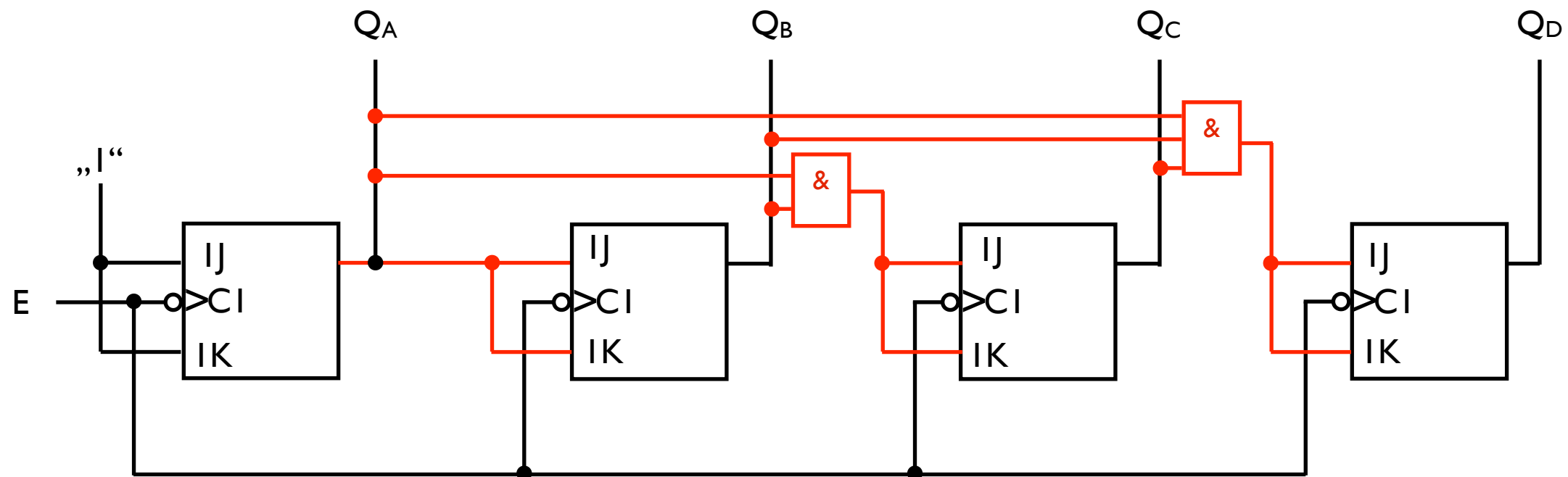
$$J_D = Q_A Q_B Q_C$$

$$K_A = 1$$

$$K_B = Q_A$$

$$K_C = Q_A Q_B$$

$$K_D = Q_A Q_B Q_C$$



III. Synchronzähler

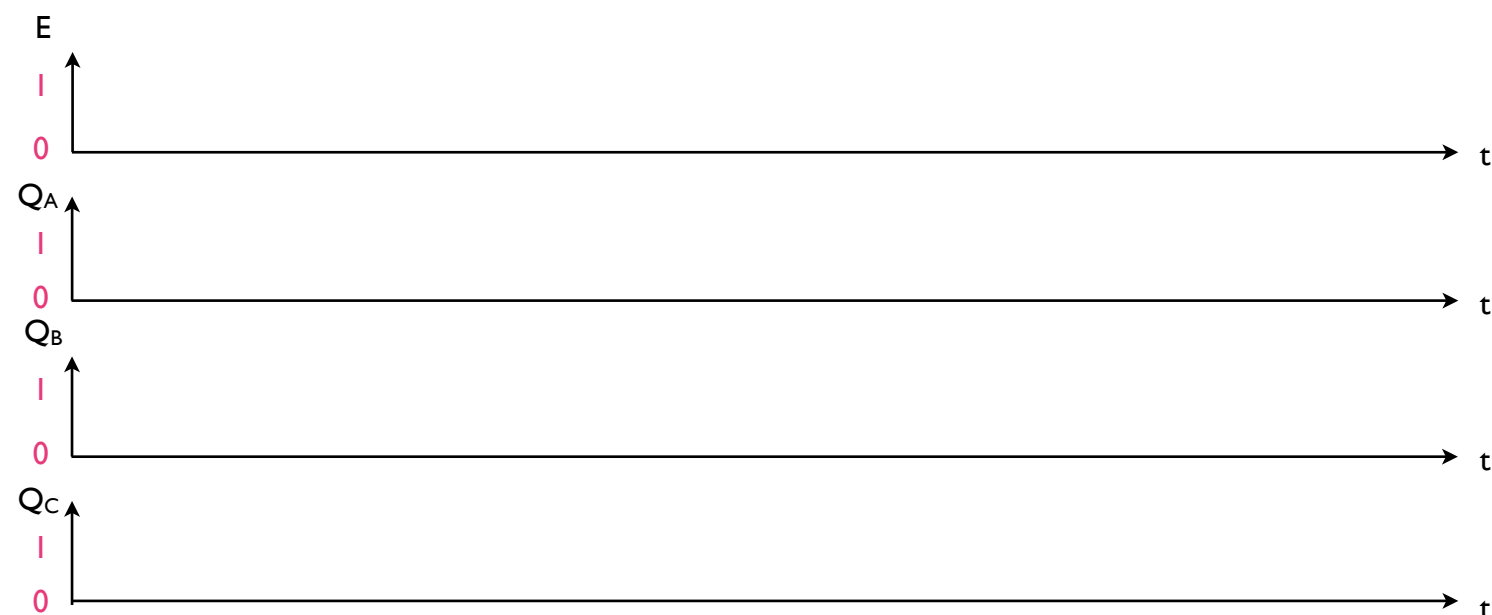
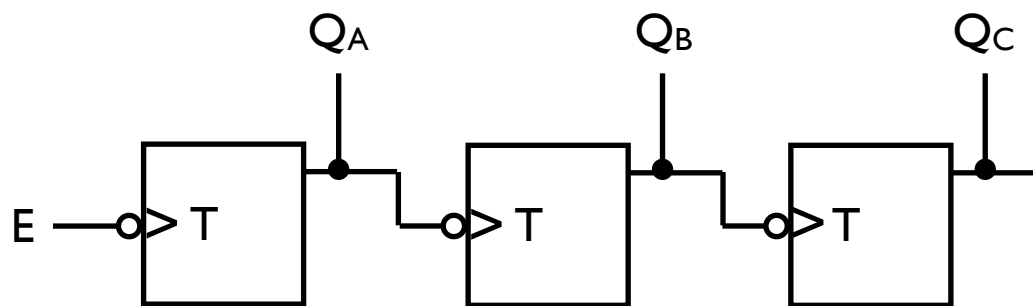
- Berechnung von Synchronzählern - 4-Bit-Synchron-BCD-Vorwärtzähler

Aufgabe: Berechnen Sie die Verknüpfungsgleichungen eines synchronen BCD-Vorwärtzählers und erstellen Sie die Schaltung. Zur Verfügung stehen JK-Master-Slave-Flipflops.



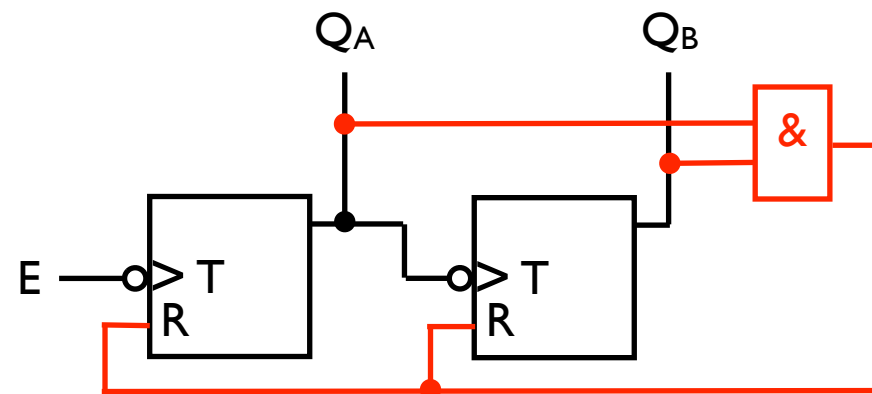
IV. Frequenzteiler

- Schaltungen, die Frequenzen in einem gewünschten Verhältnis aufteilen
- Unterscheidung zwischen Frequenzteiler mit festem und mit einstellbarem Teilverhältnis
- asynchrone Frequenzteiler mit festem Teilverhältnis
 - Einsatz von Asynchron-Dualzählern möglich
 - Anzahl der verwendeten Flipflops definiert geteilte Frequenz (jedes FF teilt Frequenz durch zwei)
 - Beispiel: asynchroner 3-Bit-Dual-Vorwärtszähler



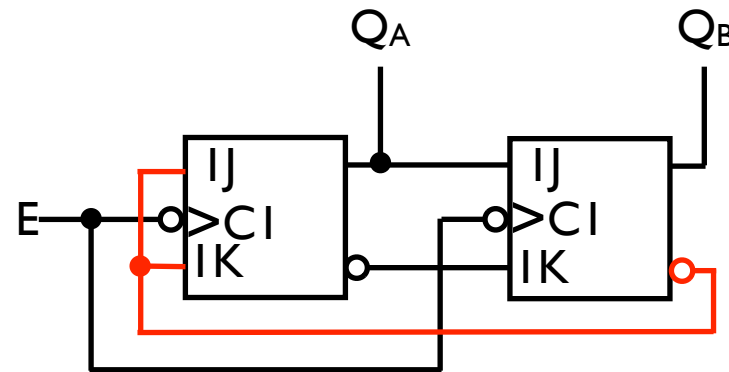
IV. Frequenzteiler

- asynchrone Frequenzteiler mit festem Teilverhältnis
 - Beispiel: asynchroner Dual-Vorwärtszähler mit Teilverhältnis 3:1
 - Verwendung von Rückstelleingängen nötig



IV. Frequenzteiler

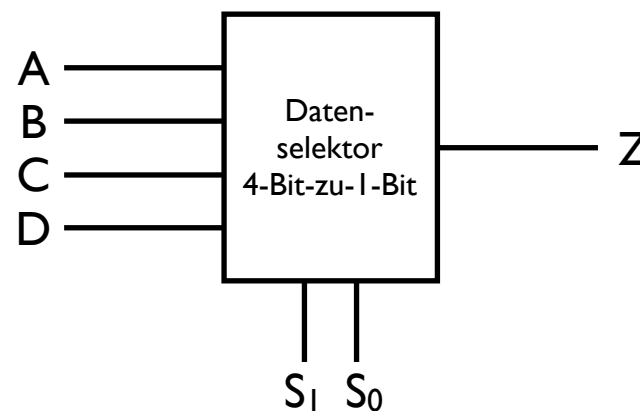
- synchrone Frequenzteiler mit festem Teilverhältnis
 - gleiches Prinzip wie bei asynchronen Frequenzteilern mit Verhältnis $2^n : 1$
 - Beschaltung der Eingänge bei ungeraden Teilverhältnissen
 - Beispiel: synchroner Dual-Vorwärtzähler mit Teilverhältnis 3:1



I 2. Digitale Auswahl- und Verbindungsschaltungen

I. Datenselektor

- Aufgabe eines Datenselektors:
aus angebotenen Daten die gewünschten Daten auswählen und an
Ausgänge weiterleiten
- Beispiel 1: 4-Bit-zu-1-Bit-Datenselektor
 - › jeder der vier Eingänge (A, B, C, D) soll wahlweise mit Hilfe der Steuerleitungen (S_0, S_1) zum
Ausgang Z durchgeschaltet werden können



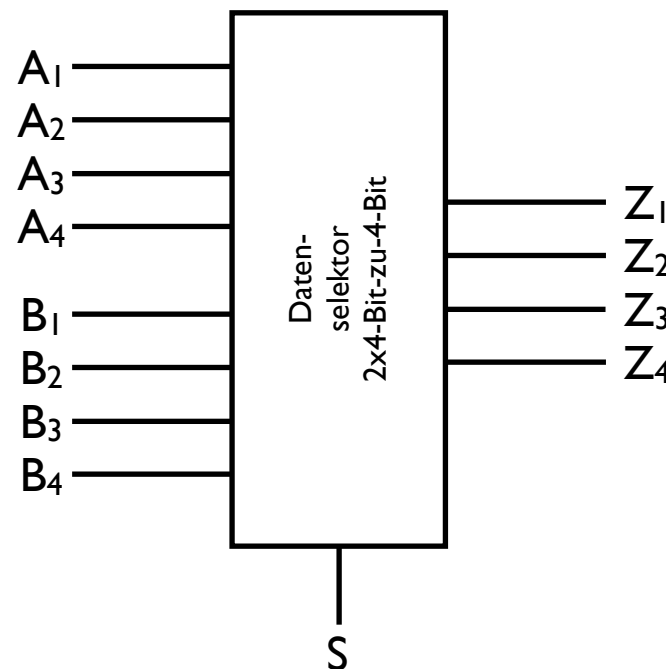
- › Zuordnung der Ein- zu den Ausgängen mit Schaltstufen (hier: 1 bis 4) → Wahrheitstabelle

Schaltstufe	S_1	S_0	Z
1	0	0	A
2	0	1	B
3	1	0	C
4	1	1	D

12. Digitale Auswahl- und Verbindungsschaltungen

I. Datenselektor

- Beispiel 2: 2 x 4-Bit-zu-4-Bit-Datenselektor
 - › jeweils vier Eingänge (A_{1-4} , B_{1-4}) sollen wahlweise mit Hilfe der Steuerleitung S zu den Ausgängen Z_{1-4} durchgeschaltet werden können



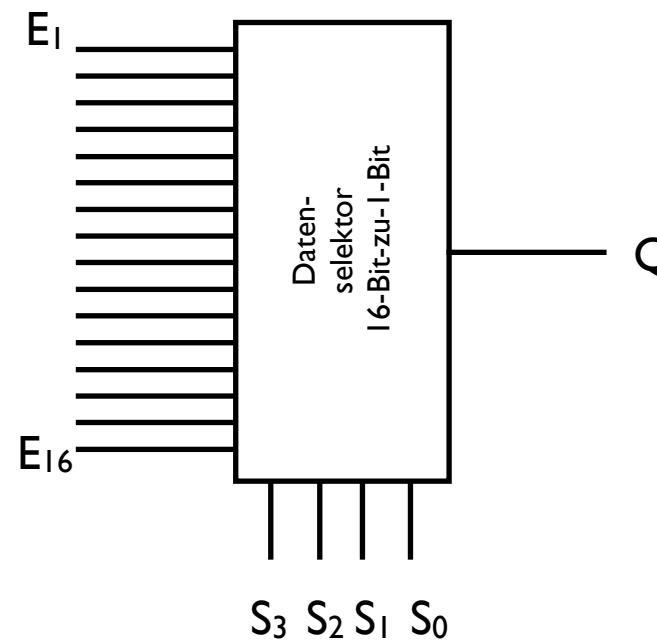
- › Zuordnung der Ein- zu den Ausgängen mit Schaltstufen (hier: 1 bis 2) → Wahrheitstabelle

Schaltstufe	S	Z_{1-4}
1	0	A_{1-4}
2	1	B_{1-4}

- › **AUFGABE:** Skizzieren Sie die Schaltung eines 2 x 4-Bit-zu-4-Bit-Datenselektors!

II. Multiplexer

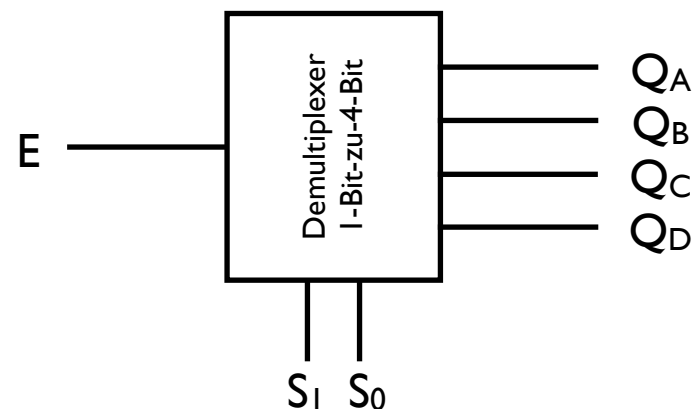
- zeitabhängig gesteuerter Datenselektor
- Beispiel: 16-Bit-zu-1-Bit-Datenselektor-Multiplexer



- gezielte zeitliche Ansteuerung der Eingänge → Multiplexer
 - Signale an Steuereingängen von „0000“ bis „1111“ mit definierter zeitlicher Dauer
 - anschließend erneutes Durchlaufen des Zyklus

II. Demultiplexer

- Arbeitsweise invers zu der des Multiplexers:
Durchschalten eines Eingangs wahlweise auf mehrere Ausgänge mit Hilfe der Steuereingänge
- Beispiel: 1-Bit-zu-4-Bit-Demultiplexer



› Wahrheitstabelle

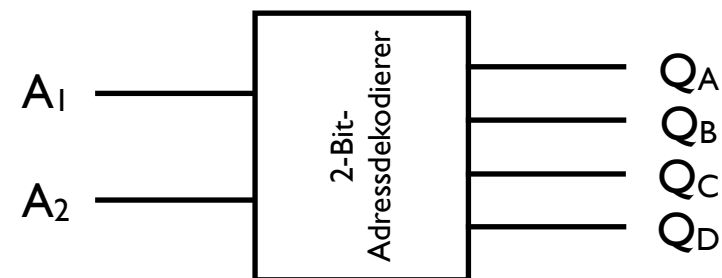
Schaltstufe	S_1	S_0	E
1	0	0	Q_A
2	0	1	Q_B
3	1	0	Q_C
4	1	1	Q_D

› **AUFGABE:** Skizzieren Sie die Schaltung eines 1-Bit-zu-4-Bit-Demultiplexers!

I 2. Digitale Auswahl- und Verbindungsschaltungen

III. Adressdekodierer

- Um Bausteine anzusteuern werden Adressen benötigt
- Adresse: binäres Wort mit definierter Länge (Anzahl von Bits)
- Adressdekodierer liefert eine „1“ an einem bestimmten Ausgang, der über die Adresseingänge angewählt werden kann
- Beispiel: 2-Bit-Adressdekodierer



› Wahrheitstabelle des 2-Bit-Adressdekodierers

Adresse	A ₂	A ₁	Q _A	Q _B	Q _C	Q _D
1	0	0	1	0	0	0
2	0	1	0	1	0	0
3	1	0	0	0	1	0
4	1	1	0	0	0	1

12. Digitale Auswahl- und Verbindungsschaltungen

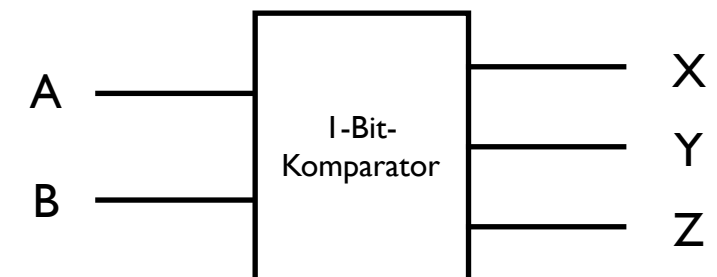
IV. Digitaler Komparator

- ermöglicht Vergleich zweier binärer Ausdrücke A und B
- Information am Ausgang: $(A < B) \vee (A = B) \vee (A > B) \rightarrow$ drei Ausgänge nötig
- Vergleich nur möglich, wenn beide Ausdrücke im gleichen Zahlensystem vorliegen
- übliche Komparatoren für duales Zahlensystem und BCD-Kode entwickelt
- Beispiel: 1-Bit-Komparator

‣ binäre Ausdrücke A und B haben eine Länge von einem Bit

‣ Zuordnung der Ausgänge:

- Ausgang X = 1, falls $A > B$
- Ausgang Y = 1, falls $A = B$
- Ausgang Z = 1, falls $A < B$



Fall	B	A	X	Y	Z	
1	0	0	0	1	0	$Y = \neg A \neg B$
2	0	1	1	0	0	$X = A \neg B$
3	1	0	0	0	1	$Z = \neg A B$
4	1	1	0	1	0	$Y = A B$

I 2. Digitale Auswahl- und Verbindungsschaltungen

IV. Digitaler Komparator

- Beispiel: 1-Bit-Komparator

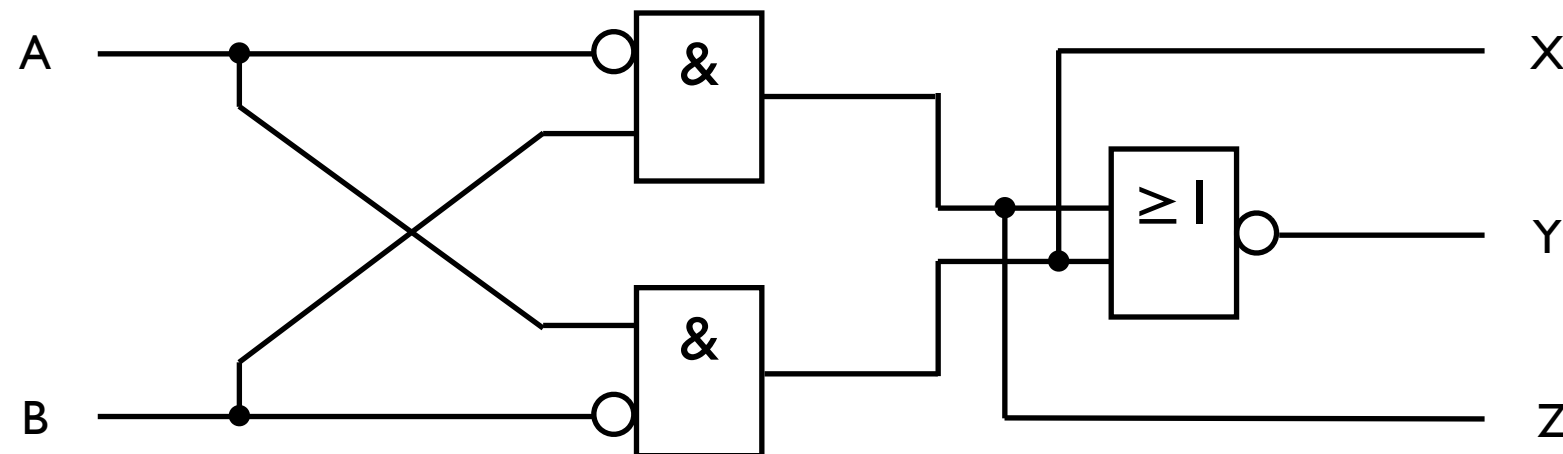
- › Zusammenfassung:

$$X = A \neg B$$

$$Y = \neg A \neg B \vee AB = \neg (\neg AB \vee A \neg B)$$

$$Z = \neg AB$$

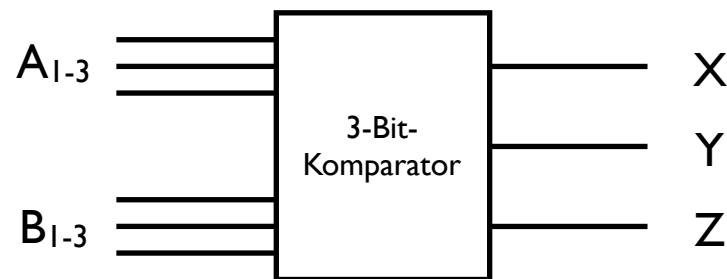
- › Schaltung des 1-Bit-Komparators



I 2. Digitale Auswahl- und Verbindungsschaltungen

IV. Digitaler Komparator

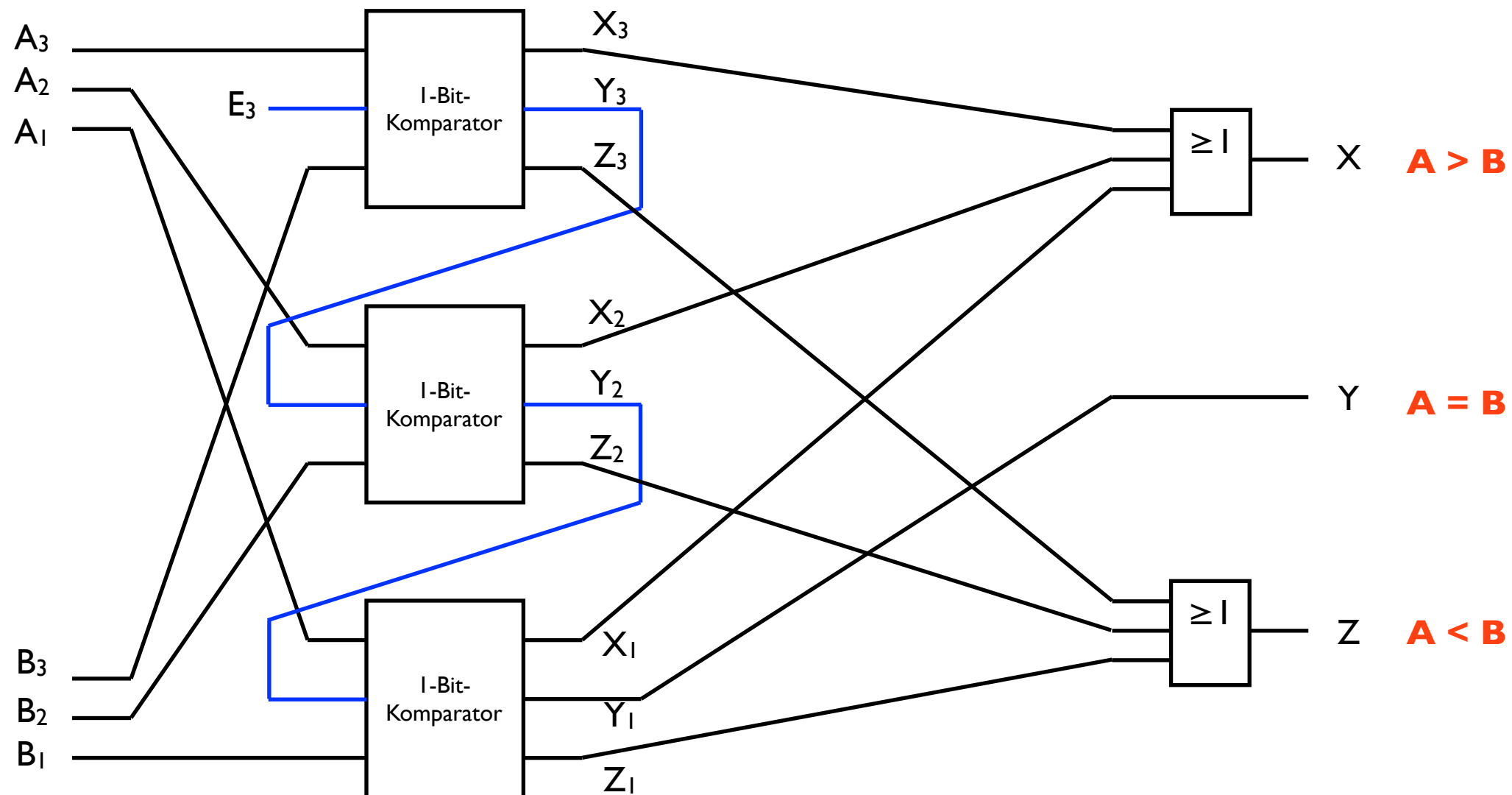
- Beispiel: 3-Bit-Komparator
 - sechs Variable → 64 Einträge in der Wahrheitstabelle
⇒ Reduzierung



Fall	2^2	2^1	2^0	$A > B$	$A = B$	$A < B$
	A_3, B_3	A_2, B_2	A_1, B_1	X	Y	Z
1	$A_3 > B_3$	x	x	1	0	0
2	$A_3 < B_3$	x	x	0	0	1
3	$A_3 = B_3$	$A_2 > B_2$	x	1	0	0
4	$A_3 = B_3$	$A_2 < B_2$	x	0	0	1
5	$A_3 = B_3$	$A_2 = B_2$	$A_1 > B_1$	1	0	0
6	$A_3 = B_3$	$A_2 = B_2$	$A_1 < B_1$	0	0	1
7	$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	0	1	0

IV. Digitaler Komparator

- Beispiel: 3-Bit-Komparator
 - › Aufbau mit 3 1-Bit-Komparatoren (mit Sperrschaltung)



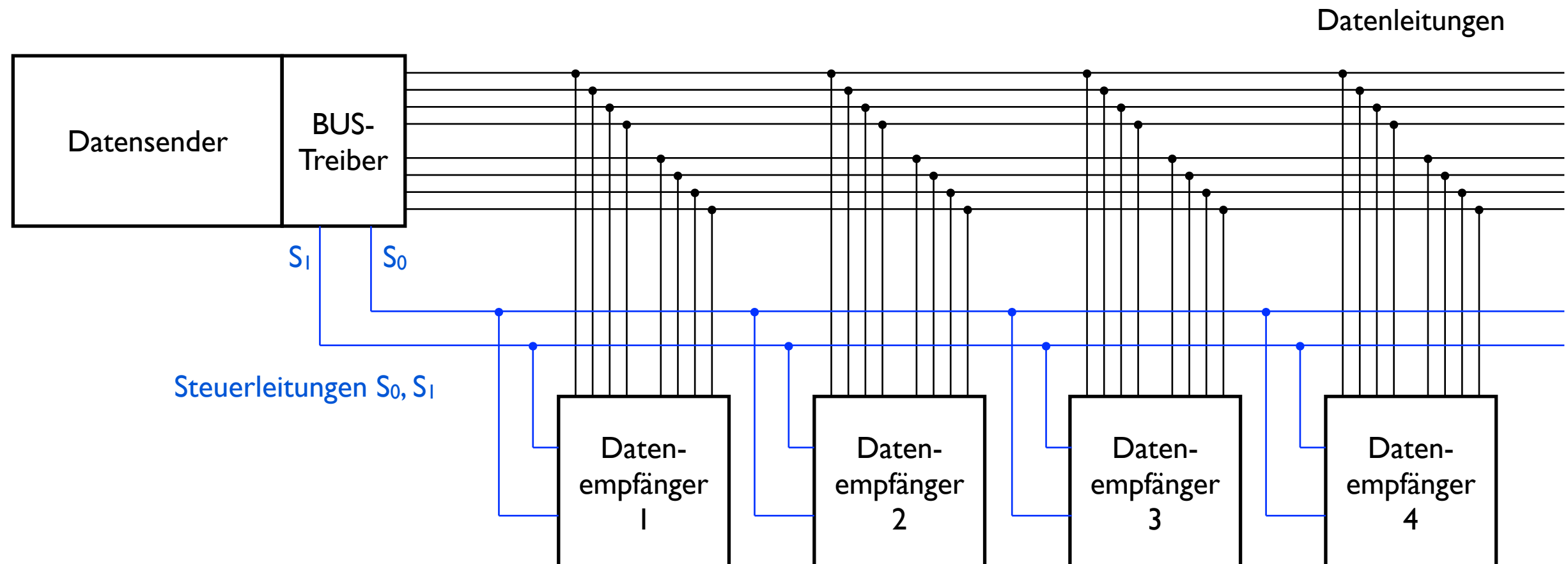
V. BUS-Schaltungen

- Bus: System zum Transport und zur Verteilung binärer Informationen
- Sender und Empfänger binärer Informationen sind durch ein BUS-System miteinander verbunden
- unidirektionaler Bus: Informationen können nur in eine Richtung transportiert werden (Einweg-Bus)
- bidirektionaler Bus: Informationsfluss ist in beide Richtungen möglich (Zweiweg-Bus)
- paralleles BUS-System: eine Leitung für jedes Bit verfügbar (Daten- und Steuerleitungen)
- serielles BUS-System: eine Leitung ausreichend; Bits werden sequentiell versendet
 - ➔ Nachteile:
 - langsamer als parallele Systeme
 - höherer Schaltungsaufwand (Einsatz von Parallel-Seriell- und Seriell-Parallel-Umsetzer)
 - ➔ Einsatz bei großen Leitungslängen (Reduzierung der Leitungskosten)

I 2. Digitale Auswahl- und Verbindungsschaltungen

V. BUS-Schaltungen

- paralleles unidirektionales BUS-System

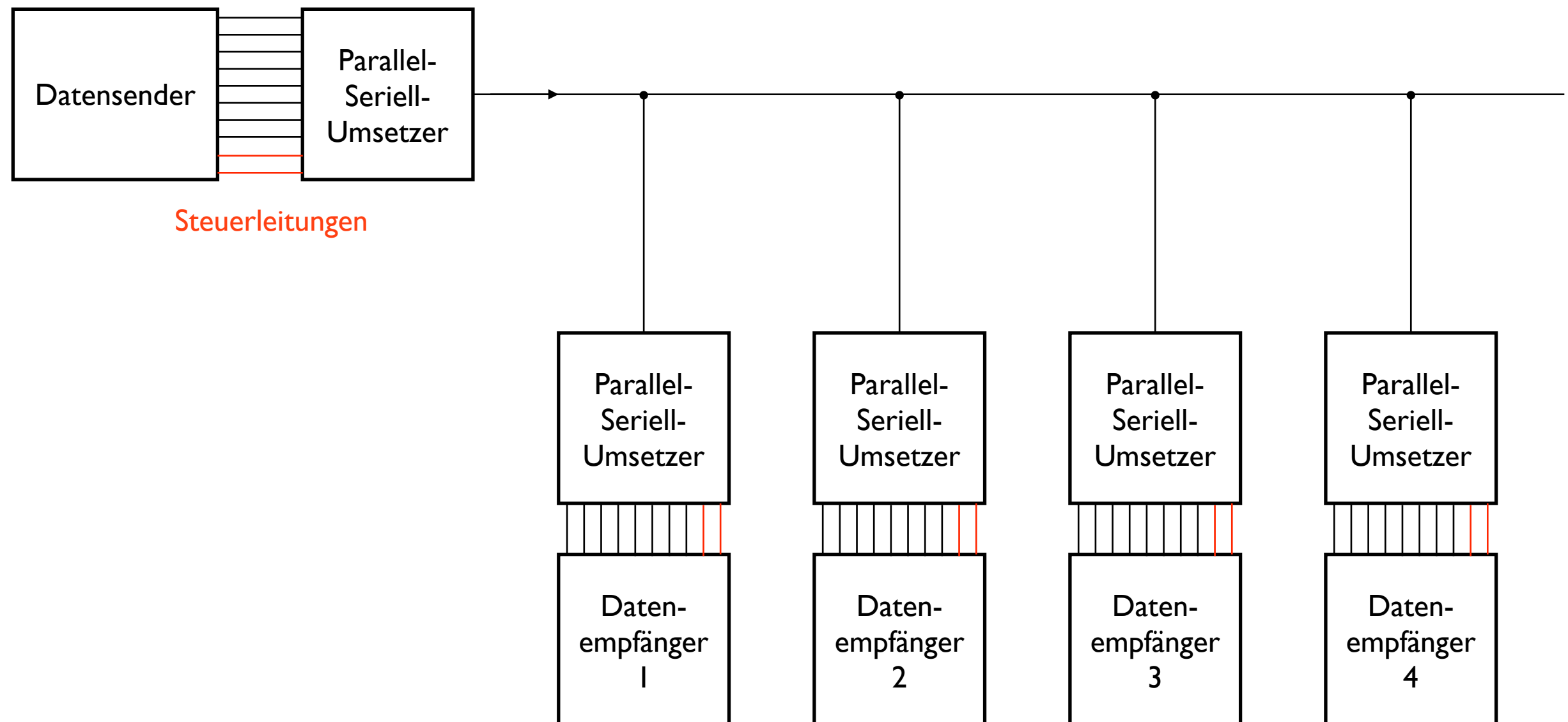


S_1	S_0	Empfänger
0	0	1
0	1	2
1	0	3
1	1	4

I 2. Digitale Auswahl- und Verbindungsschaltungen

V. BUS-Schaltungen

- serielles unidirektionales BUS-System



I 3. Register- und Speicherschaltungen

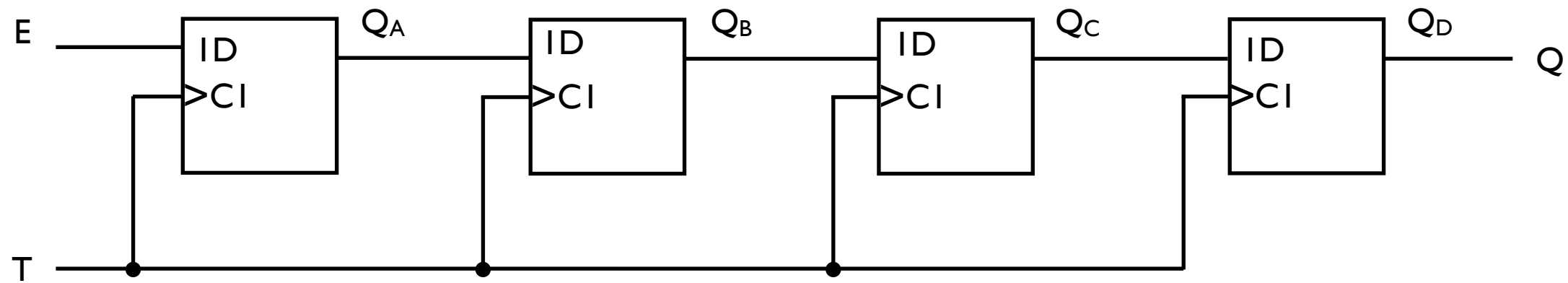
I. Schieberegister

- Arbeitsweise
 - ➔ taktgesteuerte Aufnahme von Informationen
 - ➔ Speicherung der Information
 - ➔ Ausgabe der Information
- Aufbau der Schieberegister mit Flipflops (D-, SR-, JK-FF; Master-Slave-FF)
- Herstellung von Schieberegistern als integrierte Schaltung
- Aufbau von Schieberegistern (Unterscheidung nach Ein- und Ausgabeart)
 - ➔ serielle Ein- und Ausgabe (immer gegeben)
 - ➔ zusätzliche parallele Ausgabe
 - ➔ zusätzliche parallele Ein- und Ausgabe
 - ➔ Ringregister

I 3. Register- und Speicherschaltungen

I. Schieberegister

- serielle Ein- und Ausgabe
 - ➔ Aufbau mit D-Flipflops und 4-Bit-Speicherkapazität



➔ Funktionsweise

- $(E = 1) \wedge (T: 0 \rightarrow 1) \rightarrow Q_A = 1$: Information wird am Eingang aufgenommen
- $(E = 0) \wedge (T: 0 \rightarrow 1) \rightarrow (Q_A = 0) \wedge (Q_B = 1)$: Information wird von FF_A an FF_B übergeben
- $(E = 0) \wedge (T: 0 \rightarrow 1) \rightarrow (Q_A = 0) \wedge (Q_B = 0) \wedge (Q_C = 1)$: Information wird von FF_B an FF_C übergeben
- $(E = 0) \wedge (T: 0 \rightarrow 1) \rightarrow (Q_A = 0) \wedge (Q_B = 0) \wedge (Q_C = 0) \wedge (Q_D = 1)$: Information wird von FF_C an FF_D übergeben \Rightarrow Information liegt am Ausgang des Schieberegisters an
- $(E = 0) \wedge (T: 0 \rightarrow 1) \rightarrow (Q_A = 0) \wedge (Q_B = 0) \wedge (Q_C = 0) \wedge (Q_D = 0)$: Schieberegister ist „leer“

I 3. Register- und Speicherschaltungen

I. Schieberegister

- serielle Ein- und Ausgabe

Aufgabe: Sie möchten die Zahl 1011 in ein 4-Bit-Schieberegister einlesen.

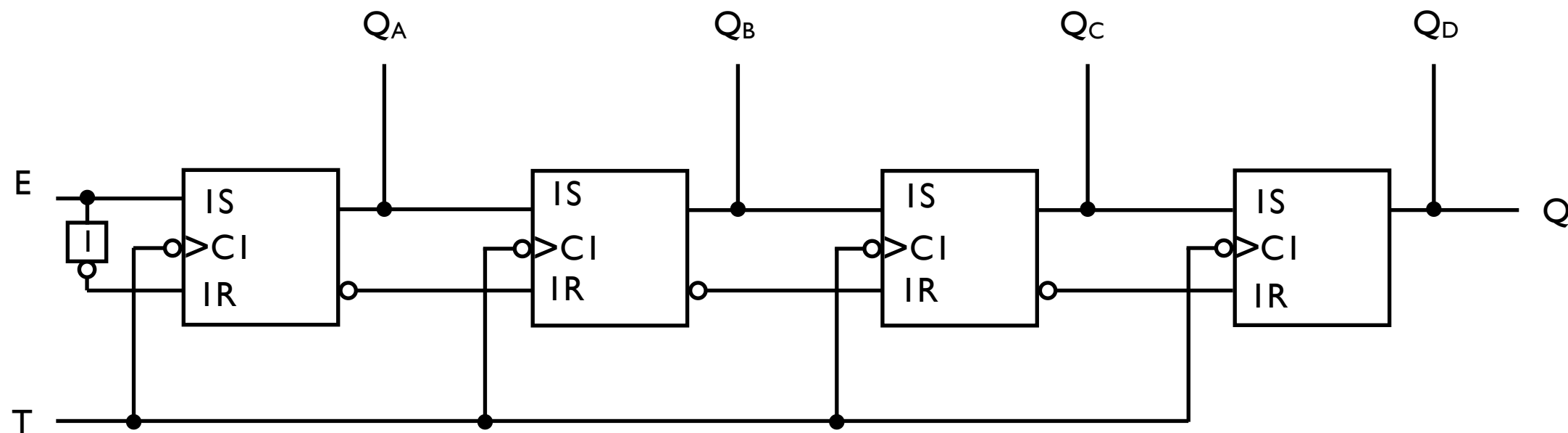
- a) Wie viele Takte sind zum Einlesen erforderlich?
- b) Zeichnen Sie das dazugehörige Zeitablaufdiagramm.
- c) Wie viele Takte benötigt man zum Auslesen?



13. Register- und Speicherschaltungen

I. Schieberegister

- parallele Ausgabe
 - ➔ Aufbau mit SR-Flipflops und 4-Bit-Speicherkapazität

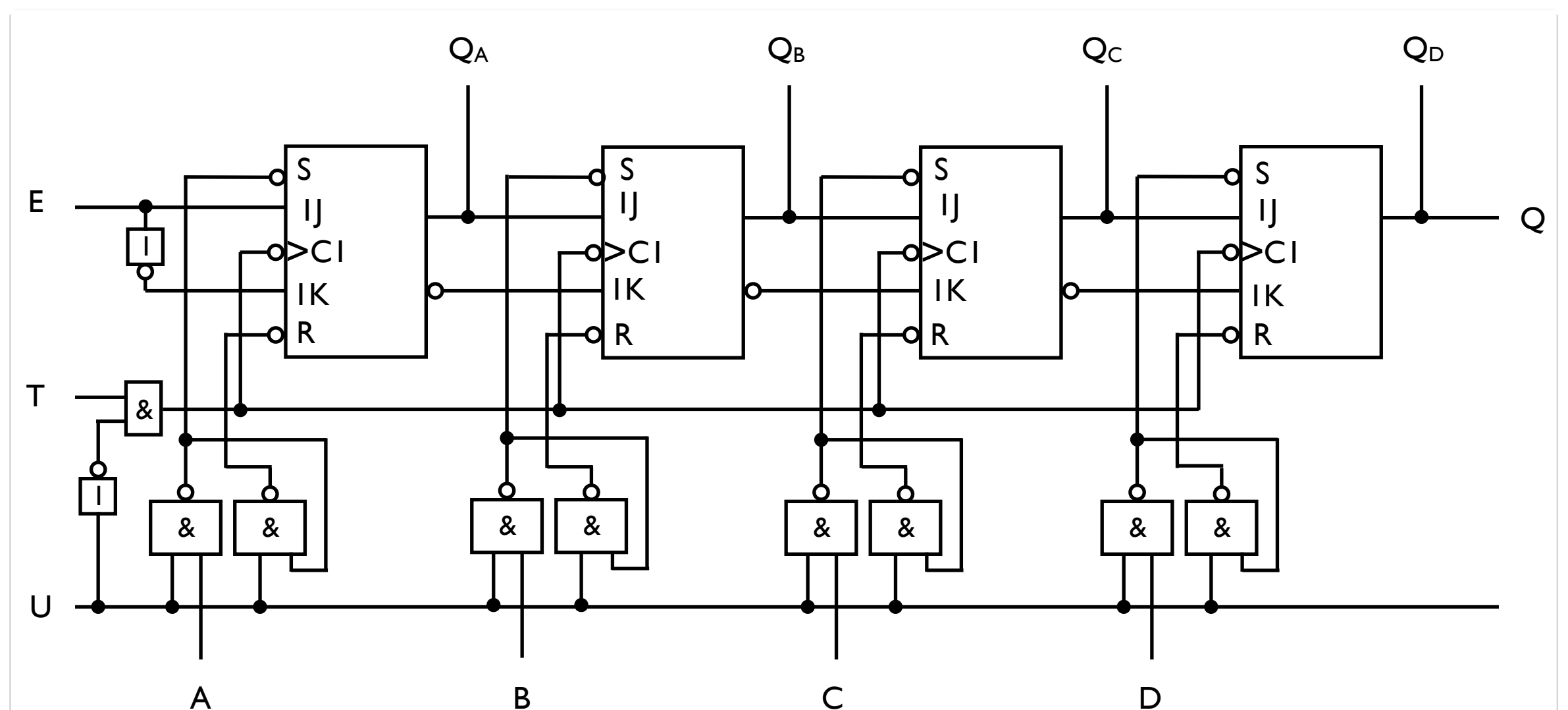


- ➔ taktunabhängige Ausgabe der Ausgänge Q_A bis Q_D
- ➔ bei weiteren Takten während der Parallelausgabe kann Ausgabewert verfälscht werden
- ➔ daher Verbot folgender Fälle
 - serielle plus parallele Datenausgabe
 - serielle Dateneingabe plus parallele Datenausgabe
- ➔ Abhilfe durch Verriegelungsschaltung

I 3. Register- und Speicherschaltungen

I. Schieberegister

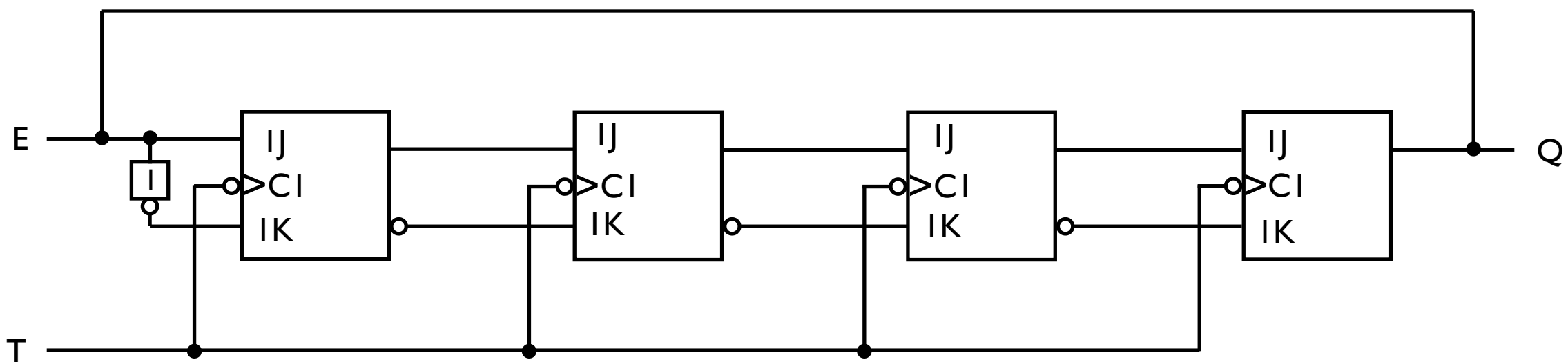
- parallele Ein- und Ausgabe
 - ➔ Aufbau mit JK-Flipflops und 4-Bit-Speicherkapazität



I 3. Register- und Speicherschaltungen

I. Schieberegister

- Ringregister
 - ➔ Schieberegister in dem der Ausgang wieder mit dem Eingang verbunden ist
 - ➔ alternative Bezeichnung: Umlaufregister
 - ➔ schematischer Aufbau:

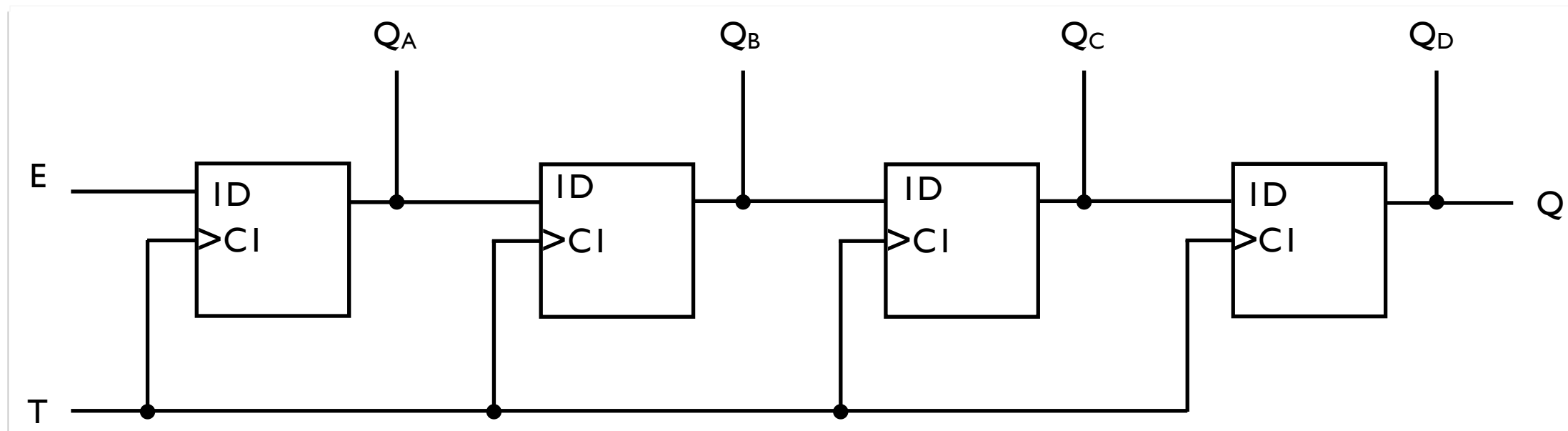


- ➔ serielle oder parallele Eingabe und Ausgabe möglich (Modifizierung der Schaltung nötig)
- ➔ Rücksetzen üblicherweise mit zusätzlichem taktunabhängigem Rücksetzeingang

I 3. Register- und Speicherschaltungen

I. Schieberegister

- umschaltbare Schieberichtung



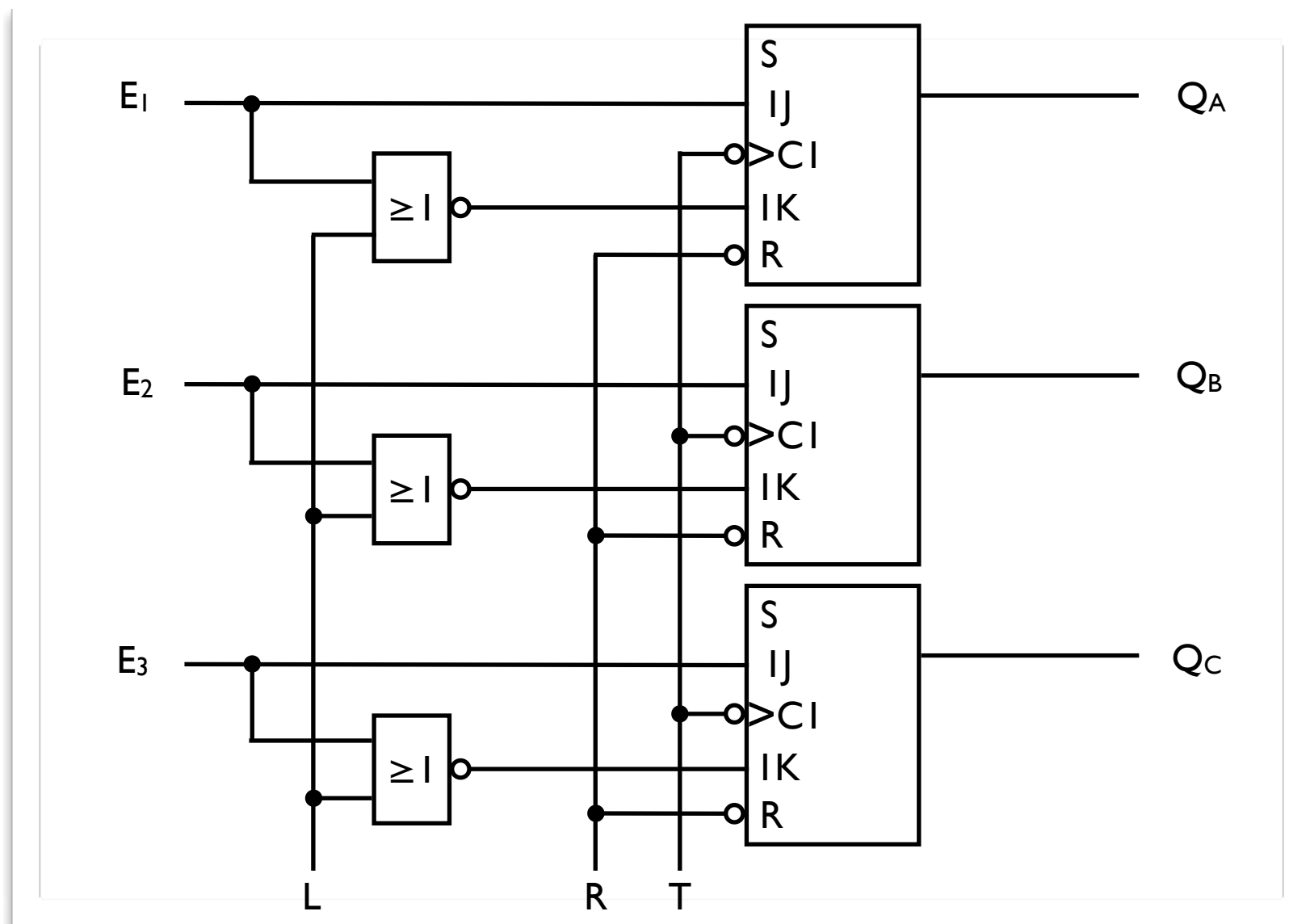
Aufgabe: Modifizieren Sie das abgebildete Schieberegister derart, dass die Schieberichtung (manuell) umgeschaltet werden kann.



I 3. Register- und Speicherschaltungen

II. Speicherregister

- kein Schieben der Informationen möglich
- Speicherung durch Setzen von Flipflops → Speicherung binärer Wörter
- Einsatz in Steuer- und Rechenschaltungen



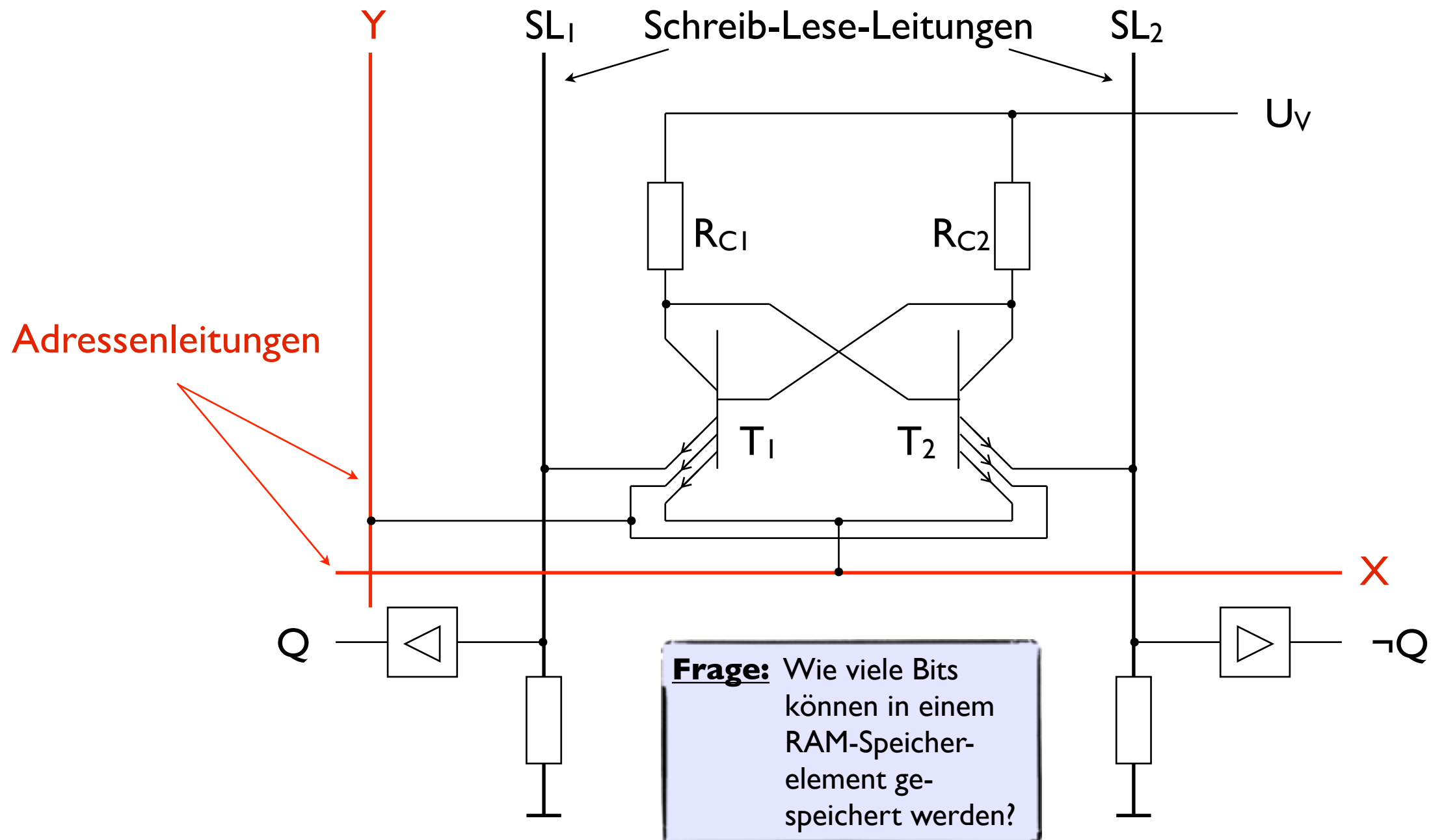
III. Random Access Memory (RAM)

- Schreib-Lese-Speicher (Speicher mit wahlfreiem Zugriff)
- spezifische Speicherkapazität pro Speicherplatz
- Adressierung der einzelnen Speicherplätze
- kein Löschen der Information nach dem Auslesen (Löschen muss gesondert durchgeführt werden)
- statischer RAM
 - ➔ Aufbau mit Flipflops (pro Bit ein Flipflop)
 - ➔ verwendete Schaltkreisfamilien:
 - TTL
 - ECL
 - N-MOS
 - C-MOS
- dynamischer RAM
 - ➔ Umsetzung durch interne (Gate-Substrat-)Kapazitäten von Transistoren
 - ➔ Problem: Ladungsverluste durch Leckströme → „Auffrischen“ nötig
 - ➔ verwendete Schaltkreisfamilien
 - sämtliche MOS-Technologien

I 3. Register- und Speicherschaltungen

IV. statischer RAM (SRAM)

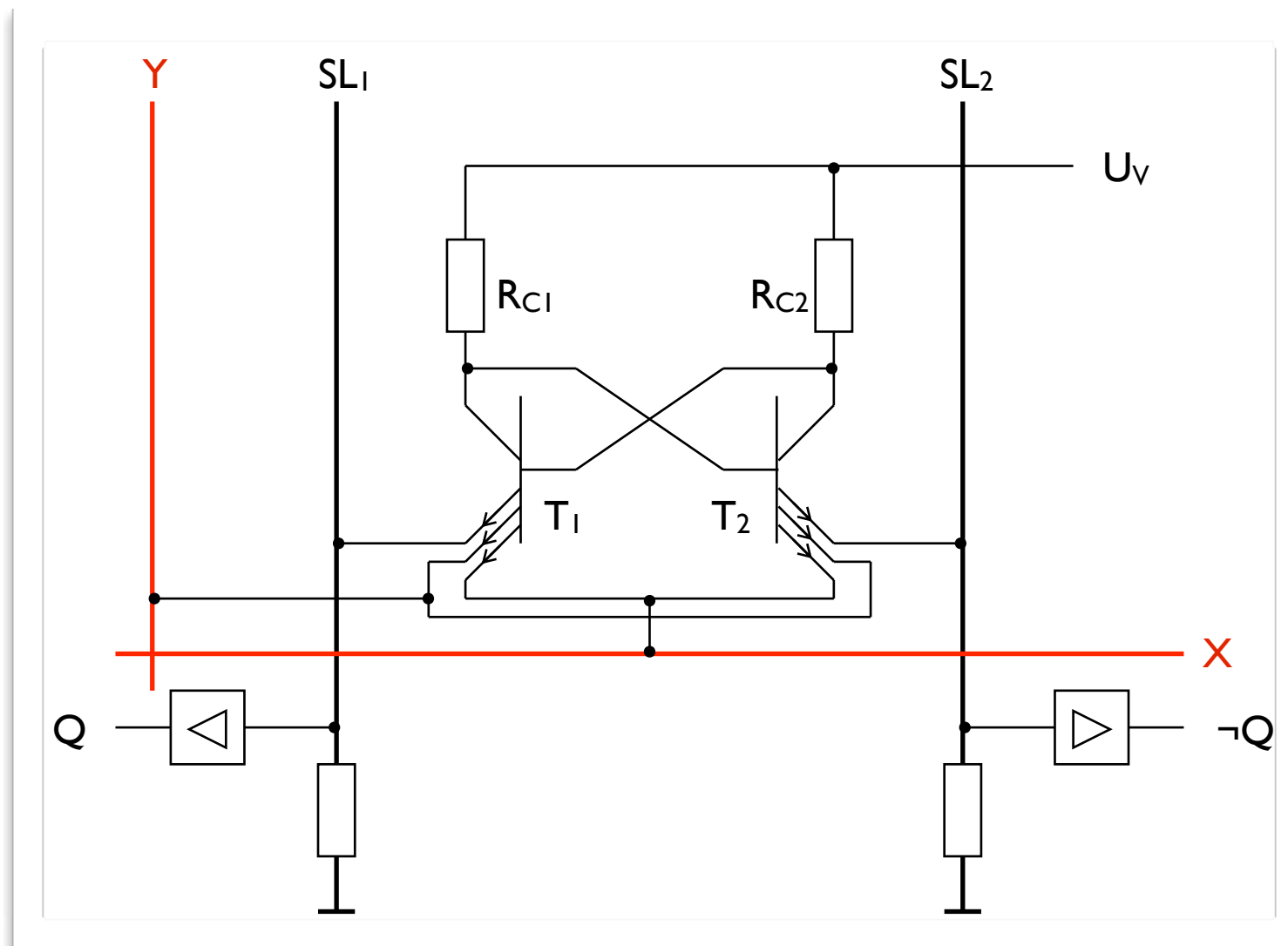
- Prinzipschaltung: Aufbau mit bipolaren Transistoren (TTL-Technik)



I 3. Register- und Speicherschaltungen

IV. statischer RAM (SRAM)

- Aktivierung der Speicherzelle (TTL-Technik)

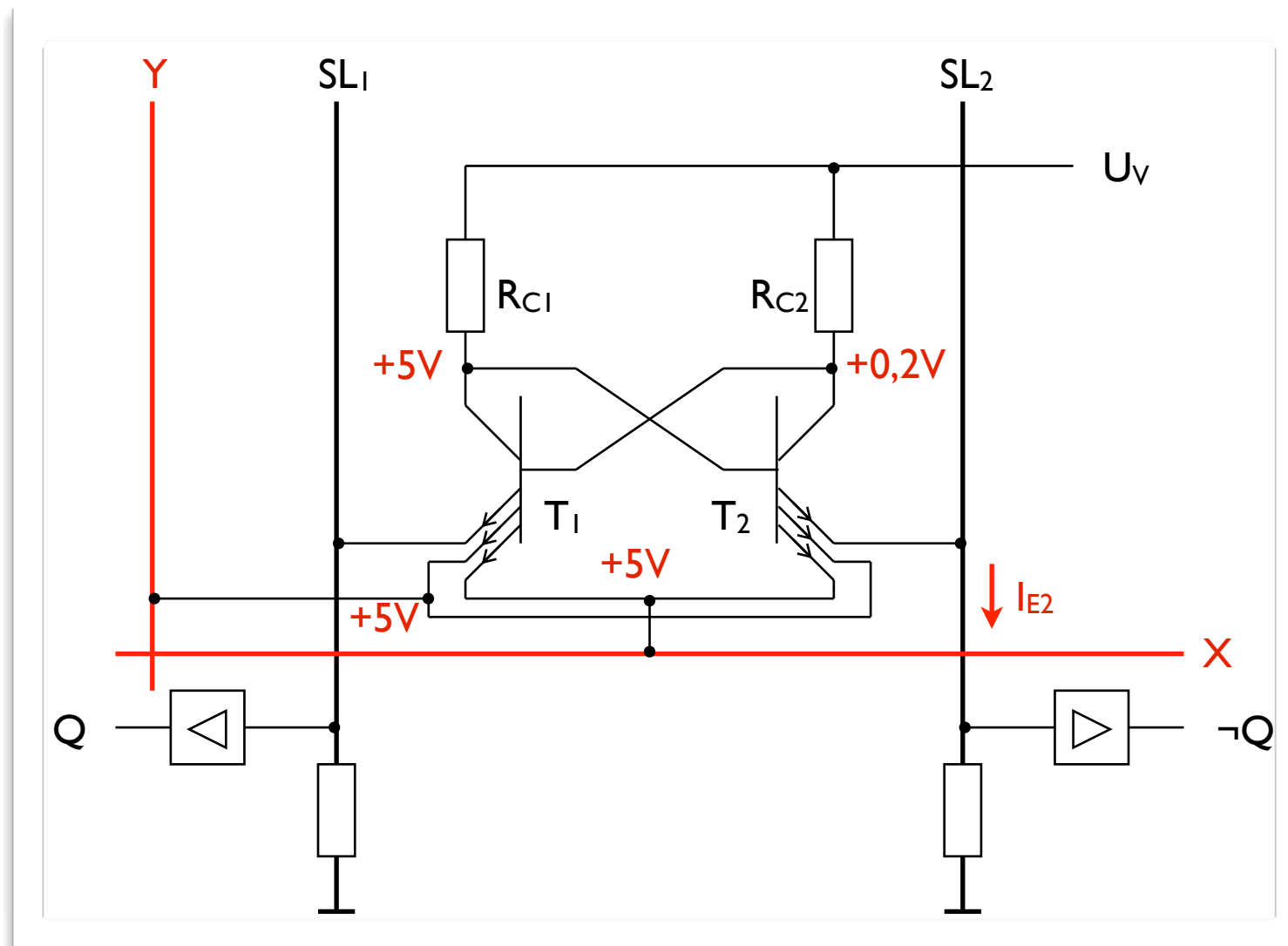


- ➔ inaktive Speicherzelle
 - „0“-Signal an beiden Adressenleitungen
 - Abfluss des Emitterstroms gegen Masse
 - „0“-Signal an einer Adressenleitung
 - Abfluss des Emitterstroms über andere Koordinatenleitung
- ➔ aktive Speicherzelle
 - „1“-Signal an beiden Adressenleitungen
 - Emitterstrom des leitenden Transistors fließt über seine SL-Leitung ab

I 3. Register- und Speicherschaltungen

IV. statischer RAM (SRAM)

- Lesevorgang (TTL-Technik)

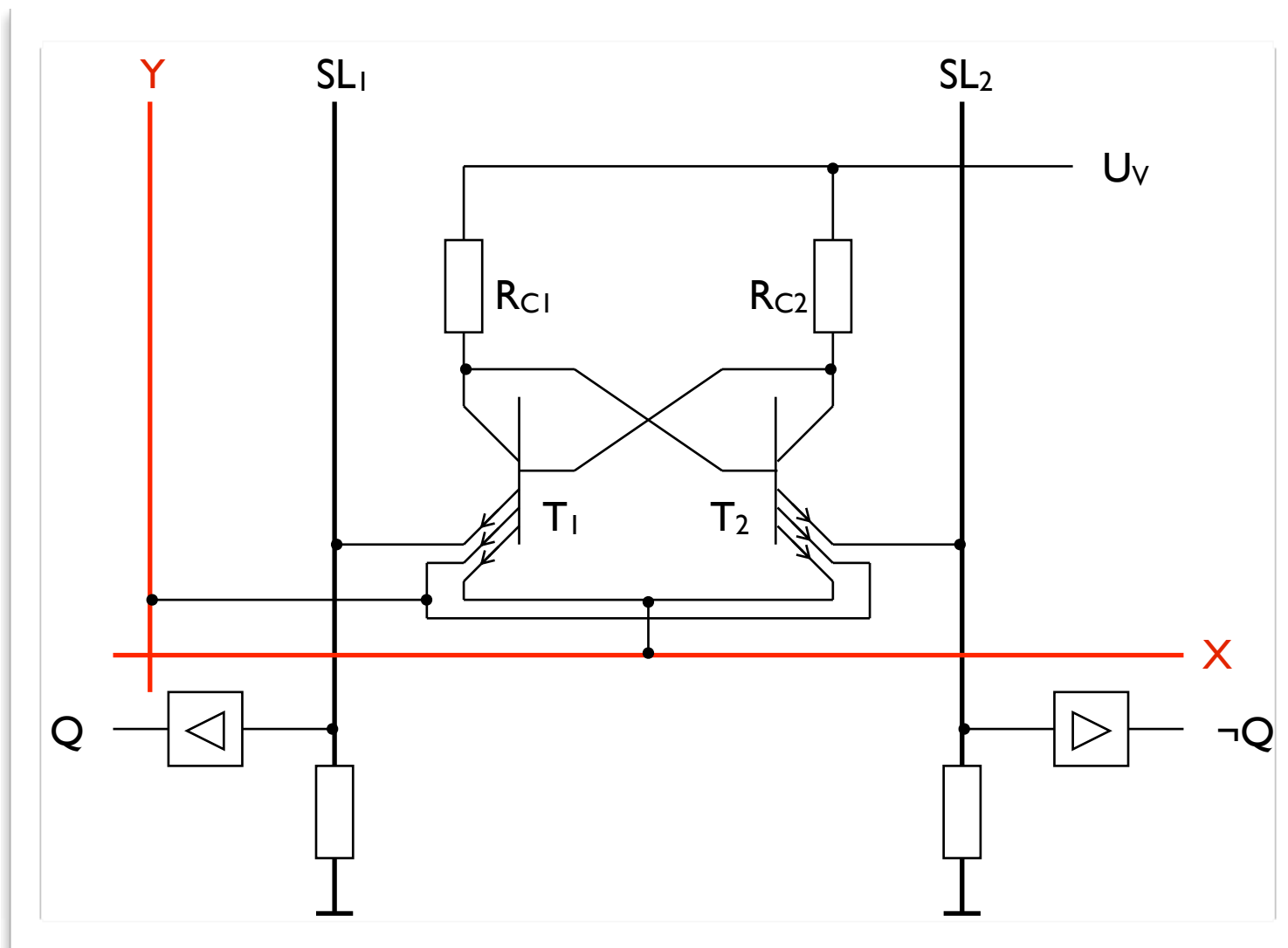


- ➔ Aktivierung der Speicherzelle nötig
- ➔ Leitender Transistor definiert Emitterstromfluss an zugehöriger SL-Leitung
- ➔ hier: T_2 leitend
- ➔ Ausgang $\neg Q$ liegt auf „1“
- ➔ Speicherelement hat den Wert „0“ gespeichert

I 3. Register- und Speicherschaltungen

IV. statischer RAM (SRAM)

- Schreibvorgang (TTL-Technik)

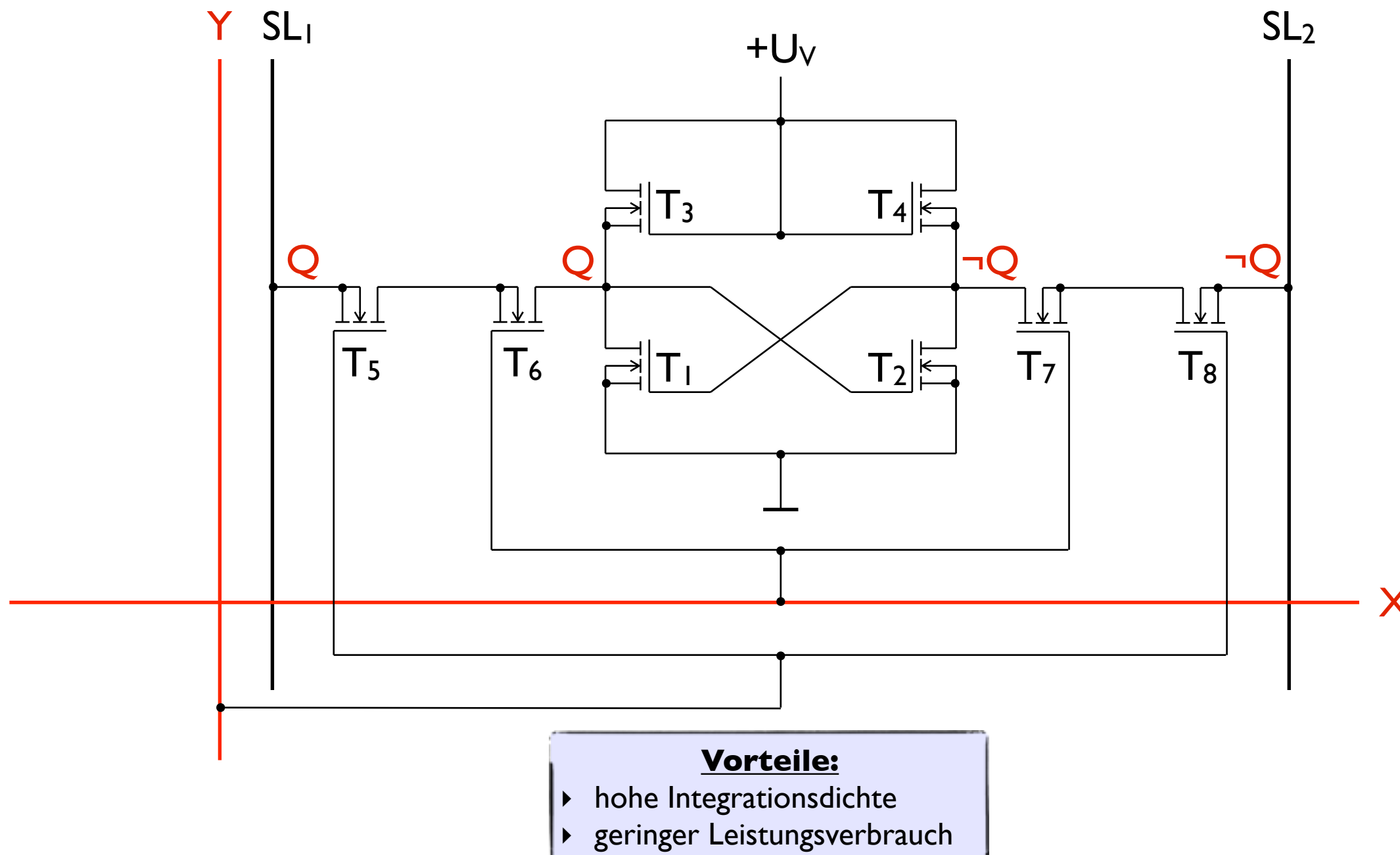


- ➔ Aktivierung der Speicherzelle nötig
- ➔ Annahme: Speicherelement hat den Wert „0“ gespeichert
- ➔ nun soll Speicherelement auf „1“ gesetzt werden
- ➔ SL_2 auf „1“ und SL_1 auf „0“ setzen
- ➔ Transistor T_2 sperrt, da alle Emitter auf „1“ liegen
- ➔ Transistor T_1 schaltet durch, da SL_1 auf „0“ liegt

I 3. Register- und Speicherschaltungen

IV. statischer RAM (SRAM)

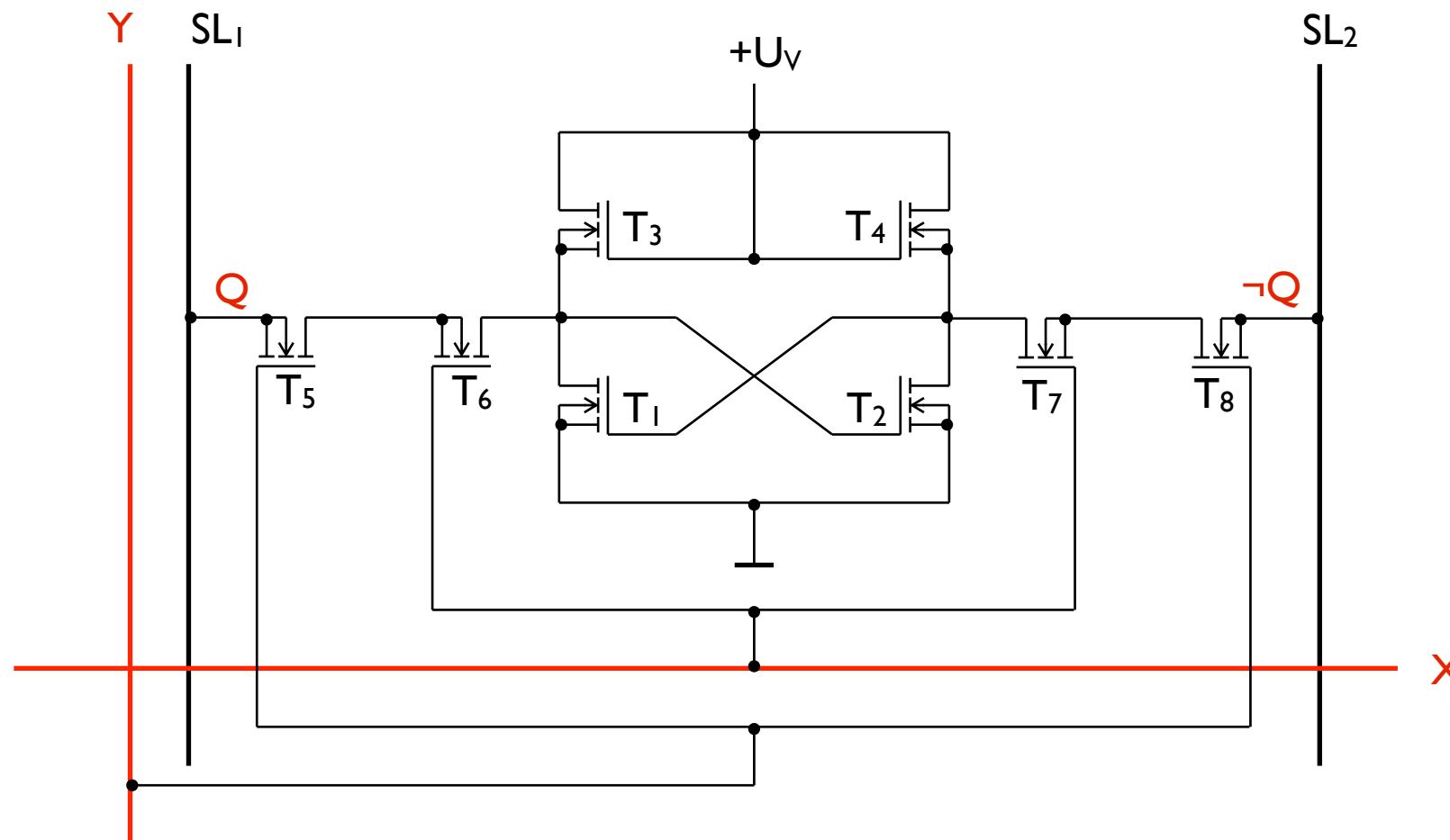
- Prinzipschaltung: Aufbau in NMOS-Technik



I 3. Register- und Speicherschaltungen

IV. statischer RAM (SRAM)

- Speicherzelle in NMOS-Technik

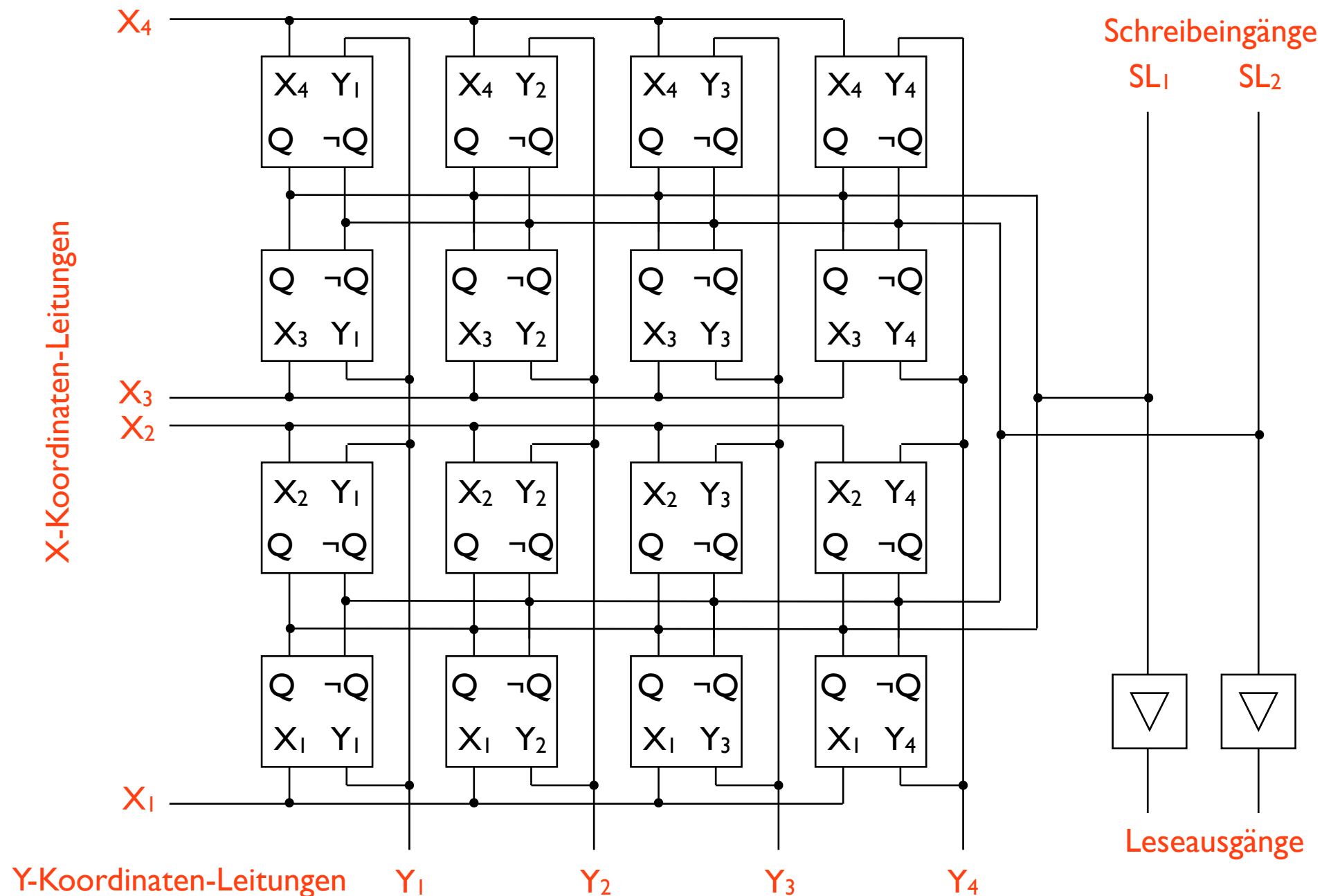


- ➡ **Aktivierung der Speicherzelle**
 - „I“-Signal an beiden Adressenleitungen
→ Transistoren T₅ bis T₈ steuern durch
 - Verbindung der Ausgänge des Flipflops mit den Schreib-Lese-Leitungen
- ➡ **Lesevorgang**
 - direkt nach Aktivierung möglich
- ➡ **Schreibvorgang**
 - Speicherelement sei auf „0“ gesetzt (T₁ leitend und T₂ gesperrt)
 - SL2 auf „0“ setzen
→ T₁ sperrt und T₂ wird leitend
 - Speicherelement liegt auf „I“

I 3. Register- und Speicherschaltungen

IV. statischer RAM (SRAM)

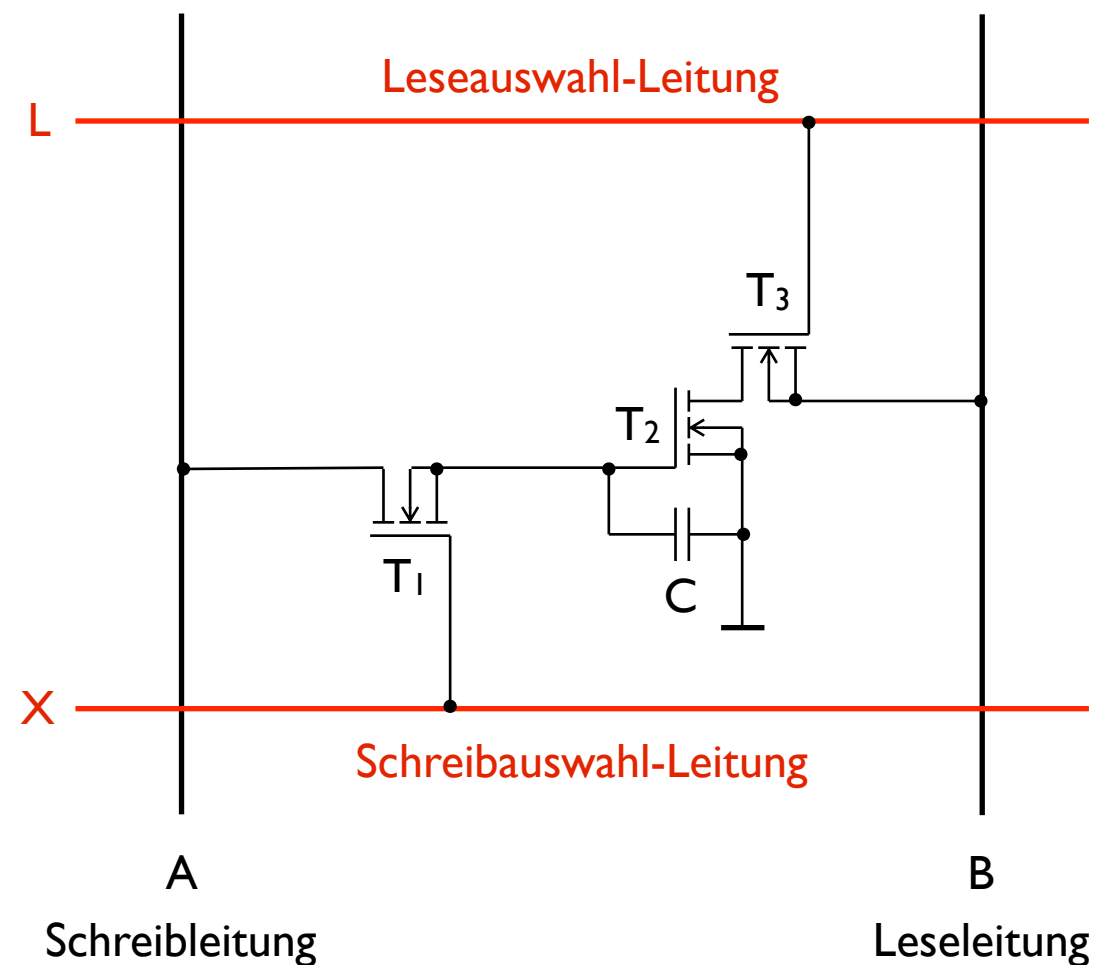
- Zusammenschaltung von Speicherelementen zu einer Speichermatrix (16Bit)



13. Register- und Speicherschaltungen

V. dynamischer RAM (DRAM)

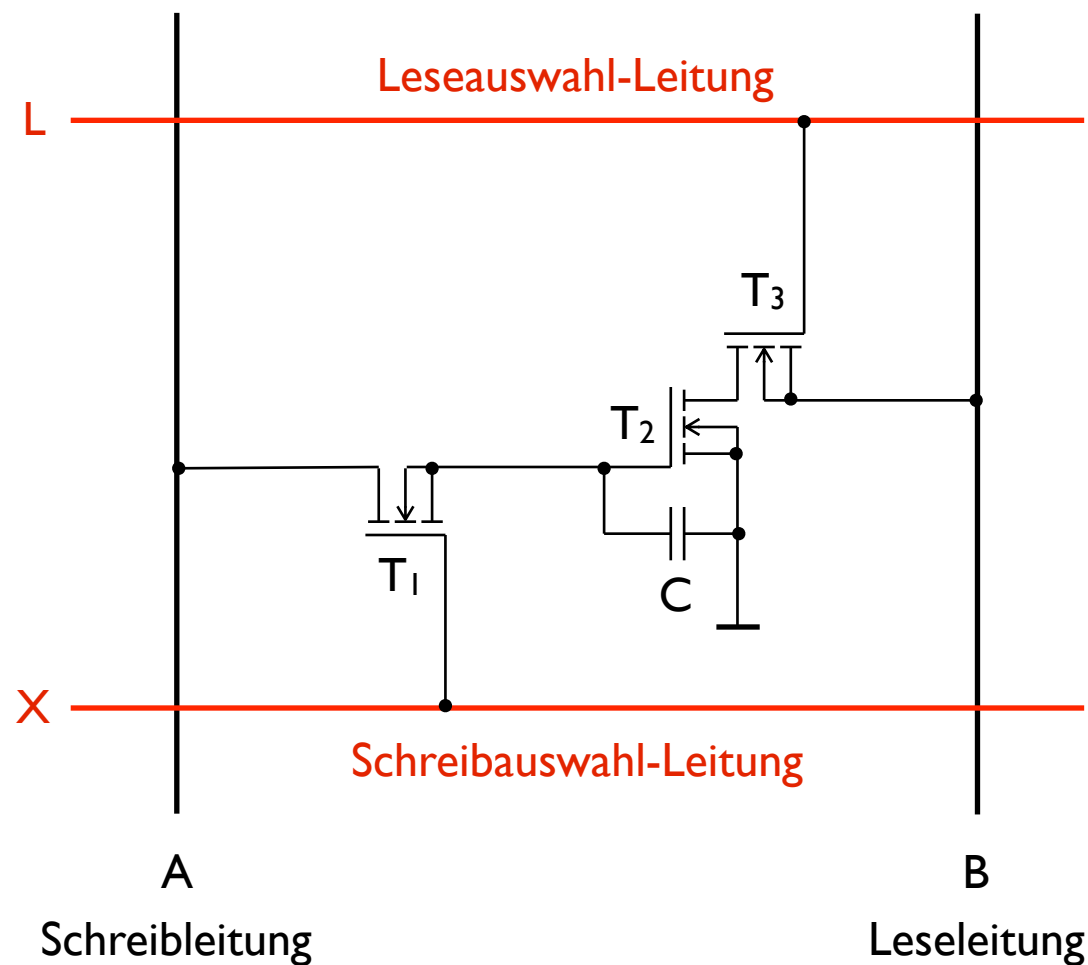
- **DRAM-Speicherelement**
 - ➔ Zusammenschaltung von drei selbstsperrenden MOSFET
 - ➔ Speicherung der Information in der Gate-Substrat-Kapazität C
 - C geladen → Speicherelement besitzt den Wert „1“
 - C entladen → Speicherelement auf „0“ gesetzt



I 3. Register- und Speicherschaltungen

V. dynamischer RAM (DRAM)

- Schreibvorgang

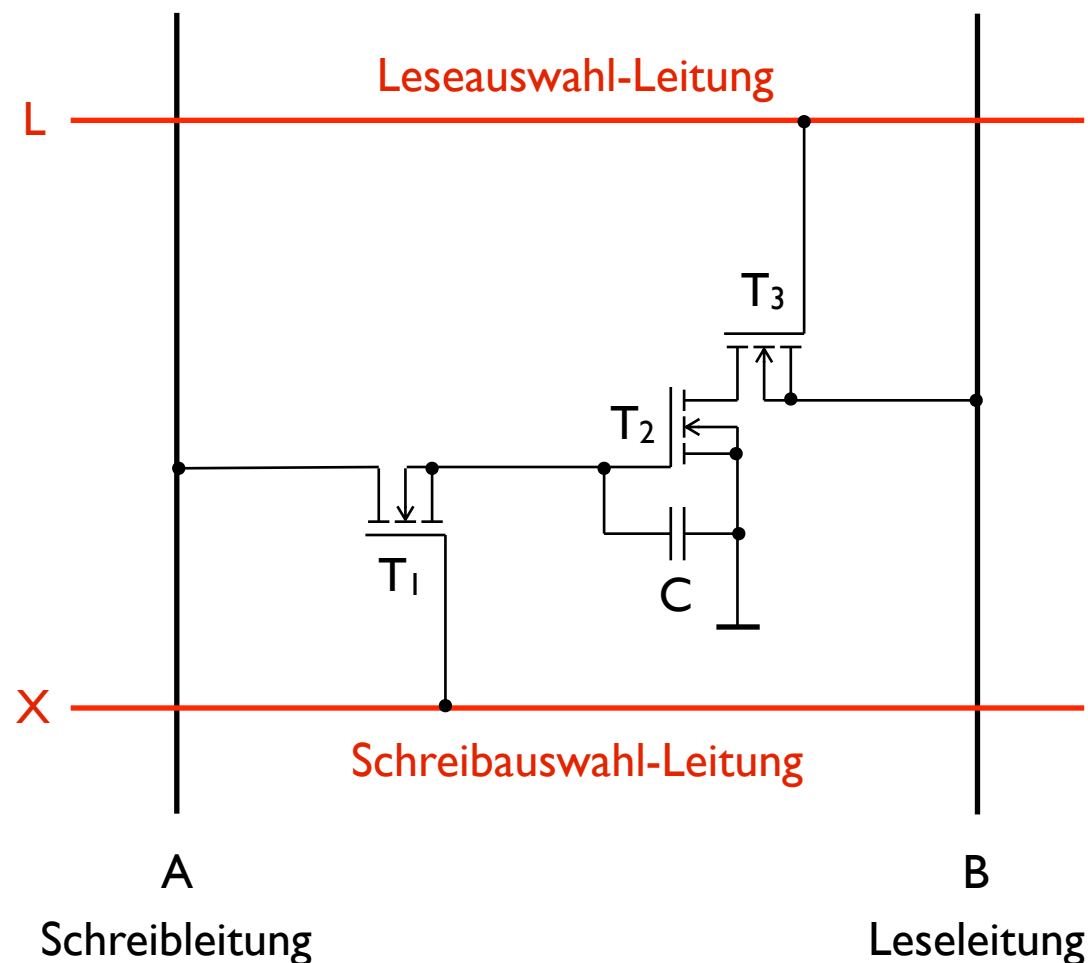


- ➔ Aktivierung der Speicherzelle durch „1“-Signal auf Schreibauswahl-Leitung X
- ➔ Source-Drain-Strecke von T₁ wird niederohmig
- ➔ „1“-Signal an Schreibleitung A führt zum Aufladen von C
 - positive Ladung am Gate von T₂
 - T₂ wird niederohmig
- ➔ Schreibauswahl-Leitung auf „0“
 - Speicherelement nicht mehr aktiv
 - T₁ hochohmig (Sperrzustand)
 - kein Abfließen der positiven Ladung von C möglich
- ➔ Speicherzelle auf „0“ setzen
 - X auf „1“ setzen → T₁ niederohmig
 - A auf „0“ legen → C entlädt sich → Speicherzelle enthält Information „0“
 - T₂ ist hochohmig

I 3. Register- und Speicherschaltungen

V. dynamischer RAM (DRAM)

- Lesevorgang

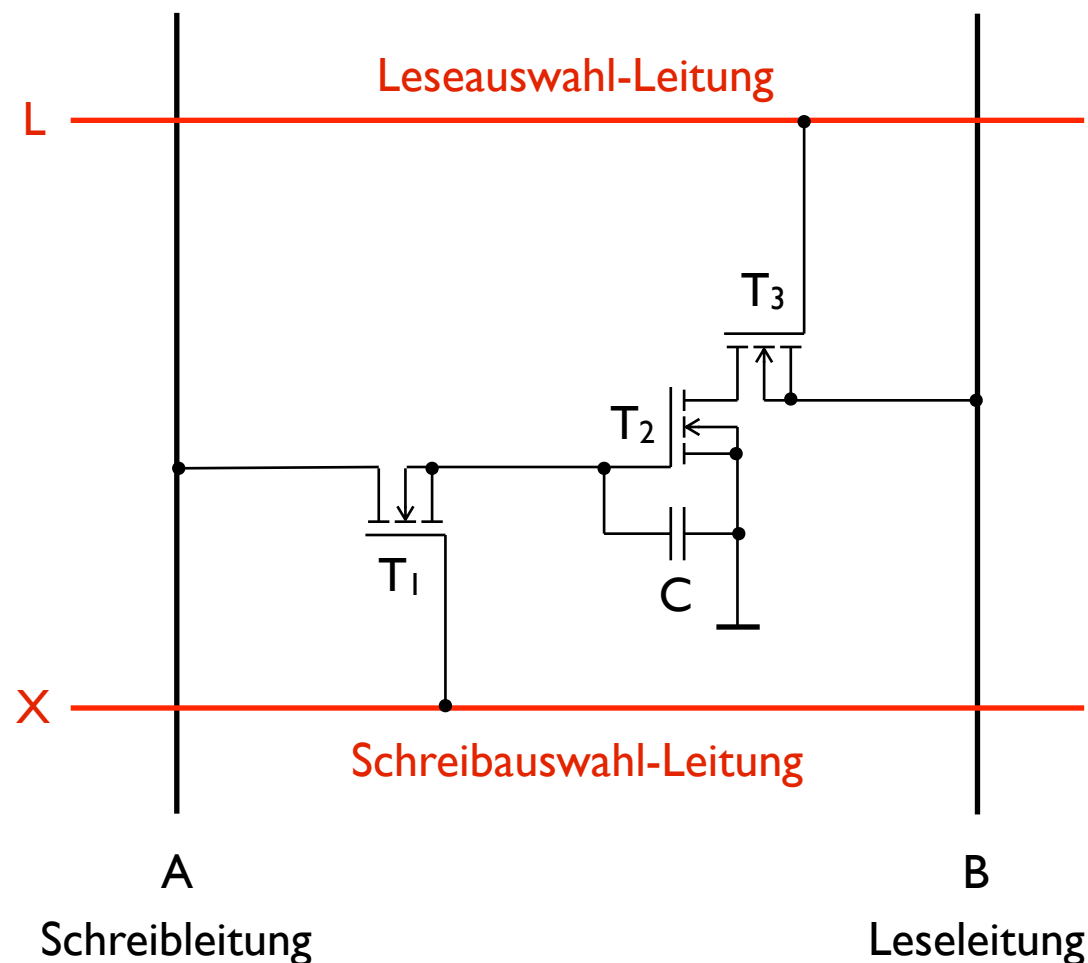


- ➡ „1“-Signal an Leseleitung B
- ➡ Aktivierung der Speicherzelle durch „1“-Signal auf Leseauswahl-Leitung L
- ➡ Source-Drain-Strecke von T₃ wird niederohmig
- ➡ Annahme: Speicherzelle auf „1“
 - T₂ ist niederohmig
 - Stromfluss von B über T₃ und T₂ nach Masse
→ Hinweis auf eingespeicherte „1“
- ➡ Annahme: Speicherzelle auf „0“
 - C ist entladen
 - T₂ hochohmig
 - kein Stromfluss von B über T₃ und T₂ nach Masse
→ Hinweis auf eingespeicherte „0“

13. Register- und Speicherschaltungen

V. dynamischer RAM (DRAM)

- Auffrischvorgang



- ➔ Problem: sehr kleine Kapazität: 0,1 - 1,0 pF
- ➔ $Q = C \cdot U \rightarrow$ nur kleine Ladungsmenge kann gespeichert werden
- ➔ kleine Leckströme bauen Ladung ab
- ➔ Auffrischung der Ladung nötig (alle 2 ms)
 - Taktgenerator und Steuerschaltung benötigt
 - in integrierten Schaltungen enthalten
- ➔ Auffrischen durch Start des Lesevorgangs
 - bei Speicherinhalt „1“ $\rightarrow T_1$ niederohmig schalten $\rightarrow C$ wird geladen
 - bei Speicherinhalt „0“ \rightarrow kein Laden von C

I 3. Register- und Speicherschaltungen

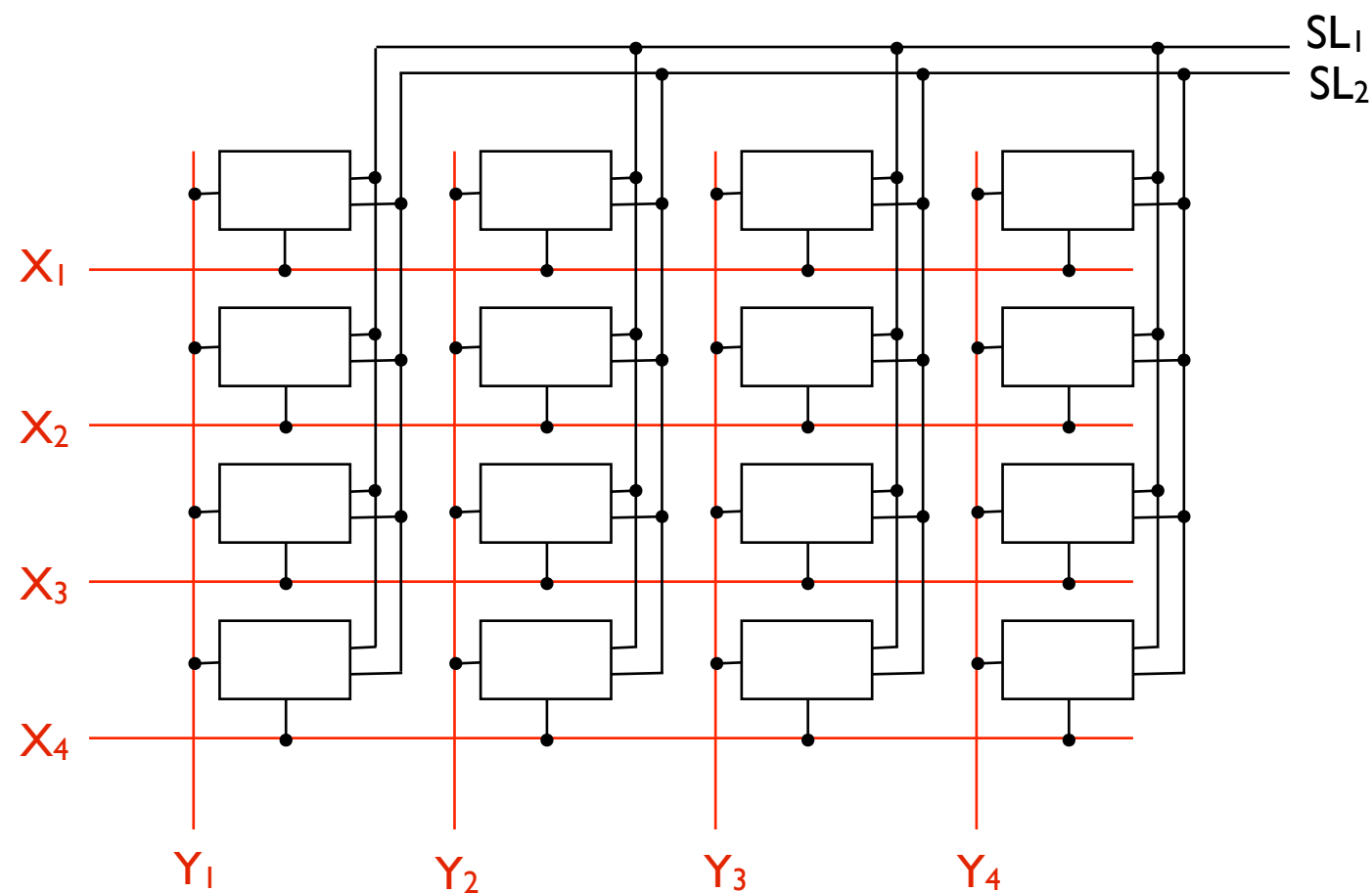
V. Diskussion dynamischer RAM (DRAM)

- ♦ trotz Auffrischvorgang sehr zuverlässiger Speicher
- ♦ große Speicherkapazität pro Chip (hohe Integrationsdichte bei MOS-Schaltungen möglich)
- relativ große Schaltzeiten (Zugriffszeit: 100 - 300 ns)
- Sperrung des Speichers während des Auffrischvorgangs
- mit steigender Temperatur steigt der Leckstrom (beim Überschreiten der maximalen Betriebstemperatur droht Informationsverlust)

I 3. Register- und Speicherschaltungen

VI. Speicheraufbau

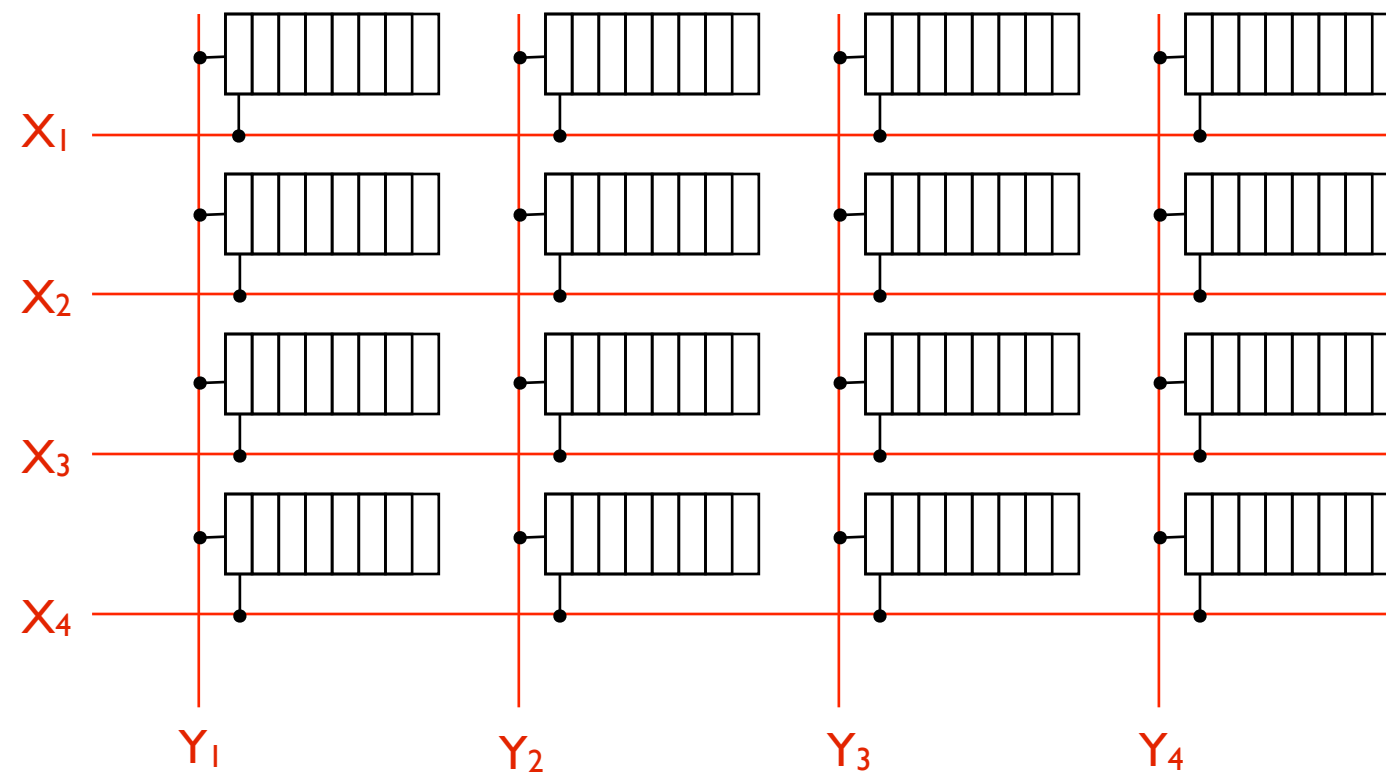
- Unterscheidung von Speichern mit Speicherzellen aus einem oder mehreren Speicherelementen
- bitorganisierter Speicher (Speicherzelle besteht aus einem Speicherelement)
 - ➔ jedes Speicherelement besitzt eigene Adresse
 - ➔ Beispiel: 16×1 -Bit-Speicher



I 3. Register- und Speicherschaltungen

VI. Speicheraufbau

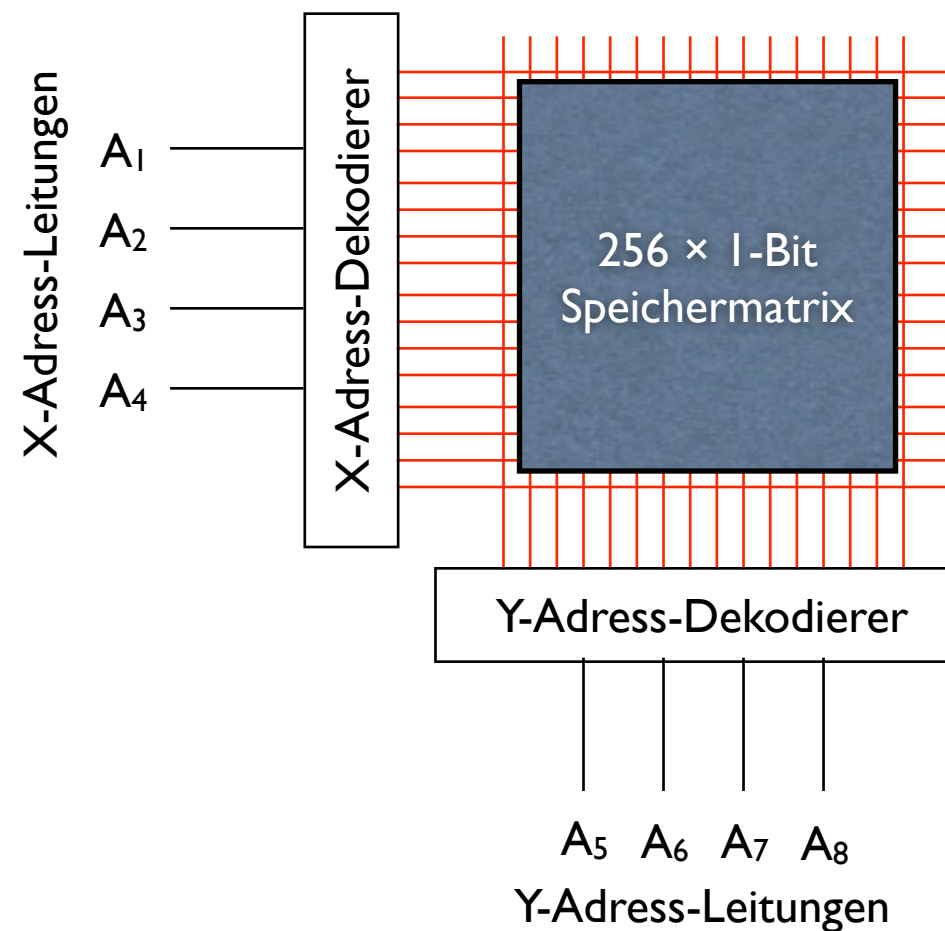
- wortorganisierter Speicher (Speicherzelle besteht aus mehreren Speicherelementen)
 - ➔ jedes x-Bit-Einheit besitzt eigene Adresse
 - ➔ Beispiel: 16 × 8-Bit-Speicher
 - 8 SL₁-Leitungen und 8 SL₂-Leitungen sind nicht eingezeichnet
 - gemeinsames Lesen und Schreiben der 8 Bit



I 3. Register- und Speicherschaltungen

VI. Speicheraufbau

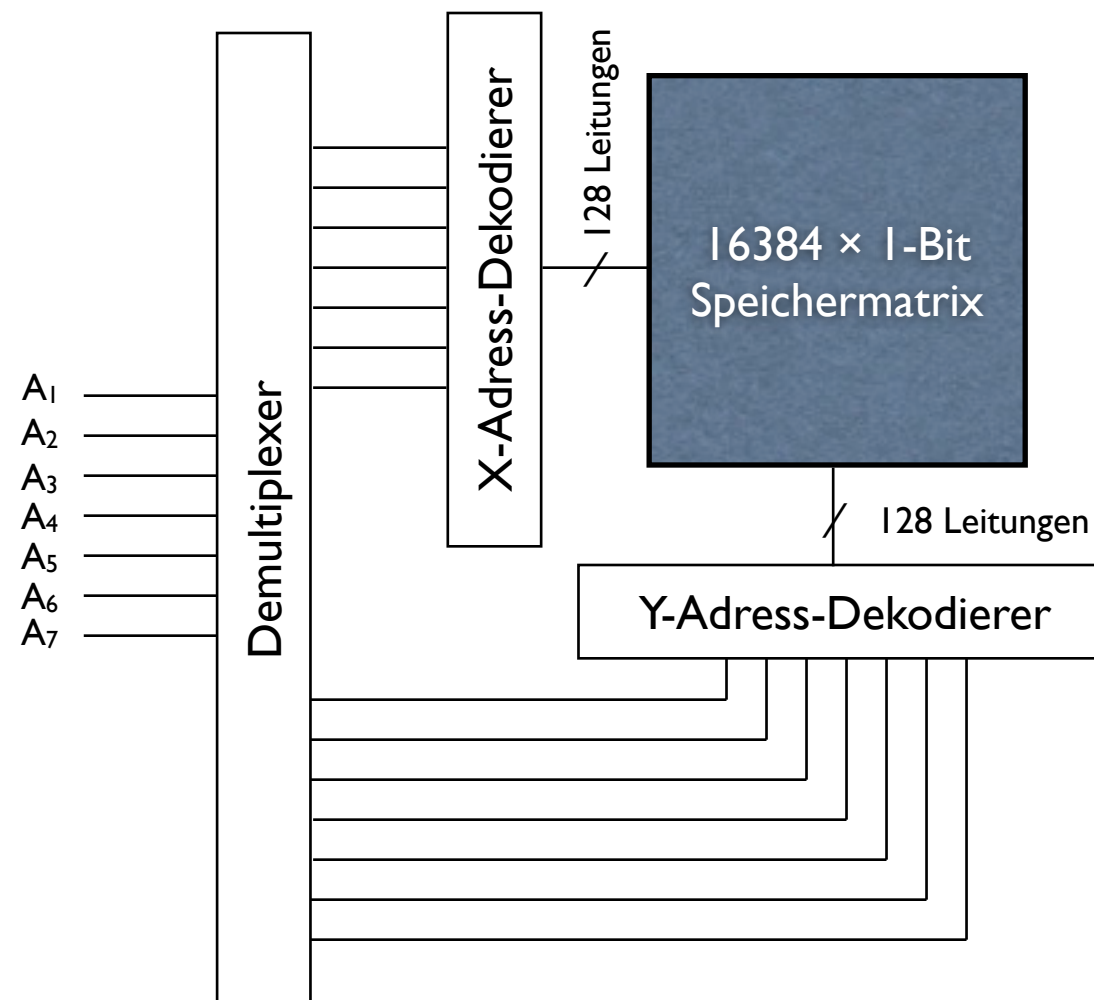
- Speicher mit zunehmender Größe
 - ➔ steigende Anzahl an Koordinatenleitungen
(z.B. 256×1 -Bit-Speicher benötigt 16 X- und 16 Y-Adressen-Leitungen)
 - ➔ Verwendung von Adressdekodierern
 - ➔ Schreib-Lese-Leitungen zur Vereinfachung ausgeblendet



13. Register- und Speicherschaltungen

VI. Speicheraufbau

- Aufbauschema großer Speicher (z.B. 16 kBit × 1-Bit-Speicher)
 - Anwahl von 16384 Bit nötig → 128 X- und 128 Y-Koordinatenleitungen
 - bei Verwendung von Adressdekodierern sind je 7 Steuerleitungen nötig → wiederum hohe Anzahl aus herauszuführenden Leitungen
 - Verwendung von Demultiplexern



I 3. Register- und Speicherschaltungen

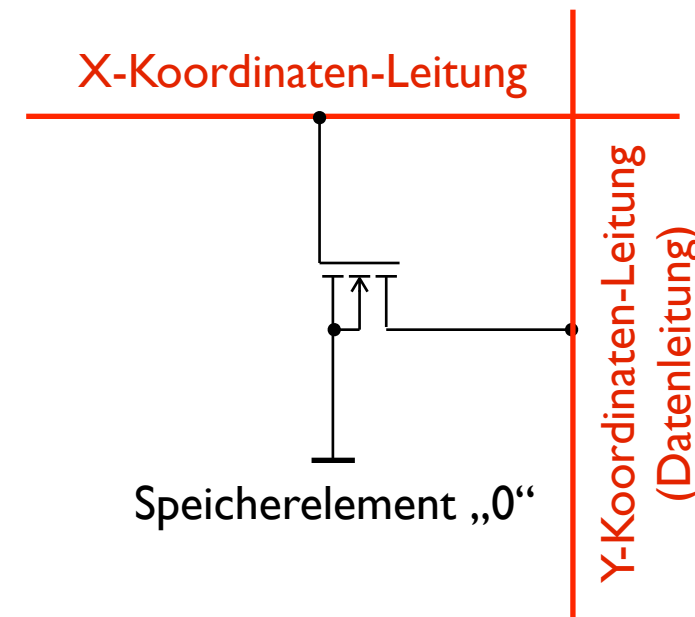
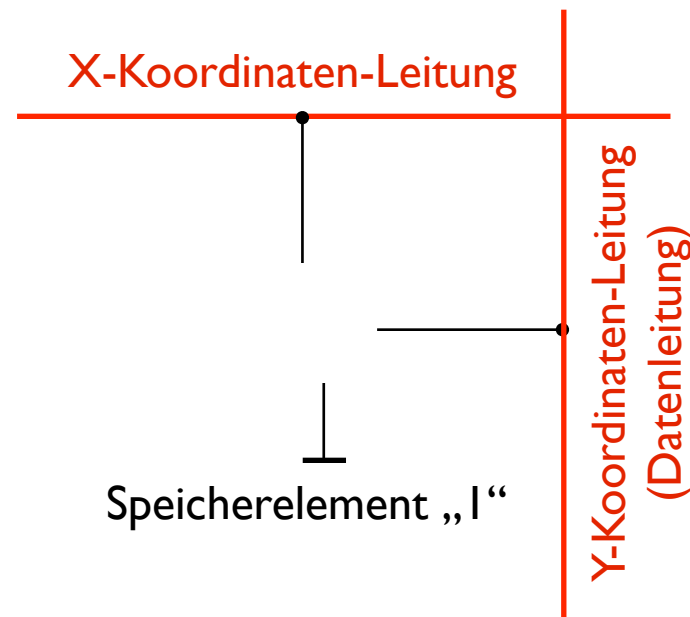
VII. Speicherkenngößen

- Speicherkapazität
 - ➔ Anzahl der speicherbaren Bits
- Speicherorganisation
 - ➔ Speicherkapazität einer Speicherzelle sowie Anwahlmöglichkeit
- Zugriffszeit
 - ➔ Zeitintervall von der Adressierung bis zur Verfügbarkeit der Information am Ausgang
- Zykluszeit
 - ➔ kürzeste Zeit zwischen zwei aufeinander folgenden Schreib-Lese-Vorgängen
- Leistungsbedarf
 - ➔ Angabe des Gesamtleistungsbedarfs bei Betrieb und im Ruhezustand
- Elektrische Betriebsbedingungen
 - ➔ Angabe der benötigten Versorgungsspannungen, der Signalpegel, der Toleranzbereiche und der elektrischen Grenzwerte
- Arbeitstemperaturbereich
 - ➔ Temperaturintervall in dem der Speicher unter vorgegebenen Betriebsbedingungen sicher arbeitet

I 3. Register- und Speicherschaltungen

VII. Read Only Memory - ROM (Festwertspeicher)

- Nur-Lese-Speicher
- im Speicher enthaltene Information ist nicht löscherbar und nicht änderbar
- zwei Arten von Speicherelementen



- Lesen eines Speicherelements
 - ➔ Aktivierung des Speicherelements: Koordinaten-Leitungen auf „1“ setzen
 - ➔ Datenleitung bleibt auf „1“ bei fehlendem Transistor
 - ➔ Datenleitung wird auf „0“ (Masse) bei vorhandenem Transistor gezogen

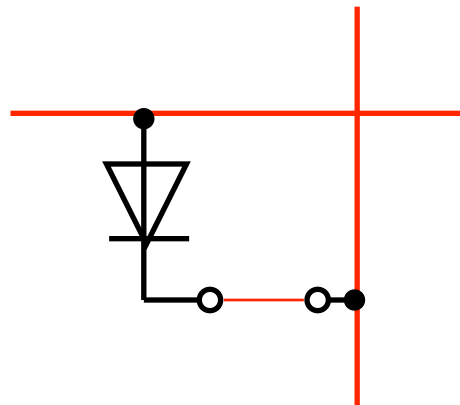
VII. Read Only Memory - ROM (Festwertspeicher)

- Speicheraufbau und -organisation analog zu RAM-Speicher
 - ➔ Speichermatrix bestehend aus Zeilen und Spalten
 - ➔ Ansteuerung der X- und Y-Koordinatenleitungen mit Adress-Dekodierern und ggf. Demultiplexern bei großen Speichern
- Maskenprogrammierbare Festwertspeicher
 - ➔ Information („0“ oder „1“) wird vor der Herstellung definiert
 - ➔ Maskenprozesse erlauben definiertes Platzieren von Transistoren
- spezifische Maske ist ein Kostentreiber
 - ➔ Wirtschaftlichkeit nur bei hohen Stückzahlen gegeben

I 3. Register- und Speicherschaltungen

VIII. Programmable Read Only Memory - PROM

- Programmierbare Festwertspeicher
- eigene Eingabe der Information in den Festwertspeicher möglich
- kleine Stückzahlen, bzw. Einzelstücke
- Programmierung durch gezieltes Durchbrennen einzelner Transistoren möglich (Herstellung einer Unterbrechung)
- Alternativ: Aufbau einer Matrix mit Dioden (Dioden-PROM)

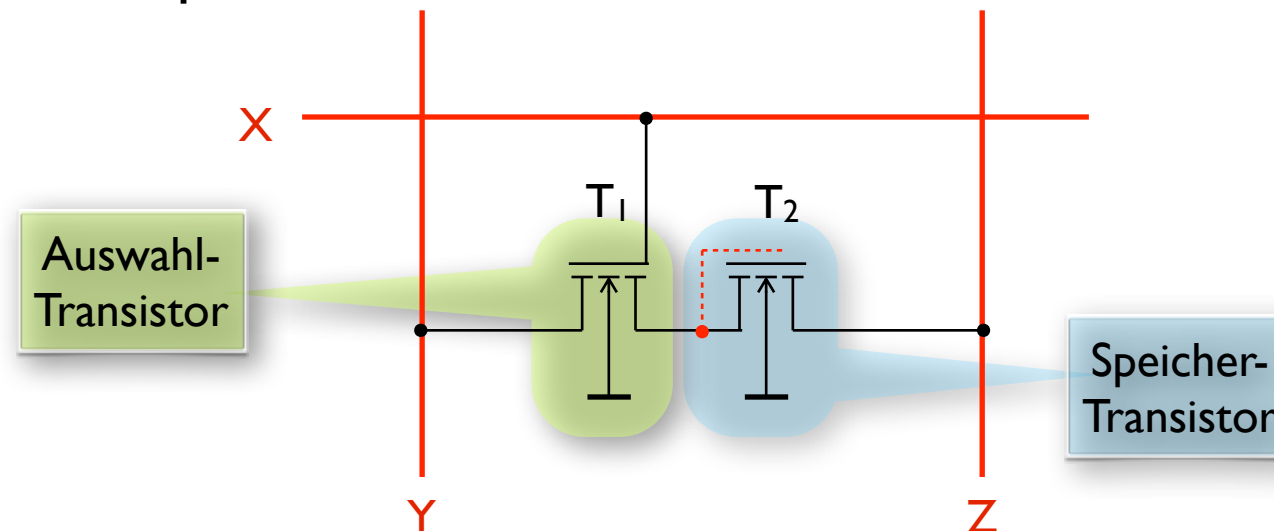


- ➔ dünne Leitung aus Chrom-Nickel-Legierung
- ➔ Durchbrennen der Leitung bei Erhöhung des Stroms über Schwellenwert
- ➔ nicht-reversibler Vorgang
- ➔ spezielles Programmiergerät erforderlich

13. Register- und Speicherschaltungen

IX. EPROM und REEPROM (löschrare und programmierbar)

- Löschen der Information durch UV-Licht möglich
 - ➔ Erasable Programmable Read Only Memory - EPROM
 - ➔ Reprogrammable Read Only Memory - REEPROM
- Aufbau eines Speicherelements



- ➔ T₂ besitzt ein Floating Gate (kein Gateanschluss, lediglich hochisolierendes Material auf dem Gate)
- ➔ Z-Leitung liegt auf Masse (0V)
- ➔ +5V an X- und Y-Koordinaten-Leitungen → T₁ schaltet durch

gelöschter Zustand: Gate ohne Ladung → T₂ gesperrt

- ➔ da T₂ gesperrt, kann Y nicht auf Masse gezogen werden (Y bleibt auf „1“)

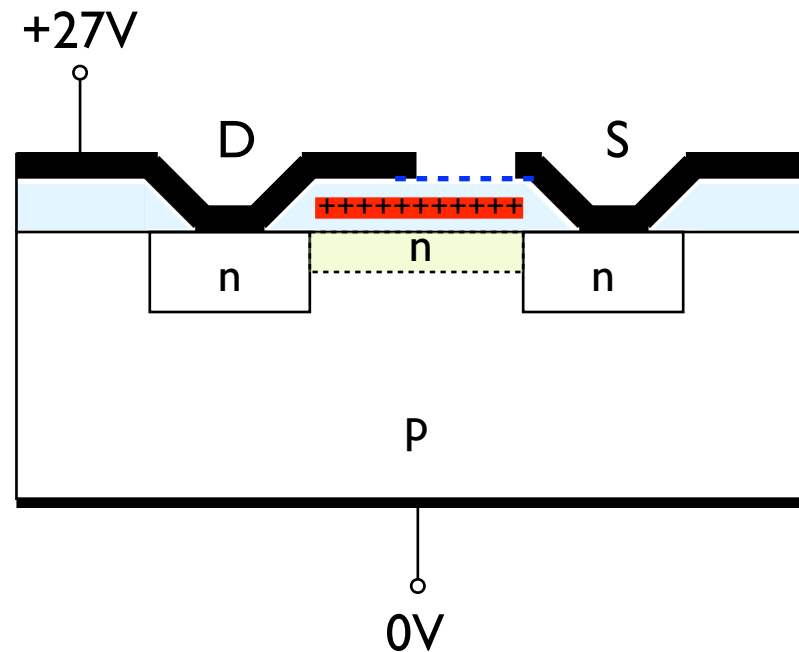
geladener Zustand: Gate positiv geladen → T₂ leitend

- ➔ da T₂ leitend, wird Y auf Masse (Zustand „0“) gezogen

13. Register- und Speicherschaltungen

IX. EPROM und REEPROM (löschar und programmierbar)

- Aufbau eines Floating-Gate Avalanche-Injection MOSFET (FAMOS-Transistor)



- Aufladen eines FAMOS-Transistors
 - Dünne Isolierschichten sowie dünnes Floating Gate
 - Anlegen eines hohen Potentials an Drain gegenüber Substrat (Programmierspannung)
 - hohes E-Feld
 - Drain „saugt“ Elektronen vom Floating Gate ab (Tunneleffekte)
 - „positive Ladung“ auf Floating Gate zieht Elektronen aus dem Substrat an
 - n-leitende Brücke entsteht

I 3. Register- und Speicherschaltungen

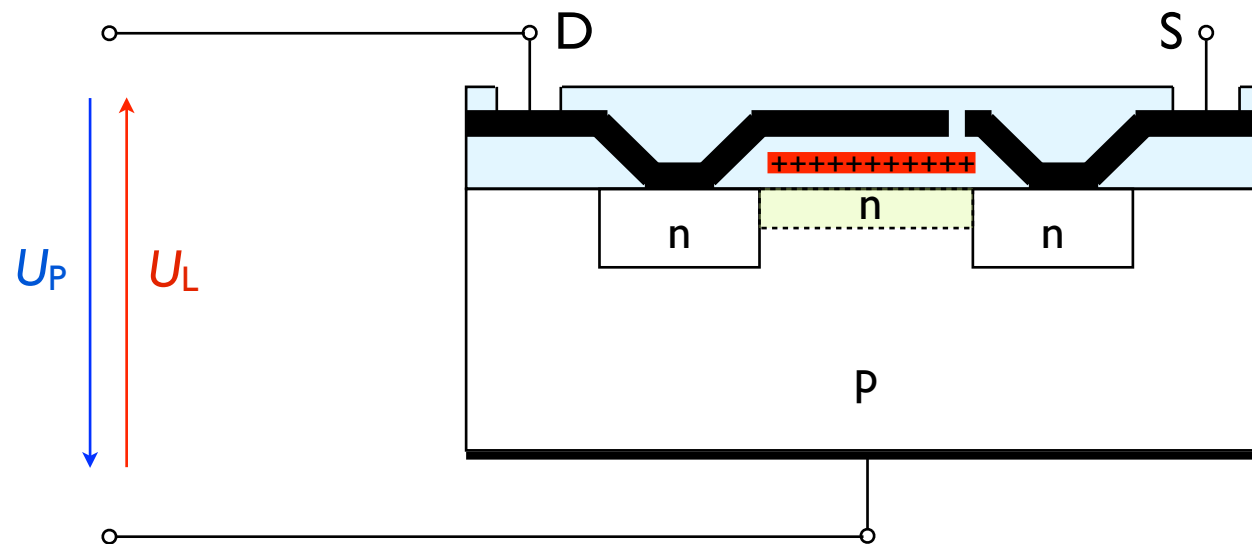
IX. EPROM und REEPROM (löschar und programmierbar)

- **Programmiervorgang**
 - ➔ Programmierung der einzelnen Speicherzellen durch sequentielle Anwähl (+5V an X- und Y-Koordinatenleitungen) der Zellen die Speicherinhalt „0“ besitzen sollen
 - ➔ Anlegen des Potentials 27V an Drain → Aufbringen positiver Ladung auf Floating Gate
 - ➔ Ladungserhaltung auf Floating Gate: 1 - 100 Jahre (Herstellerangaben)
- **Löschvorgang**
 - ➔ Löschen der Information durch Einstrahlung starken UV-Lichts auf Fenster des (R)EPROMS
 - Ionisation des Isolationsmaterials (wird schwach leitfähig)
 - Ladungsabbau des Floating Gate
 - gesamte Information des Festwertspeichers wird gelöscht
 - ➔ Vor Neuprogrammierung ist Abkühlen des Bausteins nötig (+ Abklingen der Ionisierung)
 - ➔ Unabsichtliches Löschen durch Lichtkontakt möglich
 - Sonnenlicht: Chip nach ca. 3 Tagen gelöscht
 - Leuchtstofflampe: Löschen der Information nach ca. 3 Wochen
 - Abhilfemaßnahme: Abkleben des Fensters mit lichtundurchlässigem Klebeband

13. Register- und Speicherschaltungen

X. EEPROM und EAROM (löschar und programmierbar)

- Aufbau der Speicherzelle ähnlich wie EPROM und REEPROM
- Unterschied: elektrische Löschung der Information
 - ➔ Löschen der gesamten Information des Bausteins: Electrically Erasable (elektrisch löschbares) ROM
 - ➔ Löschen eines einzelnen Bits: Electrically Alterable (elektrisch umprogrammierbares) ROM
- Aufbau des Speichertransistors (FAMOS-Transistor)



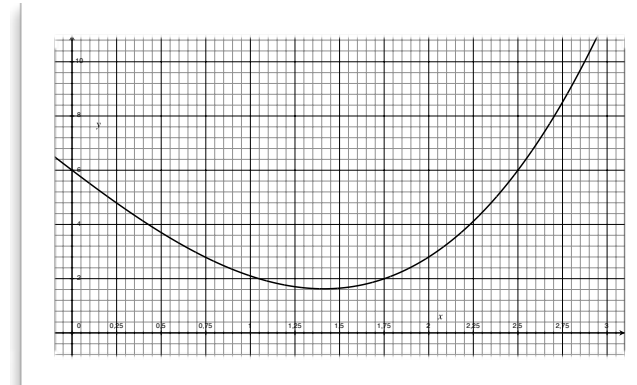
- Programmierung: Anlegen der Drain-Substrat-Spannung $U_P \approx 40\text{ V}$: Elektronenwanderung vom Floating Gate zum Drain-Anschluss
- Löschen: Umpolung der Drain-Substrat-Spannung:
 - ➔ Erzeugung eines entgegengesetzt gerichteten starken E-Feldes
 - ➔ Elektronen wandern von Drain auf Floating Gate → negative Aufladung
 - ➔ Brücke zwischen Drain und Source verschwindet

I 4. D/A-, A/D-Wandler

I. D/A-Wandler (Digital-Analog-Umsetzer)

- Prinzip: Umwandlung einer Tabelle in eine Kurve

t	1	2	3	4	5
f(t)	22	15	31	47	91

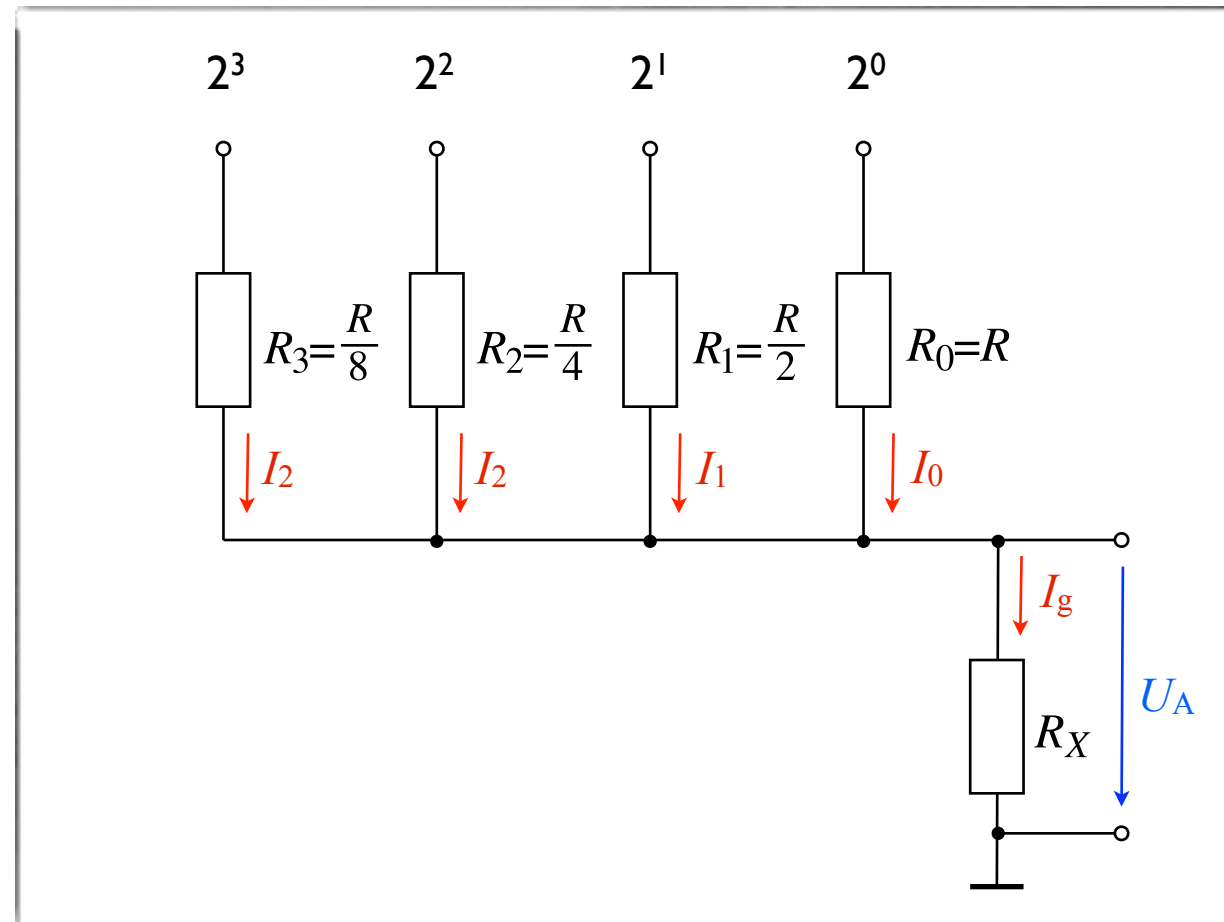


- Wandlung von bestimmten binären (bewertete) Codes in Analog-Signale möglich
 - ➔ Dual-Kode
 - ➔ BCD-Kode
 - ➔ nicht möglich: GRAY-Kode (unbewerteter Kode)
- unbewertete Codes vor D/A-Umsetzung in bewerteten Kode umwandeln
- Hamming-Kode: Redundanzstellen (Kontroll-Bits) eliminieren
- umgesetztes Analog-Signal:
 - ➔ diskrete Signalwerte (gestuftes Signal → Glättung durch Siebglieder)
 - ➔ nur bestimmte Anzahl an möglichen Amplitudenwerten (2^n)

I 4. D/A-, A/D-Wandler

I. D/A-Wandler (Digital-Analog-Umsetzer)

- Prinzipschaltung (4-Bit) mit gestuften Widerständen

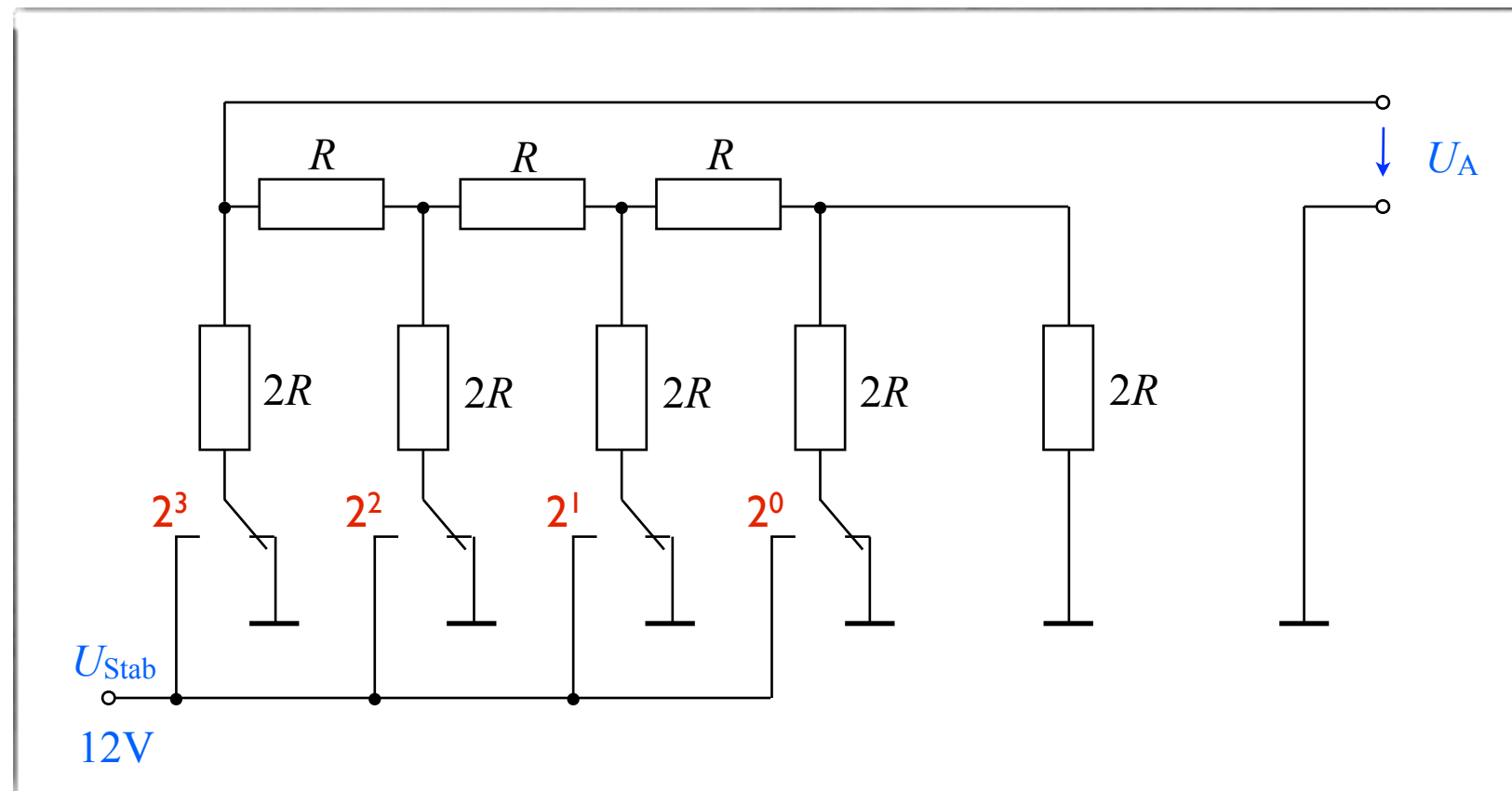


- Gleichung für die Wahl der Widerstände: $R_n = \frac{R}{2^n}$
- Widerstandswert R frei wählbar
- Problem: Spannungsschwankungen der Eingangsspannungen

I 4. D/A-, A/D-Wandler

I. D/A-Wandler (Digital-Analog-Umsetzer)

- Prinzipschaltung: R/2R-DA-Wandler (Kettenleiter)



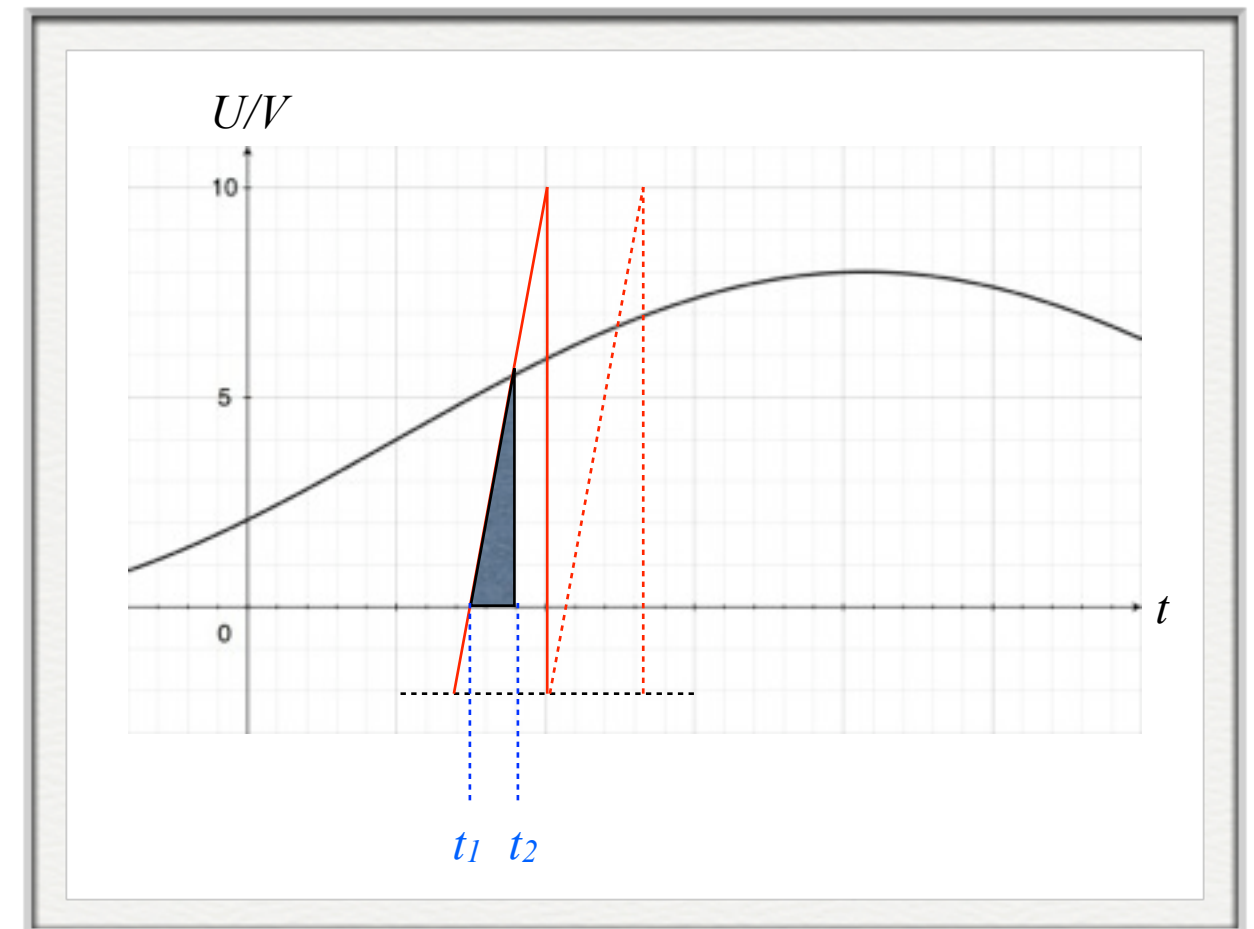
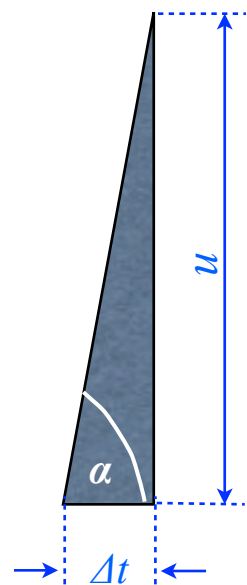
- Lediglich zwei Widerstandswerte R und $2R$ nötig
- Gleichung für Intervall der Ausgangsspannung: $\Delta U_A = \frac{U_{Stab}}{2^n}$

II. A/D-Wandler (Analog-Digital-Umsetzer)

- Abtasten des analogen Signals in bestimmten zeitlichen Intervallen liefert ein Digitalsignal
- Darstellung des digitalen Signals in verschiedenen Zahlensystemen möglich (duales Zahlensystem, BCD-Kode)
- Anzahl der verfügbaren Bits definiert das Auflösungsvermögen des A/D-Wandlers
- Genauigkeit eines A/D-Wandlers definiert sich durch den prozentualen Fehler des Ergebnisses oder des Höchstwerts.
- Abtastfrequenz des Analogsignals mindestens doppelt so groß wie die höchste zu wandelnde Frequenz
- Herstellung von A/D-Wandlern als integrierte Schaltungen (dominant in CMOS-Technologie)
- festgelegter Bereich der kleinsten und größten wandelbaren Spannung

II. A/D-Wandler (Analog-Digital-Umsetzer)

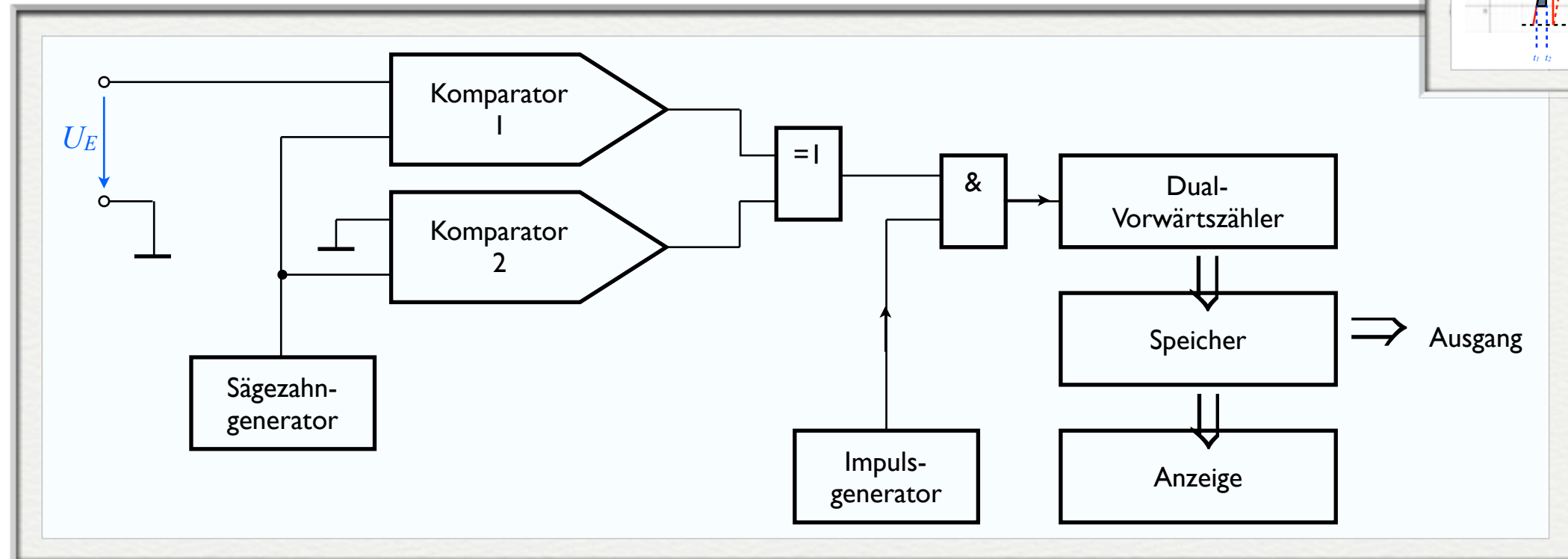
- Prinzip: Sägezahnverfahren
 - ➔ Abtasten des Analogsignals mit einer Sägezahnspannung
 - ➔ Arbeitsweise des A/D-Wandlers nach dem Sägezahnverfahren
 - Beginn der Flanke im negativen Bereich
 - Start eines Zählers beim Überschreiten der 0V-Linie
 - Stopp des Zählers beim Erreichen der analogen Spannung
 - Zählergebnis liefert Zeitintervall Δt
 $\rightarrow u = \Delta t \cdot \tan \alpha$
 - Höchstwert der Sägezahnspannung definiert größte abtastbare Spannung (hier: 10V)



14. D/A-, A/D-Wandler

II. A/D-Wandler (Analog-Digital-Umsetzer)

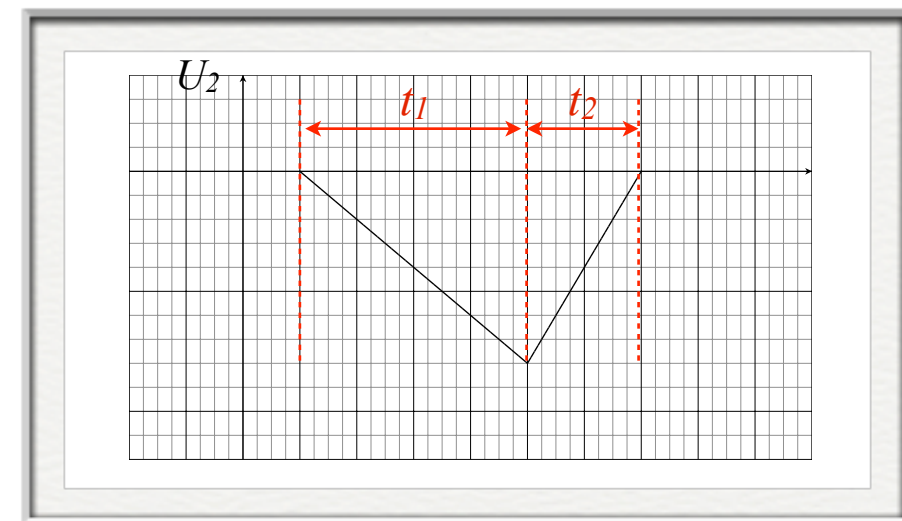
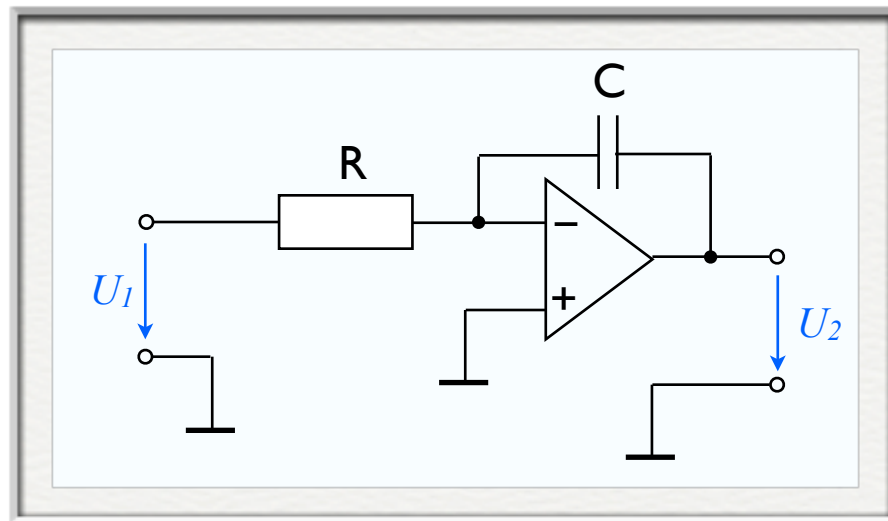
- Prinzip: Sägezahnverfahren



- Sägezahnspannung auf 2 Komparatoren geben
- Funktionsweise der Komparatoren: „1“ am Ausgang, wenn beide Eingangsspannungen gleich groß
- neg. Sägezahnspannung → „0“ am Ausgang von Komparator 2 (K_2)
- Zeitpunkt t_1 (Erreichen der 0V-Linie): $K_2 = „1“$ (K_1 immer noch auf „0“) → Ausgang XOR = „1“
- Freigabe der Generatorimpulse zum Zählen
- Sägezahnspannung erreicht die Eingangsspannung → $K_1 = „1“$ → Ausgang XOR = „0“ → Zählstopp
- Abspeichern des Zählerstands und Ausgabe

II. A/D-Wandler (Analog-Digital-Umsetzer)

- Prinzip: Dual-Slope-Verfahren
 - ➔ Zweischrittverfahren: Verwendung von zwei unterschiedlichen Flanken
 - ➔ Verwendung eines Integrators



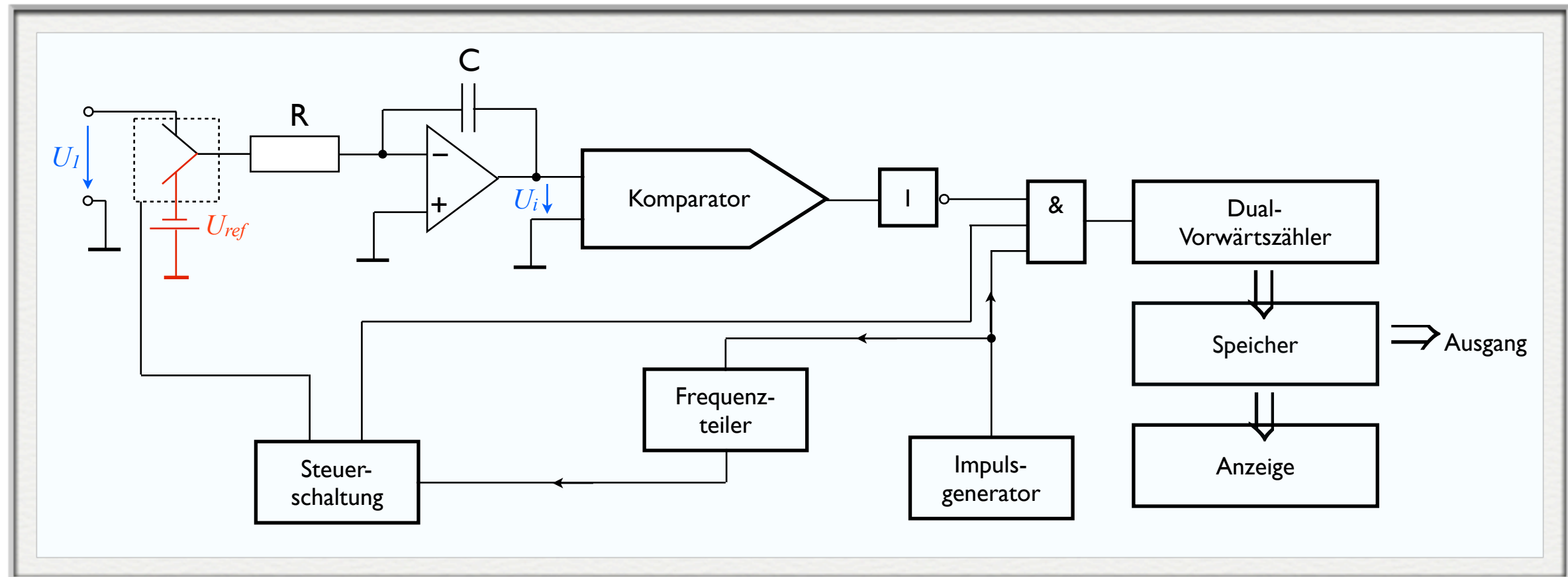
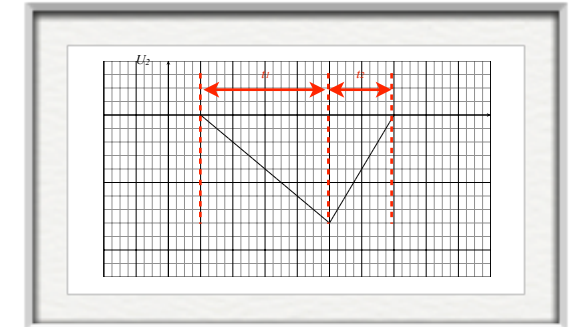
- ➔ Schritt 1: Integration einer positiven Analogspannung in einem festgelegten Zeitintervall t_1 (Laden des Kondensators C)
- ➔ Schritt 2:
 - Anlegen einer negativen Referenzspannung an den Eingang
 - Messung des Zeitintervalls t_2 bis zur vollständigen Entladung von C
 - Zeitintervall proportional zur angelegten Spannung

I 4. D/A-, A/D-Wandler

II. A/D-Wandler (Analog-Digital-Umsetzer)

- Prinzip: Dual-Slope-Verfahren

➔ Aufbau der Schaltung



➔ Steuerschaltung bringt Schalter in obere Stellung (Dauer: Intervall t_1) → C wird geladen

➔ Switch auf untere Stellung

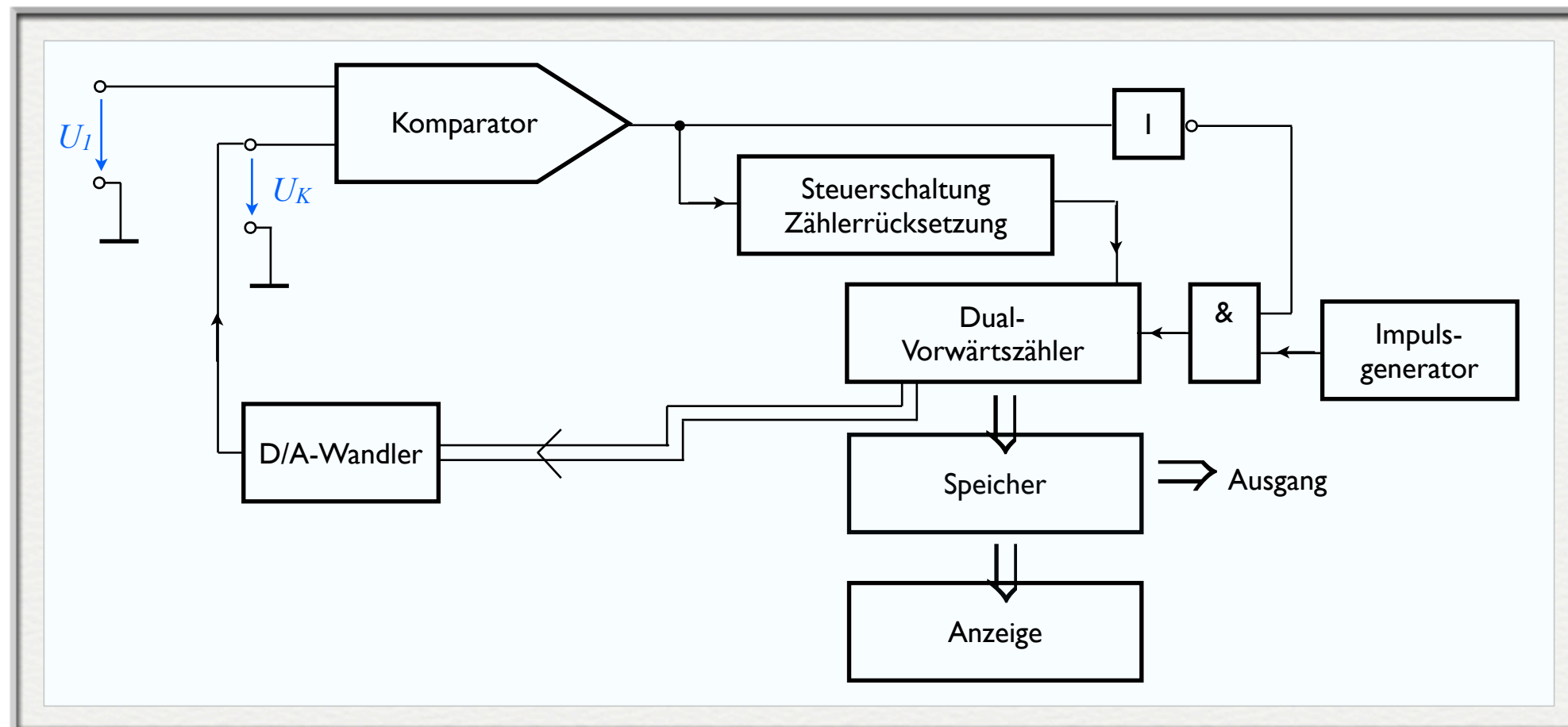
- Referenzspannung liegt an Integrator
- Freigabe des UND-Glieds über Steuerschaltung → Zähler läuft los
- C entladen → Komparator liefert „I“ am Ausgang → Zähler wird gestoppt (Intervall t_2) → $\frac{U_1}{U_{ref}} = \frac{t_2}{t_1}$

I 4. D/A-, A/D-Wandler

II. A/D-Wandler (Analog-Digital-Umsetzer)

- Kompensationsverfahren

➔ Aufbau der Schaltung

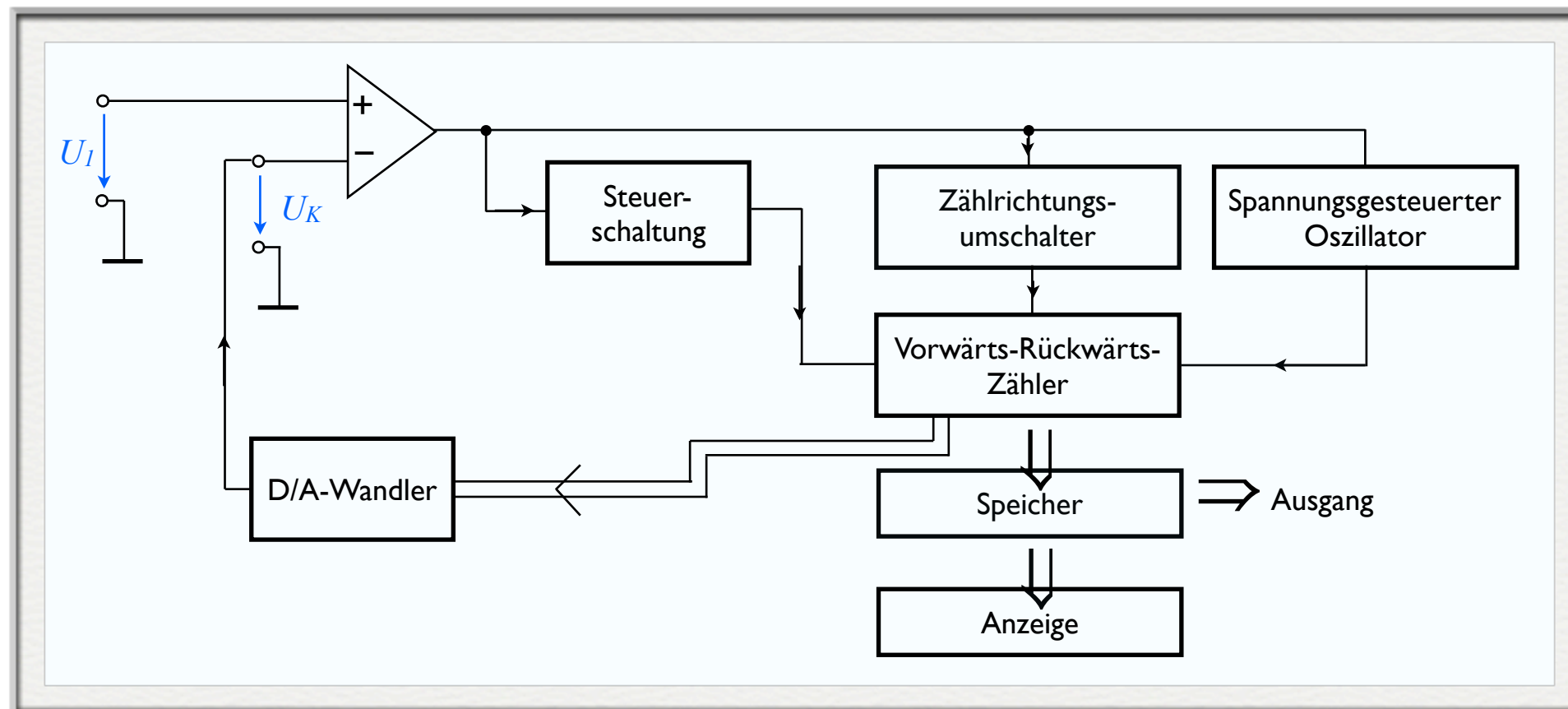


- ➔ Nullung des Zählers bei Start eines Umsetzvorgangs durch Steuerschaltung $\rightarrow U_K = „0“ \wedge K = „0“$
- ➔ Dual-Vorwärtszähler wird mit Signalen aus Impulsgenerator gestartet \rightarrow D/A-Wandler liefert ansteigendes Analogsignal
- ➔ bei $U_K = U_I \rightarrow K = „1“ \rightarrow$ Stopp des Zählvorgangs und Ablegen des Zählergebnisses im Speicher \rightarrow Nullung des Zählers

I 4. D/A-, A/D-Wandler

II. A/D-Wandler (Analog-Digital-Umsetzer)

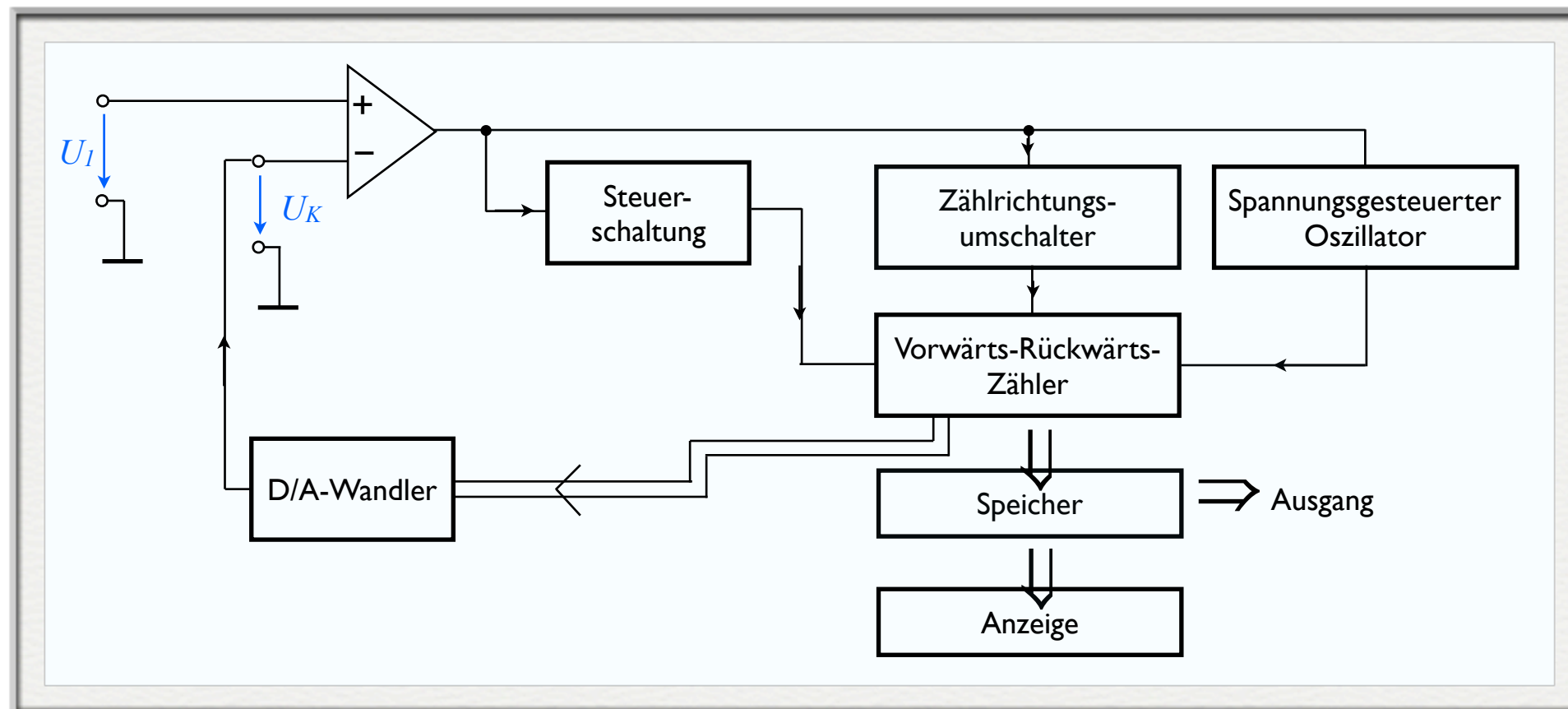
- Kompensationsverfahren
 - ➔ Modifikation der Schaltung (I/2)



- ➔ Komparator durch Differenzenverstärker ersetzt: positive Ausgangsspannung falls $U_K < U_I$ (Betrag der Ausgangsspannung beeinflusst Geschwindigkeit des spannungsgesteuerten Oszillators: je höher die Spannung, desto höher die Frequenz der Ausgangspulse)
- ➔ Vorwärts-Rückwärts-Zähler zählt in Vorwärtsrichtung → analoge Spannung U_K steigt
- ➔ Stopp des Zählers bei $U_K = U_I$ → Ablegen des Ergebnisses im Speicher

II. A/D-Wandler (Analog-Digital-Umsetzer)

- Kompensationsverfahren
 - ➔ Modifikation der Schaltung (2/2)

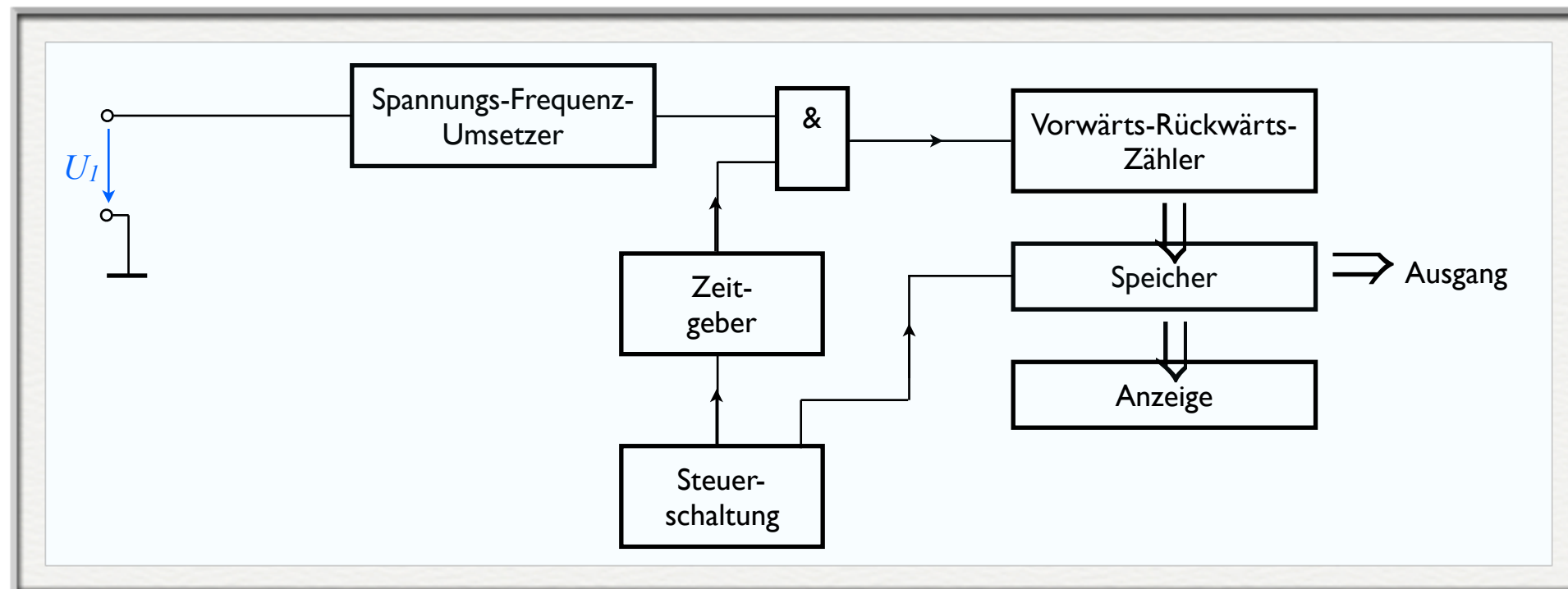


- ➔ Änderung von $U_I \rightarrow$ Ausgang des Differenzenverstärkers liefert pos. oder neg. Signal (definiert Laufrichtung des Zählers)
- ➔ \rightarrow Zähler läuft nach, bis $U_K = U_I \rightarrow$ Stopp des Zählers \rightarrow Speichern des Werts
- ➔ Vorteil: Zähler immer in der Nähe des analogen Signals (wenige Zählschritte nötig \rightarrow hohe Geschwindigkeit)

II. A/D-Wandler (Analog-Digital-Umsetzer)

- Spannungs-Frequenz-Verfahren

➔ Aufbau der Schaltung

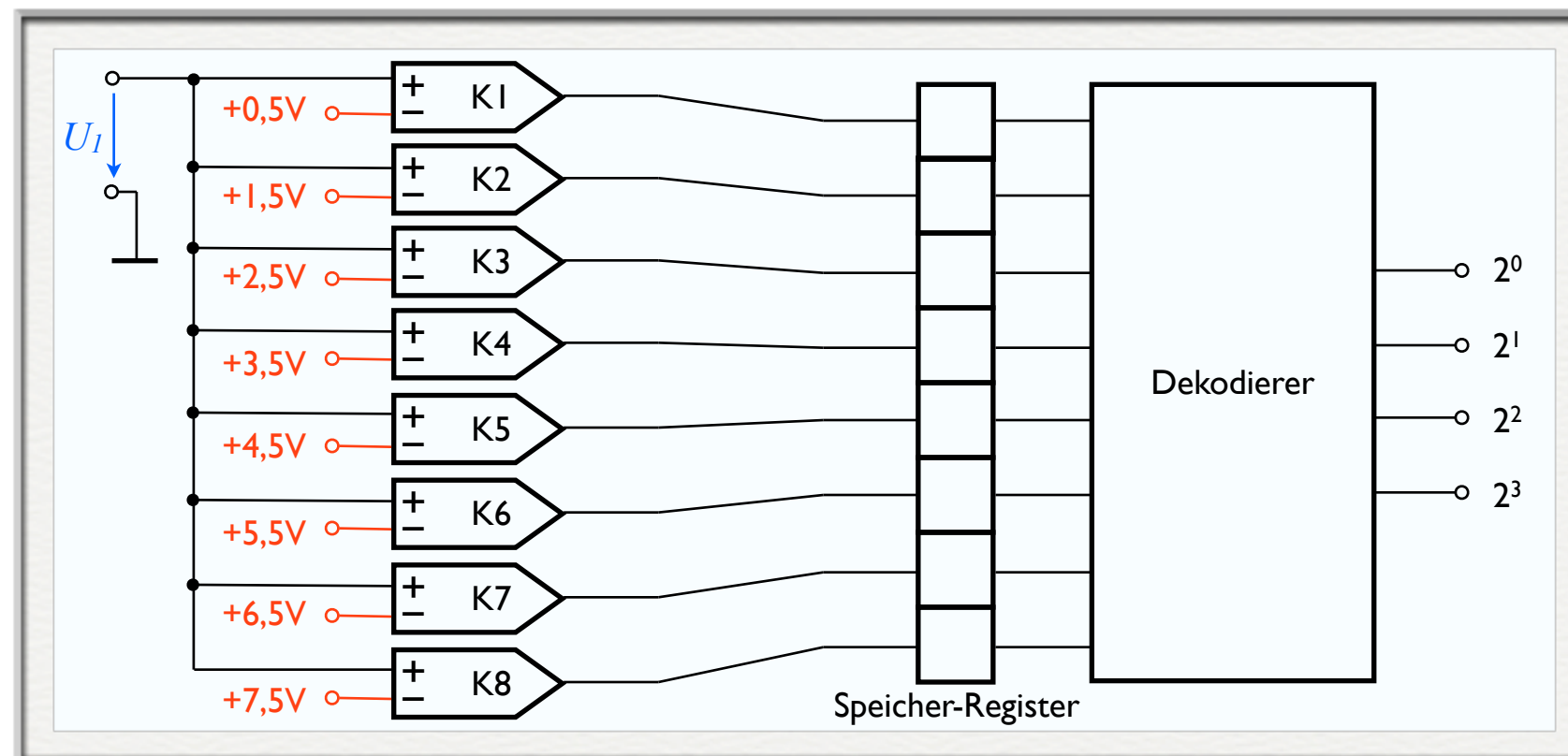


- ➔ Spannungs-Frequenz-Umsetzer liefert linearen Zusammenhang zwischen U und f
- ➔ Steuerschaltung aktiviert Wandlung → Zeitgeber liefert „1“ an ersten Eingang des UND-Glieds
- ➔ Frequenzpulse aktivieren Zählvorgang
- ➔ nach Ablauf einer festgelegten Zeit: Zeitgeber liefert „0“ an Eingang des UND-Glieds → Zähler wird gestoppt → Ergebnis in Speicher ablegen

II. A/D-Wandler (Analog-Digital-Umsetzer)

- Direktverfahren

- ➔ Aufbau der Schaltung



- ➔ Komparatoren liefern „1“, falls Eingang größer als Referenzspannung
- ➔ n Komparatoren ermöglichen n verschiedene Spannungsstufen (hier: $n = 8$)
- ➔ sehr schneller Umsetzer (Schaltzeit der Komparatoren: ca. 40 - 50 ns)
- ➔ Genauigkeit des Wandlers abhängig von Genauigkeit der Referenzspannung und von Schalttoleranzen der Komparatoren

15. Rechenschaltungen

I. Einführung

- Ziel der Rechenschaltungen: Durchführung einer bestimmten Rechenoperation mit den Eingangsvariablen
- Rechenschaltung jeweils nur für einen Kode (Zahlensystem) geeignet
- Arten von Rechenschaltungen
 - ➔ Halbaddierer
 - ➔ Volladdierer
 - ➔ Paralleladdierschaltung
 - ➔ Serielle Addierschaltung
 - ➔ Subtrahierschaltung
 - ➔ Addier-Subtrahier-Werk
 - ➔ Multiplikationsschaltung

15. Rechenschaltungen

II. Der Halbaddierer (HA)

- ermöglicht Addition zweier Dualziffern
- Rechenregeln des Halbaddierers

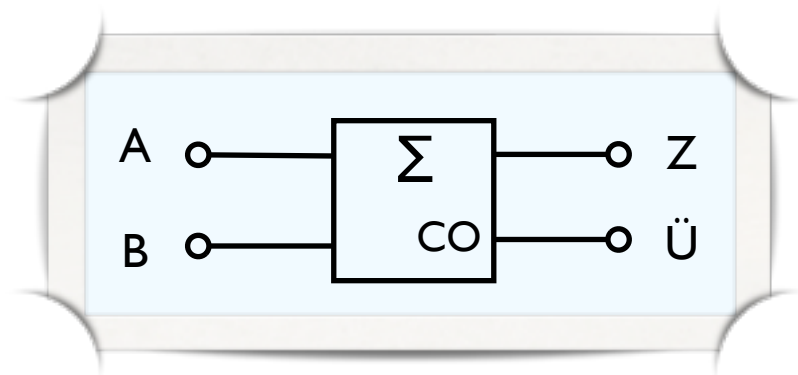
→ $0 + 0 = 0$

→ $0 + 1 = 1$

→ $1 + 0 = 1$

→ $1 + 1 = 10$

- Schaltzeichen:

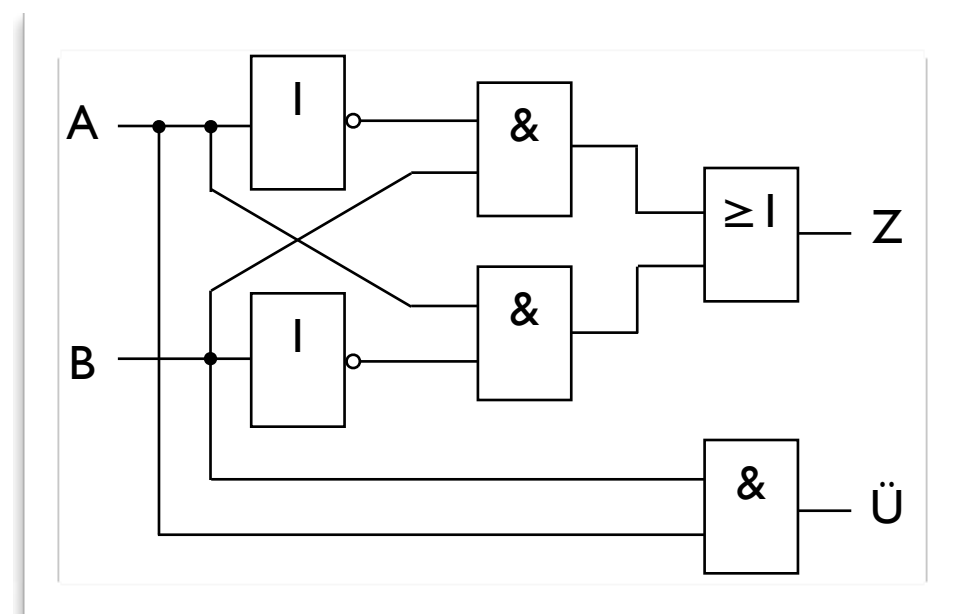


- Funktionsgleichungen:

$$Z = A \oplus B \vee \neg AB$$

$$\ddot{U} = AB$$

- Schaltung:

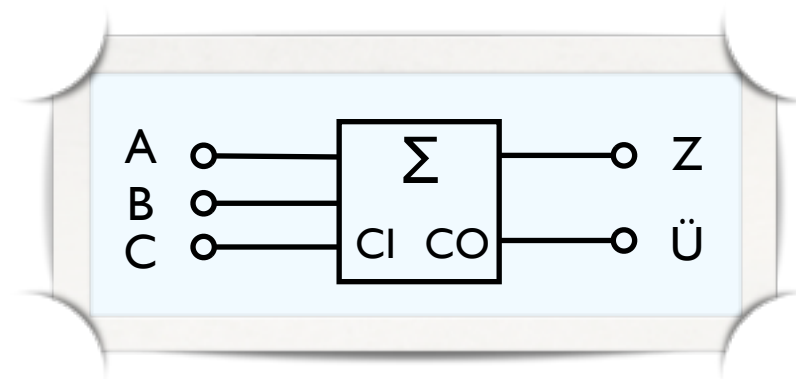


15. Rechenschaltungen

II. Der Volladdierer (VA)

- ermöglicht Addition dreier Dualziffern

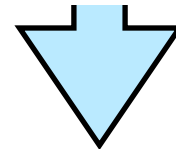
- Schaltzeichen:



- Funktionsgleichungen:

$$\begin{aligned} Z &= A \neg B \neg C \vee \neg A B \neg C \vee \neg A \neg B C \vee A B C \\ \ddot{U} &= A B \neg C \vee A \neg B C \vee \neg A B C \vee A B C \end{aligned}$$

□
KV-Diagramm

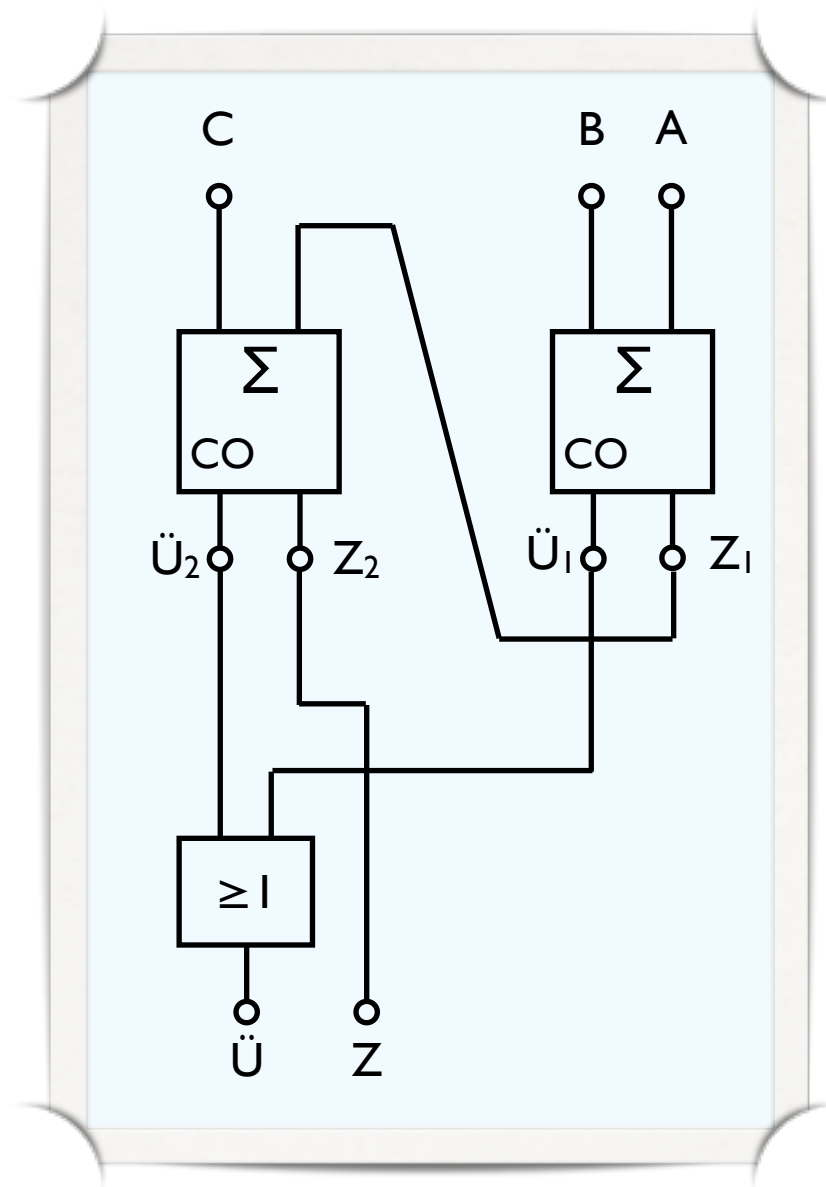


$$\begin{aligned} Z &= A \neg B \neg C \vee \neg A B \neg C \vee \neg A \neg B C \vee A B C \\ \ddot{U} &= A B \vee B C \vee A C \end{aligned}$$

15. Rechenschaltungen

II. Der Volladdierer (VA)

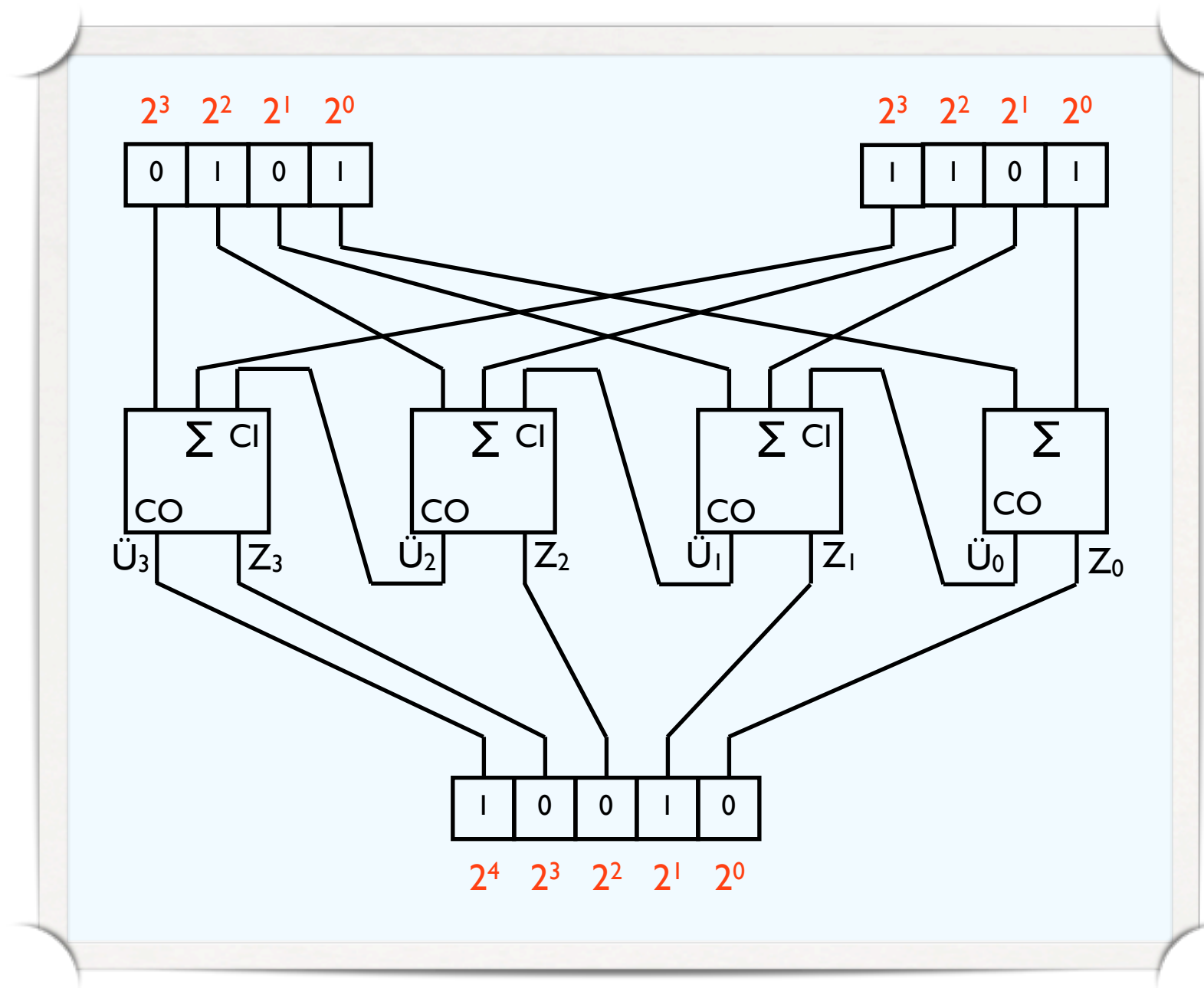
- Aufbau mit zwei Halbaddieren und einem ODER-Glied



15. Rechenschaltungen

III. Parallele Addierschaltung

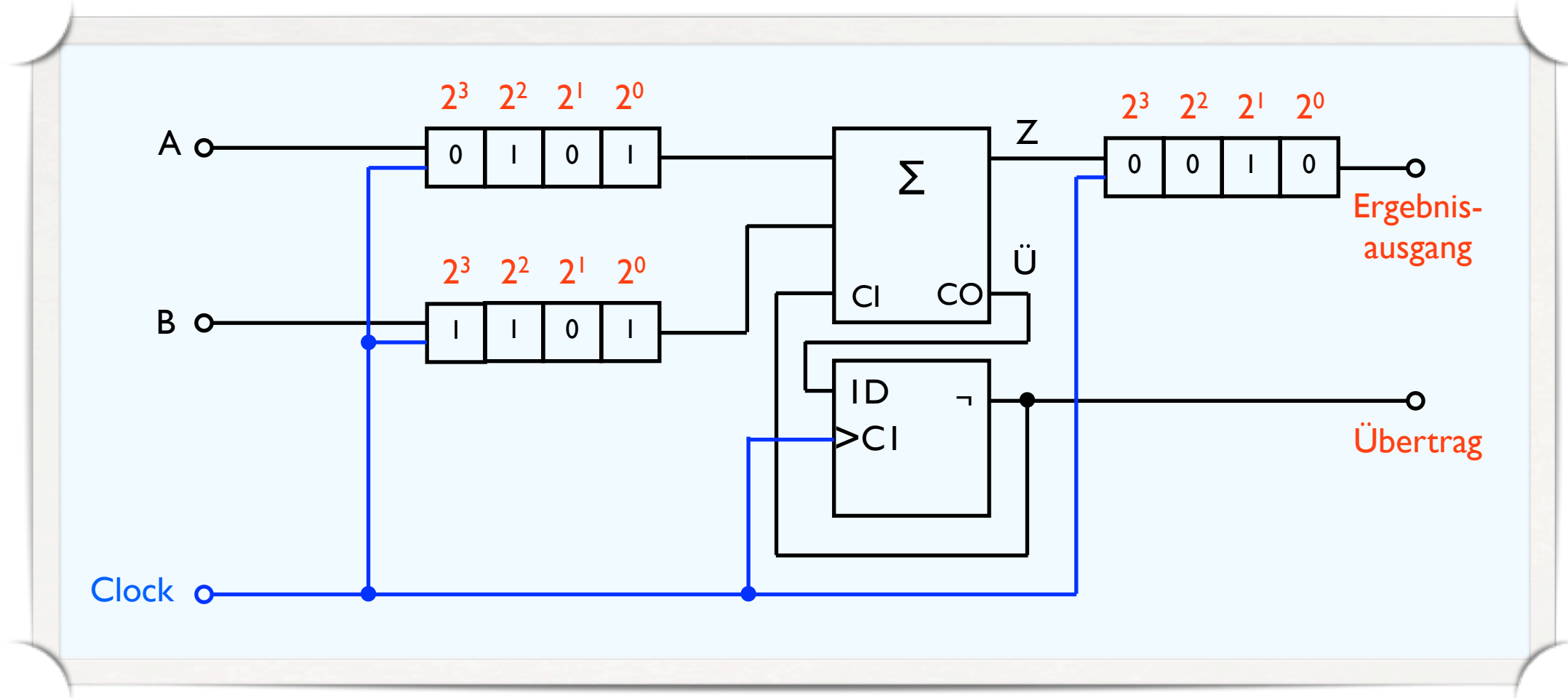
- Addition von zwei 4-Bit Dualzahlen



15. Rechenschaltungen

IV. Serielle Addierschaltung

- Addition von zwei 4-Bit Dualzahlen



- Nachteil: langsamer als parallele Addition

Aufgabe: Erstellen Sie das zugehörige Zeitablaufdiagramm!

?

15. Rechenschaltungen

V. Der Halbsubtrahierer (HS)

- ermöglicht Subtraktion zweier Dualziffern
- Rechenregeln des Halbsubtrahierers

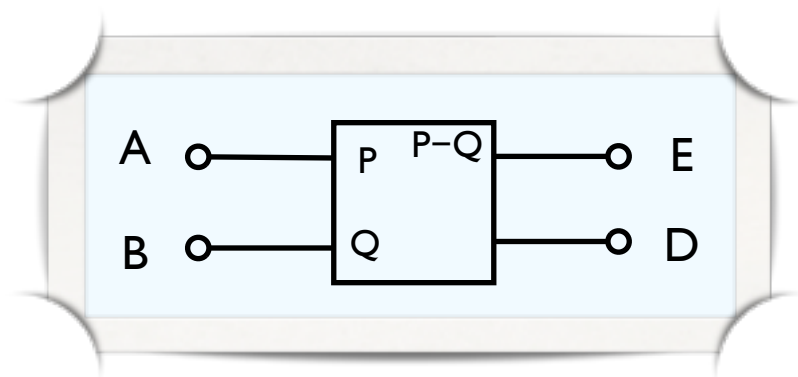
→ $0 - 0 = 0$

→ $0 - 1 = -1$

→ $1 - 0 = 1$

→ $1 - 1 = 0$

- Schaltzeichen:

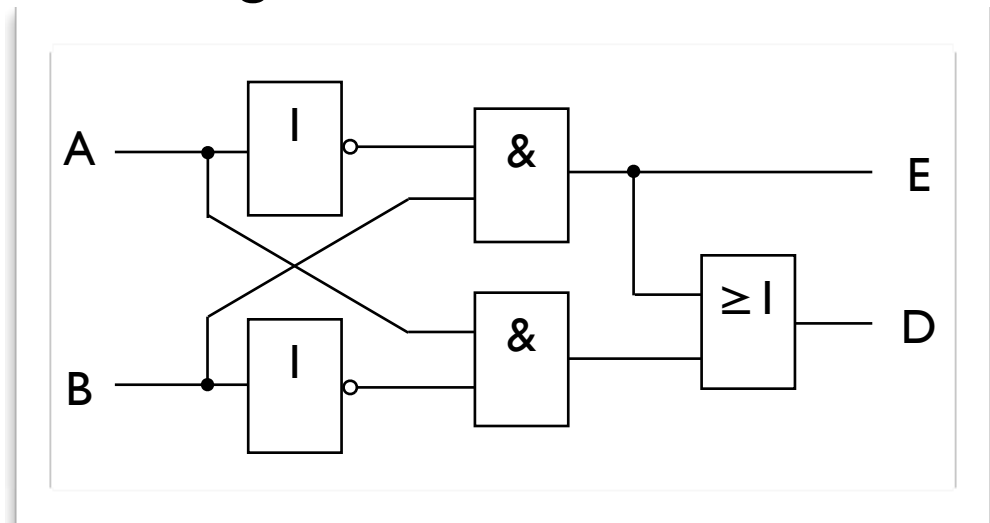


- Funktionsgleichungen:

$$D = A \neg B \vee \neg A B$$

$$E = \neg A B$$

- Schaltung:

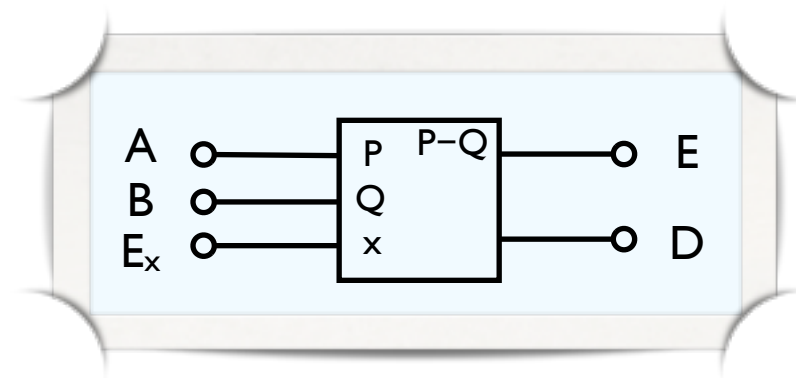


I 5. Rechenschaltungen

VI. Der Vollsubtrahierer (VS)

- ermöglicht Subtraktion dreier Dualziffern (Entleihung wird zum Subtrahend addiert oder in zweitem Schritt abgezogen)

- Schaltzeichen:



- Funktionsgleichungen:

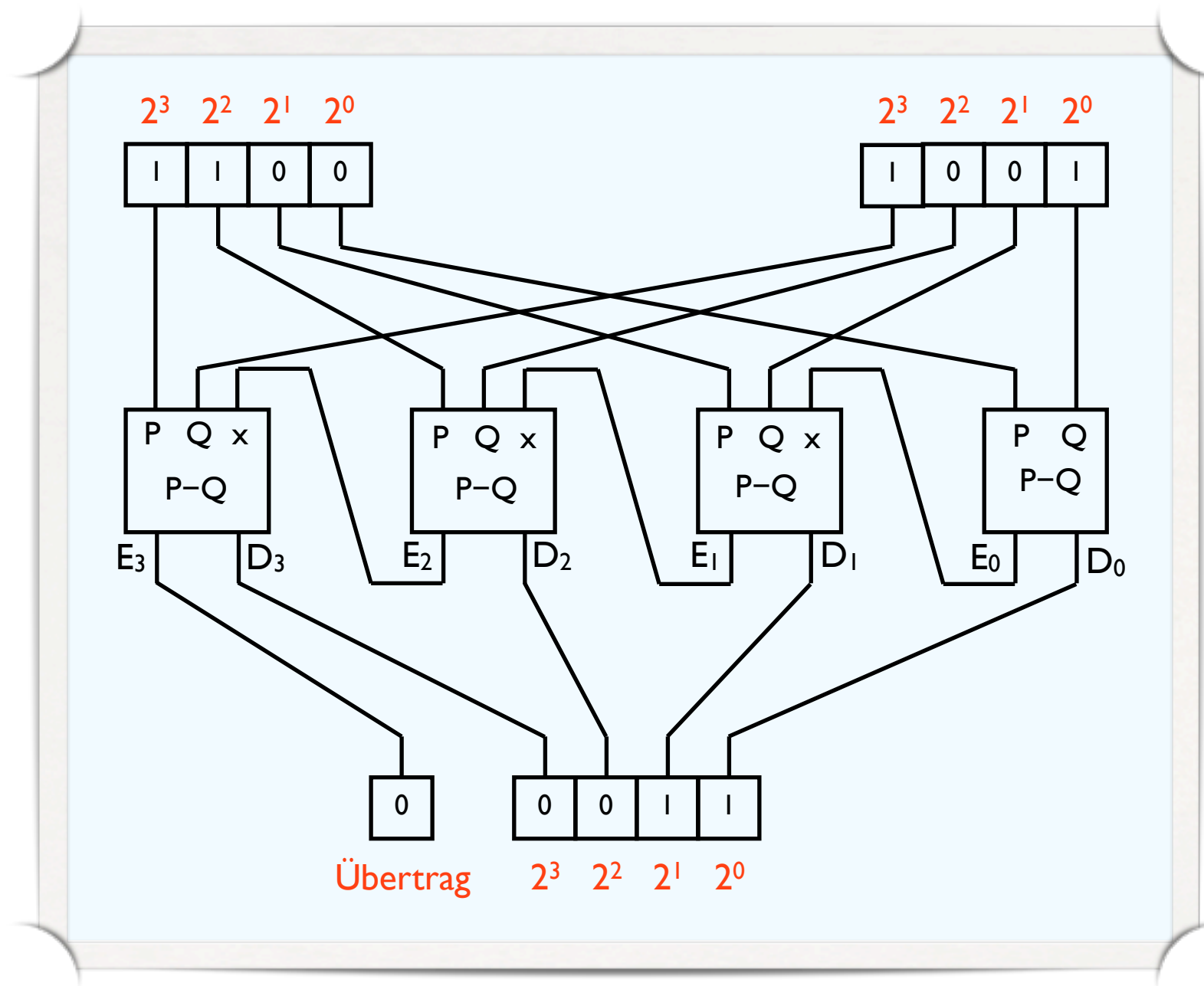
$$\begin{aligned} D &= A - (B + E_x) \\ D &= (A - B) - E_x \end{aligned}$$

- → Zwei Möglichkeiten zum Aufbau eines Vollsubtrahierers

15. Rechenschaltungen

VII. Subtrahierschaltung

- Beispiel: Subtraktion von zwei 4-Bit-Dualzahlen



15. Rechenschaltungen

VII. Subtrahierschaltung

- Beispiel: Subtraktion von zwei 4-Bit-Dualzahlen

Aufgabe: Führen Sie die folgende Subtraktion mit der vorgestellten Subtrahierschaltung durch.

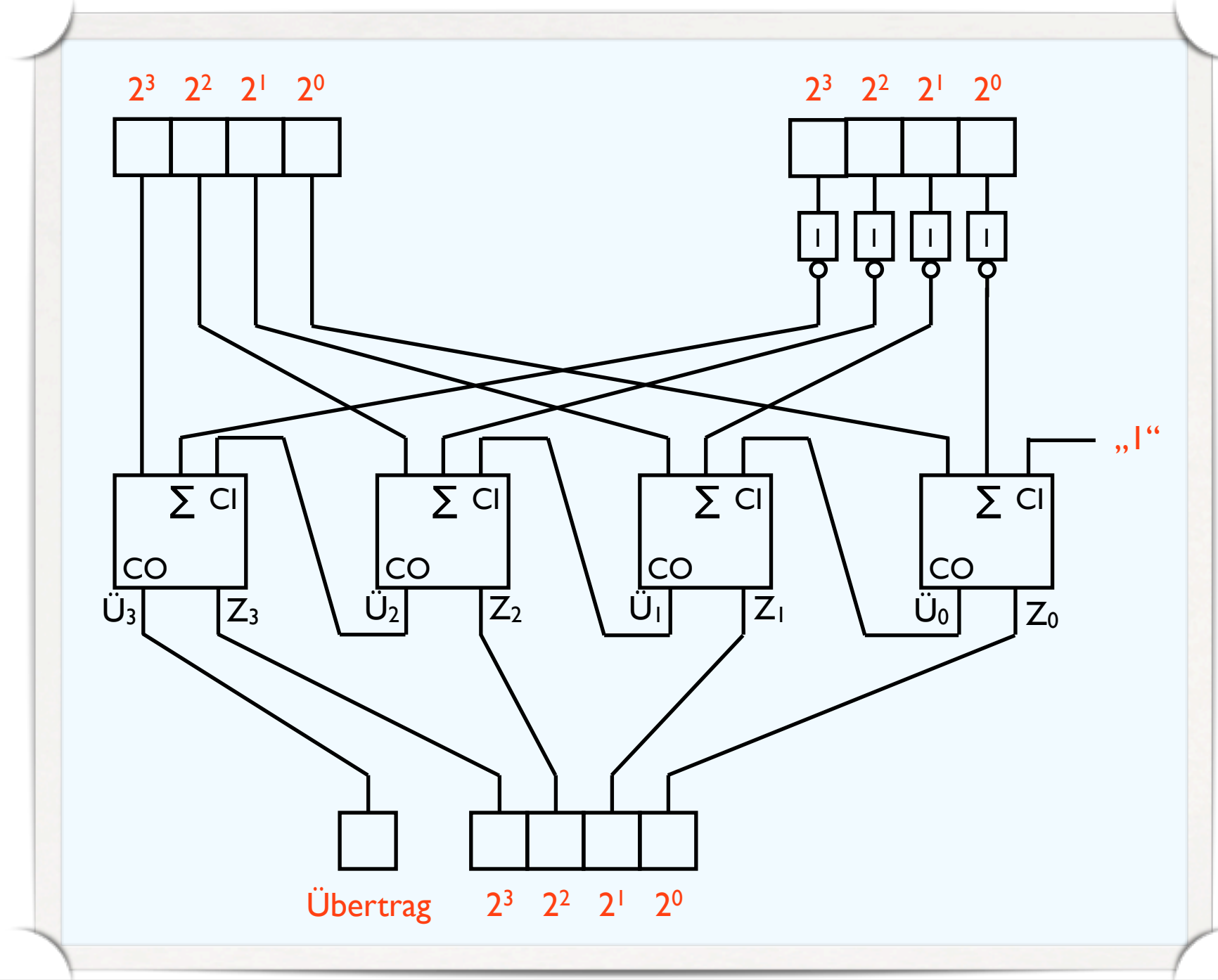
$$1010_2 - 1100_2$$



15. Rechenschaltungen

VII. Subtrahierschaltung

- Überführung der Subtraktion in eine Addition (Komplementbildung)



I 5. Rechenschaltungen

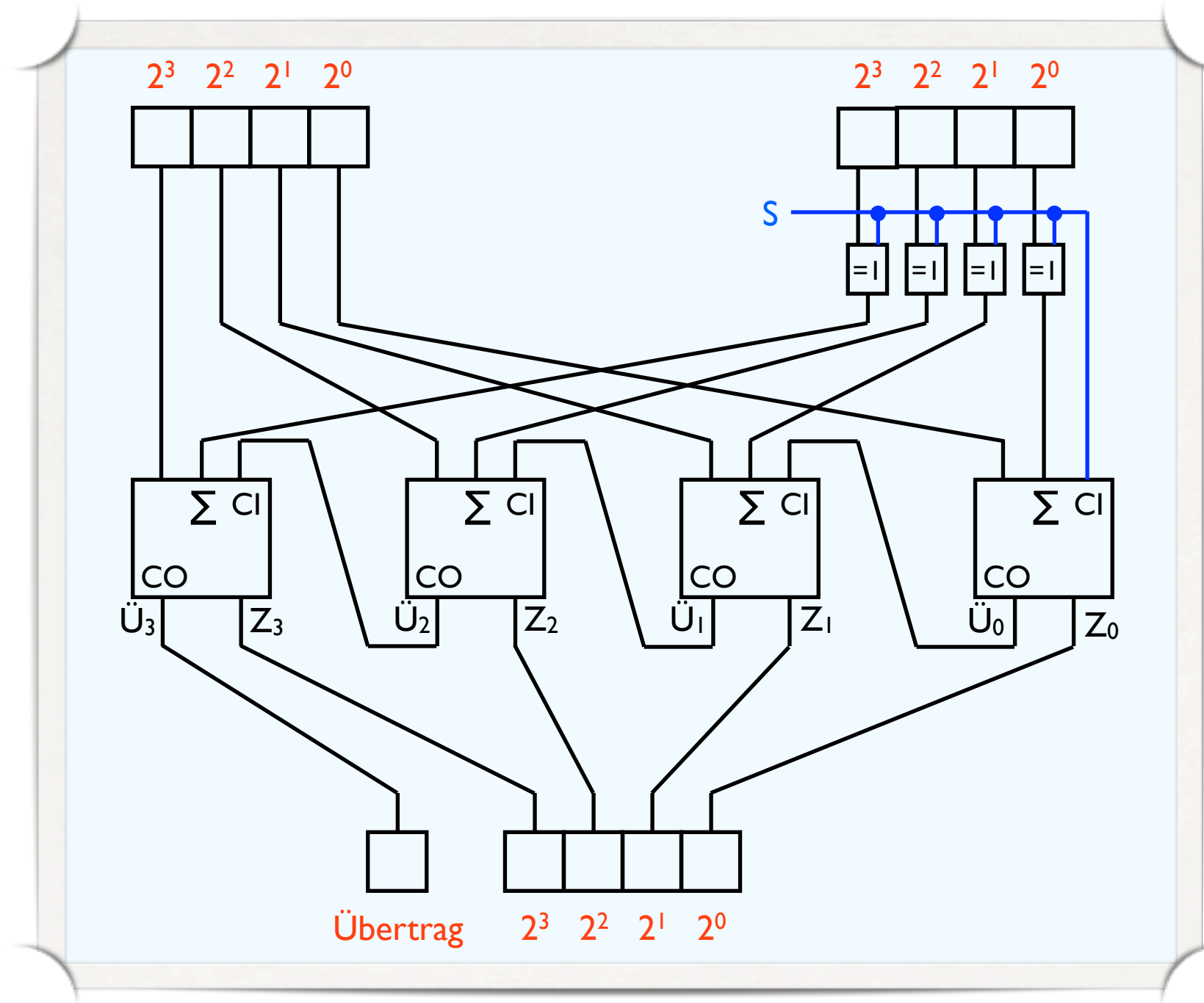
VIII. Addier-Subtrahier-Werk

- Abwandlung der Subtrahierschaltung mit Volladdierern
- keine Komplementbildung für Addier-Subtrahier-Werk:
 - ➔ keine Negation des Subtrahendregisters durchführen
 - ➔ keine Addition der „1“ am ersten Volladdierer
- separater Wahl-Eingang „S“ entscheidet über Subtraktion oder Addition
 - ➔ $S = 0$: Addition: Subtrahendregister wird nicht negiert
 - ➔ $S = 1$: Subtraktion: Komplementbildung des Registerinhalts

15. Rechenschaltungen

VIII. Addier-Subtrahier-Werk

- 4-Bit-Addier-Subtrahier-Werk



15. Rechenschaltungen

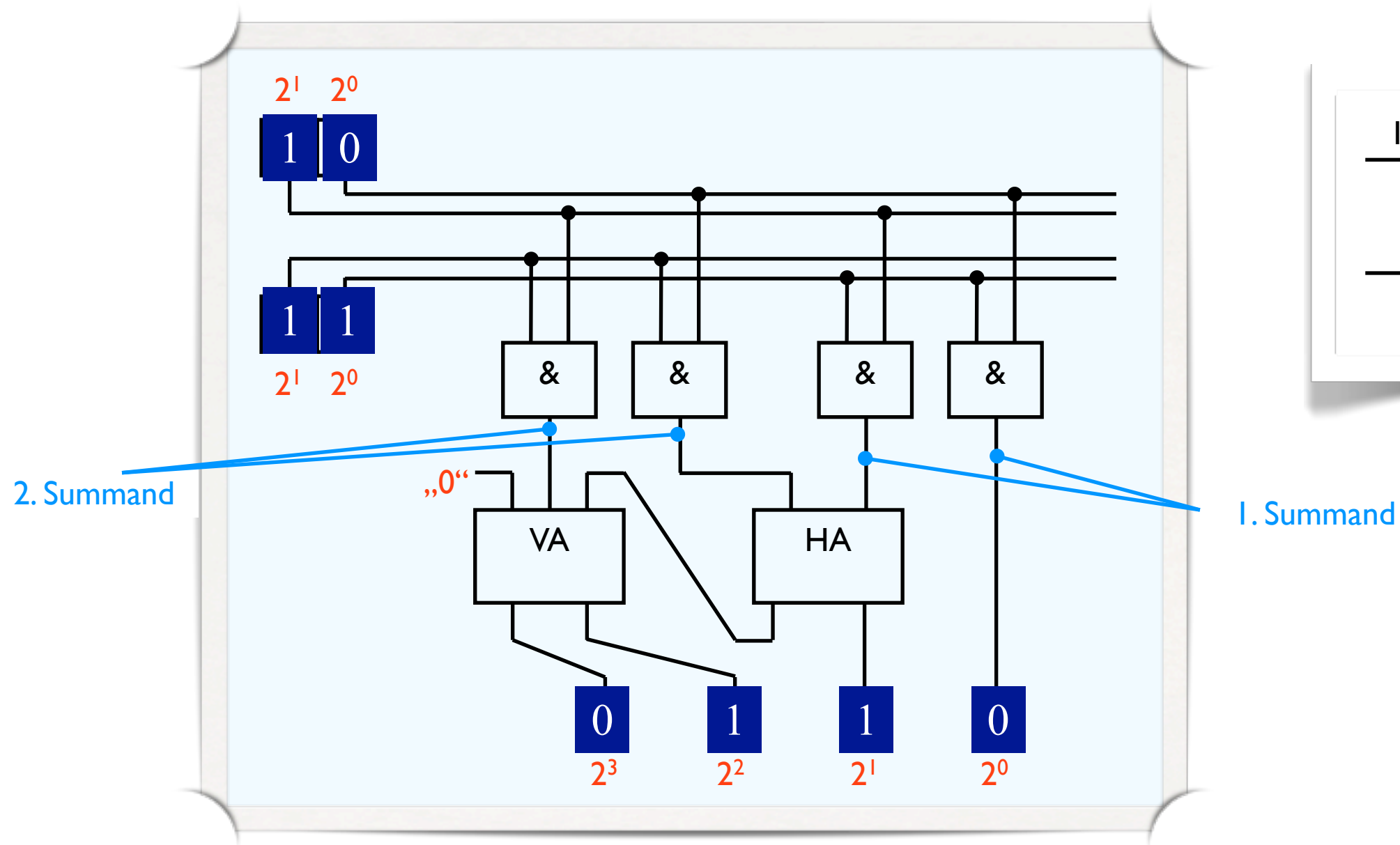
IX. Multiplikationsschaltungen

- ermöglicht Multiplikation zweier Dualziffern
- Rechenregeln des Multiplizierers
 - ➔ $0 \times 0 = 0$
 - ➔ $0 \times 1 = 0$
 - ➔ $1 \times 0 = 0$
 - ➔ $1 \times 1 = 1$
- 1-Bit-Multiplizierer entspricht einem UND-Glied
- parallele oder serielle Multiplikation möglich

15. Rechenschaltungen

IX. Multiplikationsschaltungen

- parallele Multiplikationsschaltung (2-Bit)



1. Faktor	×	2. Faktor
<hr/>		
1. Summand		
2. Summand		
<hr/>		
Summe		

15. Rechenschaltungen

IX. Multiplikationsschaltungen

- parallele Multiplikationsschaltung

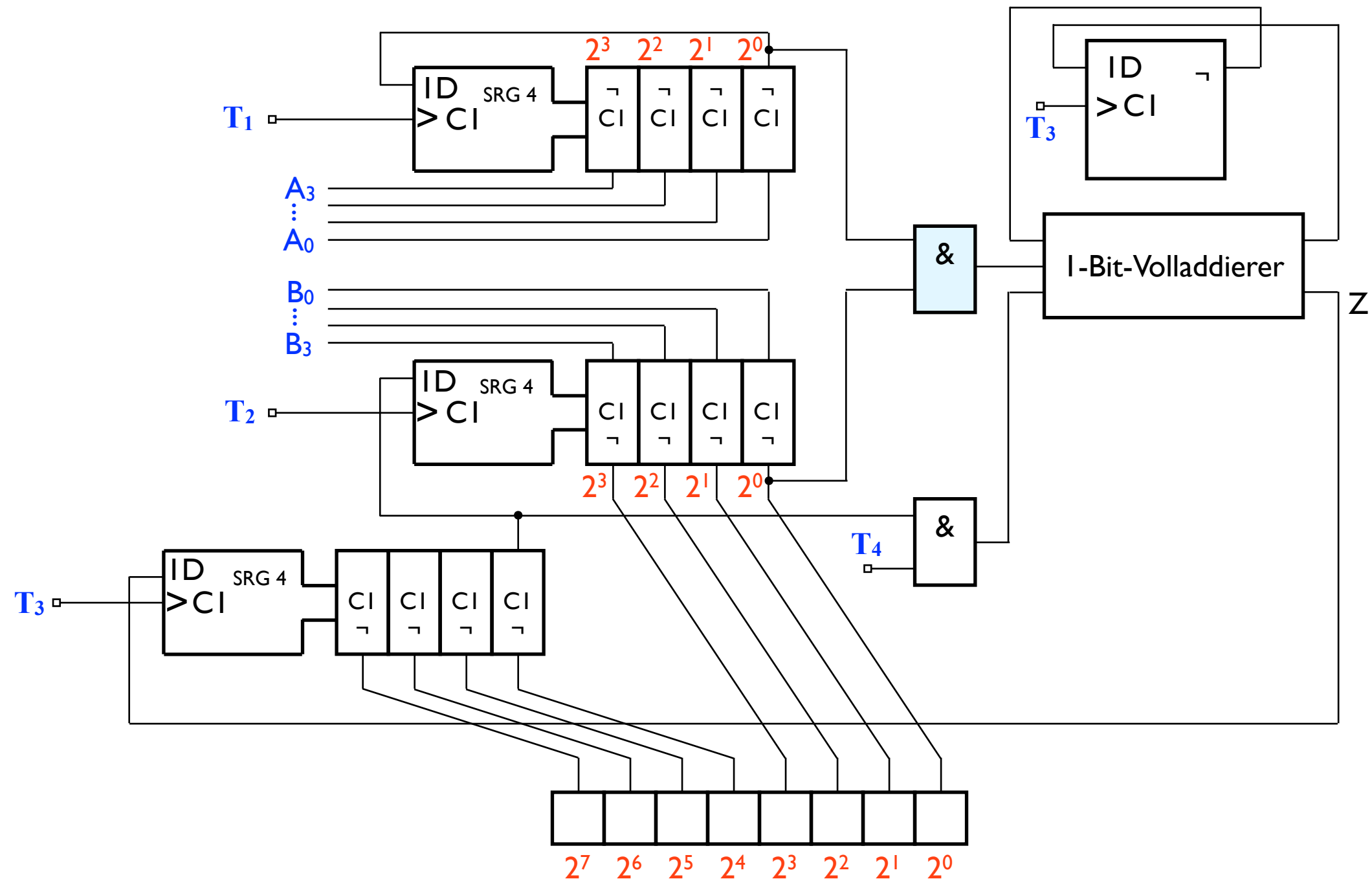
Aufgabe: Wie muss man die 2-Bit-Parallel-Multiplikationsschaltung erweitern, um das Produkt 13×8 berechnen zu können?
Skizzieren Sie die Schaltung!



15. Rechenschaltungen

IX. Multiplikationsschaltungen

- serielle Multiplikationsschaltung (4-Bit)



I 6. Programmierbare Logik

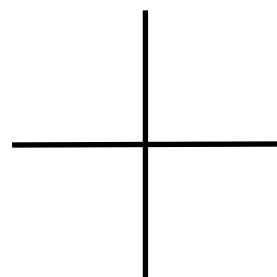
I. Grundlagen

- Unterscheidung programmierter Logikschaltungen
- durch Hersteller: Application Specific Integrated Circuit (ASIC)
 - ➔ Maskenprogrammierung durch Hersteller nach Kundenwunsch
 - ➔ Vorteile von ASIC's
 - geringer Platzbedarf (hohe Packungsdichten)
 - hohe Schaltgeschwindigkeiten
 - niedrige Verluste
 - ➔ Nachteil
 - sehr teuer → rentabel nur bei hohen Stückzahlen
- durch Anwender: Programmable Logic Devices (PLD)

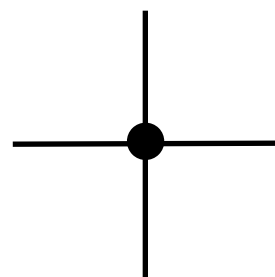
16. Programmierbare Logik

II. Programmable Logic Devices (PLD)

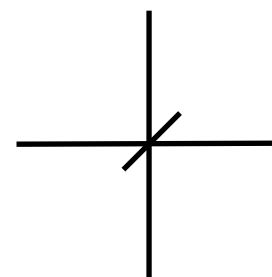
- Darstellung einer Schaltung durch ODER-Normalform möglich
- Konvention von Leitungskreuzungen



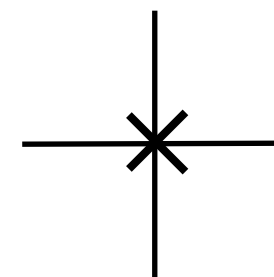
keine
Verbindung



nicht veränderbare
Verbindung



programmierbare
Verbindung
(unterbrochen)



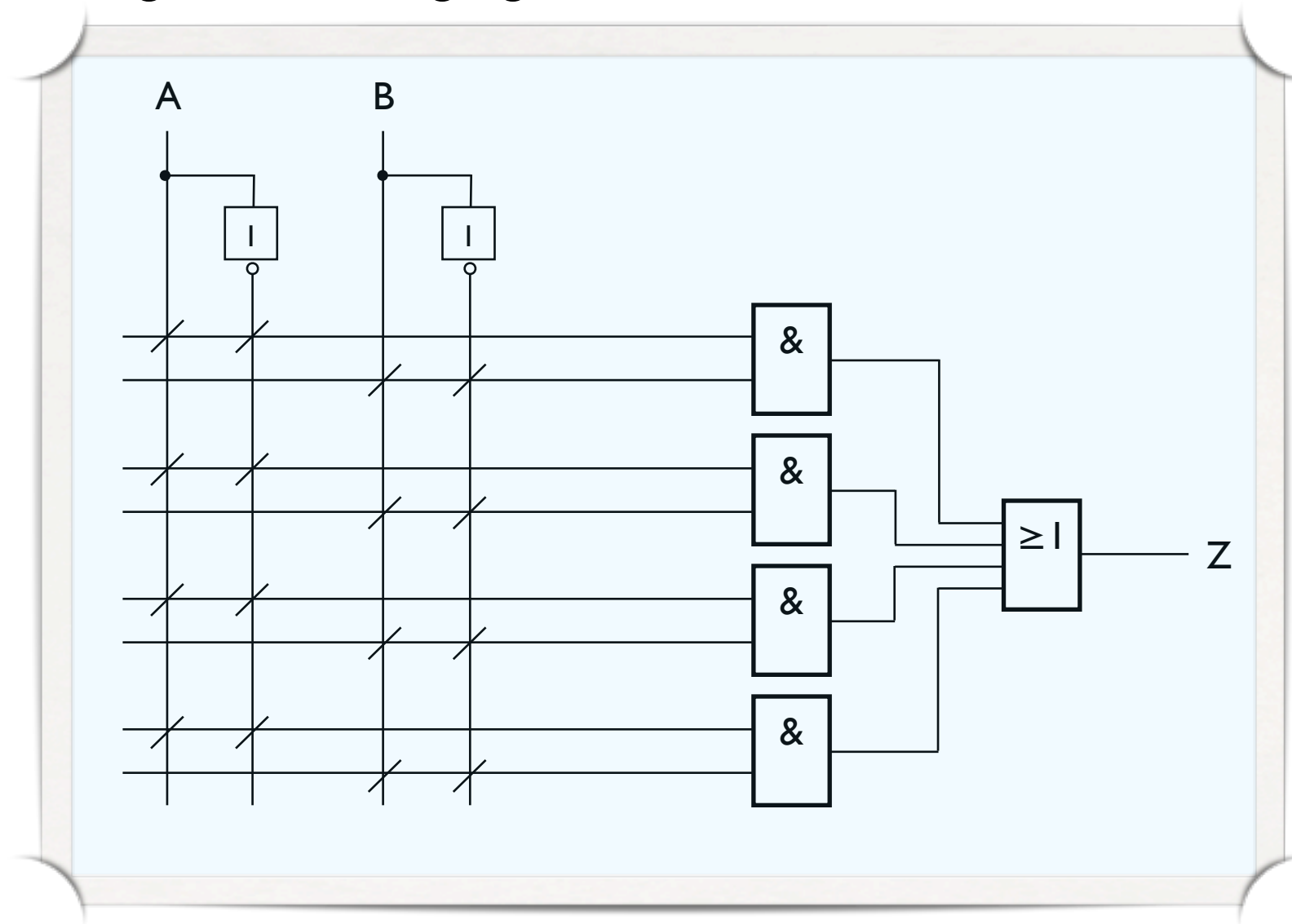
programmierbare
Verbindung
(leitend)

- Programmierung der PLD's
 - ➔ alle Kreuzungspunkte sind vorab leitend verbunden → Durchbrennen der nicht benötigten Verbindungen (irreversibel)
 - ➔ Verwendung von FAMOS-Transistoren (Aufbau analog zu EPROM und EEPROM)
 - Erasable PLD (EPLD): gesamter Baustein durch UV-Licht löschar
 - Electrical Erasable PLD (EEPLD): einzelne Zellen elektronisch löschar

16. Programmierbare Logik

II. Programmable Logic Devices (PLD)

- Schaltung mit zwei Eingängen

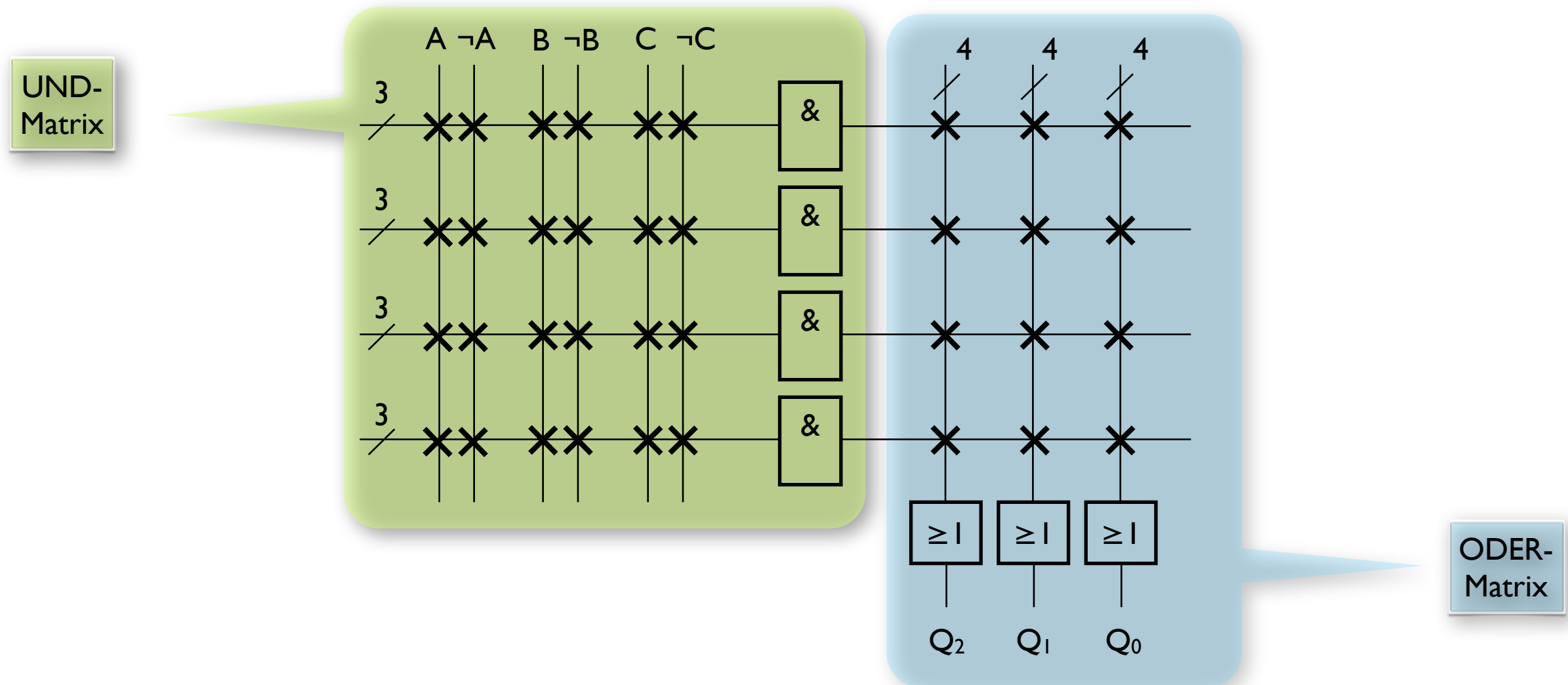


Aufgabe: Erzeugen Sie eine Antivalenzverknüpfung!

I 6. Programmierbare Logik

II. Programmable Logic Devices (PLD)

- PLD-Schaltung

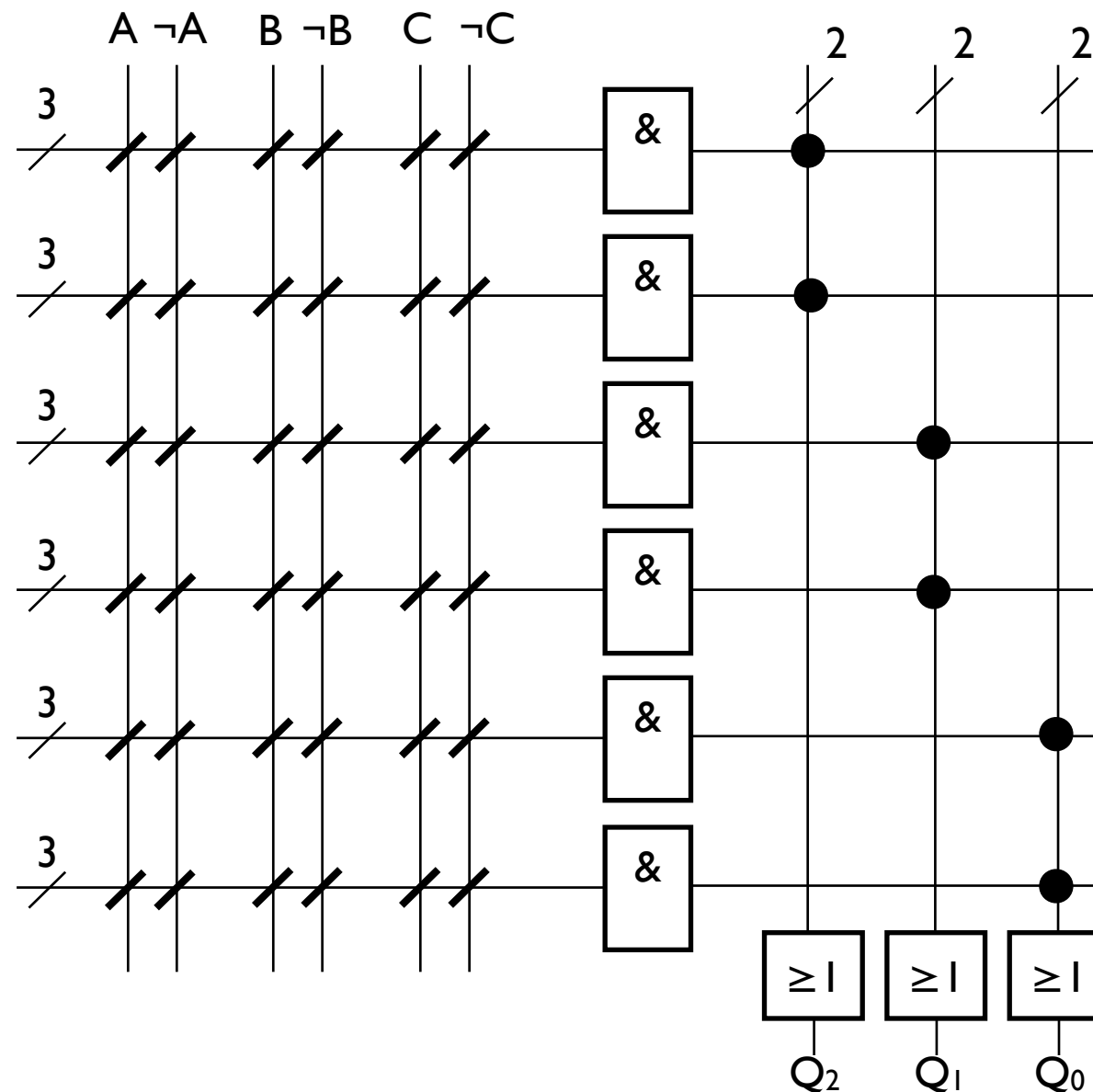


- Vereinfachung durch Verwendung mehradriger Leitungen
- Programmierung durch Lösen nicht benötigter Verbindungen

I 6. Programmierbare Logik

II. Programmable Logic Devices (PLD)

- Programmable Array Logic (PAL)
 - ➔ Programmierbare Feld-Logik
 - ➔ enthält programmierbare UND-Matrix und festverdrahtete ODER-Matrix



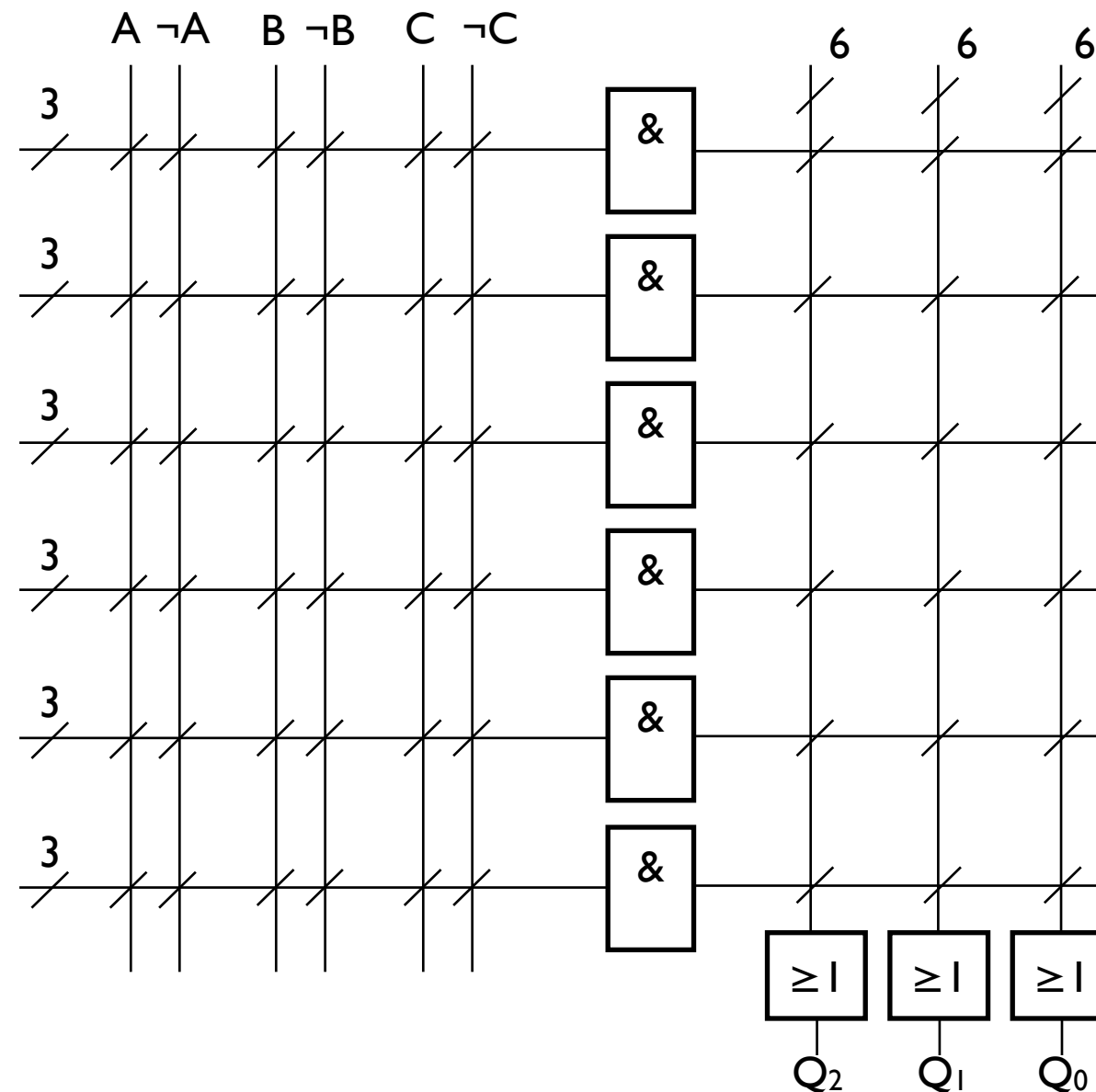
II. Programmable Logic Devices (PLD)

- Generic Array Logic (GAL)
 - ➔ Gattungsfeld-Logik (EPLD oder EEPLD)
 - ➔ enthält programmierbare UND-Matrix, Ein- und Ausgabeblocks sowie festverdrahtete ODER-Matrix
 - ➔ Output Logic Macro Cell (OLMC-Schaltung) an jedem Ausgang
 - 8-fach ODER-Glied (Darstellung der ODER-Normalform)
 - D-Flipflop (Ergebnisspeicherung)
 - Steuerungseinheit
 - Ausgang (normal und negiert)

I 6. Programmierbare Logik

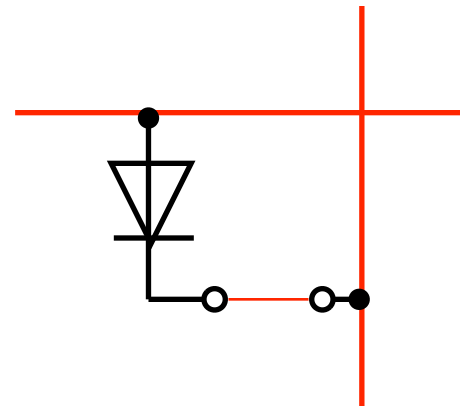
II. Programmable Logic Devices (PLD)

- Field Programming Logic Array (FPLA)
 - ➔ feldprogrammierbare Logik
 - ➔ enthält programmierbare UND-Matrix sowie programmierbare ODER-Matrix



II. Programmable Logic Devices (PLD)

- PROM-Schaltungen
 - ➔ Verwendung PROM als PLD möglich
 - ➔ Beispiel: Dioden-PROM mit Adressdekodierer



- Adressdekodierer: UND-Matrix (→ fest verdrahtet)
- Speicher: ODER-Matrix (→ programmierbar)

I 6. Programmierbare Logik

II. Programmable Logic Devices (PLD)

- Übersicht und Einteilung von PLD-Typen

SCHALTUNG			
programmierbare UND-Matrix	programmierbare ODER-Matrix	programmierbare UND- und ODER- Matrix	MACRO-Schaltung
<ul style="list-style-type: none">▶ PAL▶ GAL▶ EPAL▶ AGA▶ LCA	<ul style="list-style-type: none">▶ PROM▶ EPROM▶ EEPROM▶ PLE	<ul style="list-style-type: none">▶ FPLA▶ EPL▶ FPLAS▶ FPLS▶ FPGA▶ PL	<ul style="list-style-type: none">▶ Macrocell▶ μ PLD▶ Macrochip▶ Macrocell- Array