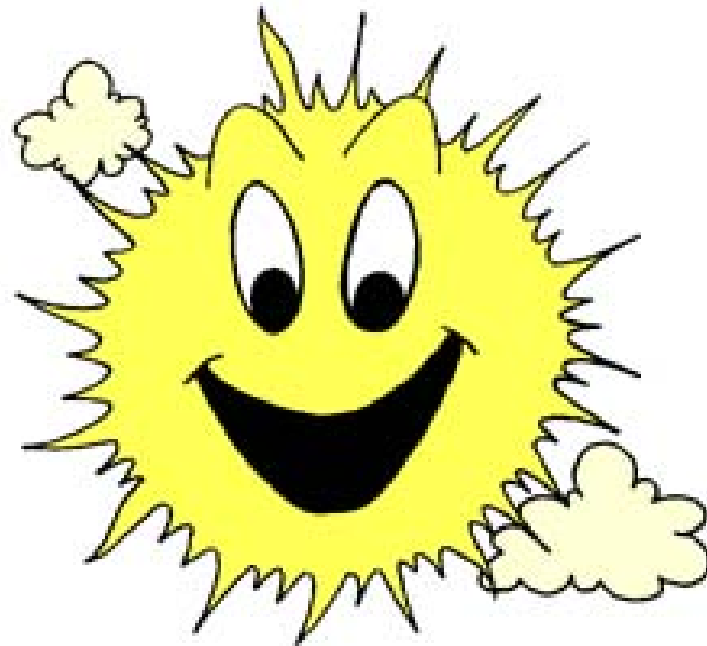


Wiederholung der 5. Vorlesung



10.2.4 Zyklische Folgeschaltung (mit JK-FF)

t	A	B	C
0	0	0	0
1	1	1	1
2	1	0	1
3	1	1	0
4	0	0	1
5	0	1	0
6	0	0	0
7	1	1	1

- Das System wird vom Takt getrieben
- 6 Zustände, zyklische Wiederholung
- 3 Flip-Flops können 8 Zustände
- Realisierung mit **JK**-Flip-Flops

10.2.4 Zyklische Folgeschaltung mit JK (2)

A	B	C	JA	KA	JB	KB	JC	KC	Index
0	0	0	1	=	1	=	1	=	0
1	1	1	-	0	=	1	-	0	7
1	0	1	-	0	1	=	=	1	5
1	1	0	=	1	=	1	1	=	6
0	0	1	0	-	1	=	=	1	1
0	1	0	0	-	=	1	0	-	2
0	0	0	1	=	1	=	1	=	0

Aufstellung der Wertetabelle für die Übergänge

- für jeden der 2 Eingänge der 3 Flipflops
- '=' don't care: mögliche Toggle-Beziehung $J=K=1$
- '-' don't care: Wert ist egal, kann beliebig gewählt werden



- 6 KV-Diagramme
- Eintrag in KV-Diagramme an den richtigen Stellen

10.2.4 Zyklische Folgeschaltung mit JK (3)

A	B	C	JA	KA	JB	KB	JC	KC	Index
0	0	0	1	=	1	=	1	=	0
1	1	1	-	0	=	1	-	0	7
1	0	1	-	0	1	=	=	1	5
1	1	0	=	1	=	1	1	=	6
0	0	1	0	-	1	=	=	1	1
0	1	0	0	-	=	1	0	-	2
0	0	0	1	=	1	=	1	=	0

Beachte bei zyklischer Folgeschaltung den **richtigen Index** für die jeweiligen Min- bzw. Max-Terme

→ KV-Diagramme

JA	C		C	
	1	0	-	-
B	0	-	-	=
			A	A

JB	C		C	
	1	1	1	-
B	=	-	=	=
			A	A

JC	C		C	
	1	=	=	-
B	0	-	-	1
			A	A

KA	C		C	
	=	-	0	-
B	-	-	0	1
			A	A

KB	C		C	
	=	=	=	-
B	1	-	1	1
			A	A

KC	C		C	
	=	1	1	-
B	-	-	0	=
			A	A

→ jeweils DMF oder KMF bestimmen, Gleichungen ermitteln, Schaltung zeichnen und realisieren

10.2.4 Zyklische Folgeschaltung mit JK (4)

JA	C	C	
1	0	-	-
B	0	-	=
		A	A

JB	C	C	
1	1	1	-
B	=	-	=
		A	A

JC	C	C	
1	=	=	-
B	0	-	1
		A	A

KA	C	C	
=	-	0	-
B	-	0	1
		A	A

KB	C	C	
=	=	=	-
B	1	-	1
		A	A

KC	C	C	
=	1	1	-
B	-	0	=
		A	A

$$JA = \bar{B} \wedge \bar{C}$$

$$JB = 1$$

$$JC = \bar{B} \vee A$$

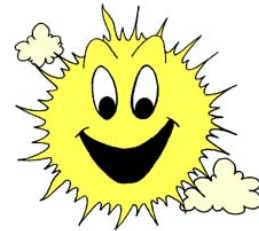
$$KA = \bar{C}$$

$$KB = 1$$

$$KC = \bar{B}$$

➔ ... Schaltung zeichnen und realisieren

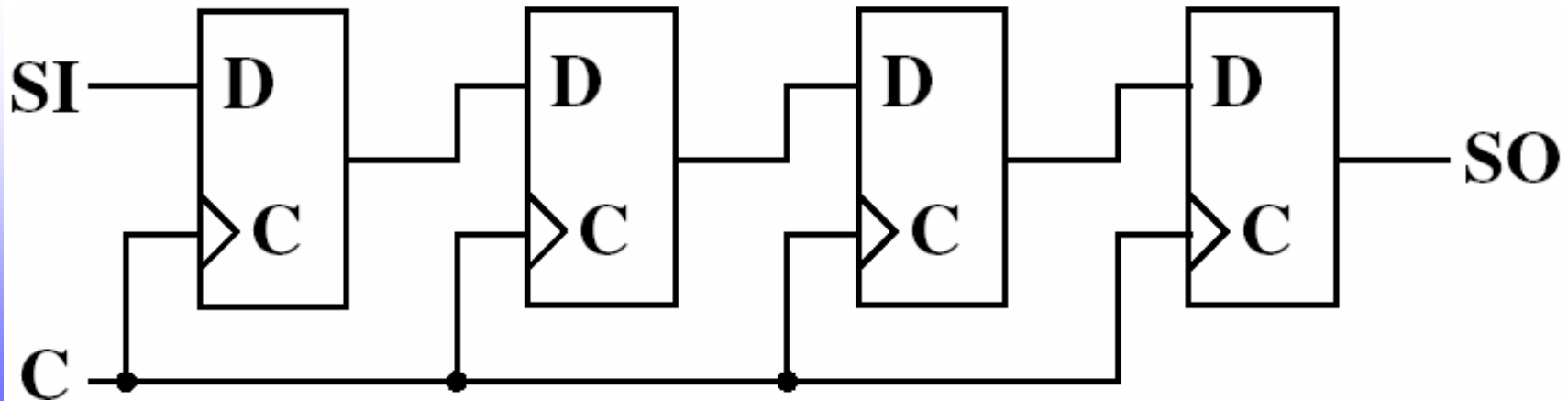
Ende der Wiederholung



11. Spezielle Schaltwerke

Im Folgenden betrachten wir einige Beispiele typischer Schaltwerke mit Flipflops, wie man sie beispielsweise in Rechnern wiederfindet.

11. Schieberegister (SISO)



- 4-bit-Schieberegister (serieller Eingang, serieller Ausgang).
- Mit jeder positiven Flanke wird jedes Bit ein Flip-Flop weiter nach rechts verschoben.

Speicherung und Wiedergabe einer Bitfolge, FIFO

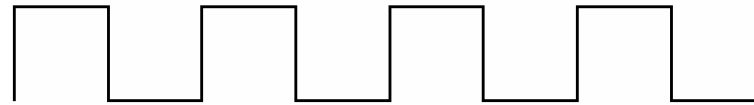
SISO = **S**erial **I**n – **S**erial **O**ut

11. Schieberegister (SISO) Impulsverlauf

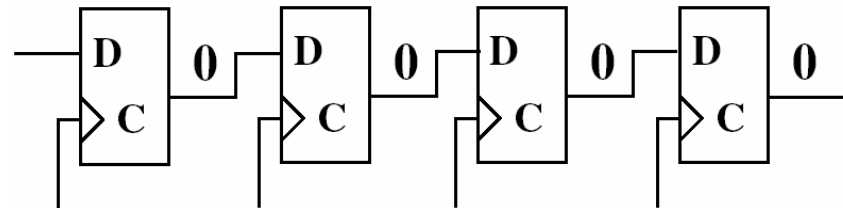
Daten



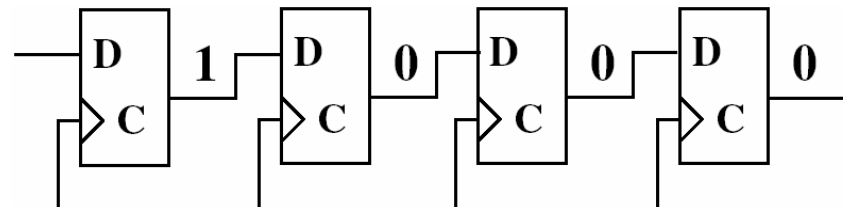
Takt



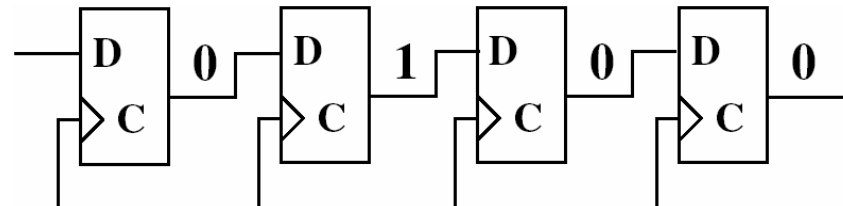
Reset



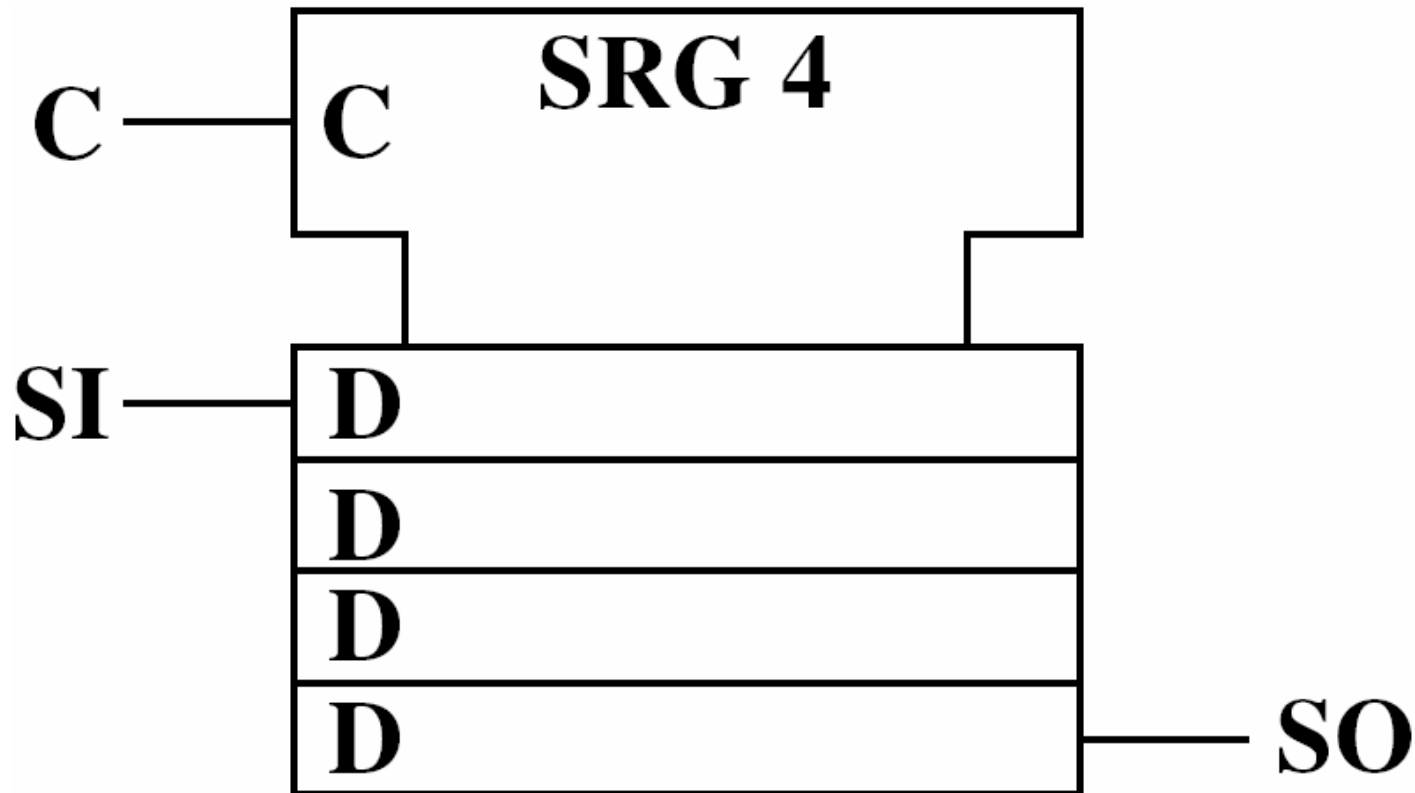
1. Takt



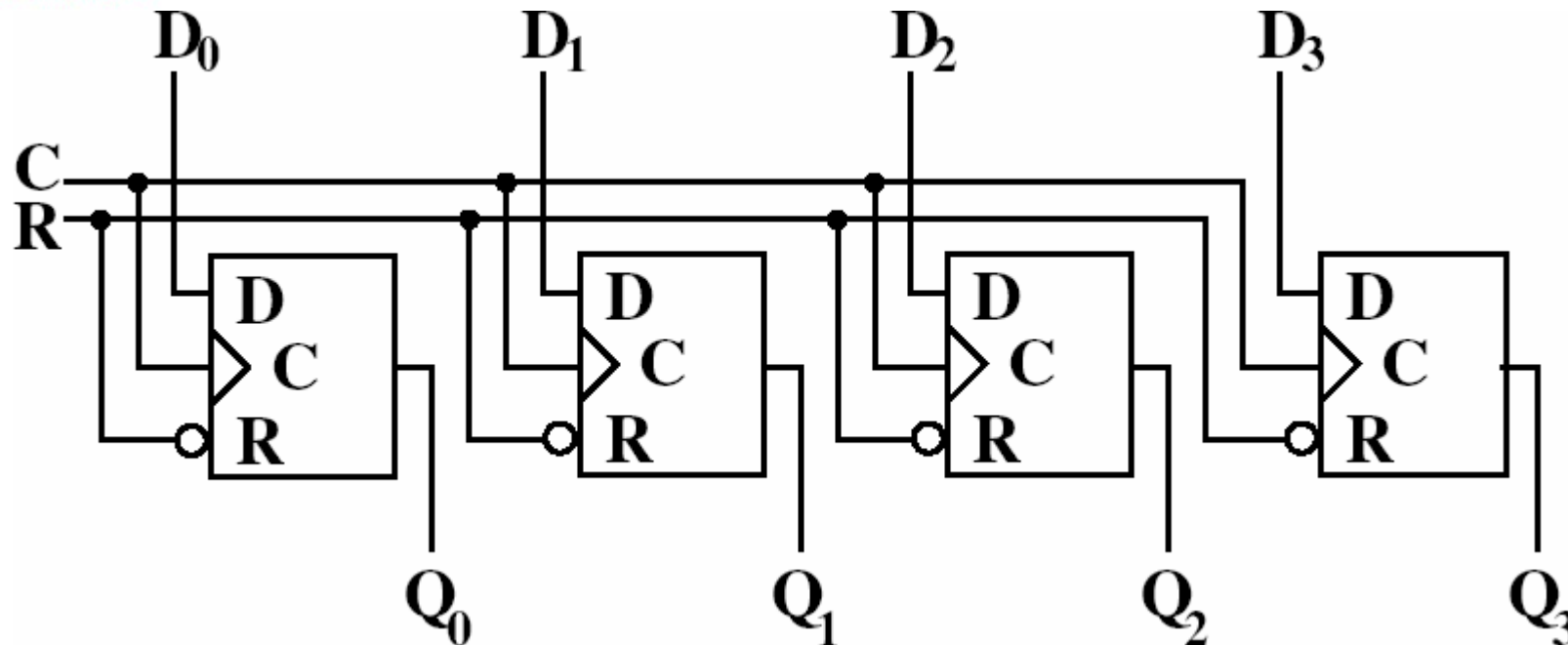
2. Takt



11. Schieberegister (SISO) Schaltsymbol



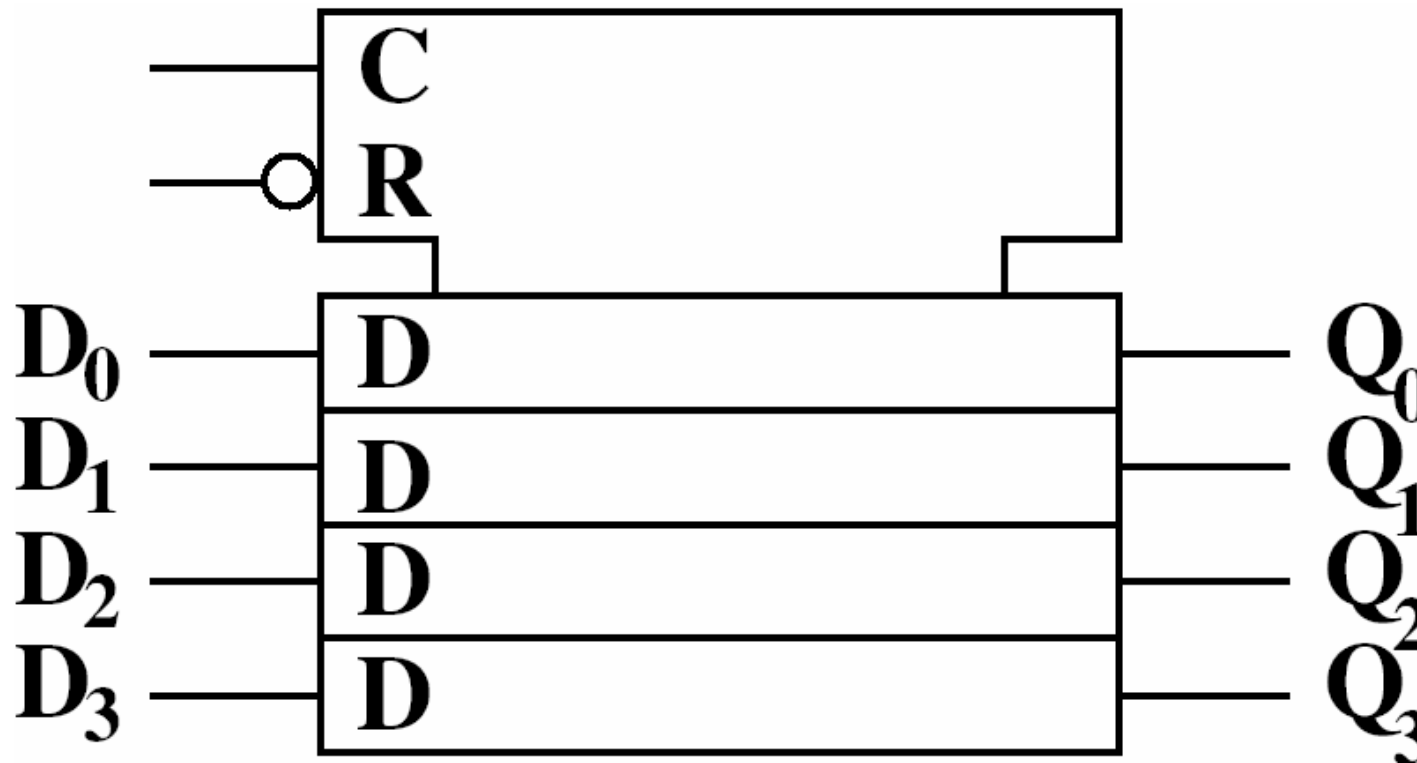
11. Auffangregister (PIPO)



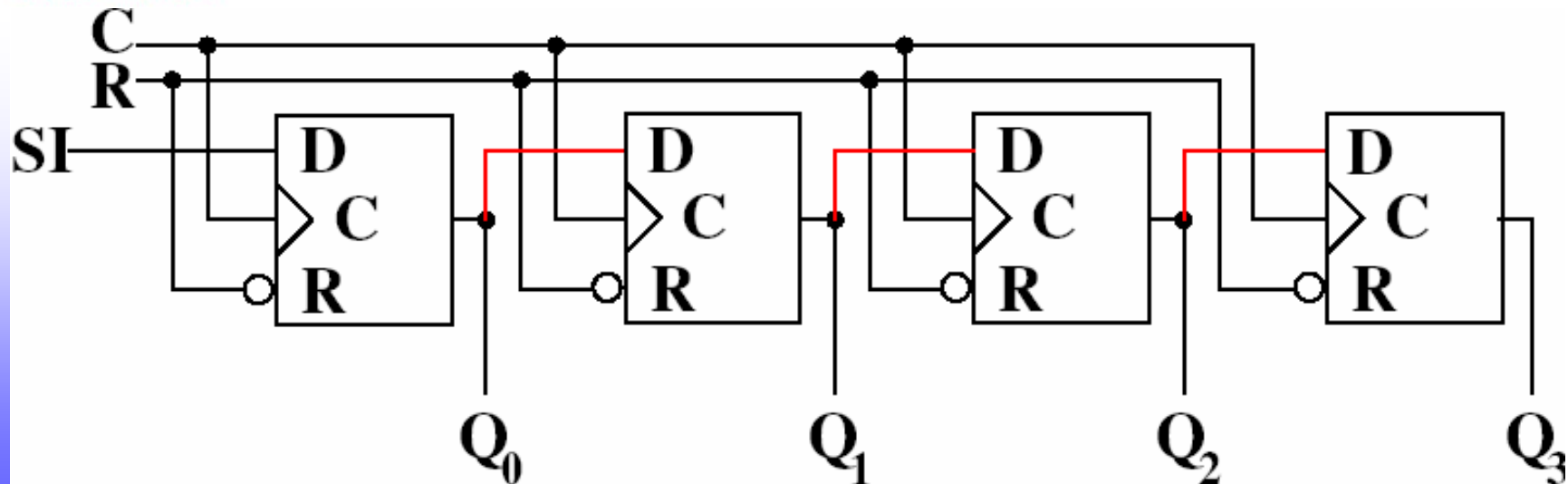
- 4 bit werden parallel gespeichert und können parallel gelesen werden
- Mit dem Reset Eingang (R) können alle Ausgänge auf 0 gesetzt werden.

PIPO = **P**arallel **I**n – **P**arallel **O**ut

11. Auffangregister (PIPO) Schaltsymbol



11. Schieberegister (SIPO)

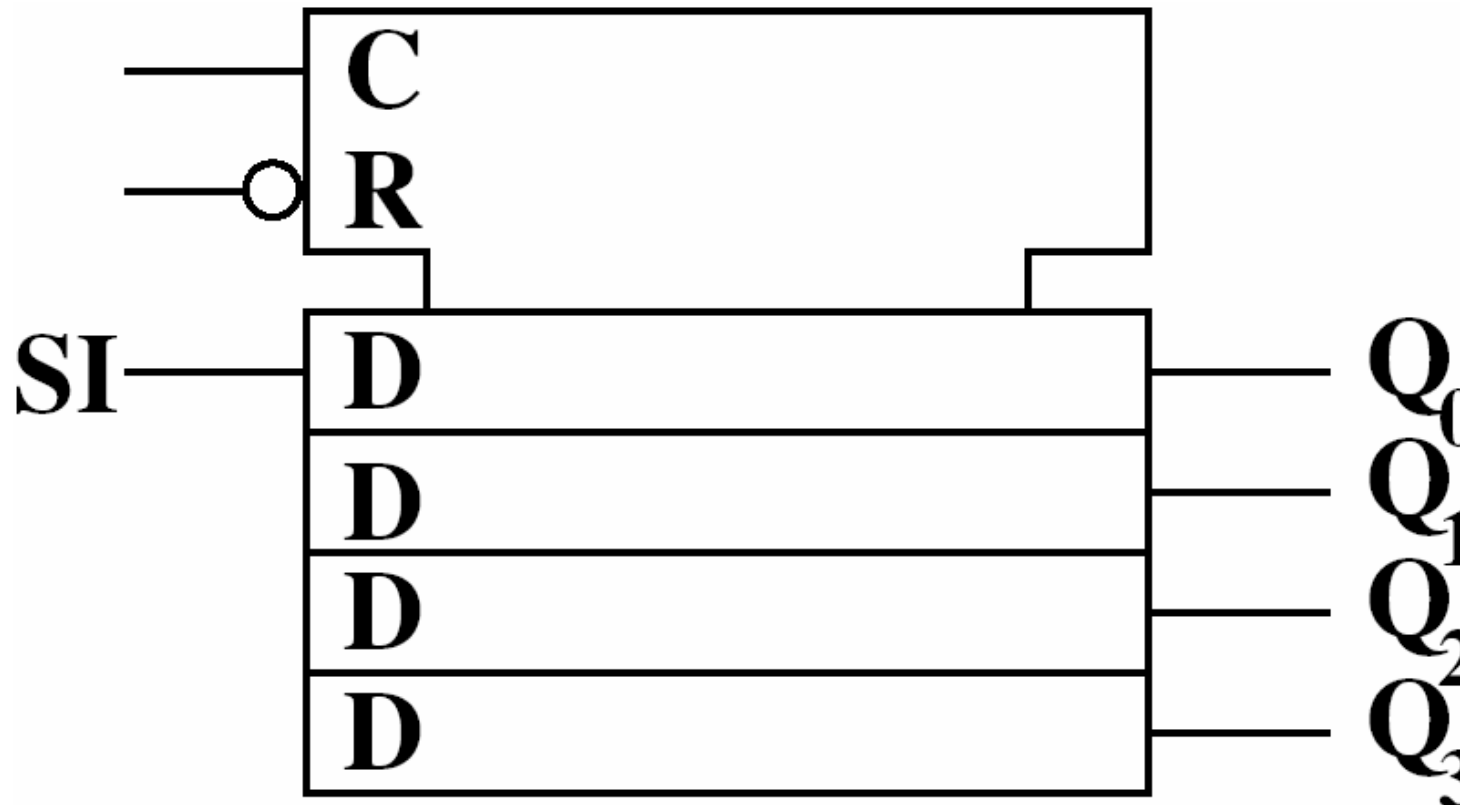


- Daten werden **seriell eingelesen** und **parallel ausgelesen**.
- Mit dem Reset (R) Eingang können alle Ausgabewerte auf 0 gesetzt.

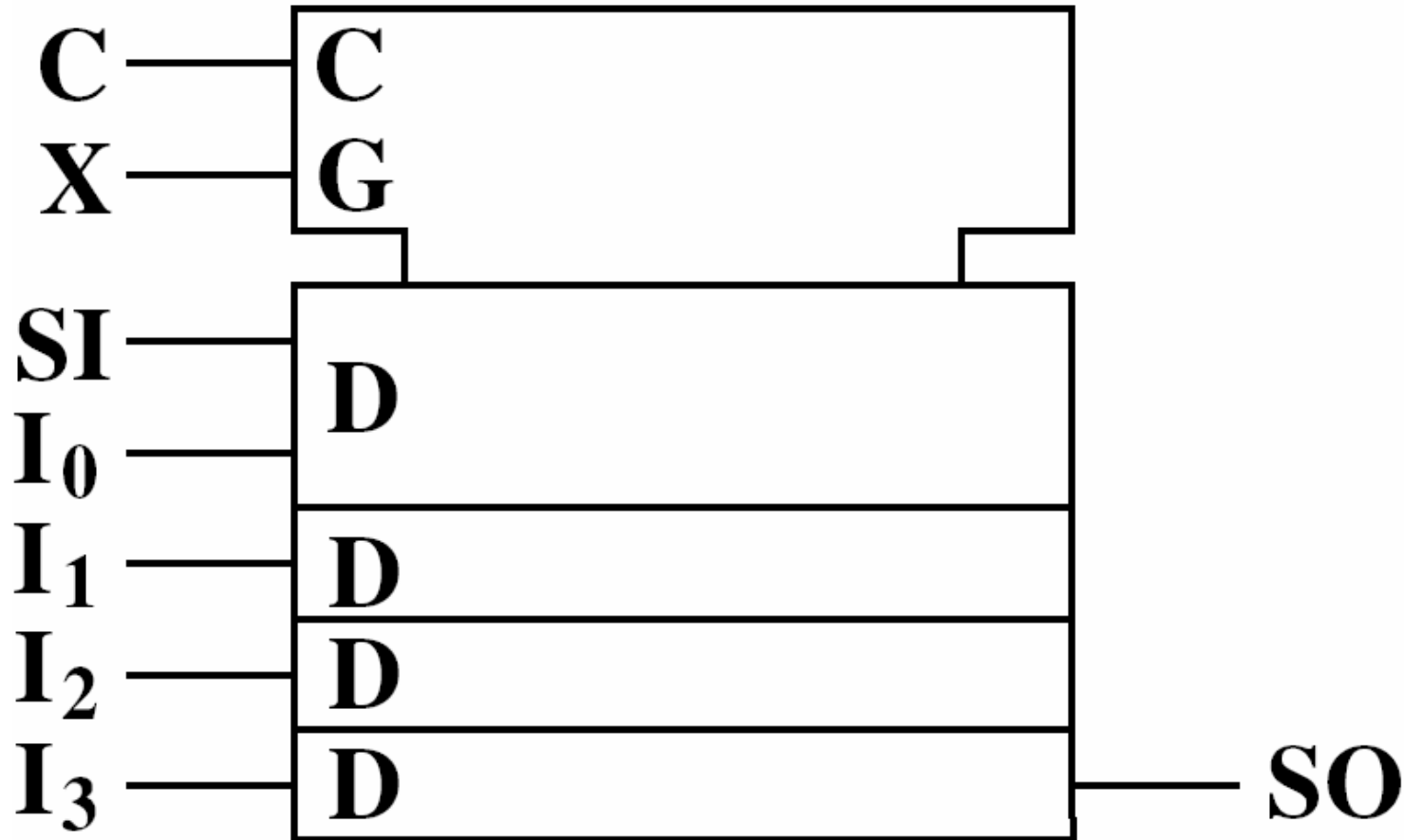
Empfangen einer seriellen Bitfolge. Daten stehen parallel am Ausgang zur Verfügung.

SIPO = **S**erial **I**n – **P**arallel **O**ut

11. Schieberegister (SIPO) Schaltsymbol

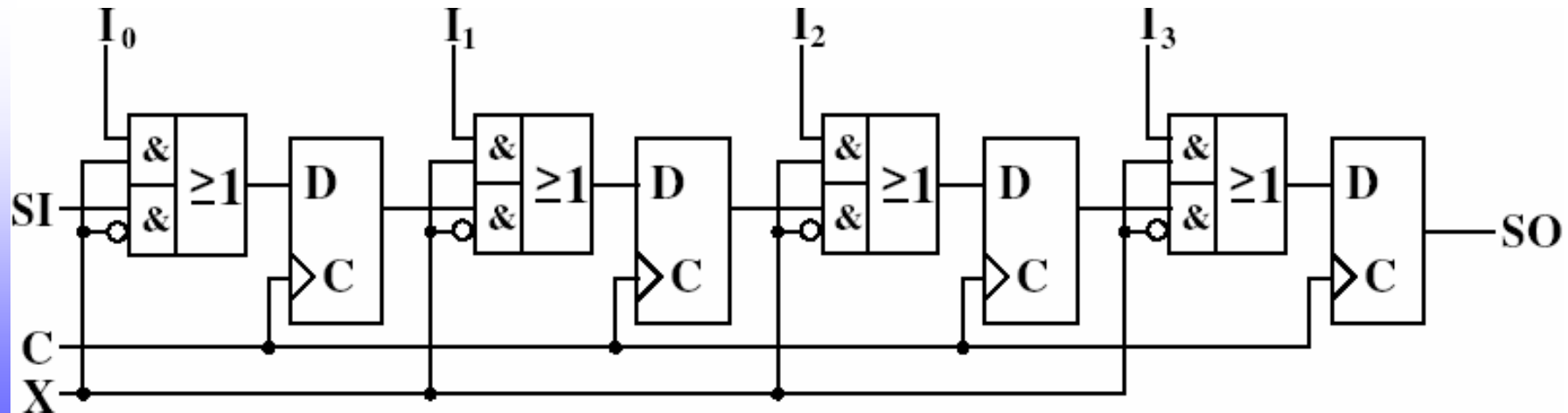


11. Schieberegister (PISO) Schaltsymbol



HS-Übung: Gemeinsamer Entwurf der Schaltung

11. Schieberegister (PISO)

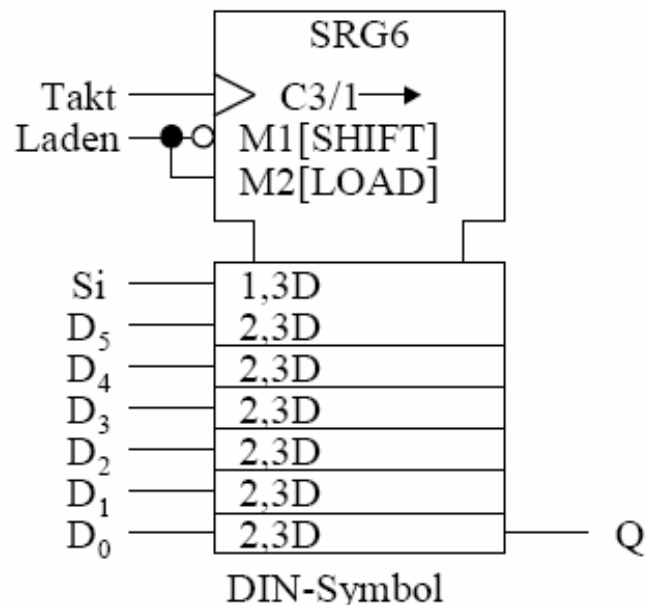
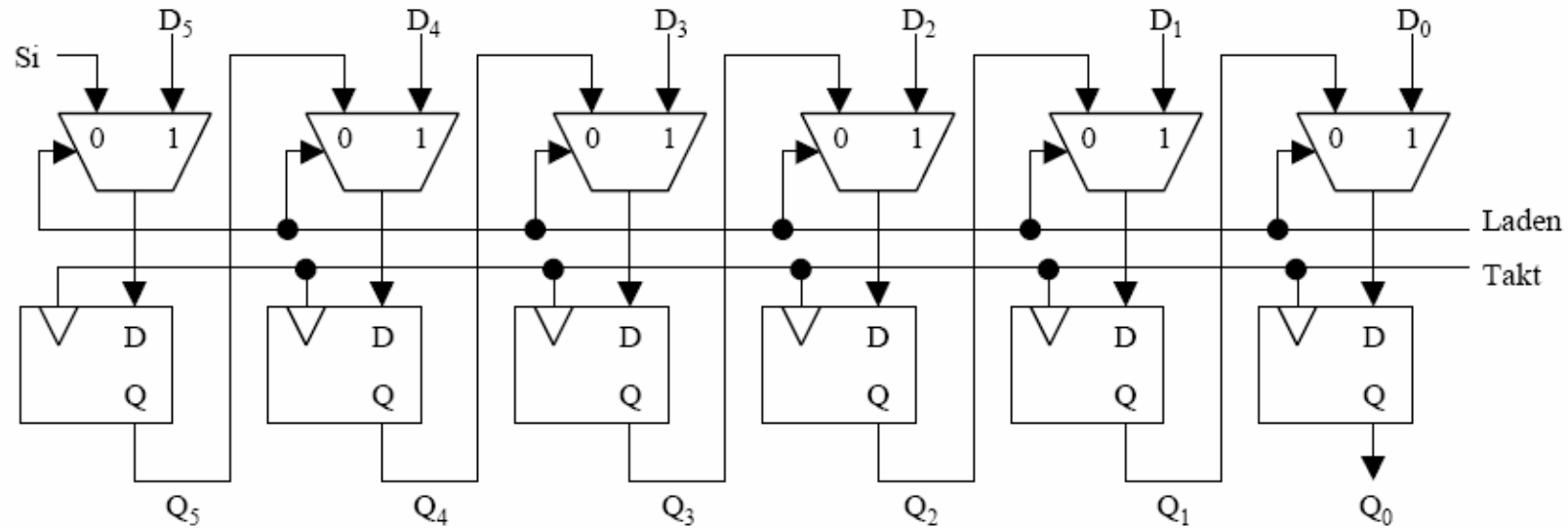


- Daten werden **parallel eingelesen** und **seriell ausgelesen**.
- Zum Auslesen müssen die Daten geshiftet werden.
- Für $X=0$ werden die Daten geshiftet, für $X=1$ können die Daten parallel eingelesen werden

Anwendung: beispielsweise Laden eines Datenworts und serielles Senden (z.B. Teil eines USART-Chips).

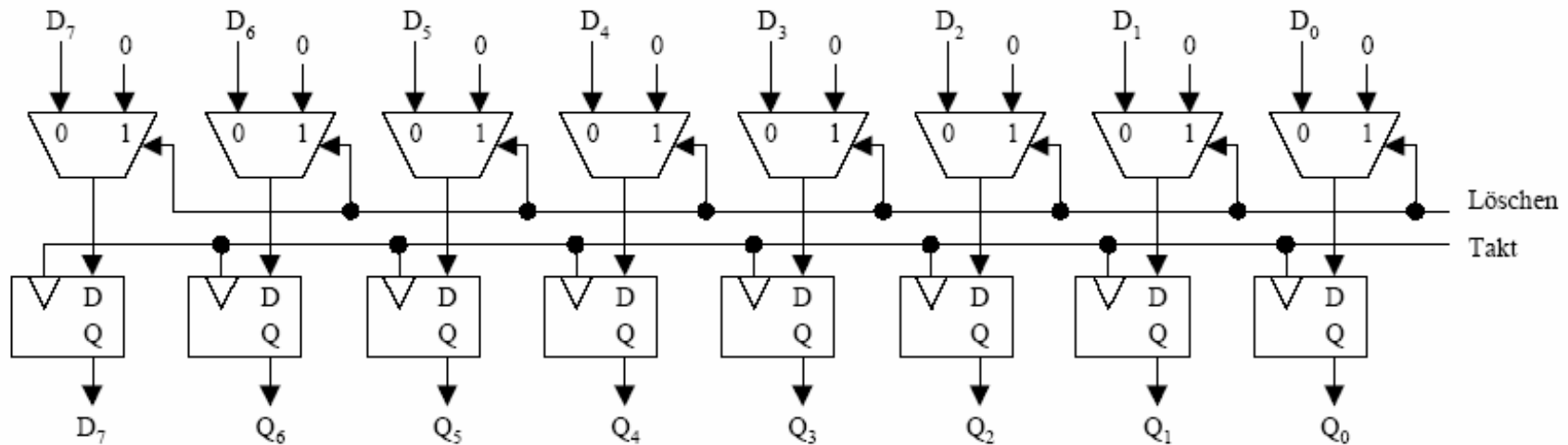
PISO = **P**arallel **I**n – **S**erial **O**ut

11. Schieberegister (PISO) (mit MUX)



PISO = Parallel In – Serial Out

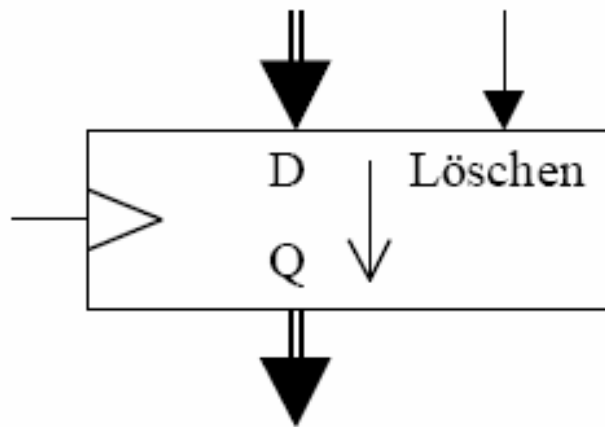
11. Register (PIPO) (synchron löschbar)



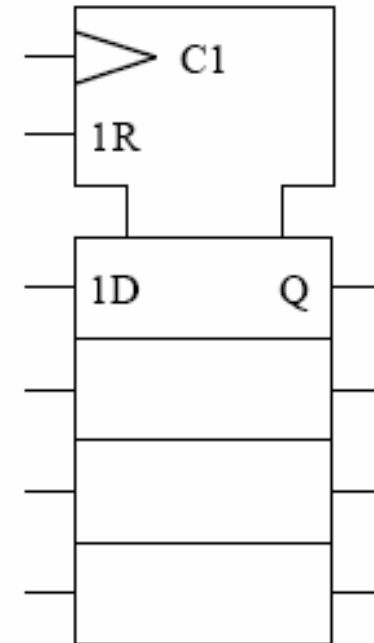
- Daten werden **parallel eingelesen** und **parallel ausgelesen**.
- Lesen erfolgt mit dem Takt (Löschen = 0).
- Löschen = 1. Mit dem nächsten Takt (synchron) werden alle Bits des Registers zurückgesetzt.

PIPO = **P**arallel **I**n – **P**arallel **O**ut

11. Register (PIPO) Schaltsymbole



Blockschaltbildsymbol



DIN-Symbol

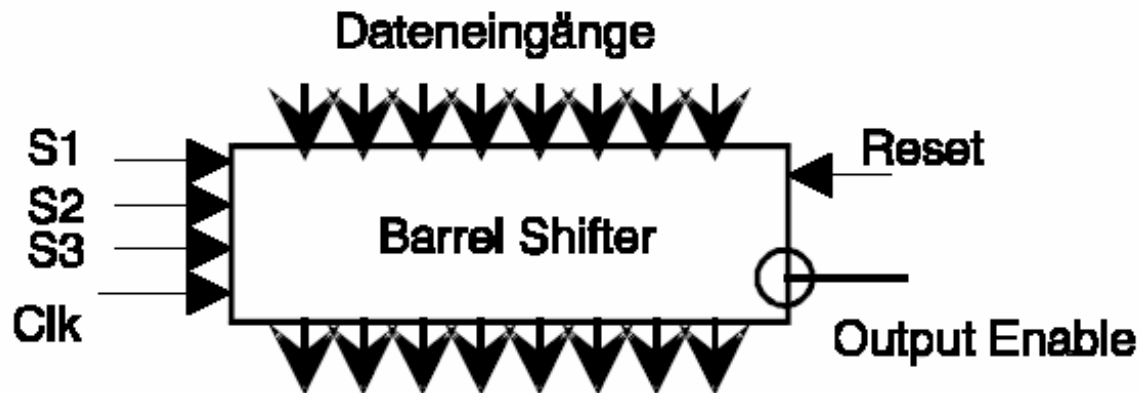
11. FIFO's, LIFO's und Shifter

Weiterhin gibt es Registertypen mit einer ganzen internen Speicherbank, auf die von zwei Seiten unabhängig zugegriffen werden kann, ohne äußere Synchronisierung der Zugriffe.

- **Fist-In – First Out:** Das FIFO gibt am Ausgang das zuerst aus, was auf der anderen Seite zuerst eingeschrieben wurde.
- **Last-In – First Out:** Das LIFO gibt das am Ausgang zuerst aus, was auf der anderen Seite als letztes eingeschrieben wurde.

11. FIFO's, LIFO's und Shifter (2)

- Ein spezialisiertes Register ist der sogenannte **Barrel Shifter**, der in einem Taktzyklus das Shiften (genauer rotieren) um eine beliebigen Anzahl von Bits erlaubt. Diese Anzahl wird über die Steuereingänge S_i festgelegt.



S1S2S3 Steuercode für die Schiebedistanz

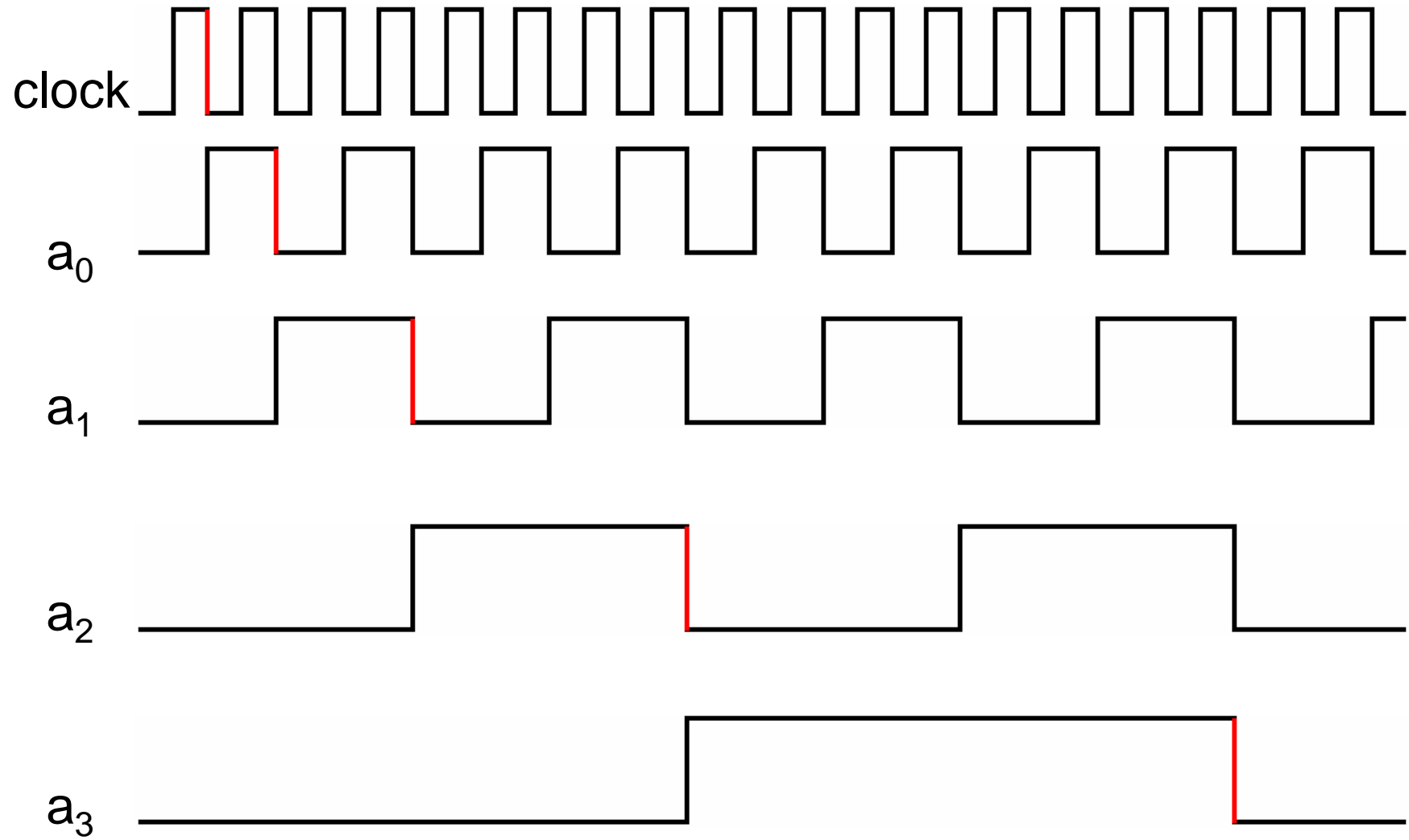
11. Zähler und Frequenzteiler

Zähler und Frequenzteiler werden sehr häufig in Anwendungen (Schaltungsrealisierungen) benötigt:

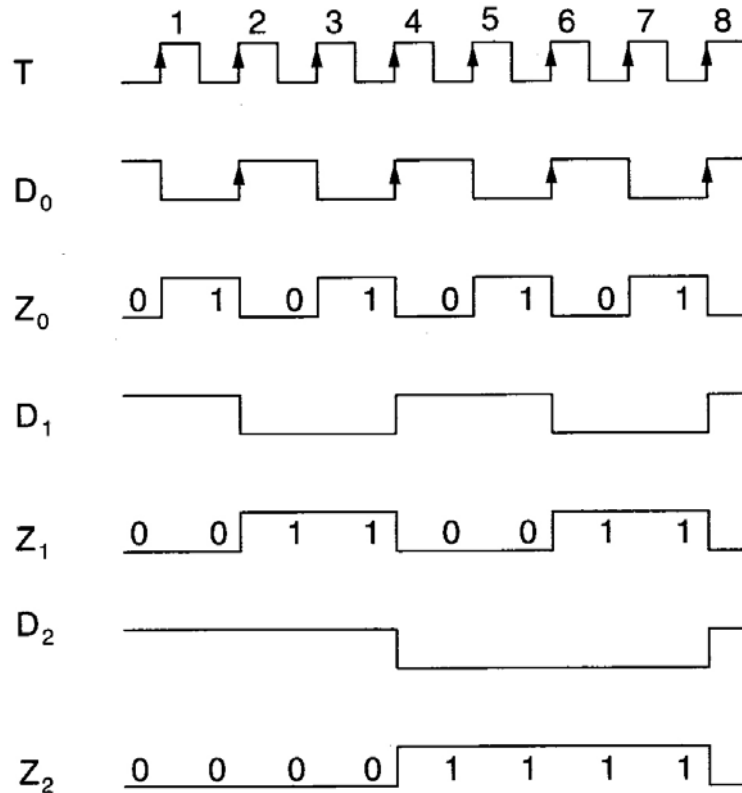
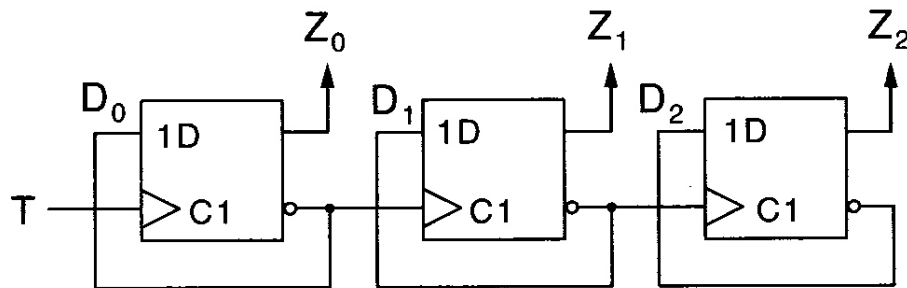
- Frequenzteiler: beispielsweise um aus dem Systemtakt (system clock) eines Computers den Takt für die Systemzeit (system time) zu gewinnen:
- Zähler: zum Beispiel Kilometerzähler im Auto, Umdrehungen der Trommel in einer Waschmaschine, Video-Zeilen im Sat-Receiver ...

Wir werden **asynchrone** und **synchrone** Zähler im Folgenden kennen lernen:

11. Frequenzteiler



11. Asynchrone Zähler

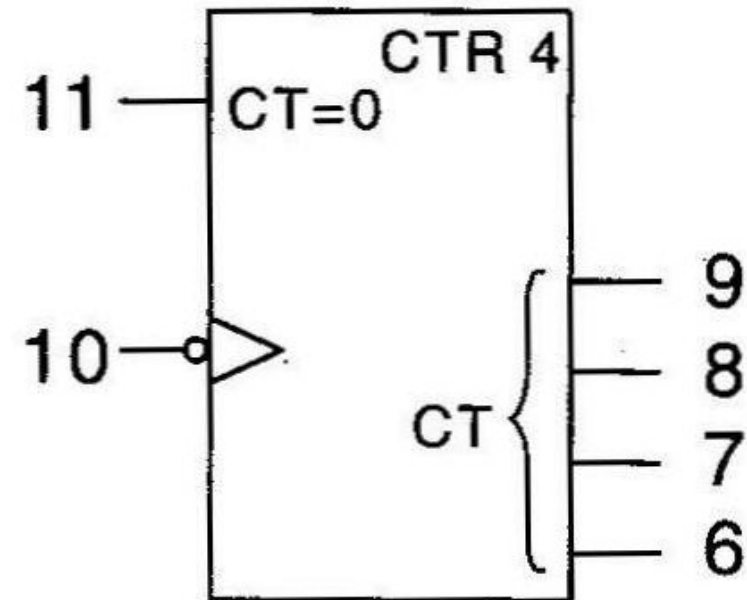
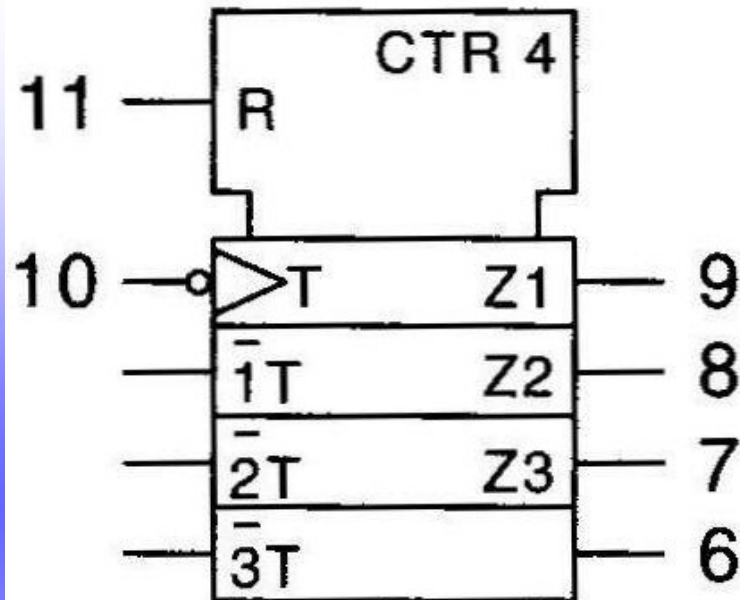


Durch Beschaltung von $\neg Q$ auf D wird ein T-FF realisiert.

FF 0 erzeugt für das nachfolgende FF 1 aus $\neg Q$ eine Taktflanke, ebenso FF 1 für FF 2.

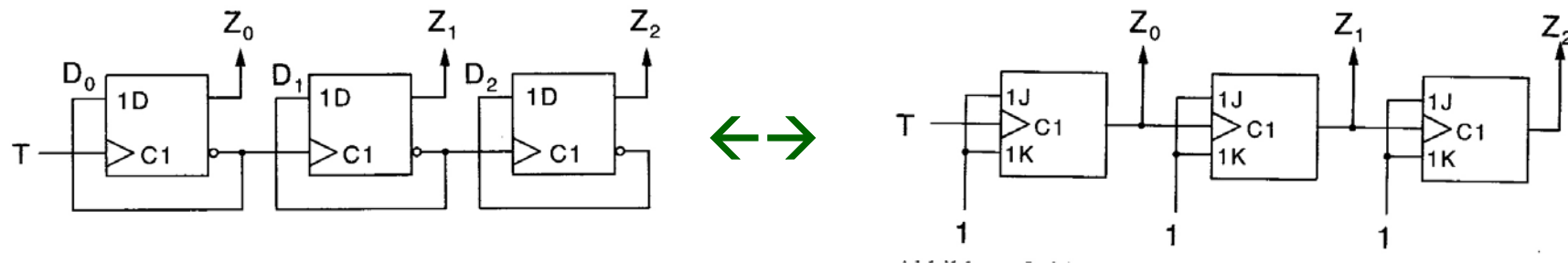
Teilungsverhältnis?
1 : 8

11. Asynchrone Zähler (2) Schaltzeichen



Anmerkung: 4-Bit Binärzähler tragen auch oft die Bezeichnung **CTRDIV16**, also Zähler, der durch 16 teilt

11. Asynchrone Zähler (3)

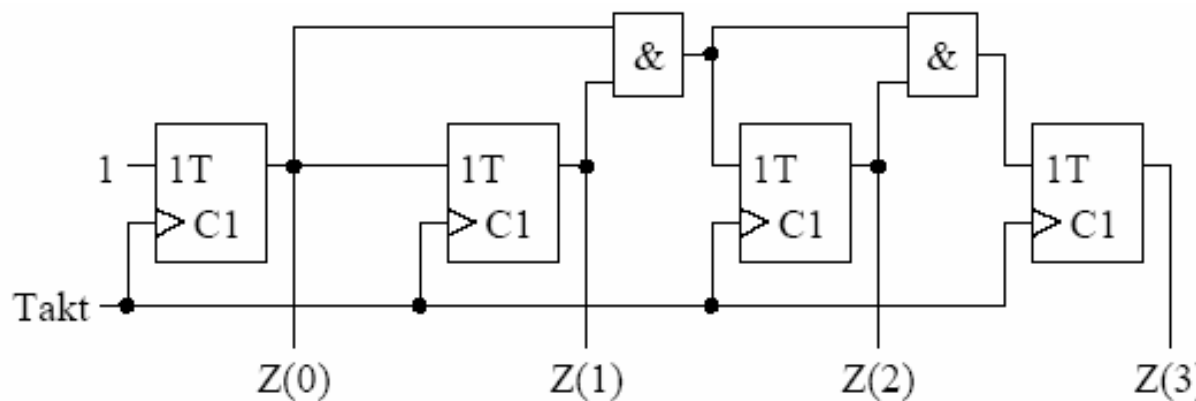


Beim asynchronen Zähler verbindet man den **Ausgang** des vorhergehenden Flip-Flops mit dem **Takteingang** des nachfolgenden Flip-Flops. Dann wird das nachfolgende Flip-Flop mit der halben Frequenz seines Vorgängers getaktet und wechselt somit auch nur halb so häufig seinen Zustandswert.

Man beachte die **Verzögerung**, die sich **von Ausgang zu Ausgang** ergibt. Sie resultiert daher, dass ein Ausgang als Taktsignal der nächsten Zählerstufe verwendet wird. Durch diese Verzögerungen wird am Ausgang nun **nicht sauber hochgezählt**, sondern an den Übergängen ergeben sich **kurzzeitig falsche Zählerwerte**.

11. Synchrone Zähler

Bei einem synchronen Zähler wird ein Zählimpuls allen Flip-Flops gleichzeitig zugeführt. Ein derartiger Zähler ist aber etwas aufwändiger als ein asynchroner Zähler.



$$\begin{aligned}
 T(0) &= 1 \\
 T(1) &= Z(0) \\
 T(2) &= Z(0) \wedge Z(1) \\
 &= T(1) \wedge Z(1) \\
 T(3) &= Z(0) \wedge Z(1) \wedge Z(2) \\
 &= T(2) \wedge Z(2)
 \end{aligned}$$

Wiederholung synchrones

Schaltwerk:

- Schaltnetz
- Speicherglieder

