Synchrone Zählschaltwerke

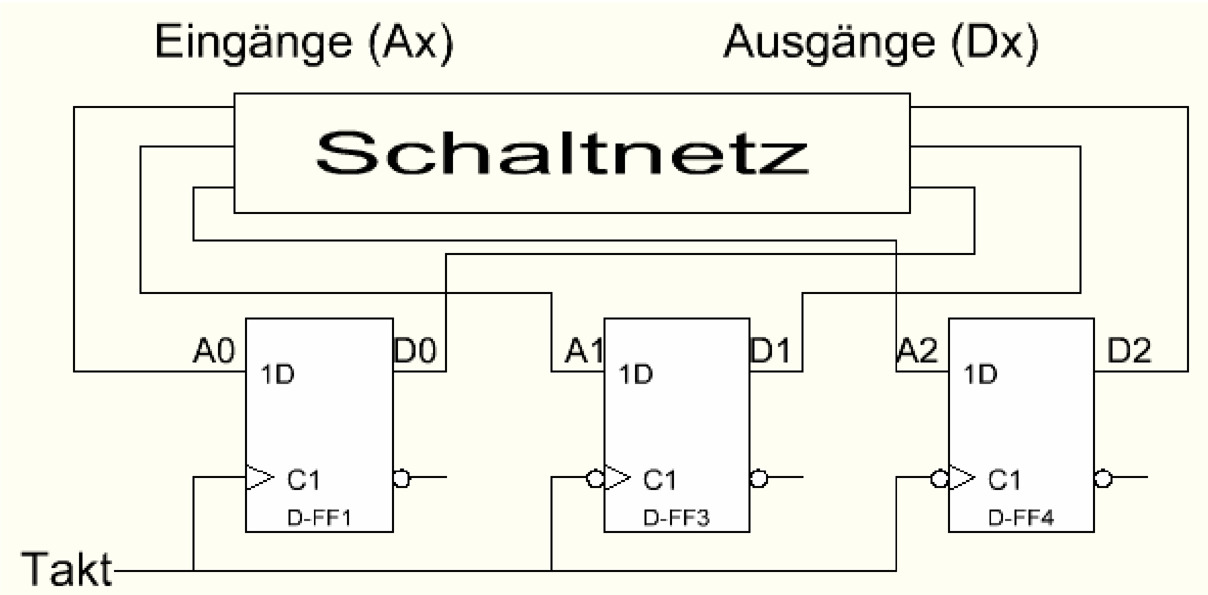
Beispiel:

- Entwicklung eines 3-Bit-Binär-Abwärtszählers
- Zählfolge: 7,6,5,4,3,2,1,0,7… (111 – 000)

Konzept:

- Zählerstand wird im Zustandsspeicher abgelegt
- Zustandszähler besteht aus 3 Flipflops, je ein Flipflop pro Binärzähler
- Schaltnetz sorgt dafür, dass vom aktuellen Zählerstand zur nächsten Ziffer weiter gezählt wird

Skizze:



Merke: Synchronzähler, weil alle Flipflops mit dem <u>gleichen</u> Taktsignal versorgt werden.

<u>Konstruktion des Schaltnetzes</u>
(Bemerkung: Der Takt ist kein Informationsträger und wird im Weiteren nicht betrachtet.)

Funktionstabelle:

x = Entsprechende Binärstelle

A_x = Eingestellter Wert, der auch angezeigt wird und in das Schaltnetz läuft

D_x = Neuer Zustand, der nach dem Durchlaufen des Flipflops erreicht wird

Zahl	A_2	A_1	A_0	D_2	D_1	D_0
0	0	0	0	1	1	1
1	0	0	1	0	0	0
2	0	1	0	0	0	1
3	0	1	1	0	1	0
4	1	0	0	0	1	1
5	1	0	1	1	0	0
6	1	1	0	1	0	1
7	1	1	1	1	1	0

KV – Diagramm für D_0

		A_1	A_1	
	1	1	1	1
A_0	0	0	0	0
			A_2	A_2

$$D_0 = \overline{A_0}$$

KV – Diagramm für D_1

		A_1	A_1	
	1	0	0	1
A_0	0	1	1	0
			A_2	A_2

$$D_1 = \left(\overline{A_0} \cdot \overline{A_1}\right) + \left(A_0 \cdot A_1\right) = \left(A_0 \equiv A_1\right)$$

KV – Diagramm für D_2

		A_1	A_1	
	1	0	1	0
A_0	0	0	1	1
			A_2	A_2

$$D_2 = \left(A_0 \cdot A_2\right) + \left(A_1 \cdot A_2\right) + \left(\overline{A_0} \cdot \overline{A_1} \cdot \overline{A_2}\right)$$

Veränderung der Zählfolge:

Zählfolge: 0,1,3,5,3,2,0…

Man muss in diesem Fall, weil die 3 doppelt vorkommt, einen Ersatzcode finden, indem die 3 z.B. durch eine 4 ersetzt wird. Diese 4 wird dann durch einen Codekonverter, der hinter das Dekodierschaltnetz geschaltet wird, wieder zu einer 3, die dann ausgegeben wird.

Entwurf eines umschaltbaren Zählers

Eine Eingangsvariable X

X = 0: Zählfolge 0,1,2,3,0…

X = 1: Zählfolge 3,2,1,0,1…

(Umschaltbarer Vorwärts/Rückwärts – Zähler der im Binärcode zählt)

Funktionstabelle:

Zahl	X	A_1	A_0	D_1	D_0
0	0	0	0	0	1
1	0	0	1	1	0
2	0	1	0	1	1
3	0	1	1	0	0
4	1	0	0	1	1
5	1	0	1	0	0
6	1	1	0	0	1
7	1	1	1	1	0

Für D_0 gilt wie im ersten Beispiel aufgrund der unveränderten Positionen der Einsen in der Funktionstabelle:

$$D_0 = \overline{A_0}$$

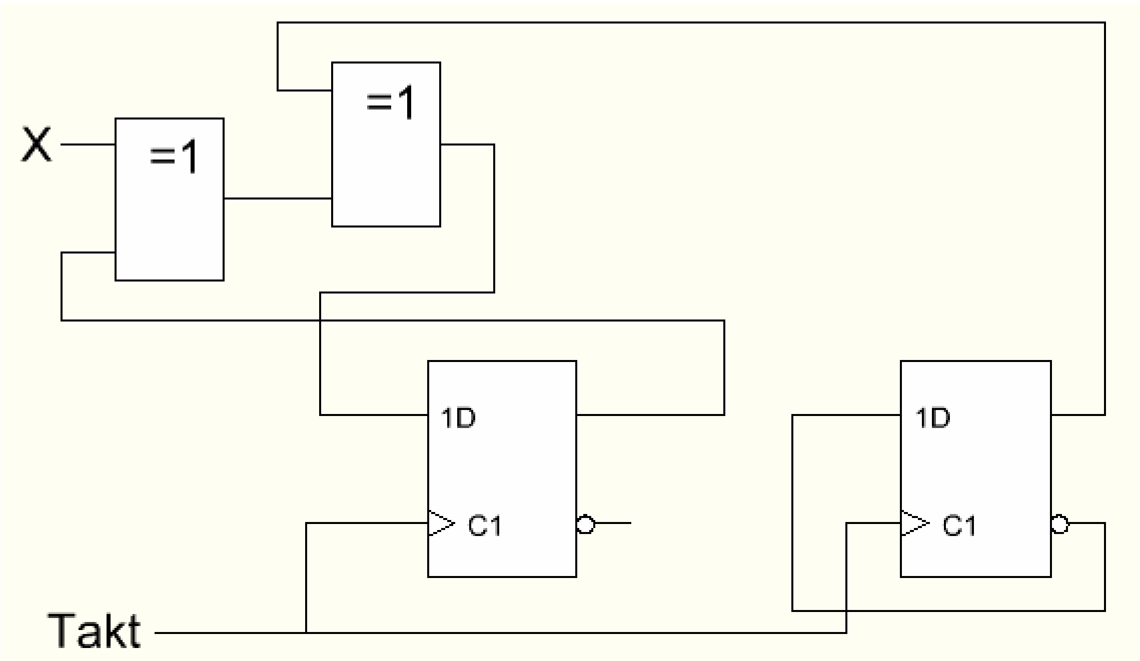
KV – Diagramm:

		A_0		
	0	1	0	1
X	1	0	1	0
			A_1	

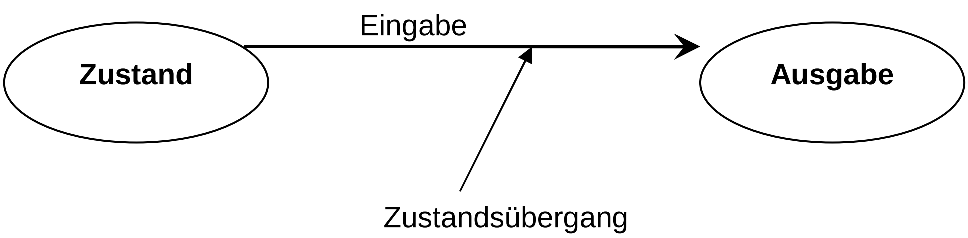
Da die Minimierung mit dem KV – Diagramm nicht gemacht werden kann, wird nach den Regeln der Boolschen Algebra minimiert.

$$
\begin{aligned}
D_1 &= \left(X \cdot \overline{A_0} \cdot \overline{A_1} \right) + \left(\overline{X} \cdot A_0 \cdot \overline{A_1} \right) + \left(X \cdot A_0 \cdot A_1 \right) + \left(\overline{X} \cdot \overline{A_0} \cdot \overline{A_1} \right) \\
&= X \cdot \left(\left(\overline{A_0} \cdot \overline{A_1} \right) + \left(A_0 \cdot A_1 \right) \right) + \overline{X} \cdot \left(\left(A_0 \cdot \overline{A_1} \right) + \left(\overline{A_0} \cdot A_1 \right) \right) \\
&= X \cdot \left(A_0 \equiv A_1 \right) + \overline{X} \cdot \left(A_0 \neq A_1 \right) \\
&= \left(X \cdot \overline{k} \right) + \left(\overline{X} \cdot k \right) \qquad\qquad k = \left(A_0 \neq A_1 \right) \\
&= X \neq k \\
&= X \neq A_0 \neq A_1
\end{aligned}
$$

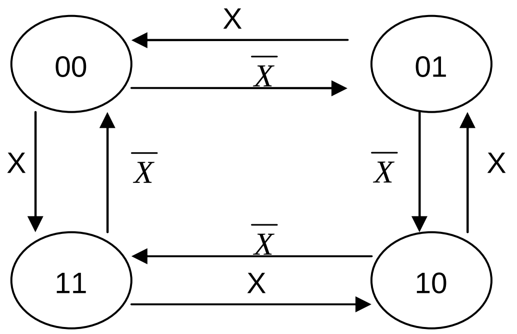
Logikplan:



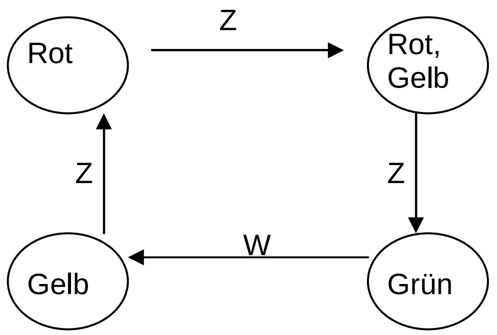
Zustandsübergangsdiagramm:



Am Beispiel Auf/Ab – Zähler



Zustandsübergangsdiagramm für eine Ampel



W = Warteknopf betätigt

Z = Zeit abgelaufen

1. Schritt: Zustandsübergangsdiagramm
2. Schritt: Zustandscodierung → Binärwerte für Zustände entwickeln
(Bsp.: Grün = 001, Gelb = 010, Rot = 100, Rot-Gelb = 110)