SCHALTWERKE (13)

[04.06.02, Folie 481, Übungen 07]

Schaltwerke sind wesentliche Funktionseinheiten eines Computers. Beispiele hierfür sind das Rechen- und das Leitwerk eines von-Neumann-Rechners [Duden: **Rechenwerk/ALU** Ausführung von arithmetischen Operationen; **Leitwerk/Steuerwerk** zentrale Komponente der Zentraleinheit (Laden der Programmbefehle, Decodierung, Interpretation, Versorgung der an der Ausführung der Befehle beteiligten Funktionseinheiten mit den nötigen Steuersignalen)]
Synchrone Schaltwerke bestehen aus einem Schaltnetz und Speichergliedern. Wichtig für ein Schaltwerk ist die funktionelle Bedeutung der Zeit: Der Zeitpunkt, in dem das Schaltwerk betrachtet wird, ist entscheidend für seinen Zustand ...

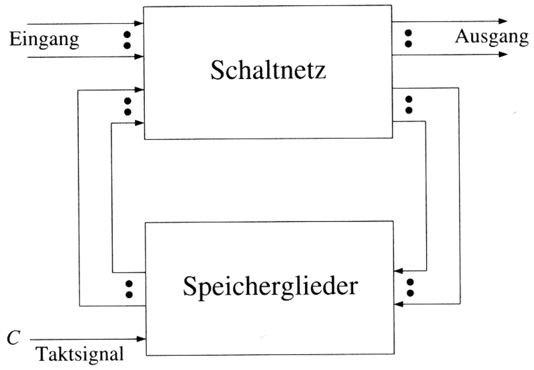


Abb. 77: Prinzipieller Aufbau eines Synchron-Schaltwerkes

Die Zeitpunkte werden durch ein Taktsignal vorgegeben. Speicherglieder können taktabhängig einen stabilen Zustand einnehmen und speichern. Der zu einem bestimmten Zeitpunkt gespeicherte Zustand der Speicherglieder heißt *innerer Zustand des Schaltwerks.*
Es gibt zwei Typen von Schaltwerken ...
- **Synchrone Schaltwerke:** Hier erfolgt der Übergang von einem stabilen Zustand in einen stabilen Folgezustand synchron mit dem Taktsignal.
- **Asynchrone Schaltwerke:** Der Übergang zwischen Zuständen erfolgt hier asynchron (ohne Taktsignal).

Wir betrachten im folgenden nur synchrone Schaltwerke.
Nach DIN40300/89 ist ein Schaltwerk eine Funktionseinheit zum Verarbeiten von Schaltvariablen, wobei der Wert am Ausgang zu einem bestimmten Zeitpunkt abhängt von den Werten am Eingang zu diesem und endlich vielen vorangegangenen Zeitpunkten.

Automaten (13.1)

Schaltwerke können mit Hilfe von *endlichen Automaten* beschrieben werden. Ein Automat in der theoretischen Informatik ist ein Modell, das die Zustände, Ein- und Ausgaben eines Systems (z.B. eines Schaltwerks) beschreibt. Ein Automat heißt endlich, wenn sein Eingabealphabet, sein Ausgabealphabet und die Zustandsmenge alle endlich sind.
Formal wird ein endlicher Automat M beschrieben als ...
$M=(X,Z,Y,z_0,F,f,g)$
Dabei ist ...
- $X=x_1,\ldots,x_n$ das Eingabebalphabet
- $Y=y_1,\ldots,y_n$ das Ausgabealphabet [Fehler im Skript korrigiert (Schiffmann256)]
- $Z=z_1,\ldots,z_n$ die Zustandsmenge
- $z_0 \in Z$ der Anfangszustand
- $F \subseteq Z$ die Menge der Endzustände
- g: $(x_i,z_j) \rightarrow z_k$ die Übergangsfunktion
- f: $(x_i,z_j) \rightarrow y_r$ die Ausgangsfunktion

[(Schiffmann256) x_n, y_m, z_m]
Es gilt ...
- Jedes Schaltwerk läßt sich als endlicher Automat darstellen.
- Jeder endliche Automat läßt sich in ein Schaltwerk umsetzen.

Allen Schaltwerken gemeinsam ist die Rückkopplung zwischen Speichergliedern und Schaltnetz ...
- Der Ausgang der Speicherglieder wirkt auf das Schaltnetz.
- Ein Teil der Ausgänge des Schaltnetzes wirkt auf die Speicherglieder.
- Der Eingang des Schaltwerkes heißt Eingangsvektor X.
- Der Ausgang entsprechend Ausgangsvektor Y.

- Der Vektor der Zustände der Speicherglieder heißt Zustandsvektor Z.
- Die Rückkopplung Schaltnetz→Speicherglieder wird durch die Schaltfunktion $g(X,Z(t_n))$ gebildet.

Mealy-Automat

Dies ergibt den Aufbau eines *Mealy-Automaten* (Abb. 78) ...

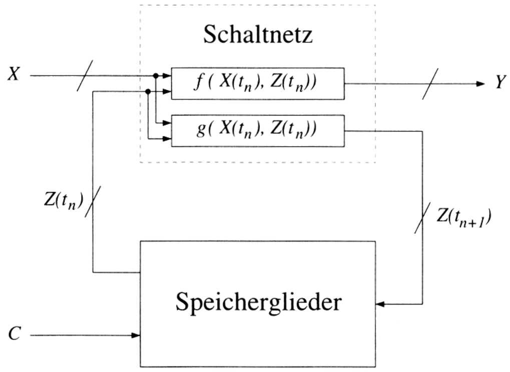


Abb. 78: Struktur eines Mealy-Automaten

Im Taktintervall t_n werden die Schaltfunktionen $f(X(t_n),Z(t_n))$ und $g(X(t_n),Z(t_n))$ gebildet ...

- Der Wert der Schaltfunktion f steht als *Ausgangsvektor* $Y=f(X(t_n),Z(t_n))$ zur Verfügung.
- Der Wert der Funktion g wird erst im folgenden Zeitintervall t_{n+1} wirksam und heißt daher *Folgezustandsvektor* $Z(t_{n+1})$.
- Die Funktion $g(X(t_n),Z(t_n))$ heißt *Übergangsfunktion*. Es gilt $Z(t_{n+1})=g(X(t_n),Z(t_n))$.
- Beim Übergang vom Zeitintervall $t_n \rightarrow t_{n+1}$ wird der Folgezustandsvektor zum neuen Zustandsvektor $Z(t_n):=Z(t_{n+1})$.

Der Mealy-Automat wird durch die Schaltfunktion ...

$Y(t_n)=f(X(t_n),Z(t_n))$ Ausgangsfunktion
$Z(t_{n+1})=g(X(t_n),Z(t_n))$ Übergangsfunktion [Fehler im Skript korrigiert]

... und die Zuweisung ...

$Z(t_n):=Z(t_{n+1})$

... bestimmt.

Es gelten folgende Zeitbedingungen ...

- Während der Taktpause (C = 0) des Zeitinvervalls t_n werden die Schaltfunktionen f und g gebildet. Die Taktpause muß größer sein als alle Signallaufzeiten durch das Schaltnetz g. Dann sind alle Hazards im Schaltnetz abgelaufen und der Folgezustandsvektor $Z(t_{n+1})$ ist stabil.
- $Z(t_{n+1})$ wird mit der steigenden Taktflanke (C=0→1) in die Master-Slave-Flipflops übernommen. Hierzu werden zuerst die Slave-Flipflops gesperrt, dann die Werte in die Master-Flipflops übernommen. Die Master-Slave-Flipflops stellen sicher, daß nicht während des Zeitintervalls t_n eine Komponente $z_i(t_{n+1})$ des Folgezustandsvektors an den Ausgang gelangt, den Zustandsvektor $Z(t_n)$ ändert, und über das Schaltnetz g wieder eine Komponente $z_j(t_{n+1})$ ändert.
- Mit der fallenden Flanke des Taktsignals (C=1→0) werden zuerst die Master-Flipflops gesperrt. Dann wird der Folgezustandsvektor $Z(t_{n+1})$ in die Slave-Flipflops übemommen und wird damit zum neuen Zustandvektor $Z(t_n)$.

Moore-Automat

Ein Sonderfall des Mealy-Automaten ist der *Moore-Automat*. Bei diesem Typ von Automat ist der Ausgangsvektor Y nicht vom Eingang, sondern nur von der Zustandsfunktion $f(Z(t_n))$ abhängig (siehe Abb. 79) ...

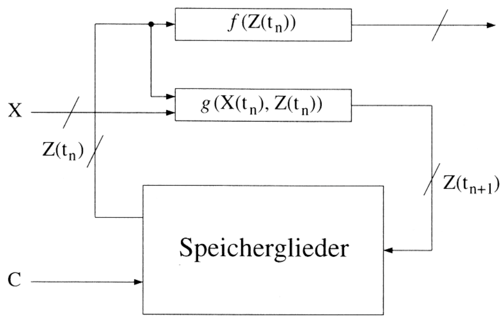


Abb.79: Moore-Automat

Ein Moore-Automat wird durch die Schaltfunktion ...

$Y(t_n)=f(Z(t_n))$ Ausgangsfunktion
$Z(t_{n+1})=g(X(t_n),Z(t_n))$ Übergangsfunktion

... beschrieben. Man sieht, daß der Eingangsvektor $X(t_n)$ keinen Einfluß auf den Ausgangsvektor $Y(t_n)$ hat. Der Eingangsvektor wirkt erst um eine Taktperiode verzögert auf den Ausgangsvektor. Es gilt ...

$Y(t_{n+1})=f(Z(t_{n+1}))=f(g(X(t_n),Z(t_n)))$

Beispiele für Moore-Automaten sind synchrone Zähler. Dabei ist der Zählzustand der Ausgangszustand und

gleichzeitig der innere Zustand des Schaltwerks. Der Eingangsvektor kann z.B. ein Signal zur Umschaltung der Zählrichtung sein. Schaltwerke, die außer dem Taktsignal kein Eingangssignal haben, nennt man *autonom* (autonome Automaten).
Für eine vereinfachte Notation des Zustandsvektors, des Folgezustandsvektors und deren Komponenten soll gelten ...
$Z:=Z(t_n)$ (in Komponenten $z_0, z_1, ...$)
$Z^+:=Z(t_{n+1})$ (in Komponenten $z_0^+, z_1^+, ...$) [Fehler im Skript korrigiert]

Funktionelle Beschreibung von Schaltwerken (13.2)

Es gibt auch bei Schaltwerken verschiedene äquivalente Beschreibungsmöglichkeiten. Die hierfür gebräuchlichsten sind ...
- Zustandsfolgetabellen
- KV-Diagramme
- Schaltfunktionen oder Vektorfunktionen
- Zustandsgraphen

Eine *Zustandsfolgetabelle* enthält bei Schaltwerken als ...
- Eingangsvariablen die Komponenten des Eingangsvektors X und des Zustandsvektors $Z(t_n)$
- Ausgangsvariablen die Komponenten des Ausgangsvektors Y und des Folgezustandsvektors $Z(t_{n+1})$

Eingangsvariablen		**Ausgangsvariablen**	
$z_0 ... z_n$	$x_0 ... x_i$	$z_0^+ ... z_n^+$	$y_0 ... y_n$
...	...	...	...

[(Schiffmann260) $y_0 ... y_j$]
Für jede Ausgangsvariable [aus der Zustandsfolgetabelle] kann ein *KV-Diagramm* erstellt werden, aus dem eine minimierte Schaltfunktion gebildet werden kann.
Ein *Zustandsgraph* beschreibt das Verhalten eines Schaltwerkes graphisch. Er besteht aus Knoten und gerichteten Kanten ...

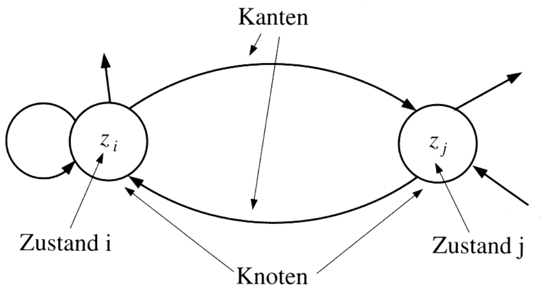


Abb. 80: Grundform eines Zustandsgraphen
- **Knoten:** Knoten stellen die inneren Zustände des Schaltwerks dar. In den Knoten stehen Zustandsnamen oder die zugehörigen Kombinationen der Zustandsvariablen.
- **Kanten:** Kanten stellen die Übergänge zwischen den Zuständen dar. An die Karten werden die Werte des Eingangsvektors geschrieben, die das Schaltwerk zum Folgezustand überführen.
  Beim Mealy-Automat schreibt man an die Kante hinter den Eingangsvektor zusätzlich auch noch den zum aktuellen Zustand gehörigen Ausgangsvektor.
  Beim Moore-Automaten ist der Ausgangsvektor eindeutig durch den momentanen Zustand (Knoten) bestimmt.
- Bei beiden Automatentypen wird der Takteingang nicht angegeben, da er kein Informationsträger ist.
- Eine Kante mit direkter Rückkopplung gibt an, daß sich bei der entsprechenden Eingabe der Zustand nicht ändert.

Ein Zustandsgraph eines Schaltwerkes mit n Variablen kann höchstens 2^n Knoten enthalten. Hat der Eingangsvektor eines Schaltwerkes m Variablen, dann können höchstens 2^m Kanten von jedem Knoten ausgehen.

Analyse von Schaltwerken (13.3)

Ein Schaltwerk zu analysieren bedeutet, sein Schaltverhalten durch ...
- Eine Zustandsfolgetabelle
- Die Schaltfunktionen (Übergangsfunktion, Ausgangsfunktion)
- Einen Zustandsgraphen

... zu beschreiben.
Die allgemeine Vorgehensweise ist wie folgt ...
(1) Aus dem Schaltnetz des Schaltplans werden die Übergangsfunktion und die Ausgangsfunktion hergeleitet.
(2) Dann wird ein Anfangszustand angenommen und mit den Werten der Eingangsvariablen und den Übergangsfunktionen der Folgezustand hergeleitet.

(3) Schritt (2) wird für alle Anfangszustände wiederholt. Dadurch entsteht eine Zustandsfolgetabelle.
(4) Aus der Zustandsfolgetabelle kann der Zustandsgraph gezeichnet werden.

Schaltwerksanalyse: Beispiel 1 (13.3.1)

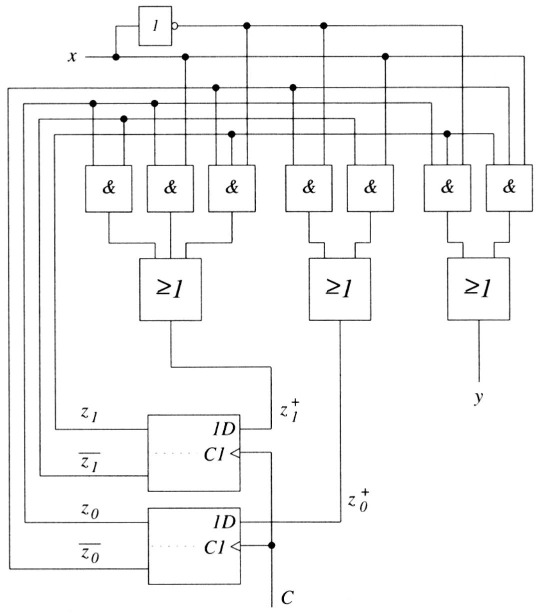


Abb. 81: Schaltwerk mit D-Flipflops

Im obigen Beispiel 1 erkennt man zuerst folgendes ...

- Es ist ein synchrones Schaltwerk.
- Der Eingangsvektor X und der Ausgangsvektor Y bestehen nur aus je einer Variablen.
- Das Schaltwerk hat zwei vorderflankengesteuerte D-Flipflops als Speicherglieder und hat daher zwei Zustandsvariablen. Es kann maximal 4 verschiedene Zustände einnehmen.
- Die Komponenten z_0^+ und z_1^+ des Folgezustandsvektors $Z(t_{n+1})$ werden durch ein Schaltnetz aus dem Eingangsvektor X und den Komponenten z_0^n und z_1^n des Zustandsvektors $Z(t_n)$ gebildet.
- Der Ausgangsvektor Y wird aus dem Eingangsvektor X und den Komponenten des Zustandsvektors $Z(t_n)$ gebildet. Daraus folgt, daß das Schaltwerk ein Mealy-Automat ist.

Aus der Analyse des Schaltnetzes von Beispiel 1 folgt als Übergangsfunktion ...

$z_0^+=(\overline{z_0}\wedge\overline{x})\vee(\overline{z_1}\wedge x)$ (15)

$z_1^+=(z_0\wedge\overline{z_1})\vee(z_0\wedge x)\vee(\overline{z_0}\wedge z_1\wedge\overline{x})$ (16)

Für den Ausgangsvektor Y ...

$y=(z_0\wedge z_1\wedge\overline{x})\vee(\overline{z_0}\wedge z_1\wedge x)$ (18)

Mit diesen Schaltfunktionen kann die Zustandsfolgetabelle erstellt werden. Für $z_0=0$, $z_1=0$ und Eingabe x=0 ergibt sich nach (15), (16) und (18), daß $z_0^+=1$, $z_1^+=0$ und y=0. Analog berechnet man alle übrigen Elemente der Tabelle. Die komplette Zustandsfolgetabelle ist in Tab. 14 angegeben ...

z_1	z_0	x	z_1^+	z_0^+	y
0	0	0	0	1	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	1	1	0
1	0	0	1	1	0
1	0	1	0	0	1
1	1	0	0	0	1
1	1	1	1	0	0

Tab. 14. Zustandsfolgetabelle für Beispiel 1

[$z_1^+=(\overline{z_1}\wedge z_0)\vee(z_0\wedge x)\vee(z_1\wedge\overline{z_0}\wedge\overline{x})$

$z_0^+=(\overline{z_0}\wedge\overline{x})\vee(\overline{z_1}\wedge x)$

$y=(z_1\wedge z_0\wedge\overline{x})\vee(z_1\wedge\overline{z_0}\wedge x)$]

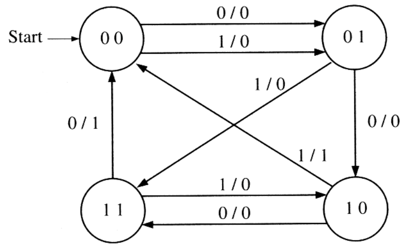


Abb. 82: Zustandsgraph für Beispiel 1
[x / y an den Kanten]

Schaltwerksanalyse: Beispiel 2 (13.3.2)

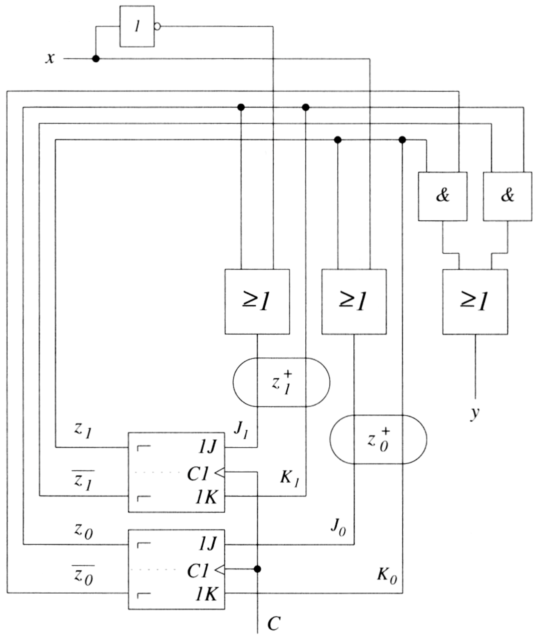


Abb. 83: Schaltwerk mit JK-Flipflops

Auch dieses Schaltwerk kann max. 2^2=4 Zustände einnehmen. Hier werden die Komponenten z_0^+ und z_1^+ nicht direkt an die Speicherglieder geführt werden, sondern als getrennte Vorbereitungseingänge J und K.
Die Analyse des Schaltnetzes führt zu den Schaltfunktionen ...

$J_0 = x \vee z_1$
$K_0 = z_1$
$J_1 = \overline{x} \vee z_0$
$K_1 = z_0$
$y = (\overline{z_1} \wedge z_0) \vee (z_1 \wedge \overline{z_0})$

Der Ausgangsvektor y enthält nur die Variablen des Zustandsvektors Z, das Schaltwerk ist also ein Moore-Automat. Damit ergibt sich folgende Zustandsfolgetabelle ...

z_1	z_0	x	K_1	J_1	K_0	J_0	z_1^+	z_0^+	y
0	0	0	0	1	0	0	1	0	0
0	0	1	0	0	0	1	0	1	0
0	1	0	1	1	0	0	1	1	1
0	1	1	1	1	0	1	1	1	1
1	0	0	0	1	1	1	1	1	1
1	0	1	0	0	1	1	1	1	1
1	1	0	1	1	1	1	0	0	0
1	1	1	1	1	1	1	0	0	0

Tab. 15. Zustandsfolgetabelle für Beispiel 2
[z_1^+ über K_1,J_1 und vorherigen Zustand z_1]

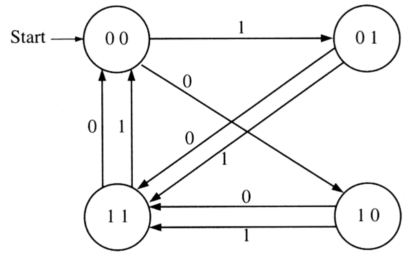


Abb. 84: Zustandsgraph für Beispiel 2

β Synthese von Schaltwerken (13.4)

[11.06.02, Folie 505, Übungen 08]

Die *Synthese von Schaltwerken* bedeutet, aus einer verbalen Aufgabenstellung ein Schaltwerk zu entwerfen. Dazu ist es notwendig, die Aufgabenstellung in einen endlichen Automaten umzuwandeln, der als Schaltwerk realisiert werden kann.
Es empfiehlt sich folgendes Vorgehen ...
(1) Festlegen der Zustandsmenge, die das Schaltwerk einnehmen soll. Daraus ergibt sich die Anzahl der Zustandsvariablen und die Anzahl der erforderlichen Speicherglieder.
(2) Festlegen des Anfangszustandes.
(3) Definition der Eingangs- und Ausgangsvariablen.
(4) Darstellung der Zustandsfolge in Form eines Zustandsgraphen.
(5) Aufstellen der Zustandsfolgetabelle.
(6) Herleitung und Minimierung der Übergangsfunktion und der Ausgangsfunktion in DNF oder KNF aus der Zustandsfolgetabelle.
(7) Darstellung des Schaltwerks in einem Schaltplan ...
- Übertragen der Schaltfunktionen in ein Schaltnetz
- Zeichnen der Speicherglieder durch Flipflops
- Kennzeichnen des Zustandsvektors und des Folgezustandsvektors
(8) Technische Realisierung des Schaltwerks (in der Praxis).

γ Schaltwerkssynthese: Beispiel 1 (Umschaltbarer Zähler) (13.4.1)

Aufgabe
Entwicklung eines zweistelligen umschaltbaren Gray-Code-Zählers, mit der Zählfolge ...
- Für x=0 gelte die Zählfolge 00, 01, 11, 10
- Für x=1 gelte die Zählfolge 00, 10, 11, 01.

Vorgehen
(1) Wir benötigen 4 Zustände, d.h. zwei Zustandsvariablen z_0,z_1.
(2) Der Anfangszustand sei 00.
(3) Die Eingangsvariable ist x, die Ausgangsvariablen sind identisch mit den Zustandsvariablen, weil der Zählzustand angezeigt werden soll. Es handelt sich also um einen Moore-Automat.
(4) Der Zustandsgraph sieht wie folgt aus ...

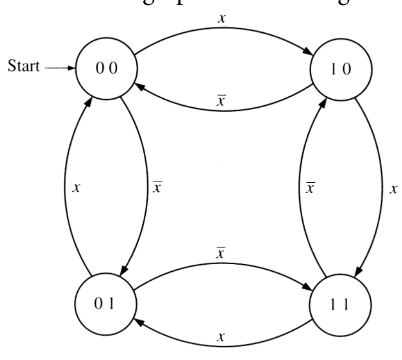


Abb.85: Zustandsgraph

[$\bar{x} \triangleq x=0$]
(5) Die Zustandsfolgetabelle lautet ...

z_1	z_0	x	z_1^+	z_0^+
0	0	0	0	1
0	0	1	1	0
0	1	0	1	1
0	1	1	0	0

1	0	0	0	0
1	0	1	1	1
1	1	0	1	0
1	1	1	0	1

Tab. 16. Zustandfolgetabelle für Gray-Code Zähler

(6) Hieraus kann man die Übergangsfunktion in DNF bestimmen ...

$z_1^+=(z_0\wedge\overline{x})\vee(\overline{z_0}\wedge x)$

$z_0^+=(\overline{z_1}\wedge\overline{x})\vee(z_1\wedge x)$

[Direkt in Tabelle 1-Paare zusammenfassen; Entspricht schon der DMF]

(7) Das Schaltwerk hat folgende Struktur ...

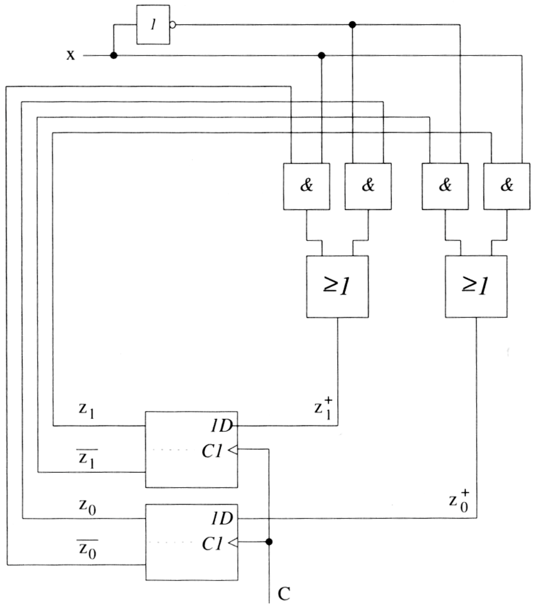


Abb. 86: Umschaltbarer Gray-Code-Zähler

Schaltwerkssynthese Beispiel 2 (Schieberegister) (13.4.2)

n.Takt		Zustände nach Takt n			
	x	z_0	z_1	z_2	y
0.	1	0	0	0	0
1.	1	1	0	0	0
2.	0	1	1	0	0
3.	0	0	1	1	0
4.	0	0	0	1	1
5.	0	0	0	0	1
6.	0	0	0	0	0

Tab. 17: Zustandsfolgetabelle für Schieberegister

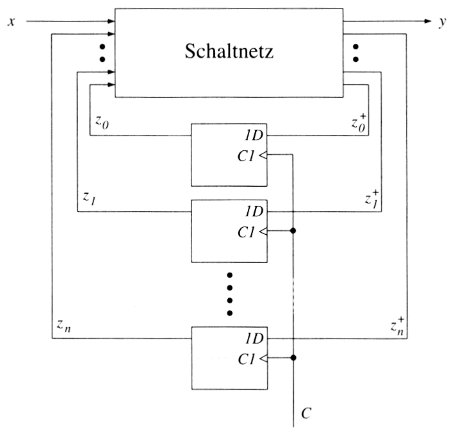


Abb. 87: Allgemeine Schaltwerksstruktur für ein Schieberegister

x	z_2	z_1	z_0	z_2^+	z_1^+	z_0^+	y
0	0	0	0	0	0	0	0

0	0	0	1	0	1	0	0
0	0	1	0	1	0	0	0
0	0	1	1	1	1	0	0
0	1	0	0	0	0	0	1
0	1	0	1	0	1	0	1
0	1	1	0	1	0	0	1
0	1	1	1	1	1	0	1
1	0	0	0	0	0	1	0
1	0	0	1	0	1	1	0
1	0	1	0	1	0	1	0
1	0	1	1	1	1	1	0
1	1	0	0	0	0	1	1
1	1	0	1	0	1	1	1
1	1	1	0	1	0	1	1
1	1	1	1	1	1	1	1

Tab. 18. Erweiterte Zustandsfolgetabelle für ein Schieberegister

Aus der Zustandsfolgetabelle ergeben sich die Übergangsfunktionen ...

$z_0^+=x$

$z_1^+=z_0$

$z_2^+=z_1$

... und die Ausgangsfunktion ...

$y=z_2$

Werden JK-Flipflops als Speicherglieder benutzt, dann lauten die Übergangsfunktionen ...

$J_0=x$, $K_0=\overline{x}$

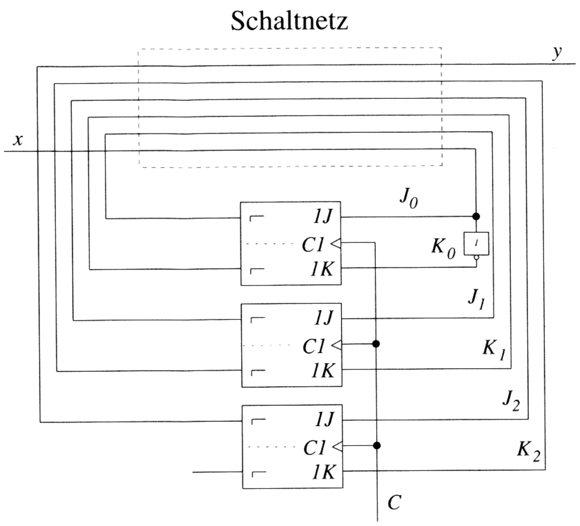
$J_1=z_0$, $K_1=\overline{z_0}$

$J_2=z_1$, $K_2=\overline{z_1}$

... und die Ausgangsfunktion ...

$y=z_2$

Damit ...



≙

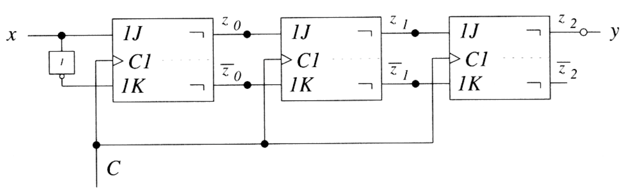


Abb. 88: Rechtsschieberegister aus JK-Flipflops

[Rechtsverschiebung ≙ Information in rechts benachbartes Speicherglied weitergegeben (Schiffmann268) [links→rechts ≙ oben→unten]; Oben spiralförmig nach außen]

Schieberegister kann man nach Art der Aufnahme und Weitergabe der Daten einteilen ...

(a) seriell laden, seriell ausgeben
(b) seriell laden, parallel ausgeben
(c) parallel laden, seriell ausgeben
(d) parallel laden, parallel ausgeben

Weiterhin kann man sie nach der Schieberichtung trennen ...

- von links nach rechts
- von rechts nach links
- nach links im Kreis
- nach rechts im Kreis

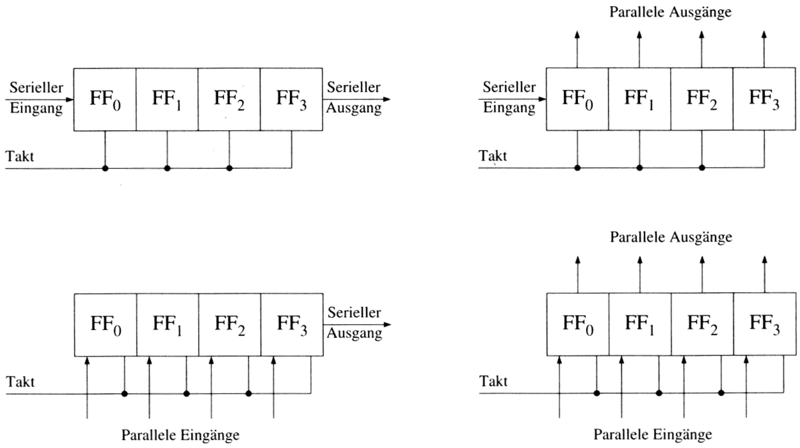


Abb. 89: Verschiedene Funktionen von Schieberegistern
[FF=Flipflop]
Alle vier Arten [aus Abb. 89] von Schieberegistern lassen sich in einer Schaltung unter Verwendung von Multiplexern realisieren ...

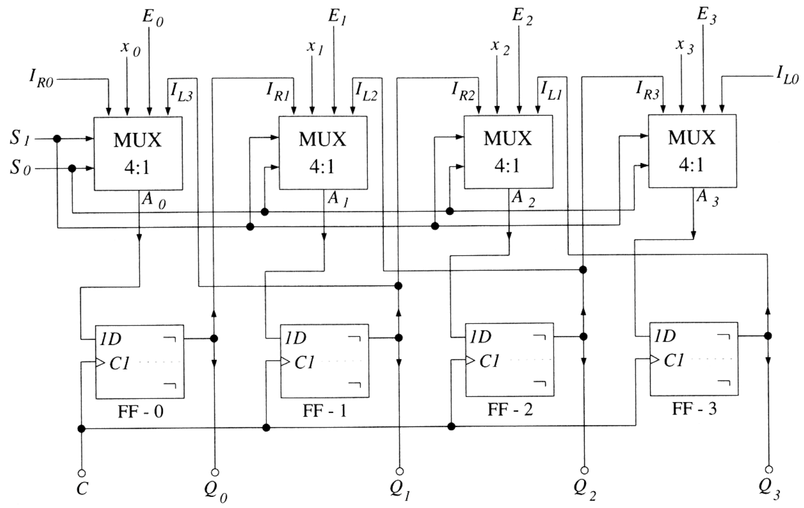


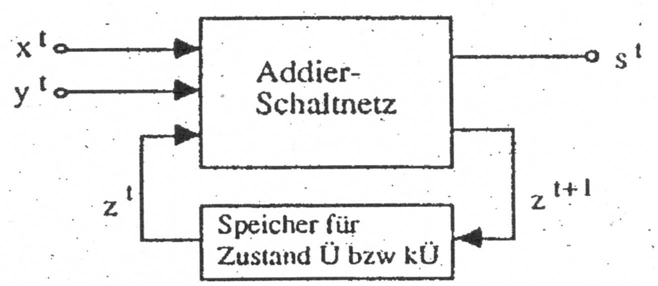
Abb. 90: 4 Bit Schieberegister mit Multiplexern

- **S_1S_0=01 (parallel laden):** Die Eingänge x_0, ..., x_3 der Multiplexer werden auf Null gesetzt und als Löscheingänge auf die MUX-Ausgänge geschaltet. Mit dem folgenden Taktimpuls werden alle Ausgänge Q_0, ..., Q_3 der D-Flipflops auf Null gesetzt [wahrscheinlich initalisieren].
- **S_1S_0=00 (seriell nach rechts schieben):** Die Eingänge I_{R0}, ..., I_{R3} werden auf die MUX-Ausgänge durchgeschaltet. Die an I_{R0} anliegende Bitkombination wird mit jedem Taktimpuls von FF-0 an um eine Stelle nach rechts verschoben.
- **S_1S_0=11(seriell nach links schieben):** Die Eingänge I_{L0}, ..., I_{L3} werden auf die MUX-Ausgänge durchgeschaltet. Die an I_{L0} anliegende Bitkombination wird mit jedem Taktimpuls von FF-3 an um eine Stelle nach links verschoben.
- **S_1S_0=10 (parallel laden):** Die Eingänge E_0, ..., E_3 werden auf die MUX-Ausgänge durchgeschaltet. Mit einem Taktimpuls wird diese Information von den D-Flipflops übernommen und liegt an den Ausgängen Q_0, ..., Q_3 an. Beim nächsten Takt kann eine neue Information von E_0, ..., E_3 übernommen werden.

[Auf diese Art können alle Funktionen entsprechend nachgeahmt werden, wenn man die richtigen Ein- und Ausgänge benutzt]

Schaltwerkssynthese Beispiel 3 (Serienaddierer) (13.4.3)

Es soll ein Serienaddierer synthetisiert werden, der zwei beliebig lange Dualzahlen stellenweise addiert. Die Addition beginnt mit der Stelle niedrigster Wertigkeit. Bei jeder nachfolgenden Stelle muß der Übertrag mit berücksichtigt werden. Die Zahlen werden bitweise eingegeben, pro Taktschritt eine Stelle. Die Ausgabe erfolgt ebenfalls bitweise, wobei die Ausgabefolge zu jedem Zeitpunkt die Summe der bisherigen Eingabefolgen (ohne den letzten Übertrag) darstellt ...



[kÜ=kein Übertrag]

Abb. 91: Blockschaltbild des Serienaddierers

(1) Wir benötigen nur ein Bit für den jeweiligen Übertrag, d.h. nur eine Zustandsvariable. Diese nennen wir z_0 bzw. ü.

(2) Der Anfangszustand ist ü=0.

(3) Die Eingangsvariablen sind x^t,y^t, die Ausgangsvariable ist s^t.

(4) Der Zustandsgraph sieht wie in Abb. 92 aus.

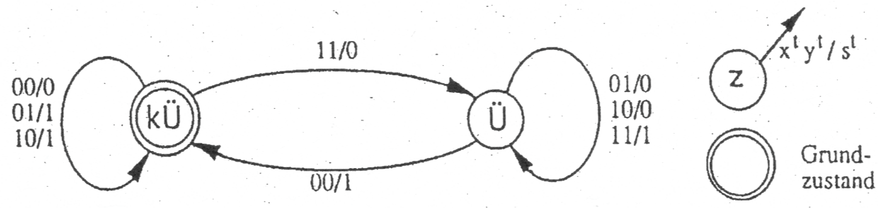


Abb. 92: Zustandsgraph des Serienaddierers

(5) Die erweiterte Zustandsfolgetabelle und die Tabelle der Ausgabefunktion sind in Tab. 19 dargestellt ...

$ü^t$	x^t	y^t	$ü^{t+1}$	s^t	J^t	K^t	
0	0	0	0	0	0	*	[J^t|K^t, 0|0,0|1 möglich ⇒ 0|*]
0	0	1	0	1	0	*	
0	1	0	0	1	0	*	
0	1	1	1	0	1	*	[1|1,1|0 ⇒ 1|* (Fehler im Skript korrigiert)]
1	0	0	0	1	*	1	[1|1,0|1]
1	0	1	1	0	*	0	[0|0,1|0]
1	1	0	1	0	*	0	
1	1	1	1	1	*	0	

Tab. 19: Erweiterte Zustandsfolgetabelle und Tabelle der Ausgabefunktion, sowie Ansteuertabelle der JK-Flipflops

(6) Minimierung der Übergangsfunktion und Ausgangsfunktion ...

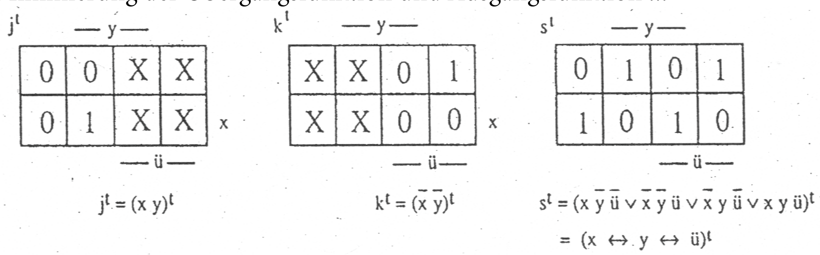


Abb. 93: KV-Diagramme der Schaltnetze für s^t, J^t, K^t

[$x \Leftrightarrow y = xy \vee \bar{x}\bar{y}$, $(x \Leftrightarrow y) \Leftrightarrow ü = ((xy \vee \bar{x}\bar{y})ü) \vee (\overline{(xy \vee \bar{x}\bar{y})}\bar{ü}) = xyü \vee \bar{x}\bar{y}ü \vee \bar{x}y\bar{ü} \vee x\bar{y}\bar{ü}$ wobei $\overline{xy \vee \bar{x}\bar{y}} = (\bar{x} \vee \bar{y}) \wedge (x \vee y) = \bar{x}x \vee \bar{x}y \vee \bar{y}x \vee \bar{y}y = \bar{x}y \vee \bar{y}x$]

$J^t=(xy)^t$

$K^t=(\bar{x}\bar{y})^t$

$s^t=(x\bar{y}\bar{ü} \vee \bar{x}\bar{y}ü \vee \bar{x}y\bar{ü} \vee xyü)^t=(x \equiv y \equiv ü)^t$

(7) Darstellung des Schaltwerks in einem Schaltplan ...