# Mitschrift
# „Schaltwerke und Rechnerorganisation“
# WS 2002/03

Benedikt Meurer[1]

16. Januar 2003

[1] `bmeurer@unix-ag.org`

# Inhaltsverzeichnis

# Kapitel 1

# Schaltwerke

Schaltnetze beschreiben einen funktionalen Zusammenhang zwischen Eingängen und Ausgängen. Zu jedem beliebigen Zeitpunkt hängen die Ausgänge nur von den Eingängen zu diesem Zeitpunkt ab. Damit sind nur Vorgänge *ohne Gedächtnis* modellierbar.

*Schaltwerke* - auch als sequentielle Schaltungen bezeichnet - sind in der Lage mit Hilfe von Speicherelementen zurückliegende Ereignisse zu registrieren. Ein einfaches Beispiel für ein Schaltwerk wäre ein Zähler. Ein weiteres Beispiel wäre der aus der Vorlesung bekannte Aufzug, der sich merken muss, in welchem Stockwerk er sich befindet, um entscheiden zu können, ob nach oben oder nach unten fahren muss, um in ein anderes Stockwerk zu gelangen.

## 1.1 Beschreibung

Zurückliegende Einwirkungen werden bei Schaltwerken durch *Zustände* erfaßt, das heißt zu einem bestimmten Zeitpunkt kann ein Schaltwerk unterschiedliche Zustände annehmen abhängig von Einwirkungen. Die Menge möglicher Zustände definiert einen *Zustandsraum* $Z$.

- $l = 2^k$ Elemente bei einem Schaltwerk mit $k$ Speicherplätzen
- Zustandsvektoren $\vec{z}_i, 1 \leq i \leq l$
- Zustandsraum $Z$: $Z = \{\vec{z}_1, \vec{z}_2, \ldots, \vec{z}_l\}, \vec{z}_i \in \mathbf{B}^k = \{0,1\}^k$
- Ausgangsvektor zum Zeitpunkt $t$: $\vec{a}^t = F(\vec{z}^t, \vec{e}^t), \vec{z}^t \in Z$
- *Zustandsübergangsverhalten*: $\vec{z}^{t+\tau} = G(\vec{z}^t, \vec{e}^t), \vec{z}^t \in Z, \vec{z}^{t+\tau} \in Z$

Ein Schaltwerk besteht allgemein aus $k$ Speicherplätzen, $m$ Ausgängen und $n$ Eingängen. Das heißt die Funktionsbündel $F$ und $G$ sind $(k+n)$-stellige Schaltfunktionen:

$$\begin{aligned} f_i : &\quad \mathbf{B}^{k+n} \rightarrow \mathbf{B}^m, \quad 1 \leq i \leq m \\ g_j : &\quad \mathbf{B}^{k+n} \rightarrow \mathbf{B}^k, \quad 1 \leq j \leq k \end{aligned}$$

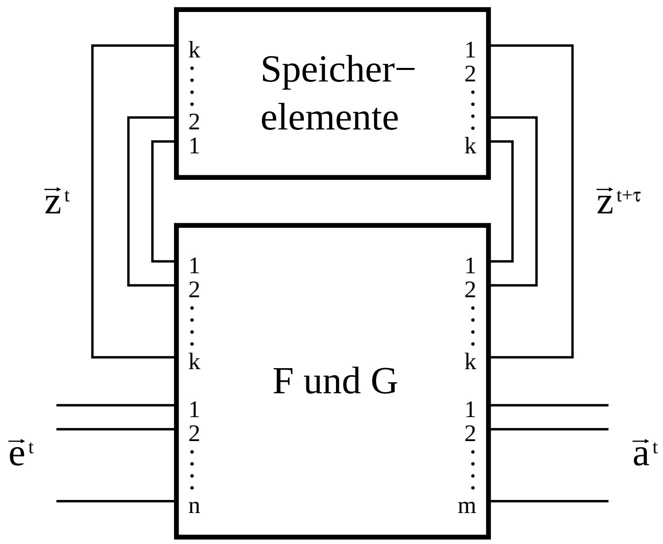



Abbildung 1.1: Schematische Darstellung eines Schaltwerks

**Definition 1.1** Die Mengen $A, E$ sind das Aus- und Eingabealphabet, das heißt $A = \mathbf{B}^m$ und $E = \mathbf{B}^n$. $Z$ ist der Zustandsraum. Die Abbildungen $F : E \times Z \to A, F(\vec{e}, \vec{z}) = \vec{a}$ und $G : E \times Z \to Z, G(\vec{e}, \vec{z}) = \vec{z}^*$ ordnen jedem geordneten Paar $(\vec{e}, \vec{z})$ einen Ausgangsvektor $\vec{a}$ bzw. einen Zustandsvektor $\vec{z}^*$ zu. Die Mengen $E$, $A$ und $Z$ repräsentieren zusammen mit den Abbildungen $F$ und $G$ einen sequentiellen Automaten, abgekürzt $(E, A, Z, F, G)$.

Der MEALY-Automat stellt den allgemeinen Fall dar, so wie in Definition 1.1 angegeben. Der MOORE-Automat unterliegt der Einschränkung, daß die Abbildung $F$ nur von $Z$ abhängt, das heißt $F(\vec{z}) = \vec{a}$.

### 1.1.1 Übergangsgraph

Zusätzlich unterscheiden sich die beiden Automaten dadurch, daß beim MEALY-Automaten die Ausgänge den Zuständen zugeordnet werden, während beim MOORE-Automaten die Ausgänge den Übergangen, und damit den Eingängen, zugeordnet werden.

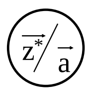



Abbildung 1.2: MEALY-Automat

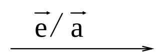



Abbildung 1.3: MOORE-Automat

## 1.2 Schaltwerksanalyse

Die Schaltwerksanalyse beschreibt die Untersuchung der Funktionen einer gegebenen Schaltung. Speicher in Schaltwerken sind rückgekoppelte Schaltelemen-

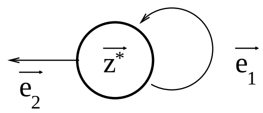



Abbildung 1.4: Stabile Zustände

te. Rückkopplungen werden aufgetrennt, so daß sich die Struktur eines Schaltwerks ergibt, wie in Abbildung 1.5 zu sehen. An der Schnittstelle entsteht eine eingangs- und eine ausgangsseitige Komponente der Zustandsvariablen $\vec{z}$ und $\vec{z}^*$. Die Schaltfunktionen aller Zustandsvariablen beschreiben das Verhalten der Schaltung. Sind $\vec{z}$ und $\vec{z}^*$ für eine Kombination identisch, dann ist der Zustand stabil, anderenfalls handelt es sich um einen Übergangszustand.

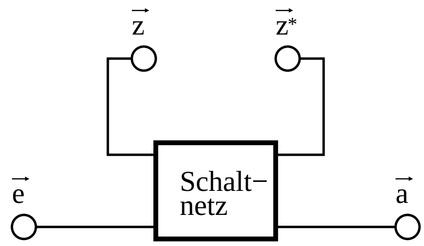



Abbildung 1.5: Schaltnetz

**Allgemeines Vorgehen bei der Analyse von Schaltungen**

1. Erkennen, ob es sich um ein Schaltnetz oder ein Schaltwerk handelt. Schaltwerke besitzen im Gegensatz zu Schaltnetzen Rückkopllungspfade, zu erkennen an vollständig geschlossenen Kreisen der Signalwege in Datenrichtung.

2. Handelt es sich um ein Schaltwerk, müssen alle Rückkopplungen aufgetrennt werden. An den Auftrennung werden zusätzliche Variablen (Zustandsvariablen), eingeführt - eingangsseitig $q_i^*$ und ausgangsseitig $q_i$. Beim Auftrennen der Rückkopllungen sollten die folgenden Regeln beachtet werden:

   - möglichst wenig Auftrennung verwenden
   - so viele Rückkopllungen auf einmal wie möglich auftrennen
   - so nah am Ausgang wie möglich auftrennen
   - möglichst direkt hinter einem Gatter auftrennen

   Nach dem Auftrennen ergibt sich die Struktur eines Schaltnetzes.

3. Zur Analyse eines Schaltnetzes (vorgegeben oder durch Auftrennungen entstanden) werden nun zunächst die Funktionsgleichungen für alle kombinatorischen Ausgänge und die durch die Auftrennung entstandenen Zustandvariablen $q_i$ aus der Schaltung bestimmt.

4. Danach wird die Funktionstabelle aufgestellt.

5. War die Ausgangsschaltung ein Schaltnetz, kann nun noch versucht werden, die Funktion des Schaltnetzes zu ermitteln.
   → **Die Schaltnetzanalyse ist damit abgeschlossen.**

6. War die Ausgangsschaltung ein Schaltwerk, werden in der Funktionstabelle nun die stabilen Zustände gekennzeichnet. Dies sind solche Zustände, bei denen bei einer bestimmten Eingangskombination auf beiden Seiten der Schnittstelle ($z_i^*$ und $z_i$) identische Werte auftreten.
   Treten unterschiedliche Variablenwerte auf beiden Seiten einer Schnittstelle auf, so ist der betreffende Übergangszustand instabil. Er kann nur kurzzeitig auftreten und muss schließlich in einen stabilen Zustand übergehen.

7. Ein Zustand kann für unterschiedliche Eingangskombinationen stabil oder instabil sein. Zur einfacheren Beschreibung des Schaltwerkes ist es i.a. ausreichend, nur diejenigen Zustände zu berücksichtigen, für die mindestens eine Eingangskombination stabil ist. Hiermit kann man die Übergangstabelle verkürzen und das Schaltverhalten anschaulich in einem Übergangsdiagramm darstellen. Instabile Zustände bewirken hierbei immer Zustandsübergange.

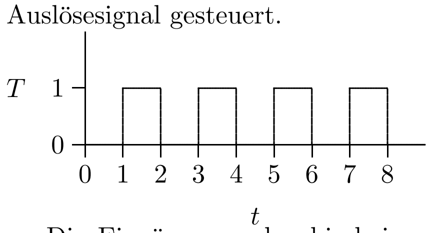
8. Anhand des Zustandsdiagramms kann versucht werden, die Funktion der vorgegebenen Schaltung zu ermitteln.
   → **Die Schaltwerksanalyse ist damit abgeschlossen.**

## 1.3 Elementare Schaltwerke

Man unterscheided im allgemeinen zwischen *asynchronen* und *synchronen* Schaltwerken.

Asynchrone Schaltungen sind durch spontane Schaltvorgänge gekennzeichnet. Mit einer Änderung der Eingangssignale ändern sich Zustand und Ausgänge. Daraus folgt eine maximale Geschwindigkeit, undefinierte Laufzeiten und hohe Störanfälligkeit.

Synchrone Schaltungen hingegen werden durch ein Signal (den *Takt*) als Auslösesignal gesteuert.




Die Eingänge werden hierbei nur zu bestimmten definitierten Zeiten ausgewertet, das heißt Schaltungsteile werden synchronisiert und zwischenzeitliche Störungen, wie sie bei asynchronen Schaltungen auftreten können, werden unterdrückt.

### 1.3.1 Schaltwerke zur Speicherung einer binären Variablen

**Anforderungen an ein Speicherelement**

**Speicherung** Es sind zwei stabile Zustände für die Speicherung der binären Variable erforderlich.

**Schreiben** Die Zustände müssen von außen einstellbar sein.

**Lesen** Die Speicherinhalte sollen negiert und nicht negiert an den Ausgängen ablesbar sein.

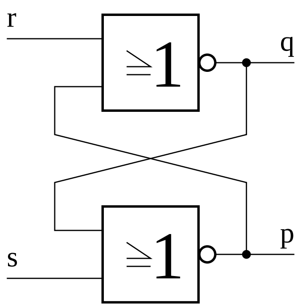



Abbildung 1.6: Schaltwerk aus zwei gekoppelten NOR-Gliedern

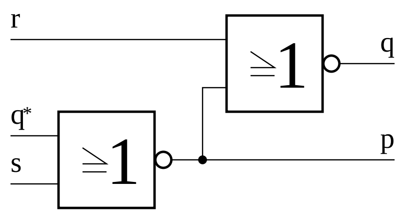



Abbildung 1.7: Schaltwerk mit aufgetrennter Rückkopplung

| Nr. | $r$ | $s$ | $q^*$ | $p$ | $q$ |
|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 0 |
| 2 | 0 | 0 | 1 | 0 | 1 |
| 3 | 0 | 1 | 0 | 0 | 1 |
| 4 | 0 | 1 | 1 | 0 | 1 |
| 5 | 1 | 0 | 0 | 1 | 0 |
| 6 | 1 | 0 | 1 | 0 | 0 |
| 7 | 1 | 1 | 0 | 0 | 0 |
| 8 | 1 | 1 | 1 | 0 | 0 |

Tabelle 1.1: Funktionstabelle der Schaltung nach Abbildung 1.6

$$\begin{aligned} p &= \overline{s + q^*} \\ q &= \overline{r + p} \\ &= \overline{r} \cdot \overline{p} \end{aligned}$$

### 1.3.2 Getaktetes RS-Flipflop

Der Pfeil auf dem Schaltbild des getakteten RS-Flipflops kennzeichnet einen sogenannten *dynamischen Eingang*, das heißt, daß nicht der Takt $T$ wirksam ist, sondern die positive Flanke des Taktes $T$.

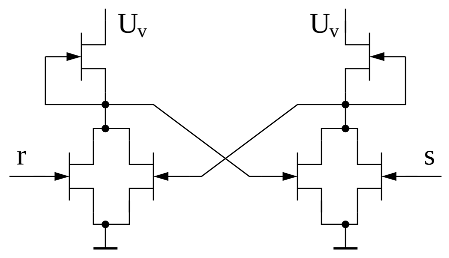



Abbildung 1.8: NOR-Schaltung rückgekoppelt (MNMOS Technologie)

| $r$ | $s$ | $q$ | Bemerkung |
|---|---|---|---|
| 0 | 0 | $q^*$ | Speichern |
| 0 | 1 | 1 | Setzen (set) |
| 1 | 0 | 0 | Zurücksetzen (reset) |
| 1 | 1 | – | nicht zulässig |

Tabelle 1.2: Verkürzte Übergangstabelle

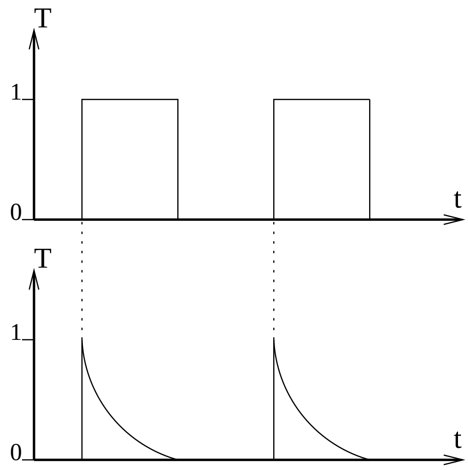



Abbildung 1.9: Taktflankensteuerung

Der Auswertezeitraum der Eingangssignale $r$ und $s$ wird gegenüber einer Zugangssteuerung stark verkürzt, man spricht von *Flankensteuerung*.

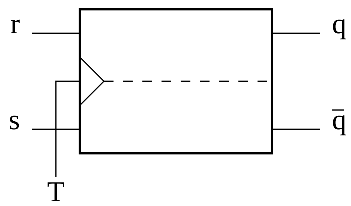



Abbildung 1.10: Getaktetes RS-Flipflop

Die Signale $r$ und $s$ werden zu *Verarbeitungseingängen*, der Takt $T$ wirkt als *Auslöseeingang.*

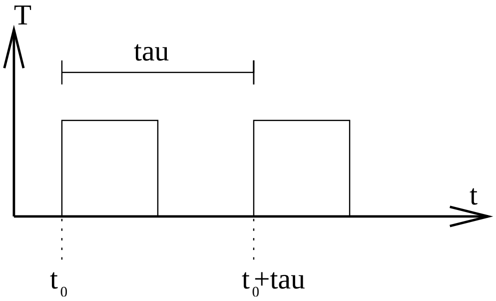



Abbildung 1.11: Taktabstände

### 1.3.3 Getaktetes Vorspeicher-Flipflop (Master-Slave-Flipflop)

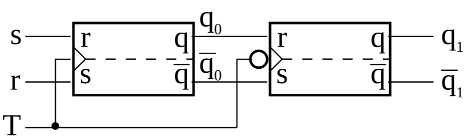



Abbildung 1.12: Schaltbild eines Vorspeicher-Flipflop

$$q_0 = \overline{(r \cdot p) + \overline{(q_0^* + (s \cdot p))}}$$
$$q_1 = \overline{(\overline{q}_0^* \cdot n) + \overline{(q_1^* + (q_0^* \cdot n))}}$$
$$(r = s = 1 \text{ unzulässig})$$

Wir treffen folgende Zuordnung:

| $q_0^*$ | $q_1^*$ | Zustand |
|---|---|---|
| 0 | 0 | Zustand 1 |
| 1 | 0 | Zustand 2 |
| 1 | 1 | Zustand 3 |
| 0 | 1 | Zustand 4 |

Tabelle 1.3: Zuordnung der Zustände

Bedingungen aus den Übergansfunktionen, unter welchen Bedingungen Zustände beibehalten bzw. unter welchen Bedingungen Zustände verlassen werden.

| **Zustand** | **Bedingung** | **Folgezustände** | | |
|---|---|---|---|---|
| Zustand 1 | | Zustand 1 | Zustand 2 | |
| $q_0^* = 0$ | $q_0 = \overline{r}sp$ | $0 = r + \overline{s} + \overline{p}$ | $1 = \overline{r}sp$ | |
| $q_1^* = 0$ | $q_1 = 0$ | $0 = 0$ | $0 = 0$ | |
| Zustand 2 | | Zustand 1 | Zustand 2 | Zustand 3 |
| $q_0^* = 1$ | $q_0 = \overline{r} + \overline{p}$ | $0 = rp$ | $1 = \overline{r} + \overline{p}$ | $1 = n$ |
| $q_1^* = 0$ | $q_1 = n$ | $0 = \overline{n}$ | $0 = \overline{n}$ | $1 = n$ |
| Zustand 3 | | Zustand 3 | Zustand 4 | |
| $q_0^* = 1$ | $q_0 = \overline{r} + \overline{p}$ | $1 = \overline{r} + \overline{p}$ | $0 = r \cdot p$ | |
| $q_1^* = 1$ | $q_1 = 1$ | $1 = 1$ | $1 = 1$ | |
| Zustand 4 | | Zustand 4 | Zustand 3 | Zustand 1 |
| $q_0^* = 0$ | $q_0 = \overline{r}sp$ | $0 = r + \overline{s} + \overline{p}$ | $1 = \overline{r}sp$ | $0 = r + \overline{s} + \overline{p}$ |
| $q_1^* = 1$ | $q_1 = \overline{n}$ | $1 = \overline{n}$ | $1 = \overline{n}$ | $0 = n$ |

Tabelle 1.4: Erweiterte Zustandsübergangstabelle für ein Master-Slave-Flipflop

### 1.3.4 JK-Flipflop

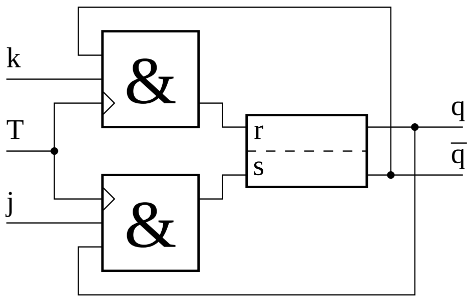



Abbildung 1.13: Schaltbild eines JK-Flipflop

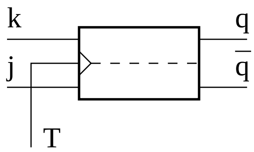



Abbildung 1.14: Schaltzeichen eines JK-Flipflop

Für die Eingänge am RS-Flipflop gelten also die folgenden Gleichungen:

$$r = k \cdot q \cdot p = k^t \cdot q^t$$
$$s = j \cdot \overline{q} \cdot p = j^t \cdot \overline{q}^t$$

Die Eingangsgrößen $j = 1, k = 1$ sind jetzt zulässig, wie die folgende kurze Zwischenrechnung zeigt:

$$\begin{aligned} r \cdot s &= (k^t \cdot q^t) \cdot (j^t \cdot \overline{q}^t) \\ &= (j^t \cdot k^t) \cdot \underbrace{(q^t \cdot \overline{q}^t)}_{=0} = 0 \end{aligned}$$

Damit sieht die Übergangsgleichung für das JK-Flipflop wie folgt aus:

$$
\begin{aligned}
q^{t+1} &= (j^t \cdot \overline{q}^t) + [\overline{(k^t \cdot q^t)} \cdot q^t] \\
&= (j^t \cdot \overline{q}^t) + (\overline{k}^t \cdot q^t) + (q^t \cdot \overline{q}^t) \\
&= (j^t \cdot \overline{q}^t) + (\overline{k}^t \cdot q^t)
\end{aligned}
$$

| $j^t$ | $k^t$ | $q^{t+1}$ | Bemerkung |
|---|---|---|---|
| 0 | 0 | $q^t$ | speichern (save) |
| 1 | 0 | 1 | setzen (set) |
| 0 | 1 | 0 | zurücksetzen (reset) |
| 1 | 1 | $\overline{q}^t$ | wechseln (toggle) |

Tabelle 1.5: Übergangstabelle des JK-Flipflop

### 1.3.5 D-Flipflop

Das D-Flipflop - das *D* steht für *Delay* (Verzögerung) - hat nur einen Vorbereitungseingang, dessen Wert bei Eintreffen eines Taktsignals gespeichert wird (Verzögerungselement). Die Übergangsfunktion für das D-Flipflop lautet $q^{t+1} = D^t$.

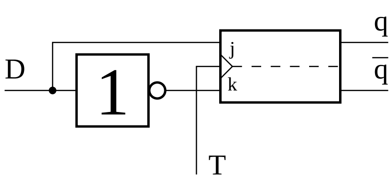



Abbildung 1.15: Schaltbild eines D-Flipflop

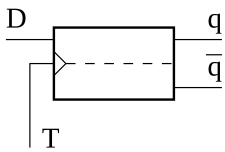



Abbildung 1.16: Schaltzeichen eines D-Flipflop

| $D^t$ | $q^{t+1}$ |
|---|---|
| 0 | 0 |
| 1 | 1 |

Tabelle 1.6: Übergangstabelle des D-Flipflop

## 1.4 Schaltwerksynthese

Schaltwerksynthese ist der *Entwurf eines Schaltwerks mit vorgegebener Funktionalität.* Dabei sind die einzelnen Schritte wie folgt:

1. Spezifikation (verbale Beschreibung)
2. Formale Beschreibung (Eingangs-/Ausgangsfunktionen, Übergangsfunktionen)
    - Festlegung der Anzahl der Zustände

- Zustandscodierung

3. Speicherelemente realisieren bzw. ansteuern

Bei asynchronen Schaltwerken werden die Speicherelemente direkt realisiert, indem für eine bestimmte Zustandsvariable $z_i$ eine Verbindung zwischen $z_i$ und $z_i^*$ in Form einer Rückkopplung hergestellt wird.

Bei synchronen Schaltungen wird für jede Zustandsvariable ein Flipflop so angesteuert, daß die Übergangsfunktion der entsprechenden Variablen realisiert wird.

### 1.4.1 Ansteuergleichungen für Speicherglieder

Das Aufstellen der Ansteuergleichungen für Speicherglieder läuft darauf hinaus eine Übereinstimmung zwischen $z_i^{t+1}$ und $q_i^{t+1}$ herzustellen. Daraus ergibt sich, daß die Ansteuerbedingungen für ein Flipflop aus der Zustandsübergangsfunktion der entsprechenden Variablen berechenbar sind, es gilt also $z_i^{t+1} = q_i^{t+1}$.

#### Methode des Koeffizientenvergleichs

$$f_i(z^t, e^t) = g_i(q_i^t, \vec{v}^t)$$

diese Gleichung läßt sich nur dann erfüllen, wenn die Koeffizienten von $q^t$ auf beiden Seiten übereinstimmen.

#### Koeffizientenvergleich

$$\begin{array}{ll} s_0^t = \overline{z}_0^t & s_1^t = z_0^t \cdot \overline{z}_1^t \\ \overline{r}_0^t = 0 \text{ bzw. } r_0^t = 1 & r_1^t = z_0^t \end{array}$$

Einbringung der Nebenbedingung

$$\begin{array}{ll} z_0^{t+1} = \overline{z}_0^t + \overbrace{\overline{z}_0^t z_0^t}^{=0} & z_1^{t+1} = z_0^t \overline{z}_1^t + \overline{(z_0^t z_1^t)} z_1^t \\ \Rightarrow s_0^t = \overline{z}_0^t \wedge r_0^t = z_0^t & \Rightarrow s_1^t = z_0^t \cdot \overline{z}_1^t \wedge r_1^t = z_0^t \cdot z_1^t \end{array}$$

#### Fallunterscheidung

*Entwicklungssatz* besagt, daß man jede Schaltfunktion nach jeder in ihr auftretenden unabhängigen Variablen entwickeln kann, indem man diese Variable zu 0 bzw. zu 1 setzt und mit der negierten bzw. nicht negierten Variablen konjunktiv verknüpft.

#### Entwurfsbeispiel

Es soll ein Schaltwerk entworfen werden, daß Dualzahlen seriell addiert. Dazu brauchen wir zwei $n$-Bit Register, die die Operanden $A$ und $B$ enthalten, ein Speicherelement, welches das Übertragsbit, das bei jeder Addition entsteht speichert, und ein Schaltnetz, das die eigentliche Addierarbeit verrichtet. Das Ergebnis der Addition soll nach Abschluß der Addition im Register $B$ bereit liegen. Abbildung 1.18 zeigt den Aufbau der Operandenregister und Abbildung 1.17 zeigt den aus unseren Überlegungen entstandenen Schaltplan für den Addierer.

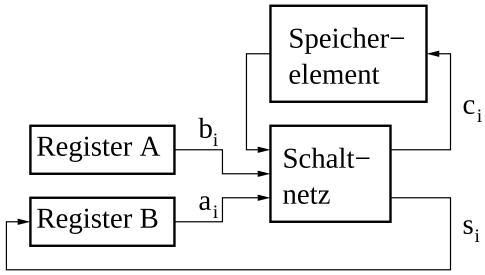



Abbildung 1.17: Schaltplan des seriellen Addierwerks

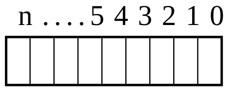



Abbildung 1.18: Aufbau der Operandenregister

Dazu muß der Übertrag $c_i$ (engl. *carry*) gespeichert werden und bei der Verarbeitung (in diesem Fall bei der Addition) des Ziffernpaares mit dem nächsthöheren Index mitberücksichtigt werden.

Benötigt wird ein Schaltnetz mit 3 Eingängen $a_i, b_i, c_i$ und zwei Ausgängen $s_i$ und $c_i$. $i$ ist bei dieser Betrachtung der räumliche Index, im nächsten Schritt gehen wir nun über zum zeitlichen Index, den man bei Schaltwerken immer hat.

Realisierung des Übertragsspeichers mit einem JK-Flipflop.