Grundlagen der Digitaltechnik Foliensatz 5: Vom Transistor zur Schaltung

G. Kemnitz

23. Februar 2021

Contents

Logikschaltungen dürfen nicht kompliziert sein

Um die heutigen digitalen Schaltkreise entwerfen und produzieren zu können, darf von den Millionen von Transistoren und Leitungen je Chip im Mittel ≤ 1 fehlerhaft sein oder ausfallen. Erreicht wird das durch

- einen hohen Automatisierungsgrad beim Entwurf und der Fertigung,
- Schaltelemente mit langer Lebensdauer, ...
- einfache Schaltungen.

Tatsache

Digitale Grundschaltungen sind einfach aufgebaut, zu verstehen und zu Systemen zu verschalten.

Schaltelemente – historische Sicht

Als Schaltelemente dienten historisch gesehen nacheinander:

- Relais (elektromagnetische Schalter): groß, großer Stromverbrauch, langsam, geringe Lebensdauer.
- Elektroröhren: schneller, aber immer noch groß, großer Stromverbrauch, geringe Lebensdauer.
- Bipolartransistoren: schneller, kleiner wesentlich langlebiger und wesentlich geringerer Stromverbrauch.
- MOS-Transistor: ps bis ns Verzögerung, extrem niedriger Stromverbrauch je Operation, hohe Integrationsdichte.

Grundkenntnisse der aktuellen Schaltungstechnik werden vor allem für die Abschätzungen der Realisierbarkeit, des Aufwands, der Geschwindigkeit und des Stromverbrauchs benötigt.

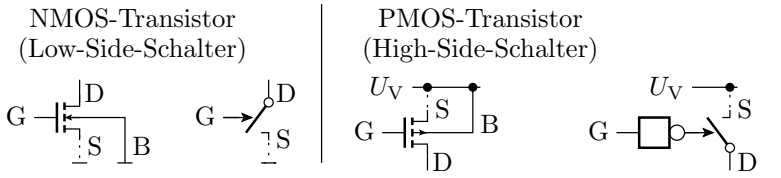
1 Gatterentwurf

1.1 MOS-Transistoren als Schalter

MOS-Transistoren als Schalter

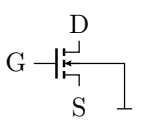
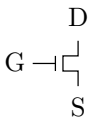
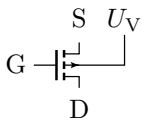
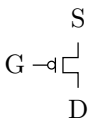
CMOS-Gatter bestehen aus zwei Arten von Transistorschaltern:

- NMOS-Transistoren zum Schalten einer Verbindung nach '0' ($\perp$, Masse, negativer Anschluss der Spannungsversorgung) und
- PMOS-Transistoren zum Schalten einer Verbindung nach '1' (U_V, positiver Anschluss der Spannungsversorgung).



S Source, Verbindung zu '0' bzw. '1'	D Drain, Verbindung zum Gatterausgang
G Gate, Steuereingang	U_V Versorgungsspannung
B Bulk (Subtrat), mit '0' bzw. '1' zu verbinden	$\perp$ Masse

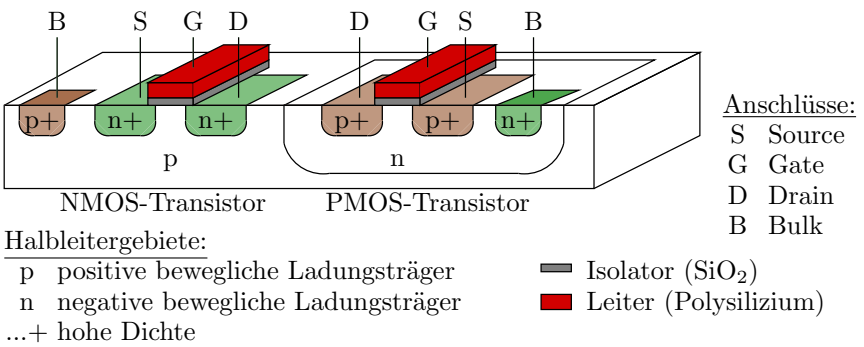
Ein NMOS-Transistor schaltet bei einer '1' (hoher Spannung) am Gate ein und ein PMOS-Transistor bei '0' (niedriger Spannung) am Gate ein, d.h. er invertiert.

Transistorschalter	Schaltsymbol komplett	Schaltsymbol vereinfacht	Funktion
Low-Side-Schalter (NMOS-Transistor)			G | S→D 0 | 0 1 | 1
High-Side-Schalter (PMOS-Transistor)			G | S→D 0 | 1 1 | 0

In der vereinfachten Schaltungsdarstellung werden die mit U_V und Masse verbundenen Bulk-Anschlüsse weggelassen.

Aufbau und Funktion von MOS-Transistoren

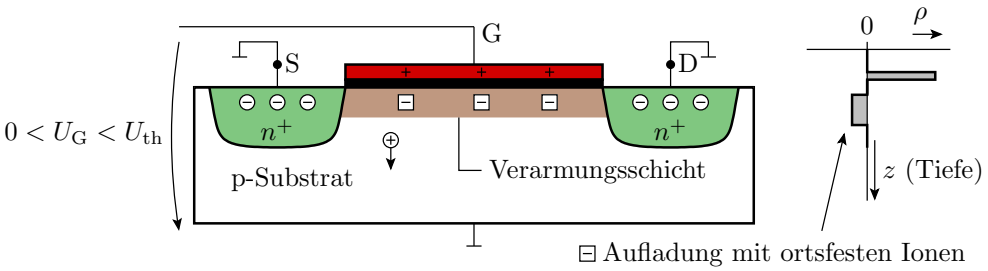
MOS-Transistoren bestehen aus n- und p-leitfähigen Gebieten. P-leitfähig bedeutet, dass die beweglichen Ladungsträger positiv geladen sind (Löcher) und n-leitfähig, dass sie negativ geladen sind (bewegliche Elektronen). Die Leitfähigkeit wird durch die gezielte Einbringung von Fremdatomen in das Kristallgitter (Dotierung) eingestellt.



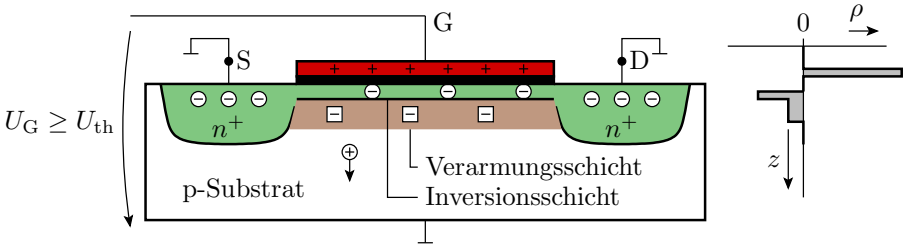
NMOS-Transistoren bestehen aus einem schwach p-leitfähigen Bulk mit eingebrachten stark n-leitfähigen Source- und Drain-Gebieten und einem stark p-leitfähigem Gebiet für den Bulk-Anschluss. Über dem Kanal zwischen dem Source- und dem Drain-Gebiet befindet sich isoliert durch eine dünne SiO_2-Schicht der Steueranschluss das Gate. Ein PMOS-Transistor ist genauso aufgebaut, nur sind n- und p-Leitfähigkeit vertauscht.
Über den Halbleitergebieten befinden sich, getrennt durch Isolationsschichten Metallebenen für die Verdrahtung der Transistoren zu Gattern und Funktionsblöcken (nicht dargestellt).

Feldeffekt (NMOS-Transistor)

Mit dem Bulk an Masse sind die pn-Übergänge von den n-leitfähigen Source- und Drain-Gebieten zum p-leitfähigen Bulk gesperrt. Die Gate-Isolator-Halbleiter-Struktur bildet einen Plattenkondensator. Bei einer schwach positiven Gate-Spannung U_G kleiner der Einschaltspannung U_{th} (Eingabe null) driften die p-Ladungen unter dem Gate weg und hinterlassen ortsfeste negative Ladungen. Die pn-Übergänge zu Source und Drain bleiben gesperrt. Die Drain-Source-Verbindung ist ausgeschaltet.

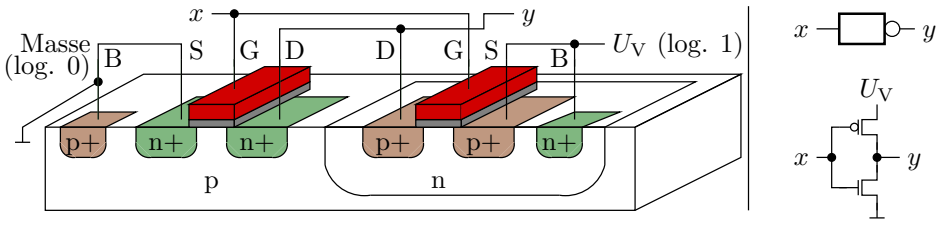


Bei einer großen Gate-Spannung $U_g > U_{th}$ (Eingabe eins) lädt sich der Kanal unter dem Gate soweit negativ auf, dass die beweglichen Elektronen vom Source[1] in den Kanal diffundieren. Es bildet sich eine n-leitfähige vom Bulk isolierte Verbindung vom Source zum Drain.



Ein PMOS-Transistor funktioniert genauso. Nur sind p- und n-Leitfähigkeiten und alle Spannungsvorzeichen umgekehrt.

Der CMOS-Inverter

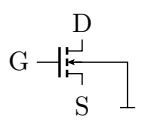
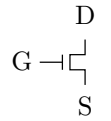
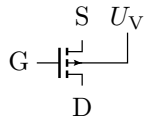
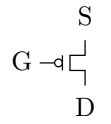


Ein CMOS-Inverter besteht aus einem NMOS-Transistor, der bei einer Eins am Eingang den Ausgang mit Masse und einem PMOS-Transistor, der bei einer Null am Eingang den Ausgang mit U_V verbindet.

x	NMOS-Transistor	PMOS-Transistor	y
0	ausgeschaltet	eingeschaltet	1
1	eingeschaltet	ausgeschaltet	0

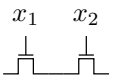
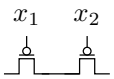
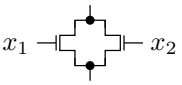
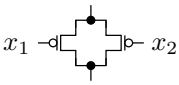
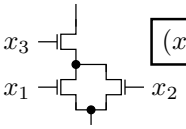
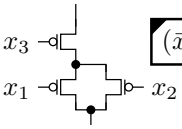
[1] Der Bezeichner Source bedeutet Quelle der beweglichen Ladungsträger. Das ist von den beiden Kanalanschlüssen beim NMOS-Transistor der mit dem niedrigeren und beim PMOS-Transistor der mit dem höheren Potential. Source und Drain können während des Betriebs ihre Rollen tauschen.

Schaltermodell

Transistorschalter	Schaltsymbol komplett	Schaltsymbol vereinfacht	Funktion
Low-Side-Schalter (NMOS-Transistor)			G | S→D: 0 | 0; 1 | 1
High-Side-Schalter (PMOS-Transistor)			G | S→D: 0 | 1; 1 | 0

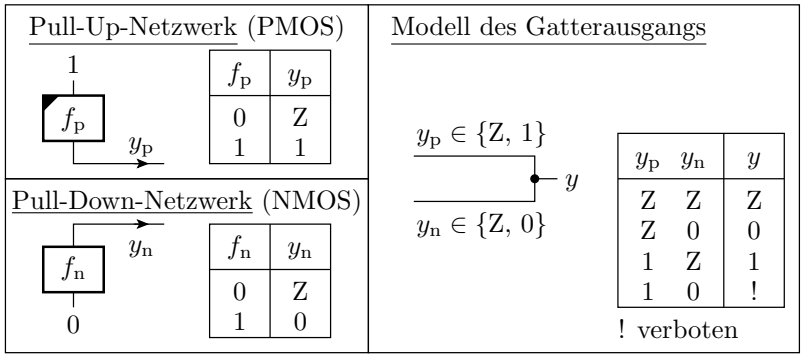
Mit der Festlegung, dass null klein oder aus und eins groß oder ein bedeutet, schaltet ein NMOS-Transistor die Verbindung zu Masse nichtinvertierend und ein PMOS-Transistor die Verbindung zur Versorgungsspannung U_V invertierend.

Geschaltete Transistornetzwerke

	NMOS-Netzwerk Struktur	NMOS-Netzwerk Funktion	PMOS-Netzwerk Struktur	PMOS-Netzwerk Funktion
Reihenschaltung		$x_1 \wedge x_2$		$\bar{x}_1 \wedge \bar{x}_2$
Parallelschaltung		$x_1 \vee x_2$		$\bar{x}_1 \vee \bar{x}_2$
gemischte Reihen- und Parallelschaltung		$(x_1 \vee x_2) \wedge x_3$		$(\bar{x}_1 \vee \bar{x}_2) \wedge \bar{x}_3$

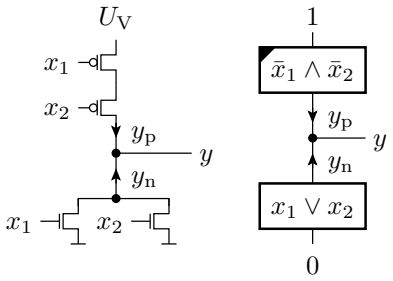
Eine Parallelschaltung ist eine ODER- und eine Reihenschaltung eine UND-Verknüpfung. NMOS-Transistoren sind zum Schalten von Verbindungen zu '1' und PMOS-Transistoren von Verbindung zu '0' elektrisch problematisch.

Vom geschalteten Netzwerk zum Gatter



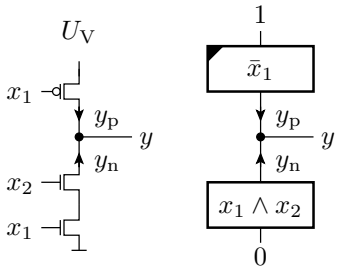
Der Gatterausgang kann entweder mit eins (U_V), mit null (Masse) verbunden oder isoliert sein ('Z', hochohmig). Ein hochohmiges Signal speichert seinen Wert in der Lastkapazität und wird nach kurzer Zeit ungültig (Wert unvorhersagbar). Null und eins setzen sich gegenüber 'Z' durch. Gleichzeitig null und eins ist nur kurzzeitig während der Schaltvorgänge erlaubt.

NOR-Gatter



x_2	x_1	f_p	f_n	y_p	y_n	y
0	0	1	0	1	Z	1
0	1	0	1	Z	0	0
1	0	0	1	Z	0	0
1	1	0	1	Z	0	0

Gatter, dessen Ausgang auch hochohmig gesteuert werden kann



x_2	x_1	f_p	f_n	y_p	y_n	y
0	0	1	0	1	Z	1
0	1	0	0	Z	Z	Z
1	0	1	0	1	Z	1
1	1	0	1	Z	0	0

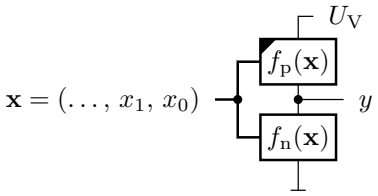
Tatsache

Der Entwurf frei strukturierter Gatter erfolgt nach einfachen logischen Regeln.

1.2 FCMOS-Gatter

FCMOS-Gatter

FCMOS bedeutet vollständig komplementär (**F**ull **C**omplementary). Der Ausgang ist stationär immer entweder mit '1' oder '0' verbunden:

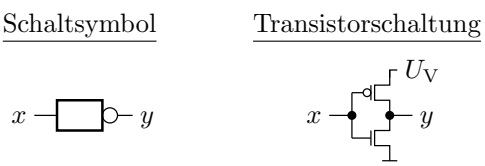


$f_n(\mathbf{x})$	$f_p(\mathbf{x})$	y
0	0	hochohmig
0	1	1
1	0	0
1	1	verboten

Die Funktion des PMOS-Netzwerks $f_p(\mathbf{x})$ ist identisch mit der des Gatters $f(\mathbf{x})$ und die Funktion des NMOS-Netzwerks $f_n(\mathbf{x})$ ist die negierte Funktion davon:

$$\begin{aligned} f_n(\mathbf{x}) &= \bar{f}(\mathbf{x}) \\ f_p(\mathbf{x}) &= f(\mathbf{x}) \end{aligned}$$

Das einfachste Gatter, der Inverter



$$\begin{aligned} f_n &= x \\ f_p &= \bar{x} \end{aligned}$$

- Jeweils die Funktion eines Einzeltransistors.

Entwurf eines NAND-Gatters

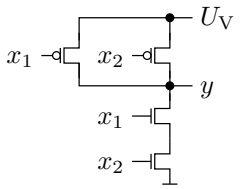
$$y(\mathbf{x}) = \overline{x_1 x_2}$$

NMOS-Netzwerk: Umformung doppelte Negation. Aus UND wird eine Reihenschaltung.

$$f_\mathrm{n}(\mathbf{x}) = \overline{\overline{x_1 x_2}} = x_1 x_2$$

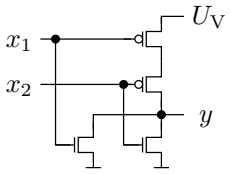
PMOS-Netzwerk: Umformung mit der De Morganschen Regel. Aus ODER wird eine Parallelschaltung.

$$f_\mathrm{p}(\mathbf{x}) = \overline{x_1 x_2} = \bar{x}_1 \vee \bar{x}_2$$



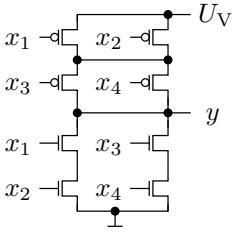
- NOR-Gatter

$$\begin{aligned} y(\mathbf{x}) &= \overline{x_1 \vee x_2} \\ f_\mathrm{n}(\mathbf{x}) &= x_1 \vee x_2 \\ f_\mathrm{p}(\mathbf{x}) &= \bar{x}_1 \bar{x}_2 \end{aligned}$$



- Komplexgatter:

$$\begin{aligned} y(\mathbf{x}) &= \overline{x_1 x_2 \vee x_3 x_4} \\ f_\mathrm{n}(\mathbf{x}) &= x_1 x_2 \vee x_3 x_4 \\ f_\mathrm{p}(\mathbf{x}) &= (\bar{x}_1 \vee \bar{x}_2)(\bar{x}_3 \vee \bar{x}_4) \end{aligned}$$



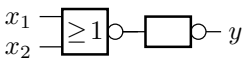
Jede Variable in den Ausdrücken von f_n und f_p kostet einen Transistor. Vor der Nachbildung durch Transistornetzwerke ist die Anzahl der Variablen in den Ausdrücken entsprechend zu minimieren. Beispiele:

$$y = \overline{(x_1 x_2 x_3) \vee x_1 \vee x_2} = \overline{x_1 \vee x_2}$$

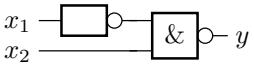
$$y = \overline{x_1 x_2 \vee x_1 x_3 \vee x_2 x_3} = \overline{x_1 (x_2 \vee x_3) \vee x_2 x_3}$$

Mit einem FCMOS-Gatter sind die Funktionen aller negierten Ausdrücke aus UND- und ODER-Verknüpfungen nachbildbar. Andere Funktionen benötigen zur Nachbildung mehrere Gatter oder zusätzliche Inverter für die zu invertierenden Ein- und Ausgangssignale.

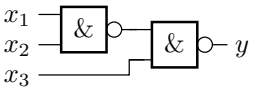
$y = x_1 \vee x_2$



$y = \overline{\bar{x}_1 x_2}$



$y = \overline{\overline{x_1 x_2} x_3}$



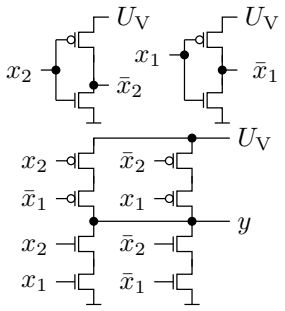
EXOR-Gatter

- 1-Bit-Addition unter Vernachlässigung des Übertrags:

$$y = x_1 \oplus x_2$$

- $y = 0$ wenn $x_1 = x_2$:

$$\begin{aligned} f_\mathrm{n} &= \overline{x_1 \oplus x_2} \\ &= x_1 x_2 \vee \bar{x}_2 \bar{x}_1 \end{aligned}$$

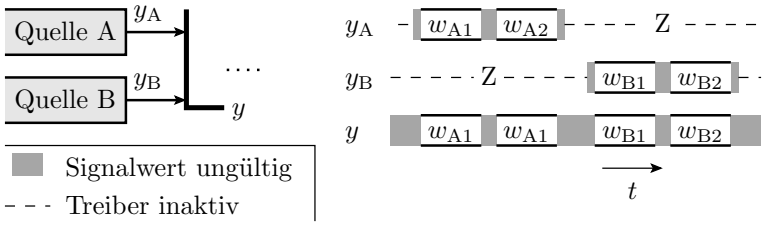


- $y = 1$ wenn $x_1 \neq x_2$:

$$\begin{aligned} f_\mathrm{p} &= x_1 \oplus x_2 \\ &= \bar{x}_1 x_2 \vee x_1 \bar{x}_2 \end{aligned}$$

1.3 Deaktivierbare Treiber

Deaktivierbare Treiber

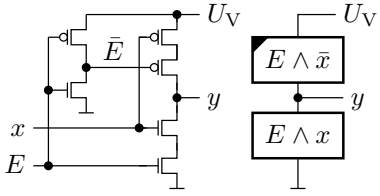


Ein Bus in einem Rechner

- ist ein zentraler Informationsknoten,
- oft mit mehreren Treibern,
- von denen nur ein Treiber gleichzeitig senden darf.

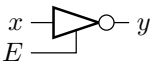
Bustreiber müssen deaktivierbar (hochohmig schaltbar) sein.

Treiber für ein Bitsignal



x	E	f_p	f_n	y
0	0	0	0	Z
0	1	1	0	1
1	0	0	0	Z
1	1	0	1	0

Schaltsymbol

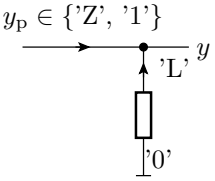


$$\begin{aligned} f_\mathrm{p} &= E \wedge \bar{x} \\ f_\mathrm{n} &= E \wedge x \end{aligned}$$

Das PMOS-Netzwerk benötigt das invertierte Freigabesignal $\bar{E}$. Bereitstellung mit einem zusätzlichen Inverter.

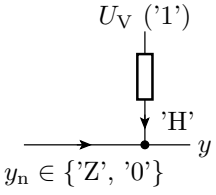
Pull-Up- und Pull-Down-Elemente

Gatter mit Pull-Down-Widerstand



y_p	y
'Z'	'L'
'1'	'1'

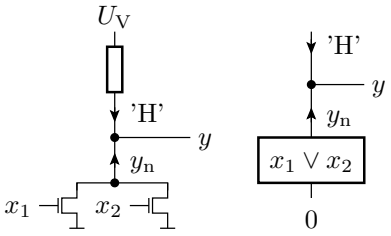
Gatter mit Pull-Up-Widerstand



y_n	y
'0'	'0'
'Z'	'H'

- Ein Pull-Down-Element erzeugt eine schwache Null, `std_logic`–Wert 'L'.
- Ein Pull-Up-Element erzeugt eine schwache Eins, `std_logic`-Wert 'H'.
- Schwache Werte überschreiben 'Z'.
- Starke Wert ('0', '1') überschreiben schwache Werte ('L', 'H', 'Z').

NOR-Gatter mit Pull-Up-Widerstand

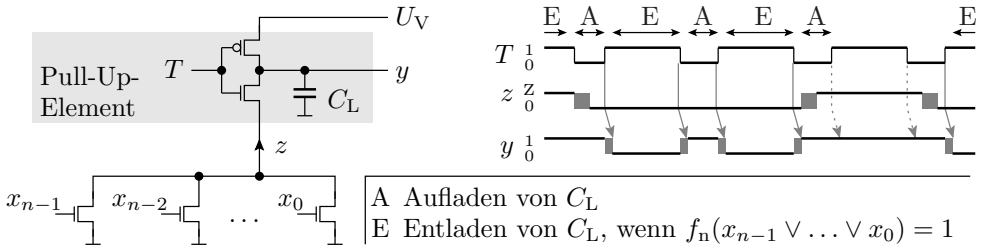


x_2	x_1	f_n	y_n	y
'0'	'0'	'0'	'H' 'Z'	'H'
'0'	'1'	'1'	'H' '0'	'0'
'1'	'0'	'1'	'H' '0'	'0'
'1'	'1'	'1'	'H' '0'	'0'

Das Pull-Up-Element ersetzt das PMOS-Netzwerk. Die Nachfolgeschaltung unterscheidet nicht zwischen schwachen und starken Signalwerten.

- Vorteil: weniger geschaltete Transistoren.
- Nachteil: Stromaufnahme außerhalb der Schaltvorgänge ist größer null.

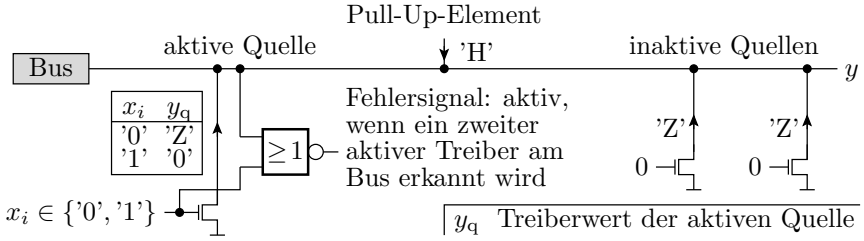
Geladene Kapazität als Pull-Up-Element



- $T = 0$: aufladen.
- $T = 1$: nur entladen, wenn $y_n = 0$.
- Pull-Up-Element ersetzt Reihenschaltung aus n Transistoren.
- Nachteil: Taktsignal erforderlich, verkürztes Gültigkeitsfenster.

Bus mit mehreren Signalquellen (Wired-AND)

- Signalquellen, bestehend aus dem Pull-Down-Netzwerk + Busverbindung mit Pull-Up-Element.
- Wenn alle Quellen 'Z', Bussignal 'H', sonst '0'.

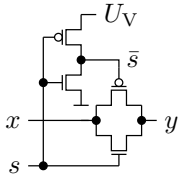


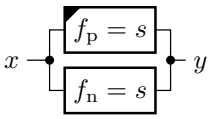
- Fehler-/Kollisionserkennung: NOR-Gatter, das kontrolliert, ob auf dem Bus eine '1' ankommt, wenn die aktive Quelle 'Z' sendet.

1.4 Transfergatter und Multiplexer

Transfergatter

Ein Transfergatter ist ein Schalter zur Weiterleitung einer »0« oder einer »1«. Es besteht aus einer Parallelschaltung eines NMOS- und eines PMOS-Transistors, die zueinander invertiert angesteuert werden[2].





x	s	y
-	0	Z
0	1	0
1	1	1

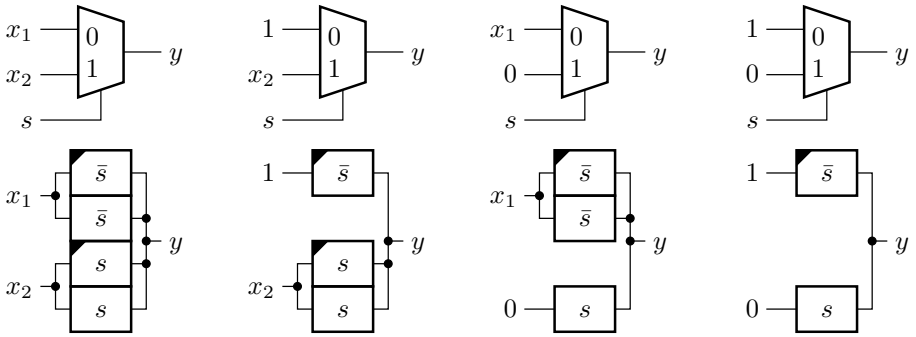
- Wert beliebig (don't care)

Ein 2:1-Multiplexer

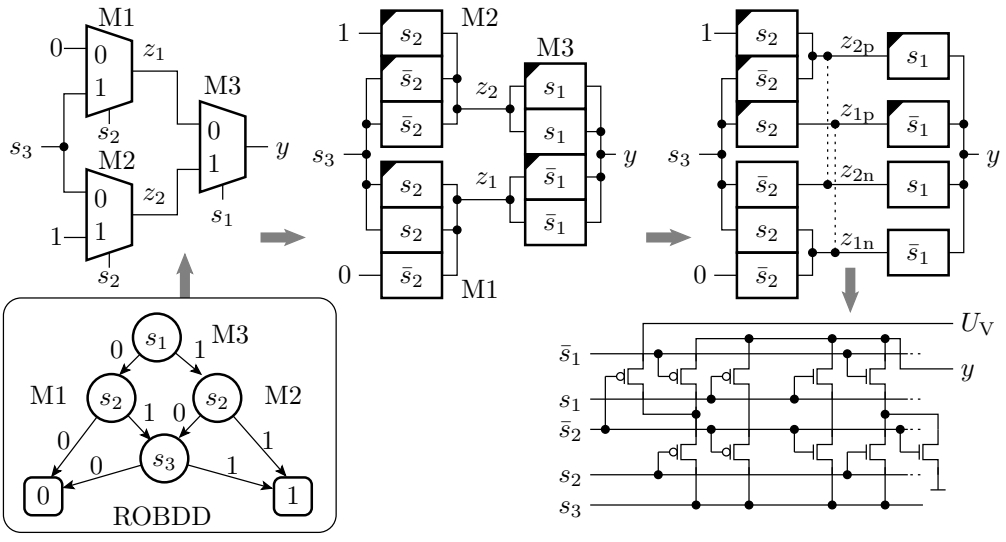
Funktion:

$$y = \begin{cases} x_1 & \text{wenn } s = 0 \\ x_2 & \text{wenn } s = 1 \end{cases}$$

Realisierbar aus zwei wechselseitig angesteuerten Transfergattern. Die Konstante null braucht keinen PMOS- und die Konstante eins keinen NMOS-Transistor zur Weiterleitung. Ein Umschalter zwischen null und eins hat dieselbe Schaltung wie ein Inverter.



ROBDD[3] ⇒ Multiplexer ⇒ Transistorschaltung

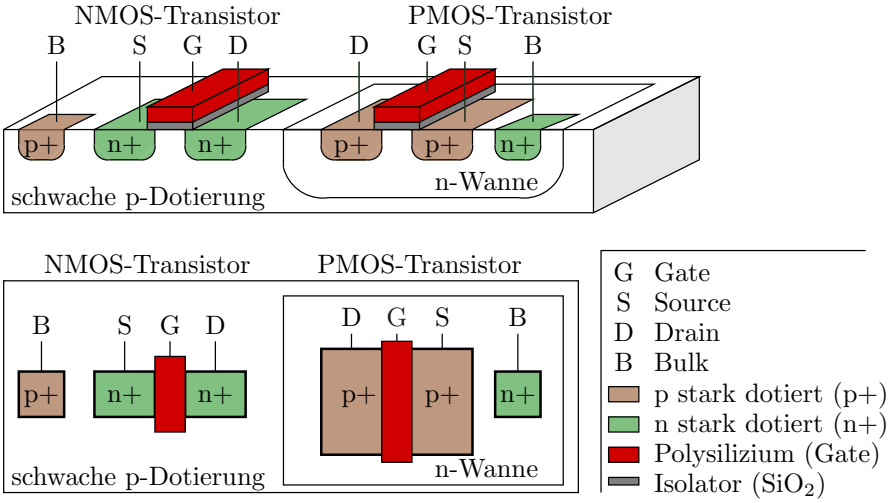


[2] Weiterleiten einer Eins mit einem NMOS-Transistor bzw. einer Null mit einem PMOS-Transistor dauert länger und liefert Ausgangsspannungen näher am verbotenen Bereich. Langsamer und unzuverlässiger.

[3] ROBDD **R**educed **O**rdered **B**inary **D**ecision **D**iagram – binäres Entscheidungsdiagramm zur Beschreibung einer logischen Funktion.

1.5 Geometrischer Entwurf

Geometrischer Entwurf

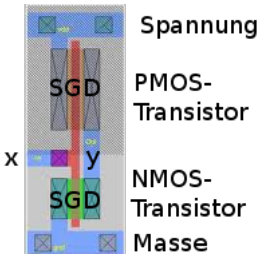


Anordnung von Flächenelementen. Die Schichtfolge in der Tiefe legt der Fertigungsprozess fest. Die Verdrahtung erfolgt darüber in mehreren isolierten Metallschichten mit Durchkontaktierungen.

Entwurf eines Inverters

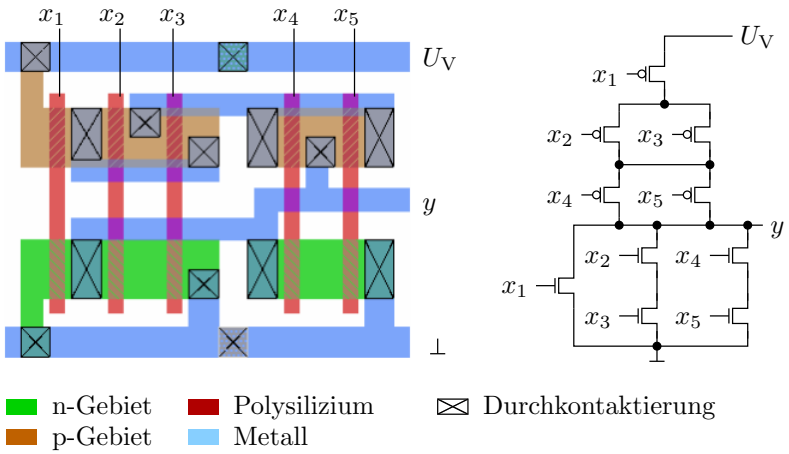
Die Source- und Drain-Gebiete sind parallel angeordnete Streifen. Das Gate ist ein Polysilizium-Streifen darüber, der bei der Fertigung als Maske für den Kanal zwischen den Source- und Draingebieten dient.

Durchkontaktierungen sind mit Metall gefüllte Löcher in Isolationsschichten. Jede Technologie hat Entwurfsregeln für Minimalabmessungen, Minimalabstände, Minimalüberlagerungen, ... die vor der Maskenerstellung automatisch geprüft werden.



Das Ziel des geometrischen Entwurfs ist eine platzsparende Anordnung:

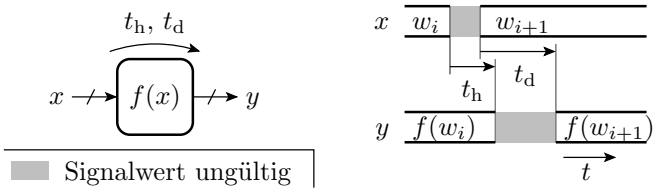
- Reihenschaltung: hochdotierter Streifen mit mehreren Gates.
- Parallelschaltung: zusätzliche Zwischenabgriffe.
- Trennung nach PMOS- und NMOS-Netzwerken.



2 Signalverzögerung

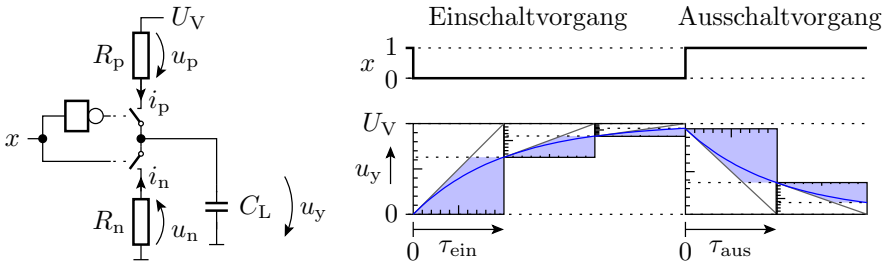
Signalverzögerung

Die Signalverzögerungen einer Verarbeitungsfunktion – Einzelgatter und Schaltungen aus Gattern – wurden bisher mit einem Toleranzschema aus einer Haltezeit t_h und einer Verzögerungszeit t_d beschrieben. Bei jeder Eingabeänderung bleibt das Ausgabesignal mindestens für die Haltezeit t_h unverändert. Eine neue gültige Eingabe bewirkt spätestens nach der Verzögerungszeit t_d eine neue gültige Ausgabe.



Dieser Abschnitt untersucht die Abhängigkeiten beider Zeitparameter von Schaltungs- und Geometriemerkmalen.

Modellierung als geschaltetes RC-Glied



$$i_\mathrm{p/n} = \frac{u_\mathrm{p/n}}{R_\mathrm{p/n}};\; u_\mathrm{C}(t) = u_\mathrm{C}(t=0) + \frac{1}{C_\mathrm{L}} \cdot \int_0^t i \cdot dt$$

$$u_\mathrm{y}(t) = \begin{cases} U_\mathrm{V} \cdot \left(1 - e^{-\frac{t}{\tau_\mathrm{ein}}}\right) & \text{mit } \tau_\mathrm{ein} = R_\mathrm{p} \cdot C_\mathrm{L} \\ U_\mathrm{V} \cdot e^{-\frac{t}{\tau_\mathrm{aus}}} & \text{mit } \tau_\mathrm{aus} = \mathrm{R_n} \cdot \mathrm{C_L} \end{cases}$$

(R_p, R_n – Widerstand des eingeschalteten PMOS- bzw. NMOS-Netzwerks; C_L – Lastkapazität).

$$u_\mathrm{y}(t) = \begin{cases} U_\mathrm{V} \cdot \left(1 - e^{-\frac{t}{\tau_\mathrm{ein}}}\right) & \text{mit } \tau_\mathrm{ein} = R_\mathrm{p} \cdot C_\mathrm{L} \\ U_\mathrm{V} \cdot e^{-\frac{t}{\tau_\mathrm{aus}}} & \text{mit } \tau_\mathrm{aus} = \mathrm{R_n} \cdot \mathrm{C_L} \end{cases}$$

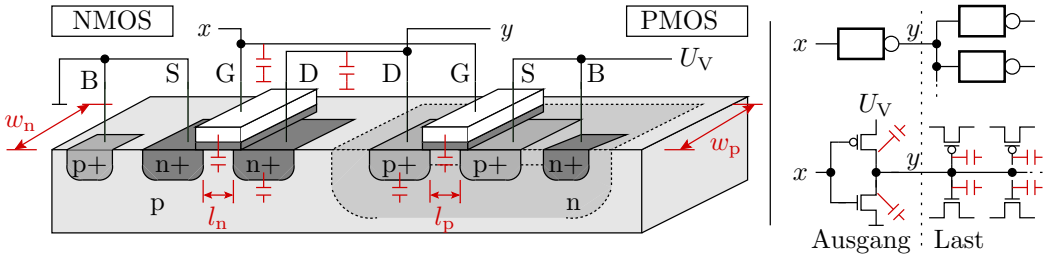
Für die nachfolgenden Überschläge:

- Annäherung der Ein- und Ausschaltzeit durch τ_ein und τ_aus
- Vernachlässigung, dass $R_\mathrm{p/n}$ von $u_\mathrm{p/n}$ abhängt[4].

[4] Vereinfacht die Lösung der DGL auf der Folie zuvor erheblich und führt am Ende zur selben empirischen Abschätzung der Gatterverzögerung.

2.1 Inverter

Einschaltwiderstände eines Inverters



$$R_p = 2^* \cdot R_{NQ} \cdot \frac{l_p}{w_p};\ R_n = R_{NQ} \cdot \frac{l_n}{w_n}$$

(R_{NQ} – Einschaltwiderstand NMOS-Transistor mit $w/l = 1$; 2^* – PMOS-Transistoren haben bei gleicher Geometrie etwa den doppelten Einschaltwiderstand).

Ausgangskapazität C_{LA} eines Inverters

Abschätzung über das Modell des Plattenkondensators:

$$C = \varepsilon \cdot \frac{A}{d}$$

(ε - Dielektrizitätskonstante; A – Fläche; d – Abstand).

Unter den Annahmen Drain-Breite und Sperrschichtdicke konstant:

$$C_{LA} = k_{CA} \cdot (w_{pA} + w_{nA})$$

($w_{...A}$ – Transistorbreiten Treiber; k_{CA} – Proportionalitätsfaktor).

Gate-Kapazitäten der getriebenen Lasttransistoren

Unter den Annahmen Drain-Breite und Sperrschichtdicke konstant, verhält sich die Summe der Gate-Kanal-Kapazitäten aller Lasttransistoren proportional zur Summe von deren Breiten:

$$C_{LL} = k_{CL} \cdot \left(\sum w_{pL} + \sum w_{nL}\right)$$

($\sum w_{...L}$ – Summe der Breiten der getriebenen Lasttransistoren; k_{CL} – Proportionalitätsfaktor).

Grund- und lastabhängige Verzögerung

$$C_L = C_{LA} + C_{LL} \approx k_{CA} \cdot (w_{pA} + w_{nA}) + k_{CL} \cdot \left(\sum w_{pL} + \sum w_{nL}\right)$$

$$t_{ein} \approx \frac{2 \cdot R_{NQ} \cdot l_{pA}}{w_{pA}} \cdot C_L;\ t_{aus} \approx \frac{R_{NQ} \cdot l_{nA}}{w_{nA}} \cdot C_L$$

$$t_{ein} \approx t_{aus} \approx \tau_A + \tau_L \cdot \frac{(\sum w_L)}{w_A}$$

Für $t_{ein} \approx t_{aus}$, d.h. $w_p \approx 2 \cdot w_n$ betragen Grundverzögerung:

$$\tau_A = \frac{R_{NQ} \cdot l_{nA} \cdot k_{CA} \cdot (w_{pA} + w_{nA})}{w_{nA}} = 3 \cdot R_{NQ} \cdot l_{nA} \cdot k_{CA}$$

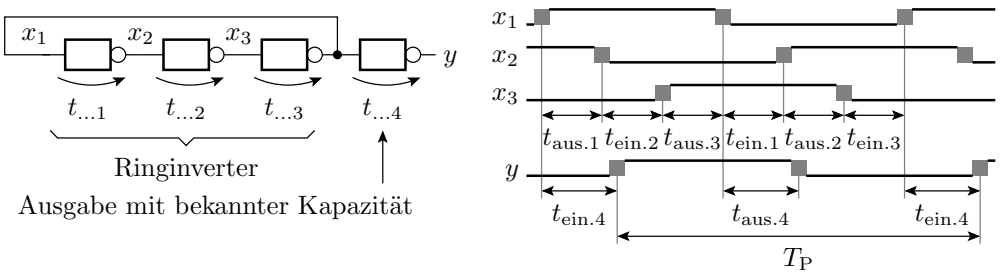
und lastabhängige Verzögerung (unabhängig von $w_{...}$):

$$\tau_L = \frac{R_{NQ} \cdot l_{nA} \cdot k_{CL} \cdot (\sum w_{pL} + \sum w_{nL})}{\sum w_{nL}} = 3 \cdot R_{NQ} \cdot l_{nA} \cdot k_{CL}$$

Schätzen von τ_A und τ_L mit Ringinvertern

Ein Ring aus einer ungeraden Anzahl von Invertern schwingt mit einer Periodendauer gleich der Summe aller Ein- und Ausschaltzeiten:

$$T_P = \sum_{i=1}^{N_{Inv}} (t_{ein.i} + t_{aus.i})$$

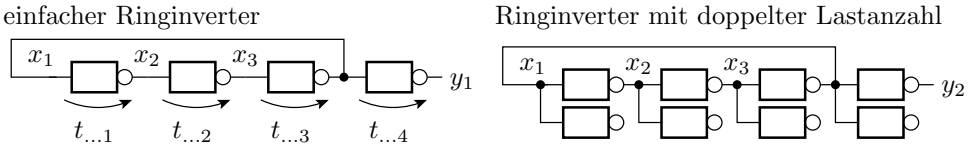


Bei vier identischen Invertern gilt für die Inverter mit einer Last $\sum w_L = w_A$ und für den Inverter mit zwei Lasten $\sum w_L = 2 \cdot w_A$:

$$\begin{aligned} T_P &= t_{aus.1} + t_{ein.2} + t_{aus.3} + t_{ein.1} + t_{aus.2} + t_{ein.3} \\ &= 6 \cdot \tau_A + 8 \cdot \tau_L \end{aligned}$$

Zur getrennten Bestimmung von τ_A und τ_L ist ein weiterer Ringinverter mit baugleichen Invertern und mehr Lasten erforderlich.

Im nachfolgenden Beispiel hat der zweite Ringinverter die doppelte Anzahl von Lasten:



$$\begin{aligned} T_{P1} &= 6 \cdot \tau_A + 8 \cdot \tau_L \\ T_{P2} &= \underbrace{2 \cdot (\tau_A + 2 \cdot \tau_L)}_{t_{ein.1}+t_{aus.1}} + \underbrace{2 \cdot (\tau_A + 2 \cdot \tau_L)}_{t_{ein.2}+t_{aus.2}} + \underbrace{2 \cdot (\tau_A + 4 \cdot \tau_L)}_{t_{ein.3}+t_{aus.3}} \\ &= 6 \cdot \tau_A + 16 \cdot \tau_L \end{aligned}$$

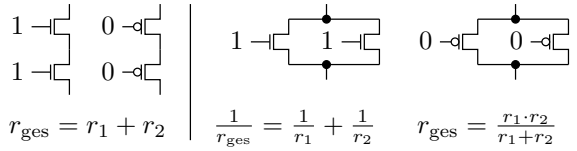
Daraus berechnet sich die Grund- und die lastabhängige Verzögerung wie folgt:

$$\begin{aligned} \tau_A &= \frac{1}{6} \cdot (2 \cdot T_{P1} - T_{P2}) \\ \tau_L &= \frac{1}{8} \cdot (T_{P2} - T_{P1}) \end{aligned}$$

2.2 Logikgatter

Gatter mit mehreren Eingängen

Bei Gattern mit mehreren Eingängen erfolgt die Auf- und Entladung der Lastkapazitäten über unterschiedliche Transistoren. Bei einer Reihenschaltung addieren sich die Ersatzwiderstände, über die der Gatterausgang umgeladen wird, und bei einer Parallelschaltung addieren sich die Kehrwerte.



(r – relative Widerstandserhöhung gegenüber einem Standard-NMOS- bzw. PMOS-Transistor).

Empirische Modellerweiterung

Unter der vereinfachten Annahme, dass die »Stockung« die Kapazität am Gatterausgang nicht ändert, erhöhen sich die Verzögerungen proportional zur Widerstandsänderung r_p bzw. r_n:

$$\begin{aligned} t_\mathrm{ein} &= c_\mathrm{p} \cdot r_\mathrm{p} \cdot \left(\tau_\mathrm{A} + \tau_\mathrm{L} \cdot \frac{(\sum w_\mathrm{L})}{w_\mathrm{A}} \right) \\ t_\mathrm{aus} &= r_\mathrm{n} \cdot \left(\tau_\mathrm{A} + \tau_\mathrm{L} \cdot \frac{(\sum w_\mathrm{L})}{w_\mathrm{A}} \right) \end{aligned}$$

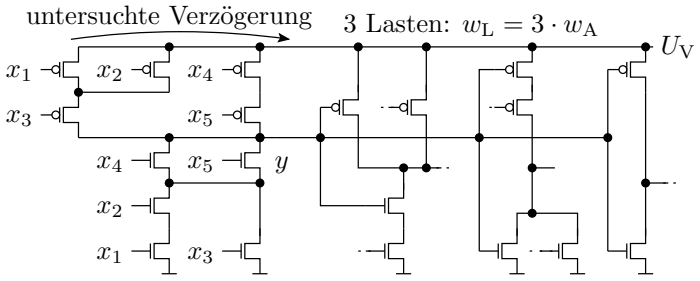
$c_\mathrm{p} = \frac{w_\mathrm{p}}{2w_\mathrm{n}}$ – Korrekturfaktor für die Breite der PMOS-Transistoren. Aus der üblichen Optimierung

$$t_\mathrm{ein.\,max} \approx t_\mathrm{aus.\,max}$$

ergibt sich als Richtwert für das Verhältnis der Transistorbreiten:

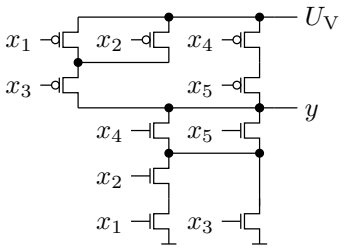
$$w_\mathrm{p} \approx 2 \cdot \frac{r_\mathrm{p.\,max}}{r_\mathrm{n.\,max}} \cdot w_\mathrm{n}$$

Beispielschaltung



- Alle NMOS-Transistoren und alle PMOS-Lasttransistoren seien Standardtransistoren.
- Alle PMOS-Transistoren des Treibergatters sollen die $2/c_\mathrm{P}$-fache Breite der NMOS-Standardtransistoren haben:

$$\begin{aligned} t_\mathrm{ein} &= c_\mathrm{P} \cdot r_\mathrm{p} \cdot (\tau_\mathrm{A} + 3 \cdot \tau_\mathrm{L}) \\ t_\mathrm{aus} &= r_\mathrm{n} \cdot (\tau_\mathrm{A} + 3 \cdot \tau_\mathrm{L}) \end{aligned}$$



x_5	x_4	x_3	x_2	x_1	r_p	r_n
0	1	0	1	⎍	$2 \cdot c_\mathrm{p}$	3
1	1	0	1	⎍	$2 \cdot c_\mathrm{p}$	2,5
1	1	⎍	0	1	$2 \cdot c_\mathrm{p}$	1,5
1	0	⎍	0	0	$1{,}5 \cdot c_\mathrm{p}$	2
		...			...	...

anzupassende Maxima

Der Parameter c_P ist so festzulegen, das die maximale Einschaltzeit gleich der maximalen Ausschaltzeit und damit gleich der Gatterverzögerungszeit ist, im Beispiel $c_\mathrm{p} = 1{,}5$ ($w_\mathrm{p} = \frac{4}{3} \cdot w_\mathrm{n}$):

- maximale Verzögerung: $t_\mathrm{d} \leq 3 \cdot (\tau_\mathrm{A} + 3 \cdot \tau_\mathrm{L})$
- minimale Haltezeit: $t_\mathrm{h} \geq 1{,}5 \cdot (\tau_\mathrm{A} + 3 \cdot \tau_\mathrm{L})$

$t_\mathrm{h} \ll t_\mathrm{d}$ für Gatter mit parallelen Transistoren unvermeidbar.

2.3 Puffer

Gepufferte CMOS-Gatter

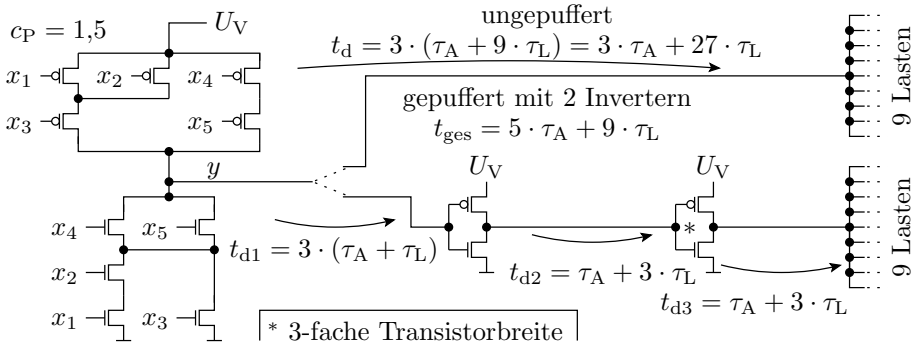
- Die Verzögerung erhöht sich gegenüber einem einfachen Inverter als Treiber um die Stockungstiefe s (Anzahl der in Reihe geschalteten Transistoren) im NMOS-Netzwerk:

$$t_{\mathrm{d}} = s \cdot (\tau_{\mathrm{A}} + N_{\mathrm{L}} \cdot \tau_{\mathrm{L}})$$

- Die lastabhängige Verzögerung wächst mit dem Produkt aus Stockungstiefe s und Lastzahl $N_{\mathrm{L}} = \frac{(\sum w_{\mathrm{L}})}{w_{\mathrm{A}}}$. Wenn alle PMOS- und NMOS-Transistoren gleichbreit sind, ist die Lastanzahl die Anzahl der getriebenen Gattereingänge.

- Zur Minimierung der Verzögerung werden zwischen Gattern mit großer Stockungstiefe und großer Lastanzahl Inverter eingefügt.

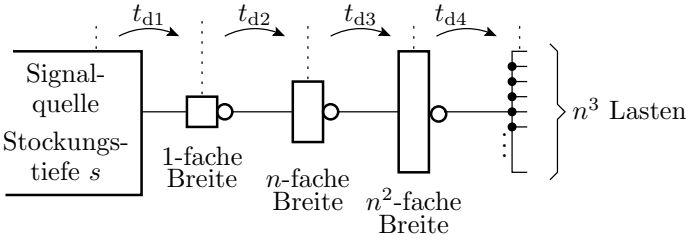
Pufferung



Im Beispiel verringern die zwei zusätzlichen Inverter die Gatterverzögerung von $3{\cdot}\tau_{\mathrm{A}}{+}27{\cdot}\tau_{\mathrm{L}}$ auf $5{\cdot}\tau_{\mathrm{A}}{+}9{\cdot}\tau_{\mathrm{L}}$. Insbesondere ein Problem bei Taktleitungen für tausende Abtastregister, Schreib-, Auswahl- und Leseleitungen in Speichern, an die hunderte Speicherzellen angeschlossen sind, ...

Treiber für große Lastanzahl

Signale für sehr viele Lasten, z.B. Takt- und Initialisierungssignale, werden mit Treiberbäumen erzeugt, in denen sich die Transistorbreiten nach einer geometrischen Reihe erhöhen:



$$t_{\mathrm{d1}} = s \cdot (\tau_{\mathrm{A}} + \tau_{\mathrm{L}}) \qquad t_{\mathrm{d2}} = \tau_{\mathrm{A}} + n \cdot \tau_{\mathrm{L}}$$
$$t_{\mathrm{d3}} = \tau_{\mathrm{A}} + n \cdot \tau_{\mathrm{L}} \qquad t_{\mathrm{d4}} = \tau_{\mathrm{A}} + n \cdot \tau_{\mathrm{L}}$$

Gesamtverzögerung: $t_{\mathrm{d.ges}} = (3 + s) \cdot \tau_{\mathrm{A}} + (s + 3 \cdot n) \cdot \tau_{\mathrm{L}}$

3 Latches und Register

Latches und Register

Latches und Register dienen

- zur Abtastung der Signalwerte zu definierten Zeitpunkten und

- Aufbewahrung von Signalwerten über eine längere Zeit.

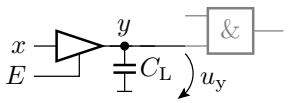
Besonderer Augenmerk ist auf die Signale, die die Zeitabläufe in einer Schaltung steuern zu legen:

- Taktsignale für Register,
- Übernahmesignale für Latches und
- asynchron wirkende Initialisierungssignale.

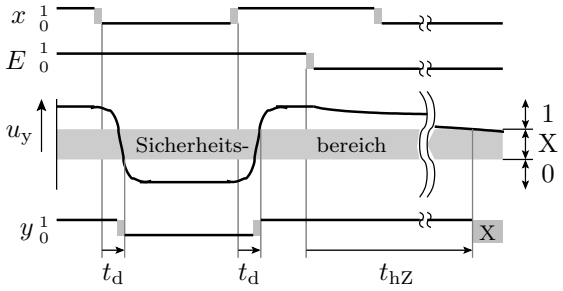
Diese müssen glitchfrei und mit tolerierbarem Zeitversatz an allen angesteuerten Speicherelementen ankommen (siehe auch später Taktversorgung).

3.1 Speicherzellen

Das Prinzip einer dynamischen Speicherzelle



E	x	y
0	0	Z
0	1	Z
1	0	0
1	1	1



Jedes Signal hat eine kurze Haltezeit, die durch Deaktivierung des Treibers (hochohmig schalten, Ausgabe Signalwert 'Z') bis zu mehreren Millisekunden verlängert werden kann. Danach muss der Inhalt aufgefrischt (gelesen und neu geschrieben) werden.

Das Prinzip einer statischen Speicherzelle

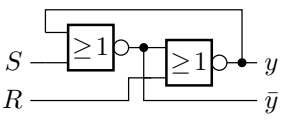
Der Kern einer statischen Speicherzelle ist ein Ring aus zwei Invertern, der in einen von zwei stabilen Zuständen kippt und seinen Zustand beibehält. Eine solche bistabile Kippstufe wird auch als Flipflop bezeichnet.

- Wie lässt sich der Zustand einstellen und ändern?

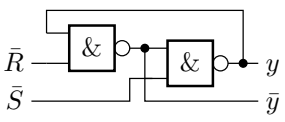
RS-Flipflop

Bei Ersatz der Inverter durch NOR- bzw. NAND-Gatter erhält das Flipflop einen Setz- und einen Rücksetzeingang. Gleichzeitiges Setzen und Rücksetzen ist verboten, weil dann die negierte gleich der direkten Ausgabe ist und das Flipflop bei zeitgleicher Deaktivierung zufällig in einen der beiden Zustände kippt.

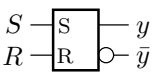
RS-Flipflop (NOR)



RS-Flipflop (NAND)



Schaltsymbol des RS-Flipflops

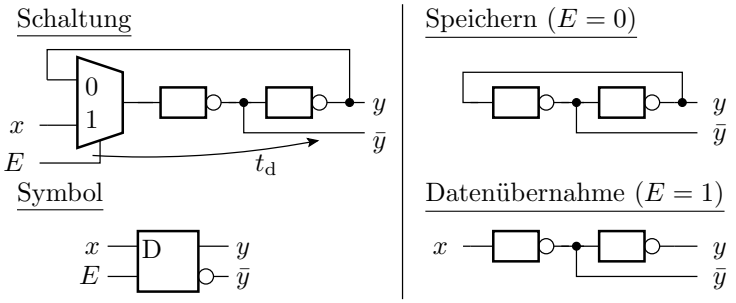


Ansteuertabelle

R	S	y
0	0	Speichern
0	1	1
1	0	0
1	1	Vermeiden

D-Flipflop

Bei einem D-Flipflop wird der Inverterring zum Beschreiben mit einem Multiplexer aufgetrennt. Während ein RS-Flipflop auf beiden Steuereingängen glitch-empfindlich ist, ist beim D-Flipflop nur der Freigabeeingang E empfindlich gegenüber Glitches[5].



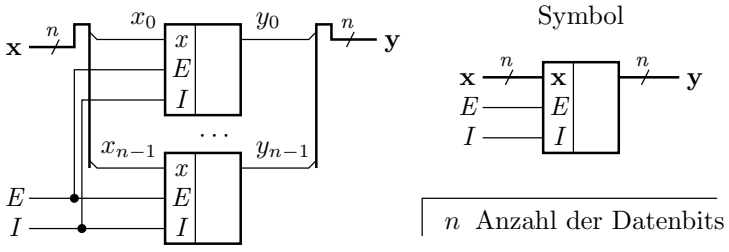
D-Flipflop mit Initialisierungseingang

Eine Speicherzelle hat nach Zuschalten der Versorgungsspannung einen unbestimmten Anfangszustand (unbestimmt, 'U'). Speicherzellen insbesondere für die Zustandsspeicherung von Automaten benötigen deshalb einen zusätzlichen Setz- oder Rücksetzeingang, realisiert durch Erweiterung eines der Inverter zu einem NAND- oder NOR-Gatter.

3.2 Latches

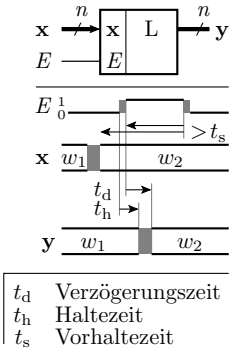
Latch

Ein Latch ist eine Zusammenfassung von $n \geq 1$ D-Flipflops mit gemeinsamem Freigabe- und bei einem initialisierbarem Latch, auch mit gemeinsamen Initialisierungseingang. In Abgrenzung zu einem Register erfolgt die Datenübernahme zustandsgesteuert.



Zeitverhalten

- Übernahmeverzögerung um t_h und t_d.
- Speicherung gültiger Werte nur, wenn bei Deaktivierung von E alle Eingabesignale für mindestens eine Vorhaltezeit t_s stabil anliegen.



[5] Glitches auf dem R- und S-Signal können ein ungewolltes Rücksetzen oder Setzen und Glitches auf dem E-Signal eine Übernahme zu einem falschen Zeitpunkt bewirken.

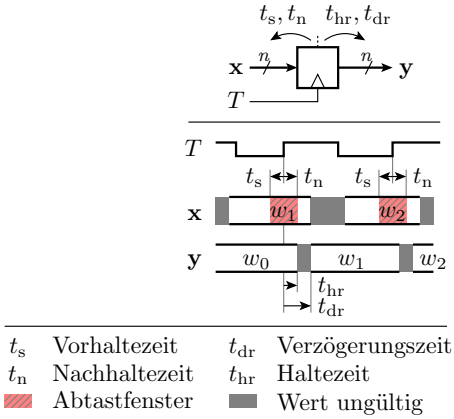
```
process(x, E)
begin
 if rising_edge(E) or (E='1' and x'event) then
  y <= <ungueltig> after th,
       x after td;
 elsif falling_edge(E) and (E'last_event < ts
       or x'last_event < ts) then
  y <= <ungueltig>;
 end if;
end process;
```

(Siehe hierzu auch Foliensatz EDS-F2.)

3.3 Register

Register und sein Zeitverhalten

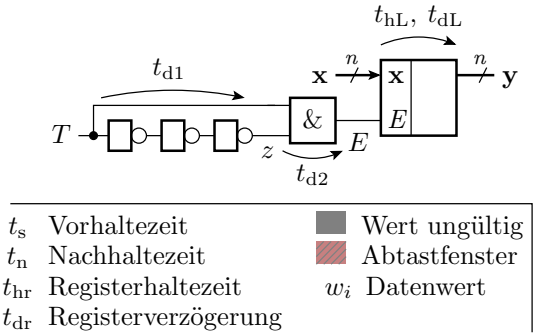
```
process(T)
begin
 if rising_edge(T) then
  y <= <ungueltig> after
            thr, x after tdr;
  -- Kontrolle Abtastfenster
  wait for tn;
  if x'last_event < ts + tn then
   y <= <ungueltig>;
  end if;
 end if;
end process;
```

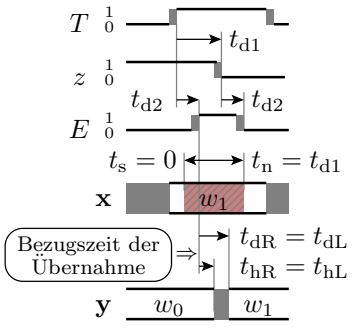


- Halte- und Verzögerungszeit beziehen sich auf die Taktflanke.
- Signalwechsel und ungültige Werte im Abtastfenster invalidieren den gespeicherten Wert. (Siehe auch Foliensatz EDS-F2.)

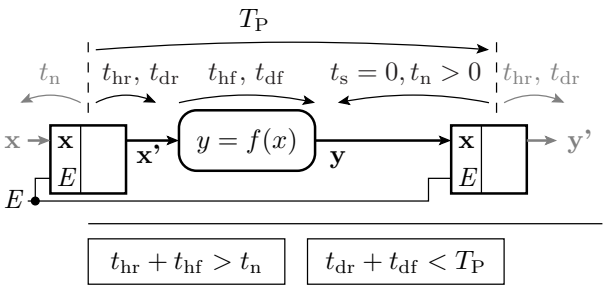
Gepulstes Latch zur Taktflankenübernahme

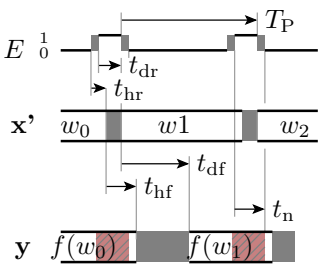
Bei Ansteuerung mit kurzen Freigabeimpulsen kann ein Latch auch als Register dienen. Die Freigabeimpulse können z.B. mit der dargestellten Schaltung zur Glitch-Erzeugung aus dem Takt gebildet werden. Das so gebildete Register benötigt keine Vorhaltezeit $t_s = 0$, dafür aber eine Nachhaltezeit $t_n > 0$.





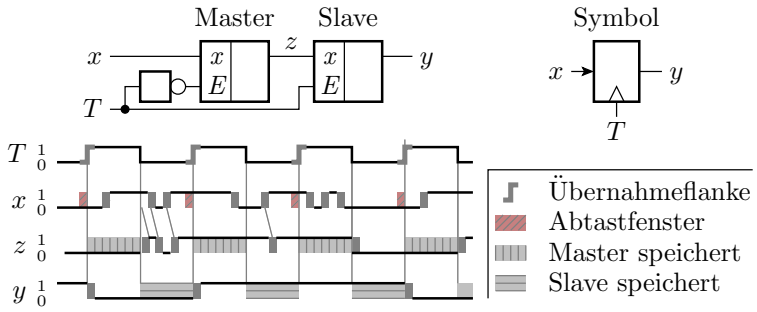
Register-Transfer-Funktion mit gepulsten Latches





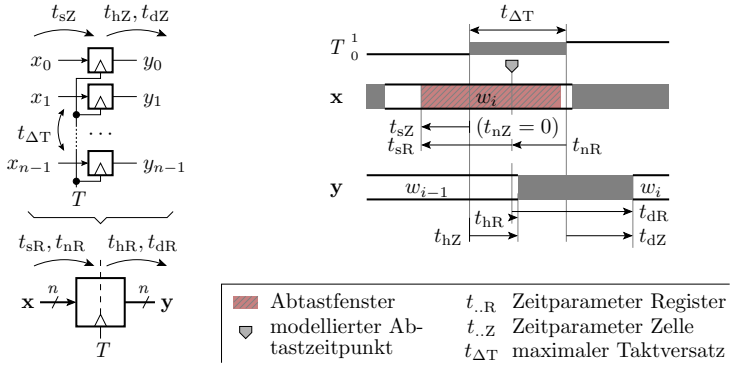
Wegen der Nachhaltezeit des Latches $t_n > 0$ ist von der Verarbeitungsfunktion eine Mindesthaltezeit zu fordern. Eine Schaltung mit gepulsten Latches als Register ist laufzeitkritischer als eine mit Master-Slave-Flipflops als Register (siehe nächste Folie). Da Zuverlässigkeit und Robustheit gegen Fehler in der Digitaltechnik meist wichtiger als Chipfläche sind, sind Schaltungen mit gepulsten Latches als Register weniger gebräuchlich.

Master-Slave-Flipflop



Ein Master-Slave-Flipflop besteht aus zwei Flipflops. Vor der aktiven Taktflanke übernimmt der Master und der Slave speichert. Nach der aktiven Taktflanke gibt der Master die gespeicherten Daten an den Slave weiter. Die Eingabedaten brauchen eine Vorhaltezeit ($t_s > 0$), aber keine Nachhaltezeit ($t_n = 0$).

Register mit Taktversatz



Bei einem Register besteht zwischen den Takteingängen der einzelnen Speicherzellen immer ein geringer Taktversatz. Das Unsicherheitsfenster wird mit in der Vor- und Nachhaltezeit des Registers berücksichtigt.

Frühestmögliche Taktflanke als Bezugszeitpunkt:

$$t_{sR} = t_{sZ} \qquad t_{hR} = t_{hZ}$$
$$t_{nR} = t_{\Delta T} \qquad t_{dR} = t_{\Delta T} + t_{dZ}$$

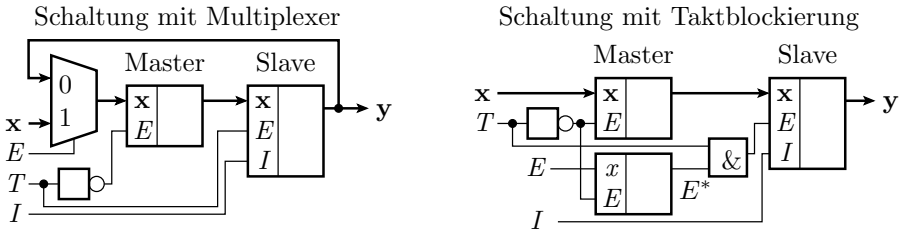
Spätestmögliche Taktflanke als Bezugszeitpunkt:

$$t_{sR} = t_{sZ} + t_{\Delta T} \qquad t_{hR} = t_{hZ} - t_{\Delta T}$$
$$t_{nR} = 0 \qquad t_{dR} = t_{dZ}$$

(Index R steht für Register und Index Z für Speicherzelle.)

Freigabe- und Initialisierungseingang

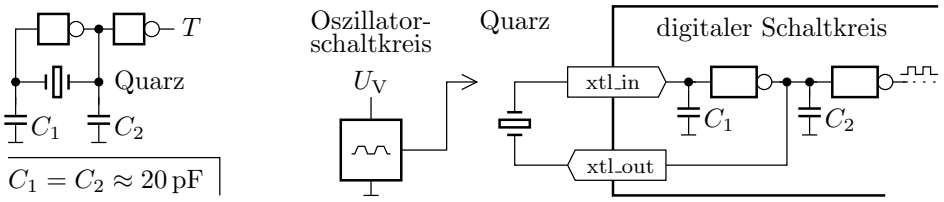
```
signal x, y: std_logic_vector(n-1 downto 0);
signal E, I, T: std_logic;
...
process(I, T):
begin
 if I='1' then  y <= aw;
 elsif E='1' and rising_edge(T) then  y<=x;
 end if;
end process;
```



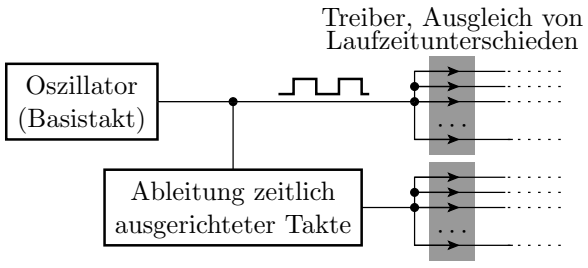
3.4 Taktversorgung

Oszillator

Takte werden mit einem RC- oder Quarzoszillator erzeugt. Bei einem Quarzoszillator wird ein Inverter mit einem Quarz oder einem keramischen Schwinger rückgekoppelt. Quarzoszillatoren sind sehr frequenzstabil. Die relativen Abweichungen von der Sollfrequenz betragen $\approx 10^{-5}$. Am Eingang für den Anschluss des Quarzes an einen Schaltkreis kann auch ein Oszillator oder eine andere Taktquelle angeschlossen werden.



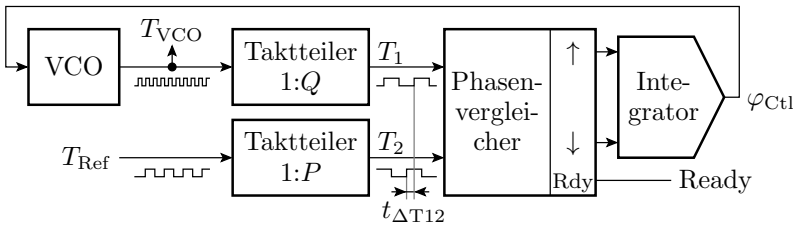
Taktnetze



Taktnetze bestehen aus Treibern und Verbindungsnetzwerken, die die Taktsignale glitchfrei und mit geringem Taktversatz an alle Speicherzellen der Register verteilen. Für die Spartan-FPGAs aus der Übung sind Taktsignale entweder über Schaltkreisanschlüsse einzuspeisen, die Zugang zum Taktnetz haben, oder über BUFG-Treiber zu leiten. Sonst gibt es eine Warnung »kombinatorisch gebildeter Takt o.ä.[6].

Phasenregelkreise

[6] Ignorieren dieser Warnungen kann schwer lokalisierbare Fehlfunktionen zur Folge haben.



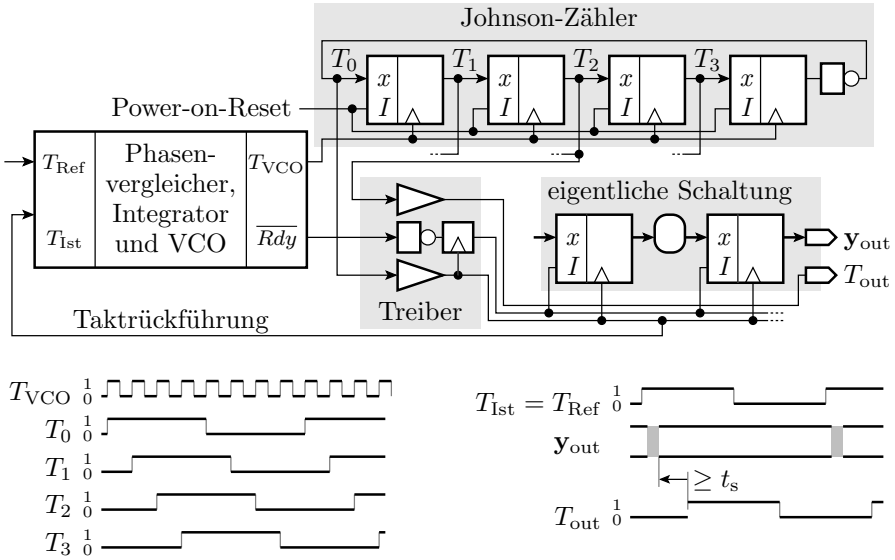
Der von einen spannungsgesteuerten Oszillator (VCO **V**oltage **C**ontrolled **O**scillator) erzeugte und optional durch Q geteilte Takt wird mit dem optional durch P geteilten Referenztakt verglichen. Bei Phasenvorlauf von T_1 gegenüber T_2 verringert und bei Phasennachlauf erhöht der Integrator die VCO-Spannung und regelt so die VCO-Frequenz ein auf:

$$f_{\mathrm{VCO}} = f_{\mathrm{Ref}} \cdot \frac{Q}{P}$$

Phasenregelkreise werden zur Frequenzvervielfachung und Phasenausrichtung genutzt. Achtung, den Rest der Schaltung nach Spannungszuschaltung erst initialisieren und benutzen, wenn der Regelkreis mit Ready Phasengleichheit signalisiert.

Definierte Phasenverschiebungen (Zeitversätze) zwischen Taktsignalen sind z.B. mit einem Johnson-Zähler als VCO-Taktteiler erzeugbar (siehe nächste Folie). Gezeigt wird dort auch eine Lösung zur zielgerichteten Ausrichtung der aktiven Taktflanke eines Ausgabetakts zu den Änderungen ausgegebener Daten (vergl. F3, Abschn. 4.4 Datensynchronisation).

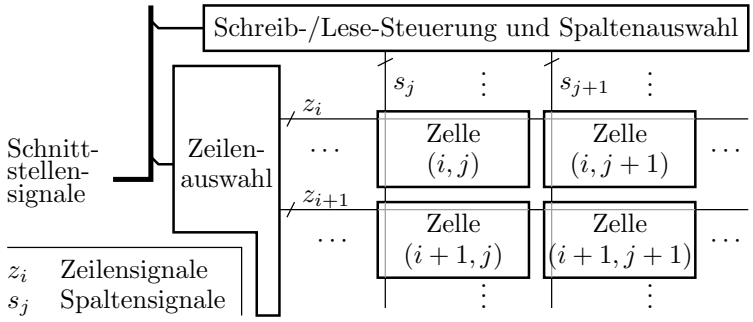
Erzeugung phasenkorrigierter Taktsignale



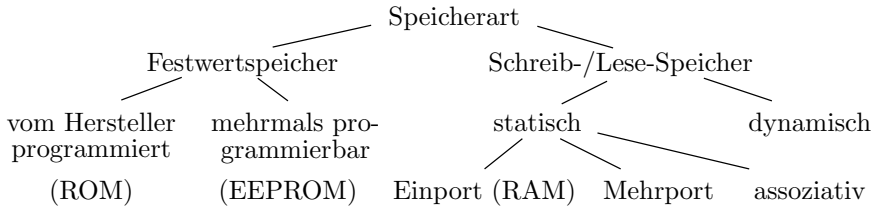
4 Blockspeicher

Blockspeicher

Ein Blockspeicher besteht im Kern aus einer regelmäßigen 2D-Anordnung von flächenminimierten Speicherzellen. Die Grundfunktionen (nur lesbar, beschreib- und lesbar, ...) hängen von der Zellenfunktion ab. Die Zeilen- und Spaltenauswahl legt die Zugriffsmöglichkeiten fest (Zugriffsbreite, Portanzahl, ...) .



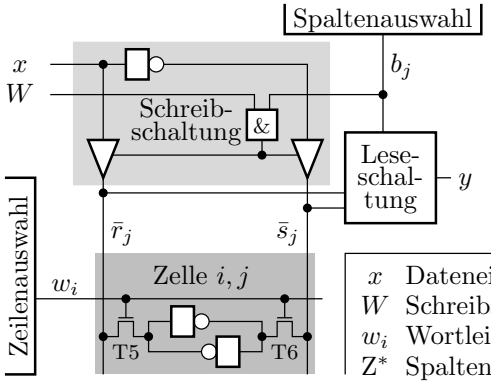
Speicherarten



- Festwertspeicher: nur einmal oder nur mit großem Zeitaufwand beschreibbar. Datenerhalt auch ohne Versorgungsspannung über Jahre.
- Schreib/Lese-Speicher: Schreibzeit in der Größenordnung der Lesezeit; Datenverlust ohne Betriebsspannung.
- Dynamische Speicher: Speichern in Kapazitäten; sehr hohe Speicherdichte, verlieren ihre Daten ohne Auffrischen nach wenigen Millisekunden.
- Mehrportspeicher: paralleler wahlfreier Zugriff.
- Assoziativspeicher: zusätzliche parallele Suchfunktion.

4.1 SRAM

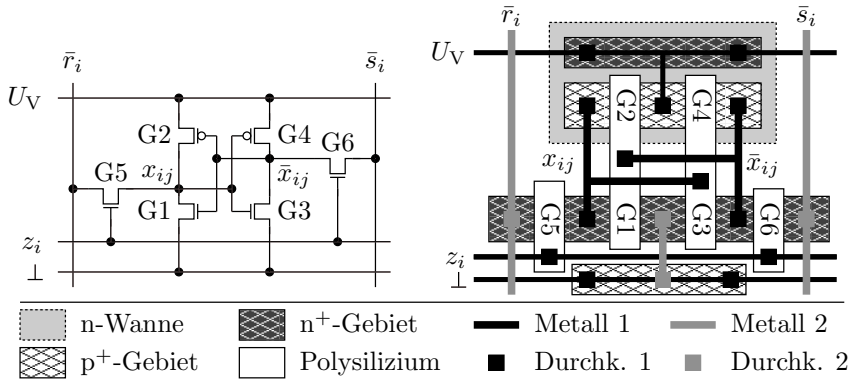
Statische Schreib/Lese-Speicher (SRAM)



	x	W	b_j	$\bar{r}_j$	$\bar{s}_j$
setzen	1	1	1	1	0
rücksetzen	0	1	1	0	1
lesen	-	0	1	Z*	Z*
keine Operation	-	-	0	Z*	Z*

x	Dateneingang	b_j	Bitleitung
W	Schreibsignal	y	Datenausgang
w_i	Wortleitung	-	beliebig (don't care)
Z*	Spaltentreiber deaktiviert		

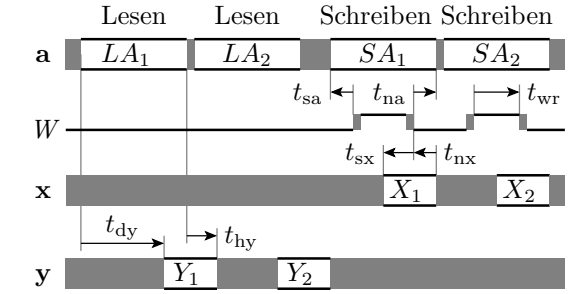
Die Speicherzelle ist ein Inverterring mit zwei Transistoren für die Lese-Schreib-Auswahl. Bei aktiviertem Zeilenauswahlsignal ist die Zelle mit den Spaltenleitungen verbunden, über die sie gesetzt, gelöscht oder gelesen werden kann.



Bei Zeilenauswahl ($z_i = 1$) bewirken die Spaltenansteuerungen

- $\bar{r} = \bar{s} =$ 'Z': Zelle kann gelesen werden.
- $(\bar{r} = 0) \wedge (\bar{s} =$ H): Schreiben einer null (Rücksetzen) und
- $(\bar{r} =$ H$) \wedge (\bar{s} = 0)$: Schreiben einer eins (Setzen).

Signalverläufe zum Lesen und Schreiben

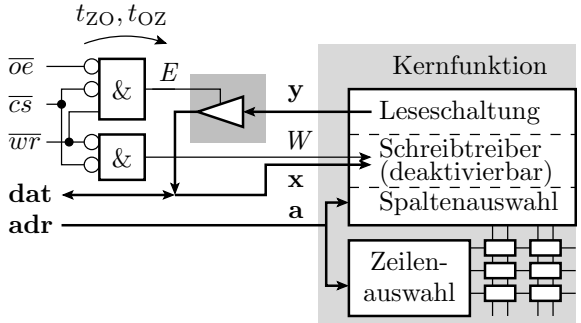


LA_i	Leseadresse
SA_i	Schreibadresse
Y_i	gelesener Wert
X_i	zu schreibender Wert

a	Adresssignal
x	Eingangsdaten
W	Schreibsignal
y	Ausgabedaten
t_{hy}	Lesehaltezeit
t_{dy}	Leseverzögerungszeit
t_{sa}	Adressvorhaltezeit
t_{na}	Adressnachhaltezeit
t_{sx}	Datenvorhaltezeit
t_{nx}	Datennachhaltezeit
t_{wr}	Schreibpulsbreite

Die Halte- und Verzögerungszeiten von Speichern sind im Vergleich zu Gattern sehr groß, weil an den Zeilen- und Spaltenleitungen viele Lasten hängen. Das Schreibsignal darf nur bei gültiger Adresse und Einhaltung der Vor- und Nachhaltebedingungen aktiviert werden, sonst Risiko für schreiben auf falsche Adresse.

Modell eines RAM-Schaltkreises



Schaltkreisanschlüsse:

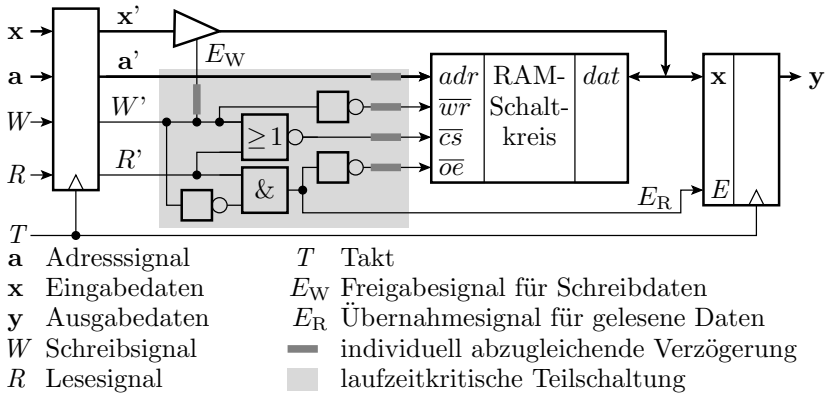
$\overline{oe}$	Ausgabefreigabesignal
$\overline{cs}$	Schaltkreisauswahlsignal
$\overline{wr}$	Schreibsignal
adr	Adresssignal
dat	bidirektionales Datensignal

Verzögerung des Ausgabetreibers:

t_{OZ}	Deaktivierungsverzögerung
t_{ZO}	Aktivierungsverzögerung (im Beispiel unten 2×4 ns)

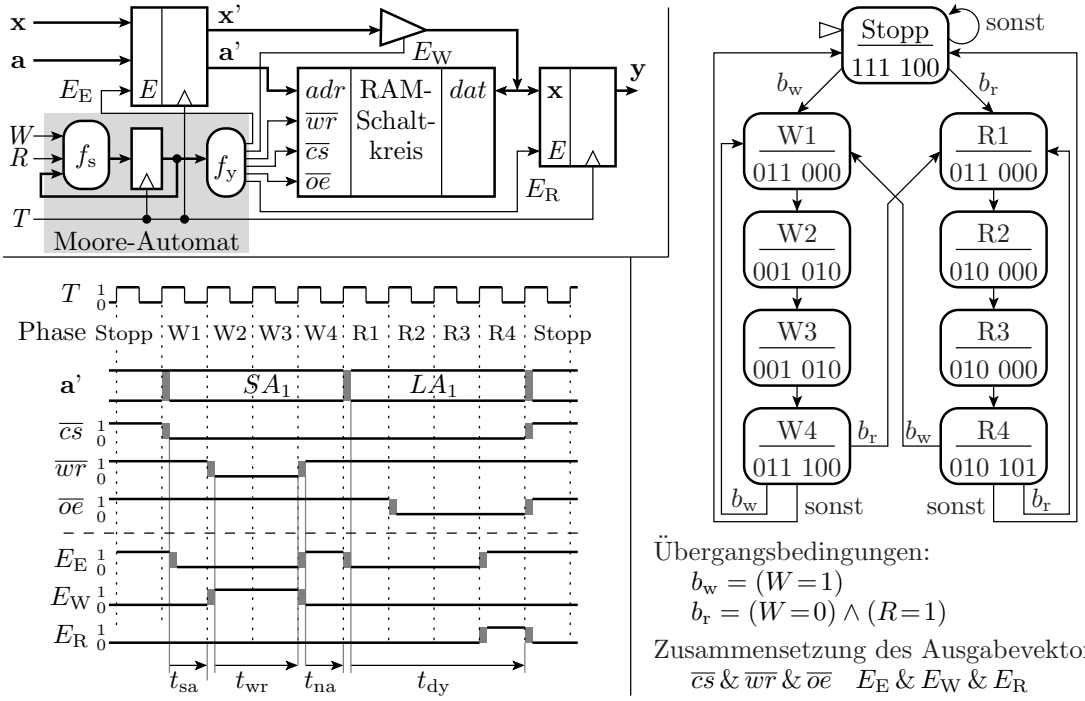
Ein RAM als Schaltkreis hat zusätzlich eine Schnittstellenschaltung. Asynchrone RAMs haben typisch einen bidirektionalen Datenbus für die Ein- und Ausgabe und die low-aktiven Steuersignaleingänge $\overline{cs}$ (**C**hip **S**elect), $\overline{wr}$ (**Wr**ite) und $\overline{oe}$ (**O**utput **E**nable). Das Simulationsmodell besteht aus dem Modell für die Kernschaltung und der Beschreibung der Schnittstellenschaltung.

Schneller laufzeitkritischer RAM-Controller



Für die Nutzung eines SRAMs ist eine Schnittstellenschaltung mit dem grau unterlegten laufzeitkritischen Teil zu entwickeln. Die Laufzeiten dieser Teilschaltung müssen auf wenige Nanosekunden genau stimmen. Das geht nur mit einer Handverdrahtung mit Constraints oder dem MIG (**M**emory **I**nterface **G**enerator).

Einfach zu entwerfender RAM-Controller

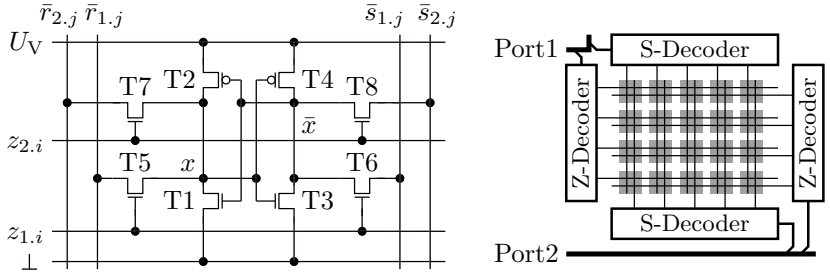


In dieser Lösung ist die Schaltung zur Generierung der Steuersignale für das Schreiben und Lesen ein Automat, dessen Ausgabeänderungen an Taktflanken ausgerichtet sind. Damit sind die Zeitbedingungen viel einfacher einzuhalten, insbesondere ohne eine Handverdrahtung.

4.2 Mehrport- und Assoziativspeicher

Mehrportspeicher

Ein Mehrportspeicher erlaubt den zeitgleichen unabhängigen Zugriff auf Speicherplätze unterschiedlicher Adresse. Das verlangt Speicherzellen mit mehreren Paaren von Auswahltransistoren (4 Transistoren für den Inverterring und 2 Transistoren für jeden Port), einen Satz von Zeilen- und Spaltenleitungen sowie einen Zeilen- und einen Spalten-Decoder je Port.

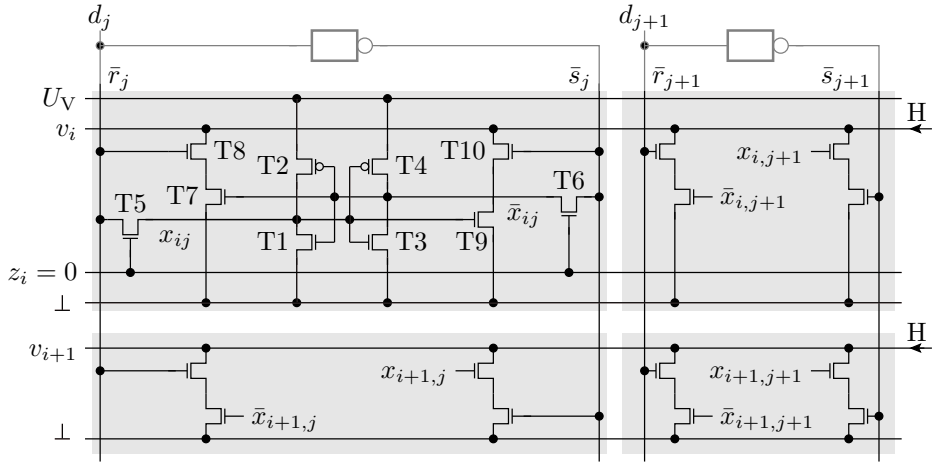


Assoziativspeicher

Ein Assoziativ- oder inhaltsadressierbarer Speicher ist ein Schreib-Lese-Speicher (RAM) mit einer zusätzlichen parallelen Vergleichsfunktion des Eingabevektors mit den Inhalten aller Speicherplätze. In der Zusatzbetriebsart »Vergleich« werden die Eingabedaten mit den zuvor geschrieben Inhalten aller Speicherplätze verglichen und die Adresse mit der ersten Übereinstimmung ausgegeben. Dazu erhält jede Speicherzelle zusätzlich 4 Transistoren für den Pull-Down-Zweig eines minimierten Bitvergleichers. Zeilenweises AND der Signale für Bitgleichheit.

Einsatz in Rechnern:

- Übersetzungspuffer von virtuellen in physikalische Adressen und
- Tag-Speicher[7] für mehrfach assoziative Caches.



- Vergleichsschaltung auf Zellenebene (T7 bis T10)

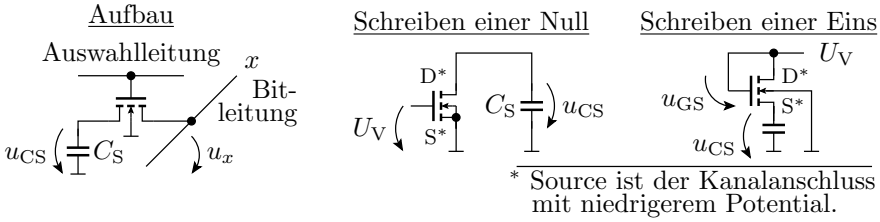
$$f_{ij} = (\bar{r}_j \wedge \bar{x}_{ij}) \vee (\bar{s}_j \wedge x_{ij}) = (d_j \wedge \bar{x}_{ij}) \vee (\bar{d}_j \wedge x_{ij})$$

- Parallelschaltung innerhalb der Zeile: $f_i = \bigvee_{j=1}^{N_S} f_{ij}$

4.3 DRAM

DRAM-Zelle

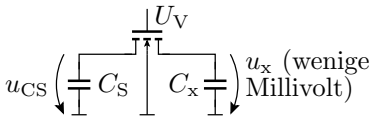
DRAMs (**D**ynamic **R**andom **A**ccess **M**emory) haben die kleinsten mit MOS-Technik realisierbaren Speicherzellen (nur ein NMOS- Transistor je Bit), sind dafür aber kompliziert anzusteuern.



[7] Der Tag ist Adressteil der Cache-Einträge.

Zum Schreiben wird der Schreibwert
auf die Datenleitung gelegt und der Transistor eingeschaltet. Zum Schreiben einer Null arbeitet der Transistor normal, aber beim Schreiben einer Eins in einer »unvorteilhaften« Betriebsart, in der die Speicherkapazität C_S nur auf eine Spannung $u_\mathrm{CS} < U_\mathrm{V}$ aufgeladen wird (U_V – Versorgungsspannung; u_CS – Zellenspannung).

Lesen einer DRAM-Zelle



Das Lesen beginnt mit einem Ladungsausgleich zwischen Zelle und Datenleitung. Wegen der viel größeren Kapazität der Datenleitung C_x gegenüber der Kapazität der Speicherzelle C_S verringert sich die Amplitude auf wenige Prozent:

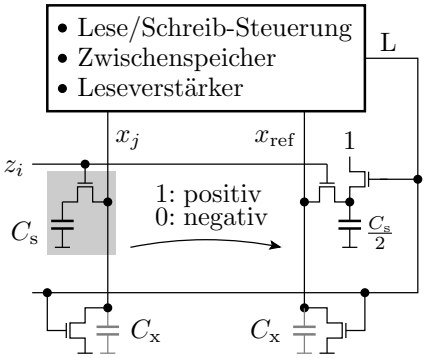
$$U_\mathrm{x}^{(+)} = \frac{C_\mathrm{S}}{C_\mathrm{S}+C_\mathrm{x}} \cdot \begin{cases} 0 & \text{für eine gespeicherte »0«} \\ u_\mathrm{CS1} & \text{für eine gespeicherte »1«} \end{cases}$$

Damit die Amplitudendifferenz auswertbar ist, darf an einer Datenleitung nur eine begrenzte Anzahl von Speicherzellen angeschlossen sein. Die Datenleitung muss vor jedem Lesevorgang entladen werden. Die Auswertung erfolgt über Differenzbildung.

DRAMs haben zwei symmetrische Hälften. Auf der einen Seite wird die Speicherzelle und auf der anderen Seite eine geladene Referenzzelle mit der halben Kapazität entladen. Die Spannungsdifferenz wird mit einer bistabilen Kippschaltung ähnlich einem RS-Flipflop ausgewertet. Das Lesen löscht die gespeicherte Information. Deshalb müssen die gelesenen Werte zurückgeschrieben werden.

Gesamter Leseablauf

- Datenleitungen entladen und Referenzkapazität aufladen.
- Ladungsausgleich.
- Differenz auswerten.
- Gelesene Werte zurückschreiben.



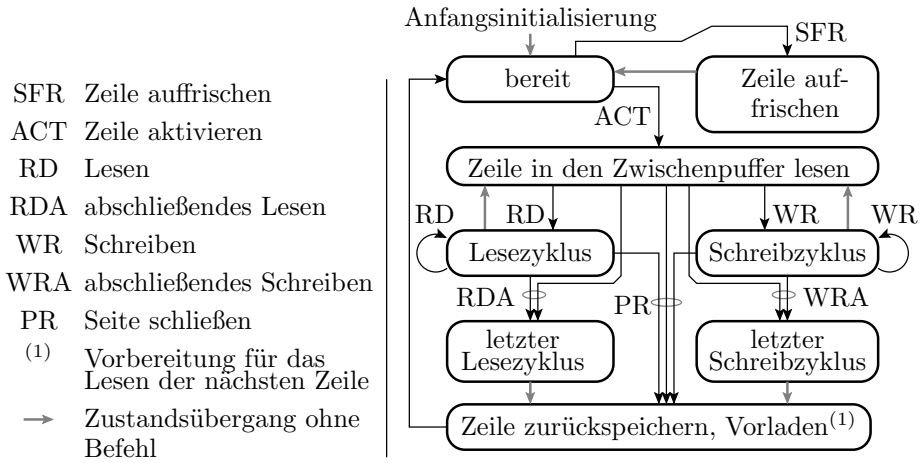
Bei jedem Lesevorgang wird eine komplette Zeile gelesen und wieder zurückgespeichert, egal wie viele Bits davon benötigt werden.

Zum Schreiben wird die Zeile mit dem adressierten Platz komplett gelesen, die zu schreibenden Bits verändert und zurückgeschrieben.

Eine DRAM-Zelle hat nur eine begrenzte Datenhaltezeit von wenigen Millisekunden. In dieser Zeit muss jede Zelle einmal gelesen und zurückgeschrieben werden. Das verlangt etwa 100.000 Refresh-Zyklen pro Sekunde, die zwischen den normalen Speicherzugriffen einzuschieben sind.

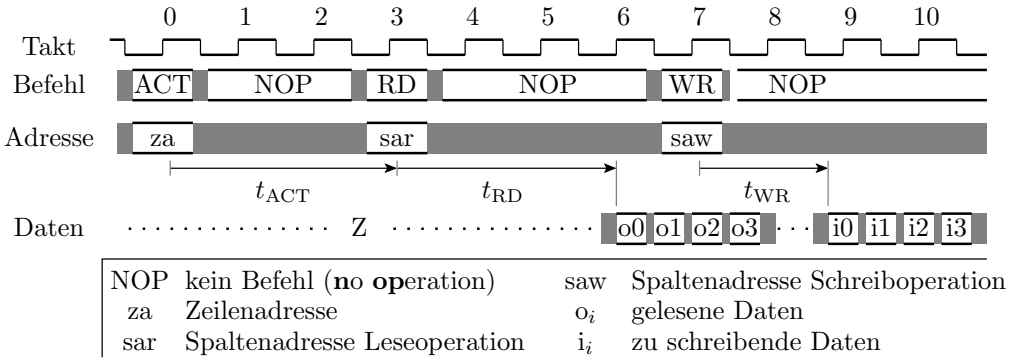
Ein DRAM-Schaltkreis enthält dafür eine Ablaufsteuerung. Die externe Ansteuerung eines DRAMs unterscheidet sich erheblich von der eines SRAMs. Insbesondere erfolgt der Lese- und Schreibzugriff meist blockorientiert und die übergeordnete Schaltung muss sich auf variable Wartezeiten einstellen.

Ansteuerung



- Übertragung der Zeilen- und Spaltenadressen nacheinander.
- Auffrischen hat Vorrang, kann Zugriff verzögern.
- Datenaustausch in größeren Blöcken sinnvoll.

Ansteuerung eines DDR-DRAMs[8]



Die Aktivierung einer Zeile dauert relativ lange und erfordert die erste Hälfte der Adresse. Nach der Aktivierung ist ein schneller wahlfreier Zugriff mit der zweiten Adresshälfte auf den Zwischenspeicher möglich. Ein DRAM hat deshalb nur halb so viele Adressleitungen wie Adressbits.

Der DRAM-interne Ablauf wird mit Befehlen gesteuert (ACT, RD, WR, NOP (**N**o **Op**eration) und Refresh). Im Beispielsignalverlauf werden nach der Aktivierung einer Zeile bei jeder Lese- und Schreibanforderung mehrere aufeinanderfolgende Bits abwechselnd mit der steigenden und der fallenden Taktflanke übertragen[9].

4.4 Festwertspeicher

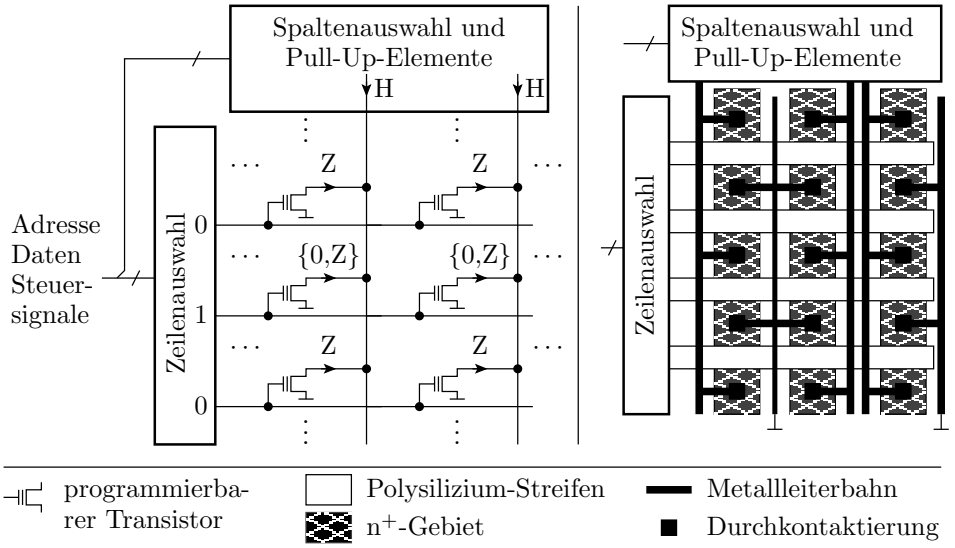
Das Prinzip von Festwertspeichern

[8] DDR-DRAM **D**ouble **D**ate **R**ate **D**ynamic **R**andom **A**ccess **M**emory.
[9] Das verdoppelt die Datenrate und ist gemeint mit »**D**ouble **D**ate **R**ate«.

Die Speicherelemente eines MOS-Festwertspeichers sind einzelne Transistoren, die durch Programmierung entweder ein- und ausschaltbar oder nur einen der beiden Schaltzustände haben.

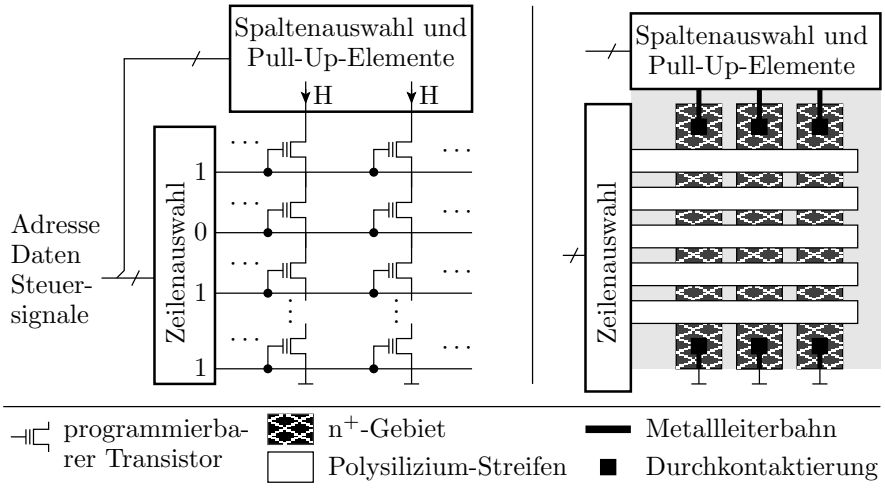
- NOR-ROM: Parallelschaltung programmierter NMOS-Transistoren. Die Zeilenauswahlschaltung steuert pro Spalte nur den ausgewählten Transistor mit »1« und alle anderen mit »0« an. Die nicht ausgewählten Transistoren sind aus. Schaltet der ausgewählte Transistor nicht ein, sperrt die Parallelschaltung, sonst schaltet sie ein.
- NAND-ROM: Reihenschaltung programmierter Transistoren. Die Zeilenauswahlschaltung steuert den ausgewählten Transistor mit »0« und alle anderen mit »1« an. Die nicht ausgewählten Transistoren schalten ein. Wenn der ausgewählte Transistor nicht ausschaltet, schaltet die Reihenschaltung ein, sonst aus.

NOR-ROM



Schneller als der NAND-ROM.

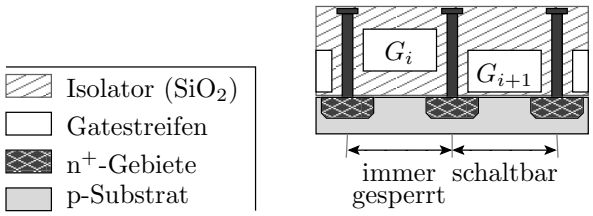
NAND-ROM



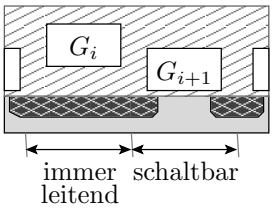
Flächensparender, weniger Stromverbrauch, aber langsamer als NOR-ROM.

Deaktivierung von Transistoren bei der Fertigung

- NOR-ROM: Erhöhung der Einschaltspannung, z.B. durch belassen eines dicken Gate-Oxids:

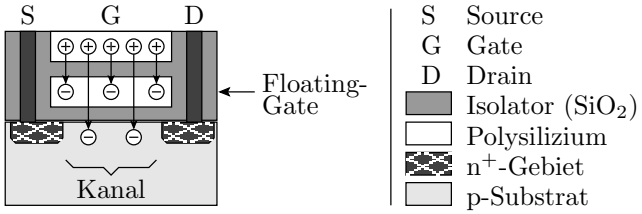


- NAND-ROM: Kurzschluss von Drain und Source:



Elektrisch einstellbare Einschaltspannung

Transistoren mit elektrisch einstellbarer Einschaltspannung haben ein zweites vollständig im Gateoxid eingeschlossenes »Floating Gate«. Positive Ladungen auf dem »Floating Gate« vergrößern und negative verringern die Einschaltspannung. Die Datenhaltezeit für die Ladungen hat eine Größenordnung von 10 Jahren. Danach müssen derartige Speicher (Programmspeicher von Mikrorechnern, Memory-Sticks, etc.) aufgefrischt werden.



Elektrische Programmierung

Für das Auf- und Entladen der »Floating Gates« werden folgende physikalische Mechanismen genutzt:

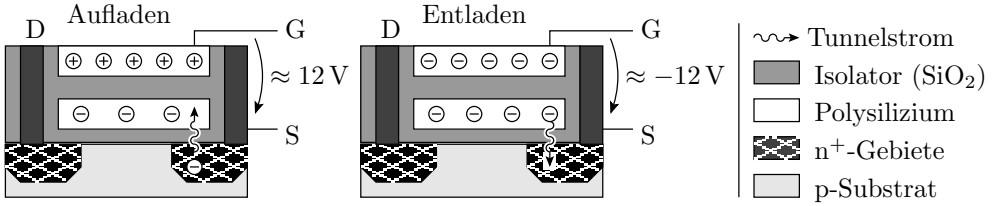
- Heiße Elektronen: Elektronen mit sehr hoher Geschwindigkeit werden durch Gitterstreuungen zum Teil in das Gate abgelenkt und können so das »Floating Gate« negativ aufladen.

- Hochenergetische Strahlung (UV-Licht) erzeugt eine geringe Leitfähigkeit des Gateoxids, so dass sich die »Floating Gates« entladen können.

- Tunnelströme: Das ist ein quantenmechanisches Phänomen, bei dem Ladungsträger eine dünne Isolationsschicht, überwinden, indem sie plötzlich auf der anderen Seite sind. Voraussetzung sind sehr hohe Feldstärken. Geeignet zum Auf- und Entladen.

Heutige EE- (**E**lectrically **E**rasable) PROMs nutzen letzteres.

Programmieren mit Tunnelströmen



Bei elektrisch programmier- und löschbarer Einschaltspannung haben die »Floating Gates« dünne Tunnelfenster (Dicke $\approx 10\,\text{nm}$) zum Source oder Drain. Tunnelströme nehmen exponentiell mit der Spannung zu. Die Programmierung erfolgt mit einer stark überhöhten Spannung. Das Aufladen dauert in der Größenordnung von Millisekunden. Bei normaler Betriebsspannung ist die Datenhaltezeit etwa 10 Jahre.

Die interne Steuerung eines EEPROMs ist noch komplizierter als bei einem DRAM. Interner Spannungsvervielfacher zur Erzeugung der Programmierspannung. Kontrollfunktionen für die eingestellten Einschaltspannungen. Ablaufsteuerung, die den Programmiervorgang fortsetzt, bis die Einschaltspannungen der gerade programmierten Zellen im Soll-Bereich liegen. Rekonfigurationsmöglichkeiten zum Ersatz von Speicherblöcken mit defekten Zellen. Puffer für Schreibdaten und Funktionen für das zeitgleiche Beschreiben einer Pufferseite. ...

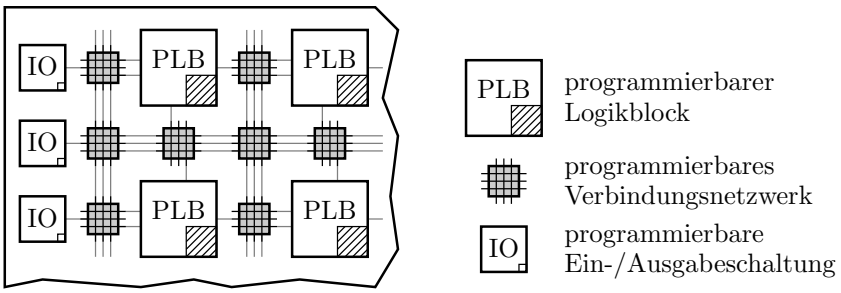
Ein Flash-Speicher ist ein EEPROM, in dem die Speicherzellen nicht einzeln, sondern nur in Blöcken gelöscht werden können. Das vereinfacht die interne Programmierung und Schaltung, spart Chipfläche und ist für viele Anwendungen ausreichend.

5 Programmierbare Logikschaltkreise

Programmierbare Logikschaltkreise

Programmierbare Logikschaltkreise bestehen aus

- programmierbaren Logikblöcken,
- programmierbaren Verbindungsnetzwerken,
- programmierbaren Ein-/Ausgabeschaltungen, ...

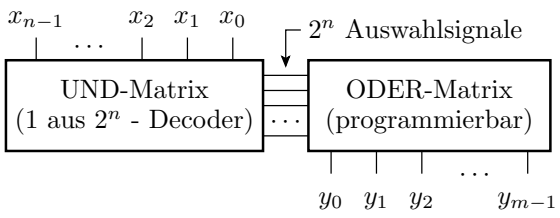


Sie ersetzen Schaltungen, die früher aus vielen Schaltkreisen bestanden und sind hervorragend für Prototypen, Kleinserien und studentische Praktika geeignet.

Programmierbare Logikblöcke

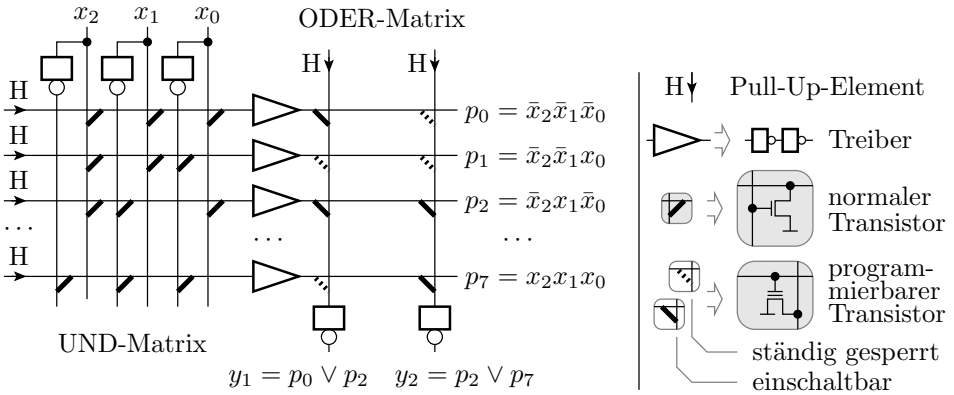
Die Grundstruktur einer universell programmierbaren Logikfunktion ist ein programmierbarer Speicherblock aus einer UND-Matrix für die Zeilenauswahl und einer ODER-Matrix zur Verknüpfung von Zeilenauswahlsignalen.

Mit einer UND-Matrix, die für jede der 2^n (n – Anzahl der Eingänge) genau eine Zeile auswählt, lässt sich jede beliebige Wertetabelle mit n-Eingabebits einprogrammieren.



Alternative ist eine programmierbare UND-Matrix.

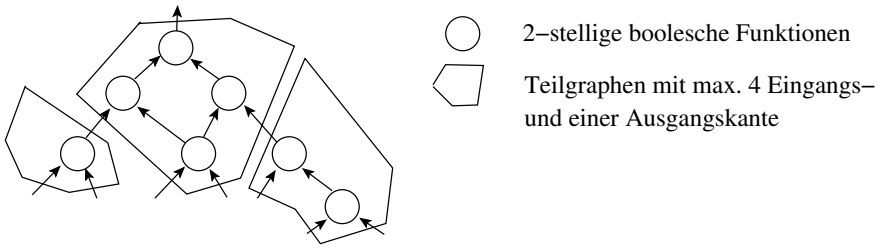
Programmieren einer Beispielfunktion



Funktionen mit mehr als n Eingabebits werden aus mehreren Tabellenfunktionen zusammengesetzt und auf mehrere programmierbare Logikblöcke verteilt.

Abdeckung mit Tabellenfunktionen

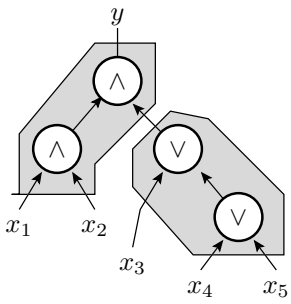
Gegeben sein eine $m > n$-stellige Funktion. Diese soll aus n-stelligen Funktionen ($n \in 3, 4, 5$) zusammengesetzt werden. Aufstellung des Berechnungsflusses, z.B. als Ausdruck oder Entscheidungsbaum und Abdeckung von Teilgraphen mit nicht mehr als n Eingängen:



Die Zusammensetzung aus mehreren n-stelligen Funktionen verlangt zusätzlich ein programmierbares Verbindungsnetzwerk.

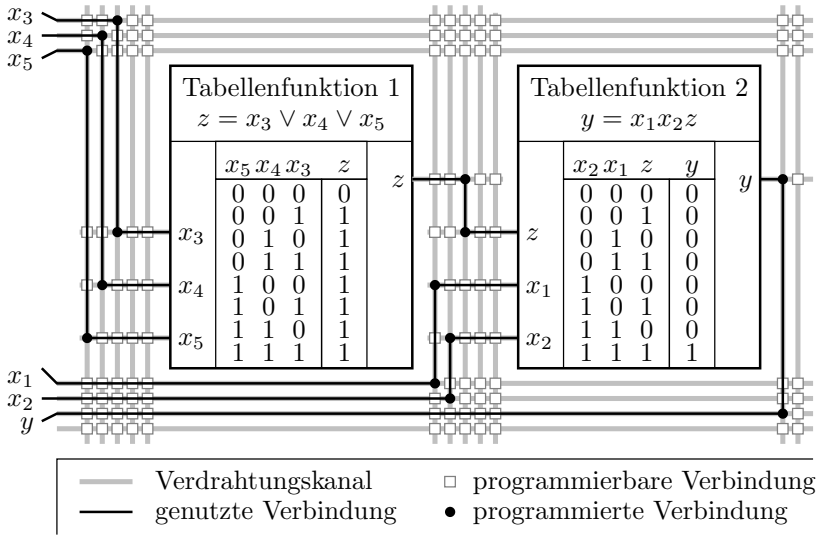
Beispiel sei die Abdeckung einer 5-stelligen mit zwei 3-stelligen Logikfunktionen:

$$y = x_1x_2\,(x_3 \vee x_4 \vee x_5)$$



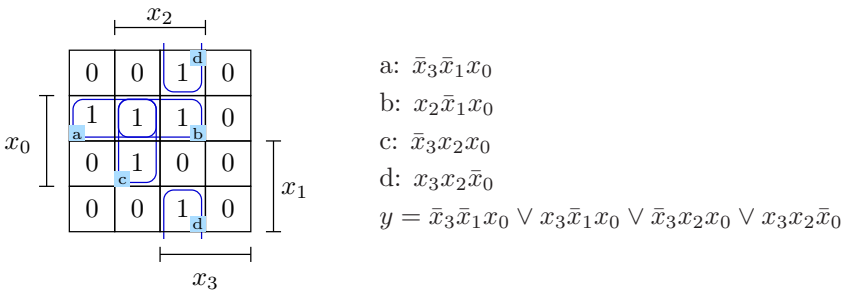
Tabellenfunktion 1 : $z = x_3 \vee x_4 \vee x_5$
Tabellenfunktion 2 : $y = x_1x_2z$

Tabellenfunktion 1 : $z = x_3 \vee x_4 \vee x_5$
Tabellenfunktion 2 : $y = x_1x_2z$

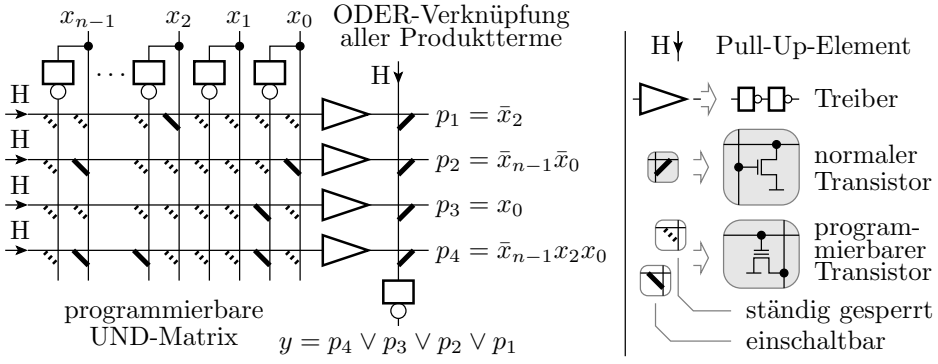


Programmierbare UND-Matrix

Bei der Schaltungsminimierung mit KV-Diagrammen oder nach Quine und McCluskey ist das Ergebnis eine ODER-Verknüpfung minimierter Konjunktionen.



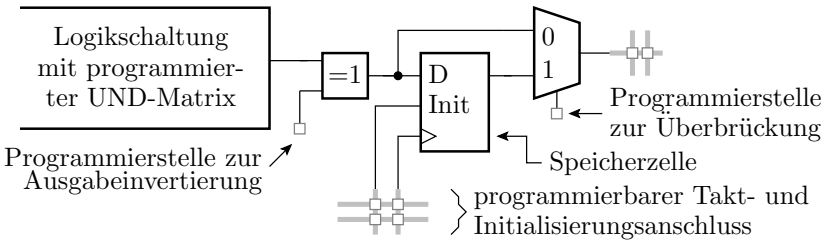
Für die Programmierung so minimierter Ausdrücke ist eine Schaltungsstruktur, bei der programmierbare UND-Terme ODER-verknüpft werden, günstiger.



Die UND-Matrix, ein Draht-UND hat programmierbare Transistoren für die direkten und negierten Werte aller Eingänge. Alle UND-Terme sind ODER verknüpft. Diese Struktur erlaubt bei gleicher Anzahl von Programmierstellen Funktionen mit mehr Eingängen, typ. 8 bis 16 Eingänge bei 4 bis 16 Produkttermen. Tabellenfunktionen in programmierbaren Schaltkreisen haben nicht mehr als 3 bis 5 Eingänge.

Weitere Programmierelemente

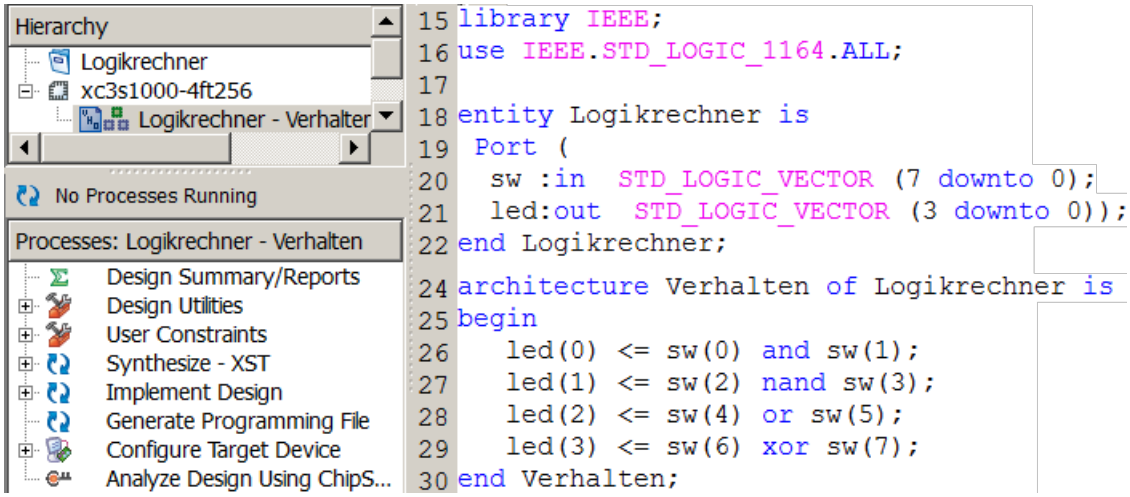
Logikschaltkreise mit programmierbarer UND-Matrix haben in der Regel zusätzlich eine programmierbare Ausgabeinvertierung, damit die Funktion auch nach den Nullen entwickelt werden kann:



Jede programmierbare Logikfunktion hat ein überbrückbares Abtastregister. Größere programmierbare Logikschaltkreise haben zusätzlich konfigurierbare Rechenwerke, Taktverteiler, Blockspeicher, Prozessorkerne, ...

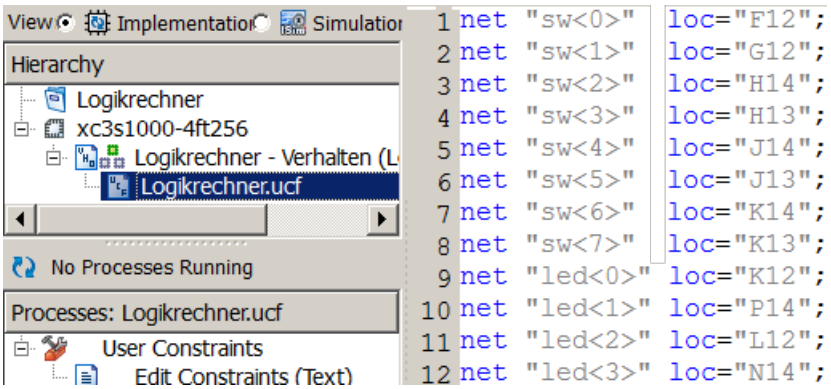
6 Schaltungsentwurf mit FPGAs

Rechnereingabe in VHDL



- Projekt anlegen, einige Konfigurationen vornehmen, ...
- Beschreibung eingeben, Syntaxtest, optional Simulation, ...

Das Constraint-File

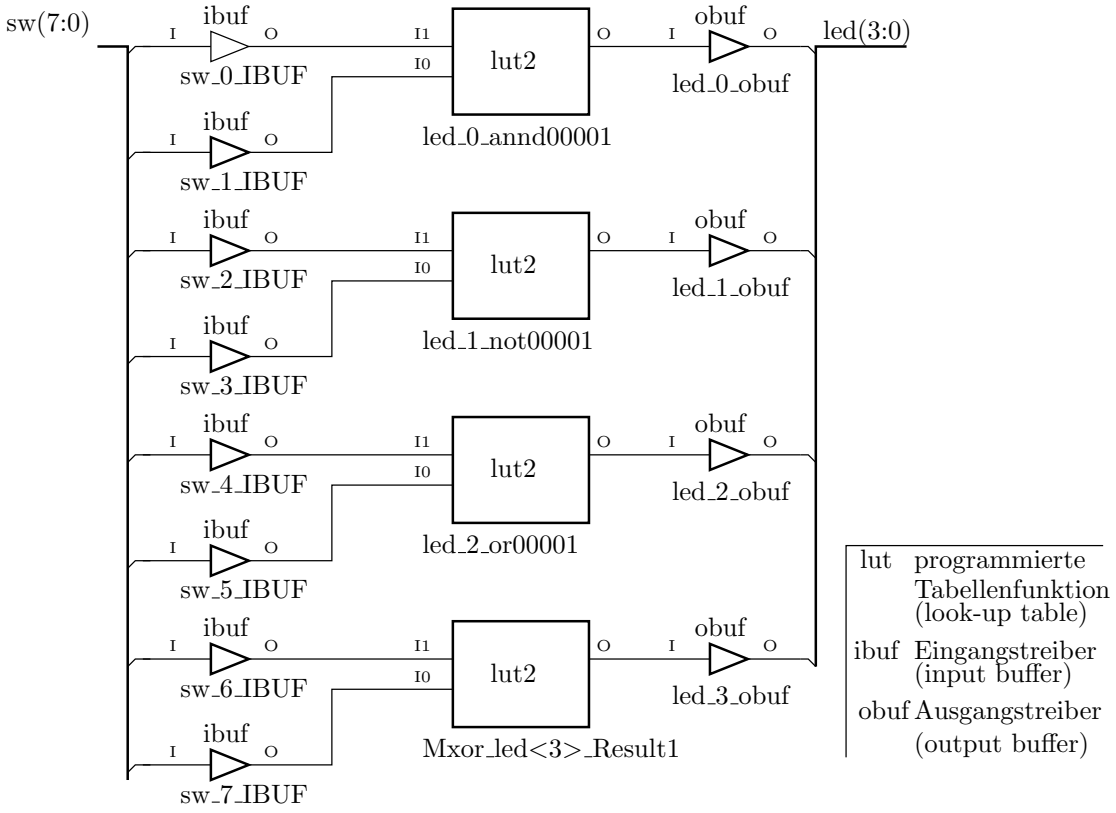


Die Constraint-Datei enthält alle zusätzlichen Informationen zur Vorgabe der Zielfunktion, die nicht in der VHDL-Datei stehen: die Pin-Namen der Schaltungsanschlüssen (s.o.), Taktfrequenz, ...

Synthese

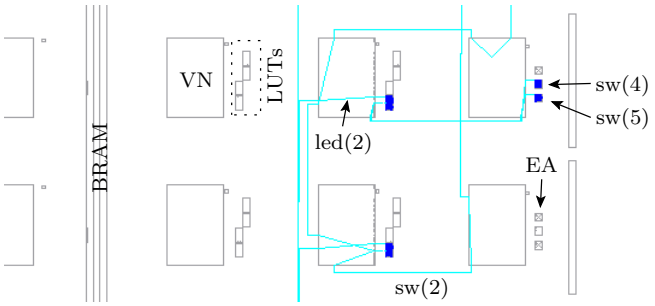
Berechnung einer Schaltung aus der VHDL-Beschreibung. Im Beispiel ist das trivial, weil die Schaltung nur aus vier Gattern besteht. Für größere Entwürfe wird die Zielfunktion mit Bitvektoren, arithmetischen Operatoren, Fallunterscheidungen, Unterprogrammen, ... beschrieben. Die Synthese muss daraus die logischen Funktionen extrahieren, optimieren, mit Teilschaltungen nachbilden, ...

Unser programmierbarer Schaltkreis hat statt Gatter als logische Grundbausteine Tabellenfunktionen (LUT Look-Up Table, kleine programmierbare Speicher). An den Anschlüssen werden Buffer eingefügt, die die internen kleineren Spannungspegel (0/1V) auf die größeren Anschlusspegel (0/2,5...3,3V) umsetzen.

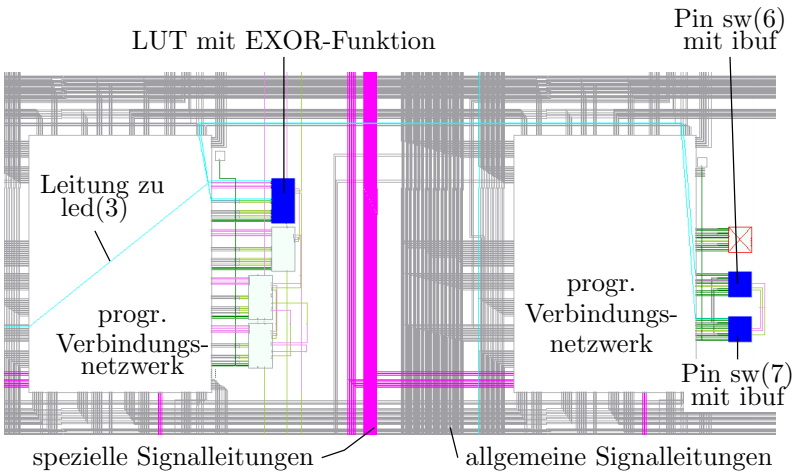


Verdrahtung

Nach der Synthese folgt die Platzierung der einzelnen Funktionsblöcke und ihre Verdrahtung. Ein FPGA besteht aus konfigurierbaren Funktionsblöcken (LUTs), EA-Schaltungen, Multiplizierern, Blockspeichern (BRAM),... und programmierbare Verbindungsnetzwerken (VN).



Ein vergrößerter Ausschnitt mit dem Logikblock, in den das EXOR-Gatter programmiert ist, und den beiden Eingangstreibern vor dem EXOR. Handverdrahtung ist möglich, aber nicht zu empfehlen.


LUT mit EXOR-Funktion
Pin sw(6)
mit ibuf
Leitung zu
led(3)
progr.
Verbindungs-
netzwerk
progr.
Verbindungs-
netzwerk
Pin sw(7)
mit ibuf
spezielle Signalleitungen
allgemeine Signalleitungen