

Kapitel 4 – Sequentielle Logik

1. Speichernde Elemente
2. Sequentielle Schaltkreise
3. Entwurf sequentieller Schaltkreise
4. **SRAM**
5. Anwendung: Datenpfade von ReTI

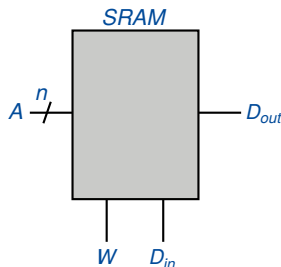
Albert-Ludwigs-Universität Freiburg

Prof. Dr. Christoph Scholl

Institut für Informatik
WS 2015/16

SRAM: Ein-/Ausgänge und Zeichen

- SRAM = Static Random-Access Memory.
- Sei $n \in \mathbb{N}$, $N = 2^n$. Ein N -Bit statischer Speicher oder SRAM hat:
 - n Eingänge $A = (A_{n-1}, \dots, A_0)$ „Adresse“,
 - Dateneingang D_{in} ,
 - Datenausgang D_{out} ,
 - Kontrollsignal W „write“

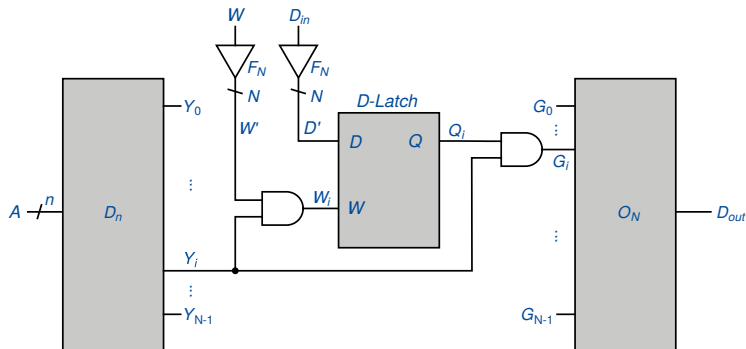


- Der Speicher enthält N Speicherzellen L_0, \dots, L_{N-1} , die je ein Bit speichern können.
- Zelle $L_{\langle A \rangle}$ wird mit Hilfe der Adresse A ausgewählt.
 - An D_{out} erscheint der Inhalt von $L_{\langle A \rangle}$.
 - Durch Schreibpuls an W wird D_{in} nach $L_{\langle A \rangle}$ übernommen.

N -Bit-SRAM, $N \times s$ -Bit-SRAM: Aufbau

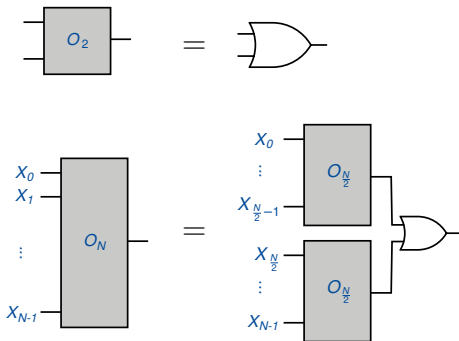
- Ein $N \times s$ -Bit-SRAM besteht aus s N -Bit SRAMs mit gemeinsamen Adress- und Schreibsignalen.
 - s heißt **Bitbreite** des $N \times s$ -Bit-SRAMs.
- Ein N -Bit-SRAM besteht im Prinzip aus 3 **Hilfsschaltkreisen**:
 - mehrfaches ODER
 - Treiberbäume
 - Dekodierer
- Motivation von Treiberbäumen durch **Fanout-Beschränkung**: Aus elektrischen Gründen kann eine Leitung nicht auf beliebig viele Gattereingänge verzweigen

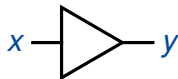
SRAM: Schaltbild



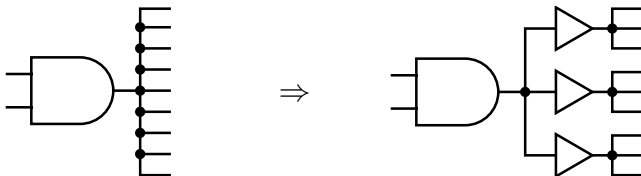
O_N : Mehrfaches ODER

- Ein **N-faches ODER** O_N mit $N = 2^n$ ist ein Schaltkreis, der **N-faches ODER** berechnet.
- **Balancierter Baum**, um Verzögerungszeit zu minimieren (Tiefe $O(\log N) = O(n)$).





- Ein **Treiber** ist ein Gatter mit einem Eingang X und einem Ausgang Y , das die **Identität** $Y = X$ berechnet.
 - Eingesetzt, um **Fanout-Beschränkung** zu überwinden.
 - Beispiel: Fanout-Beschränkung von 3.



F_N : Treiberbäume im SRAM

- Zur Erinnerung: Ein **Baum** ist ein azyklischer gerichteter Graph $G = (V, E)$ mit:
 - Genau einer Quelle w ,
 - $\text{indeg}(v) = 1$ für alle $v \in V \setminus \{w\}$
 - Blätter = Knoten $v \in V$ mit $\text{outdeg}(v) = 0$
 - Innere Knoten = Knoten $v \in V$ mit $\text{outdeg}(v) \geq 1$.
- Im SRAM für Realisierung von F_N eingesetzt.
- Wir betrachten hier eine Fanout-Beschränkung von $10 \rightarrow 10$ -äre Bäume.

10-äre Bäume (1/2)

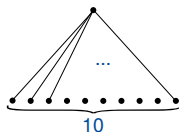
B_0 :



- Anzahl der Blätter von B_s : $L(B_s)$.

- Entspricht dem erreichten Verzweigungsgrad.

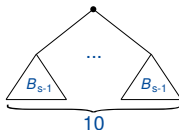
B_1 :



- Anzahl der inneren Knoten von B_s : $I(B_s)$.

- Entspricht der benötigten Anzahl von Treibern.

B_s :



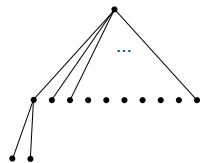
$$L(B_s) = 10^s$$

$$I(B_s) = \sum_{i=0}^{s-1} 10^i = \frac{10^s - 1}{10 - 1} < \frac{L(B_s)}{9}$$

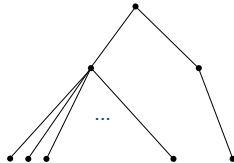
- Benutze also B_s zum 10^s -fachen Vervielfältigen eines Signals
- Innere Knoten des Baumes werden durch Treiber ersetzt
- \Rightarrow **Treiberbaum** mit Fanoutbeschränkung 10

Treiberbäume: Allgemeiner Fall (1/2)

- Angenommen, ein Signal soll x -fach vervielfältigt werden mit $10^{s-1} < x < 10^s$ keine Zehnerpotenz.
- Ziel: **Balancierte Treiberbäume**, d.h. alle Pfade von der Wurzel zu einem Blatt haben gleiche Länge.



$x = 11$, unbalanciert



$x = 11$, balanciert

Treiberbäume: Allgemeiner Fall (2/2)

- **Idee:** “Fülle Bäume von links her 10-är auf” und Sorge zusätzlich für gleiche Tiefe der Blätter!
- Beispiel: ...

Lemma

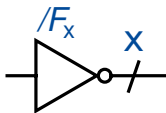
$\forall s \in \mathbb{N}$ und $x \in \{10^{s-1} + 1, \dots, 10^s\}$ gibt es einen Baum $T(x)$ mit Ausgangsgrad ≤ 10 an jedem inneren Knoten und den folgenden Eigenschaften:

- 1 $T(x)$ hat x Blätter.
- 2 $T(x)$ hat $\leq \frac{x}{9} + s$ innere Knoten.
- 3 Alle Pfade von der Wurzel zu einem Blatt haben genau die Länge $s = \lceil \log_{10} x \rceil$ mit $\lceil \log_{10} x \rceil < \frac{1}{3} \log_2 x + 1$.

- Beweis: Induktion über $s \Rightarrow$ Übung

Notation: Invertierender Treiberbaum

- Invertierender Treiberbaum $/F_x$:
Ersetze den Treiber an der Wurzel durch einen Inverter.



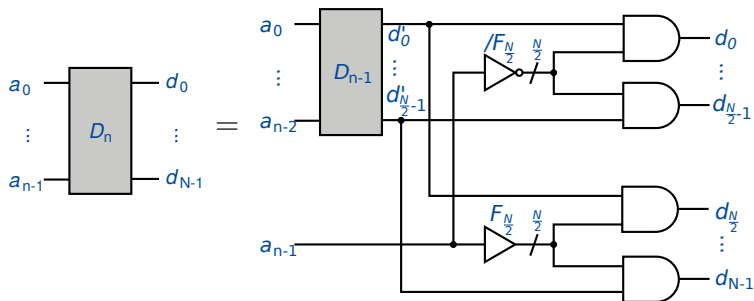
- Sei $n \in \mathbb{N}$, $N = 2^n$. Ein n -Bit-Dekodierer D_n ist ein Schaltkreis, der die Funktion $d : \mathbb{B}^n \rightarrow \mathbb{B}^N$ berechnet, wobei gilt:

$$d_i(a) = \begin{cases} 1, & \text{falls } \langle a \rangle = i \\ 0, & \text{sonst} \end{cases} \quad \forall i = 0, \dots, N-1$$

($d_i(a)$ ist Bit i des N -Tupels $d(a)$.)

- Induktive Konstruktion von D_n : Siehe nächste Folie.

Dekodierer: Rekursiver Aufbau



$$\text{depth}(D_1) = 1 \quad (1)$$

$$\text{depth}(D_2) = 2 \quad (2)$$

$$\text{depth}(D_n) = \max(\text{depth}(D_{n-1}), \text{depth}(F_{\frac{N}{2}})) + 1 \quad (3)$$

$$\leq \max(\text{depth}(D_{n-1}), \frac{1}{3} \log_2\left(\frac{N}{2}\right) + 1) + 1 \quad (4)$$

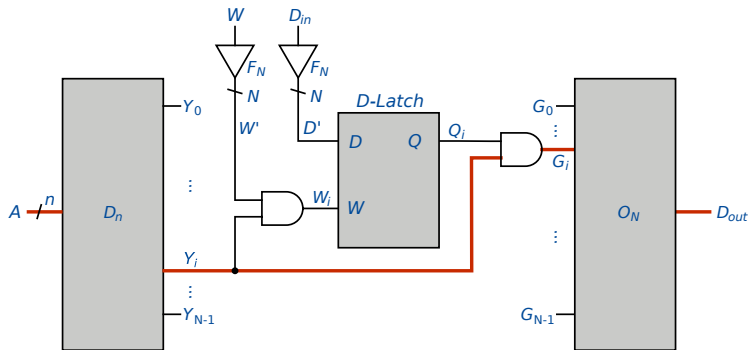
$$= \max(\text{depth}(D_{n-1}), \frac{1}{3}n + \frac{2}{3}) + 1 \quad (5)$$

$$(6)$$

■ Durch Induktion folgt: $\text{depth}(D_n) = n$.

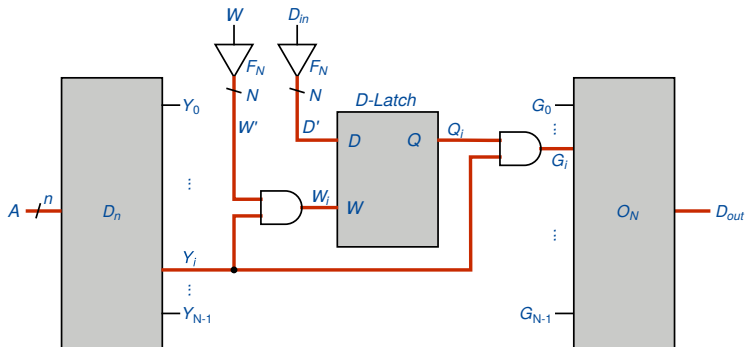
SRAM: Lesevorgang ($W = 0$)

- D_n setzt $Y_i = 1$ für $i = \langle A \rangle$, $Y_j = 0$ für $j \neq i$.
- Der Inhalt der i -ten Zelle L_i steht an G_i , für alle $j \neq i$ steht an G_j der Wert 0.



SRAM: Schreibvorgang (Puls auf W)

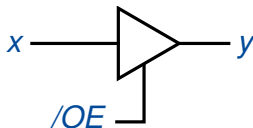
- D_{in} an D -Eingänge sämtlicher Latches angelegt.
- Schreibpuls nur am W -Eingang von L_j (da $Y_j = 1$).



Tristate-Treiber und Busse

- **Tristate-Treiber** sind Treiber mit Eingangssignal x und zusätzlichem Signal $/OE$, dem **Output-Enable-Signal**.
- Am Ausgang y erscheint

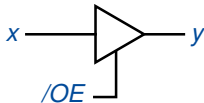
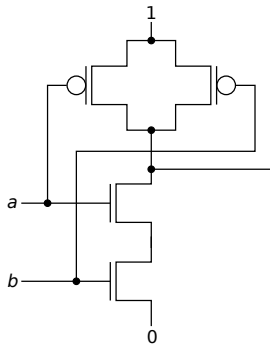
$$y = \begin{cases} x, & \text{falls } /OE = 0 \\ Z, & \text{sonst } /OE = 1 \end{cases}$$



- Z bezeichnet den **Zustand hoher Impedanz** (high-Z).

Zustand hoher Impedanz

- Wir haben bisher Schaltungen betrachtet, die aus **CMOS-Gattern** bestehen. Dort ist jede Leitung zu jedem Zeitpunkt entweder mit V_{DD} (logisch-1) oder Masse (logisch-0) verbunden.
- Eine Leitung im Zustand **Z**, also der Ausgang eines Treibers mit $\text{/OE} = 1$, ist **weder mit V_{DD} noch mit Masse** verbunden. Man sagt, der Treiber ist disabled ($\text{/OE} = 0$: enabled).



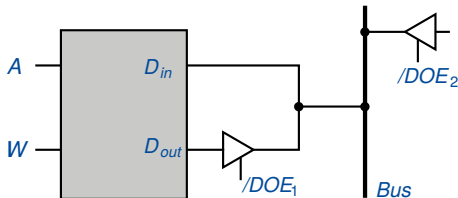
- **n -Bit-Treiber**: n Treiber mit gemeinsamen $/OE$.
- Im Gegensatz zu Ausgängen üblicher Gatter kann man **Ausgänge** von Tristate-Treibern **zusammenschalten**. Man muss dafür sorgen, dass zu jeder Zeit **höchstens ein Treiber enabled** ist.
- Ein **n Bit breiter Bus** ist ein Bündel aus n Leitungen, welches die Ausgänge von **mehreren n -Bit-Treibern** verbindet.



- k Tristate-Treiber, die durch einen Bus verbunden sind, wirken ähnlich wie ein k -fach-Multiplexer.
- Vorteile Bus gegenüber Multiplexer:
 - Leicht erweiterbar.
 - Datentransport in verschiedene Richtungen zu verschiedenen Zeiten.
- Nachteil von Bus:
 - Man muss **Bus Contention** vermeiden, d.h. es darf nie mehr als ein Treiber auf einem Bus gleichzeitig enabled sein (sonst Folgen bis hin zur physikalischen Zerstörung der Schaltung)!

Bus zur Kommunikation mit SRAM

- SRAM mit gemeinsamem Datenein- und -ausgang.



- **Lesezugriff** auf den Speicher: $/DOE_1$ enabled, alle anderen Treiber, z.B. $/DOE_2$, disabled.
- **Schreibzugriff**: D_{in} nimmt den Wert vom Bus, $/DOE_1$ disabled.