

Übungen zur Grundlagen der Technischen Informatik

Übung 9 – CMOS, PAL, NAND, Latches und Flipflops

Florian Frank

Friedrich-Alexander-Universität Erlangen-Nürnberg

Wintersemester 2018/19



Was machen wir heute?

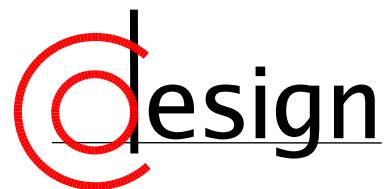
Aufgabe 1 – CMOS-Gatterschaltungen

Aufgabe 2 – NAND-Technik

Aufgabe 3 – PAL-Implementierung

Aufgabe 4 – Latches und Flipflops

Aufgabe 1 – CMOS-Gatterschaltungen



Aufgabe 1 – CMOS-Gatterschaltungen

Sei die Schaltfunktion $f_1(x_3, x_2, x_2, x_0) = x_0 \overline{x_1 x_2} + x_0 \overline{x_1 x_3}$ gegeben.

- a) Standardzellen sind vorgefertigte CMOS-Realisierungen einfacher Schaltfunktionen, wie zum Beispiel Und, Oder oder Nicht, die im Baukastenprinzip zusammengesetzt werden können. Schalten Sie die folgenden Standardzellen so zusammen, dass sie f_1 realisieren (Hinweis: die einzelnen Standardzellen sind gestrichelt umrahmt).

Aufgabe 1 – CMOS: Begriffsklärung

Transistor (TRANSFER RESISTOR)

Ein Transistor ist ein steuerbarer Widerstand. Wir nutzen ihn als einen durch Spannung steuerbaren Schalter .

Aufgabe 1 – CMOS: Begriffsklärung

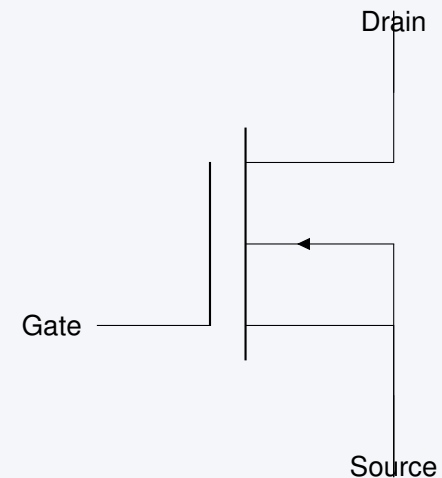
Transistor (TRANSFER RESISTOR)

Ein Transistor ist ein steuerbarer Widerstand. Wir nutzen ihn als einen durch Spannung steuerbaren Schalter .

MOSFETs (*metal-oxid-semiconductor-field-effect-transistor*)

Ein MOSFET ist ein Feldeffekttransistor mit isoliertem Gate mit – historisch – einer Metall-Isolator-Halbleiter-Struktur.

Wir unterscheiden zwischen NMOS und PMOS-Transistoren:



Aufgabe 1 – CMOS: Begriffsklärung

Transistor (TRANSFER RESISTOR)

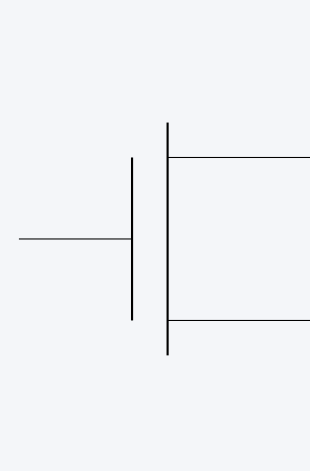
Ein Transistor ist ein steuerbarer Widerstand. Wir nutzen ihn als einen durch Spannung steuerbaren Schalter .

MOSFETs (*metal-oxid-semiconductor-field-effect-transistor*)

Ein MOSFET ist ein Feldeffekttransistor mit isoliertem Gate mit – historisch – einer Metall-Isolator-Halbleiter-Struktur.

Wir unterscheiden zwischen NMOS und PMOS-Transistoren:

NMOS n-dotiert → öffnet bei logischer 1
Kommt im „pull-down“-Netzwerk vor.



Aufgabe 1 – CMOS: Begriffsklärung

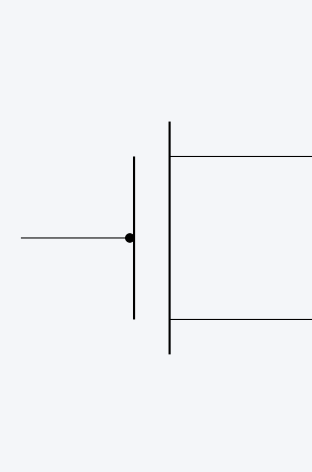
Transistor (TRANSFER RESISTOR)

Ein Transistor ist ein steuerbarer Widerstand. Wir nutzen ihn als einen durch Spannung steuerbaren Schalter .

MOSFETs (*metal-oxid-semiconductor-field-effect-transistor*)

Ein MOSFET ist ein Feldeffekttransistor mit isoliertem Gate mit – historisch – einer Metall-Isolator-Halbleiter-Struktur.

Wir unterscheiden zwischen NMOS und PMOS-Transistoren:



PMOS p-dotiert → schließt bei logischer 1
Kommt im „pull-up“-Netzwerk vor.

Aufgabe 1 – CMOS: Begriffsklärung

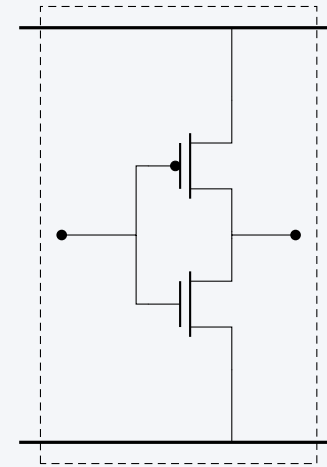
MOSFETs (*metal-oxid-semiconductor-field-effect-transistor*)

Ein MOSFET ist ein Feldeffekttransistor mit isoliertem Gate mit – historisch – einer Metall-Isolator-Halbleiter-Struktur.

Wir unterscheiden zwischen NMOS und PMOS-Transistoren:

NMOS n-dotiert → öffnet bei logischer 1
Kommt im „pull-down“-Netz vor.

PMOS p-dotiert → schließt bei logischer 1
Kommt im „pull-up“-Netz vor.



CMOS (COMPLEMENTARY MOS(FET)s)

Man versteht unter der CMOS-Technologie eine Logikfamilie, sowie den dazu verwendeten Halbleiterprozess.

Der Grundgedanke dieser ist die Kombination von PMOS und NMOS-Transistoren. Die gewünschte Operation wird dabei sowohl im PMOS-Netz (das „pull-up“-Netz) als auch im NMOS-Netz (das „pull-down“-Netz) realisiert.

Aufgabe 1 – CMOS: Ein Beispiel (I)

Realisieren Sie die Schaltfunktion $f_B(x, y) = \overline{x + y}$.
Wir gehen dazu wie folgt vor:

Aufgabe 1 – CMOS: Ein Beispiel (I)

Realisieren Sie die Schaltfunktion $f_B(x, y) = \overline{x + y}$.

Wir gehen dazu wie folgt vor:

Pull-Up-Netzwerk Hier müssen alle Literale negiert auftreten („ein PMOS ist ein negierter NMOS“). Wir formen also die Funktion wie folgt um:

Aufgabe 1 – CMOS: Ein Beispiel (I)

Realisieren Sie die Schaltfunktion $f_B(x, y) = \overline{x + y}$.

Wir gehen dazu wie folgt vor:

Pull-Up-Netzwerk Hier müssen alle Literale negiert auftreten („ein PMOS ist ein negierter NMOS“). Wir formen also die Funktion wie folgt um:

$$f_{B_{PMOS}} = \overline{x + y} = \overline{x} \cdot \overline{y}$$

Aufgabe 1 – CMOS: Ein Beispiel (I)

Realisieren Sie die Schaltfunktion $f_B(x, y) = \overline{x + y}$.

Wir gehen dazu wie folgt vor:

Pull-Up-Netzwerk Hier müssen alle Literale negiert auftreten („ein PMOS ist ein negierter NMOS“). Wir formen also die Funktion wie folgt um:

$$f_{B_{PMOS}} = \overline{x + y} = \overline{x} \cdot \overline{y}$$

Wir sehen, dass alle Literale negiert vorkommen. Wir brauchen also zwei PMOS-Transistoren, die

Aufgabe 1 – CMOS: Ein Beispiel (I)

Realisieren Sie die Schaltfunktion $f_B(x, y) = \overline{x + y}$.

Wir gehen dazu wie folgt vor:

Pull-Up-Netzwerk Hier müssen alle Literale negiert auftreten („ein PMOS ist ein negierter NMOS“). Wir formen also die Funktion wie folgt um:

$$f_{B_{PMOS}} = \overline{x + y} = \overline{x} \cdot \overline{y}$$

Wir sehen, dass alle Literale negiert vorkommen. Wir brauchen also zwei PMOS-Transistoren, die in Serie geschaltet sind.

Aufgabe 1 – CMOS: Ein Beispiel (I)

Realisieren Sie die Schaltfunktion $f_B(x, y) = \overline{x + y}$.

Wir gehen dazu wie folgt vor:

Pull-Up-Netzwerk Hier müssen alle Literale negiert auftreten („ein PMOS ist ein negierter NMOS“). Wir formen also die Funktion wie folgt um:

$$f_{B_{PMOS}} = \overline{x + y} = \overline{x} \cdot \overline{y}$$

Wir sehen, dass alle Literale negiert vorkommen. Wir brauchen also zwei PMOS-Transistoren, die in Serie geschaltet sind. Die Eingänge sind dann aber x und y .

Aufgabe 1 – CMOS: Ein Beispiel (I)

Realisieren Sie die Schaltfunktion $f_B(x, y) = \overline{x + y}$.

Wir gehen dazu wie folgt vor:

Pull-Up-Netzwerk Hier müssen alle Literale negiert auftreten („ein PMOS ist ein negierter NMOS“). Wir formen also die Funktion wie folgt um:

$$f_{B_{PMOS}} = \overline{x + y} = \overline{x} \cdot \overline{y}$$

Wir sehen, dass alle Literale negiert vorkommen. Wir brauchen also zwei PMOS-Transistoren, die in Serie geschaltet sind.

Die Eingänge sind dann aber x und y .

Pull-Down-Netzwerk Hier muss die Funktion negiert werden („das PDN ist zum PUN komplementär“):

Aufgabe 1 – CMOS: Ein Beispiel (I)

Realisieren Sie die Schaltfunktion $f_B(x, y) = \overline{x + y}$.

Wir gehen dazu wie folgt vor:

Pull-Up-Netzwerk Hier müssen alle Literale negiert auftreten („ein PMOS ist ein negierter NMOS“). Wir formen also die Funktion wie folgt um:

$$f_{B_{PMOS}} = \overline{x + y} = \overline{x} \cdot \overline{y}$$

Wir sehen, dass alle Literale negiert vorkommen. Wir brauchen also zwei PMOS-Transistoren, die in Serie geschaltet sind.

Die Eingänge sind dann aber x und y .

Pull-Down-Netzwerk Hier muss die Funktion negiert werden („das PDN ist zum PUN komplementär“):

$$f_{B_{NMOS}}(x, y) = \overline{\overline{x + y}} = x + y$$

Aufgabe 1 – CMOS: Ein Beispiel (I)

Realisieren Sie die Schaltfunktion $f_B(x, y) = \overline{x + y}$.

Wir gehen dazu wie folgt vor:

Pull-Up-Netzwerk Hier müssen alle Literale negiert auftreten („ein PMOS ist ein negierter NMOS“). Wir formen also die Funktion wie folgt um:

$$f_{B_{PMOS}} = \overline{x + y} = \overline{x} \cdot \overline{y}$$

Wir sehen, dass alle Literale negiert vorkommen. Wir brauchen also zwei PMOS-Transistoren, die in Serie geschaltet sind.

Die Eingänge sind dann aber x und y .

Pull-Down-Netzwerk Hier muss die Funktion negiert werden („das PDN ist zum PUN komplementär“):

$$f_{B_{NMOS}}(x, y) = \overline{\overline{x + y}} = x + y$$

Mit diesen Funktionen lässt sich nun das CMOS-Netz bilden.

Aufgabe 1 – CMOS: Ein Beispiel (II)

Realisieren Sie die Schaltfunktion $f_B(x, y) = \overline{x + y}$.

Bauen des Pull-Up-Netzwerkes: $f_{PMOS} = \overline{x} \cdot \overline{y}$

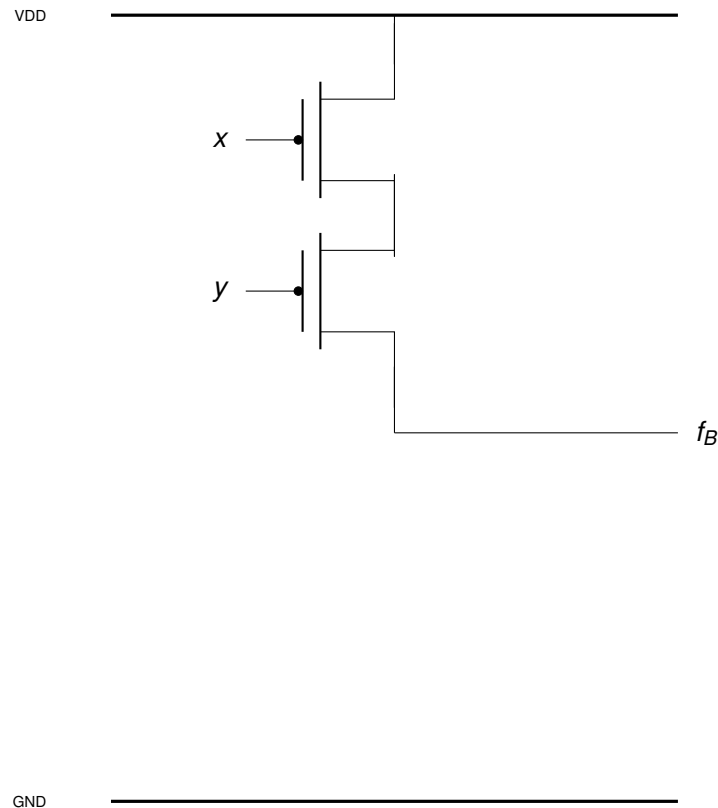
VDD _____

GND _____

Aufgabe 1 – CMOS: Ein Beispiel (II)

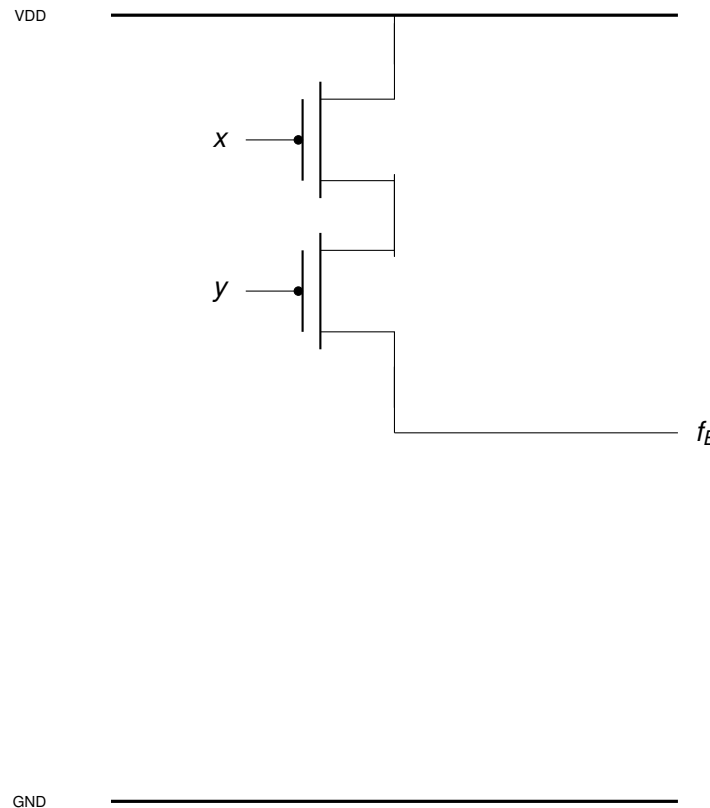
Realisieren Sie die Schaltfunktion $f_B(x, y) = \overline{x + y}$.

Bauen des Pull-Up-Netzwerkes: $f_{PMOS} = \overline{x} \cdot \overline{y}$



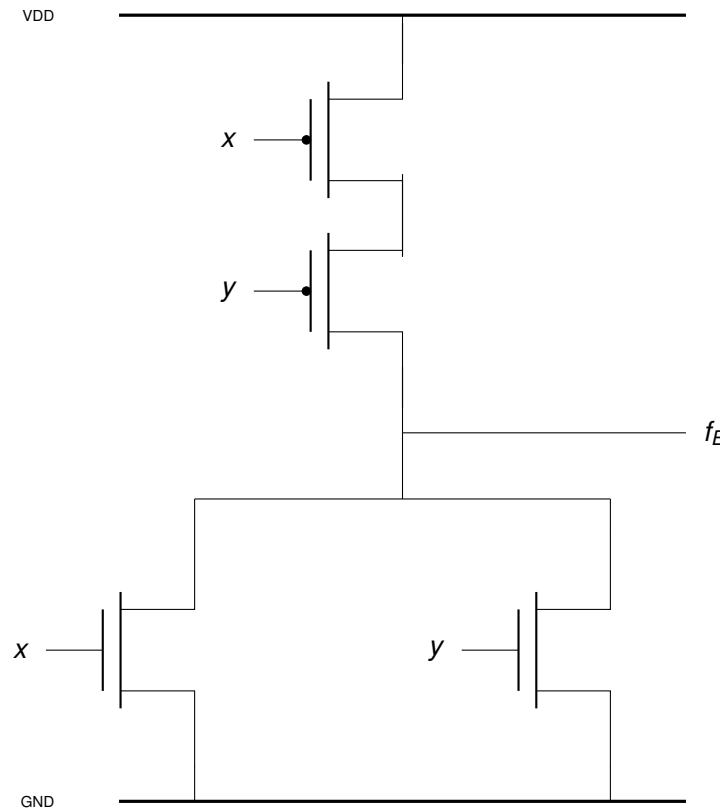
Aufgabe 1 – CMOS: Ein Beispiel (II)

Realisieren Sie die Schaltfunktion $f_B(x, y) = \overline{x + y}$.
 Bauen des Pull-Down-Netzwerkes: $f_{NMOS} = x + y$



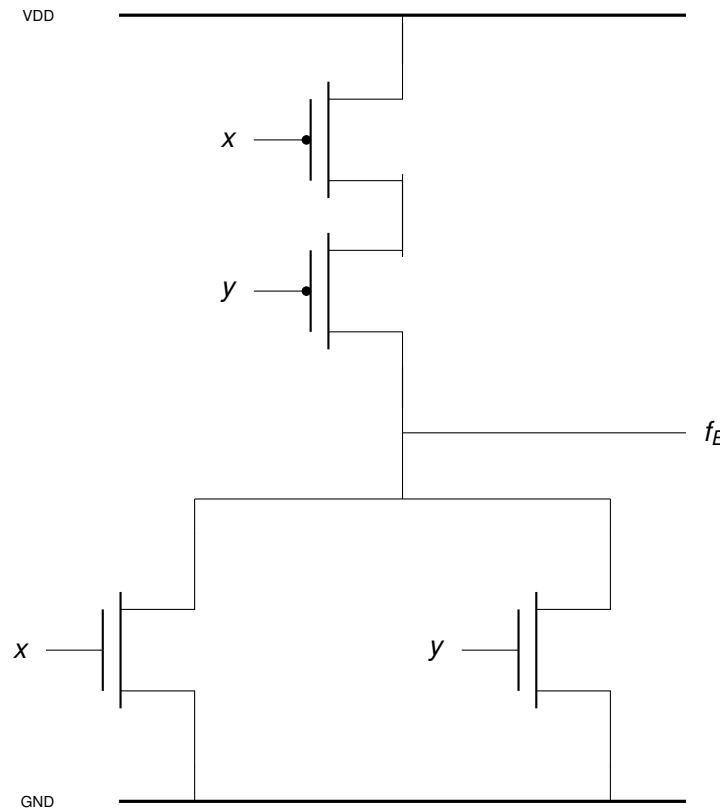
Aufgabe 1 – CMOS: Ein Beispiel (II)

Realisieren Sie die Schaltfunktion $f_B(x, y) = \overline{x + y}$.
 Bauen des Pull-Down-Netzwerkes: $f_{NMOS} = x + y$



Aufgabe 1 – CMOS: Ein Beispiel (II)

Realisieren Sie die Schaltfunktion $f_B(x, y) = \overline{x + y}$.
 Bauen des Pull-Down-Netzwerkes: $f_{NMOS} = x + y$



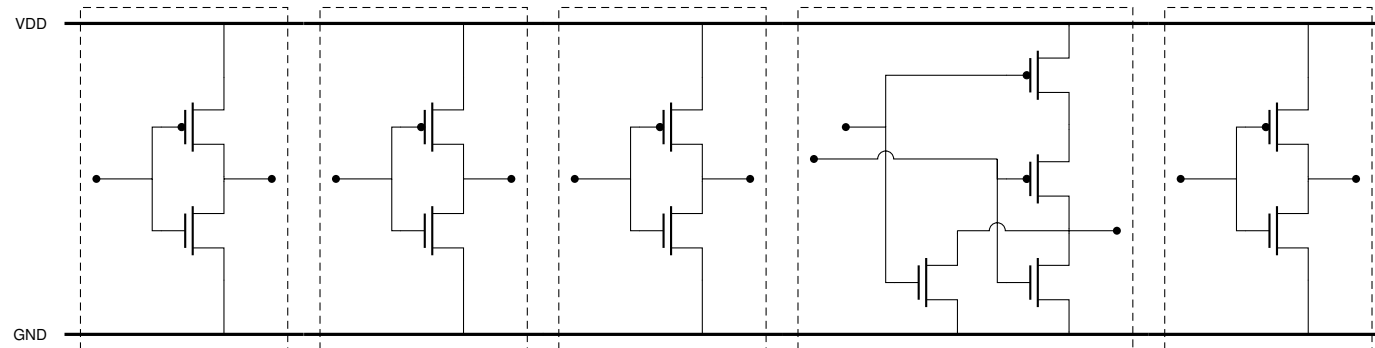
Weiteres Tafelbeispiel: $f_{B2}(x_1, x_2) = x_1 + x_2$

Aufgabe 1 – CMOS-Gatterschaltungen

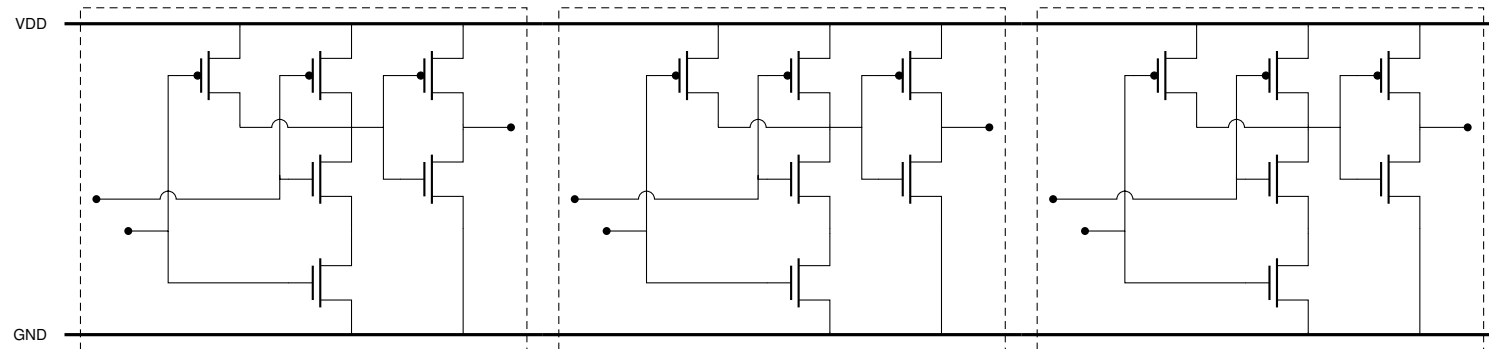
Sei die Schaltfunktion $f_1(x_3, x_2, x_2, x_0) = x_0 \overline{x_1 x_2} + x_0 \overline{x_1 x_3}$ gegeben.

- a) Standardzellen sind vorgefertigte CMOS-Realisierungen einfacher Schaltfunktionen, wie zum Beispiel Und, Oder oder Nicht, die im Baukastenprinzip zusammengesetzt werden können. Schalten Sie die folgenden Standardzellen so zusammen, dass sie f_1 realisieren (Hinweis: die einzelnen Standardzellen sind gestrichelt umrahmt).

Aufgabe 1 – CMOS-Gatterschaltungen



- X₃ ●
- X₂ ●
- X₁ ●
- X₀ ●



Aufgabe 1 – CMOS-Gatterschaltungen

Sei die Schaltfunktion $f_1(x_3, x_2, x_2, x_0) = x_0 \overline{x_1 x_2} + x_0 \overline{x_1 x_3}$ gegeben.

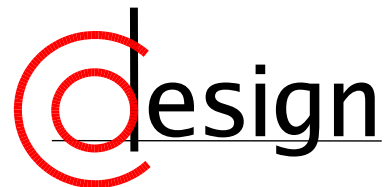
- b) Realisieren Sie die Schaltfunktion f_1 als CMOS-Schaltung mit möglichst wenig Transistoren, wobei alle Eingänge nur in der nicht invertierten Form zur Verfügung stehen.

Aufgabe 1 – CMOS-Gatterschaltungen

Sei die Schaltfunktion $f_1(x_3, x_2, x_2, x_0) = x_0 \overline{x_1 x_2} + x_0 \overline{x_1 x_3}$ gegeben.

- c) Vergleichen Sie die Anzahl benötigter Transistoren in a) und b). Für welche Anwendungsfälle eignen sich die beiden Entwurfsmethoden jeweils?

Aufgabe 2 – NAND-Technik



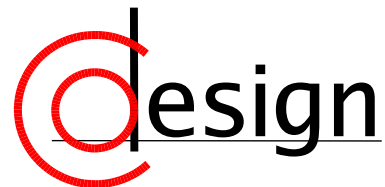
Aufgabe 2 – NAND-Technik

Realisieren Sie die Schaltfunktion

$$f_2 = \overline{DCA} + \overline{CA} + CB + \overline{DB}$$

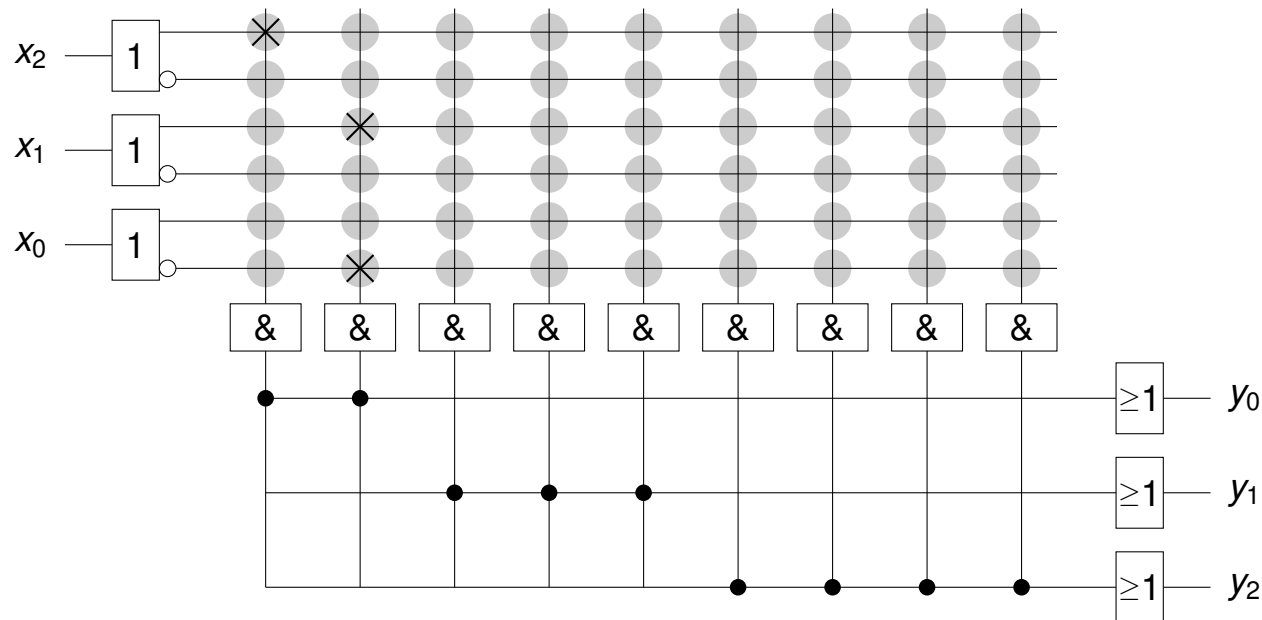
unter ausschließlicher Verwendung von NAND-Gattern mit zwei Eingängen.
Wie viele NAND-Gatter sind dann erforderlich?

Aufgabe 3 – PAL-Implementierung



Aufgabe 3 – PAL-Implementierung

Realisieren Sie einen Codeumsetzer, der eine 3-Bit-Binärzahl in einen zyklischen Gray-Code umwandelt. Orientieren Sie sich dazu an unten gegebener *Programmable Array Logic* (PAL), in der beispielhaft die Funktion $y_0(x_2, x_1, x_0) = x_2 + (x_1 \cdot \overline{x_0})$ programmiert ist:



Aufgabe 3 – PAL-Implementierung: Begriffsklärung

PAL (PROGRAMMABLE ARRAY LOGIC)

Effektiv ein Baustein, in dem konjunktive Terme über ein Feld frei programmiert werden können. Damit ist ein PAL ein Baustein zur Repräsentation einer DMF:

Erste Stufe Auswahl der Literale für konjunktive Terme (programmierbar)

Zweite Stufe Auswahl der konjungenierten Terme (fest)

Aufgabe 3 – PAL-Implementierung: Begriffsklärung

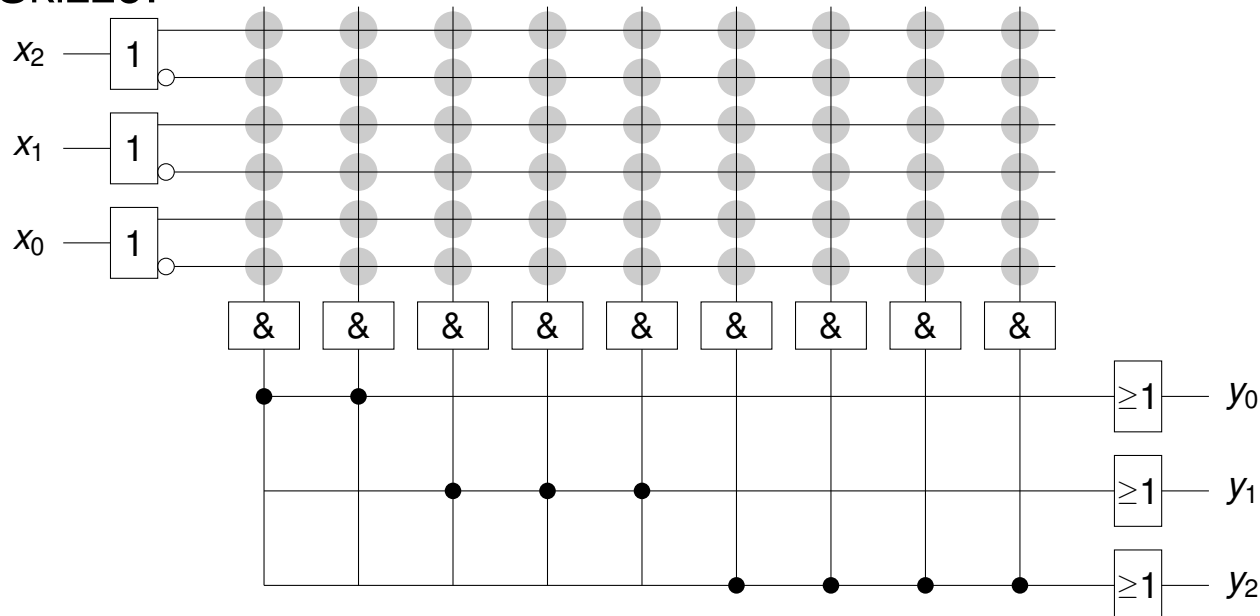
PAL (PROGRAMMABLE ARRAY LOGIC)

Effektiv ein Baustein, in dem konjunktive Terme über ein Feld frei programmiert werden können. Damit ist ein PAL ein Baustein zur Repräsentation einer DMF:

Erste Stufe Auswahl der Literale für konjunktive Terme (programmierbar)

Zweite Stufe Auswahl der konjungenierten Terme (fest)

Skizze:



Aufgabe 3 – PAL-Implementierung: Begriffsklärung

PAL (PROGRAMMABLE ARRAY LOGIC)

Effektiv ein Baustein, in dem konjunktive Terme über ein Feld frei programmiert werden können. Damit ist ein PAL ein Baustein zur Repräsentation einer DMF:

Erste Stufe Auswahl der Literale für konjunktive Terme (programmierbar)

Zweite Stufe Auswahl der konjungenierten Terme (fest)

Es gibt auch noch andere Schaltungstypen, wie zum Beispiel:

PAL PROGRAMMABLE ARRAY LOGIC – UND-Terme sind programmierbar, ODER-Terme fest.

Aufgabe 3 – PAL-Implementierung: Begriffsklärung

PAL (PROGRAMMABLE ARRAY LOGIC)

Effektiv ein Baustein, in dem konjunktive Terme über ein Feld frei programmiert werden können. Damit ist ein PAL ein Baustein zur Repräsentation einer DMF:

Erste Stufe Auswahl der Literale für konjunktive Terme (programmierbar)

Zweite Stufe Auswahl der konjungenierten Terme (fest)

Es gibt auch noch andere Schaltungstypen, wie zum Beispiel:

PLA PROGRAMMABLE LOGIC ARRAY – UND-Terme sind programmierbar, ODER-Terme ebenfalls.

Aufgabe 3 – PAL-Implementierung: Begriffsklärung

PAL (PROGRAMMABLE ARRAY LOGIC)

Effektiv ein Baustein, in dem konjunktive Terme über ein Feld frei programmiert werden können. Damit ist ein PAL ein Baustein zur Repräsentation einer DMF:

Erste Stufe Auswahl der Literale für konjunktive Terme (programmierbar)

Zweite Stufe Auswahl der konjungenierten Terme (fest)

Es gibt auch noch andere Schaltungstypen, wie zum Beispiel:

ULA UNIVERSAL LOGIC ARRAY – UND-Terme sind fest, ODER-Terme ebenfalls, man kann aber programmieren, welche Minterme ausgewählt werden.

Aufgabe 3 – PAL-Implementierung: Begriffsklärung

PAL (PROGRAMMABLE ARRAY LOGIC)

Effektiv ein Baustein, in dem konjunktive Terme über ein Feld frei programmiert werden können. Damit ist ein PAL ein Baustein zur Repräsentation einer DMF:

Erste Stufe Auswahl der Literale für konjunktive Terme (programmierbar)

Zweite Stufe Auswahl der konjungenierten Terme (fest)

Es gibt auch noch andere Schaltungstypen, wie zum Beispiel:

ROM READ-ONLY MEMORY – UND-Terme sind fest, ODER-Terme dafür programmierbar.

Aufgabe 3 – PAL-Implementierung: Begriffsklärung

PAL (PROGRAMMABLE ARRAY LOGIC)

Effektiv ein Baustein, in dem konjunktive Terme über ein Feld frei programmiert werden können. Damit ist ein PAL ein Baustein zur Repräsentation einer DMF:

Erste Stufe Auswahl der Literale für konjunktive Terme (programmierbar)

Zweite Stufe Auswahl der konjungenierten Terme (fest)

Es gibt auch noch andere Schaltungstypen, wie zum Beispiel:

PAL PROGRAMMABLE ARRAY LOGIC – UND-Terme sind programmierbar, ODER-Terme fest.

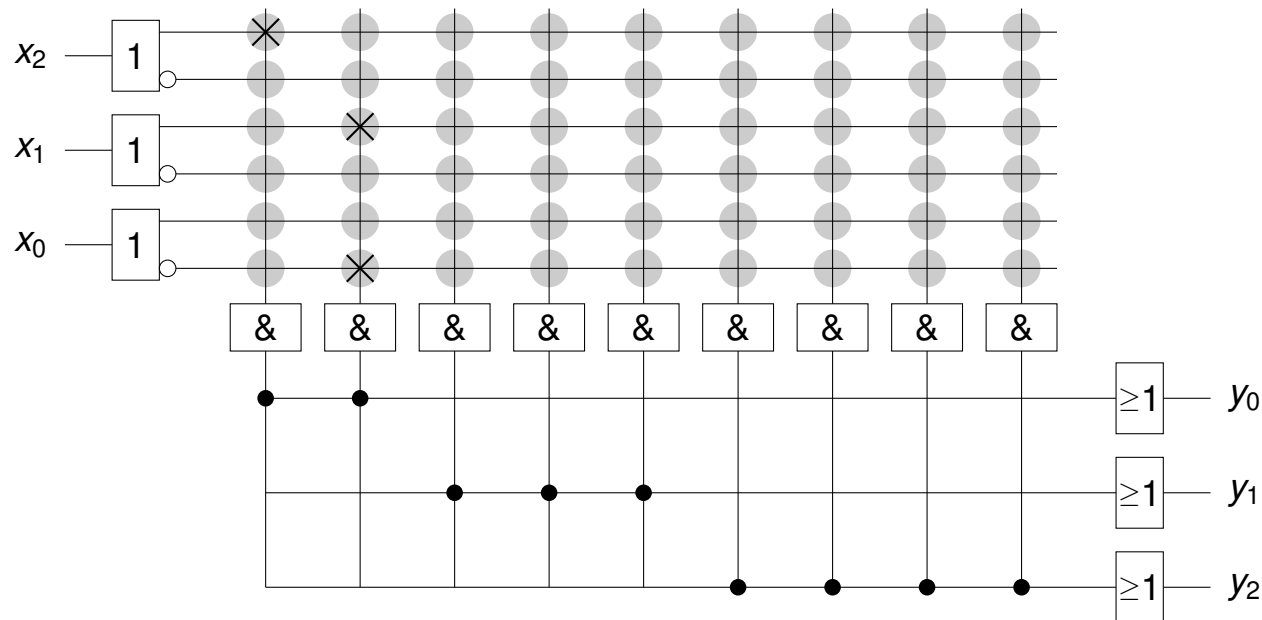
PLA PROGRAMMABLE LOGIC ARRAY – UND-Terme sind programmierbar, ODER-Terme ebenfalls.

ULA UNIVERSAL LOGIC ARRAY – UND-Terme sind fest, ODER-Terme ebenfalls, man kann aber programmieren, welche Minterme ausgewählt werden.

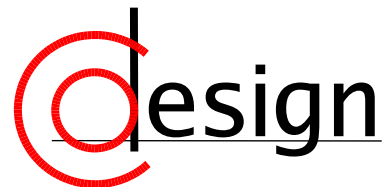
ROM READ-ONLY MEMORY – UND-Terme sind fest, ODER-Terme dafür programmierbar.

Aufgabe 3 – PAL-Implementierung

Realisieren Sie einen Codeumsetzer, der eine 3-Bit-Binärzahl in einen zyklischen Gray-Code umwandelt. Orientieren Sie sich dazu an unten gegebener *Programmable Array Logic* (PAL), in der beispielhaft die Funktion $y_0(x_2, x_1, x_0) = x_2 + (x_1 \cdot \overline{x_0})$ programmiert ist:



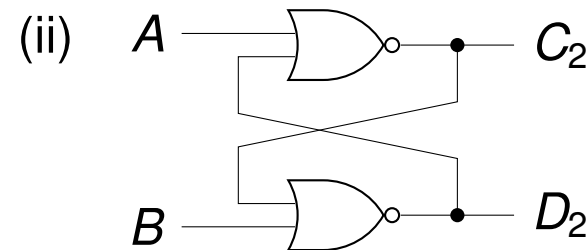
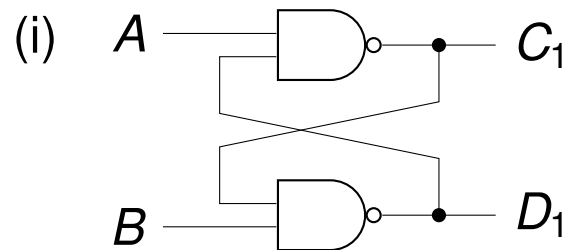
Aufgabe 4 – Latches und Flipflops



Aufgabe 4 – Latches und Flipflops

Die Verzögerungszeit jedes Logikgatters in dieser Aufgabe betrage $\tau = 1$ ns.

- a) An die Eingänge (A , B) der unten stehenden Schaltungen (i) und (ii) werden nacheinander folgende Werte angelegt: (0, 1), (0, 0), (1, 1), (1, 0), (1, 1) und (0, 0). Geben Sie jeweils die Ausgangswerte von (i) und (ii) an und benennen Sie die Signale A bis D_2 sinnvoll.



Aufgabe 4 – Latches und Flipflops: Begriffklärung

Speicherelement

Ein Speicherelement ist ein Gerät/Modul/Element, das einen vorher angelegten Wert speichert.

Aufgabe 4 – Latches und Flipflops: Begriffklärung

Speicherelement

Ein Speicherelement ist ein Gerät/Modul/Element, das einen vorher angelegten Wert speichert.

Eigenschaften von Speicherelementen

Speicherelemente können ...

Operation	Q^t	Q^{t+1}
S(etze)	—	1
R(ücksetze)	—	0
N(ix)	0	0
	1	1

Operationen eines Speicherelements

Aufgabe 4 – Latches und Flipflops: Begriffklärung

Speicherelement

Ein Speicherelement ist ein Gerät/Modul/Element, das einen vorher angelegten Wert speichert.

Eigenschaften von Speicherelementen

Speicherelemente können ...

asynchron sein, wenn Änderungen jederzeit möglich sind.

Operation	Q^t	Q^{t+1}
S(etze)	—	1
R(ücksetze)	—	0
N(ix)	0	0
	1	1

Operationen eines Speicherelements

Aufgabe 4 – Latches und Flipflops: Begriffklärung

Speicherelement

Ein Speicherelement ist ein Gerät/Modul/Element, das einen vorher angelegten Wert speichert.

Eigenschaften von Speicherelementen

Speicherelemente können ...

asynchron sein, wenn Änderungen jederzeit möglich sind.

synchron sein, wenn Änderungen nur zu vorher festgelegten Momenten möglich sind. Diese Momente können ...

Operation	Q^t	Q^{t+1}
S(etze)	—	1
R(ücksetze)	—	0
N(ix)	0	0
	1	1

Operationen eines Speicherelements

Aufgabe 4 – Latches und Flipflops: Begriffklärung

Speicherelement

Ein Speicherelement ist ein Gerät/Modul/Element, das einen vorher angelegten Wert speichert.

Eigenschaften von Speicherelementen

Speicherelemente können ...

asynchron sein, wenn Änderungen jederzeit möglich sind.

synchron sein, wenn Änderungen nur zu vorher festgelegten Momenten möglich sind. Diese Momente können ...

pegelabhängig sein. Mann nennt diese Elemente dann auch **pegelgesteuert**.

Operation	Q^t	Q^{t+1}
S(etze)	—	1
R(ücksetze)	—	0
N(ix)	0	0
	1	1

Operationen eines Speicherelements

Aufgabe 4 – Latches und Flipflops: Begriffklärung

Speicherelement

Ein Speicherelement ist ein Gerät/Modul/Element, das einen vorher angelegten Wert speichert.

Eigenschaften von Speicherelementen

Speicherelemente können ...

asynchron sein, wenn Änderungen jederzeit möglich sind.

synchron sein, wenn Änderungen nur zu vorher festgelegten Momenten möglich sind. Diese Momente können ...

pegelabhängig sein. Mann nennt diese Elemente dann auch **pegelgesteuert**.

taktflankenabhängig sein. Mann nennt diese Elemente dann auch **taktflankengesteuert**.

Es können zu **steigender**, **fallender** oder **zu beiden** Flanken Änderungen möglich sein.

Operation	Q^t	Q^{t+1}
S(etze)	—	1
R(ücksetze)	—	0
N(ix)	0	0
	1	1

Operationen eines Speicherelements

Aufgabe 4 – Latches und Flipflops: Begriffklärung

Speicherelement

Ein Speicherelement ist ein Gerät/Modul/Element, das einen vorher angelegten Wert speichert.

Flipflops

Flipflops sind Elemente zur Speicherung eines Bits.

Sie sind – bei uns – **rein taktflankengesteuert**.

Aufgabe 4 – Latches und Flipflops: Begriffklärung

Speicherelement

Ein Speicherelement ist ein Gerät/Modul/Element, das einen vorher angelegten Wert speichert.

Latches

Latches sind ebenfalls Elemente zur Speicherung eines Bits, ähnlich zu den Flipflops.

Im Gegensatz zu Flipflops sind sie aber **rein pegelgesteuert..**

Aufgabe 4 – Latches und Flipflops: Begriffklärung

Speicherelement

Ein Speicherelement ist ein Gerät/Modul/Element, das einen vorher angelegten Wert speichert.

Flipflops

Flipflops sind Elemente zur Speicherung eines Bits.

Sie sind – bei uns – **rein taktflankengesteuert**.

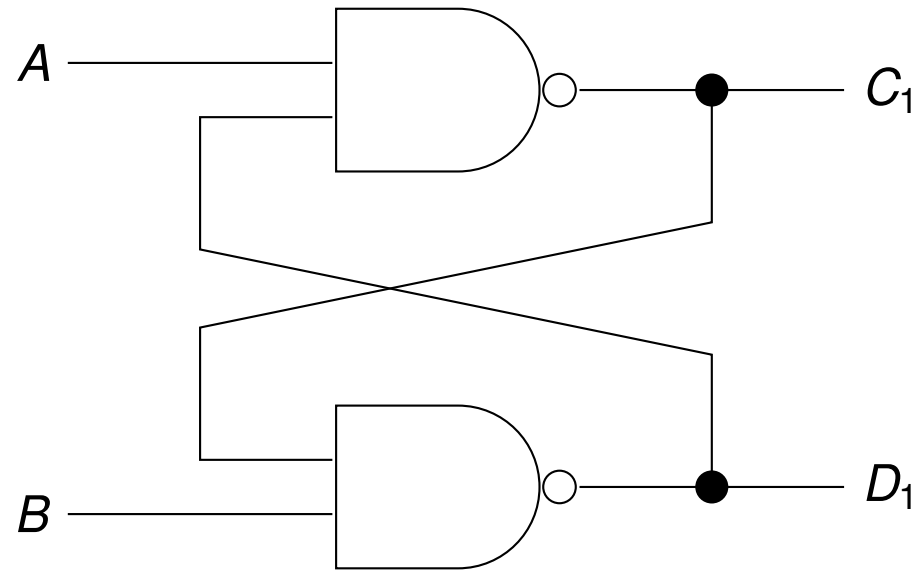
Latches

Latches sind ebenfalls Elemente zur Speicherung eines Bits, ähnlich zu den Flipflops.

Im Gegensatz zu Flipflops sind sie aber **rein pegelgesteuert**.

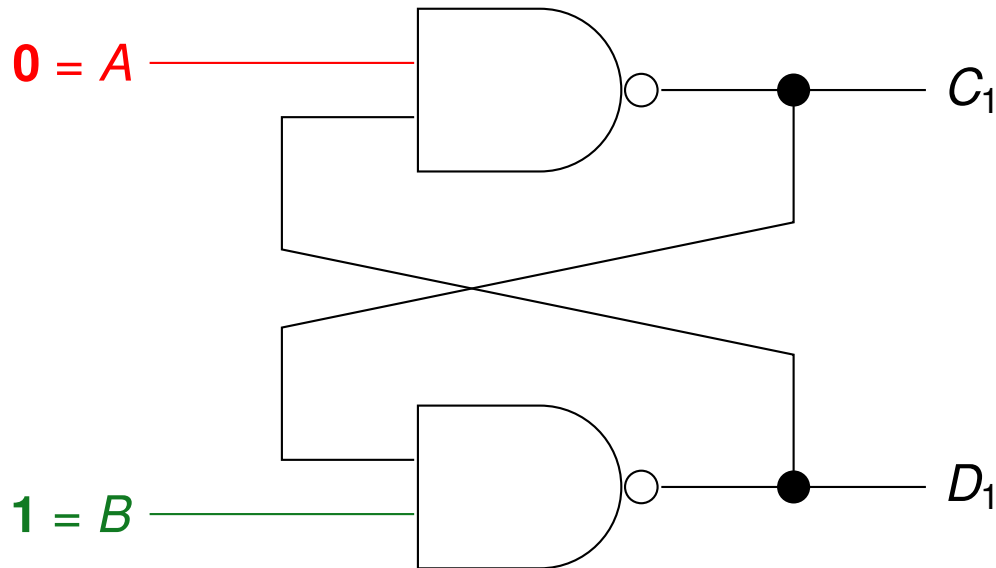
Aufgabe 4 – Latches und Flipflops

Anliegender Wert (**fett** hervorgehoben):



Aufgabe 4 – Latches und Flipflops

Anliegender Wert (**fett** hervorgehoben): **(0,1)**

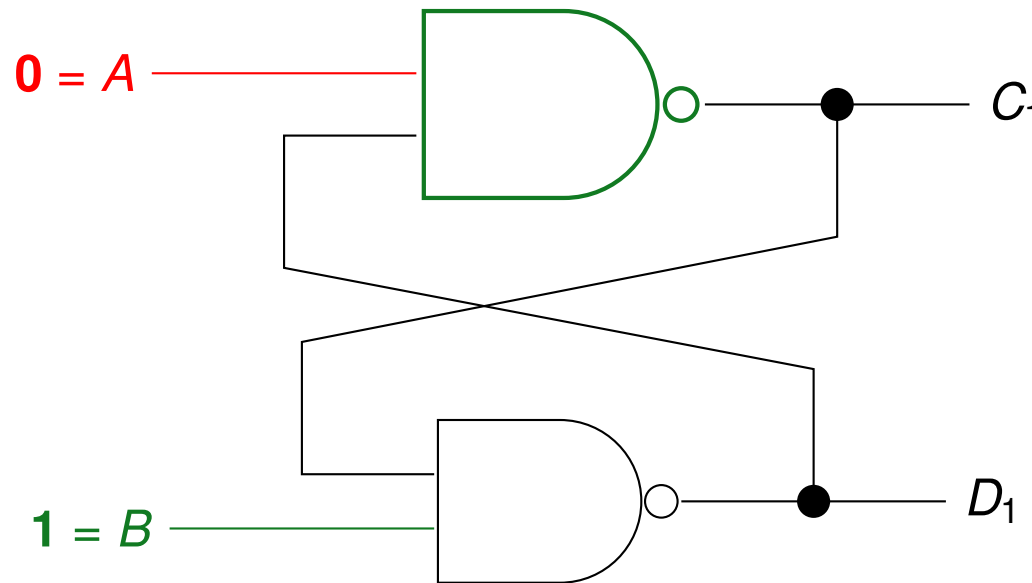


Hat ein NAND mindestens **eine** 0 im Eingang, so ist der Ausgang 1.

Hat ein NAND **keine** 0 im Eingang (sprich: nur Einsen), so ist der Ausgang 0.

Aufgabe 4 – Latches und Flipflops

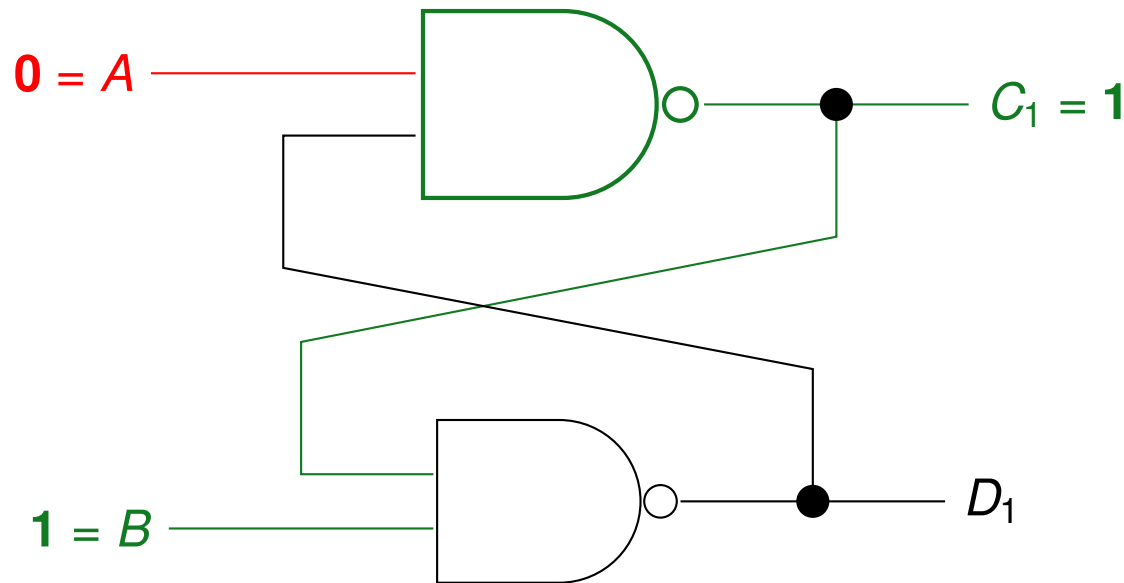
Anliegender Wert (**fett** hervorgehoben): **(0,1)**



Hat ein NAND mindestens **eine** 0 im Eingang, so ist der Ausgang 1.
 Hat ein NAND **keine** 0 im Eingang (sprich: nur Einsen), so ist der Ausgang 0.

Aufgabe 4 – Latches und Flipflops

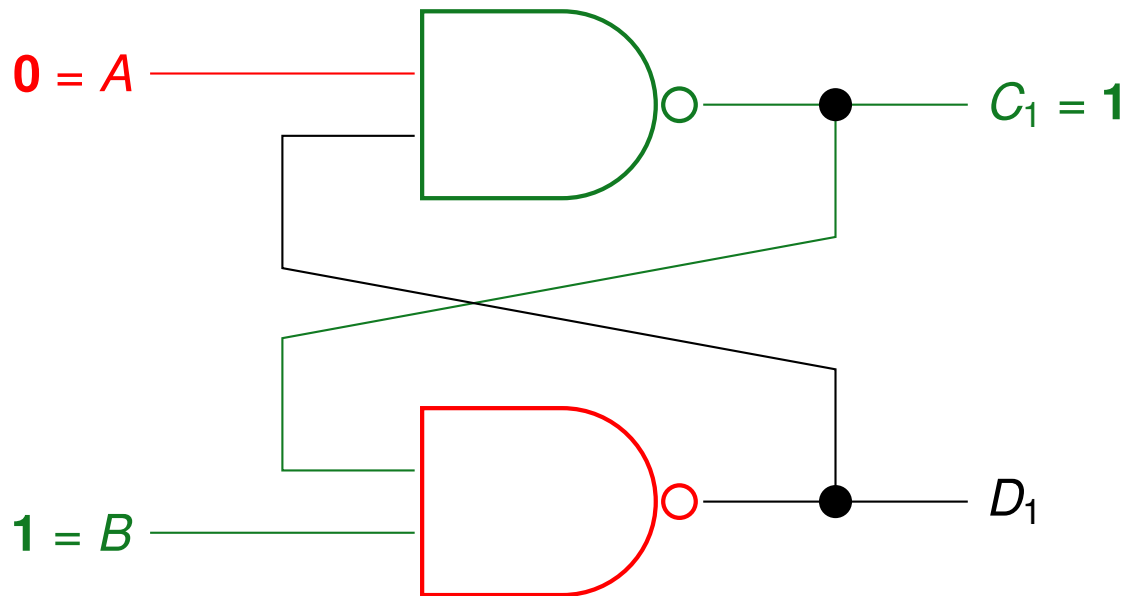
Anliegender Wert (**fett** hervorgehoben): **(0,1)**



Hat ein NAND mindestens **eine** 0 im Eingang, so ist der Ausgang 1.
 Hat ein NAND **keine** 0 im Eingang (sprich: nur Einsen), so ist der Ausgang 0.

Aufgabe 4 – Latches und Flipflops

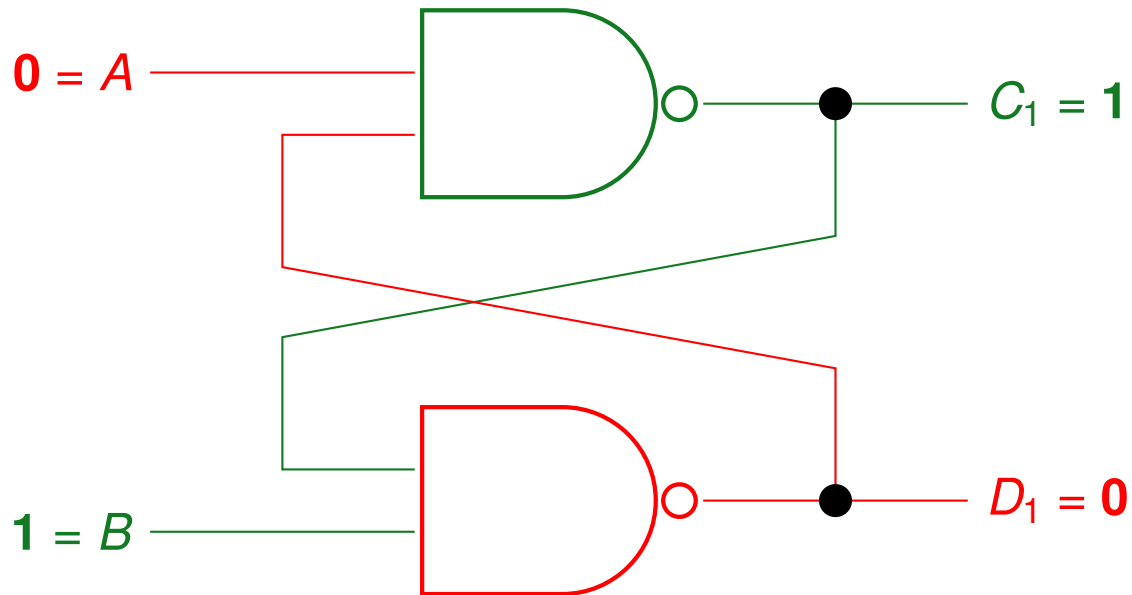
Anliegender Wert (**fett** hervorgehoben): **(0,1)**



Hat ein NAND mindestens **eine** 0 im Eingang, so ist der Ausgang 1.
 Hat ein NAND **keine** 0 im Eingang (sprich: nur Einsen), so ist der Ausgang 0.

Aufgabe 4 – Latches und Flipflops

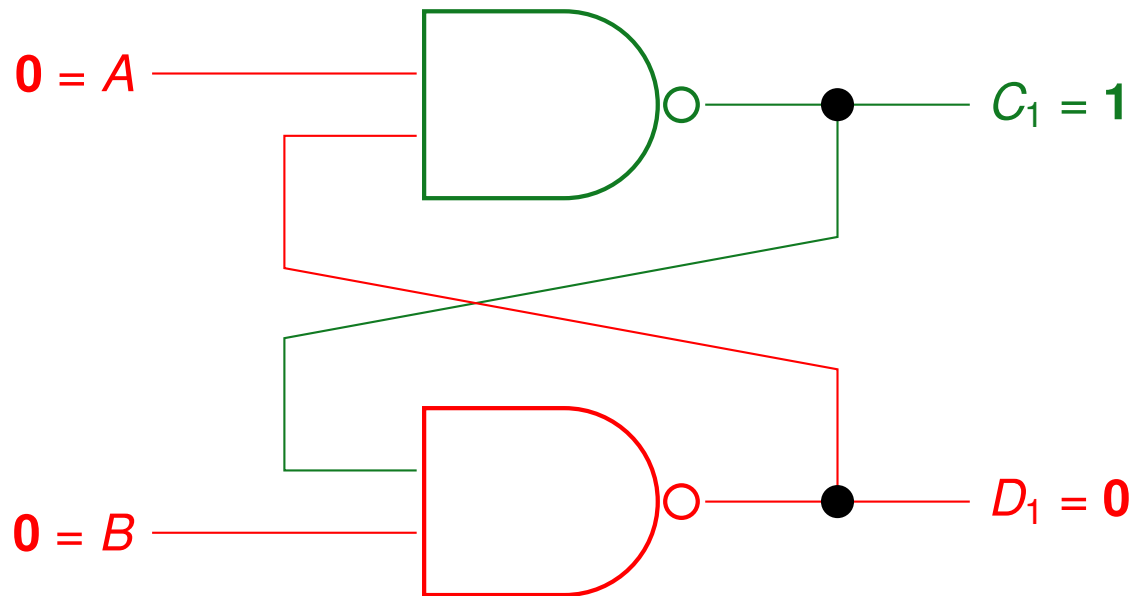
Anliegender Wert (**fett** hervorgehoben): **(0,1)**



Hat ein NAND mindestens **eine** 0 im Eingang, so ist der Ausgang 1.
 Hat ein NAND **keine** 0 im Eingang (sprich: nur Einsen), so ist der Ausgang 0.

Aufgabe 4 – Latches und Flipflops

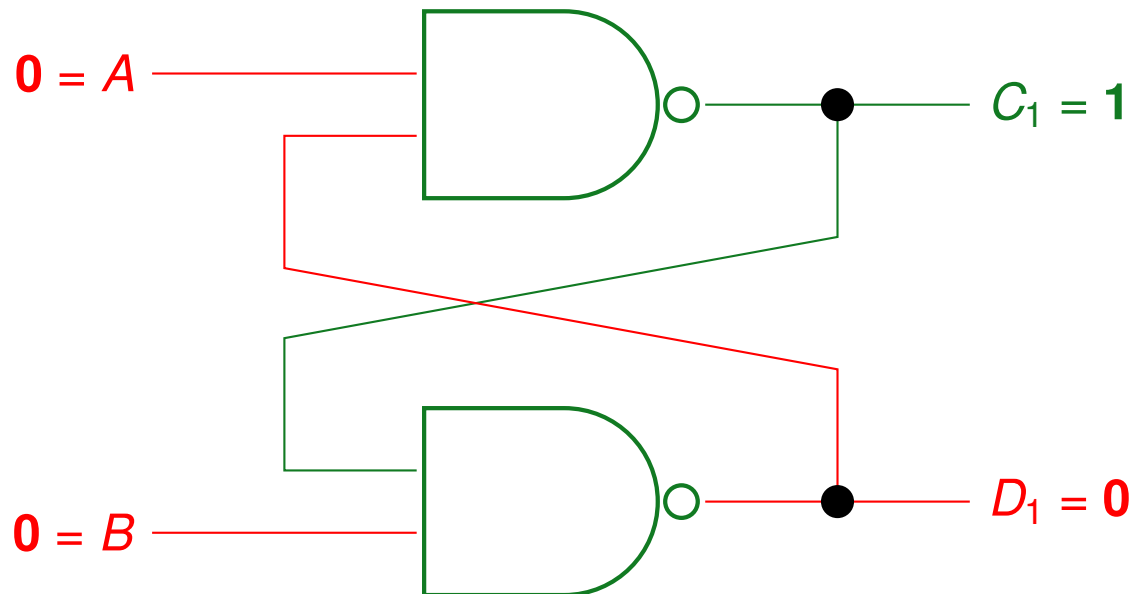
Anliegender Wert (**fett** hervorgehoben): $(0,1) \mapsto (0, 0)$



Hat ein NAND mindestens **eine** 0 im Eingang, so ist der Ausgang 1.
 Hat ein NAND **keine** 0 im Eingang (sprich: nur Einsen), so ist der Ausgang 0.

Aufgabe 4 – Latches und Flipflops

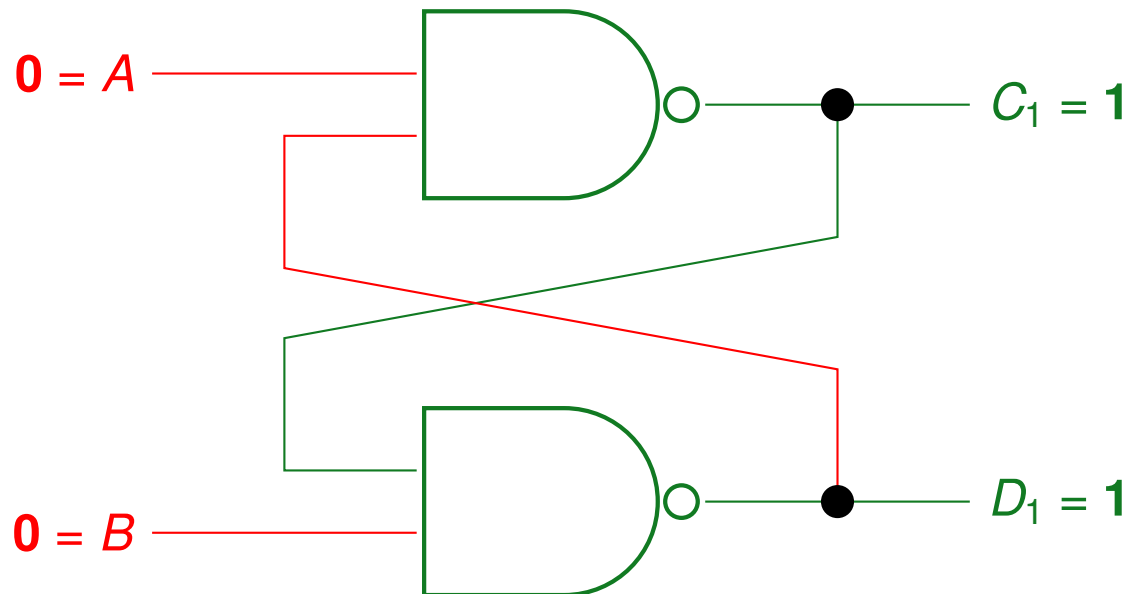
Anliegender Wert (**fett** hervorgehoben): $(0,1) \mapsto (0, 0)$



Hat ein NAND mindestens **eine** 0 im Eingang, so ist der Ausgang 1.
 Hat ein NAND **keine** 0 im Eingang (sprich: nur Einsen), so ist der Ausgang 0.

Aufgabe 4 – Latches und Flipflops

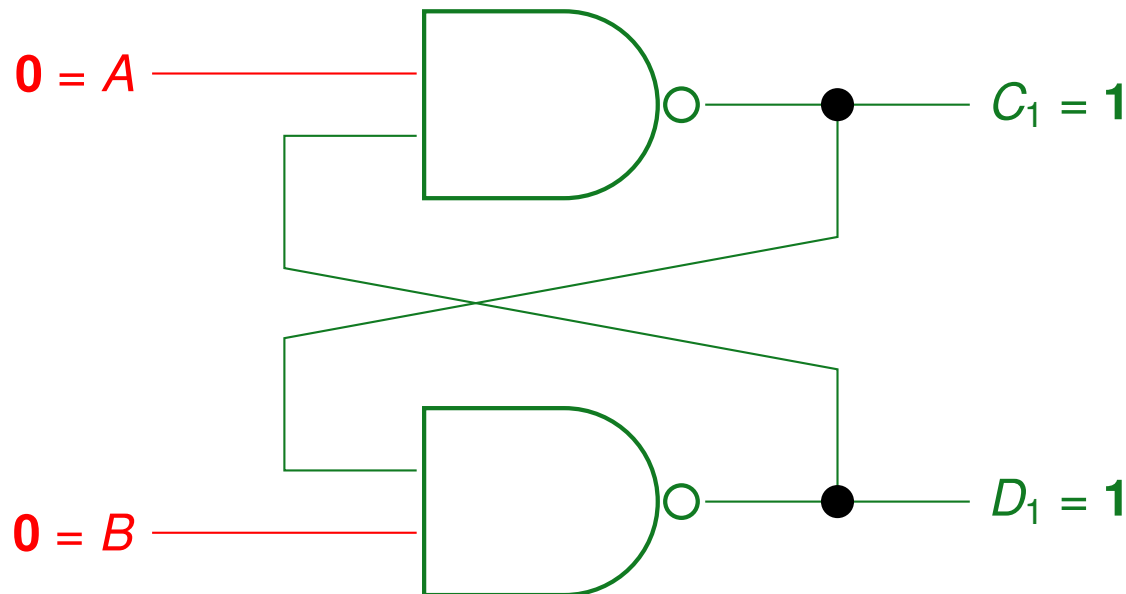
Anliegender Wert (**fett** hervorgehoben): $(0,1) \mapsto (0, 0)$



Hat ein NAND mindestens **eine** 0 im Eingang, so ist der Ausgang 1.
 Hat ein NAND **keine** 0 im Eingang (sprich: nur Einsen), so ist der Ausgang 0.

Aufgabe 4 – Latches und Flipflops

Anliegender Wert (**fett** hervorgehoben): $(0,1) \mapsto (0, 0)$

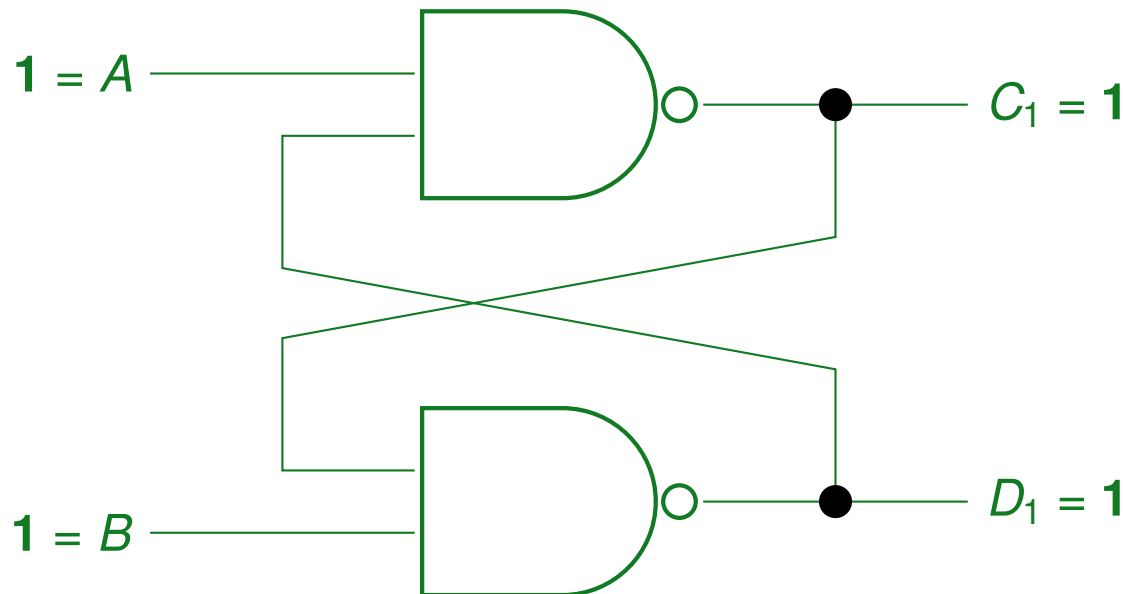


Hat ein NAND mindestens **eine** 0 im Eingang, so ist der Ausgang 1.

Hat ein NAND **keine** 0 im Eingang (sprich: nur Einsen), so ist der Ausgang 0.

Aufgabe 4 – Latches und Flipflops

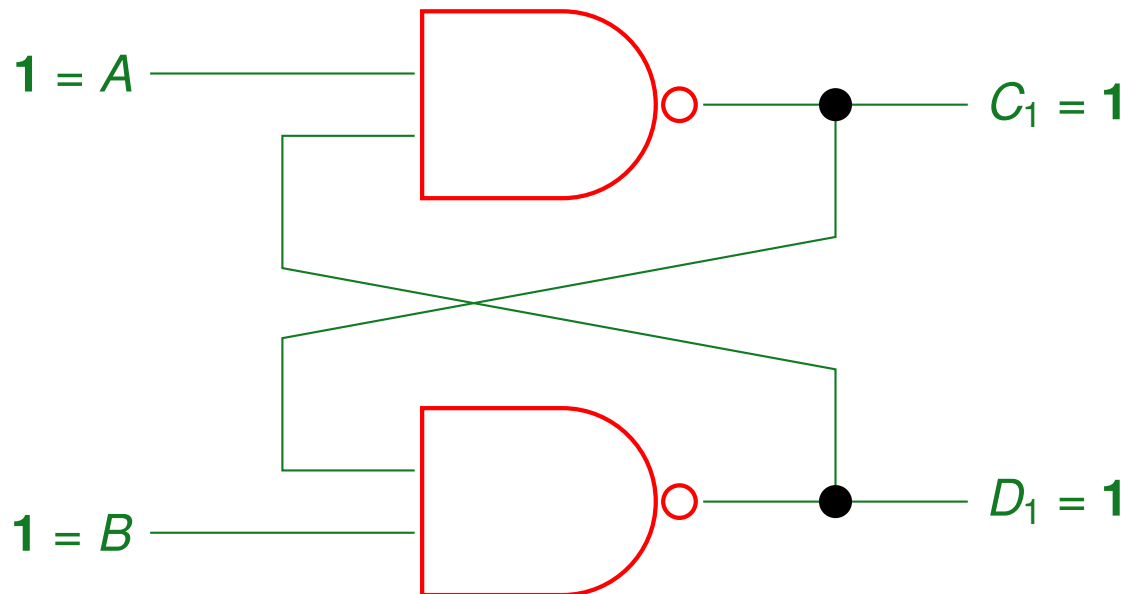
Anliegender Wert (**fett** hervorgehoben): $(0, 1) \mapsto (0, 0) \mapsto (1, 1)$



Hat ein NAND mindestens **eine** 0 im Eingang, so ist der Ausgang 1.
 Hat ein NAND **keine** 0 im Eingang (sprich: nur Einsen), so ist der Ausgang 0.

Aufgabe 4 – Latches und Flipflops

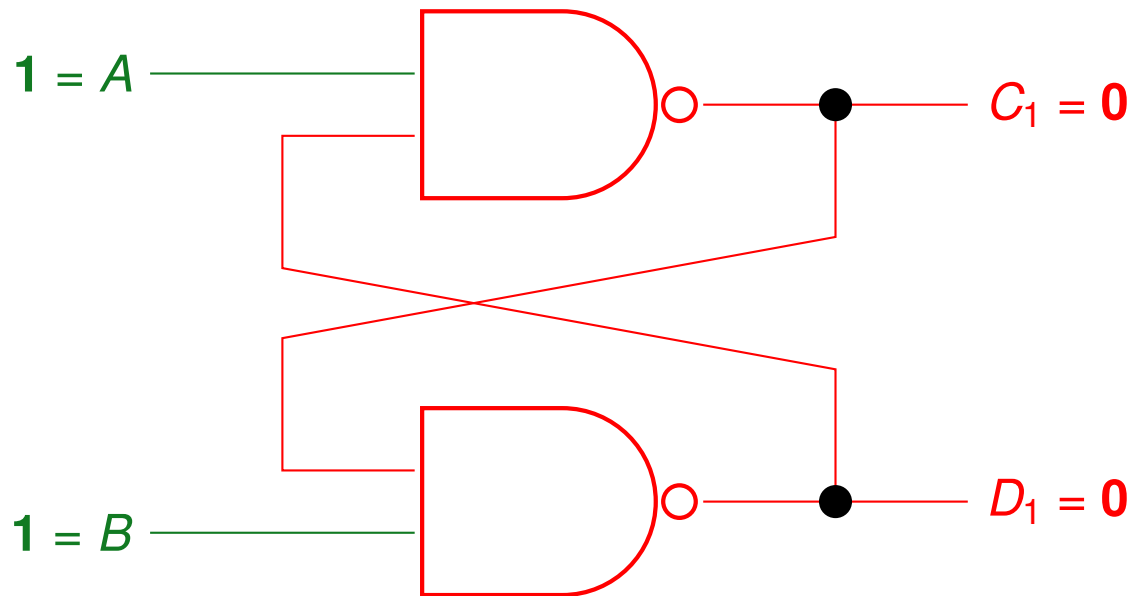
Anliegender Wert (**fett** hervorgehoben): $(0, 1) \mapsto (0, 0) \mapsto (1, 1)$



Hat ein NAND mindestens **eine** 0 im Eingang, so ist der Ausgang 1.
 Hat ein NAND **keine** 0 im Eingang (sprich: nur Einsen), so ist der Ausgang 0.

Aufgabe 4 – Latches und Flipflops

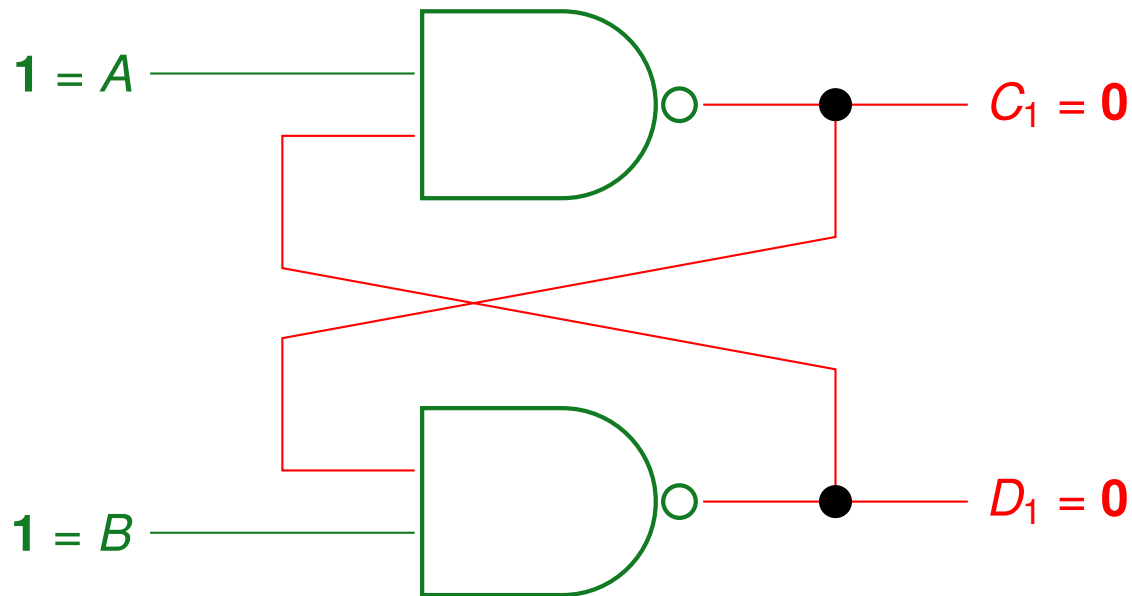
Anliegender Wert (**fett** hervorgehoben): $(0,1) \mapsto (0, 0) \mapsto (1, 1)$



Hat ein NAND mindestens **eine** 0 im Eingang, so ist der Ausgang 1.
 Hat ein NAND **keine** 0 im Eingang (sprich: nur Einsen), so ist der Ausgang 0.

Aufgabe 4 – Latches und Flipflops

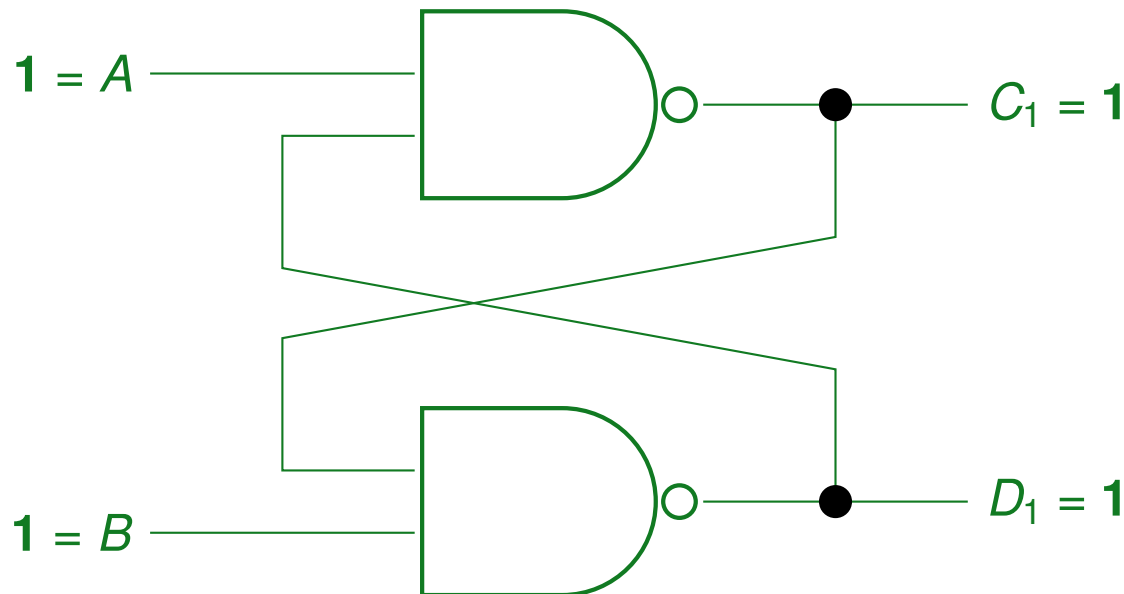
Anliegender Wert (**fett** hervorgehoben): $(0,1) \mapsto (0, 0) \mapsto (1, 1)$



Hat ein NAND mindestens **eine** 0 im Eingang, so ist der Ausgang 1.
 Hat ein NAND **keine** 0 im Eingang (sprich: nur Einsen), so ist der Ausgang 0.

Aufgabe 4 – Latches und Flipflops

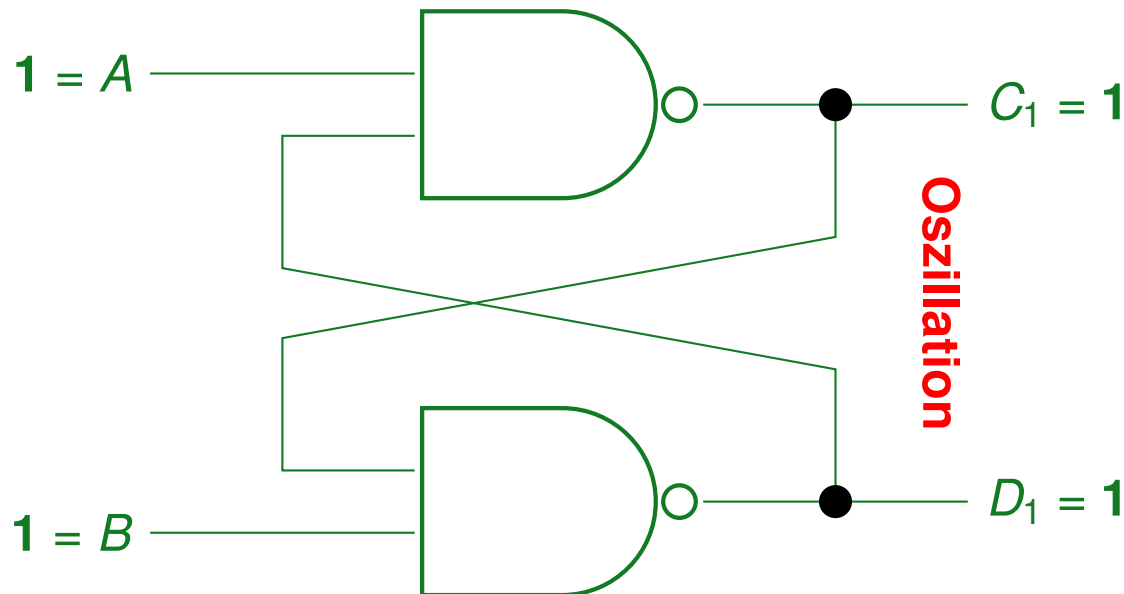
Anliegender Wert (**fett** hervorgehoben): $(0,1) \mapsto (0, 0) \mapsto (1, 1)$



Hat ein NAND mindestens **eine** 0 im Eingang, so ist der Ausgang 1.
 Hat ein NAND **keine** 0 im Eingang (sprich: nur Einsen), so ist der Ausgang 0.

Aufgabe 4 – Latches und Flipflops

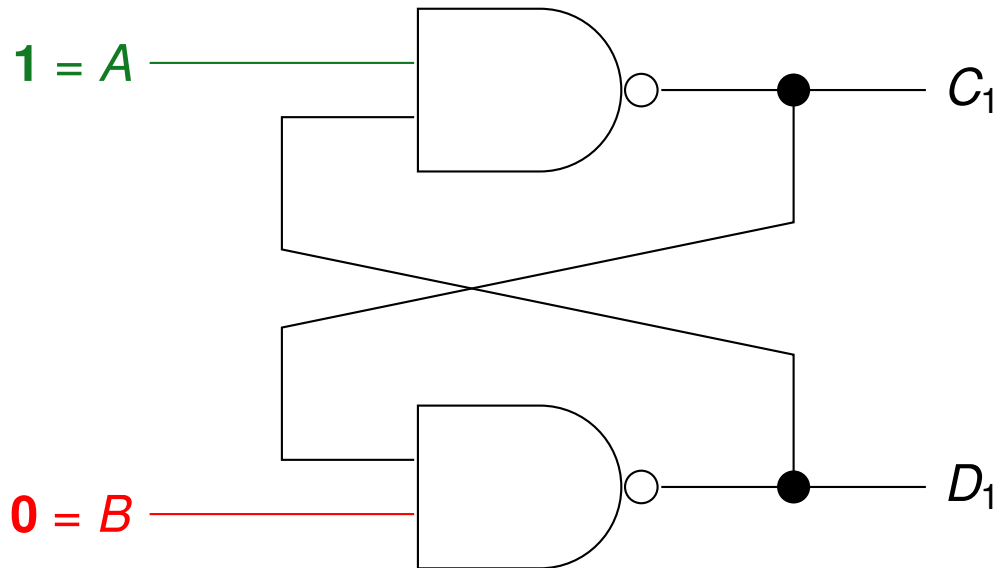
Anliegender Wert (**fett** hervorgehoben): $(0,1) \mapsto (0, 0) \mapsto (1, 1)$



Hat ein NAND mindestens **eine** 0 im Eingang, so ist der Ausgang 1.
 Hat ein NAND **keine** 0 im Eingang (sprich: nur Einsen), so ist der Ausgang 0.

Aufgabe 4 – Latches und Flipflops

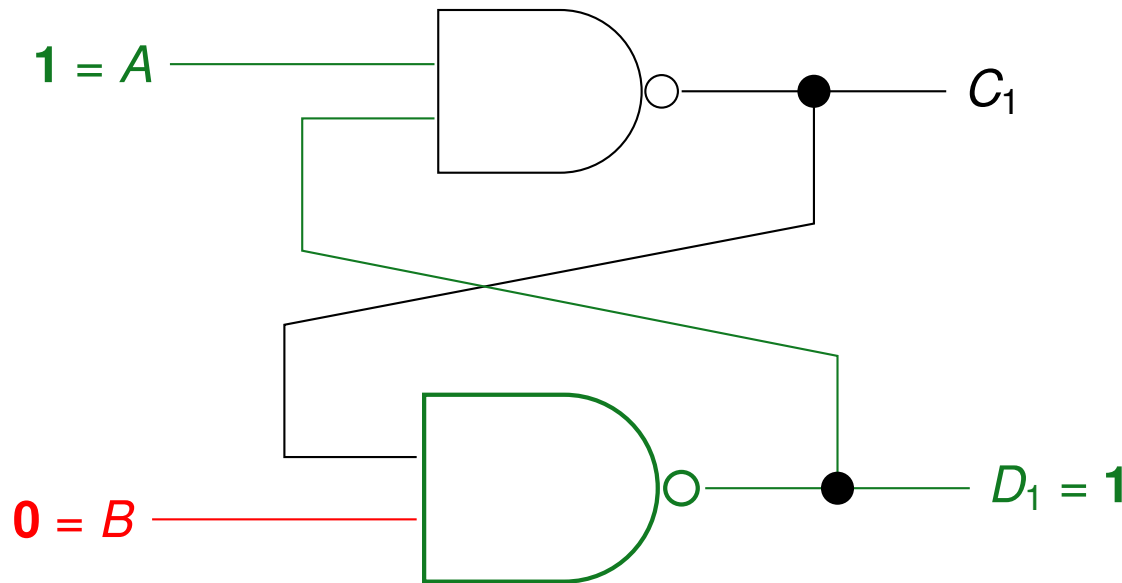
Anliegender Wert (**fett** hervorgehoben): ... $\mapsto (0, 0) \mapsto (1, 1) \mapsto (1, 0)$



Hat ein NAND mindestens **eine** 0 im Eingang, so ist der Ausgang 1.
 Hat ein NAND **keine** 0 im Eingang (sprich: nur Einsen), so ist der Ausgang 0.

Aufgabe 4 – Latches und Flipflops

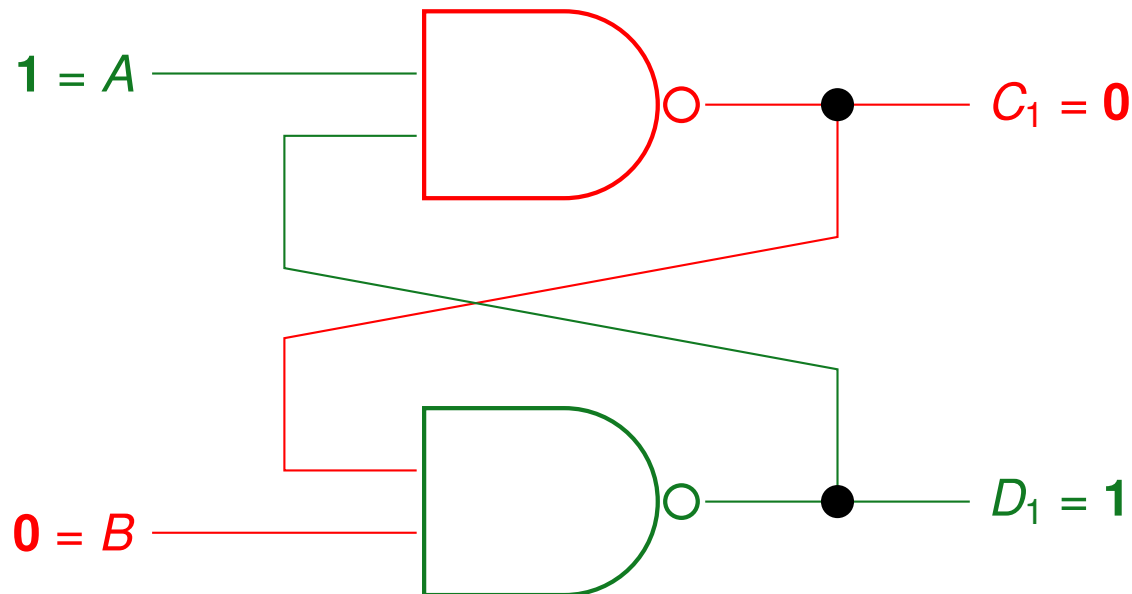
Anliegender Wert (**fett** hervorgehoben): ... $\mapsto (0, 0) \mapsto (1, 1) \mapsto (1, 0)$



Hat ein NAND mindestens **eine** 0 im Eingang, so ist der Ausgang 1.
 Hat ein NAND **keine** 0 im Eingang (sprich: nur Einsen), so ist der Ausgang 0.

Aufgabe 4 – Latches und Flipflops

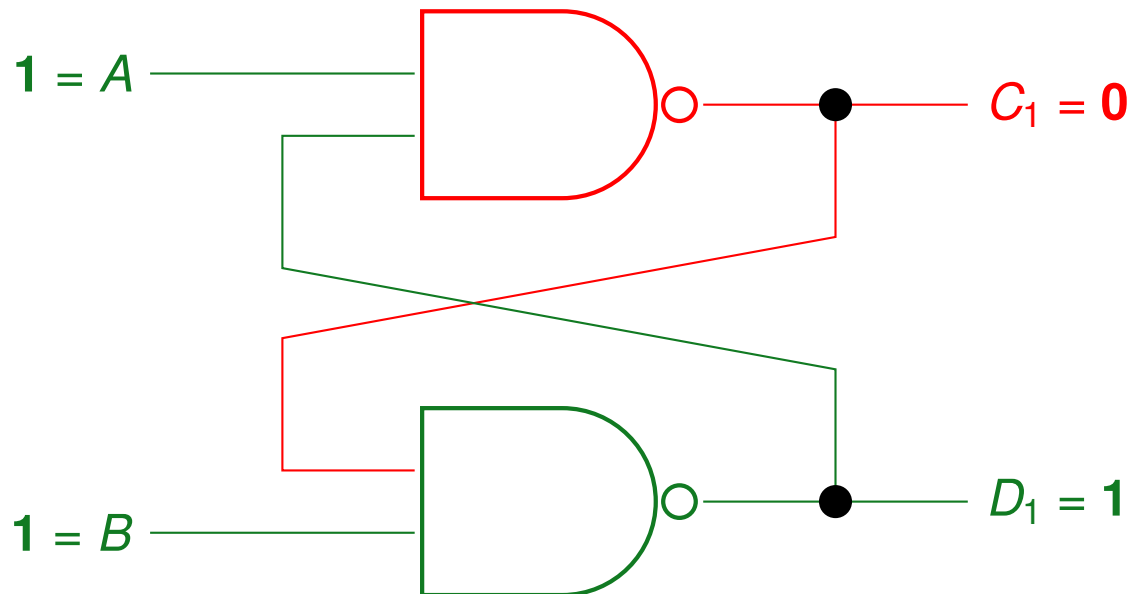
Anliegender Wert (**fett** hervorgehoben): ... $\mapsto (0, 0) \mapsto (1, 1) \mapsto (1, 0)$



Hat ein NAND mindestens **eine** 0 im Eingang, so ist der Ausgang 1.
 Hat ein NAND **keine** 0 im Eingang (sprich: nur Einsen), so ist der Ausgang 0.

Aufgabe 4 – Latches und Flipflops

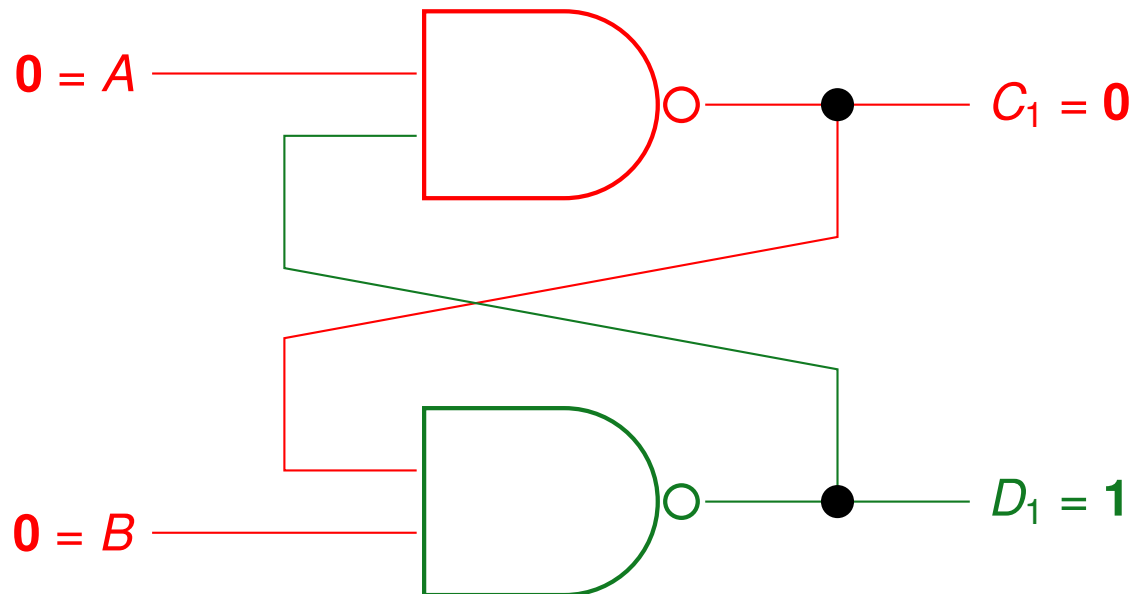
Anliegender Wert (**fett** hervorgehoben): ... $\mapsto (1, 1) \mapsto (1, 0) \mapsto (1, 1)$



Hat ein NAND mindestens **eine** 0 im Eingang, so ist der Ausgang 1.
 Hat ein NAND **keine** 0 im Eingang (sprich: nur Einsen), so ist der Ausgang 0.

Aufgabe 4 – Latches und Flipflops

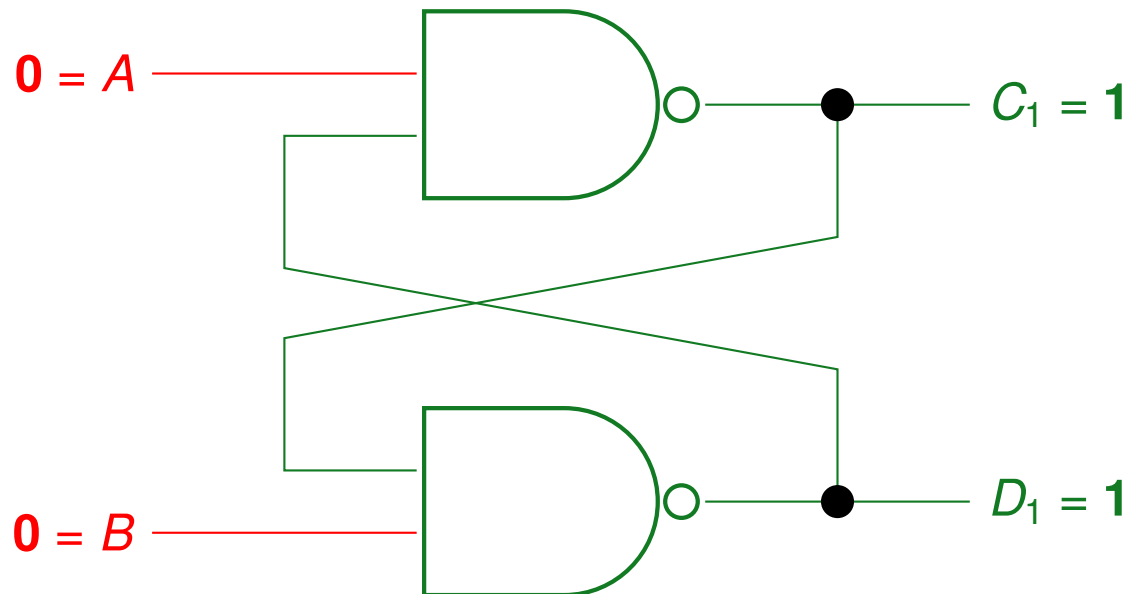
Anliegender Wert (**fett** hervorgehoben): ... $\mapsto (1, 0) \mapsto (1, 1) \mapsto (0, 0)$



Hat ein NAND mindestens **eine** 0 im Eingang, so ist der Ausgang 1.
 Hat ein NAND **keine** 0 im Eingang (sprich: nur Einsen), so ist der Ausgang 0.

Aufgabe 4 – Latches und Flipflops

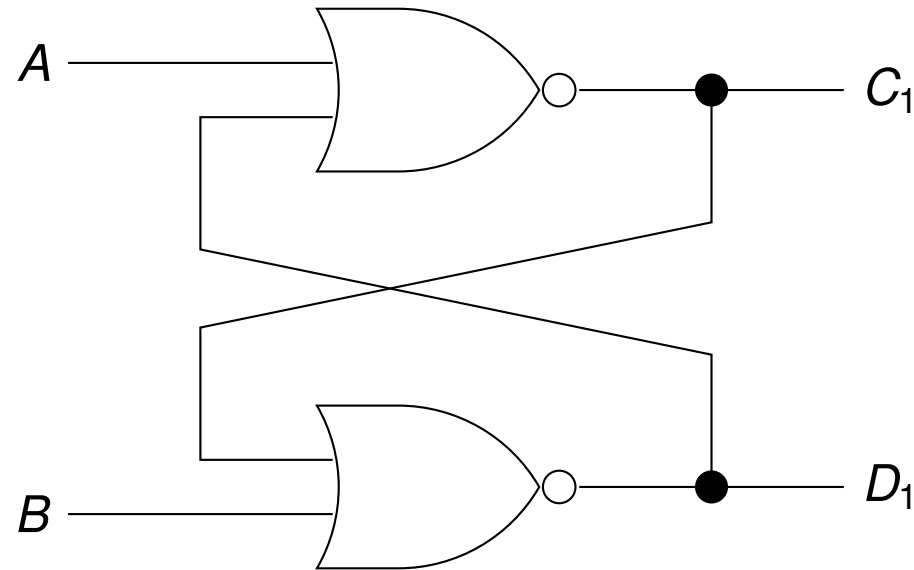
Anliegender Wert (**fett** hervorgehoben): ... $\mapsto (1, 0) \mapsto (1, 1) \mapsto (0, 0)$



Hat ein NAND mindestens **eine** 0 im Eingang, so ist der Ausgang 1.
 Hat ein NAND **keine** 0 im Eingang (sprich: nur Einsen), so ist der Ausgang 0.

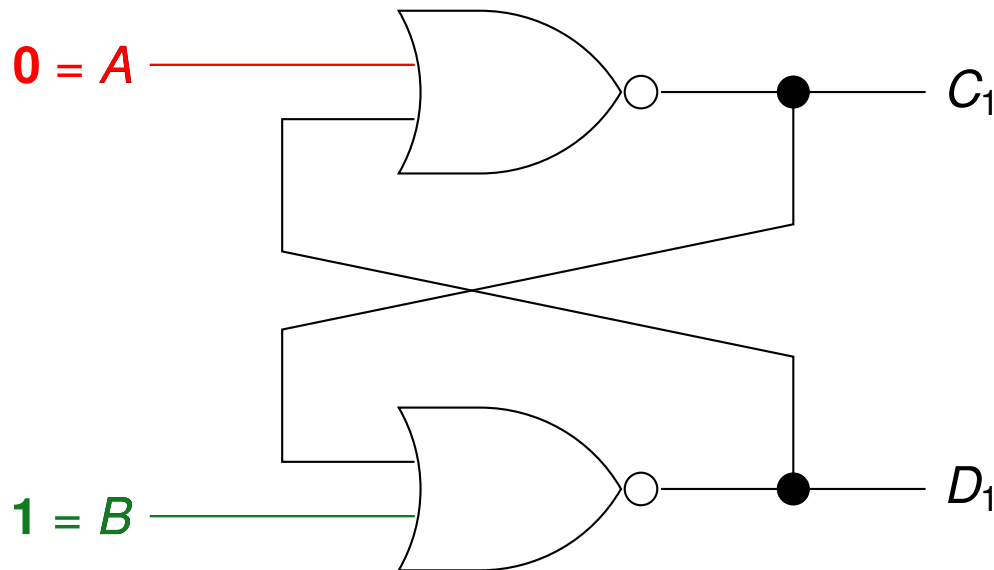
Aufgabe 4 – Latches und Flipflops

Anliegender Wert (**fett** hervorgehoben):



Aufgabe 4 – Latches und Flipflops

Anliegender Wert (**fett** hervorgehoben): **(0,1)**

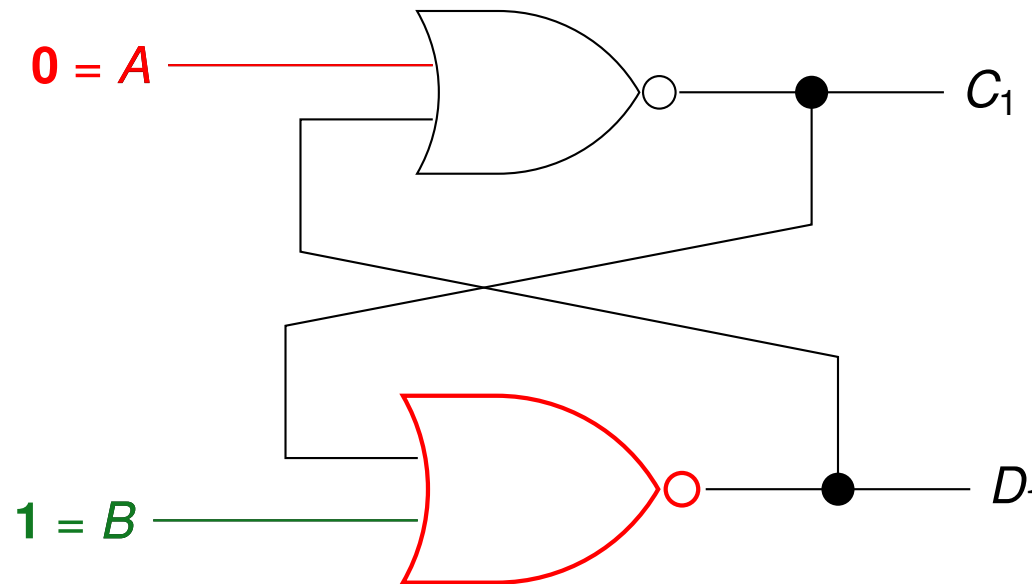


Hat ein NOR mindestens **eine** 1 im Eingang, so ist der Ausgang 0.

Hat ein NOR **keine** 1 im Eingang (sprich: nur Nullen), so ist der Ausgang 1.

Aufgabe 4 – Latches und Flipflops

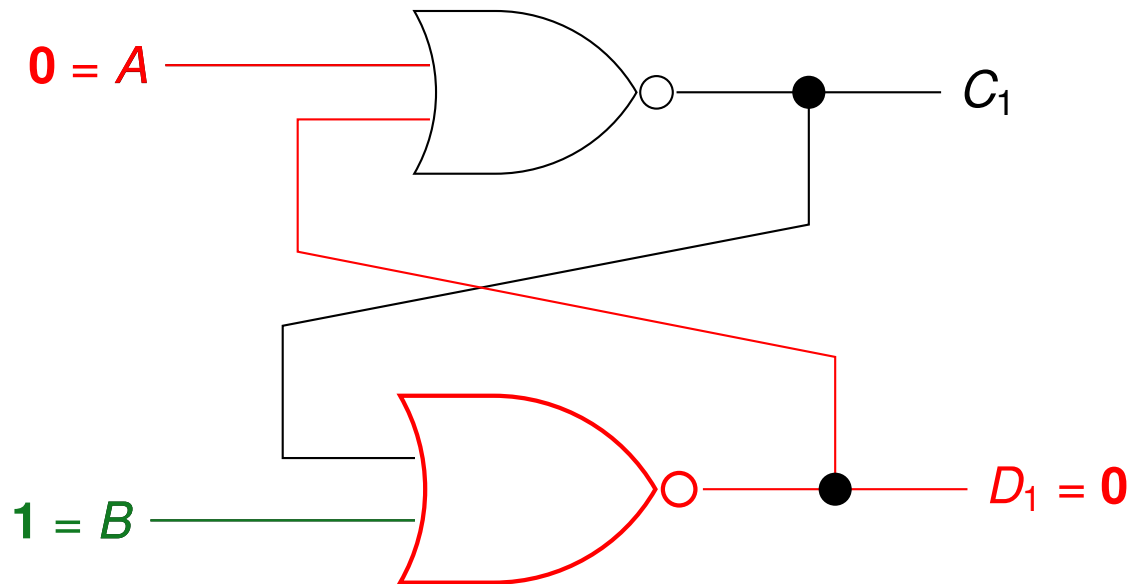
Anliegender Wert (**fett** hervorgehoben): **(0,1)**



Hat ein NOR mindestens **eine** 1 im Eingang, so ist der Ausgang 0.
 Hat ein NOR **keine** 1 im Eingang (sprich: nur Nullen), so ist der Ausgang 1.

Aufgabe 4 – Latches und Flipflops

Anliegender Wert (**fett** hervorgehoben): **(0,1)**

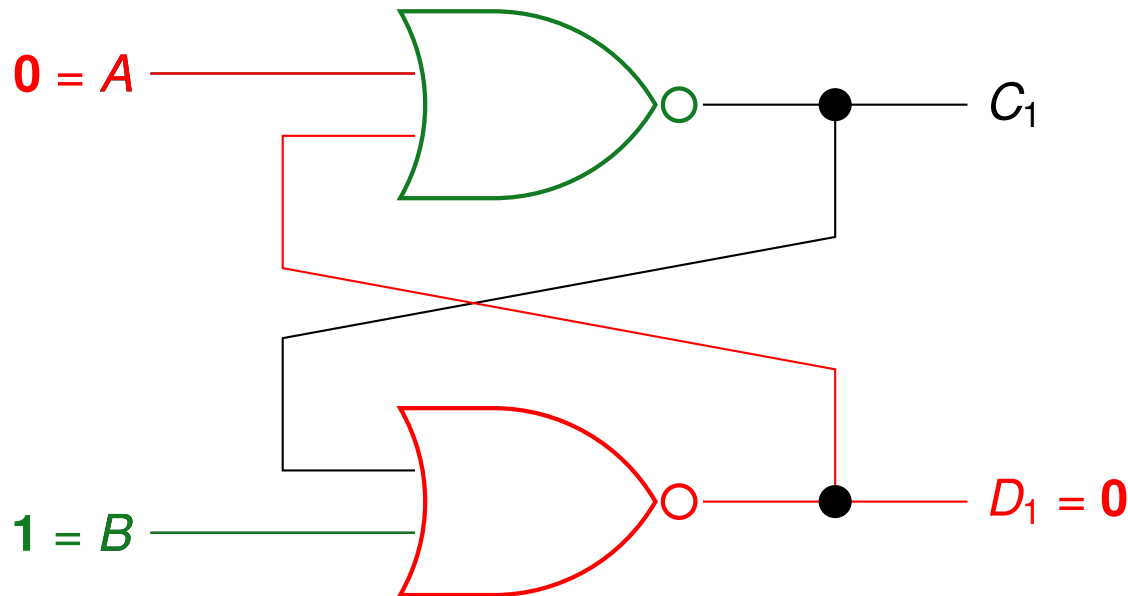


Hat ein NOR mindestens **eine** 1 im Eingang, so ist der Ausgang 0.

Hat ein NOR **keine** 1 im Eingang (sprich: nur Nullen), so ist der Ausgang 1.

Aufgabe 4 – Latches und Flipflops

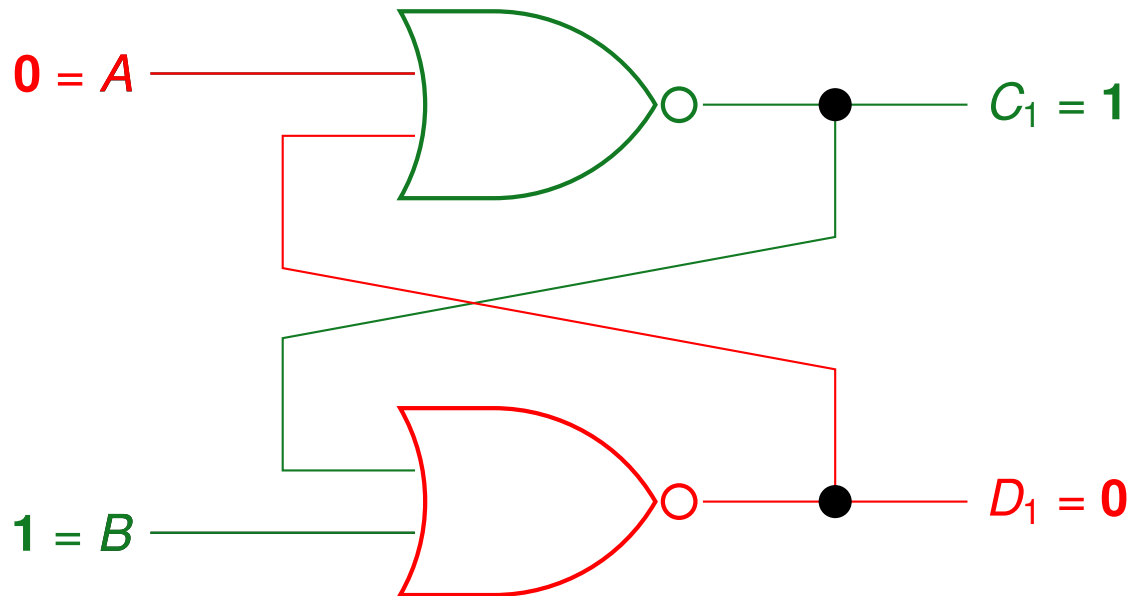
Anliegender Wert (**fett** hervorgehoben): **(0,1)**



Hat ein NOR mindestens **eine** 1 im Eingang, so ist der Ausgang 0.
 Hat ein NOR **keine** 1 im Eingang (sprich: nur Nullen), so ist der Ausgang 1.

Aufgabe 4 – Latches und Flipflops

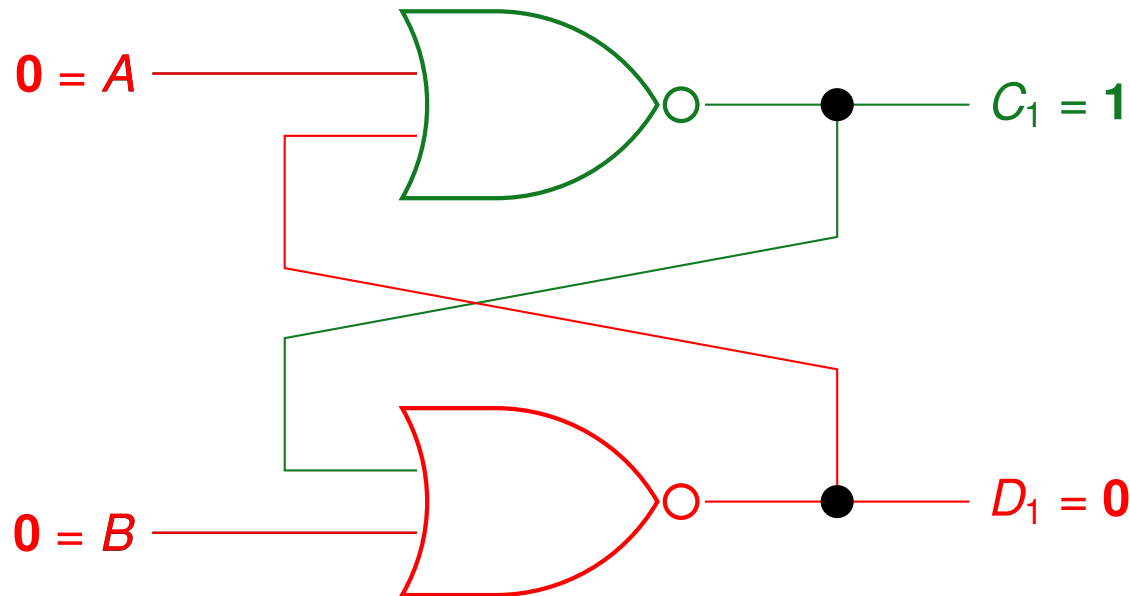
Anliegender Wert (**fett** hervorgehoben): **(0,1)**



Hat ein NOR mindestens **eine** 1 im Eingang, so ist der Ausgang 0.
 Hat ein NOR **keine** 1 im Eingang (sprich: nur Nullen), so ist der Ausgang 1.

Aufgabe 4 – Latches und Flipflops

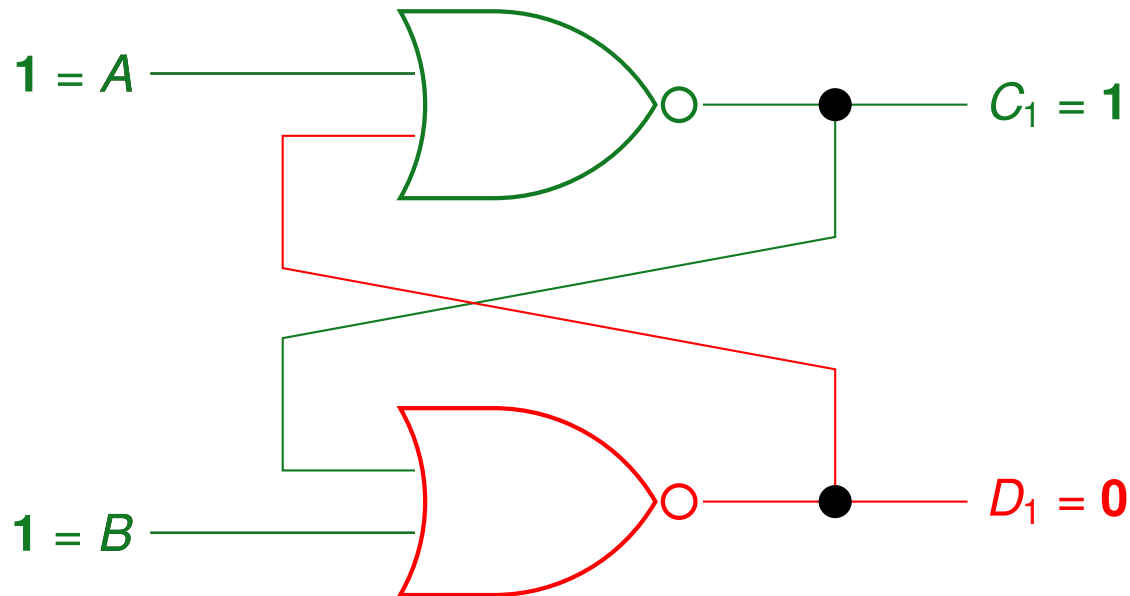
Anliegender Wert (**fett** hervorgehoben): $(0,1) \mapsto (0, 0)$



Hat ein NOR mindestens **eine** 1 im Eingang, so ist der Ausgang 0.
 Hat ein NOR **keine** 1 im Eingang (sprich: nur Nullen), so ist der Ausgang 1.

Aufgabe 4 – Latches und Flipflops

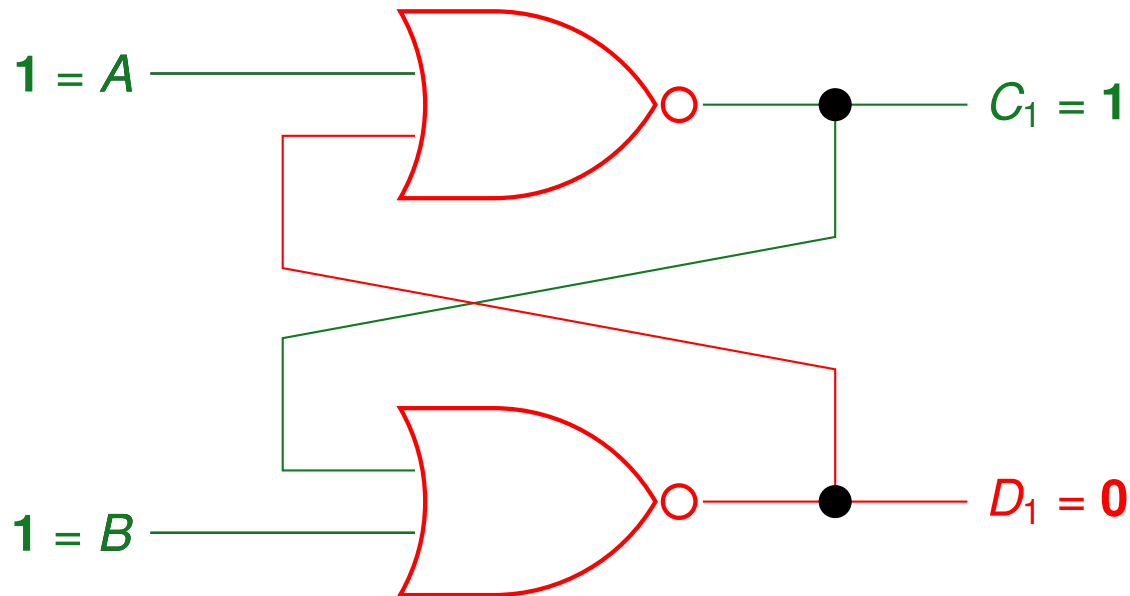
Anliegender Wert (**fett** hervorgehoben): $(0,1) \mapsto (0, 0) \mapsto (1, 1)$



Hat ein NOR mindestens **eine** 1 im Eingang, so ist der Ausgang 0.
 Hat ein NOR **keine** 1 im Eingang (sprich: nur Nullen), so ist der Ausgang 1.

Aufgabe 4 – Latches und Flipflops

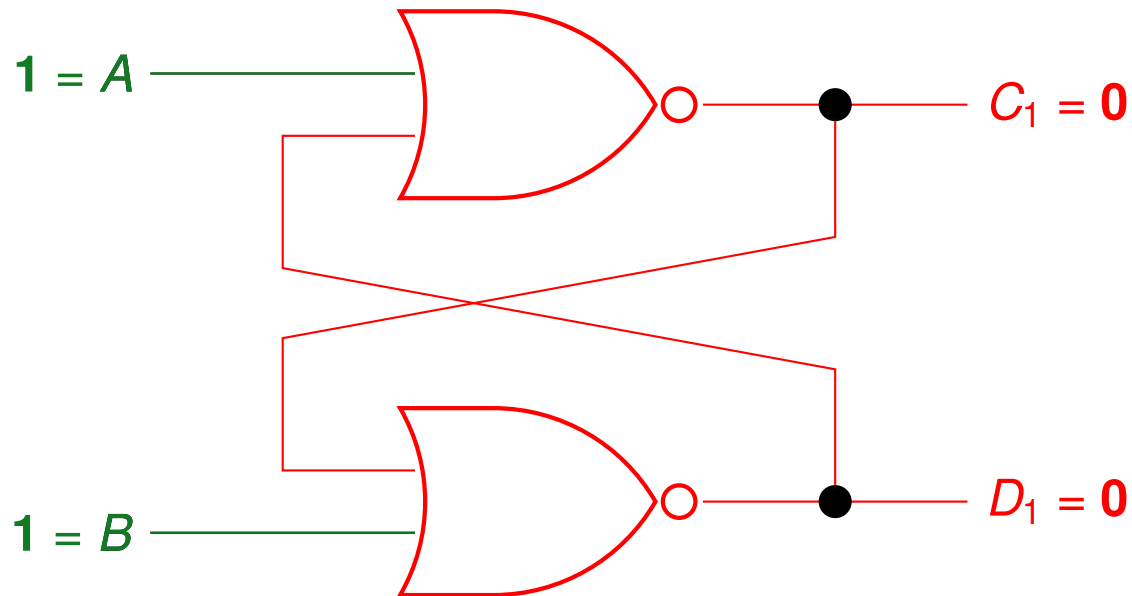
Anliegender Wert (**fett** hervorgehoben): $(0,1) \mapsto (0, 0) \mapsto (1, 1)$



Hat ein NOR mindestens **eine** 1 im Eingang, so ist der Ausgang 0.
 Hat ein NOR **keine** 1 im Eingang (sprich: nur Nullen), so ist der Ausgang 1.

Aufgabe 4 – Latches und Flipflops

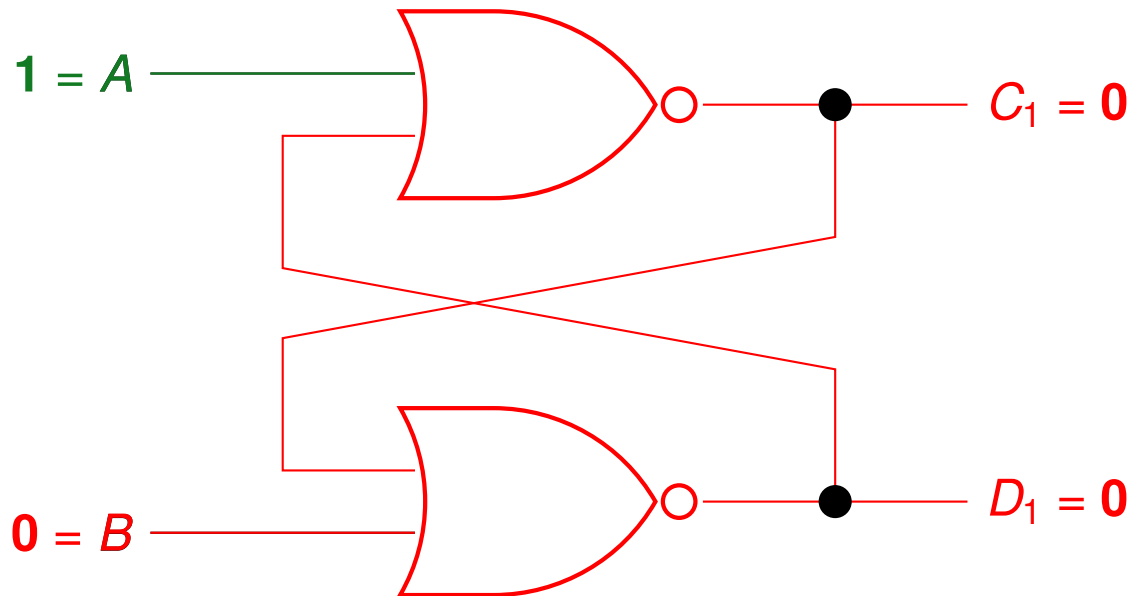
Anliegender Wert (**fett** hervorgehoben): $(0,1) \mapsto (0, 0) \mapsto (1, 1)$



Hat ein NOR mindestens **eine** 1 im Eingang, so ist der Ausgang 0.
 Hat ein NOR **keine** 1 im Eingang (sprich: nur Nullen), so ist der Ausgang 1.

Aufgabe 4 – Latches und Flipflops

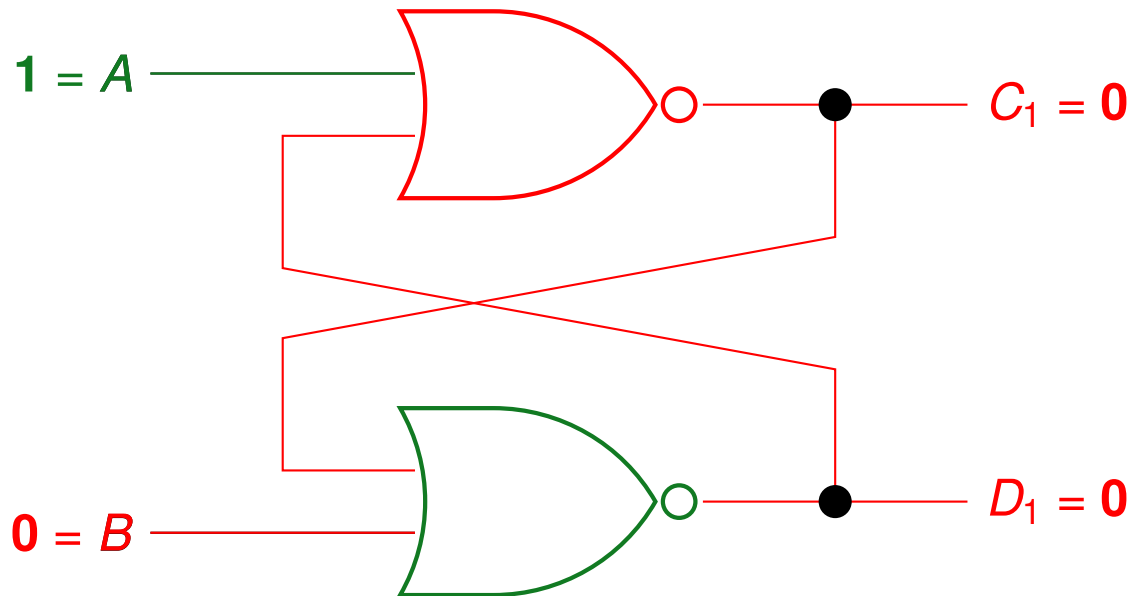
Anliegender Wert (**fett** hervorgehoben): ... $\mapsto (0, 0) \mapsto (1, 1) \mapsto (1, 0)$



Hat ein NOR mindestens **eine** 1 im Eingang, so ist der Ausgang 0.
 Hat ein NOR **keine** 1 im Eingang (sprich: nur Nullen), so ist der Ausgang 1.

Aufgabe 4 – Latches und Flipflops

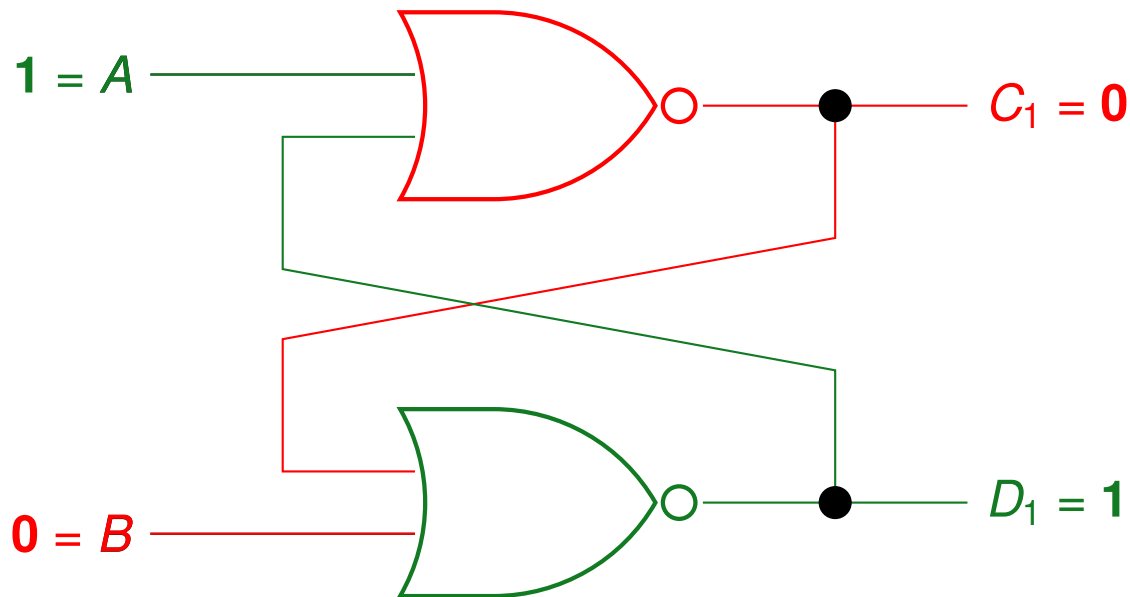
Anliegender Wert (**fett** hervorgehoben): ... $\mapsto (0, 0) \mapsto (1, 1) \mapsto (1, 0)$



Hat ein NOR mindestens **eine** 1 im Eingang, so ist der Ausgang 0.
 Hat ein NOR **keine** 1 im Eingang (sprich: nur Nullen), so ist der Ausgang 1.

Aufgabe 4 – Latches und Flipflops

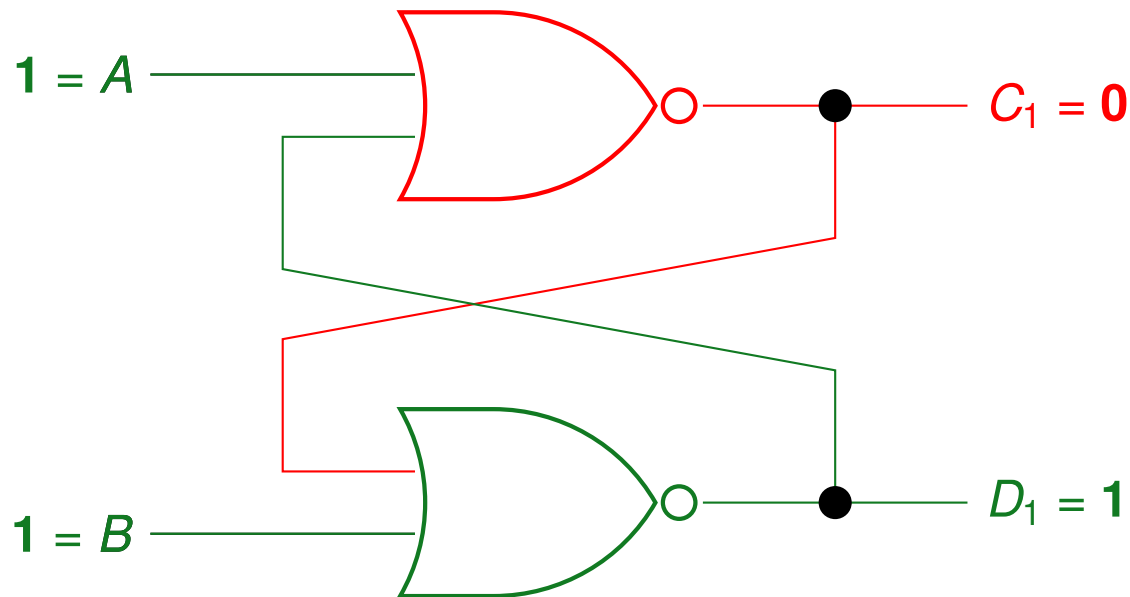
Anliegender Wert (**fett** hervorgehoben): ... $\mapsto (0, 0) \mapsto (1, 1) \mapsto (1, 0)$



Hat ein NOR mindestens **eine** 1 im Eingang, so ist der Ausgang 0.
 Hat ein NOR **keine** 1 im Eingang (sprich: nur Nullen), so ist der Ausgang 1.

Aufgabe 4 – Latches und Flipflops

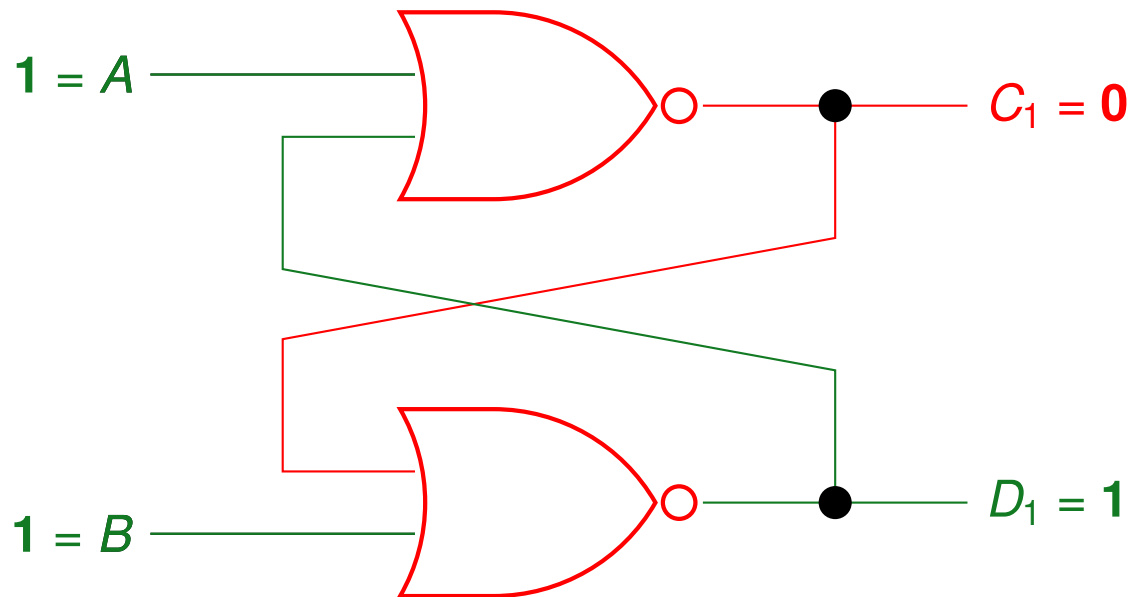
Anliegender Wert (**fett** hervorgehoben): ... $\mapsto (1, 1) \mapsto (1, 0) \mapsto (1, 1)$



Hat ein NOR mindestens **eine** 1 im Eingang, so ist der Ausgang 0.
 Hat ein NOR **keine** 1 im Eingang (sprich: nur Nullen), so ist der Ausgang 1.

Aufgabe 4 – Latches und Flipflops

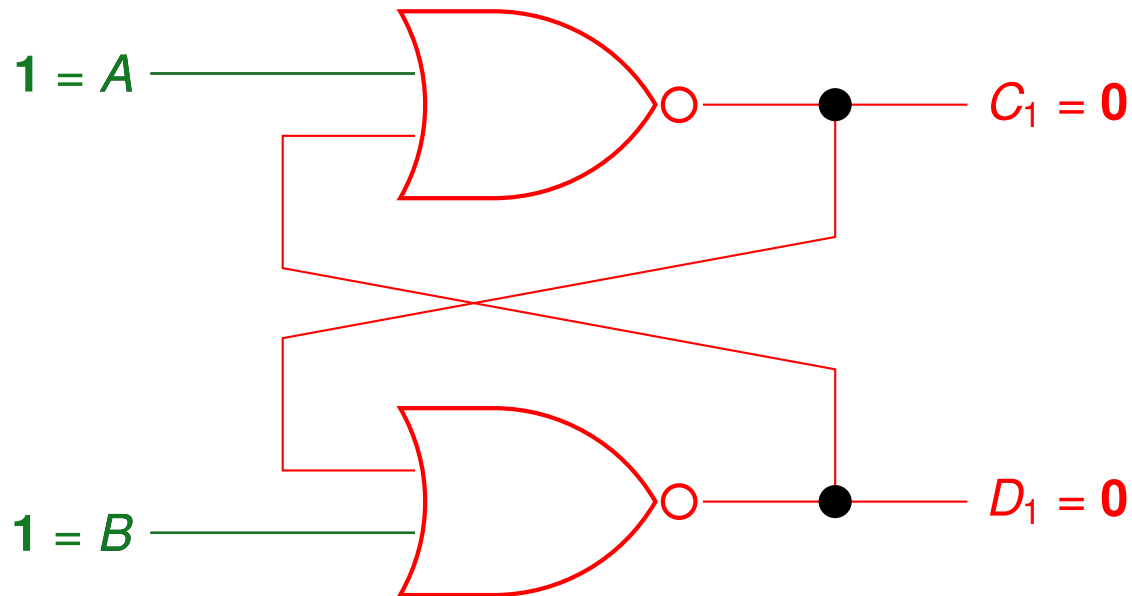
Anliegender Wert (**fett** hervorgehoben): ... $\mapsto (1, 1) \mapsto (1, 0) \mapsto (1, 1)$



Hat ein NOR mindestens **eine** 1 im Eingang, so ist der Ausgang 0.
 Hat ein NOR **keine** 1 im Eingang (sprich: nur Nullen), so ist der Ausgang 1.

Aufgabe 4 – Latches und Flipflops

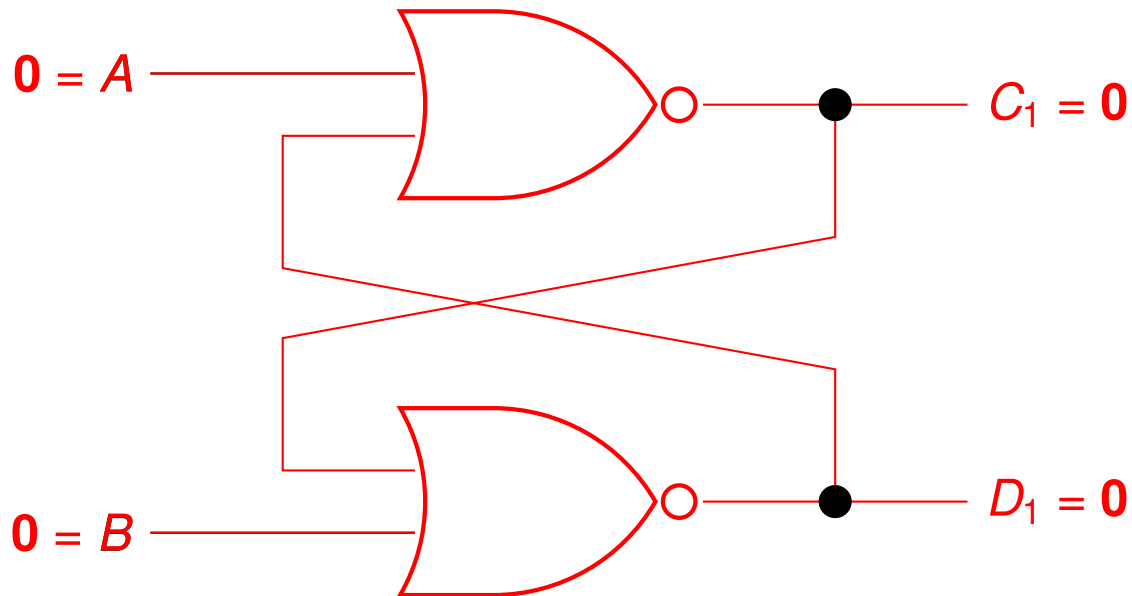
Anliegender Wert (**fett** hervorgehoben): ... $\mapsto (1, 1) \mapsto (1, 0) \mapsto (1, 1)$



Hat ein NOR mindestens **eine** 1 im Eingang, so ist der Ausgang 0.
 Hat ein NOR **keine** 1 im Eingang (sprich: nur Nullen), so ist der Ausgang 1.

Aufgabe 4 – Latches und Flipflops

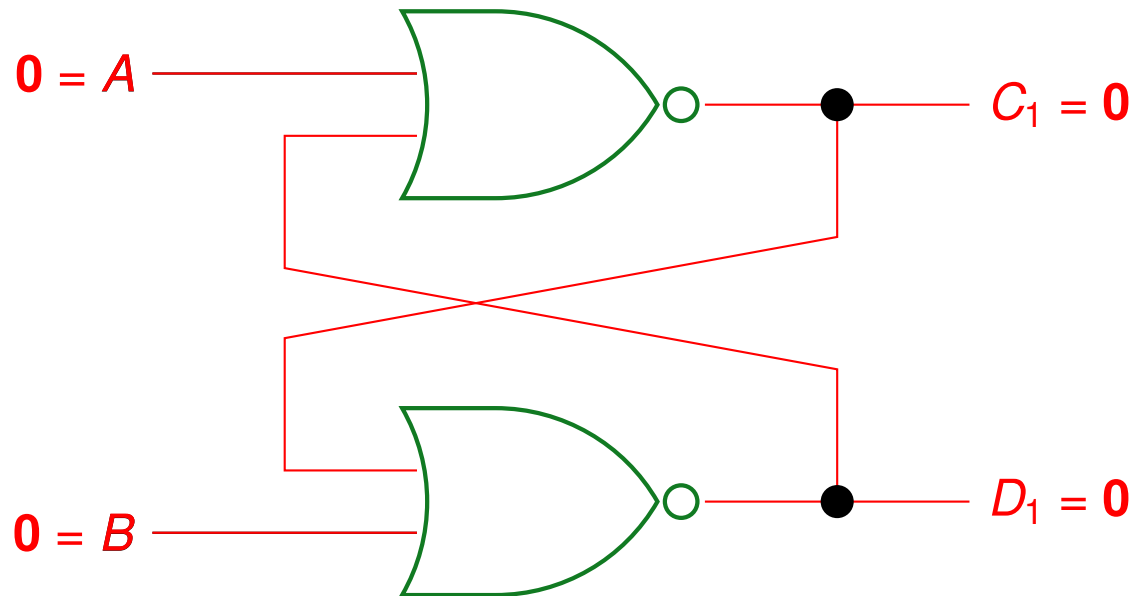
Anliegender Wert (**fett** hervorgehoben): ... $\mapsto (1, 0) \mapsto (1, 1) \mapsto (0, 0)$



Hat ein NOR mindestens **eine** 1 im Eingang, so ist der Ausgang 0.
 Hat ein NOR **keine** 1 im Eingang (sprich: nur Nullen), so ist der Ausgang 1.

Aufgabe 4 – Latches und Flipflops

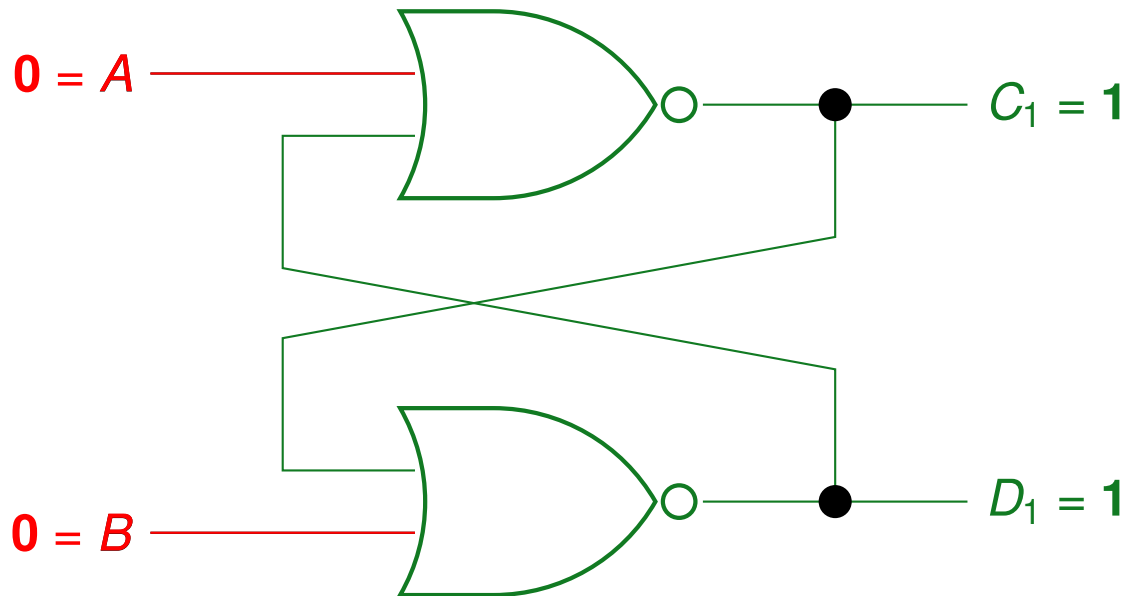
Anliegender Wert (**fett** hervorgehoben): ... $\mapsto (1, 0) \mapsto (1, 1) \mapsto (0, 0)$



Hat ein NOR mindestens **eine** 1 im Eingang, so ist der Ausgang 0.
 Hat ein NOR **keine** 1 im Eingang (sprich: nur Nullen), so ist der Ausgang 1.

Aufgabe 4 – Latches und Flipflops

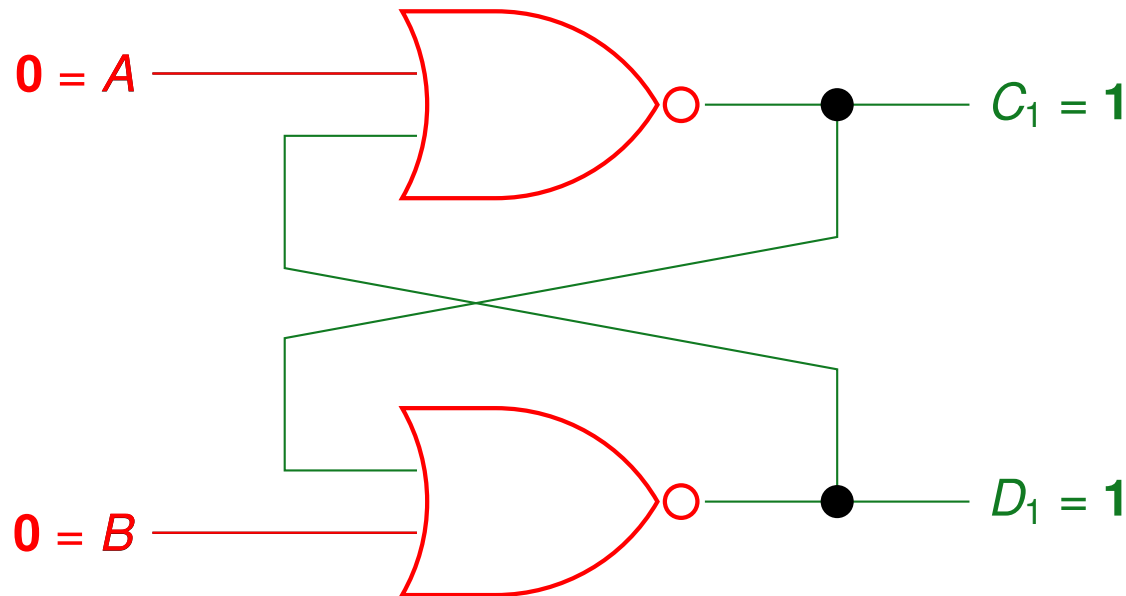
Anliegender Wert (**fett** hervorgehoben): ... $\mapsto (1, 0) \mapsto (1, 1) \mapsto (0, 0)$



Hat ein NOR mindestens **eine** 1 im Eingang, so ist der Ausgang 0.
 Hat ein NOR **keine** 1 im Eingang (sprich: nur Nullen), so ist der Ausgang 1.

Aufgabe 4 – Latches und Flipflops

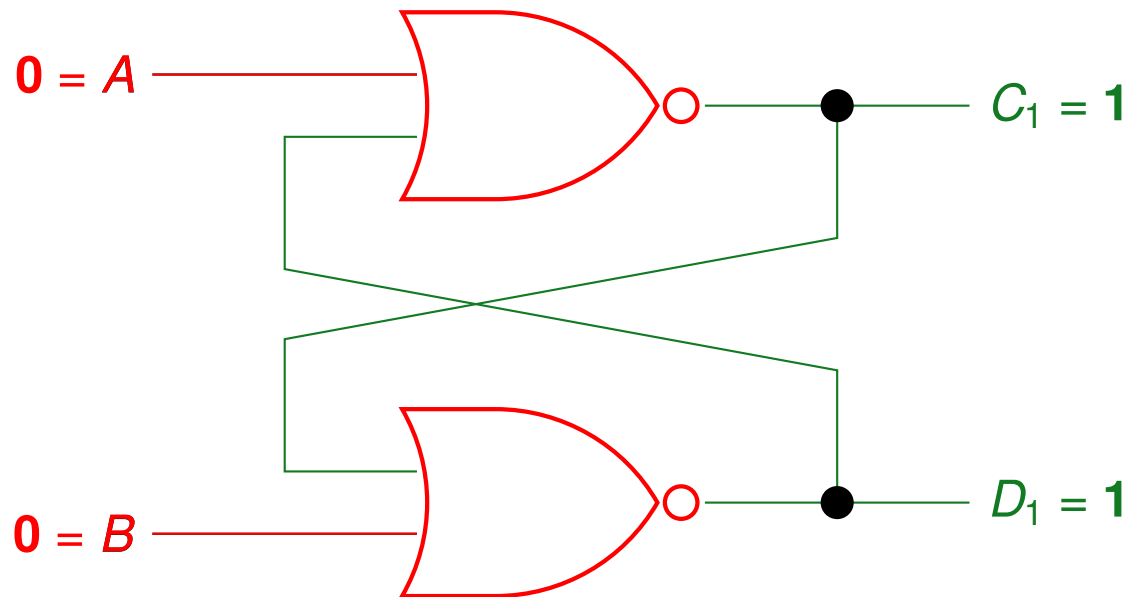
Anliegender Wert (**fett** hervorgehoben): ... $\mapsto (1, 0) \mapsto (1, 1) \mapsto (0, 0)$



Hat ein NOR mindestens **eine** 1 im Eingang, so ist der Ausgang 0.
 Hat ein NOR **keine** 1 im Eingang (sprich: nur Nullen), so ist der Ausgang 1.

Aufgabe 4 – Latches und Flipflops

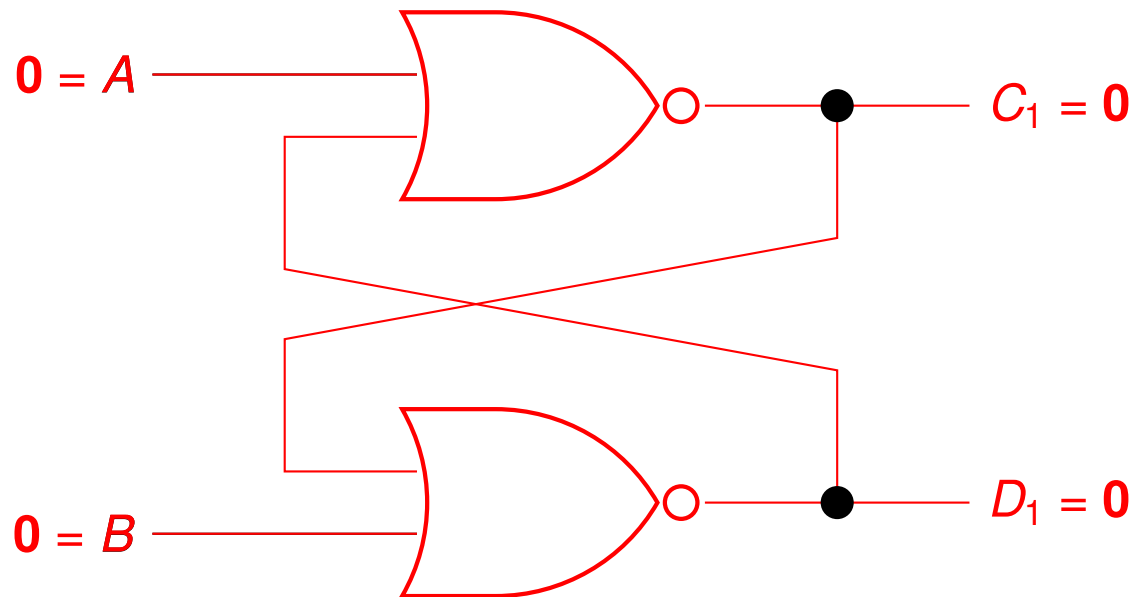
Anliegender Wert (**fett** hervorgehoben): ... $\mapsto (1, 0) \mapsto (1, 1) \mapsto (0, 0)$



Hat ein NOR mindestens **eine** 1 im Eingang, so ist der Ausgang 0.
 Hat ein NOR **keine** 1 im Eingang (sprich: nur Nullen), so ist der Ausgang 1.

Aufgabe 4 – Latches und Flipflops

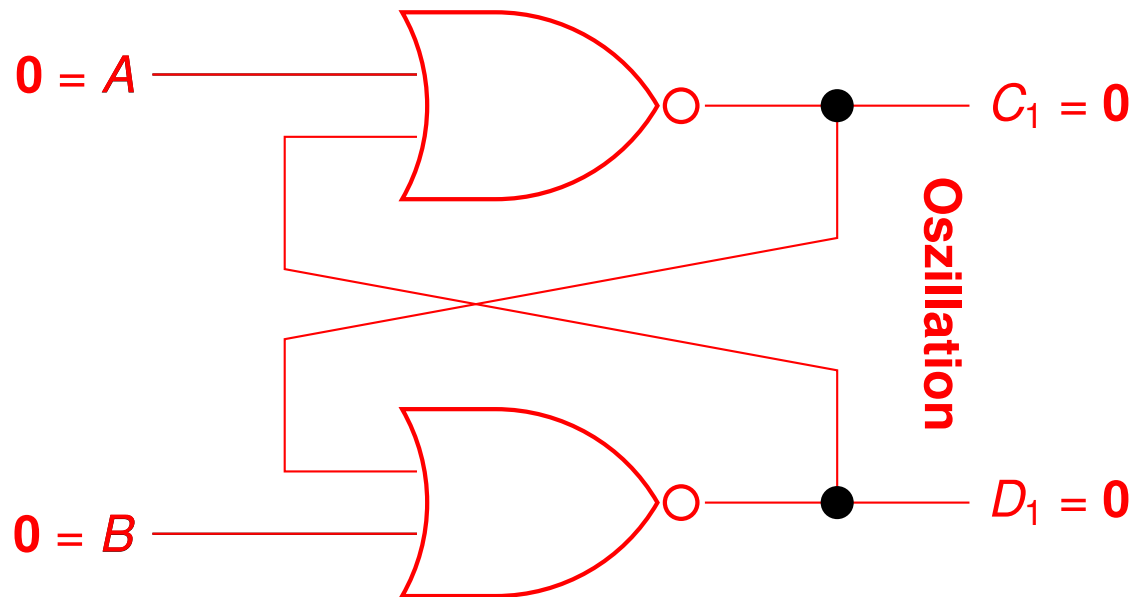
Anliegender Wert (**fett** hervorgehoben): ... $\mapsto (1, 0) \mapsto (1, 1) \mapsto (0, 0)$



Hat ein NOR mindestens **eine** 1 im Eingang, so ist der Ausgang 0.
 Hat ein NOR **keine** 1 im Eingang (sprich: nur Nullen), so ist der Ausgang 1.

Aufgabe 4 – Latches und Flipflops

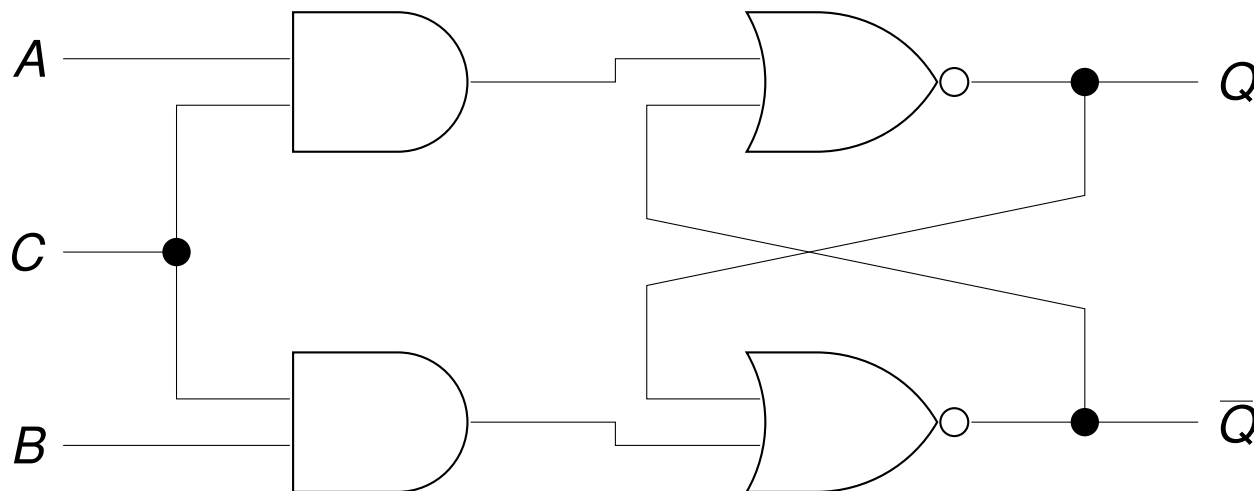
Anliegender Wert (**fett** hervorgehoben): ... $\mapsto (1, 0) \mapsto (1, 1) \mapsto (0, 0)$



Hat ein NOR mindestens **eine** 1 im Eingang, so ist der Ausgang 0.
 Hat ein NOR **keine** 1 im Eingang (sprich: nur Nullen), so ist der Ausgang 1.

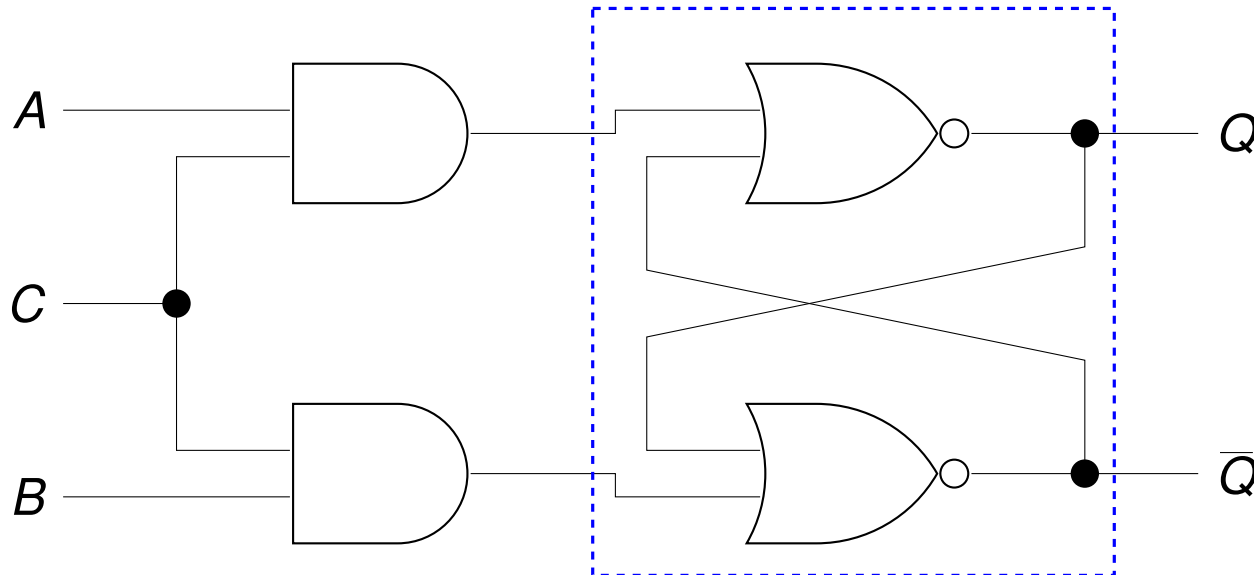
Aufgabe 4 – Latches und Flipflops

b) Sei C ein Taktsignal. Wie bezeichnet man dann das hier abgebildete Speicherelement?



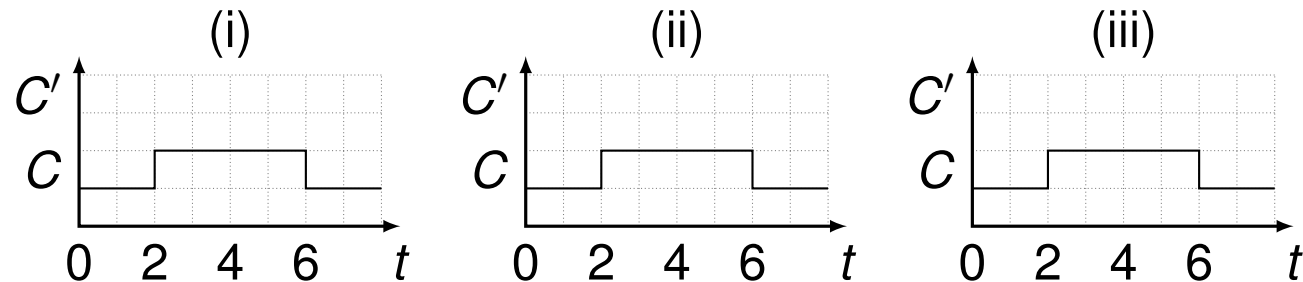
Aufgabe 4 – Latches und Flipflops

b) Sei C ein Taktsignal. Wie bezeichnet man dann das hier abgebildete Speicherelement?



Aufgabe 4 – Latches und Flipflops

b) Dieses soll so erweitert werden, dass es (i) nur bei der steigenden, (ii) der fallenden und (iii) bei jeder Flanke von C auf die Eingänge A und B reagiert. Geben Sie jeweils das Schaltnetz der Flankenerkennung $C \mapsto C'$ an und vervollständigen Sie die folgenden Wellenformdiagramme:



Aufgabe 4 – Latches und Flipflops

- c) Erweitern Sie nun die Schaltung aus Teilaufgabe b) dahingehend, dass keine undefinierten Zustände, wie sie in Aufgabe a) der Fall waren, mehr auftreten.