

B1

"Wahr" ~ "1" ~ "high" ~ "H" ~ "spannig"

Tatsächlich basiert die ganze Logik und Arithmetik in einem Computer auf unseren wenigen logischen Grundfunktionen. Diese müssen aber hardware-mäßig realisiert werden.

Diese Logik-Schaltungen liegen auf der Grenze zwischen Mathematik und Physik oder zwischen der Logik und der Elektronik.

Die verwendeten Schaltensymbole stellen genauso eine elektronische Komponente wie eine logische Funktion dar als auf ein Bauteil, das man kaufen kann!

Mit diesen 'Logikgattern' kann sich gleichzeitig komplexe logische Abhängigkeiten darstellen und das Verdrahtungsschema von elektronischen Komponenten aufzeichnen.

Die Inputs (Wahrheitswerte, Daten) werden elektrisch durch Spannungen realisiert (~~und nicht eher~~ ^{statt z.B.} mechanisch durch irgendwelche Hebelstellungen ...). Sie müssen ^{sich} mit der folgenden äquivalenten Darstellung befassen:

Logik-Schreibweise		wahr		falsch
Digitale Schreibweise		1		0
Elektronik-Schreibweise	high	H		L low
Spannungsspekt *		$< 0,8 \text{ V}$ (max $< 0,1 \text{ V}$)		$> 2,8 \text{ V}$ (max $> 4,8 \text{ V}$)

*) gilt für 5 Volt - TTL-Bausteine, eine schon etwas belastete, aber immer noch erhältliche Bausteinserie.

Die Input-Leitungen sind immer links, der Output ist rechts angeordnet. Jeweill sollen in den Schaltungen die Informationen / Signale vom irgend möglich von links nach rechts fließen. Die folgende Tabelle gibt die Logikgatter mit den modernen Ekt-Symbolen. (In der USA werden immer noch andere, weniger eingängige Symbole verwendet.) Sie mache 2 einfache Beispiele.

~~B3~~

Tabelle der Logik-Gatter


Die Kopie stellt alle Logik-Glieder zusammen, die wir verwenden.

Links : Der Name der Logik Funktion

Mitte : Das Schaltzeichen (moderne EU-Variante)

Rechts : Funktionsformel , Wiederholung der Definition

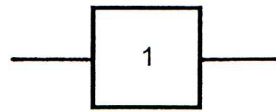
Die Informations flüsse immer von links nach rechts durch diese Schaltglieder . Die Anschlüsse links sind also Input-Leitungen , der Anschluss rechts ist die Output-Leitung .

Ein  vor dem Output bedeutet die Negation des ganzen Symbols (vgl. OR / NOR , AND / NAND , XOR / EXOR , 1 / NOT)

Im nächsten Abschnitt (B4) lernen wir , Schaltungen aus mehreren solchen Gliedern zu analysieren .

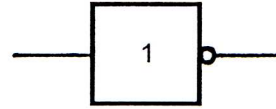
Logikbausteine

Verstärker, Treiber



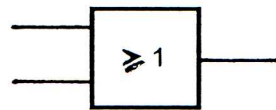
In	Out
1	1
0	0

Inverter, NOT



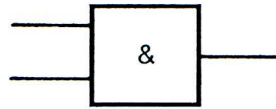
In	Out
1	0
0	1

OR



In	Out
1 1	1
1 0	1
0 1	1
0 0	0

AND



In	Out
1 1	1
1 0	0
0 1	0
0 0	0

NOR



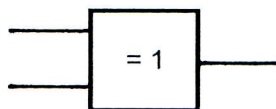
In	Out
1 1	0
1 0	0
0 1	0
0 0	1

NAND



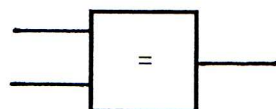
In	Out
1 1	0
1 0	1
0 1	1
0 0	1

XOR

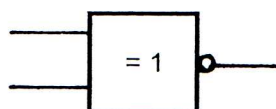


In	Out
1 1	0
1 0	1
0 1	1
0 0	0

EQUI

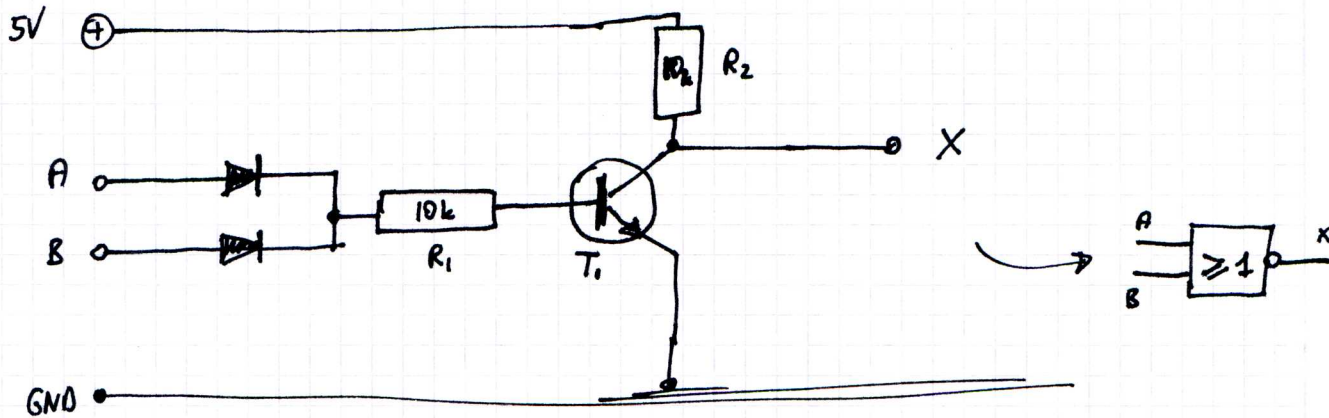


oder



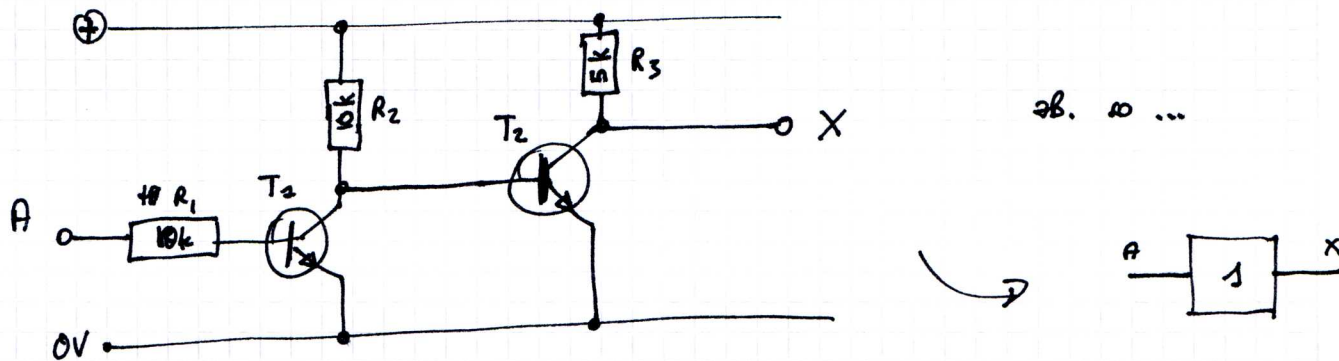
In	Out
1 1	1
1 0	0
0 1	0
0 0	1

Bauw ist aber noch ein simple NOR-Glied:



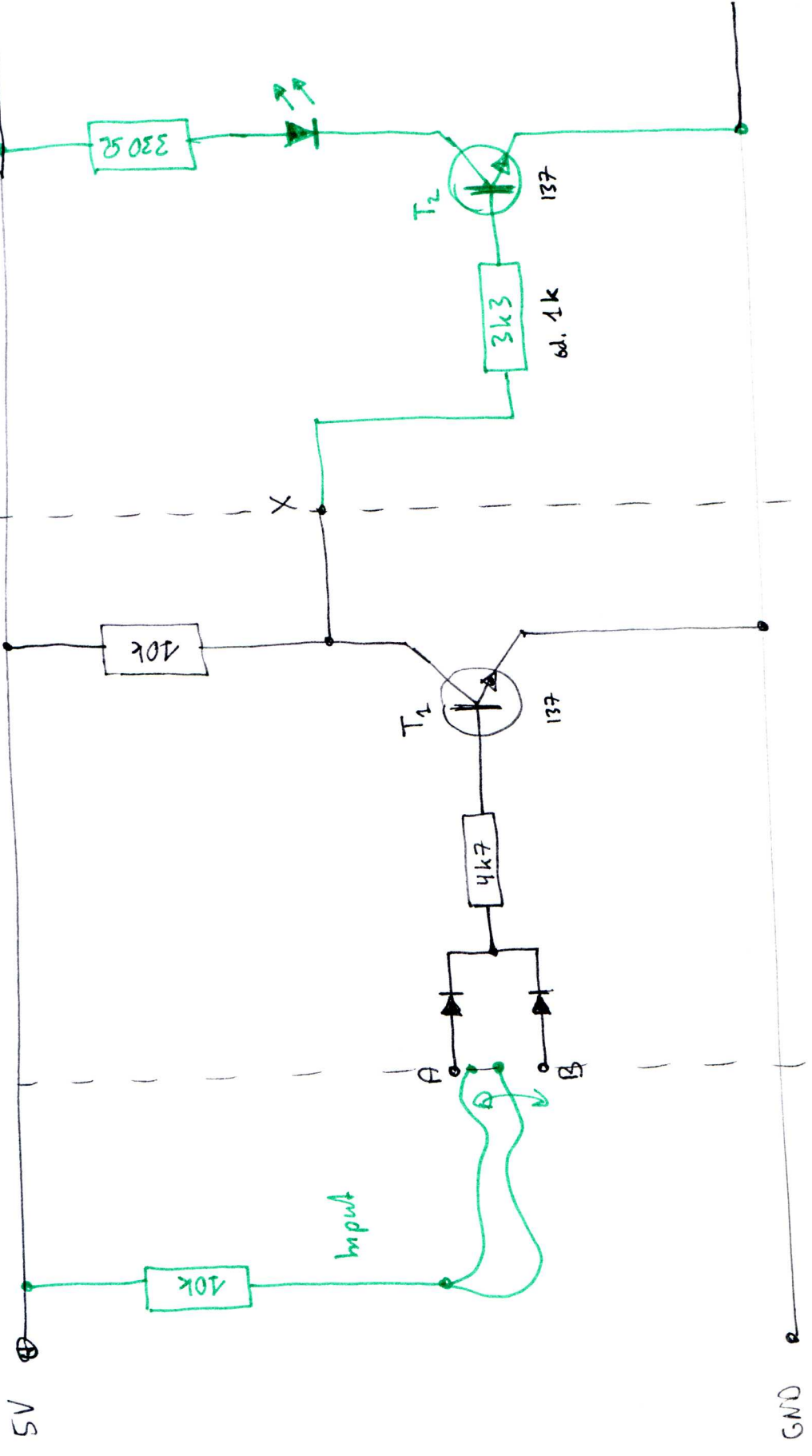
- Ist einer der beiden (oder beide!) Eingänge high, so wird T_1 durchgeschaltet und X geht auf low, da der Widerstand von T_1 dann viel kleiner ist als $R_2 = 10k\Omega$.
- Sind beide Eingänge low, so sperrt T_1 und X ist high.
- Die beide Dioden dienen dem, dass A high und B low konfliktfrei möglich sind.

Wie kann man jetzt z.B. einen Verstärker bauen??



Alle Bauteile bestätigen nicht die Input- und Output-Füßchen von GND und V_C für die Stromversorgung!

$$X = (A \text{ NOR } B)$$



X high \Rightarrow LED on

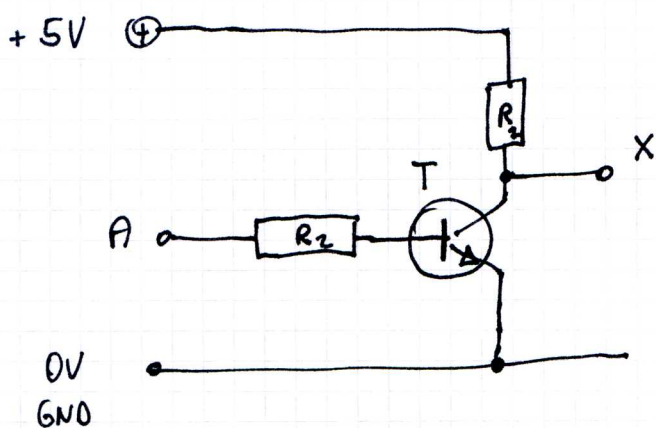
Done ✓ 25.6.2010

ⓑ3 RTL, TTL, CMOS, HCT usw am Beispiel des Inverters

a) RTL - Technologie (~~im 1960er-Jahre~~) ~~der 50er-Jahre~~ *späte 50er-Jahre*

RTL ~ Resistor-Transistor-Logik

Die Schaltungen werden aus Widerständen und (meist npn-) Transistoren aufgebaut:



$$R_1 = R_2 = 10 \text{ k}\Omega$$

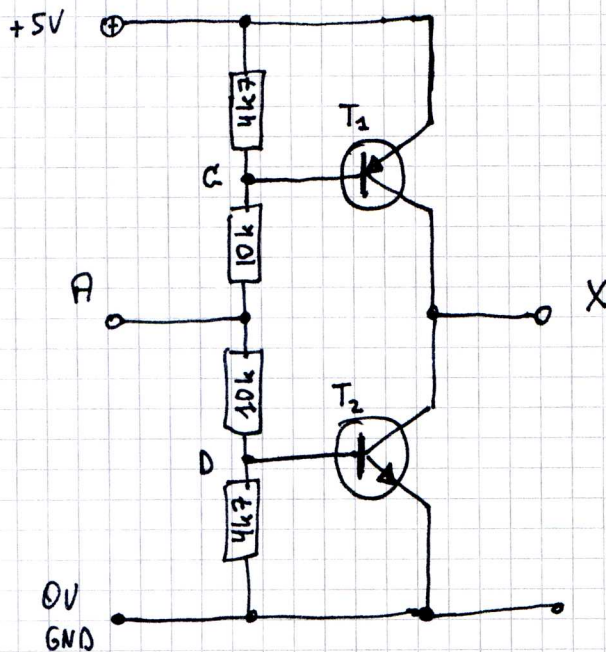
Bestreibung in B2

Nachteil: Im Falle von $A = \text{high}$ fließt durch ein ~~gering~~ beträchtlicher Strom ($5\text{V} / 10\text{k}\Omega = 0,5\text{mA}$) durch T \rightarrow Wärme!

b) TTL - Technologie (~~im 1970er-Jahre~~) ~~der 70er-Jahre~~ *ab 1961*

In der TTL-Technologie (Transistor-Transistor-Logik) wird der Widerstand R_2 durch ein ~~eine~~ ~~zweiten~~ (oder mehrere) gesteuerten Transistor ersetzt.

Sind für eine Versorgung mit 5V angelegt



$A \text{ high} \Rightarrow C \text{ high} \Rightarrow T_2 \text{ gesperrt}$
 $\Rightarrow R(T_2) \sim M\Omega$

$A \text{ high} \Rightarrow D \sim 2V \Rightarrow T_2 \text{ leitet}$
 $\Rightarrow R(T_2) < 10\Omega$

also insgesamt $X = \text{low}!$

$A \text{ low} \Rightarrow T_2 \text{ gesperrt} \& T_2 \text{ leitet}$
 $\Rightarrow X = \text{high}$

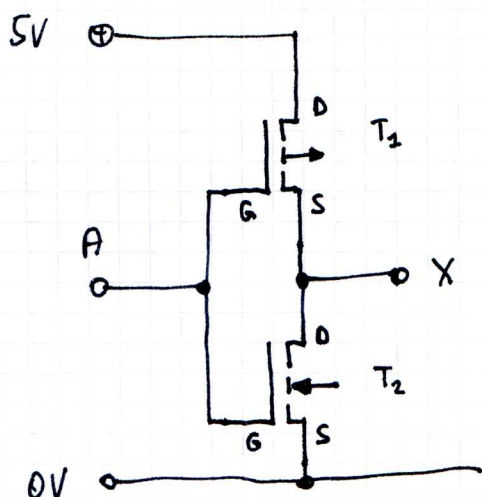
Es gibt viele Varianten an TTL-Invertoren, sie können ja mal mit "wiki TTL invertor" googeln ...

c) CMOS-Technologie ~~des 30er-Jahr~~ ~~ab 1968~~ ~~ab 1968~~ ab 1968

Nach wieher konnte der Stromverbrauch mit der Halbleitertechnologie gesenkt werden durch die Verwendung von MOS-Transistoren. Bei ihnen fließt nur ein winziger Strom beim Ein- oder Ausschalten, bis das elektrische Feld auf- oder abgebaut ist, welches den 'Kanal' leitend oder wieder gesperrt macht.

Das C in CMOS bedeutet 'komplementär', es werden meist 2 Typen verwendet, die sowohl gesperrt sind. Der PMOS-Transistor wird durch eine negative Spannung von D zu G leitet, der NMOS durch eine positive Spannung von G zu S.

$V_{DD} = \text{Anspannung} \approx 4-15V$ Versorgungsspannung betriebe werden



T_1 PMOS, normal gesperrt, leitet falls ~~U_{GS}~~ $U_{GS} > 0.8V$

T_2 NMOS, normal gesperrt, leitet falls $U_{GS} > 0.8V$

Die Transistoren schalten extrem schnell (mehrere GHz möglich!)

$$\begin{aligned} R(\text{gesperrt}) &= \text{einige M}\Omega \\ R(\text{leitend}) &< 1 \Omega \quad !! \end{aligned}$$

} nahezu ideale Schalter!

Problem: $A = 2.5V \rightarrow$ beide Schalter durch \rightarrow Kurzschluss zwischen 5V und 0V !!

Es braucht also noch eine Zusatzschaltung welche verhindert, dass A während ein neuer Wert $Z_{\text{et}} > 0.8V$ und $< 4.2V$ ist !!

d) HCT - Bausteine (frühe 80er-Jahre)

Diese lösen genau das Problem der CMOS-Schaltkreise, indem sie am Eingang eine Schaltung mit periodischen Transistoren haben, die dafür sorgt, dass der Wert von U_{GS} auf 10V (z.B.) sehr schnell erfolgt ("Schicht-Transistor"). *Zudem Abgrenzung \leftarrow TTL-Beleg.*

Die Bausteine der Serie 74 HCT sind daher recht beliebt. Sie vermeiden ein weiteres Schwachpunkt von CMOS-Schaltkreisen: Sie sind viel weniger empfindlich gegen elektrostatische Entladungen.

e) Bi CMOS - Technologie

Nun kann man auch noch am Ausgang von CMOS-Schaltern
 ein Verstärker mit gewöhnlichen Transistoren anbringen. Damit
 können dann diese Logik-Glieder direkt Thyristoren oder
 Triacs (das Schalte für größere Leistungen) ansteuern.

wiki 'CMOS Inverter'

http://tams-www.informatik.uni-hamburg.de/applets/cmos/cmos_dt.html

In unserer 2. Serie von Kursvorlesungen werden wir sehen, wie man
 alle unsere Logikgatter in RTL, TTL und CMOS-Technik
 realisieren kann.

84

Analyse von einfachen Logikschaltungen

Egal ob die Aufgabe in der Sprache der Logik oder ~~als~~ als Schaltplan der Digitaltechnik gestellt wird: Sie kann nun komplexe (d.h. zusammen gesetzte ...) Situationen analysieren?

Nur in einfachen Fällen kann man mithilfe der Sätze der Strukturansätze:

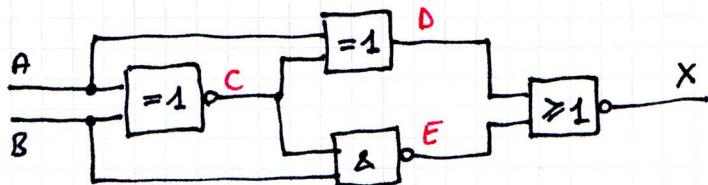
Bsp. 1 $X = \text{NOT}(A \text{ NAND } B) = \text{NOT}(\text{NOT}(A \text{ AND } B)) = A \text{ AND } B$



→ NAND und NOR können auch als Inverter plant werden!

Was geschieht aber hier:

Bsp. 2



Hier hilft eine systematische Analyse weiter. Früher Hilfspositionen C, D, E ein und analysiere sie links und rechts fastfortschritt:

A	B	C	D	E	X
1	1	1	0	0	1
1	0	0	1	1	0
0	1	0	0	1	0
0	0	1	1	1	0

A EQUI B
 A XOR C
 B NAND C
 E NOR D

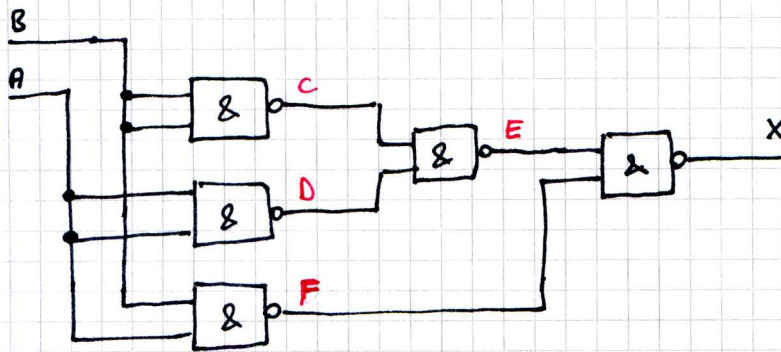
⇒ $X = A \text{ AND } B$!

Das Beispiel ist etwas absurd - dennoch könnte man ...



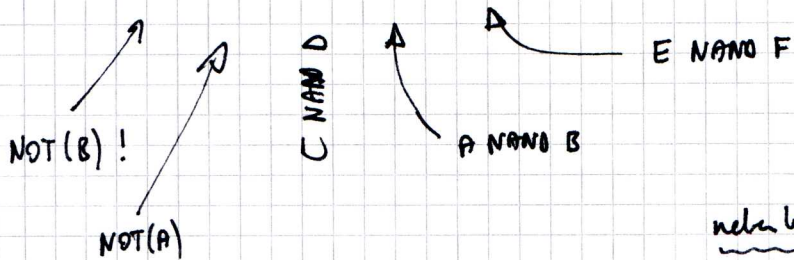
erhalten...

Bsp. 3 Eine kleine NAND-Logik :



A	B	C	D	E	F	X
1	1	0	0	1	0	1
1	0	1	0	1	1	0
0	1	0	1	1	1	0
0	0	1	1	0	1	1

$\Rightarrow X = A \text{ EQUI } B$



nebenbei: $E = A \text{ OR } B$

\Rightarrow Weitere Beispiele als Übungsaufgaben !