Rechnen-
und
Systemtechnik

Skript und Unterrichtsmitschrift

Christian Klisch

April 2002

Inhaltsverzeichnis

1 Grundbausteine der Digitaltechnik

Die Grundbausteine der Digitaltechnik haben in der Regel einen oder mehrere Eingänge und einen Ausgang. Um den Zusammenhang zwischen den Ein- und Ausgangssignalen zu erkennen, stellt man sogenannte Wahrheitstabellen auf.

Die Funktion der Schaltung beschreibt man durch eine Schaltung der Schaltalgebra. Die Signalzustände an den Ein- und Ausgängen kennzeichnet man in der Digitaltechnik mit den Binärzeichen I und 0 oder H (high) und L (low).
Diesen Binärzeichen sind Spannungen zugeordnet.

Beispiel:
H -> 2V ... 5V
L -> 0V ... 0,8V

1.1 UND-Verknüpfungen (Konjunkion)

Wahrheitstabellen:

$X=A\wedge B$

Fall	B	A	X
	0	0	0
	0	1	0
	1	0	0
	1	1	1

$X=A\wedge B\wedge C$

Fall	C	B	A	X
	0	0	0	0
	0	0	1	0
	0	1	0	0
	0	1	1	0
	1	0	0	0
	1	0	1	0
	1	1	0	0
	1	1	1	1

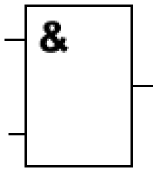
Der Ausgang hat immer nur dann ein High-Signal, wenn <u>alle</u> Eingänge gleichzeitig auch ein High-Signal haben

$X=A\wedge B\wedge C\wedge D$

Fall	D	C	B	A	X
	0	0	0	0	0
	0	0	0	1	0
	0	0	1	0	0
	0	0	1	1	0
	0	1	0	0	0
	0	1	0	1	0
	0	1	1	0	0
	0	1	1	1	0
	1	0	0	0	0
	1	0	0	1	0

	1	0	1	0	0
	1	0	1	1	0
	1	1	0	0	0
	1	1	0	1	0
	1	1	1	0	0
	1	1	1	1	1

Schaltzeichen:



1.2 ODER-Baustein

Technische Beispiele:
Motorrad: Zündung funktioniert nur, wenn der Ständer oben, Leerlauf/Kupplung drin ist
Elektrische Türöffneranlage im Mehrfamilienhaus

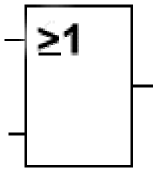
Der Ausgang hat High-Signal, wenn mindestens ein Eingang High-Signal hat.

Mathematische Gleichung:

$X=A\vee B$
$X=A\vee B\vee C\vee D...$

Fall	B	A	X
	0	0	0
	0	1	1
	1	0	1
	1	1	1

Schaltzeichen:



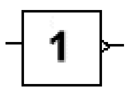
1.3 NICHT-Verknüpfung (Negation)

Fall	A	X
1	0	1
2	1	0

Mathematische Formel:

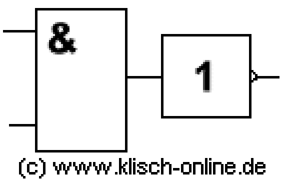
$X=\overline{A}$

Schaltzeichen:



2 Zusammengesetzte Glieder

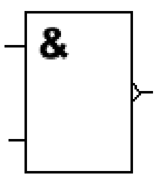
2.1 NAND -Baustein



NOT AND

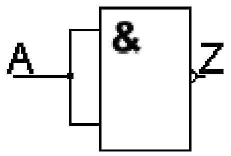
Fall	B	A	x	y
1	0	0	0	I
2	0	I	0	I
3	I	0	0	I
4	I	I	I	0

Schaltzeichen:



Am Ausgang des NAND-Bausteins ist immer dann High-Signal, wenn alle Eingänge auf Low geschaltet sind.

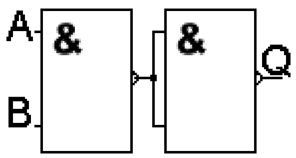
Negation aus NAND



(c) www.klisch-online.de

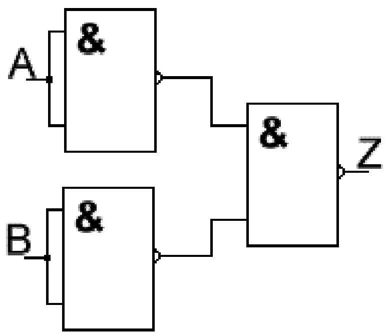
Fall	A	Z
1	0	I
2	I	0

UND aus NAND



Fall	B	A	Q
1	0	0	0
2	0	I	0
3	I	0	0
4	I	I	I

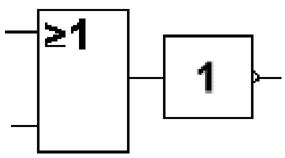
ODER aus NAND



Fall	B	A	Z
1	0	0	0
2	0	I	I
3	I	0	I
4	I	I	I

(c) www.klisch-online.de

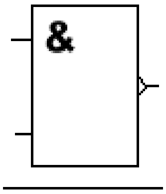
2.2 NOR-Bausteine



(c) www.klisch-online.de

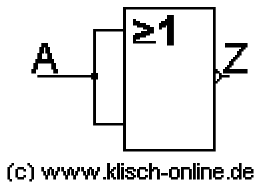
Fall	B	A	Y
1	0	0	I
2	0	I	0
3	I	0	0
4	I	I	0

Schaltzeichen:



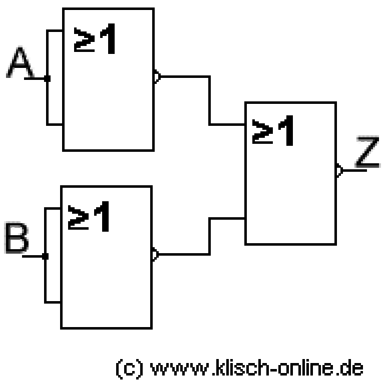
Am Ausgang eines NOR-Bausteins ist nur dann High-Signal, wenn alle Signale Low geschaltet sind

Negation aus NOR



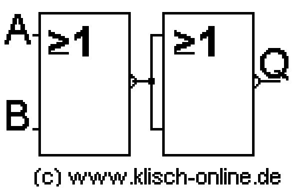
Fall	A	Z
1	0	I
2	I	0

UND aus NOR



Fall	B	A	Z
1	0	0	0
2	0	I	0
3	I	0	0
4	I	I	I

ODER aus NOR



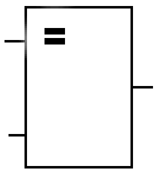
Fall	B	A	Q
1	0	0	0
2	0	I	I
3	I	0	I
4	I	I	I

2.3 Äquivalenz-Baustein *(Gleichwertig)*

Fall	B	A	x
1	0	0	I
2	0	I	0
3	I	0	0
4	I	I	I

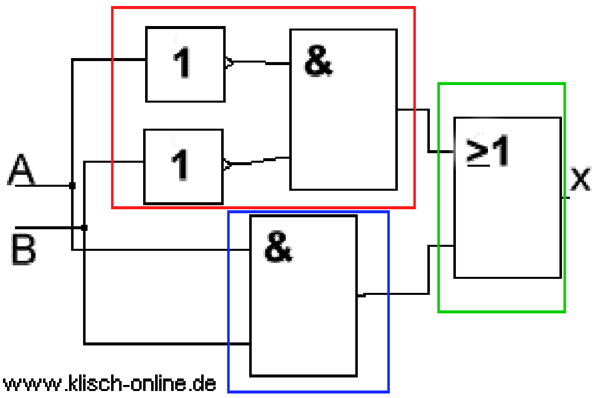
Ein Äquivalenz-Baustein hat immer dann am Ausgang ein High-Signal, wenn beide Eingänge ein gleiches Signal haben.

Schaltzeichen:



$x = A \wedge B \vee \overline{A} \wedge \overline{B}$

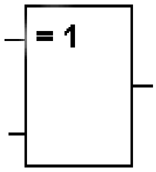
$$x = A \wedge B \vee \overline{A} \wedge \overline{B}$$



2.4 Antivalenz (XOR)

Fall	B	A	X
1	0	0	0
2	0	I	I
3	I	0	I
4	I	I	0

Schaltzeichen:



$x = A \wedge \overline{B} \vee \overline{A} \wedge B$

Beim Antivalenz-Baustein ist dann am Ausgang ein High-Signal, wenn die Signale am Eingang unterschiedliche sind.

3 Binärcodes und Zahlensysteme

3.1 Dezimales Zahlensystem

Basis 10 => 10 Ziffern

$10^0 = 1$
$10^1 = 10$
$10^2 = 100$
$10^3 = 1000$

3.2 Duales Zahlensystem

Im Zeitraum 1670 - 1703 wurde von dem Spanier Juan Caramuel und dem deutschen Gottfried Leibnitz das Dualsystem entwickelt.
Weil das Dualsystem nur 2 Ziffern benötigt, die durch die Spannungen wie z.B.. 0V und +5V verschlüsselt werden können, wurde das Dualsystem zur Grundlage moderner Computer.

3.2.1 Aufbau des Dualen Zahlensystems

2^6	2^5	2^4	2^3	2^2	2^1	2^0	
64	32	16	8	4	2	1	
			0	I	I	I	= 7
			0×2^3	$+ 1 \times 2^2$	$+ 1 \times 2^1$	$+ 1 \times 2^0$	= 7

3.2.4 Dualzahlen mit Kommastellen

Es kann sein, dass eine Dezimalzahl mit Kommastellen sich nicht ohne Rest in eine Dualzahl mit Kommastellen umwandeln lässt. Man muss dann entscheiden, auf wie viel Stellen nach dem Komma man die Dualzahl berechnen will.

3.2.5 Addition von Dualzahlen

Regeln:
0 + 0 = 0
0 + I = I
I + I = 0 + Übertrag
I + I + I = I + Übertrag

Beispiel:

```
  I I I I
+ 0 I I I
I I I I
I 0 I I 0
```

3.2.6 Subtraktion von Dualzahlen

Subtraktion durch Addition des Zweier-Komplements

Regeln:

1. Die Abzuziehende Zahl (Subtrahend) auf volle Stellenzahl durch vorsetzen von Nullen erweitern
2. Das Einer-Komplement bilden, das heißt die abzuziehende Zahl zu invertieren
3. Das Zweier-Komplement bilden, das heißt zur invertierten Zahl eine I hinzuaddieren
4. Das Zweier-Komplement mit dem Minuend addieren. Hierbei den Übertrag in der Werthöchste Stelle einklammern (Vorzeichen Stelle)

Beispiel:

```
  I0II0I Minuend (45)
- 00II0I Subtrahend (13)
```

OOIIOI invertieren

IIOOIO -> Einer-Komplement

+OOOOOI

IIOOII -> Zweier-Komplement

IOIIOI

+ IIOOII

(I)IOOOOO = 32

Negative Dualzahlen

Wenn bei der Addition des Zweier-Komplements in n-stelliger Darstellung kein Übertrag in die Stelle n+1 auftritt, ist das Ergebnis eine negative Zahl.
Um den Betrag der negativen Zahl festzustellen, ist vom Ergebnis das Zweierkomplement zu bilden. Bei der Darstellung negativer Zahlen ist die Werthöchste Stelle stets I, diese Stelle kann als Vorzeichenstelle angesehen werden. Computer arbeiten bei der Zahlendarstellung stets mit festgelegter Stellenzahl (Datenbusbreite z.B. 8, 16 oder 32 Bit). Die Werthöchste Stelle ist somit stets bekannt und kann als Vorzeichenstelle verwendet werden, ohne dass Irrtümer entstehen.

3.3 Rechenschaltung

3.3.1 Halbaddierer

Übung:

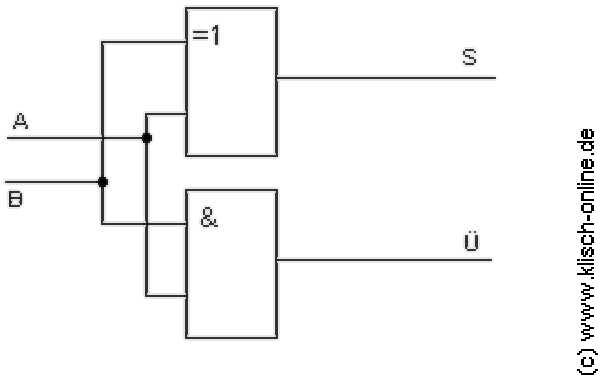
Eine Digitalschaltung soll in der Lage sein, zwei Dualziffern nach den Regeln der Addition im Dualsystem zu addieren. Die Schaltung hat 2 Eingänge A und B und 2 Ausgänge S = Summe und Ü = Übertrag.

a) Stelle die Wahrheitstabelle auf
b) Entwirf und skizziere die Schaltung
c) Bau die Schaltung nach und teste die Funktionen

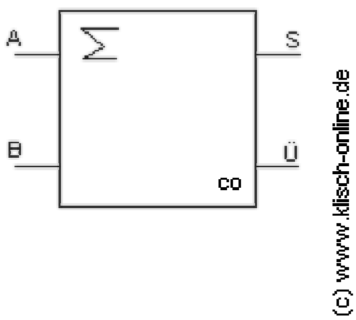
a)

B	A	S	Ü
0	0	0	0
0	I	I	0
I	0	I	0
I	I	0	I

b)



<u>Schaltbild</u>



co ^= carry out

<u>3.3.1 Volladdierer</u>

Übung:

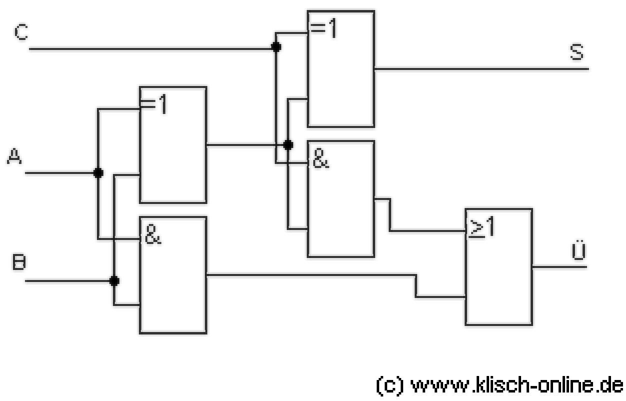
Zum Aufbau von Addierwerken werden Schaltungen benötigt, die 3 Dualziffern addieren können, da bei der Addition von 2 Dualzahlen die Übergänge mit Addiert werden müssen. Die zu entwickelnde Digitalschaltung sollte also in der Lage sein 3 Dualzahlen nach den Regeln der Addition im Dualsystem zu addieren. Die Schaltung hat 3 Eingänge A, B, C und 2 Ausgänge S und Ü.

a) Stelle die Wahrheitstabelle auf
b) Entwirf und skizziere die Schaltung
c) Bau die Schaltung nach und teste die Funktionen

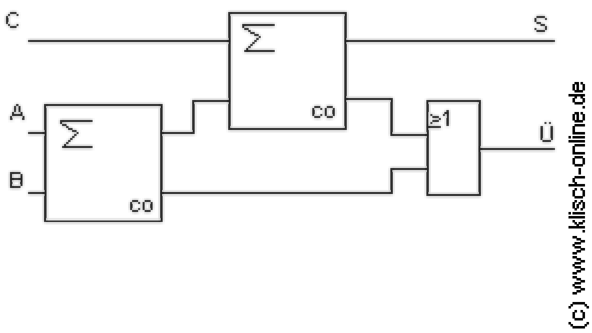
a)

C	B	A	S	Ü
0	0	0	0	0
0	0	I	I	0
0	I	0	I	0
0	I	I	0	I
I	0	0	I	0
I	0	I	0	I
I	I	0	0	I
I	I	I	I	I

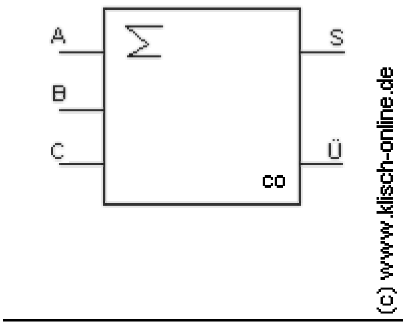
b)



Vereinfachte Darstellung:



<u>Schaltzeichen:</u>



3.5 Hexadezimales Zahlensystem

Basis = 16 ----> 16 Ziffern

0 1 2 3 4 5 6 7 8 9 A B C D E F

...	16^3	16^2	16^1	16^0
	4096	256	16	1

3.5.1 Umwandlung von Hexadezimal- in Dezimalzahlen

Beispiel: 1AE7

1*4096 + 10*256 + 14*16 + 7*1

10 steht für das A
256 steht für 16^2

3.5.2 Umwandlung von Dezimal- in Hexadezimalzahlen

Beispiel: $3999_{(10)}$

3999 : 256 = 15 15x passt 256 in 3999, mit dem Rest wird weiter dividiert
159 : 16 = 9
15

$F9F_{(Hex)} = 3999_{(10)}$

Das F steht für die 15

3.5.3 Umwandlung von Dual- in Dezimalzahlen

Beispiel:

000I. IIII. 0III

1 F 7

Das Viererblockergebnis in Dez wird in Hex umgewndelt

3.5.4 Umwandlung von Hexadezimal- in Dualzahlen

Beispiel:

A F F $E_{(Hex)}$

I0I0. IIII. IIII. III0

4 Grundlagen der Schaltalgebra

Mit Hilfe der Schaltalgebra können Digitalschaltungen optimal entwickelt werden

4.1 Theoreme Rechenregeln

Hier können schaltalgebraische Gleichungen teilweise extrem vereinfacht werden, sodass die Digitalschaltung den kleinstmöglichen Aufwand an Digitalbausteinen aufweist

4.1.1 Theoreme der UND-Verknüpfung

$\overline{A} \wedge \overline{A} = 0$

$\overline{A} \wedge A = 0$

$A \wedge \overline{A} = 0$

$A \wedge A = I$

4.1.2 Theoreme der ODER-Verknüpfung

$\overline{A} \vee \overline{A} = 0$
$\overline{A} \vee A = I$
$A \vee \overline{A} = I$
$A \vee A = I$

4.1.3 Theoreme der Negation

$\overline{\overline{A}} = A$

4.2 Kommutativgesetze (Vertauschungsgesetze)

4.2.1 Kommutativgesetz der UND-Verknüpfung

$x = A \wedge B \wedge C \wedge D$
$x = B \wedge D \wedge A \wedge C$

Anwendung:
$z = A \wedge B \wedge C \wedge \overline{A}$
$z = A \wedge \overline{A} \wedge B \wedge C$

Die Variablen können vertauscht werden

4.2.2 Kommutativgesetz der ODER-Verknüpfung

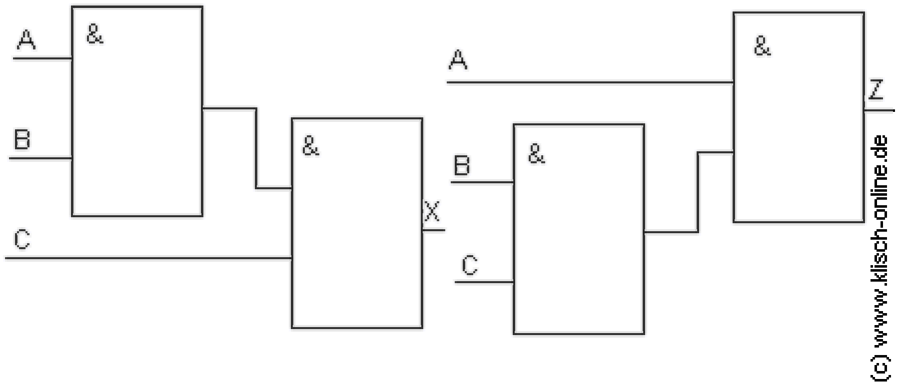
$x = A \vee B \vee C \vee D$
$x = B \vee D \vee A \vee C$

Anwendung:
$z = A \vee B \vee C \vee \overline{A}$
$z = A \vee \overline{A} \vee B \vee C$

Die Variablen können vertauscht werden

4.3 Assoziativgesetze (Verbindungsgesetz)

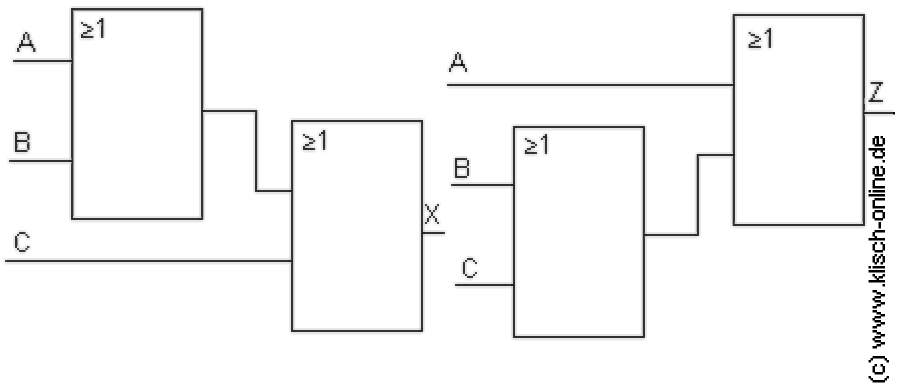
4.3.1 Assoziativgesetz der UND-Verknüpfung



X = (A ∧ B) ∧ C   Z = A ∧ (B ∧ C)

(A ∧ B) ∧ C = A ∧ (B ∧ C)

4.3.2 Assoziativgesetz der ODER-Verknüpfung



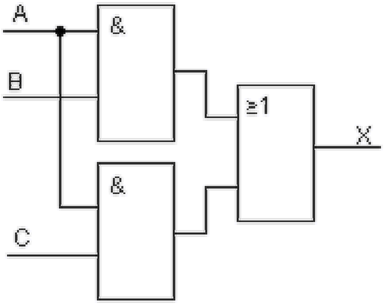
X = (A ∨ B) ∨ C   Z = A ∨ (B ∨ C)

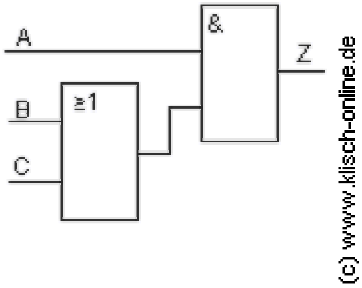
(A ∨ B) ∨ C = A ∨ (B ∨ C)

4.4 Distributivgesetz

Das Distributivgesetz heißt zu Deutsch Verteilungsgesetz. Es entspricht den Regeln über das Ausmultiplizieren und Ausklammern eines Faktors in der normalen Algebra.

4.4.1 Konjunktives Distributivgesetz



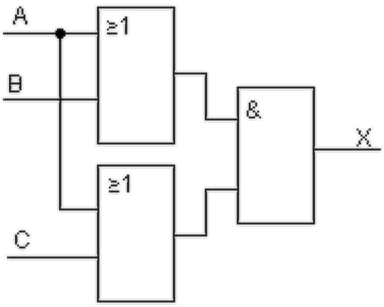


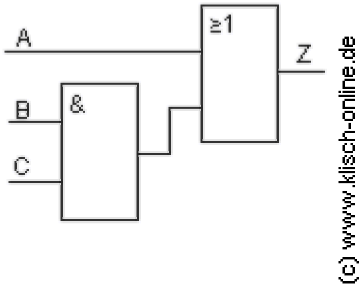
$X = (A \wedge B) \vee (A \wedge C)$

$Z = A \wedge (B \vee C)$

$$(\underline{A \wedge} B) \vee (\underline{A \wedge} C) = \underline{A \wedge} (B \vee C)$$

4.4.2 Disjunktives Distributionsgesetz





$X = (A \vee B) \wedge (A \vee C)$

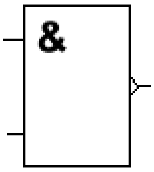
$Z = A \vee (B \vee C)$

$$(A \vee B) \wedge (A \vee C)= A \vee (B \vee C)$$

4.5 Morganische Gesetze

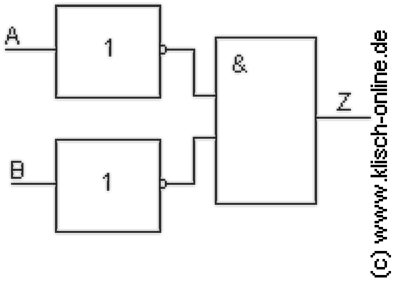
Die morganischen Gesetze haben eine große praktische Bedeutung, wenn Schaltungen ausschließlich aus NAND oder NOR Bausteinen aufgebaut werden soll.

4.5.1 1. Morganisches Gesetz



B	A	X
0	0	I
0	I	I
I	0	I
I	I	0

$X = \overline{(A \wedge B)}$

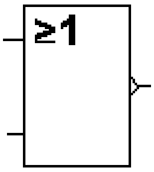


B	A	X
0	0	I
0	I	I
I	0	I
I	I	0

$Z = \overline{A} \vee \overline{B}$

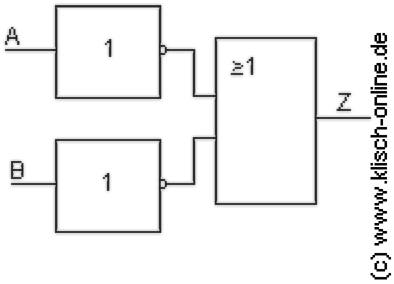
$\overline{(A \wedge B)} = Z = \overline{A} \vee \overline{B}$

4.5.2 2. Morganisches Gesetz



B	A	X
0	0	I
0	I	0
I	0	0
I	I	0

$X = \overline{(A \vee B)}$



B	A	X
0	0	I
0	I	0
I	0	0
I	I	0

$Z = \overline{A} \wedge \overline{B}$

$\overline{(A \vee B)} = Z = \overline{A} \wedge \overline{B}$

4.6 Bindungsregeln

Beispiel:

$x = A \vee B \wedge C$

$x = A \vee (B \wedge C)$

4.7 NAND-Funktion

Jede gewünschte Verknüpfungsschaltung lässt sich nur mit NAND-Bausteinen aufbauen. Hierzu muss eine Schaltalgebraische Gleichung entsprechend umgeformt werden. Ein Weg, der meistens zum Ziel führt, beginnt mit der Vorname einer Doppelnegation. Eine Doppelnegation ändert den Inhalt einer Gleichung nicht.

5 Schaltungssynthese

Digitale Schaltungen werden als sogenannte logische Verknüpfungen für die unterschiedlichsten Steuerung- und Rechenzwecke benötigt. Der Entwurf digitaler Schaltungen wird Schaltsynthese genannt. Die Schaltungssynthese erfolgt nach folgenden Schritten.

1. Exakte Funktionsbeschreibung der gesuchten Schaltung
2. Festlegung der Eingangs- und Ausgangsvariablen und der Bedeutung von Low und High
3. Aufstellen der Wahrheitstabelle
4. Bestimmung der Schaltalgebraischen Gleichung (Disjunktive bzw. Konjunktive Normalform)
5. Vereinfachung und ggf. Umformung (Full-NAND) der Gleichung
6. Dokumentation der Schaltung (Schaltplan + Besonderheiten)

5.1 Disjunktive / Konjunktive Normalform

Definition (Disjunktiv):
Eine Disjunktive Normalform besteht in der Regel aus mehreren Vollkonjunktionen, die durch ODER verknüpft sind. Sie kann auch aus einer einzigen Vollkonjunktion bestehen.

Beispiel:

Fall	C	B	A	X	Z
1	0	0	0	I	I
2	0	0	I	0	I
3	0	I	0	0	I
4	0	I	I	I	I
5	I	0	0	I	I
6	I	0	I	0	I
7	I	I	0	0	I
8	I	I	I	0	0

$X = \overline{A} \wedge \overline{B} \wedge \overline{C} \vee \overline{A} \wedge B \wedge C \vee A \wedge \overline{B} \wedge \overline{C}$

$\overline{Z} = A \wedge B \wedge C$
$\overline{\overline{Z}} = \overline{(A \wedge B \wedge C)}$
$\overline{Z} = A \wedge B \wedge C$ (Disjunktive Normalform)

Die Konjunktive Normalform zieht man der Disjuktiven Normalform vor, wenn man die Wahrheitstabelle am Ausgang häufiger den Wert High als Low hat.

5.2 KV-Diagramme

(Karnaugh-Veitch)

5.2.1 KV-Diagramme für 2 Variablen

Ein KV-Diagramm hat stehts so viel Plätze, wie Vollkonjunktionen möglich sind.
Daraus folgt: 2 Variablen => 4 Plätze

Q	A	$\overline{A}$
B	$A \wedge B$	$A \wedge B$
$\overline{B}$	$A \wedge B$	$A \wedge B$

Die entsprechende Vollkonjunktion wird im KV-Diagramm durch eine 1 dargestellt
Beispiel:

B	A	Q
0	0	I
0	I	I
I	0	0
I	I	0

Q	A	$\overline{A}$
B		
$\overline{B}$	1	1

Vereinfachungen:

Sind Vollkonjunktionen, das heißt, Einsen im KV-Diagramm "benachbart", können sie in Päckchen zusammengefasst werden.
Variablen, die als Koordinaten negiert und nicht negiert auftrete, entfallen.

Beispiele:

Q	A	$\overline{A}$
B		1
$\overline{B}$	1	

nicht benachbart

Q	A	$\overline{A}$
B		
$\overline{B}$	1	1

=> $Q = \overline{B}$

Q	A	$\overline{A}$
B	1	
$\overline{B}$	1	

=> $Q = A$

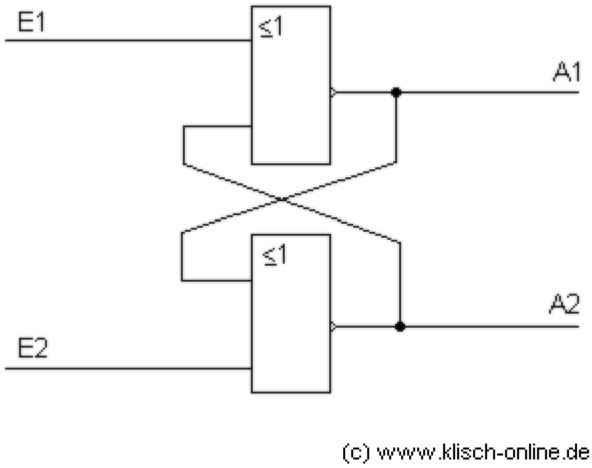
X	A	$\overline{A}$
B		1
$\overline{B}$	1	1

=> $X = \overline{A} \vee \overline{B}$

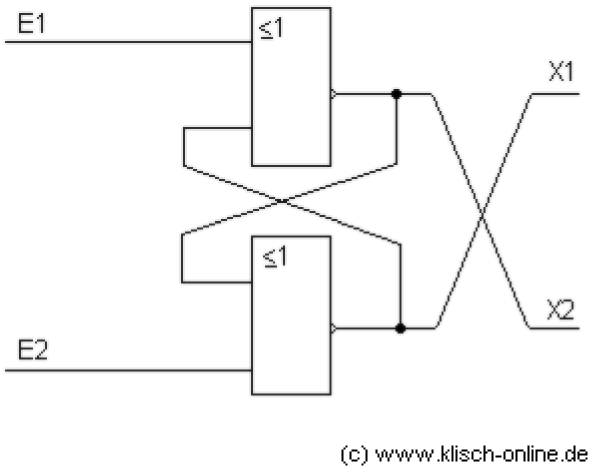
6 Signalspeicher

In der Digitaltechnik müssen Signale häufig gespeichert werden. Diese Aufgaben übernehmen die sogenannten Flip-Flops.

6.1 NOR-Latch

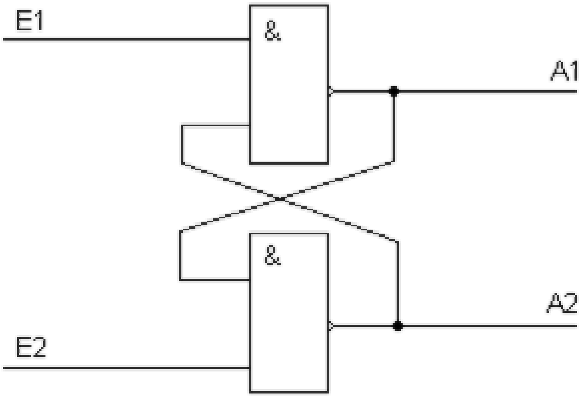


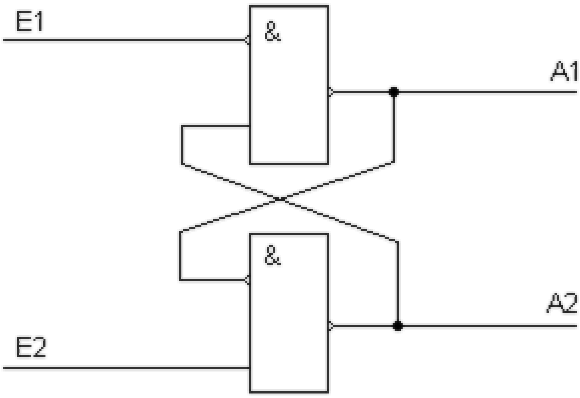
Nach den allgemeinen Regeln für Flip-Flops soll der Setzeingang (E1) den gegenüberliegenden Ausgang setzen. Der Rücksetzeingang (E2) soll den gegenüberliegenden Ausgang setzen. Soll aus dem NOR-Latch ein RS-Flip-Flop aufgebaut werden, müssen die Ausgänge vertauscht werden.



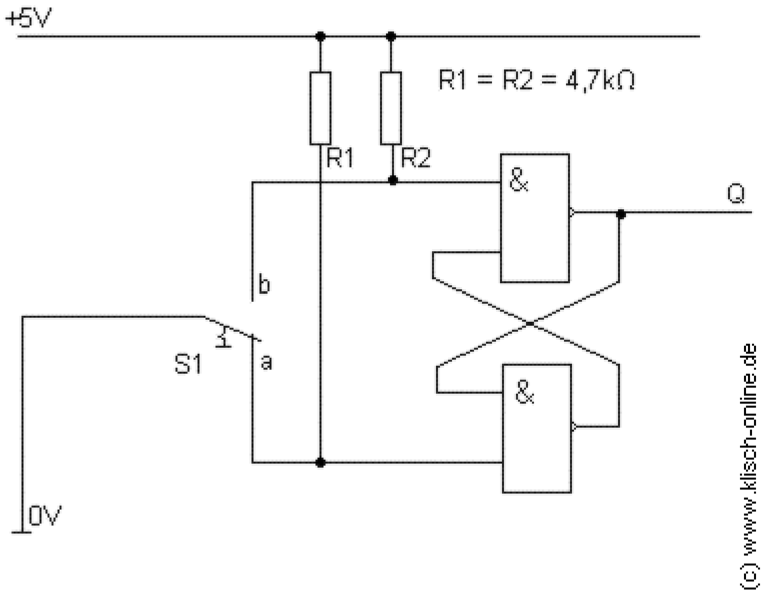
S	R	X1	X2	
I	0	I	0	Setzen
0	I	0	I	Rücksetzen
0	0	X1n	X2n	Speicher
I	I	(0)	(0)	-> Unbestimmt

6.2 NAND-Latch





6.2.1 Schaltungsbeispiel "Prellfreier Schalter"



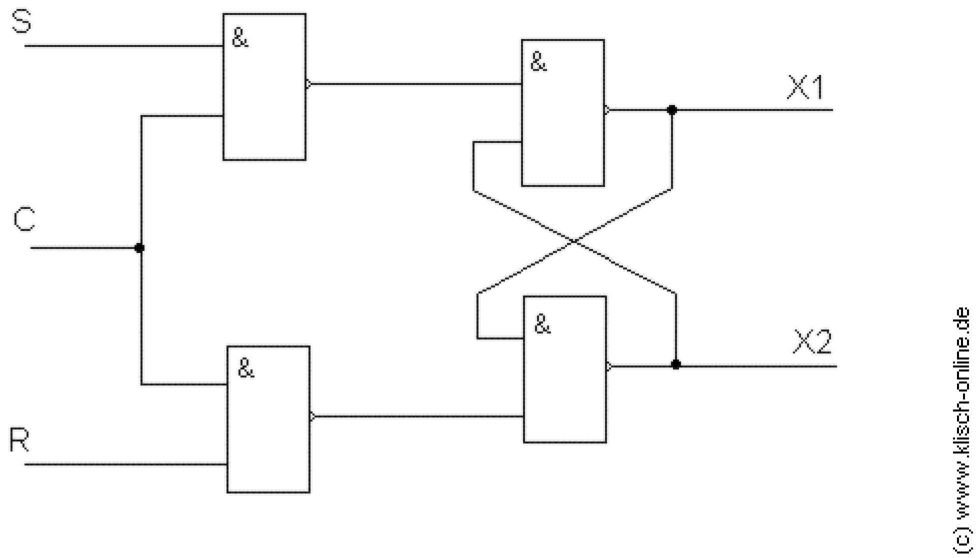
6.3 Taktzustandsgesteuerte Flip-Flops

In der Digitaltechnik ist es häufig wünschenswert, dass erst zu einem bestimmten Zeitpunkt oder in einem bestimmten Zeitbereich die Eingangsinformationen an einen Ausgang gelangen und so gespeichert werden.

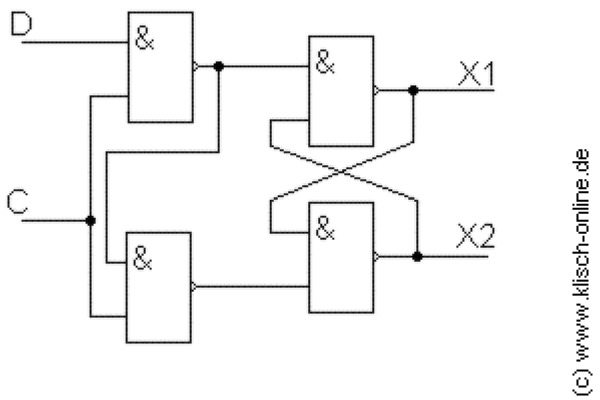
Vorteile:

1. Die Übernahme eines Eingangssignals zum Ausgang kann durch einen besonderen Befehl ausgelöst werden
2. Störimpulse können außerhalb dieser Zeit keinen Einfluss ausüben.

6.3.1 Taktzugangsgesteuertes SR-Flip-Flop

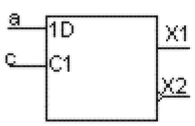


6.3.2 D-Flip-Flop (Delay-Flip-Flop)



Das D-Flip-Flop übernimmt und speichert das Signal 0 oder 1 vom D-Eingang immer dann, wenn der takt C auf High gesetzt wird. Ein D-Flip-Flop speichert also 1 Bit. 8 D-Flip-Flops speichern 1 Byte.
Ein D-Flip-Flop ist ein Grundbaustein in statischen Schreib-Lese-Speichern (RAM) von Mikrocomputern.

Schaltzeichen:

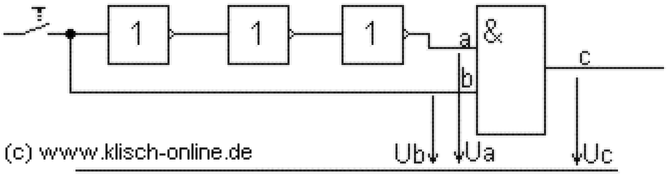


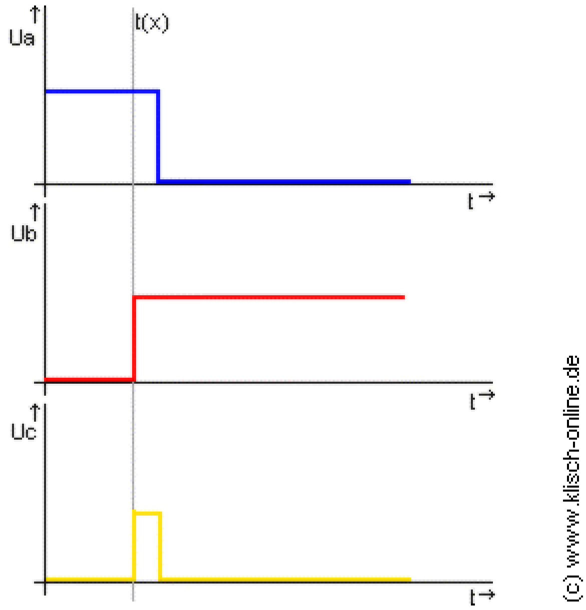
6.4 Einflankengesteuerter Flip-Flop

Mit der Taktflankensteuerung erreicht man ein sehr genaues gleichzeitiges Schalten vieler Flip-Flops. Selbst bei größeren Fertigungstoleranzen ergeben sich fast keine Abweichungen vom Soll- Zeitschaltpunkt.
Ein weiterer Vorteil der Taktflankensteuerung ist die Verminderung der Störanfälligkeit. Störsignale an den Eingängen können nur dann Störungen verursachen, wenn sie in dem sehr kurzen Zeitraum des Schaltens gerade anliegen.

Erzeugen der Taktflanke

Funktionsprinzip: Ausnutzung der Signallaufzeit von Digitalbausteinen.





Negative Taktflankensteuerung