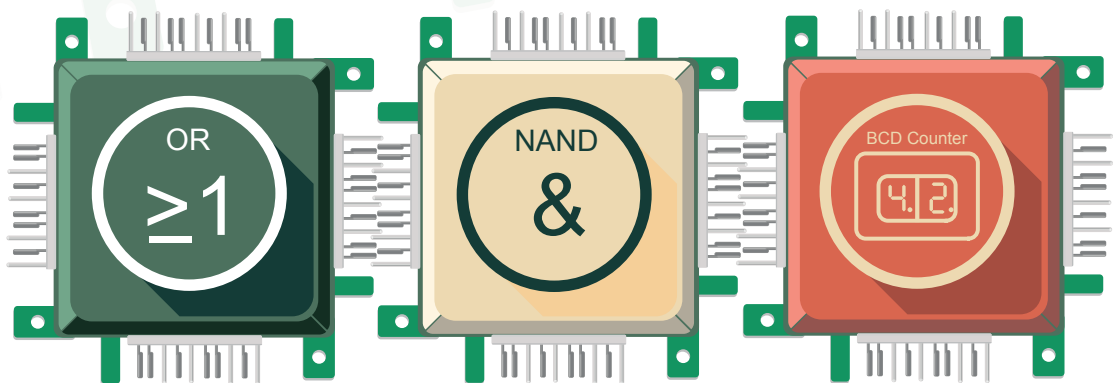


Brick 'R'
knowledge

Logic Set

Experimental kit by Brick'R'knowledge
Experimentierkasten von Brick'R'knowledge



Impressum

Brick'R'knowledge Logic Set Anleitung
Rev. 1.1
Datum: 26.04.2017

ALLNET® und Brick'R'knowledge® sind eingetragene Warenzeichen der ALLNET® GmbH Computersysteme.

ALLNET® GmbH Computersysteme
Brick'R'knowledge
Maistraße 2
D-82110 Germering

© Copyright 2017 ALLNET GmbH Computersysteme. Alle Rechte vorbehalten.

Alle in dieser Anleitung enthaltenen Informationen wurden mit größter Sorgfalt und nach bestem Wissen zusammengestellt. Dennoch sind Fehler nicht ganz auszuschließen. Für die Mitteilung eventueller Fehler sind wir jederzeit dankbar. Bitte sende diese an info@brickrknowledge.de.



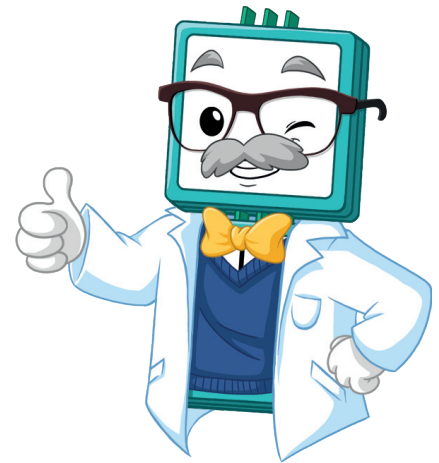
Inhaltsverzeichnis

1. Sicherheitshinweise	5
2. Grundlagen des Brick'R'knowledge Systems	6
2.1 Der Masse-Brick	6
2.2 Die Spannungsversorgung	6
2.3 Die Steckverbinder	7
2.4 Signalführung	7
3. Das Logic Set	8
3.1 Die Bausteine im Überblick	8
4. Grundlagen der digitalen Schaltungstechnik	16
4.1 Was heißt eigentlich digital?	16
4.2 Logische Funktionen	16
4.3 Zahlensysteme im Vergleich	18
4.4 Logikpegel	19
4.4.1 Spannungsbereiche	19
4.4.2 Logische Zustände	20
4.5 Schaltnetz	21
4.6 Schaltwerk	21
4.7 Glitches	22
4.8 Programmierbare Logikbausteine	23
4.8.1 Programmable Array Logic (PAL) bzw. Generic Array Logic (GAL)	23
4.8.2 Complex Programmable Logic Device (CPLD)	23
4.8.3 Field Programmable Gate Array (FPGA)	23
4.8.3.1 FPGA-Grundstruktur	24
4.8.3.2 Konfigurierbare Logik-Blöcke (CLBs)	24
4.8.3.3 Ein-/Ausgangs-Blöcke (IOBs)	25
5. Schaltungsbeispiele	26
5.1 Grundsaltungen	26
5.1.1 Inverter	26
5.1.2 AND-Gatter	27
5.1.3 OR-Gatter	28
5.1.4 NAND-Gatter	29
5.1.5 NOR-Gatter	30
5.1.6 XOR-Gatter	31
5.1.7 XNOR-Gatter	32
5.2 Entprellschaltung	33
5.2.1 Entprellung auf Gatterebene	33
5.2.2 Bricks als entprellte Taktquelle	33
5.3 Digitales Rechenwerk	34
5.3.1 1-Bit-Halbaddierer	34
5.3.2 1-Bit-Volladdierer	35
5.3.3 4-Bit-Volladdierer	37
5.4 Flipflop / Bistabile Kippstufe	38
5.4.1 Flipflop-Typen	39
5.4.1.1 Nicht taktgesteuerte Flipflops	39
5.4.1.2 Zustandsgesteuerte Flipflops	39
5.4.1.3 Einflankengesteuerte Flipflops	39
5.4.1.4 Zweiflankengesteuerte Flipflops	39
5.4.2 RS-Flipflop	40
5.4.3 D-Flipflop	41
5.4.4 JK-Flipflop	42

5.5 Schieberegister	44
5.5.1 Schieberegister mit seriellem Laden	44
5.6 Zähler	45
5.6.1 Asynchroner 4-Bit-Binär-Zähler	46
5.6.1.1 Asynchroner Aufwärtszähler	47
5.6.1.2 Asynchroner Abwärtszähler	48
5.6.1.3 Frequenzteiler	49
5.6.2 Asynchroner 4-Bit BCD-Zähler	50
5.6.3 Synchroner 3-Bit Binär-Zähler	52
5.6.4 BCD-Zähler mit Übertrag und Reset	54
6. Brick Community	56
7. Brick Sets im Überblick	59

Vorwort

Das Brick'R'knowledge Experimentiersystem wurde zum ersten mal auf der HAM Radio Ausstellung am 28.06.2014 von Rolf-Dieter Klein (Amateurfunkrufzeichen: DM7RDK) vorgestellt. Das Besondere an unseren Elektroniksets ist, dass die einzelnen Bausteine über ein Stecker-System verbunden werden, bei dem die zusammenzufügenden Teile baugleich sind (Hermaphrodite). So können auch knifflige Stromkreise realisiert werden. Auch das Zusammenstecken der einzelnen Bausteine in verschiedenen Winkeln ist möglich! Für die Rückführung der Masse (0Volt) sind gleich zwei Kontakte vorhanden! Damit lassen sich kompakte Schaltungen aufbauen, bei der die Masse-Rückführung für eine stabile Spannungsversorgung der Bausteine sorgt. Eine weitere Besonderheit ist, dass man solche Schaltungen sehr leicht erklären und dokumentieren kann.



Viel Spass mit dem Logic Set wünscht

Rolf-Dieter Klein

Quellen- und Literaturverzeichnis:

Unter den folgenden Links finden Sie umfassendes Grundlagenwissen über digitale Schaltungstechnik und die Gesetzmäßigkeiten der Schaltalgebra.

- Elektronik-Kompendium von Patrick Schnabel: <http://www.elektronik-kompendium.de/sites/dig/index.htm>
- Elektroniktutor von Detlef Mietke: <http://www.elektroniktutor.de/digital1.html>
- Wikipedia: <https://www.wikipedia.de>

Die aktuellste Handbuch-Version finden Sie stets unter:
<http://www.brickrknowledge.de/downloads>

1. Sicherheitshinweise

Achtung, die Bausteine des Elektroniksets NIE direkt an das Stromnetz (230V) anschließen, andernfalls besteht Lebensgefahr!

Zur Spannungsversorgung empfehlen wir das mitgelieferte 9V-Netzteil (oder opt. 9V-Batterie-Brick). Die Versorgungsspannung beträgt hier gesundheitsungefährliche 9Volt bei einem Stromfluss von max. 1 Ampere. Bitte tragen Sie auch Sorge dafür, dass offen herumliegende Drähte nicht in Berührung oder Kontakt mit Steckdosenleisten (gewöhnliche Zimmerverteiler) kommen bzw. in diese hineinfallen, auch hier besteht andernfalls die Gefahr eines gesundheitsgefährlichen Stromschlags bzw. elektrischen Schocks. Schauen Sie niemals direkt in eine Leuchtdiode (LED), da hier die Gefahr besteht, die Netzhaut zu schädigen (Blendung). Die Netzhaut befindet sich im Auge und hat die Aufgabe, die einfallenden Lichtreize durch die auf ihr befindlichen Zapfen (das Farbsehen) und die ebenfalls auf ihr befindlichen Stäbchen (Hell-, Dunkelsehen) in für das Gehirn verwertbare Reize umzuwandeln. Es ist unbedingt darauf zu achten, dass mitgelieferte Netzteil nach den Versuchsaufbauten wieder von allen Bausteinen zu trennen, andernfalls besteht die Gefahr eines Elektrobrandes!

Bausteine oder andere Teile des Elektroniksets nicht verschlucken, andernfalls sofort einen Arzt aufsuchen!

2. Grundlagen des Brick'R'knowledge Systems

2.1 Der Masse-Brick

Der Masse-Brick ist ein besonderer Baustein des Brick'R'knowledge Systems. Er spart zusätzliche Verbindungen mit Hilfe anderer Bricks oder Leitungen. Hier wird das Geheimnis unserer vierpoligen Verbinders offenbart. Die mittleren zwei Kontakte sind für die Signalübertragung reserviert, so wie es der Aufdruck verrät. Die äußeren Kontakte werden zum Schließen des Stromkreises, also der Rückführung des Stromflusses zur Spannungsquelle benutzt. Das realisiert der Masse-Brick. Dieser Brick heißt deshalb Masse-Brick, weil in der Elektronik mit der Bezeichnung „Masse“ nicht etwa das Gewicht eines Gegenstandes beschrieben wird, sondern das Bezugspotential auf das sich alle anderen Potentiale beziehen. Der Masse-Brick stellt in allen Brick'R'knowledge-Sets genau diese Verbindung zu 0V her.

In unserer Schaltung sind das 9 Volt gegenüber 0 Volt: Man spricht einfach nur „Neun Volt“. Man erstellt in der Elektronik Schaltungen so, dass nachdem alle Bauelemente in ihrer Funktionsweise in die mehr oder weniger komplexen Stromkreise eingebracht sind, diese mit der „Masse“ verbunden werden. Schaltpläne sind nur so zu lesen.

In der Praxis verbindet unser Masse-Brick (rechts) die beiden mittleren Kontakte mit den beiden äußeren. Doch keine Angst, wir verursachen damit keinen Kurzschluss, denn der Strom durchfließt ja über die mittleren Kontakte die Bauelemente in unserem Brick-Stromkreis. Der rote Pfeil in der Abbildung symbolisiert den Pluspol und die braunen Pfeile zeigen die Masserückführung zum Minuspol der Spannungsversorgung.

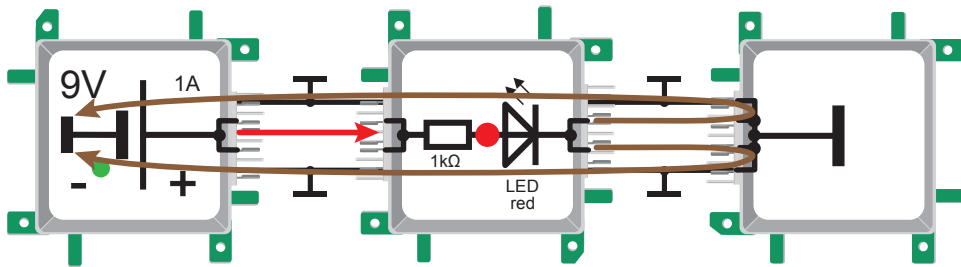


Abb. 1: Der Masse-Brick

2.2 Die Spannungsversorgung



Abb. 2: Netzteil-Adapter

Die Spannungsversorgung des Logic Sets erfolgt über das mitgelieferte 9V-Steckernetzteil (ALL-BRICK-0221). Das Netzteil liefert eine stabilisierte Gleichspannung von 9V und einen Maximalstrom von 1 A. Bei Überlastung schaltet das Netzteil ab, d. h. es ist kurzschlussicher. Eine LED zeigt an, sobald der Brick Spannung bereitstellt.

Gatter-, Flipflop und Zähler-Bricks sind aktive Bauelemente. "Unter der Haube" kommen klassische Logik-Bausteine in CMOS-Technologie (siehe auch Kap. 4.4 „Logikpegel“) zum Einsatz, die eine eigene Versorgung benötigen. Beachten Sie dies stets beim "Verdrahten" Ihrer Bricks. Theoretisch können Sie die Logik-Bricks an jeder stabilisierten Gleichstromquelle im Bereich +5V bis +15V betreiben, wir empfehlen jedoch das mitgelieferte 9V-Steckernetzteil zu verwenden.

Bitte stecken Sie den Versorgungs-Brick stets als letzten Brick an Ihre Schaltung, nachdem Sie diese nochmals kontrolliert haben. Am Ende der Versuchsdurchführung muss das Netzteil vom Stromnetz getrennt werden!

Optional steht auch ein Versorgungs-Brick via 9V-Blockbatterie (ALL-BRICK-0001) zur Verfügung.

2.3 Die Steckverbinder

Beim Zusammenstecken der Bricks muss darauf geachtet werden, dass sich die Kontakte richtig berühren, da sonst die Gefahr von Unterbrechungen oder sogar Kurzschlüssen besteht!

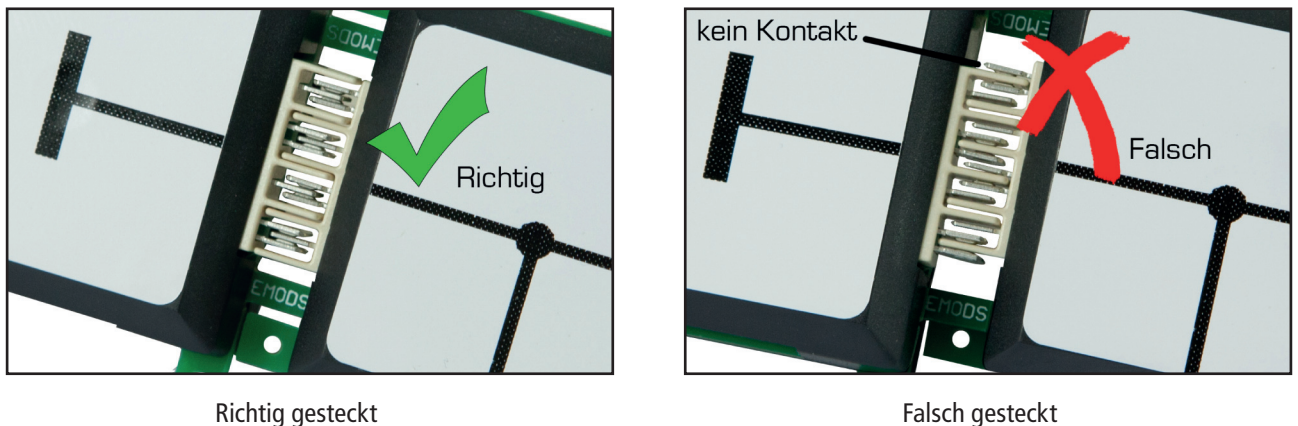


Abb. 3: Die Steckverbinder

Im linken Bild sieht Ihr eine richtig gesteckte Verbindung. Die Verbindung besteht jeweils aus kleinen Stiften, die sich mechanisch verklemmen und dabei eine elektrische Verbindung herstellen. Um eine Isolation zwischen den Kontakten zu gewährleisten und einen Kurzschluss zu verhindern sind dazwischen Stege aus Kunststoff eingebracht, die den elektrischen Strom nicht leiten.

Ein Beispiel einer fehlerhaften Verbindung ist im rechten Bild zu sehen. Hier treffen Isolierstege auf Kontakte, sodass kein Strom fließen kann. Der Stromkreis bleibt „offen“ oder ist instabil und die Funktion der Schaltung ist nicht gegeben.

Achtung: Es ist wichtig, grundsätzlich immer den richtigen Sitz der Kontaktstifte zu kontrollieren. Weichen diese zu weit voneinander ab, kann es zu einem Kurzschluss kommen. Dann findet der Stromfluss nicht durch unsere Bauelemente mit der erhofften Wirkung statt, sondern sucht sich den kürzesten Weg zurück zur Spannungsquelle.

Ein Kurzschluss führt zum Maximalstromfluss, da der einzige Widerstand den der elektrische Strom überwinden muss, der Innen-Widerstand der Spannungsquelle ist. Dieser Widerstand ist anschaulich sehr klein, sodass der Kurzschlussstrom bei längerer Dauer zur Überhitzung führen kann. Es besteht Brandgefahr!



Wichtig: Immer die richtige Stellung der Kontakte überprüfen!

2.4 Signalführung

Für die Kennzeichnung der beiden mittleren Signalpins der Brick'R'knowledge Stecker gilt:

Falls nur ein einpoliges Signal über die Anschlüsse des Bricks geführt wird, zeigt das Label nur eine Linie, die mittig zum jeweiligen Stecker führt. In diesem Fall sind die beiden mittleren Signalpins stets verbunden (linkes Beispiel).

Sofern es sich um einen Brick handelt, der zwei getrennte Signale über einen oder mehrere seiner Stecker führt, werden grundsätzlich beide Signalpins gekennzeichnet. Falls an einem Stecker beide Signalpins verbunden sind, wie am oberen und unteren Stecker in Abb. 4 rechts zu sehen, wird dies entsprechend dargestellt.



Abb. 4: Signalführung einpolig (links) und zweipolig (rechts)

3. Das Logic Set

Das Brick'R'knowledge Logic Set soll die Grundlagen der digitalen Schaltungstechnik auf leicht verständliche Weise vermitteln. Dies geschieht zum einen durch dieses ausführliche, didaktische Handbuch und zum anderen durch das praxisorientierte Zusammenstecken und Experimentieren mit den Bricks. Wir beginnen mit einfachen logischen Verknüpfungen über Halb- und Voll-Addierer bis hin zu Flipflop-Schaltungen und einem 4-stelligen BCD-Zähler. Als Taktgenerator kommt der bekannte Timer-Baustein 555 zum Einsatz.

3.1 Die Bausteine im Überblick

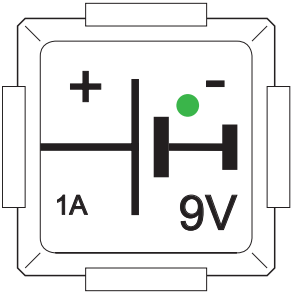
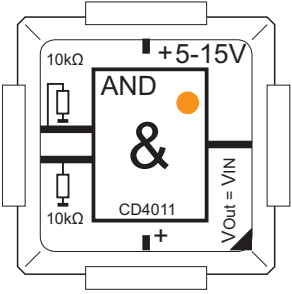
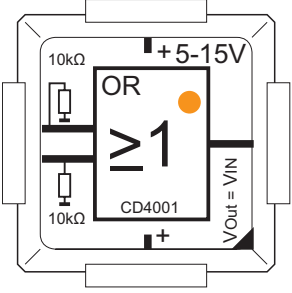
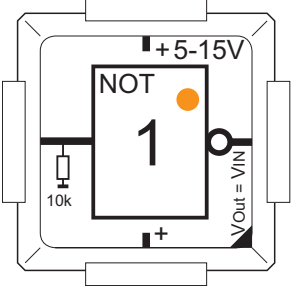
Abbildung	Anzahl	Art.-Nr. / Brick-ID	Kurzbeschreibung
	1	Art.-Nr.: 118627 Brick-ID: ALL-BRICK-0221	9V Netzteiladapter Der 9V-Netzadapter liefert maximal 1 A Strom für die Bricks. Er ist stabilisiert und vor Kurzschlüssen gesichert. Eine LED zeigt an sobald er in Betrieb ist. Der Pluspol ist herausgeführt und der Minuspol ist mit Masse verbunden. Setzen Sie diesen Brick als letztes ein, nachdem Sie die Schaltung noch einmal kontrolliert haben.
	2	Art.-Nr.: 128276 Brick-ID: ALL-BRICK-0437	AND-Gatter Das Gatter realisiert eine logische UND-Verknüpfung mit 2 Eingängen, basierend auf 1/4 CD4011. Die Eingänge sind über je einen 10 kΩ Pulldown-Widerstand mit Masse verbunden. Versorgungsspannung: +5..15V.
	1	Art.-Nr.: 128277 Brick-ID: ALL-BRICK-0438	OR-Gatter Das Gatter realisiert eine logische ODER-Verknüpfung mit 2 Eingängen, basierend auf 1/4 CD4001. Die Eingänge sind über je einen 10 kΩ Pulldown-Widerstand mit Masse verbunden. Versorgungsspannung: +5..15V.
	1	Art.-Nr.: 135015 Brick-ID: ALL-BRICK-0634	Inverter Inverter gibt das Eingangssignal negiert am Ausgang aus. Der Eingang ist über einen 10 kΩ Pulldown-Widerstand mit Masse verbunden. Versorgungsspannung: +5..15V.

Abbildung	Anzahl	Art.-Nr. / Brick-ID	Kurzbeschreibung
	1	Art.-Nr.: 113684 Brick-ID: ALL-BRICK-0057	NAND-Gatter Das Gatter realisiert eine logische UND-Verknüpfung mit 2 Eingängen und negiertem Ausgang, basierend auf 1/4 CD4011. Die Eingänge sind über je einen 10 kΩ Pulldown-Widerstand mit Masse verbunden. Versorgungsspannung: +5..15V.
	2	Art.-Nr.: 113685 Brick-ID: ALL-BRICK-0058	NOR-Gatter Das Gatter realisiert eine logische ODER-Verknüpfung mit 2 Eingängen und negiertem Ausgang, basierend auf 1/4 CD4001. Die Eingänge sind über je einen 10 kΩ Pulldown-Widerstand mit Masse verbunden. Versorgungsspannung: +5..15V.
	2	Art.-Nr.: 128278 Brick-ID: ALL-BRICK-0439	XOR-Gatter Das Gatter realisiert eine logische Exklusiv-ODER-Verknüpfung mit 2 Eingängen, basierend auf 1/4 CD4070. Die Eingänge sind über je einen 10 kΩ Pulldown-Widerstand mit Masse verbunden. Versorgungsspannung: +5..15V.
	1	Art.-Nr.: 128279 Brick-ID: ALL-BRICK-0440	XNOR-Gatter Das Gatter realisiert eine logische Exklusiv-ODER-Verknüpfung mit 2 Eingängen und negiertem Ausgang, basierend auf 1/4 CD4070. Die Eingänge sind über je einen 10 kΩ Pulldown-Widerstand mit Masse verbunden. Versorgungsspannung: +5..15V.
	4	Art.-Nr.: 113683 Brick-ID: ALL-BRICK-0056	JK-Flipflop JK-Flipflop (einflankengesteuert) basierend auf 1/2 CD4027. JK-Eingänge und Takt sind über je einen 10 kΩ Pulldown-Widerstand mit Masse verbunden. Versorgungsspannung: +5..15V. Beachte: Am Takteingang C sollte keine weitere Last wie z. B. eine LED angeschlossen werden!

Abbildung	Anzahl	Art.-Nr. / Brick-ID	Kurzbeschreibung
	4	Art.-Nr.: 135012 Brick-ID: ALL-BRICK-0632	<p>D-Flipflop D-Flipflop (flankengesteuert). Daten-Eingang und Takt sind über je einen 10 kΩ Pull-down-Widerstand mit Masse verbunden. Versorgungsspannung: +5..15V.</p> <p>Beachte: Am Takteingang C sollte keine weitere Last wie z. B. eine LED angeschlossen werden!</p>
	1	Art.-Nr.: 135014 Brick-ID: ALL-BRICK-0633	<p>D-Flipflop mit Set und Reset D-Flipflop (flankengesteuert) mit Set- und Reset-Eingang im 2x1-Format. Alle Eingänge sind über je einen 10 kΩ Pull-down-Widerstand mit Masse verbunden. Versorgungsspannung: +5..15V.</p> <p>Beachte: Am Takteingang C sollte keine weitere Last wie z. B. eine LED angeschlossen werden!</p>
	2	Art.-Nr.: 135011 Brick-ID: ALL-BRICK-0631	<p>BCD-Counter Der BCD-Counter-Brick basiert auf dem Synchronzähler-Baustein CD4518. Damit kann man auf einfache Weise einen 2-stelligen BCD-Zähler aufbauen. Takt- und Reset-Eingang sind über je einen 10 kΩ Pull-down-Widerstand mit Masse verbunden. Zur Kaskadierung mehrerer BCD-Zähler ist der low-aktive Übertrag (Carry Out) der Zehner-Stelle herausgeführt. Der Takt-Eingang (Clock) zählt mit positiver Flanke. Versorgungsspannung: +9..15V.</p>
	1	Art.-Nr.: 137826 Brick-ID: ALL-BRICK-0643	<p>Taktgeber (Clock) Der Taktgeber-Brick dient als Taktquelle für die Logik-Bricks. Die Schaltung basiert auf dem Universal-Timer-Baustein 555. Über das Potentiometer ist ein Rechtecksignal von ca. 0,5Hz bis 100Hz einstellbar. Um einzelne Rechteckimpulse zu erzeugen kann der Taktgeber 1:1 gegen den Brick "Taster entprellt" (ALL-BRICK-0641) getauscht werden. Versorgungsspannung: +5..15V.</p>

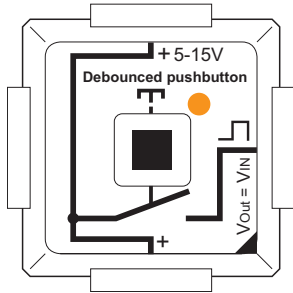
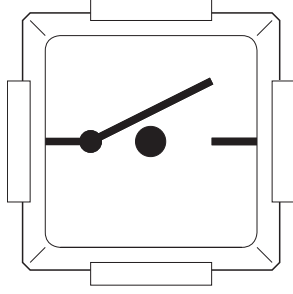
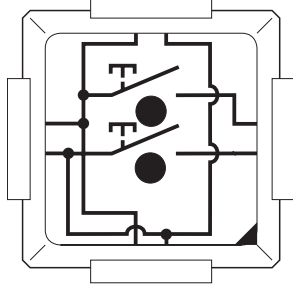
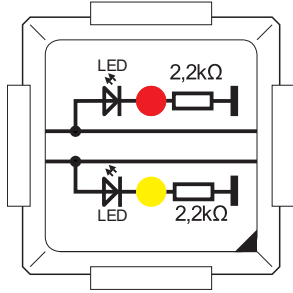
Abbildung	Anzahl	Art.-Nr. / Brick-ID	Kurzbeschreibung
	1	Art.-Nr.: 137825 Brick-ID: ALL-BRICK-0641	<p>Taster entprellt (Debounced pushbutton) Mit diesem Spezialtaster kann man auf einfache Weise einen entprellten Einzeltakt für Flipflop-Schaltungen erzeugen. Er kann 1:1 gegen den Taktgeber-Brick (ALL-BRICK-0643) getauscht werden. Man erhält die Möglichkeit, die Einzelschritte eines Schaltwerks in Ruhe zu studieren. Die Versorgung ist von oben nach unten durchgeschleift. Versorgungsspannung: +5..15V.</p> <p>Unter der Haube kommt ein D-Flipflop-Baustein vom Typ CD4013B mit asynchronem Set- und Reset-Eingang zum Einsatz.</p>
	3	Art.-Nr.: 113644 Brick-ID: ALL-BRICK-0017	<p>Taster-Brick einfach Der Taster ist als einpoliger Schließer ausgeführt. Durch Drücken wird eine elektrische Verbindung zwischen linkem und rechtem Anschluss hergestellt. Beachte: ein Taster ist ein elektromechanisches Bauteil und liefert kein entprelltes Signal. Dies kann bei Verwendung mit Logik-Schaltungen zu Fehlimpulsen führen (siehe auch Kap. 5.2 auf Seite 33 zur Tastenentprellung).</p>
	1	Art.-Nr.: 137824 Brick-ID: ALL-BRICK-0642	<p>Doppeltaster Dieser Brick enthält 2 einpolige Schließer. Damit können die Eingänge von Gattern oder Flipflops bequem beschaltet werden. Zusätzlich sind die Signalpfade getrennt vom oberen zum unteren Anschluss durchverbunden. Schließen Sie die Versorgung am oberen und/oder unteren Anschluss an und sparen Sie so in vielen Situationen zahlreiche Leitungs-Bricks.</p>
	1	Art.-Nr.: 125693 Brick-ID: ALL-BRICK-0410	<p>Dual-LED auf Masse (rot/gelb) In dem Baustein sind zwei LEDs (rot/gelb) untergebracht, die intern mit Masse verbunden sind. Die Signalleitungen sind getrennt durchverbunden. Beide LEDs sind über einen 2,2kΩ Vorwiderstand vor zu hohem Stromfluss geschützt. Sie sind für 2 mA Strom ausgelegt. Die beiden Widerstände sind intern mit Masse verbunden sodass man den Baustein direkt anschließen kann. Zusätzlich werden die beiden Signale durchgeschleift.</p>

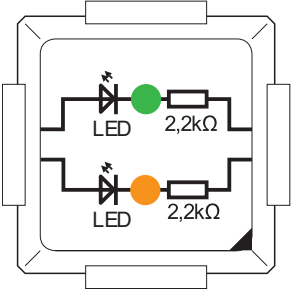
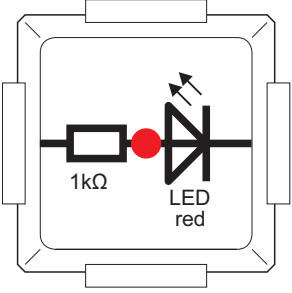
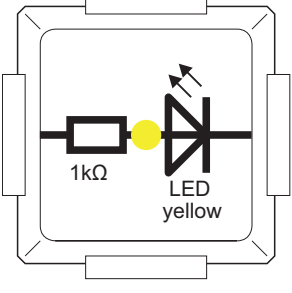
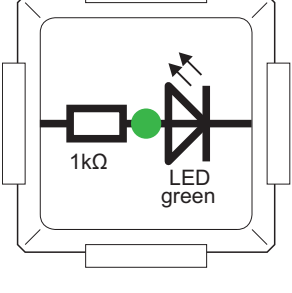
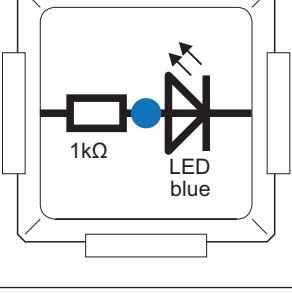
Abbildung	Anzahl	Art.-Nr. / Brick-ID	Kurzbeschreibung
	1	Art.-Nr.: 125682 Brick-ID: ALL-BRICK-0409	Dual-LED (grün/orange) In dem Baustein sind zwei LEDs (grün/orange) untergebracht. Beide LEDs sind über einen 2,2 kΩ Vorwiderstand vor zu hohem Stromfluss schützt. Sie sind für 2 mA Strom ausgelegt. Die Widerstände sind mit der anderen Seite des Bricks verbunden und getrennt geführt. Generell ist darauf zu achten, dass die Kathode mit Masse und die Anode mit positivem Potential verbunden werden muss.
	1	Art.-Nr.: 113636 Brick-ID: ALL-BRICK-0009	LED rot Rote LED mit eingebautem 1 kΩ Vorwiderstand. Die Kathode (Minuspole) ist mit einem Querstrich am LED-Symbol gekennzeichnet.
	1	Art.-Nr.: 113638 Brick-ID: ALL-BRICK-0011	LED gelb Gelbe LED mit eingebautem 1 kΩ Vorwiderstand. Die Kathode (Minuspole) ist mit einem Querstrich am LED-Symbol gekennzeichnet.
	1	Art.-Nr.: 113639 Brick-ID: ALL-BRICK-0012	LED grün Grüne LED mit eingebautem 1 kΩ Vorwiderstand. Die Kathode (Minuspole) ist mit einem Querstrich am LED-Symbol gekennzeichnet.
	1	Art.-Nr.: 113637 Brick-ID: ALL-BRICK-0010	LED blau Blaue LED mit eingebautem 1 kΩ Vorwiderstand. Die Kathode (Minuspole) ist mit einem Querstrich am LED-Symbol gekennzeichnet.

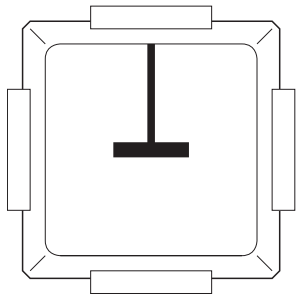
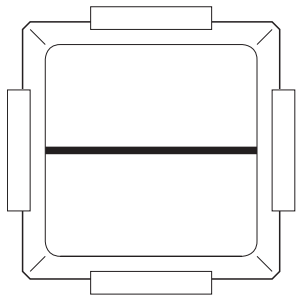
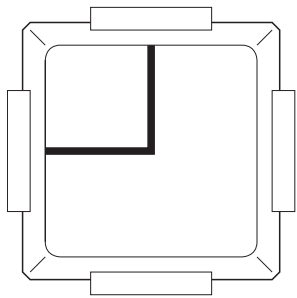
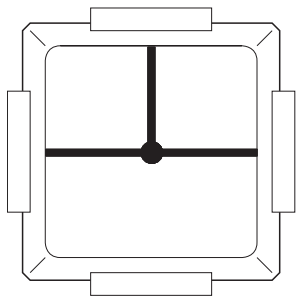
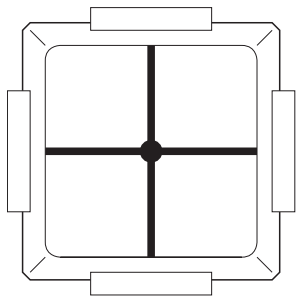
Abbildung	Anzahl	Art.-Nr. / Brick-ID	Kurzbeschreibung
	4	Art.-Nr.: 113630 Brick-ID: ALL-BRICK-0003	Masse-Brick Mit dem Masse-Brick kann man auf einfache Weise an jede Stelle der Schaltung Masse anlegen. Er ist notwendig um den/die Stromkreis(e) zu schließen. Der Masse-Brick verbindet die beiden mittleren Kontakte des Anschlusses mit den beiden außen liegenden Masseleitungen.
	10	Art.-Nr.: 113631 Brick-ID: ALL-BRICK-0004	Gerade Die gerade Leitung verbindet zwei gegenüberliegende Anschlüsse miteinander.
	8	Art.-Nr.: 113632 Brick-ID: ALL-BRICK-0005	Ecke Mit dem Eck-Brick werden zwei angrenzende Seiten miteinander verbunden.
	8	Art.-Nr.: 113633 Brick-ID: ALL-BRICK-0006	T-Kreuzung Mit der T-Kreuzung werden Abzweigungen hergestellt. Er kann auch anstelle eines Eck-Bricks verwendet werden.
	3	Art.-Nr.: 113634 Brick-ID: ALL-BRICK-0007	Kreuzung verbunden Dieser Brick verbindet alle vier Seiten miteinander. Er kann auch anstelle eines T-Bricks oder Eck-Bricks verwendet werden.

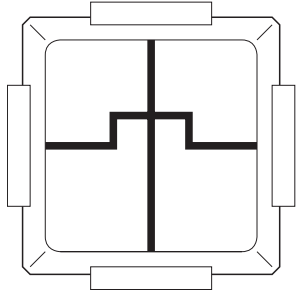
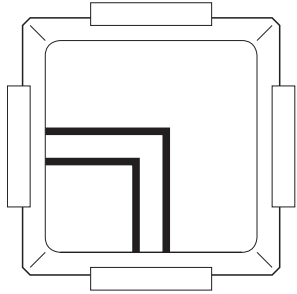
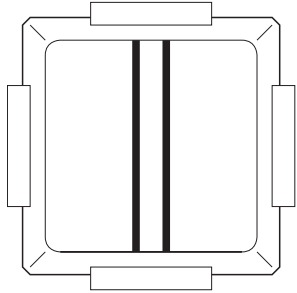
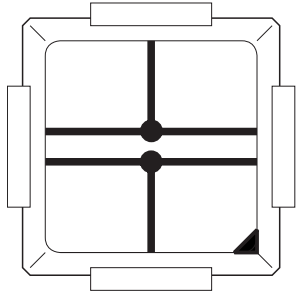
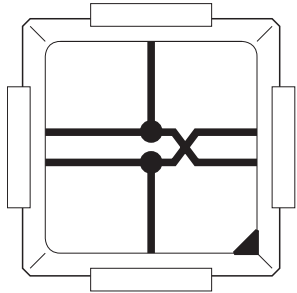
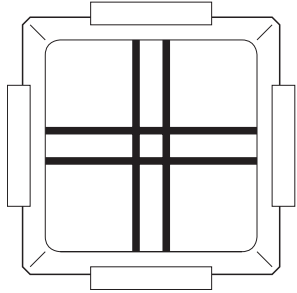
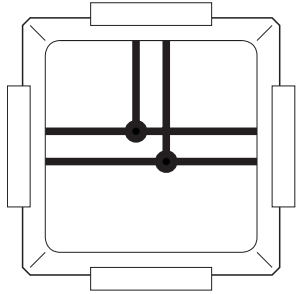
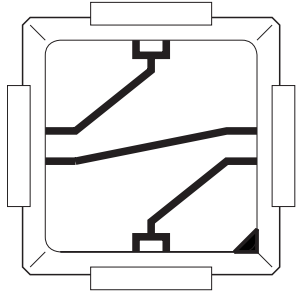
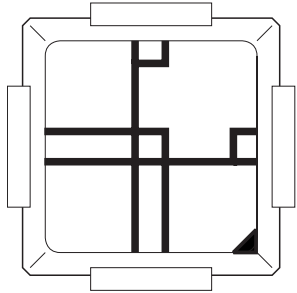
Abbildung	Anzahl	Art.-Nr. / Brick-ID	Kurzbeschreibung
	7	Art.-Nr.: 113635 Brick-ID: ALL-BRICK-0008	Kreuzung nicht verbunden Dieser Brick verbindet die jeweils gegenüberliegenden Anschlüsse. Links mit rechts und oben mit unten, ohne dass die beiden Leitungen verbunden sind.
	1	Art.-Nr.: 113681 Brick-ID: ALL-BRICK-0054	Leitung doppelt Ecke Eck-Brick für zwei getrennt geführte Signale.
	1	Art.-Nr.: 113676 Brick-ID: ALL-BRICK-0049	Leitung doppelt gerade Gerade Verbindung für zwei getrennt geführte Signale auf den beiden inneren Kontakten.
	2	Art.-Nr.: 122443 Brick-ID: ALL-BRICK-0381	Doppel-T-Kreuzung links/rechts Doppel-T-Kreuzung links/rechts zur Verwendung mit Gatter oder Flipflops.
	1	Art.-Nr.: 113675 Brick-ID: ALL-BRICK-0048	Leitung doppelt überkreuzt Mit diesem Baustein kann man die mittleren Leitungen getrennt weiterleiten und gleichzeitig kreuzen.

Abbildung	Anzahl	Art.-Nr. / Brick-ID	Kurzbeschreibung
	1	Art.-Nr.: 113678 Brick-ID: ALL-BRICK-0051	Doppel-Kreuzung nicht verbunden Dieser Brick ermöglicht die Kreuzung getrennt geführter Signale ohne dass diese verbunden werden.
	4	Art.-Nr.: 113679 Brick-ID: ALL-BRICK-0052	Doppel-T-Kreuzung links Mit der Doppel-T-Kreuzung links werden Abzweigungen getrennt geführter Signale hergestellt. Der linke Abzweig nach oben ist mit der oberen Leitung verbunden und der rechte Abzweig mit der unteren Leitung. Optional auch als Doppel-T-Kreuzung rechts erhältlich (Art.-Nr. 113680, Brick-ID: ALL-BRICK-0053).
	4	Art.-Nr.: 122442 Brick-ID: ALL-BRICK-0380	Doppel-Weiche links/rechts Die Leitungsführung dieses Bricks erlaubt die Verteilung getrennt geführter Signale und den Übergang von getrennter Signalführung auf Einzelleitung. Universell einsetzbar z. B. zur Taktzuführung an Flipflops oder Signalarückkopplungen.
	5	Art.-Nr.: 113677 Brick-ID: ALL-BRICK-0050	Leitung doppelt speziell Dieser Spezial-Leitungs-Brick erlaubt die Verteilung getrennt geführter Signale und den Übergang von getrennter Signalführung auf Einzelleitung. Typische Verwendung ist das Verschalten von Logik-Bricks.

4. Grundlagen der digitalen Schaltungstechnik

4.1 Was heißt eigentlich digital?

Der Begriff "Digital" kommt aus dem Lateinischen und heißt soviel wie Finger. Mit den Fingern hat man schon als Kind gezählt und gerechnet. Sie dienen quasi als elementares Rechenwerk. Wir beschränken uns auf die binäre Digitaltechnik, welche nur zwei mögliche Signalzustände kennt:

- Logisch Null "0" oder "L" (englisch low), und
- Logisch Eins "1" oder "H" (englisch high)

Im Gegensatz dazu steht der Begriff "Analog" (z. B. Analogkamera versus Digitalkamera). Den Unterschied kann man gut mit Hilfe eines Vergleichs zwischen einer Schräge und einer Treppe erklären. Die Höhenänderung an einer Schräge findet kontinuierlich statt, die an einer Treppe in kleinen diskreten Stufen. So wie die digitale Informationsverarbeitung nur zwischen logisch Null "0" und Eins "1" unterscheidet, werden in der Analogtechnik Signale wertkontinuierlich verarbeitet.

Die Hauptkomponenten digitaler Schaltungen sind Logikgatter wie NOT, AND und OR. Mit diesen Logikfunktionen lassen sich alle anderen Gatter, Zähler, Flipflops, etc. aufbauen. Komplexere Schaltungen sind Speicherschaltkreise und Prozessoren sowie frei programmierbare Logik-Bausteine.

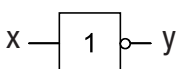
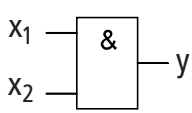
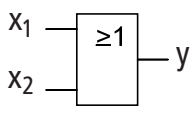
Wenn ein Spannungs-High-Pegel mit "1" und ein Low-Pegel mit "0" dargestellt wird, spricht man von positiver Logik, wenn ein Spannungs-High-Pegel mit "0" und ein Low-Pegel mit "1" dargestellt wird, spricht man von negativer Logik.

4.2 Logische Funktionen

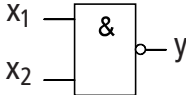
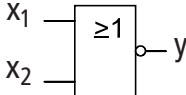
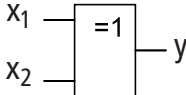
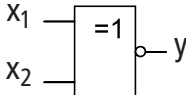
Logische Funktionen (auch Boolesche Funktionen genannt), werden auf Hardware-Ebene als Logikgatter oder einfach nur Gatter bezeichnet. Es gibt die drei logischen Grundfunktionen NOT, AND und OR aus denen die Funktionen NAND, NOR, XOR und XNOR zusammengesetzt werden können. Jedes Logikelement lässt sich durch eine Schaltfunktion beschreiben. Übliche Darstellungsformen sind die Boolesche Gleichung und die Wahrheitstabelle.

Enthält eine digitale Schaltung lediglich Logikelemente ohne Rückkopplung von Ausgängen auf Eingänge, so spricht man von einer rein kombinatorischen Logik oder auch **Schaltnetz** genannt (siehe auch Kap. 4.5 auf Seite 21). Neben logischen Funktionen können digitale Schaltungen aber auch speichernde Elemente wie Flipflops enthalten, die takt- oder zustandsgesteuert arbeiten. Sobald mindestens ein Ausgang auf einen Eingang zurückgekoppelt wird, spricht man von Schaltwerk oder auch Automat genannt (siehe auch 4.6 auf Seite 21).

Grundfunktionen

Schalt-Symbol	Bezeichnung	Gleichung	Wahrheitstabelle															
	NOT (Negation)	$y = \bar{x}$	<table border="1"> <thead> <tr> <th>x</th> <th>y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	x	y	0	1	1	0									
x	y																	
0	1																	
1	0																	
	AND (Konjunktion)	$y = x_1 \wedge x_2$	<table border="1"> <thead> <tr> <th>x₁</th> <th>x₂</th> <th>y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	x ₁	x ₂	y	0	0	0	0	1	0	1	0	0	1	1	1
x ₁	x ₂	y																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
	OR (Disjunktion)	$y = x_1 \vee x_2$	<table border="1"> <thead> <tr> <th>x₁</th> <th>x₂</th> <th>y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	x ₁	x ₂	y	0	0	0	0	1	1	1	0	1	1	1	1
x ₁	x ₂	y																
0	0	0																
0	1	1																
1	0	1																
1	1	1																

Zusammengesetzte Funktionen

Schalt-Symbol	Bezeichnung	Gleichung	Wahrheitstabelle															
	NAND (Exclusion)	$y = \overline{x_1 \wedge x_2}$	<table border="1"> <thead> <tr> <th>x_1</th> <th>x_2</th> <th>y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	x_1	x_2	y	0	0	1	0	1	1	1	0	1	1	1	0
x_1	x_2	y																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
	NOR (Nihilation)	$y = \overline{x_1 \vee x_2}$	<table border="1"> <thead> <tr> <th>x_1</th> <th>x_2</th> <th>y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	x_1	x_2	y	0	0	1	0	1	0	1	0	0	1	1	0
x_1	x_2	y																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
	XOR (Antivalenz)	$y = (\overline{x_1} \wedge x_2) \vee (x_1 \wedge \overline{x_2})$	<table border="1"> <thead> <tr> <th>x_1</th> <th>x_2</th> <th>y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	x_1	x_2	y	0	0	0	0	1	1	1	0	1	1	1	0
x_1	x_2	y																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
	XNOR (Äquivalenz)	$y = (x_1 \wedge x_2) \vee (\overline{x_1} \wedge \overline{x_2})$	<table border="1"> <thead> <tr> <th>x_1</th> <th>x_2</th> <th>y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	x_1	x_2	y	0	0	1	0	1	0	1	0	0	1	1	1
x_1	x_2	y																
0	0	1																
0	1	0																
1	0	0																
1	1	1																

Hinweis: Die Darstellung der Schaltsymbole entspricht IEC 60617-12.

4.3 Zahlensysteme im Vergleich

Wertigkeit der Stellen	Binärzahl				Dezimalzahl		Hexadezimalzahl*
	8 (2 ³)	4 (2 ²)	2 (2 ¹)	1 (2 ⁰)	10 (10 ¹)	1 (10 ⁰)	1 (16 ⁰)
Null	0	0	0	0	0	0	0
Eins	0	0	0	1	0	1	1
Zwei	0	0	1	0	0	2	2
Drei	0	0	1	1	0	3	3
Vier	0	1	0	0	0	4	4
Fünf	0	1	0	1	0	5	5
Sechs	0	1	1	0	0	6	6
Sieben	0	1	1	1	0	7	7
Acht	1	0	0	0	0	8	8
Neun	1	0	0	1	0	9	9
Zehn	1	0	1	0	1	0	A
Elf	1	0	1	1	1	1	B
Zwölf	1	1	0	0	1	2	C
Dreizehn	1	1	0	1	1	3	D
Vierzehn	1	1	1	0	1	4	E
Fünfzehn	1	1	1	1	1	5	F

Das jeweilige Zahlensystem wird durch Angabe der dem Zahlensystem zugrunde liegenden Basis gekennzeichnet. Dies erfolgt durch Tiefstellung der in runde Klammern gesetzten Basis, z. B. $1100_{(2)} = 12_{(10)} = C_{(16)}$.

Neben dem Binärsystem mit der Basis 2, wird in der Digitaltechnik häufig auch das Hexadezimalsystem* mit der Basis 16 verwendet. Dies hat den Hintergrund, dass Binärzahlen relativ lang und schwer zu überschauen sind. Da 16 eine Potenz von 2 ist, ist es besonders einfach möglich, Binärzahlen in Hexadezimalzahlen umzurechnen. Dazu werden je vier Stellen der Binärzahl durch eine Hexadezimalstelle ersetzt, was die Übersichtlichkeit deutlich steigert. Die Hexadezimalziffern mit dem Wert 0-9 werden durch die Ziffernsymbole 0-9 und die Werte 10-15 durch die Großbuchstaben A-F dargestellt. Dadurch sind sie relativ gut lesbar.

So lässt sich zum Beispiel leicht feststellen, dass $ADF5_{(16)}$ größer ist als $ABF1_{(16)}$ während sich die entsprechenden Binärzahlen $1010110111110101_{(2)}$ und $1010101111110001_{(2)}$ nicht so schnell überblicken lassen.

Umrechnung vom Binärsystem ins Dezimalsystem

Um eine Binärzahl in die entsprechende Dezimalzahl umzurechnen, werden alle Ziffern jeweils mit ihrer Wertigkeit (entsprechende Zweierpotenz) multipliziert und dann addiert.

Beispiel:

$$1100_{(2)} = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 8 + 4 + 0 + 0 = 12_{(10)}$$

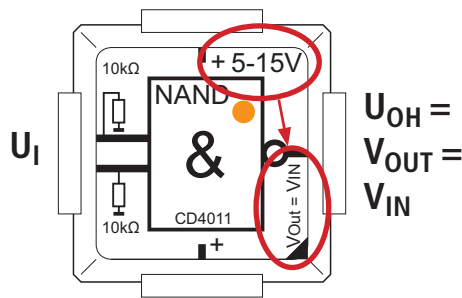
*Die lateinisch-griechische Wortkombination „Hexadezimal“ leitet sich von griechisch hexa „sechs“ und lateinisch decem „zehn“ ab. Der Bezeichnung "Hexadezimalzahl" ist übrigens gleichbedeutend mit "Sedezimalzahl", letztere wird jedoch in der Praxis nicht so häufig verwendet.

Siehe auch: <https://de.wikipedia.org/wiki/Hexadezimalsystem>.

4.4 Logikpegel

Um digitale Signale verarbeiten und anzeigen zu können, werden Spannungspegel definiert, die den logischen Zuständen entsprechen. Bei binär codierten Signalen reichen zwei Spannungsbereiche, welche die Information repräsentieren. Diese werden High-Pegel (auch H-Pegel, High oder H) und Low-Pegel (L-Pegel, Low oder L) genannt.

Der High-Pegel, also die höhere Spannung entspricht meist der Betriebsspannung (z. B. 9 Volt bei den Bricks); der Low-Pegel als niedrigere Spannung liegt idealerweise bei 0 Volt (oft auch Masse genannt). Der mehr oder weniger steile Übergang vom Low-Pegel zum High-Pegel wird auch als steigende oder positive Flanke bezeichnet und der Übergang vom High- zum Low-Pegel als fallende oder negative Flanke. Der Wechsel zwischen den beiden Pegeln muss mit einer Mindest-Flankensteilheit erfolgen, die in der Regel im Hersteller-Datenblatt definiert ist.



Bei unseren Logik-Bricks richtet sich der Ausgangspegel nach der Versorgungsspannung, die zwischen +5V und +15V variieren darf (Ausnahme BCD-Counter-Brick: 9-15V). Bei Verwendung des mitgelieferten Netzteiladapters oder bei Versorgung mit einer 9V-Blockbatterie sind dies 9V. Daraus folgt, dass der Ausgangs-High-Pegel U_{OH} – wie bei CMOS-Bausteinen üblich – annähernd der Versorgungsspannung entspricht.

Abb. 5: Ausgangspegel der Bricks

4.4.1 Spannungsbereiche

Die Spannungsbereiche werden durch die verschiedenen Logik-Familien vorgegeben. Als Logik-Familien bezeichnet man elektronische Schaltkreise (ICs), deren Logikpegel, zeitliches Verhalten, Treibereigenschaften, Herstellungsprozess etc. identisch sind.

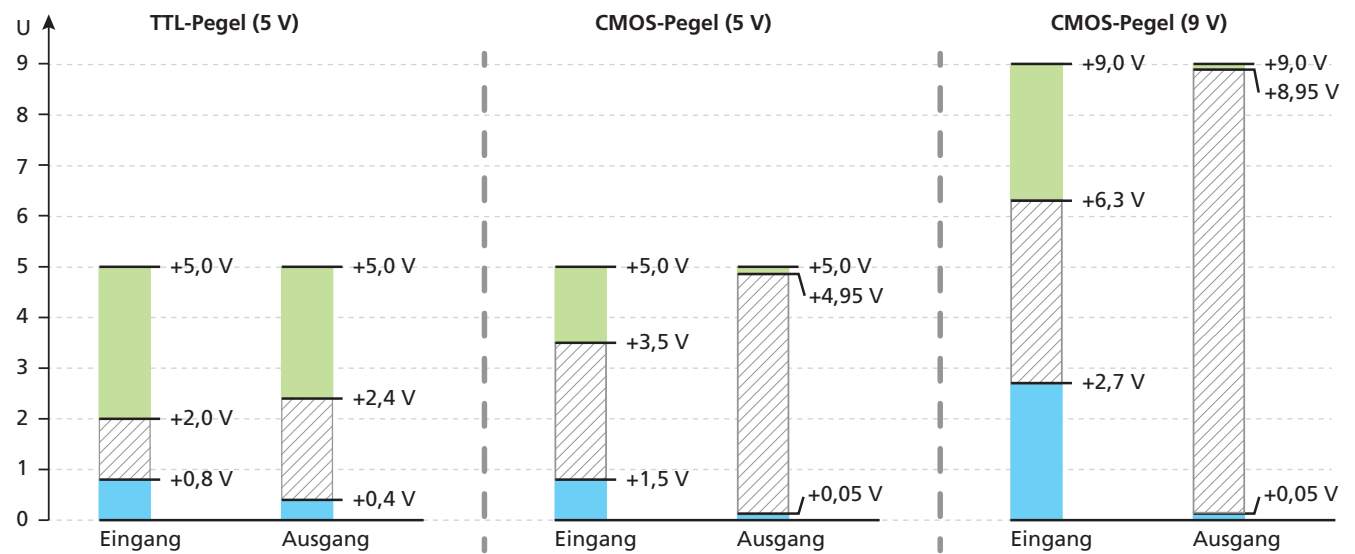


Abb. 6: Logikpegel für TTL- und CMOS-Technologie

Zur Darstellung der beiden Logikwerte dienen relativ große Pegelbereiche (in der Grafik blau und grün). Dies ist sinnvoll damit reale Logikschaltungen die Zustände trotz Toleranzen sicher erkennen können. Der Bereich zwischen beiden Eingangs-Pegelbereichen eines Logikgatters, also zwischen U_{IL} und U_{IH} ist nicht zulässig (verbotener Bereich), die Signalspannung ist dort nicht eindeutig einem Logikwert zuzuordnen (in der Grafik schraffiert). Deshalb wird für High-Pegel ausgangsseitig eine minimale Ausgangsspannung U_{OH} garantiert und eingangsseitig eine minimale Eingangsspannung U_{IH} gefordert. Die Ausgangsspannung U_{OH} ist immer größer als die Eingangsspannung U_{IH} , die Differenz $U_{OH} - U_{IH}$ wird statischer Störabstand genannt und sorgt für die Betriebssicherheit der Schaltungen. Bei Low-Pegel gibt es entsprechend eine maximale Ausgangsspannung U_{OL} , die maximale Eingangsspannung U_{IL} und den statischen Störabstand $U_{IL} - U_{OL}$.

4.4.2 Logische Zustände

Die Pegelangaben L und H dürfen nicht mit den logischen Zuständen 0 und 1 verwechselt werden. Die Angaben L und H geben stets den realen Spannungspegel an. Zum Beispiel 0 V (Low) oder 5 V (High). Mit diesen Pegelangaben beziehen wir uns auf die realen elektrischen Spannungspegel einer Schaltung. Will man die logische Arbeitsweise einer Schaltung beschreiben, so müssen die Pegelangaben den logischen Zuständen zugeordnet werden. Man unterscheidet in diesem Zusammenhang positive und negative Logik.

Positive Logik

Bei Verwendung der positiven Logik entspricht die logische 0 dem Low-Pegel und die logische 1 dem High-Pegel.

Negative Logik

Bei der Verwendung der negativen Logik entspricht die logische 0 dem High-Pegel und die logische 1 dem Low-Pegel.

Bezeichnungen low-aktiver Signale werden üblicherweise mit einer Überstreichung versehen. Alternativ werden Sternchen oder Schrägstriche voran- oder hintenangestellt. Die Schreibweisen $\overline{\text{RESET}}$, *RESET sowie /RESET sollen alle andeuten, dass das Signal RESET low-aktiv ist.

4.5 Schaltnetz

Der Begriff Schaltnetz bezeichnet eine kombinatorische Schaltung, die aus einfachen Grundgattern (wie z. B. AND, OR, NOT) besteht. Ein oder mehrere Ausgangsvariablen hängen zu jedem beliebigen Zeitpunkt von einer oder mehreren Eingangsvariablen ab. Es gibt keine Rückkopplungen, d. h. der Ausgang eines Gatters wird nicht an den Eingang desselben rückgeführt. Man könnte auch sagen, eine solche Schaltung ist ein vergessliches Verfahren. Signallaufzeiten werden auf der logischen Ebene vernachlässigt (siehe auch Kap. „Glitches“ auf Seite 22).

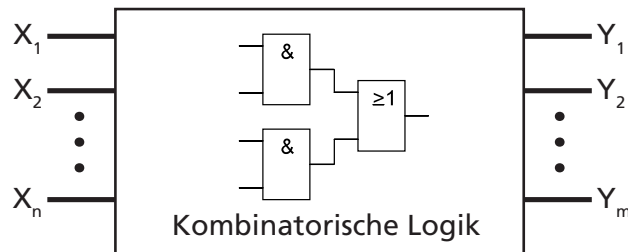


Abb. 7: Schaltnetz

4.6 Schaltwerk

Um Aufgaben zu lösen, die nicht nur von einer Momentaufnahme abhängig sind, benötigt man Schaltungen mit "Gedächtnis". Also eine Schaltung, die Eingangsvariablen zu einem bestimmten Zeitpunkt mit vor diesem Zeitpunkt entstandenen Werten verknüpft. Durch die Rückkopplung der Ausgänge auf die Eingänge, bekommt die Schaltung einen speichernden Charakter. Wir bekommen ein sog. Schaltwerk oder endlichen Zustandsautomaten.

Ein solches Schaltwerk besteht aus einem Speicherblock (Flipflop) und einem Block mit kombinatorischer Logik. Aus dem aktuellen Zustand des Speicherblocks und den Eingangsvariablen X bildet der kombinatorische Schaltungsteil den gewünschten Folgezustand Z_{t+1} und die Ausgangsvariablen Y_1 . Dieser Teil wird auch Mealy-Automat genannt. Während sich die Ausgangsvariablen Y_1 innerhalb eines Zustandes ändern können, sind die Ausgänge Y_2 getaktet und daher frei von möglichen Fehlimpulsen (auch Moore-Automat genannt).

Ein Schaltwerk heißt „synchron“ wenn die Eingänge und Rückkopplungen durch Taktsignale synchronisiert sind (siehe rechtes Bild), andernfalls heißt es „asynchron“ (siehe linkes Bild).

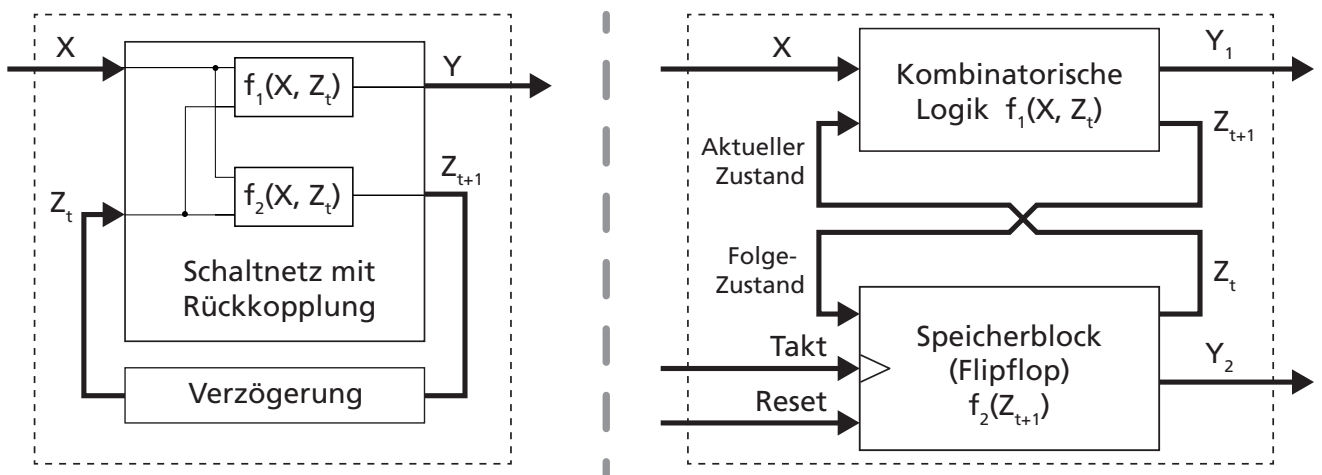


Abb. 8: Asynchron-Schaltwerk (links) und Synchron-Schaltwerk (rechts)

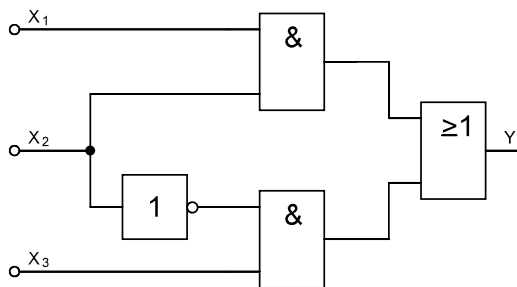
In der Realität erzeugt jedes Schaltnetz sein Ergebnis erst nach einer gewissen Zeit, die von seinem Aufbau abhängt. Um also zu einem bestimmten Zeitpunkt anliegende Eingangswerte mit den richtigen zwischengespeicherten Werten zu verknüpfen, werden die Eingangssignale und die rückgekoppelten Ausgänge mit Hilfe von Flipflops und einem Taktsignal synchronisiert.

Die Frequenz des Taktsignals muß dabei so groß gewählt werden, dass in der Zwischenzeit alle Schaltnetze im Schaltwerk ihre Berechnungen abschließen können, d. h. alle beteiligten Gatterlaufzeiten verstrichen sind.

4.7 Glitches

In der Elektronik bezeichnet man mit Glitch eine kurzzeitige Falschaussage in logischen Schaltungen und temporäre Verfälschung einer booleschen Funktion. Diese tritt auf, weil die Signallaufzeiten der einzelnen Gatter in der Realität niemals vollkommen gleich sind. Die Anfälligkeit für Glitches steigt mit der Komplexität, höheren Taktraten und der Miniaturisierung der Schaltungen, kann aber auch bei sehr einfachen Schaltungen vorkommen. Sie stellen ein wesentliches Problem bei der Entwicklung moderner elektronischer Schaltungen und schneller Mikroprozessoren dar.

Beispiel für die Entstehung von Glitches

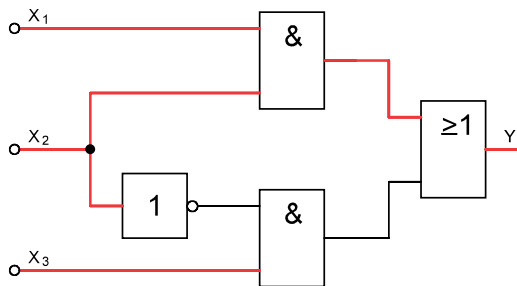


Die Schaltung

Es sei eine Schaltung gegeben, die drei Eingänge besitzt: x_1 , x_2 und x_3 . Der Ausgang y soll den Wert „1“ liefern, wenn mindestens eine der beiden Bedingungen erfüllt ist:

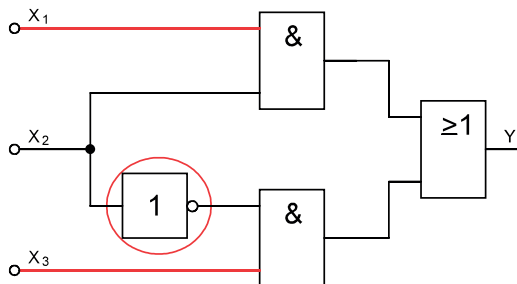
- x_2 und x_1 sind gleichzeitig „1“ ODER
- x_2 ist gleich „0“ und x_3 gleichzeitig „1“

Trifft nicht wenigstens eine der beiden Bedingungen zu, soll y den Wert „0“ ausgeben.



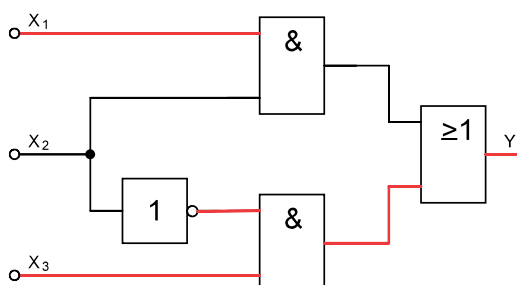
Zustand 1 - Die Schaltung liefert wie gewünscht eine 1

Die Schaltung befinde sich jetzt in Zustand 1. Laut unseren Vorgaben ist die erste Bedingung erfüllt, nämlich x_2 und x_1 sind „1“. Die Verzweigungen, die die Information „1“ tragen, sind rot dargestellt. Der Inverter wandelt die eingehende „1“ in eine „0“ um. Daher lässt die nachfolgende UND-Verknüpfung kein Signal mehr durch, gibt also eine „0“ aus. Die gesamte Schaltung (ODER-Verknüpfung) liefert aber dennoch eine „1“, da die andere UND-Verknüpfung die „1“ liefert.



Zustand 2 - Inverter verursacht einen Glitch

In Zustand 2 soll $x_2 = 0$ und $x_1 = 1$ sein. Die Schaltung soll weiterhin eine „1“ ausgeben. Der Inverter benötigt allerdings eine gewisse Zeit, um die Umwandlung des x_2 -Signals von „0“ in eine „1“ wahrzunehmen. Für kurze Zeit ist sowohl $x_2 = 0$, als auch $\bar{x}_2 = 0$. Dieser Umstand wird so verarbeitet, als ob keine der Bedingungen erfüllt ist und gibt folglich eine „0“ aus. Diese Situation bezeichnet man als Glitch.



Zustand 3 - Die Schaltung gibt wieder den korrekten Wert aus

Nach einiger Zeit – in der Größenordnung von Nanosekunden – befindet sich die Schaltung in Zustand 3: der Inverter hat die neue Information verarbeitet. Die jetzt ausgegebene „1“ läuft in das UND-Gatter, welches (wieder nach kurzer Verzögerung) dann auch eine „1“ liefert. Nun gibt die ODER-Verknüpfung bei y die gewünschte „1“ aus.

Abb. 9: Entstehung von Glitches

In der Praxis existieren Laufzeitunterschiede auch in Gattern desselben Typs oder in unterschiedlich langen Leitungen. Möchte man den exakten Wert der Funktion wissen, muss man eine entsprechende Zeit warten bis alle Signale eingeschwungen sind. Diese Tatsache beschränkt wesentlich die Taktfrequenz moderner Prozessoren.

Die Auswirkungen von Glitches können in synchronen Schaltungsdesigns durch nachgeschaltete D-Flipflops verhindert werden. Die Idee dabei ist, dass die Ausgänge der kombinatorischen Schaltungsteile, bestehend aus diversen Gattern unterschiedlicher Laufzeit, erst dann gültige Zustände annehmen müssen, wenn die Taktflanke die Ausgangswerte in die D-Flipflops übernimmt. In der Zeit zwischen zwei Taktflanken können im kombinatorischen Teil durch Laufzeiteffekte beliebig viele Glitches auftreten, da diese Zwischenzustände nicht durch das nachgeschaltete D-Flipflop beachtet werden.

Das Verfahren, die Ausgänge von kombinatorischen Schaltungsteilen immer mit D-Flipflops zu versehen, ist eines der wesentlichen Grundlagen für stabile, digitale Schaltungsdesigns in komplexen, freiprogrammierbaren FPGAs (siehe Kap. 4.8.3 auf Seite 23).

4.8 Programmierbare Logikbausteine

Herkömmliche, nicht-programmierbare Logik-Bausteine haben eine fest definierte Funktion. Ihr könnt diese Bausteine als integrierte Schaltkreise (Integrated Circuit = IC) kaufen. Diese Art Bausteine sind auch in den Logik-Bricks verbaut. Einer der bekanntesten Logik-Bausteine ist das NAND-Gatter mit der Typenbezeichnung 7400 (TTL-Technologie) bzw. 4011 (CMOS-Technologie). Im Gegensatz dazu erhalten PLDs (Programmable Logic Devices), also programmierbare Logik-Bausteine ihre Funktion erst nach der Herstellung durch entsprechende Programmierung des Schaltungsentwicklers (auch Personalisierung genannt). Manche Chip-Hersteller stellen kostenfreie Software-Tools für Schaltungsentwurf, Implementierung (Place & Route) und Simulation zur Verfügung.

Diejenigen, die sich etwas tiefer mit diesem spannenden Thema – insbesondere den FPGA-Bausteinen – beschäftigen möchten, finden im Folgenden ein grobes Überblick (in aufsteigender Komplexität):

4.8.1 Programmable Array Logic (PAL) bzw. Generic Array Logic (GAL)

Ein PAL stellt ein programmierbares UND-Array mit einem festen ODER-Array dar. Ein GAL ist im Gegensatz zu einem PAL wiederbeschreibbar. Die Personalisierung erfolgt mit Hilfe eines Programmiergeräts durch den Schaltungsentwickler. Technologischer Nachfolger ist der CPLD (siehe nächstes Kapitel).

4.8.2 Complex Programmable Logic Device (CPLD)

Ein CPLD besteht aus Blöcken, die untereinander verbunden werden können. Zentrales Element ist die sogenannte programmierbare logische Anordnung (PLA), die aus einem UND-Array und einem ODER-Array besteht, wobei beide Arrays programmierbar sind. Außerdem gibt es Ein- und Ausgangsblöcke sowie eine programmierbare Rückkopplung. In der Regel ist für jeden I/O-Pin auch ein Flipflop enthalten. Diese UND/ODER-Matrix erlaubt beliebige kombinatorische Verknüpfungen. Aufgrund seines Aufbaus können beim CPLD – im Gegensatz zum FPGA – die Durchlaufzeiten stets exakt bestimmt werden. Ein weiterer Unterschied zum FPGA ist die dauerhafte Programmierung durch den Schaltungsentwickler, sodass die Funktionalität auch nach Abschalten der Versorgung erhalten bleibt. Der CPLD muss nicht bei jedem Start des Geräts neu geladen werden.

4.8.3 Field Programmable Gate Array (FPGA)

Ein FPGA besteht ähnlich wie ein CPLD aus untereinander vernetzten Blöcken, jedoch sind diese komplexer. Auch die Möglichkeiten, diese Blöcke untereinander zu verbinden, sind gegenüber dem CPLD stark erweitert. Durch spezifische Konfiguration der intern vorhandenen Elemente können in einem FPGA völlig verschiedene Schaltungen und Funktionen realisiert werden. Diese reichen von Schaltungen geringer Komplexität, wie z. B. einem einfachen Synchronzähler oder Decoder, bis hin zu hochkomplexen Schaltungen wie Speicher-Controller und digitaler Signalverarbeitung. Moderne FPGAs enthalten oft auch integrierte Funktionsblöcke wie RAM, PLLs oder ganze CPU-Kerne.

FPGAs werden in allen Bereichen der Digitaltechnik eingesetzt, vor allem aber dort, wo es auf schnelle Signalverarbeitung und hohe Flexibilität bei Änderungen ankommt. So sind auch nachträglich Änderungen oder Verbesserungen an der implementierten Schaltung möglich, ohne teure und zeitintensive Hardware-Änderungen vornehmen zu müssen. Diese hohe Flexibilität ist ideal für Prototypen und die kostengünstige Fertigung von kleinen und mittleren Serien.

Die Personalisierung erfolgt "im Feld" beim Anwender entweder durch Auslesen der Konfigurationsdaten aus einem externen nichtflüchtigen Speicher wie z. B. einem PROM (Programmable Read Only Memory) – in diesem Fall ist der FPGA "Master" – oder durch Download der Daten vom Rechner in den FPGA (Slave-Modus). Nach dem Einschalten der Versorgungsspannung ist der FPGA zunächst "dumm". Erst nach Abschluss der sogenannten Konfigurationsphase von einigen Millisekunden ist der Baustein betriebsbereit. FPGAs können in der Schaltung ohne Programmiergerät jederzeit umkonfiguriert werden.

4.8.3.1 FPGA-Grundstruktur

Die Grundstruktur von FPGAs stellt eine Matrix-Anordnung aus konfigurierbaren Logik-Blöcken (Configurable Logic Blocks = CLBs) sowie Ein-/Ausgangs-Blöcken (Input/Output Blocks = IOBs) dar. Letztere sind in der Regel mit den Pins am Chip verbunden. Siehe Abb. 10 auf Seite 24.

Zwischen den Blöcken (CLBs und IOBs) liegt ein Gitter aus Verbindungswegen, man spricht auch von der Interconnect Area bestehend aus einer Hierarchie horizontaler und vertikaler "Lines". An diese Leitungen werden die Ein- und Ausgänge der Blöcke angeschlossen. Entferntere Verbindungen werden über programmierbare Verbindungspunkte in den Schaltmatrizen (engl. Switch-Matrix) des Gitters "verdrahtet". Auf diese Weise ist die Signalführung (auch Routing genannt) über den gesamten Chip möglich.

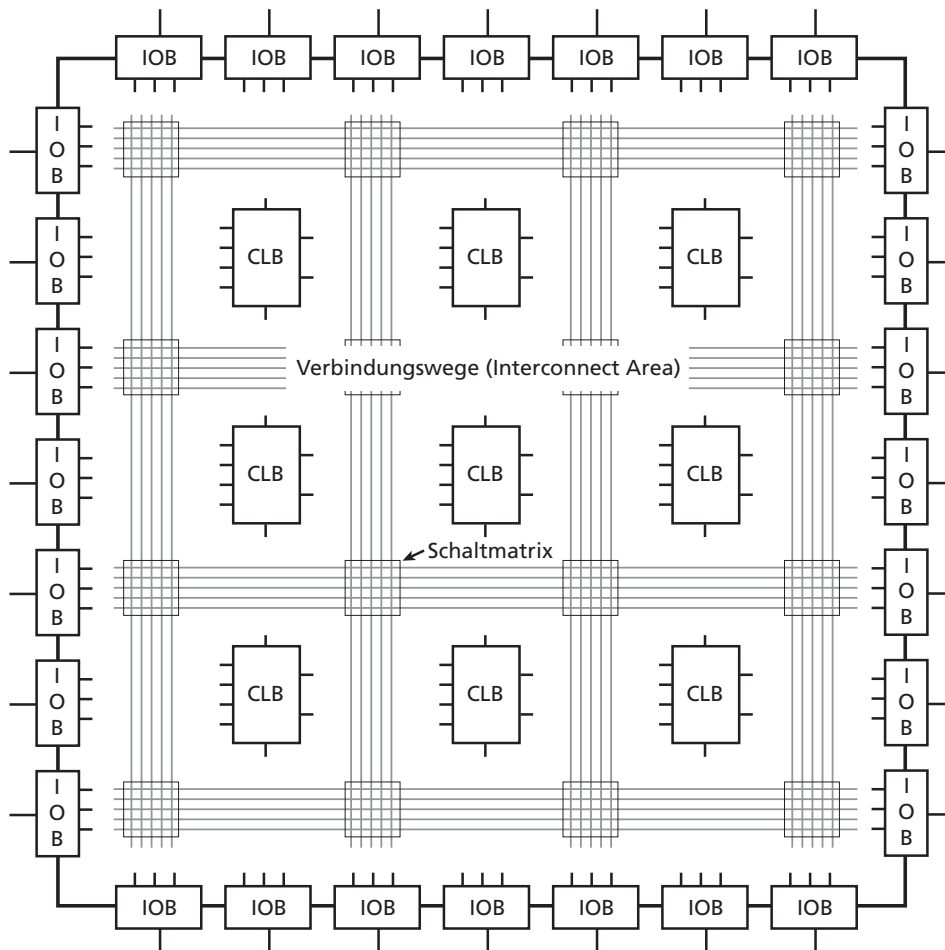


Abb. 10: FPGA Grundstruktur

4.8.3.2 Konfigurierbare Logik-Blöcke (CLBs)

Die Grundstruktur eines FPGAs ist ein Feld (engl. Array) aus konfigurierbaren Blöcken mit einer programmierbaren Wahrheitstabelle (engl. Lookup Table = LUT) und einem 1-Bit-Register (D-Flipflop). Die LUT kann, je nach Anzahl der verfügbaren Eingänge, jede beliebige n-stellige Binärfunktion realisieren. Üblich sind LUT-Strukturen mit vier binären Eingängen und mehr. Neben den LUTs ermöglichen Multiplexer in den Basisblöcken sehr schnelle lokale Signalpfade, zum Beispiel zur Einbindung oder Umgehung des Flipflops, zur Rückkopplung von dessen Ausgang, zur Verbindung von Nachbarblöcken und ähnlichem.

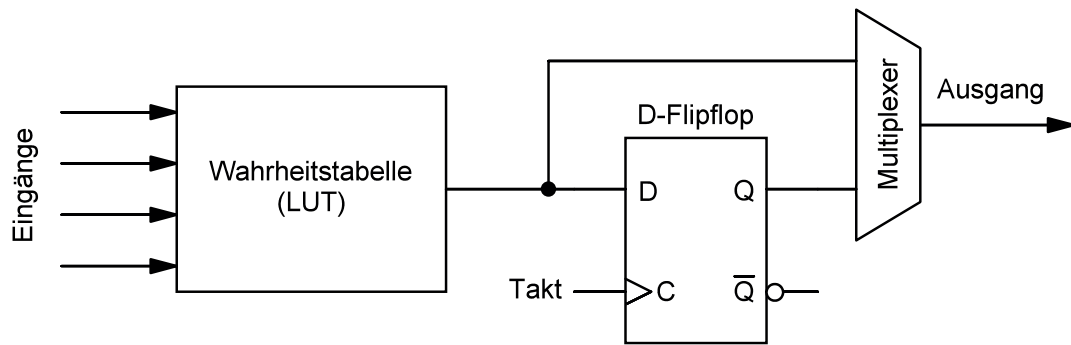


Abb. 11: Logik-Block Beispiel

4.8.3.3 Ein-/Ausgangs-Blöcke (IOBs)

Die Ein-/Ausgangs-Blöcke (IOBs) bilden die Schnittstelle zur Aussenwelt. Über sie werden die Anschlüsse des FPGA mit der Schaltmatrix verbunden. Je nach Anwendung kann der Spannungspegel der Ein- und Ausgänge an den jeweiligen Schnittstellen-Standard (TTL, LVDS, etc.) angepasst werden. Außerdem können der Treiberstrom der Ausgänge und die Flankensteilheit der Ausgangssignale definiert werden sowie Tristate-Puffer (Pins die hochohmig geschaltet werden können) zum Aufbau einer Bus-Architektur aktiviert werden.

5. Schaltungsbeispiele

5.1 Grundsaltungen

5.1.1 Inverter

Der Inverter-Brick bildet die NOT-Funktion ab und wird auch als Negation oder Invertierung bezeichnet. D. h. wir können das Eingangssignal invertieren. Legt man am Eingang des Inverter-Bricks High-Pegel an, erhalten wir am Ausgang Low-Pegel und umgekehrt.

In unserer Brick-Schaltung wird am Eingang des Inverter-Bricks ein Taster angeschlossen und mit der Versorgungsspannung verbunden. Damit der Inverter-Brick bei geöffnetem Taster (Ruhezustand) weiß, was er tun soll, hat er einen integrierten Pull-Down-Widerstand. Dieser bewirkt, dass der Eingang zunächst auf Low-Pegel liegt. Am Ausgang wird eine beliebiger LED-Brick angeschlossen um die Zustandsänderung anzuzeigen.

Durch Drücken des Tasters wird am Eingang High-Pegel angelegt, der Ausgang geht auf Low-Pegel und das LED geht aus.

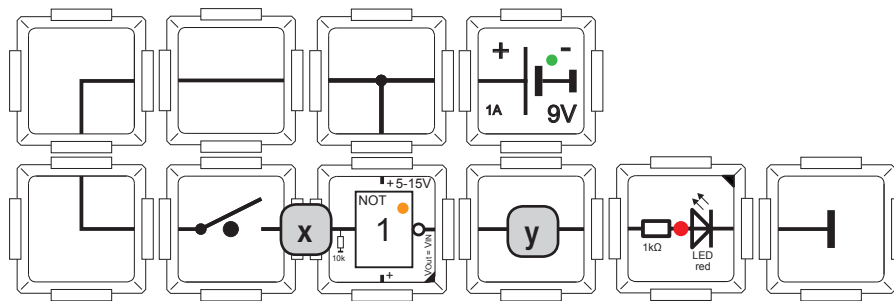


Abb. 12: Inverter als Brick-Schaltung

Schalt-Symbol	Bezeichnung	Gleichung	Wahrheitstabelle						
	NOT (Negation)	$y = \bar{x}$	<table border="1"> <thead> <tr> <th>x</th> <th>y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	x	y	0	1	1	0
x	y								
0	1								
1	0								

5.1.2 AND-Gatter

Das AND-Gatter realisiert eine logische UND-Verknüpfung von zwei oder mehreren Eingängen. Im Vergleich zum NAND-Gatter wird der Ausgang hier nicht invertiert.

In unserem Beispiel werden die beiden Eingänge des AND-Gatters über Taster mit der Versorgungsspannung verbunden. Solange kein Taster gedrückt wird, werden die beiden Eingänge über die integrierten Pull-Down-Widerstände auf Masse (Low-Pegel) gezogen, sodass an den Eingängen ein definierter Pegel anliegt. Die LED leuchtet erst, wenn beide Tasten gedrückt werden und der Ausgang gemäß Wahrheitstabelle auf High-Pegel wechselt.

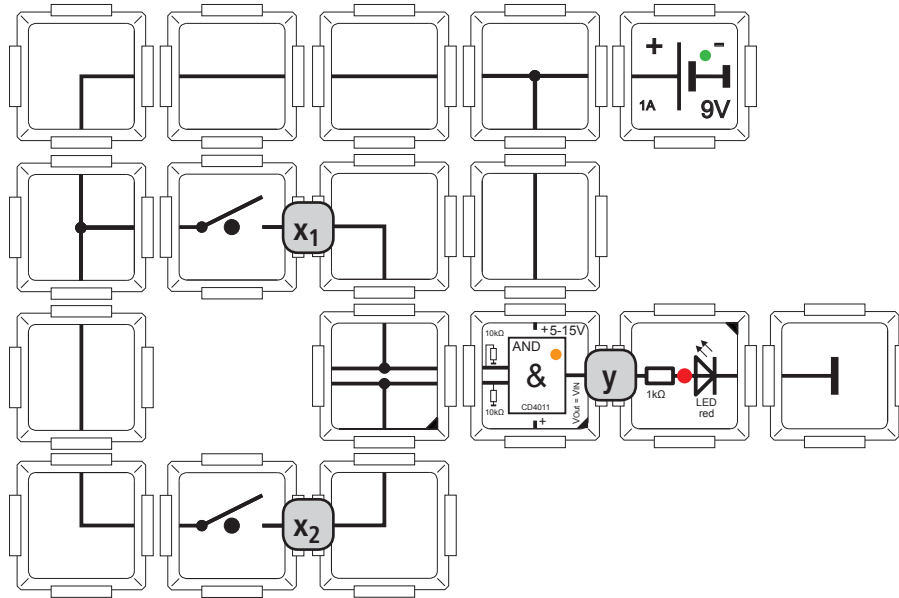


Abb. 13: AND-Gatter als Brick-Schaltung

Schalt-Symbol	Bezeichnung	Gleichung	Wahrheitstabelle															
	AND (Konjunktion)	$y = x_1 \wedge x_2$	<table border="1"> <thead> <tr> <th>x_1</th> <th>x_2</th> <th>y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	x_1	x_2	y	0	0	0	0	1	0	1	0	0	1	1	1
x_1	x_2	y																
0	0	0																
0	1	0																
1	0	0																
1	1	1																

Eine AND-Schaltung kann man auch ganz einfach aus zwei Taster bauen, die hintereinander in den Stromkreis geschaltet werden – man spricht auch von Reihenschaltung. Nur wenn beide Tasten (x_1 und x_2) gedrückt werden leuchtet die LED (y). Teste die Funktion anhand obiger Wahrheitstabelle.

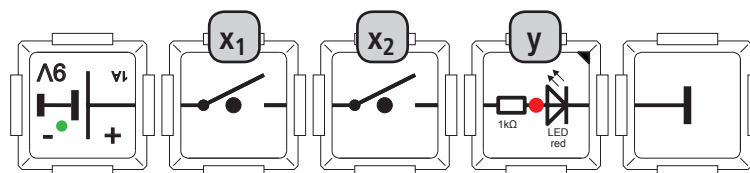


Abb. 14: Alternative AND-Schaltung mit Taster

5.1.3 OR-Gatter

Das OR-Gatter realisiert eine logische ODER-Verknüpfung von zwei oder mehreren Eingängen. Im Vergleich zum NOR-Gatter wird der Ausgang hier nicht invertiert.

In unserem Beispiel werden die beiden Eingänge des OR-Gatters über Taster mit der Versorgungsspannung verbunden. Solange kein Taster gedrückt wird, werden die beiden Eingänge über die integrierten Pull-Down-Widerstände auf Masse (Low-Pegel) gezogen, sodass an den Eingängen ein definierter Pegel anliegt. Die LED leuchtet erst, wenn mindestens eine der beiden Tasten gedrückt wird. Der Ausgang wechselt gemäß Wahrheitstabelle auf High-Pegel.

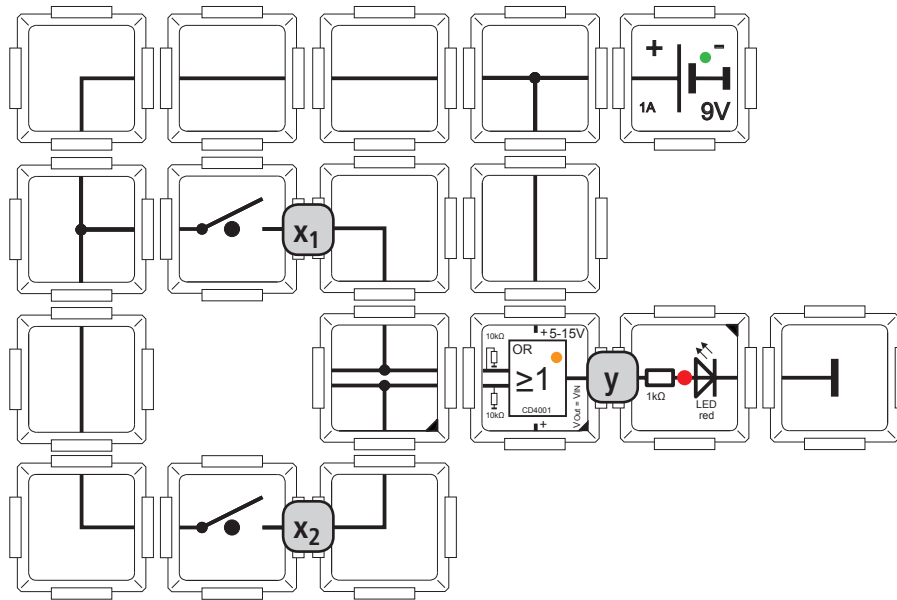


Abb. 15: OR-Gatter als Brick-Schaltung

Schalt-Symbol	Bezeichnung	Gleichung	Wahrheitstabelle															
	OR (Disjunktion)	$y = x_1 \vee x_2$	<table border="1"> <thead> <tr> <th>x_1</th> <th>x_2</th> <th>y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	x_1	x_2	y	0	0	0	0	1	1	1	0	1	1	1	1
x_1	x_2	y																
0	0	0																
0	1	1																
1	0	1																
1	1	1																

Eine OR-Schaltung kann man auch ganz einfach aus zwei Taster bauen, die parallel in den Stromkreis eingebaut werden – man spricht auch von Parallelschaltung. Sobald mindestens einer der beiden Taster (x_1 oder x_2) gedrückt wird, leuchtet die LED (y). Teste die Funktion anhand obiger Wahrheitstabelle.

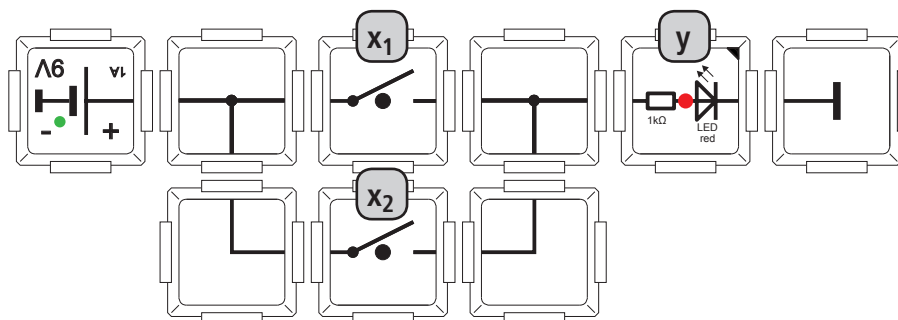


Abb. 16: Alternative OR-Schaltung mit Taster

5.1.4 NAND-Gatter

Das NAND-Gatter realisiert eine logische UND-Verknüpfung von zwei oder mehreren Eingängen mit invertierendem Ausgang. Neben dem NOR-Gatter ist es der am häufigsten verwendete Gattertyp. Formal logisch existiert keine Funktion, die nicht aus diesen beiden Typen zusammengesetzt werden kann. Dies macht deutlich, warum diese beiden Gattertypen in der Praxis sehr häufig vorkommen.

In unserem Beispiel werden die beiden Eingänge des NAND-Gatters über Taster mit der Versorgungsspannung verbunden. Solange kein Taster gedrückt wird, werden die beiden Eingänge über die integrierten Pull-Down-Widerstände auf Masse (Low-Pegel) gezogen. Die LED leuchtet, da aufgrund des invertierenden Ausgangs dieser einen High-Pegel ausgibt. Erst wenn beide Tasten gedrückt werden, geht der Ausgang auf Low-Pegel (siehe auch Wahrheitstabelle), die LED geht aus.

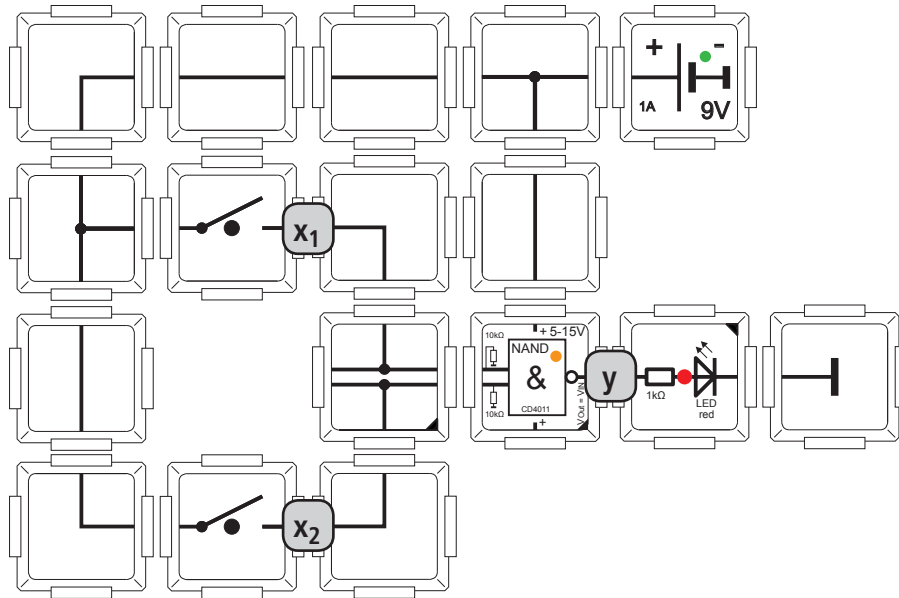


Abb. 17: NAND-Gatter als Brick-Schaltung

Schalt-Symbol	Bezeichnung	Gleichung	Wahrheitstabelle															
	NAND (Exclusion)	$y = \overline{x_1 \wedge x_2}$	<table border="1"> <thead> <tr> <th>x_1</th> <th>x_2</th> <th>y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	x_1	x_2	y	0	0	1	0	1	1	1	0	1	1	1	0
x_1	x_2	y																
0	0	1																
0	1	1																
1	0	1																
1	1	0																

Alternativ können Sie die NAND-Schaltung auch kompakter bauen – die Funktionalität ist die gleiche:

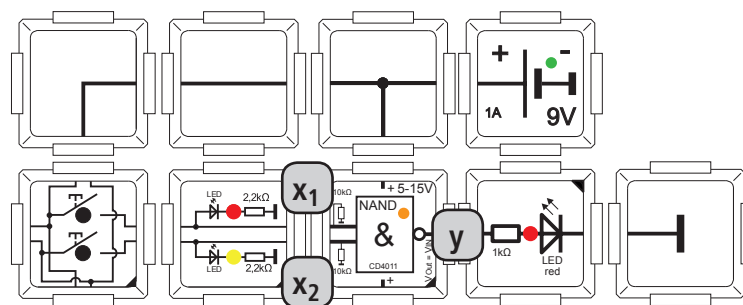


Abb. 18: Alternative NAND-Schaltung

5.1.5 NOR-Gatter

Das NOR-Gatter realisiert eine logische ODER-Verknüpfung von zwei oder mehreren Eingängen mit invertierendem Ausgang. Neben dem NAND-Gatter ist es der am häufigsten verwendete Gattertyp. Formal logisch existiert keine Funktion, die nicht aus diesen beiden Typen zusammengesetzt werden kann. Dies macht deutlich, warum diese beiden Gattertypen in der Praxis sehr häufig vorkommen.

In unserem Beispiel werden die beiden Eingänge des NOR-Gatters über Taster mit der Versorgungsspannung verbunden. Solange kein Taster gedrückt wird, werden die beiden Eingänge über die integrierten Pull-Down-Widerstände auf Masse (Low-Pegel) gezogen. Die LED leuchtet, da aufgrund des invertierenden Ausgangs dieser einen High-Pegel ausgibt. Sobald mindestens eine der beiden Tasten gedrückt wird, geht der Ausgang auf Low-Pegel (siehe auch Wahrheitstabelle), die LED geht aus.

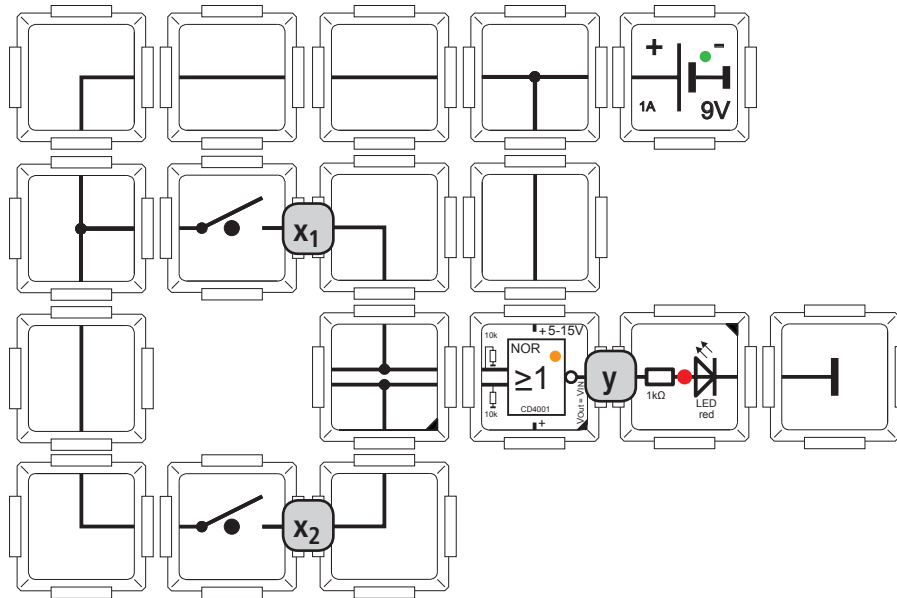


Abb. 19: NOR-Gatter als Brick-Schaltung

Schalt-Symbol	Bezeichnung	Gleichung	Wahrheitstabelle															
	NOR (Nihilation)	$y = \overline{x_1 \vee x_2}$	<table border="1"> <thead> <tr> <th>x₁</th> <th>x₂</th> <th>y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	x ₁	x ₂	y	0	0	1	0	1	0	1	0	0	1	1	0
x ₁	x ₂	y																
0	0	1																
0	1	0																
1	0	0																
1	1	0																

Alternativ können Sie die NOR-Schaltung auch kompakter bauen – die Funktionalität ist die gleiche:

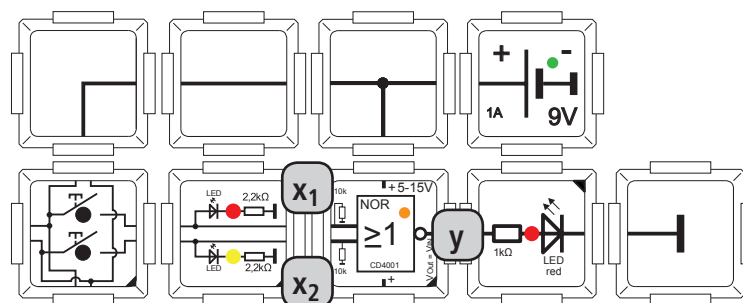


Abb. 20: Alternative NOR-Schaltung

5.1.6 XOR-Gatter

Das XOR-Gatter ist in der Elektrotechnik häufig anzutreffen. Es realisiert eine logische Exklusiv-ODER-Verknüpfung von zwei oder mehreren Eingängen. Ein einfaches Beispiel ist eine klassische Wechselschaltung, wie sie z. B. für das Flurlicht zuhause verwendet wird. Jeder Schalter kann zum Ein- oder Ausschalten verwendet werden, unabhängig welcher Schalter zuerst betätigt wird. Im allgemeinen Sprachgebrauch kann man die Exklusiv-Oder-Verknüpfung auch als "entweder, oder" beschreiben. Die XOR-Verknüpfung ist auch ein wichtiger Bestandteil des Addierers, den wir in Kap. 5.3 „Digitales Rechenwerk“ noch näher kennenlernen.

In unserem Beispiel werden die beiden Eingänge des XOR-Gatters über Taster mit der Versorgungsspannung verbunden. Solange kein Taster gedrückt wird, werden die beiden Eingänge über die integrierten Pull-Down-Widerstände auf Masse (Low-Pegel) gezogen. Die LED leuchtet nicht, da der Ausgang einen Low-Pegel ausgibt. Sobald "entweder" die eine "oder" die andere Taste gedrückt wird, geht der Ausgang auf High-Pegel und die LED leuchtet. Falls kein oder beide Taster gedrückt werden geht der Ausgang gemäß Wahrheitstabelle wieder auf Low-Pegel.

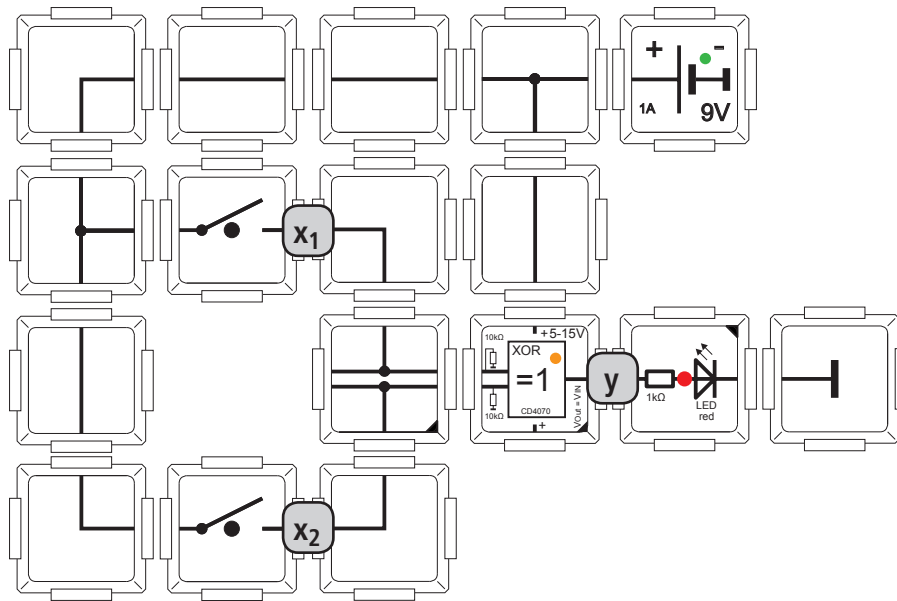


Abb. 21: XOR-Gatter als Brick-Schaltung

Schalt-Symbol	Bezeichnung	Gleichung	Wahrheitstabelle															
	XOR (Antivalenz)	$y = (\bar{x}_1 \wedge x_2) \vee (x_1 \wedge \bar{x}_2)$	<table border="1"> <thead> <tr> <th>x₁</th> <th>x₂</th> <th>y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	x ₁	x ₂	y	0	0	0	0	1	1	1	0	1	1	1	0
x ₁	x ₂	y																
0	0	0																
0	1	1																
1	0	1																
1	1	0																

Alternativ können Sie die XOR-Schaltung auch kompakter bauen – die Funktionalität ist die gleiche:

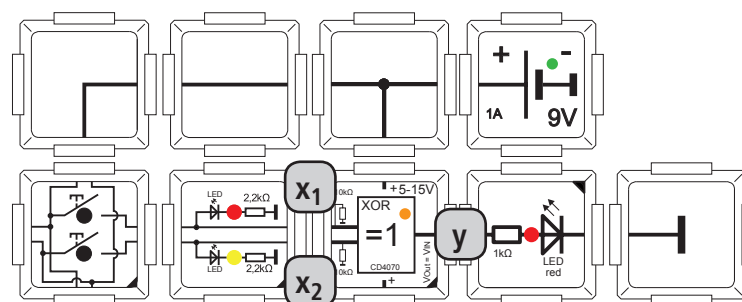


Abb. 22: Alternative XOR-Schaltung

5.1.7 XNOR-Gatter

Das XNOR-Gatter realisiert eine logische Exklusiv-ODER-Verknüpfung von zwei oder mehreren Eingängen. Neben NAND- und NOR-Gatter gehört es zu den Gattern mit invertierendem Ausgang. Die invertierende Exklusiv-ODER-Verknüpfung signalisiert uns die Gleichheit ihrer Eingänge. Dies kann z. B. zum Vergleich von zwei verschiedenen Bitmustern verwendet werden. Im Falle der Bitmuster-Gleichheit (beide Eingänge "0" oder beide Eingänge "1") kann dann ein Interruptsignal an den Computer geschickt werden oder einfach ein akustischer oder optischer Alarm ausgelöst werden.

In unserem Beispiel werden die beiden Eingänge des XNOR-Gatters über Taster mit der Versorgungsspannung verbunden. Solange kein Taster gedrückt wird, werden die beiden Eingänge über die integrierten Pull-Down-Widerstände auf Masse (Low-Pegel) gezogen. Dies heißt im Falle der XNOR-Verknüpfung, dass der Ausgang High-Pegel ausgibt. Unsere rote "Alarm"-LED leuchtet. Das gleiche gilt, wenn wir beide Taster gleichzeitig drücken. Sobald jedoch nur ein Taster gedrückt wird, geht der Ausgang wieder auf Low-Pegel und die LED geht aus.

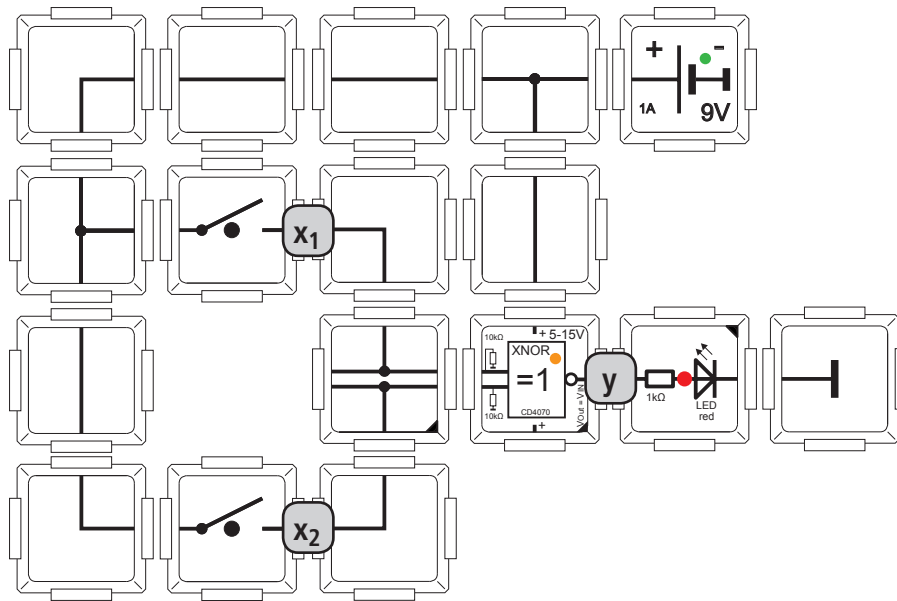


Abb. 23: XNOR-Gatter als Brick-Schaltung

Schalt-Symbol	Bezeichnung	Gleichung	Wahrheitstabelle															
	XNOR (Äquivalenz)	$y = (x_1 \wedge x_2) \vee (\overline{x_1} \wedge \overline{x_2})$	<table border="1"> <thead> <tr> <th>x_1</th> <th>x_2</th> <th>y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	x_1	x_2	y	0	0	1	0	1	0	1	0	0	1	1	1
x_1	x_2	y																
0	0	1																
0	1	0																
1	0	0																
1	1	1																

Alternativ können Sie die XNOR-Schaltung auch kompakter bauen – die Funktionalität ist die gleiche:

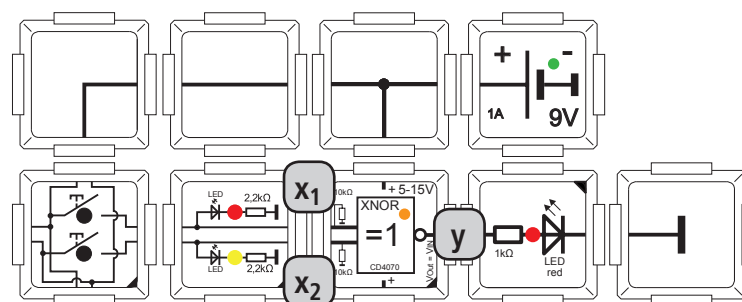


Abb. 24: Alternative XNOR-Schaltung

5.2 Entprellschaltung

Taster und Schalter haben den Nachteil, dass bei Betätigung der mechanische Kontakt (oft ist eine Feder im Spiel) ein mehrfaches Schließen und Öffnen verursacht. Diesen Störeffekt nennt man in der Digitaltechnik auch "Prellen".

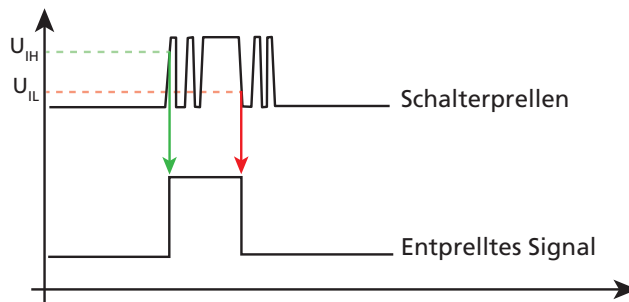


Abb. 25: Typischer Störimpuls durch Kontaktprellen

5.2.1 Entprellung auf Gatterebene

Das störende Prellen kann man mit einem einfachen RS-Flipflop beseitigen. Ein solches RS-Flipflop kann man auf Gatterebene wahlweise aus zwei NAND- oder NOR-Gatter aufbauen. Wir verwenden die Variante aus NOR-Gatter, deren Eingänge bereits mit Pull-Down-Widerständen ($10\text{k}\Omega$) bestückt sind. Wegen der hohen Schaltgeschwindigkeit des RS-Flipflops bleibt es beim ersten Kontakt stabil in diesem Zustand stehen und speichert den logischen Wert, bis der andere Eingang beschaltet wird. Die Pulldown-Widerstände bewirken, dass unbeschaltete Eingänge, stets einen definierten Zustand haben. Schalten Sie zwischen dem S(et)- und R(ese)-Eingang hin und her, um am Q- bzw. \bar{Q} -Ausgang einen entprellten Impuls zu erhalten.

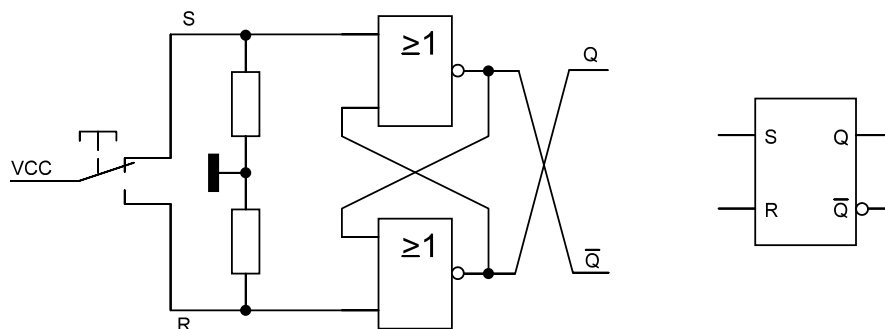


Abb. 26: Entprellschaltung aus NOR-Gatter (links) und RS-Flipflop (rechts)

Wie Sie mit dem 2x1 D-Flipflop-Brick eine Tastenentprellung bauen können erfahren Sie Kap. 5.4.3 auf Seite 41. Sie brauchen dazu nur den Set- und Reset-Eingang beschalten.

5.2.2 Bricks als entprellte Taktquelle

Wählen Sie je nach Anwendungsfall zwischen den Bricks "Taster entprellt" (engl.: "Debounced pushbutton") und Taktgeber (engl. "Clock" (generator)). Aus didaktischen Gründen ist der entprellte Taster oft vorteilhaft. So können z. B. die Einzelschritte eines Schieberegisters oder Zählers in Ruhe nachvollzogen werden. Im Falle des BCD-Zähler-Bricks ist möglicherweise der Clock-Brick vorteilhaft, der ein einstellbares Rechtecksignal von 0,5 Hz bis 100 Hz ausgibt. Dies ist ideal um beispielsweise das Übertragungssignal des BCD-Zähler-Bricks genauer zu studieren (siehe Kap. 5.6.4 auf Seite 54).

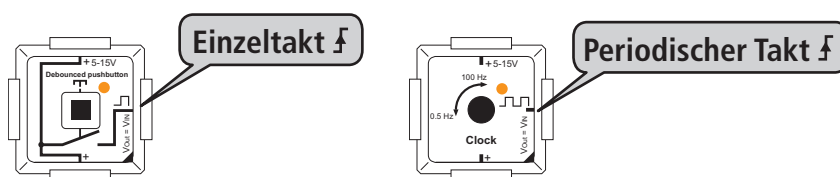


Abb. 27: Links: Entprellter Taster (ALL-BRICK-0641), rechts: Taktgeber (ALL-BRICK-0643)



Zur Verwendung der beiden Bricks beachten Sie bitte die Hinweise bei den einzelnen Übungen. In vielen Fällen können die beiden Bricks 1:1 gegeneinander getauscht werden. Beachten Sie beim Stecken stets die Ausrichtung des Bricks!

Um Fehlfunktionen zu vermeiden, an den Taktausgängen keine weitere Last, wie z. B. LEDs anschließen!

5.3 Digitales Rechenwerk

Aus den bereits bekannten Logikgattern lassen sich digitale Schaltungen zusammenbauen um Rechenoperationen im dualen Zahlensystem durchzuführen. Ein solches Rechenwerk wird im englischen Sprachgebrauch auch ALU (Arithmetic Logic Unit) genannt. Sie kann mit elementaren, arithmetischen und logischen Operationen zwei Eingangswerte miteinander verknüpfen. Wird nun die ALU mit einem Steuerwerk (Zustandsfolger) und einem Ergebnisregister erweitert erhält man eine sog. zentrale Verarbeitungseinheit besser bekannt unter dem englischen Begriff CPU (Central Processing Unit). Das Herzstück jeder CPU ist also eine ALU und deren elementarer Bestandteil wiederum der Addierer. Damit können im einfachsten Fall zwei einstellige Dualzahlen A und B addiert werden. Im ersten Schritt bauen wir einen 1-Bit Halbaddierer, den wir im zweiten Schritt zu einem Volladdierer mit Übertrag ausbauen.

5.3.1 1-Bit-Halbaddierer

Der 1-Bit Halbaddierer (HA) ist die einfachste Rechenschaltung und kann zwei einstellige Dualziffern addieren. Der Eingang A soll mit Eingang B addiert werden. Die Summe S wird mittels XOR-Verknüpfung generiert und der Übertrag (Ü) mit einer UND-Verknüpfung. Der Übertrag signalisiert, dass der Zahlenraum dieser Stelle durch die Addition überschritten wird. Dies ist prinzipiell die gleiche Vorgehensweise wie bei der schriftlichen Addition im Dezimalsystem. Vielleicht erinnern Sie sich: Sobald die Zehn erreicht ist, wird bei der nächsthöheren Stelle ein entsprechender Übertrag notiert.

Die beiden Tasten entsprechen den Summanden A und B welche addiert werden sollen. Durch drücken einer Taste wird der Binärwert "1" angelegt. Das rote Summen-LED am Ausgang des XOR-Gatters leuchtet ($1 + 0 = 1$). Sobald beide Tasten gedrückt werden, d.h. wenn wir $1 + 1$ rechnen, ergibt das im dualen System die Summe "0" und den Übertrag "1" (grüne LED leuchtet).

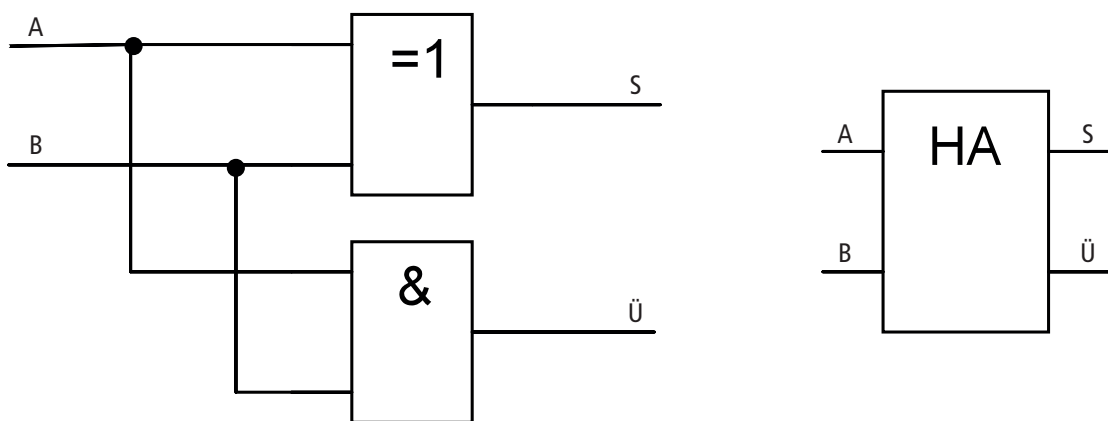


Abb. 28: Schaltung Halbaddierer (links) und Blockschaltbild (rechts)

Wahrheitstabelle 1-Bit-Halbaddierer

Summand A	Summand B	Summe S	Übertrag Ü
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

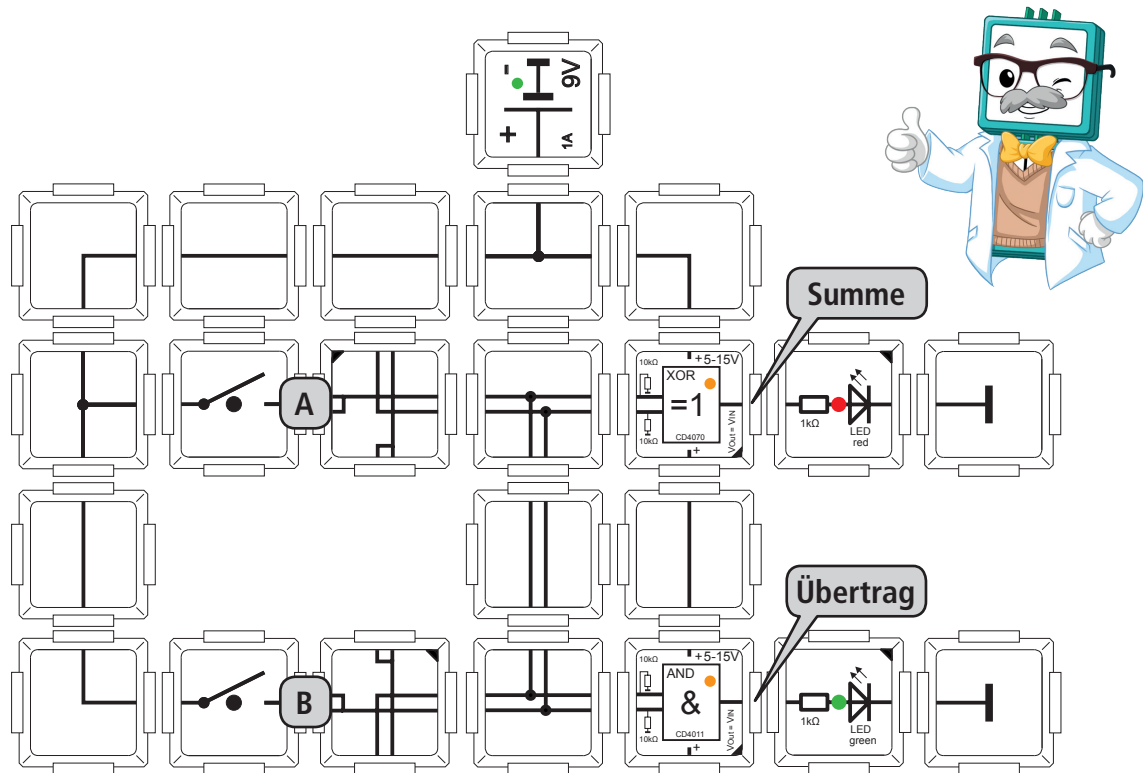


Abb. 29: 1-Bit Halbaddierer als Brick-Schaltung

5.3.2 1-Bit-Volladdierer

Möchte man mehrstellige Dual-Zahlen addieren, muss der Übertrag der jeweils niederwertigeren Stelle in die Summe S miteinbezogen werden. Wir bezeichnen diesen zusätzlichen Summanden als Übertragseingang \ddot{U}_E . Schaltungstechnisch kombinieren wir zwei Halbaddierer (HA) zu einem sog. Volladdierer (VA). Damit können wir zu den beiden Summanden A und B zusätzlich den Übertrag \ddot{U}_E addieren. Das Ergebnis wird als Summe S (rote LED) und als Übertragsausgang \ddot{U}_A (grüne LED) angezeigt.

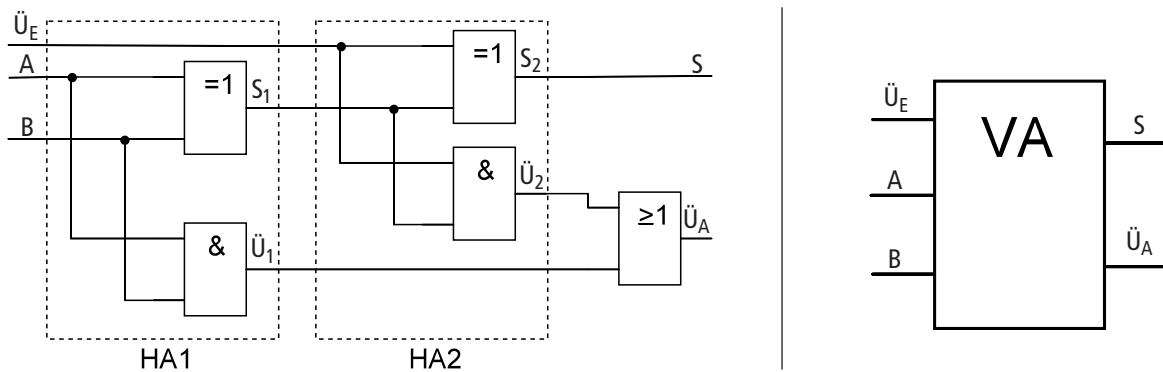


Abb. 30: Schaltung Volladdierer (links) und Blockschaltbild (rechts)

Wahrheitstabelle 1-Bit-Volladdierer

Übertragseingang \ddot{U}_E	Summand A	Summand B	Summe S	Übertragsausgang \ddot{U}_A
0	0	0	0	0
0	1	0	1	0
0	0	1	1	0
0	1	1	0	1
1	0	0	1	0
1	1	0	0	1
1	0	1	0	1
1	1	1	1	1

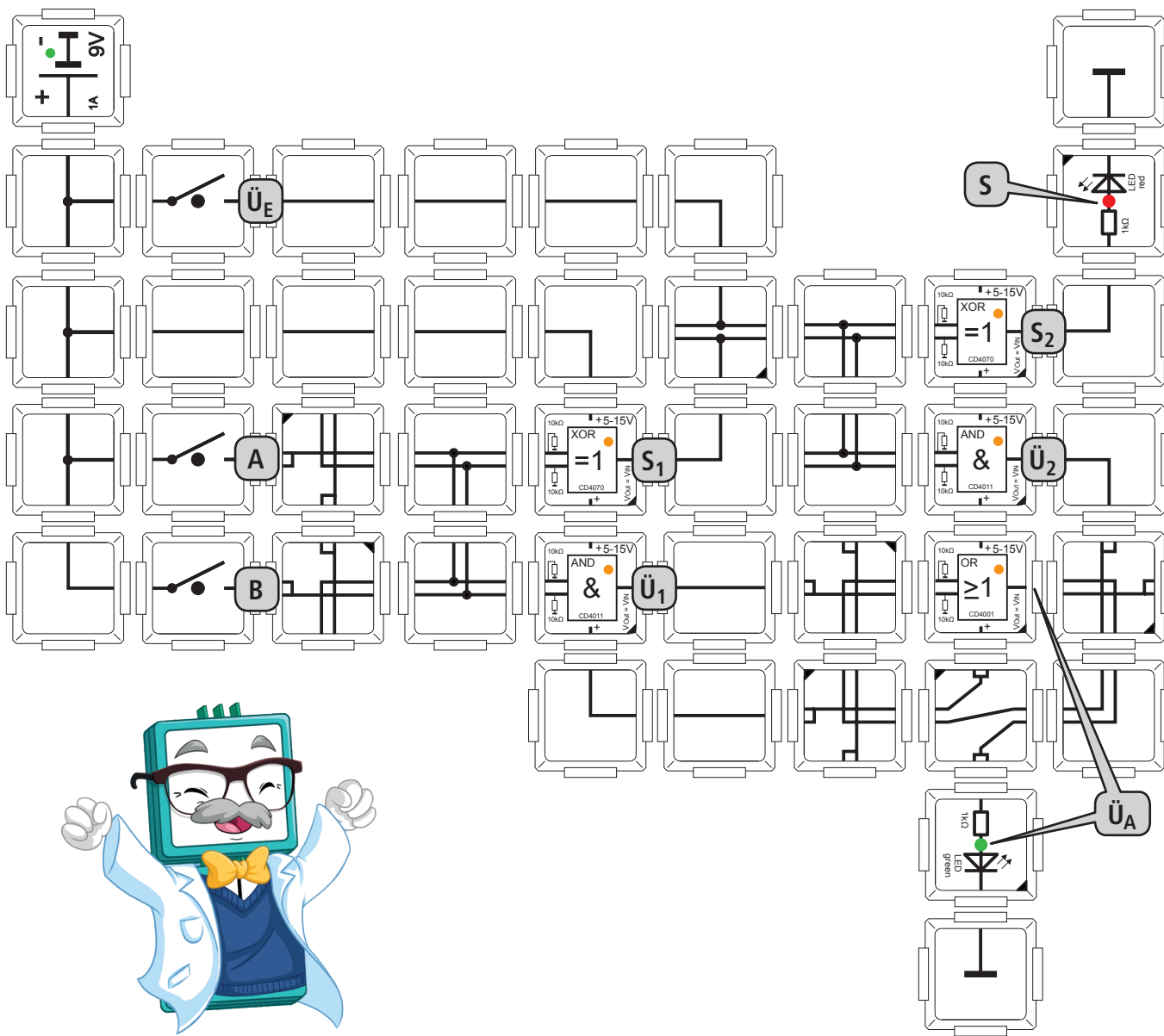


Abb. 31: 1-Bit Volladdierer als Brick-Schaltung

5.3.3 4-Bit-Volladdierer

Das folgende Blockschaltbild zeigt einen 4-Bit-Volladdierer (VA0 bis VA3), der aus vier 1-Bit-Volladdierern aufgebaut wurde. Damit lassen sich zwei vierstellige Dualzahlen addieren.

Der Eingang \ddot{U}_{E0} kann mit Masse verbunden werden, da für die niederwertigste Stelle kein Übertrag berücksichtigt werden muss. Die Überträge von VA0 bis VA3 werden jeweils durch die Verbindung der Übertragsausgänge \ddot{U}_{An} mit den Übertragungseingängen des jeweils höherwertigeren Addierers \ddot{U}_{En+1} weitergereicht. \ddot{U}_{A3} wird hier nicht weiter verwendet.

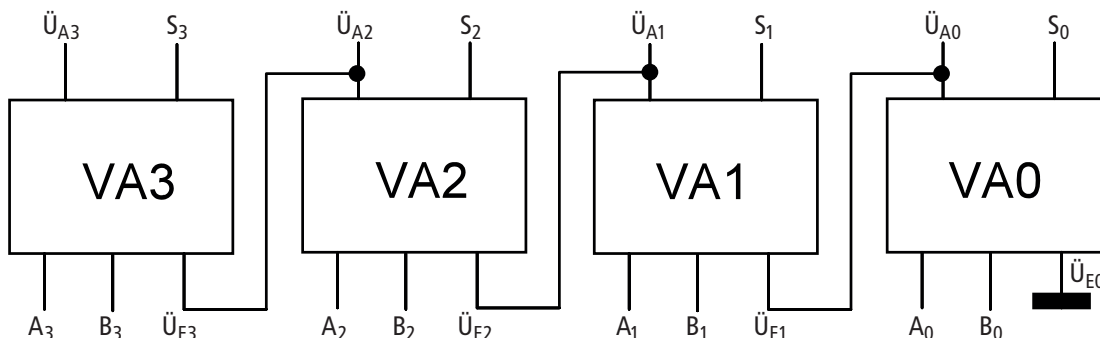


Abb. 32: Blockschaltbild 4-Bit-Volladdierer

Übrigens: In der Digitaltechnik ist es üblich, die Indizes für Busleitungen oder logische Einheiten mit "0" beginnen zu lassen. Dies hat auch den Hintergrund, dass die Wertigkeit der Stellen im Binärsystem mit Zweierpotenzen 2^n – beginnend bei $n = 0$ – beschrieben werden kann. Zum Beispiel:

$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
-----------	-----------	-----------	-----------



Aufgrund der Komplexität der Schaltung für den 4-Bit-Volladdierer kann dieser nicht mit den Bricks des Logic Sets aufgebaut werden.



5.4 Flipflop / Bistabile Kippstufe

Angenommen Sie erwarten eine Nachricht sind aber gerade nicht zuhause, um diese entgegennehmen zu können. Gut, dass es einen Briefkasten gibt, damit der Absender bzw. Postbote diese dort für Sie einwerfen kann. So geht die Nachricht nicht verloren und Sie können die Nachricht aus dem Briefkasten holen und lesen sobald Sie Zeit haben. Man könnte den Briefkasten auch als eine Art Zwischenspeicher bezeichnen. Der rote Winker zeigt an, ob eine Information bereit liegt oder nicht. Er kann also zwei Zustände signalisieren. So ähnlich können Sie sich das sogenannte Flipflop als kleinstes digitales Speicherelement (1 Bit) vorstellen.



Der Begriff Flipflop oder auch bistabile Kippstufe genannt, bezeichnet in der Digitaltechnik eine elektronische Schaltung, die zwei stabile elektrische Zustände ("0" oder "1") einnehmen kann. Durch entsprechende Eingangssignale kann die Kippstufe von einem Zustand in den anderen überführt werden.

Damit haben wir die Möglichkeit eine digitale Information (1 Bit) zu speichern – solange die Versorgungsspannung anliegt. Das Flipflop ist das Herzstück sequentieller Schaltungen und wird in verschiedenen Ausführungen eingesetzt. In Mikroprozessoren werden mehrere Flipflops zu sog. Registern parallelgeschaltet. Die Breite eines Registers gibt an, wieviele Flipflops zu einem Register zusammengefasst wurden (gängige Einheiten sind: 8 bit = 1 Byte; 16 bit = 1 Word; 32 bit = 1 Longword).

Das RS-Flipflop wird oft auch mit einer mechanischen Wippe verglichen. Die Zustände Q und \bar{Q} korrespondieren dabei mit den nichtinvertierenden bzw. invertierenden Flipflop-Ausgängen:

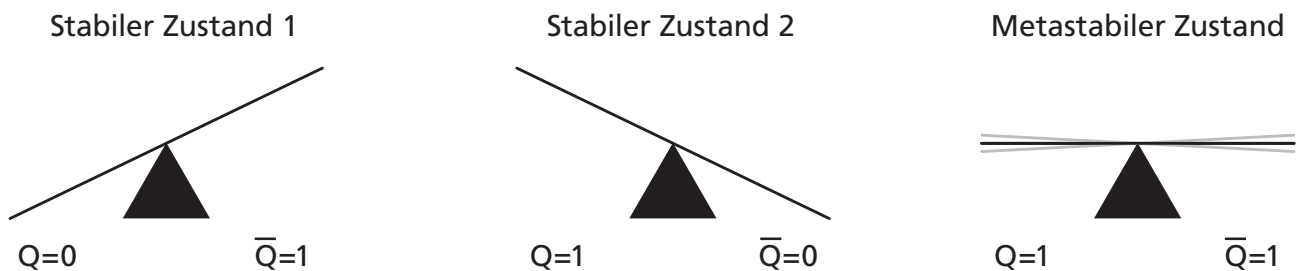


Abb. 33: Bistabile Kippstufe am Beispiel einer Wippe

Eine Wippe besitzt zwei stabile Zustände: Einerseits, wenn ihr linkes Ende den Boden berührt, andererseits, wenn ihr rechtes Ende auf dem Boden aufliegt. Durch geeigneten Kraftaufwand kann man die Wippe von einem stabilen Zustand in den anderen stabilen Zustand überführen (bei einem Flipflop geschieht dies durch geeignete Aktivierung der Eingänge, d. h. Reset = 0, Set = 1 bzw. Reset = 1, Set = 0). Lässt die steuernde Kraft nach (entspricht Reset = 0 und Set = 0), so verharrt die Wippe in dem zuvor eingestellten Zustand: Dieser ist damit gleichsam gespeichert. Eine horizontale Ausrichtung der Wippe führt jedoch zu einem metastabilen Zustand (entspricht dem Fall Reset = 1 und Set = 1): Da die Wippe in der Praxis nie exakt symmetrisch gebaut ist und nicht alle äußeren Störeinflüsse ausgeschaltet werden können, kippt die Wippe nach einer bestimmten Zeit und nimmt einen der beiden stabilen Zustände ein, sobald die steuernde Kraft nachlässt. In der Regel kann nicht vorhergesagt werden, ob ihr linkes Ende oder ihr rechtes Ende in Richtung des Erdbodens kippen wird, da nicht alle Störungen exakt genug bekannt sind. Ein getaktetes Flipflop entspricht dabei einer Wippe, bei der die steuernde Kraft nur während einer durch ein äußeres Taktsignal bestimmten Zeit wirken kann.

5.4.1 Flipflop-Typen

Die verschiedenen Flipflop-Typen unterscheiden sich in Art und Anzahl der Eingänge und Ihrem Verhalten. Sie haben unterschiedlich wirkende Eingänge und ändern ihren Zustand nur bei bestimmten festgelegten Bedingungen. Ein einfaches Flipflop hat zwei Eingänge und zwei Ausgänge. Taktabhängige Flipflops haben noch einen entsprechenden Takteingang.

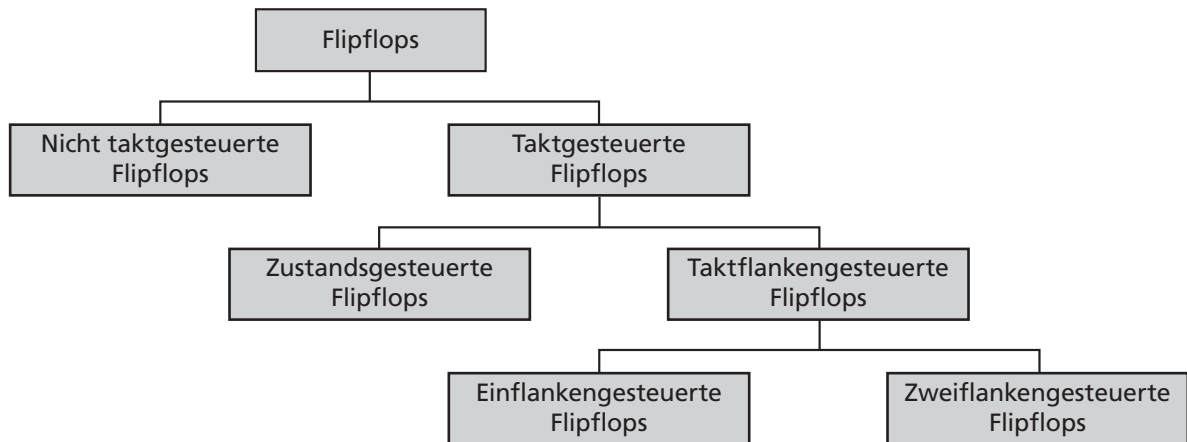
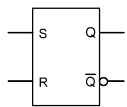


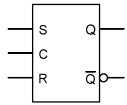
Abb. 34: Flipflop-Typen

5.4.1.1 Nicht taktgesteuerte Flipflops



Das RS-Flipflop ohne Takteingang ist vollständig taktunabhängig. Seine Setz- und Rücksetzeingänge S und R lassen sich jederzeit ansprechen. Siehe auch Kap. 5.4.2 auf Seite 40

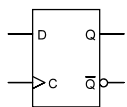
5.4.1.2 Zustandsgesteuerte Flipflops



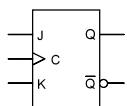
Die Setz- und Rücksetzeingänge S und R eines zustandsgesteuerten RS-Flipflops sind nur wirksam, solange am Takteingang C ein Signalpegel (hier High-Pegel) anliegt. Man kann diesen Flipflop-Typ vom RS-Flipflop ableiten, indem man vor jeden Eingang eines aus NOR-Gattern aufgebauten RS-Flipflops ein AND-Gatter schaltet und je einen Eingang als Takteingang zusammenfasst.

5.4.1.3 Einflankengesteuerte Flipflops

Die Störanfälligkeit eines flankengesteuerten Flipflops gegenüber einem zustandsgesteuerten Flipflop wird heruntergesetzt. Die Taktflankensteuerung wird im Schaltzeichen durch das Dreieck gekennzeichnet.



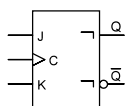
Das D-Flipflop kann einfach vom einflankengesteuerten RS-Flipflop abgeleitet werden, indem der Rücksetzeingang invertierend zum Setzeingang beschaltet wird (siehe auch Kap. 5.4.3 auf Seite 41). Je nach Pegel am D(aten)-Eingang wird der Zustand mit der Flanke am Takteingang C (hier positive Flanke) am Q-Ausgang übernommen. D-Flipflops kommen beispielsweise bei Schieberegistern zum Einsatz (siehe Kap. 5.5.1 auf Seite 44).



Eine weiterer Vertreter der einflankengesteuerten Flipflops ist das sog. JK-Flipflop. Beim Anlegen einer Flanke am Takteingang C (hier positive Flanke) wechselt der Ausgangszustand, sofern an den beiden Eingängen J und K eine 1 anliegt. Dieses Verhalten wird auch als Toggeln (hin- und herschalten) bezeichnet und führt uns zum T-Flipflop. Es kann aus einem einflankengesteuerten JK-Flipflop nachgebildet werden, indem J- und K-Eingang verbunden werden und gemeinsam als Freigabe-Eingang T fungieren. Es wird z. B. in Synchronzählern verwendet. Siehe auch Kap. 5.6.3 auf Seite 52.

5.4.1.4 Zweiflankengesteuerte Flipflops

Zweiflankengesteuerte Flipflops, deren Eingangszustände verzögert am Ausgang erscheinen, werden auch Master-Slave-Flipflops genannt. Im Schaltsymbol ist dies am rechten Winkel am Ausgang erkennbar. Ein bekannter Vertreter dieses Typs ist das sogenannte JK-Master-Slave-Flipflop.

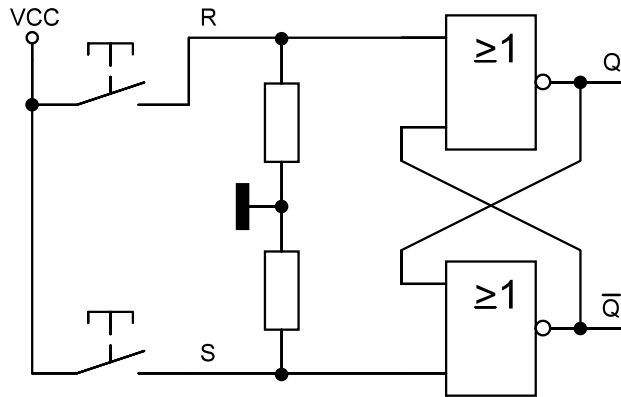


Das zweiflankengesteuerte JK-Flipflop nimmt während der ersten Taktflanke (hier positive Flanke) den Eingangszustand auf und gibt diesen mit der folgenden (negativen) Taktflanke aus. Die Störanfälligkeit wird minimiert. Die Taktflankensteuerung wird im Schaltzeichen durch das Dreieck gekennzeichnet. Siehe auch Kap. 5.4.4 auf Seite 42.

In den folgenden Kapiteln werden die im Brick'R'knowledge Logic Set enthaltenen Flipflop-Typen vorgestellt.

5.4.2 RS-Flipflop

Das nicht-taktgesteuerte RS-Flipflop kann auch aus zwei NOR-Gatter aufgebaut werden. Diese Grundschaltung nennt man auch NOR-Flipflop. Prinzipiell ist die Schaltung auch mit zwei NAND-Flipflops möglich, erfordert jedoch negierte Eingänge.



Elegante Variante mit Wechseltaster (nicht im Logic Set enthalten) und Schaltzeichen für RS-Flipflop!

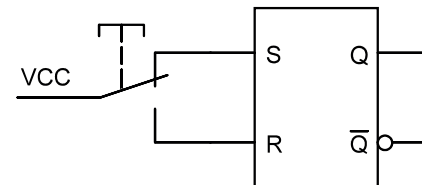
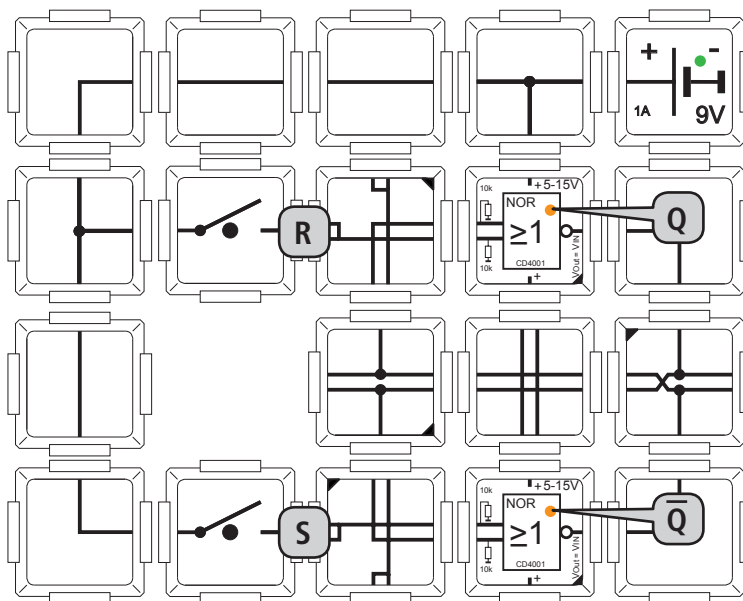


Abb. 35: RS-Flipflop aus zwei NOR-Gattern (links) und Schaltzeichen RS-Flipflop (rechts)

Die Setz- und Rücksetz-Eingänge werden auch mit R und S bezeichnet, wie im Schaltzeichen des RS-Flipflops zu erkennen ist. Die beiden Ausgänge Q und \bar{Q} sind zueinander negiert.

Wahrheitstabelle für nicht-taktgesteuertes RS-Flipflop

Setz-Eingang S	Rücksetz-Eingang R	Ausgang Q	Ausgang \bar{Q}	Erläuterung
1	0	1	0	Setzen (SET)
0	0	Q	\bar{Q}	Keine Änderung
0	1	0	1	Rücksetzen (RESET)
1	1	0	0	Verbotener Zustand



! Nach dem Anlegen der Versorgungsspannung befinden sich die Ausgänge in einem zufälligen, nicht definierten Zustand (Low oder High)!

Abb. 36: RS-Flipflop aus NOR-Bricks aufgebaut

- Setzen: Durch drücken der Taste am S-Eingang wird der Ausgang Q auf "1" gesetzt.
- Keine Änderung: Wird keine Taste gedrückt, so bleiben die Ausgänge unverändert.
- Rücksetzen: Durch drücken der Taste am R-Eingang, wird der Ausgang Q auf "0" gesetzt.

- Verbotener Zustand: Werden beide Tasten gedrückt, d. h. die Eingänge auf "1" gesetzt, gehen beide Ausgänge auf "0". Dieser Zustand kann nicht gespeichert werden. Kritisch ist dies jedoch nur dann, wenn beide Eingänge nach diesem Zustand gleichzeitig "0" werden. Dieser Folgezustand ist undefiniert, weil nicht klar ist, welcher Ausgang zunächst "1" werden soll.

5.4.3 D-Flipflop

Das D-Flipflop besteht aus einem RS-Flipflop, bei dem der Rücksetzeingang über einen Inverter mit dem Setzeingang verbunden ist. Dadurch wird verhindert, dass der undefinierte Zustand eintritt.

Das D-Flipflop gibt es als taktzustandsgesteuertes und als taktflankengesteuertes Flipflop wie in unserem Fall. Sofern ein D-Flipflop herausgeführte Setz- und Rücksetz-Eingänge hat, lässt es sich über diese auch taktunabhängig (asynchron) setzen bzw. zurücksetzen (siehe auch Kap. 5.2 auf Seite 33).

Das D-Flipflop stellt das Grundelement Schreib-Lese-Speicher dar. In der Praxis werden meist viele D-Flipflops parallel geschaltet und über einen gemeinsamen Takt synchronisiert. Eine solche Anordnung aus meist 4, 8, 16 oder 32 D-Flipflops nennt man Register. Eine praktische Anwendung ist z. B. ein Status-Register welches gelesen werden soll um Informationen über ein Gerät zu erhalten oder ein Befehlsregister um Kommandos an ein Gerät zu übermitteln.

Anschluss D ist der Daten-Eingang. In unserem Fall eines einflankengesteuerten D-Flipflops, wird der Wert am Daten-Eingang ("0" oder "1") mit der positiven Flanke am Takt-Eingang C übernommen.

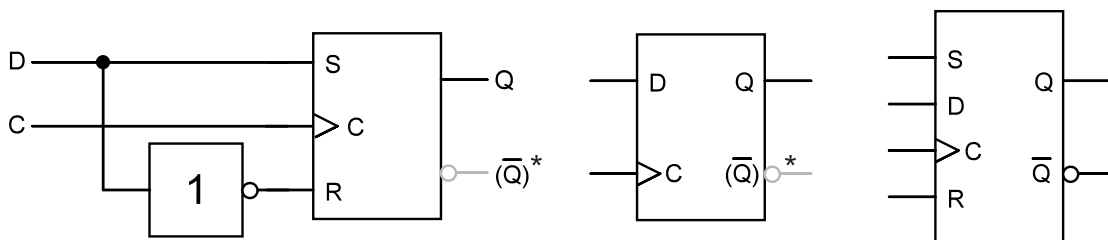


Abb. 37: D-Flipflop aus flankengesteuertem RS-Flipflop (links), Schaltzeichen eines einfachen D-Flipflops (Mitte) und mit RS-Eingängen (rechts)

Wahrheitstabelle für positiv flankengesteuertes D-Flipflop (opt. mit Setz- und Rücksetz-Eingängen)

Daten-Eingang D	Takt-Eingang C	(Eingang S)	(Eingang R)	Ausgang Q	(Ausgang \bar{Q})	Erläuterung
1	0 → 1	0	0	1	(0)	Setzen
0	0 → 1	0	0	0	(1)	Rücksetzen
1	1 → 0	0	0	Q	(\bar{Q})	Keine Änderung
0	1 → 0	0	0	Q	(\bar{Q})	Keine Änderung
X	X	1	0	1	0	Asynchrones Setzen
X	X	0	1	0	1	Asyn. Rücksetzen

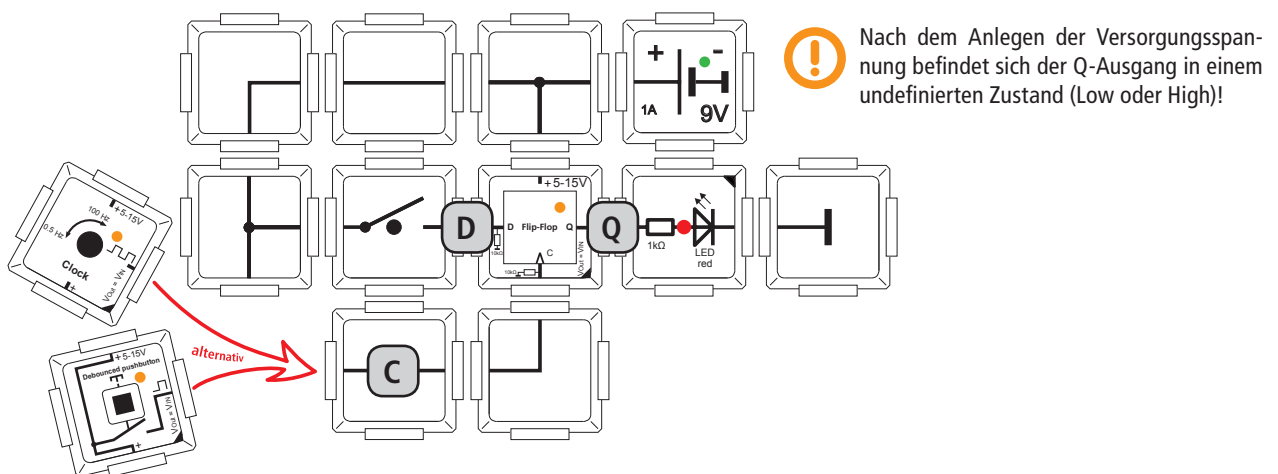


Abb. 38: Brick-Schaltung mit Standard D-Flipflop

*Beachte, dass beim Standard-D-Flipflop-Brick (siehe Abb. 38), der \bar{Q} -Ausgang nicht herausgeführt ist. Im Vergleich dazu, stehen bei dem in Abb. 39 verwendeten D-Flipflop, neben den beiden Ausgängen Q und \bar{Q} auch die asynchronen Setz (S)- und Rücksetz (R)-Eingänge zur Verfügung!

D-Flipflop mit asynchronem Set- und Reset-Eingang

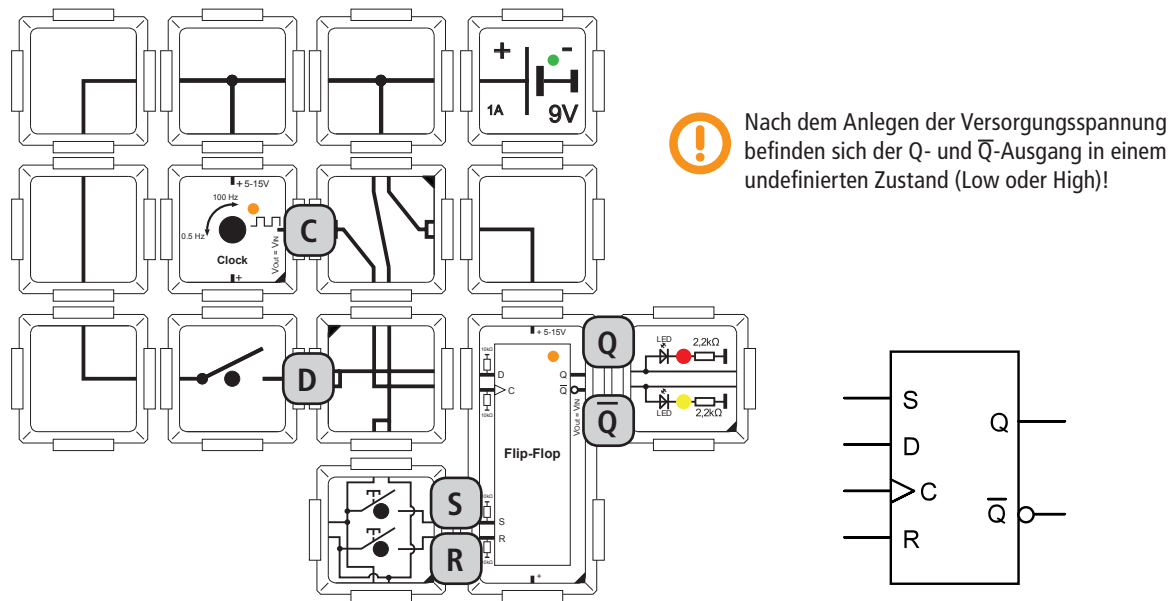


Abb. 39: Brick-Schaltung mit D-Flipflop und asynchronem Set- und Reset-Eingang (links), Schaltzeichen für verwendetes D-Flipflop (rechts)

5.4.4 JK-Flipflop

Das JK-Flipflop gibt es als taktflankengesteuertes und taktzustandgesteuertes Flipflop. Im Brick'R'knowledge Logic Set sind vier (ein)flankengesteuerte JK-Flipflops enthalten. Beim Anlegen einer positiven Taktflanke wechselt der Ausgangszustand, sofern an den beiden Eingängen J und K eine "1" anliegt. Dieses Verhalten wird auch als Toggeln (hin-und-herschalten) bezeichnet. Siehe auch T-Flipflop, bei dem die beiden Eingänge J und K verbunden sind und als T(oggle)-Eingang zur Verfügung stehen.

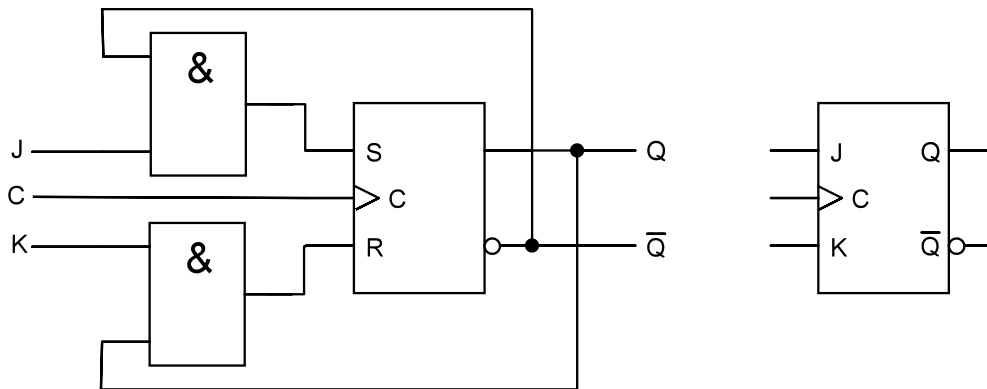


Abb. 40: Aufbau eines JK-Flipflops (links) und Schaltzeichen für einflankengesteuertes JK-Flipflop (rechts)

Am Takteingang C (vom englischen "Clock") des JK-Flipflops muss ein Rechtecksignal angelegt werden. Das von uns verwendete JK-Flipflop reagiert stets auf eine positive Taktflanke, also den Übergang an C von "0" nach "1". Die beiden Eingänge J und K sind Steuereingänge. Die Ausgänge Q und \bar{Q} werden in Abhängigkeit der Eingänge J und K gesteuert.

Wahrheitstabelle für positiv flankengesteuertes JK-Flipflop

Eingang J	Eingang K	Takt-Eingang C	Ausgang Q	Ausgang \bar{Q}	Erläuterung
0	0	0 → 1	Q	\bar{Q}	Keine Änderung
1	0	0 → 1	1	0	Setzen
0	1	0 → 1	0	1	Rücksetzen
1	1	0 → 1	\bar{Q}	Q	Ausgänge toggeln

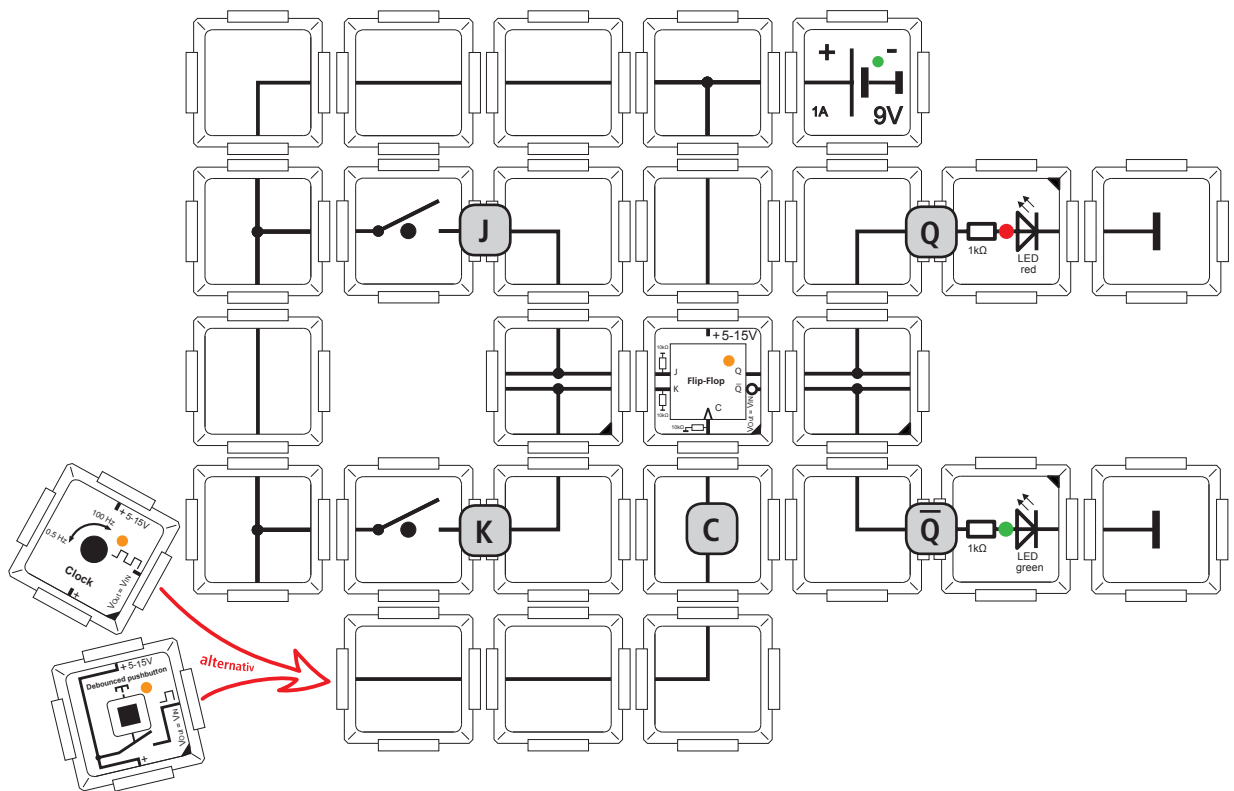


Abb. 41: JK-Flipflop als Brick-Schaltung

- Keine Änderung: Wird keine Taste gedrückt, so bleiben die Ausgänge Q und \bar{Q} unverändert.
- Setzen: Durch drücken der Taste am J-Eingang wird der Ausgang Q auf "1" und \bar{Q} auf "0" gesetzt.
- Rücksetzen: Durch drücken der Taste am K-Eingang (Taste J nicht gedrückt) wird der Ausgang Q auf "0" und \bar{Q} auf "1" gesetzt.
- Ausgänge toggeln: Werden beide Tasten gedrückt, toggeln die Ausgänge. D.h. für den Q-Ausgang: eine "1" vor der Taktflanke wird zu "0" nach der Flanke und eine "0" vor der Taktflanke wird zu einer "1" nach der Flanke. Der \bar{Q} -Ausgang verhält sich entsprechend invertierend.

Alternative JK-Flipflop-Schaltung

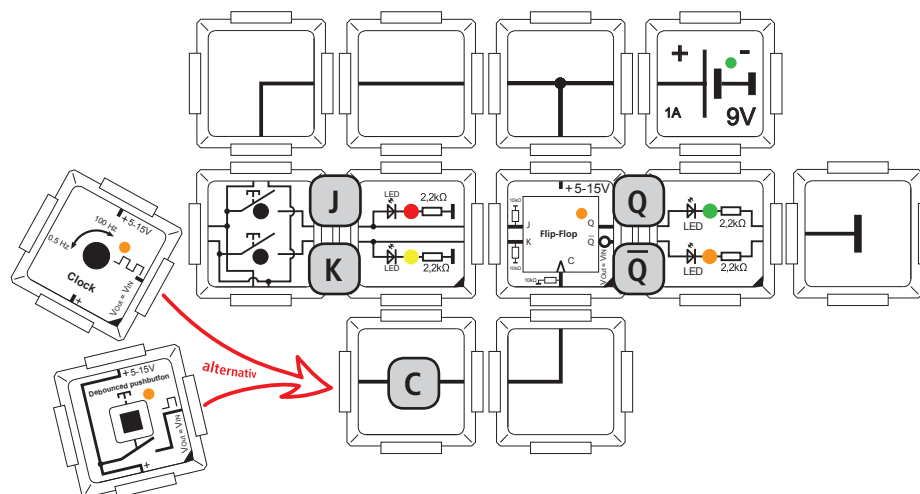


Abb. 42: Alternative JK-Flipflop-Schaltung

Aus didaktischen Gründen empfehlen wir, den Takt (Clock-Brick) möglichst langsam einzustellen (0,5Hz). Alternativ können Sie auch den entprellten entprellten Taster (Debounce pushbutton) verwenden um das Verhalten in Ruhe zu studieren (siehe auch 5.2.2 auf Seite 33).

5.5 Schieberegister

Schieberegister bestehen aus in Serie geschalteten Flipflops, die synchron getaktet werden. Das bitweise Schieben zählt neben der Addition (siehe Kap. 5.3 auf Seite 34) zu den elementaren Operationen einer arithmetischen Recheneinheit. Wird beispielsweise die eingelesene Dualzahl um eine Stelle nach rechts geschoben, entspricht das dezimale Ergebnis einer Division mit 2. Zur Multiplikation muss die entsprechende Dualzahl um eine Stelle nach links geschoben werden. Je nach Anwendungsfall haben Schieberegister unterschiedliche Bit-Breite, schieben nach rechts oder links und können seriell oder parallel geladen werden. Für viele Aufgaben eignen sich Schieberegister als integrierte Schaltkreise (ICs), die bereits alle Standardfunktionalitäten beinhalten.

Einsatzmöglichkeiten für Schieberegister sind:

- Arithmetische Operationen (Multiplikation, Division)
- Ringspeicher
- Seriell-Parallel-Wandler und Parallel-Seriell-Wandler
- Verzögerung von Signalen

5.5.1 Schieberegister mit seriellem Laden

Wird ein Schieberegister am D-Eingang mit einer "1" geladen, so wird dieser Wert mit jedem Takt um eine Stelle (also um ein Flipflop) weitergeschoben.

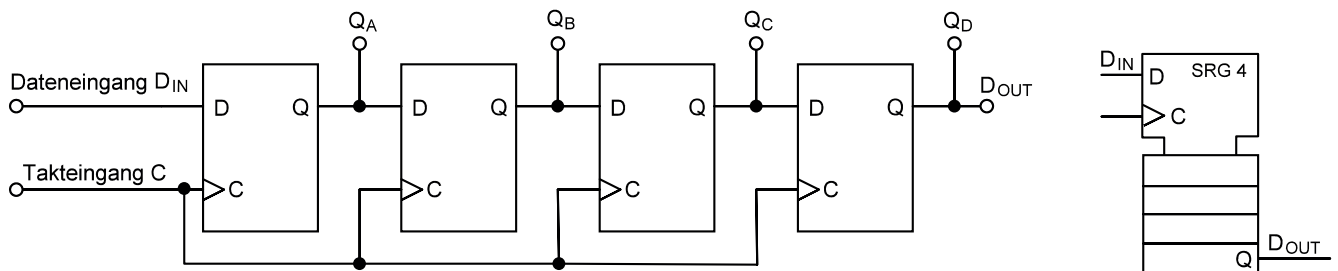


Abb. 43: 4-Bit-Schieberegister aus D-Flipflops mit serieller Ein- und Ausgabe (links), Schaltzeichen (rechts)

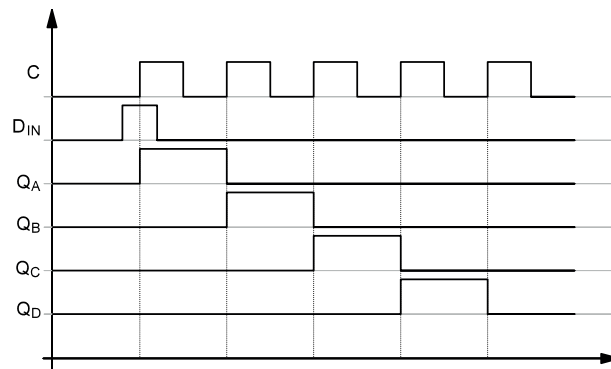


Abb. 44: Timing-Diagramm Schieberegister

In der folgenden Brick-Schaltung ist das Schieberegister aus vier D-Flipflops aufgebaut. Solange der Taster gedrückt wird, liegt am Daten-Eingang des ersten Flipflops ein High-Pegel an. Mit der nächsten positiven Flanke des Taktes C wird die "1" übernommen und an Q_A ausgegeben. Zugleich wird der Pegel von Q_A an den Dateneingang des zweiten Flipflops weitergegeben. Das Prinzip setzt sich fort, bis nach vier Taktzyklen das 4-Bit-breite Datenwort im Schieberegister (Q_A bis Q_D) steht. Q_D ist in unserem Beispiel zugleich der serielle Datenausgang D_{OUT} . Dieser könnte alternativ auch mit D_{IN} verbunden werden um ein Ringregister zu bauen in dem die Bits quasi im Kreis geschoben werden.

Der Takt wird mit dem Taktgeber-Brick (Clock) erzeugt. Unter der Haube kommt der bekannte Timer-Baustein vom Typ 555 zum Einsatz. Die Taktfrequenz ist zwischen 0,5 Hz und 100 Hz einstellbar (siehe auch 5.2.2 auf Seite 33). Der entprellte Taster wird für diese Übung nicht empfohlen!



Nach dem Anlegen der Versorgungsspannung befinden sich die Q-Ausgänge in einem undefinierten Zustand (Low oder High)!

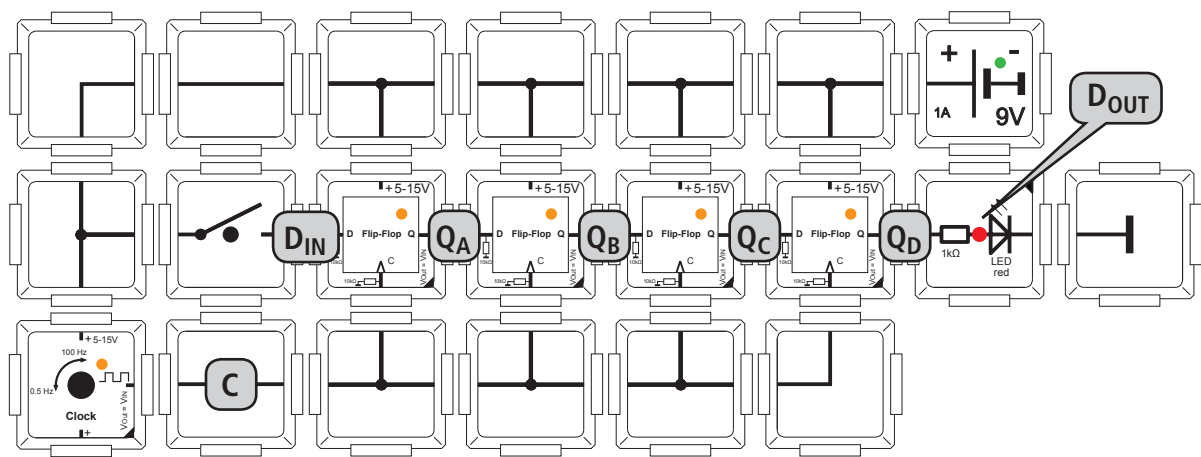


Abb. 45: Schieberegister als Brick-Schaltung

5.6 Zähler

Als digitale Zähler eignen sich Flipflop-Schaltungen ideal. Die Binärzähler werden entsprechend ihrer Ansteuerung und der Zählrichtung eingeteilt. Nach der Art Ihrer Taktung unterscheidet man zwei grundlegende Zählertypen die asynchronen und die synchronen Zähler. Das Taktsignal sollte stets prellfrei sein, um Zählfehler durch Kontaktprellen zu vermeiden (siehe auch Kap. 5.2 auf Seite 33).

Bei **Synchronzählern** steuert der Zähltakt alle Speicher-Flipflops gleichzeitig an. Eine Addition der Signallaufzeiten, wie sie bei Asynchronzählern durch die Taktweitergabe von Stufe zu Stufe entsteht, gibt es hier nicht. Die Ausgangspegel aller Stufen erscheinen gleichzeitig nach der steuernden Taktflanke. Im Vergleich zu Asynchronzählern erlaubt das Synchronverfahren wesentlich höhere maximale Zählfrequenzen. Der Schaltungsaufwand ist bei Synchronzählern allerdings größer.

Die Taktsteuerung von **Asynchronzählern** erfolgt für alle Zählerbaugruppen sequentiell. Der Zählertakt steuert nur das erste Flipflop. Alle folgenden werden von Ausgangspegel ihres Vorgängers gesteuert. Wie bei allen asynchronen Abläufen, kann es aber zu unübersichtlichen Schaltungen mit Laufzeitproblemen kommen. Dafür ist der Gatteraufwand geringer. Im englischen Sprachraum werden Asynchronzähler auch als "Ripple Counter" bezeichnet, da sich der Steuertakt einer Welle vergleichbar durch die Flipflop-Stufen fortpflanzt.

Sowohl Synchron- als auch Asynchron-Zähler gibt es als Aufwärtszähler oder Abwärtszähler. Mit Zusatzschaltungen kann die Zählrichtung wahlweise umschaltbar gemacht und der Zählbereich eingeschränkt werden. Bei Bedarf kann der Zählwert unterschiedlich codiert ausgegeben werden (z. B. Binär-, BCD- oder Gray-Zähler). Siehe auch Tabelle auf Seite 18.

Der universelle Grundbaustein eines Dualzählers ist ein als Toggle-Flipflop geschaltetes JK-Flipflop. Jedes Flipflop entspricht einer Binärstelle und speichert ein Bit. Durch geeignete Verknüpfung der Flipflops (siehe folgende Kapitel) entstehen mehrstellige Dualzähler.

Die Zählkapazität K eines Dualzählers ist von der Anzahl n der Einzelspeicher (Flipflops) abhängig. Für einen Binär-Zähler gilt folgende Formel:

$$K = 2^n - 1. \quad \text{Die Werte für } n=2 \text{ bis } 10 \text{ finden Sie in folgender Tabelle:}$$

Abhängigkeit von Zähltiefe und Zählkapazität

Anzahl der Flipflops n / (Zähltiefe in Bits)	Zählkapazität (K)
2	3
3	7
4	15
5	31
6	63
7	127
8	255
9	511
10	1023

In der Praxis kommen meist 4- oder 8-Bit-breite Zählerbausteine zum Einsatz. Bei Bedarf können beispielsweise zwei 8-Bit-Zähler zu einem 16-Bit-Zähler verknüpft werden. Im modernen Schaltungsdesign werden Zähler als Logikblöcke in programmierbare Bausteine wie z. B. FPGAs (Field-Programmable Gate-Array) integriert (siehe auch Kap. 4.8 auf Seite 23).



5.6.1 Asynchroner 4-Bit-Binär-Zähler

Asynchron arbeitende Zähler haben keinen gemeinsamen Takt. Die Flipflops werden hintereinandergeschaltet, d. h. der Ausgang des ersten Flipflops wird mit dem Takteingang des zweiten Flipflops verbunden usw. Aufgrund von Signallaufzeiten bekommt jedes Flipflop die Taktflanke zu unterschiedlichen Zeiten am Takteingang. Die Signallaufzeiten der einzelnen Speicherstufen summieren sich. Dies wirkt sich besonders bei Binär-Zählern mit großer Zähltiefe und hohen Taktfrequenzen aus. Für ein korrektes Zählen muss die Taktzeit stets größer als die Gesamtlaufzeit des Signals durch den kompletten Zähler sein.

Ist bei n gleichen Flipflops die kürzeste Schaltzeit t (in Sekunden) eines Speichers bekannt, errechnet sich die maximale Frequenz (Grenzfrequenz f_g in Hertz [Hz]) des Zähltakts zu:

$$f_g = \frac{1}{(n + 1) \times t}$$

Bei Flipflops in CMOS- oder TTL-Technologie dauert die Signallaufzeit wenige Nanosekunden. Je größer die Zähltiefe (z. B. 16 bit), desto länger dauert es, bis sich der Impuls vom ersten Flipflop bis zum letzten Flipflop fortpflanzt. Diese Laufzeit des Zählimpulses kann zu Störungen und so zu Fehlern beim Zählen führen. Je höher die Zählfrequenz, desto eher treten Probleme auf. Solange nur im Bereich bis 100 Hz gezählt wird – wie mit unserem Taktgeber-Brick – ist ein 4-Bit Asynchrnzähler kein Problem.

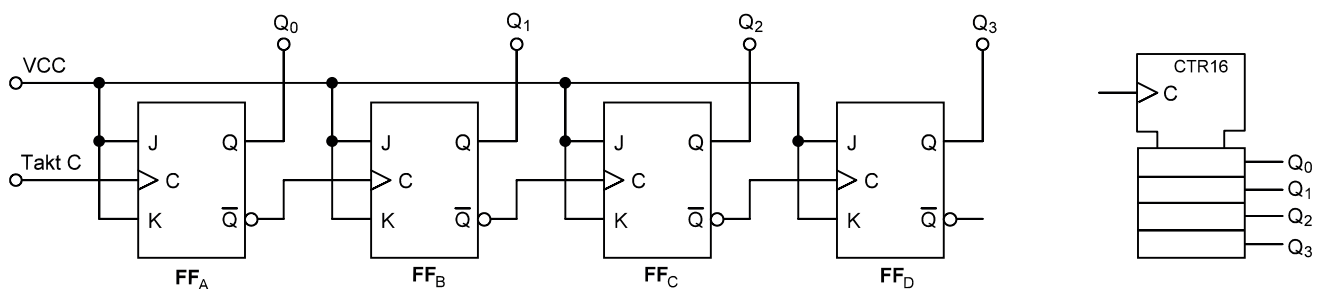


Abb. 46: Asynchroner 4-Bit-Binär-Zähler (links), Schaltzeichen (rechts)

5.6.1.1 Asynchroner Aufwärtszähler

Die folgende Brick-Schaltung stellt einen asynchronen Aufwärtszähler dar. Den Zählvorgang kann man anhand der orangenen LEDs in den Flipflop-Bricks, welche intern mit dem Q-Ausgang verbunden sind, gut beobachten (siehe auch Timing-Diagramm Abb. 48). Die Hintereinanderschaltung der Flipflop-Stufen erfolgt durch die Verbindung des Ausgangs \bar{Q}_0 der ersten Stufe mit dem Takteingang C der zweiten Stufe usw. Da wir einflankengesteuerte JK-Flipflops verwenden, bewirkt jeweils der Übergang vom Setz- in den Rücksetz-zustand eine positive Flanke am \bar{Q} -Ausgang. Da sowohl der J-Eingang als auch der K-Eingang mit High-Pegel verbunden sind, bewirkt dies ein Toggeln des jeweils nachfolgenden Flipflops.

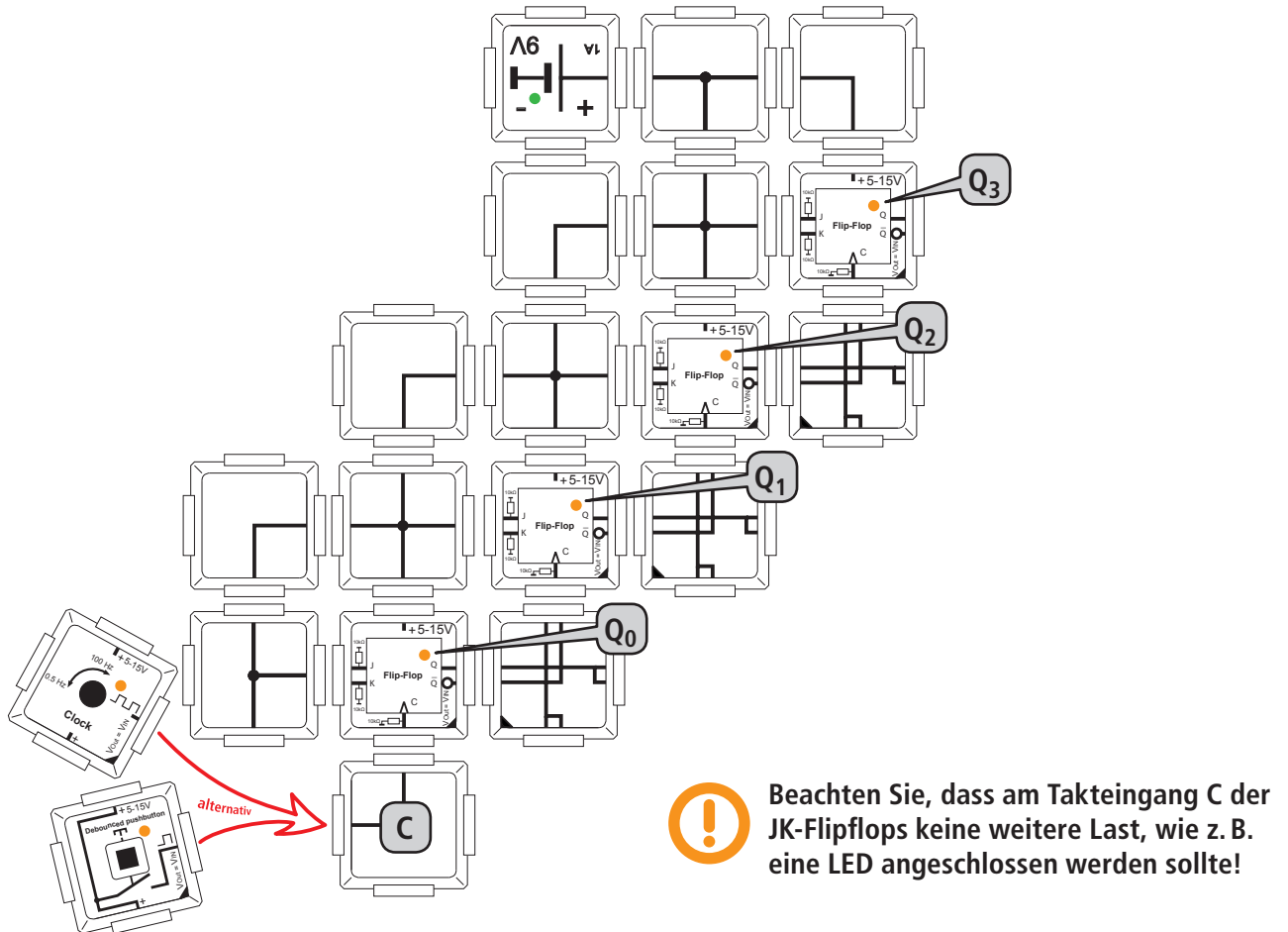


Abb. 47: Asynchroner 4-Bit-Binär-Aufwärtszähler aus JK-Flipflops als Brick-Schaltung

Im folgenden Timing-Diagramm sehen Sie einen 4-Bit-Binär-Aufwärtszähler, der von 0 bis 15 zählt und danach wieder bei 0 beginnt. Beachten Sie, dass der Takteingang C prellfrei beschaltet werden muss. Bei Verwendung des Taktgeber-Bricks (Clock) ist dies stets gewährleistet. Alternativ können Sie den entprellten Taster (Debounced pushbutton) verwenden um die Zähl Schritte in Ruhe nachzuvollziehen.

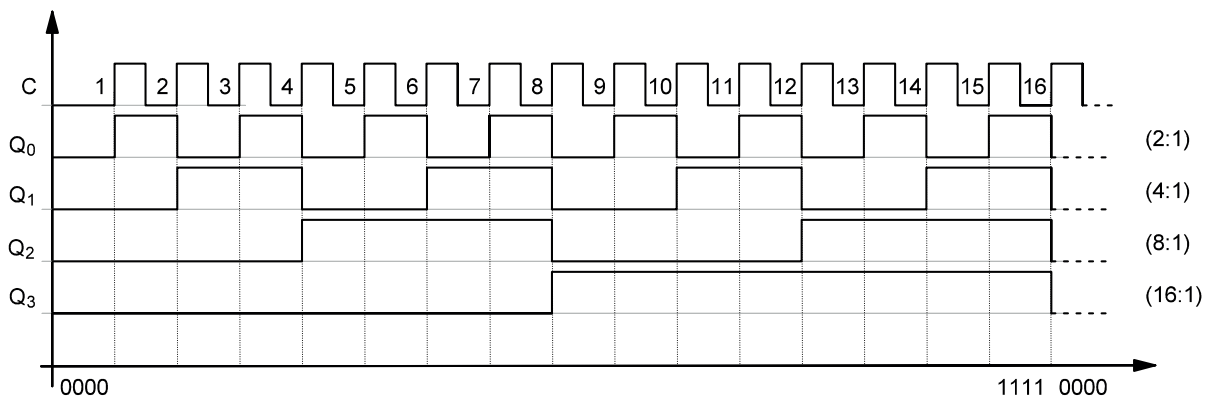


Abb. 48: Timing-Diagramm asynchroner 4-Bit-Binär-Aufwärtszähler

5.6.1.2 Asynchroner Abwärtszähler

Die folgende Brick-Schaltung zeigt einen asynchronen Abwärtszähler. Den Zählvorgang kann man anhand der im Flipflop eingebauten orangenen LEDs ablesen, welche mit dem Q-Ausgang (siehe auch Timing-Diagramm Abb. 50). Die Hintereinanderschaltung der Flipflop-Stufen erfolgt durch die Verbindung des Ausgangs Q_0 der ersten Stufe mit dem Takteingang C der zweiten Stufe usw. Da wir einflankengesteuerte JK-Flipflops verwenden, bewirkt jeweils der Übergang vom Rücksetz- in den Setzzustand eine positive Flanke am Q-Ausgang. Da sowohl der J-Eingang als auch der K-Eingang mit High-Pegel verbunden sind, bewirkt dies ein Toggeln des nachfolgenden Flipflops.

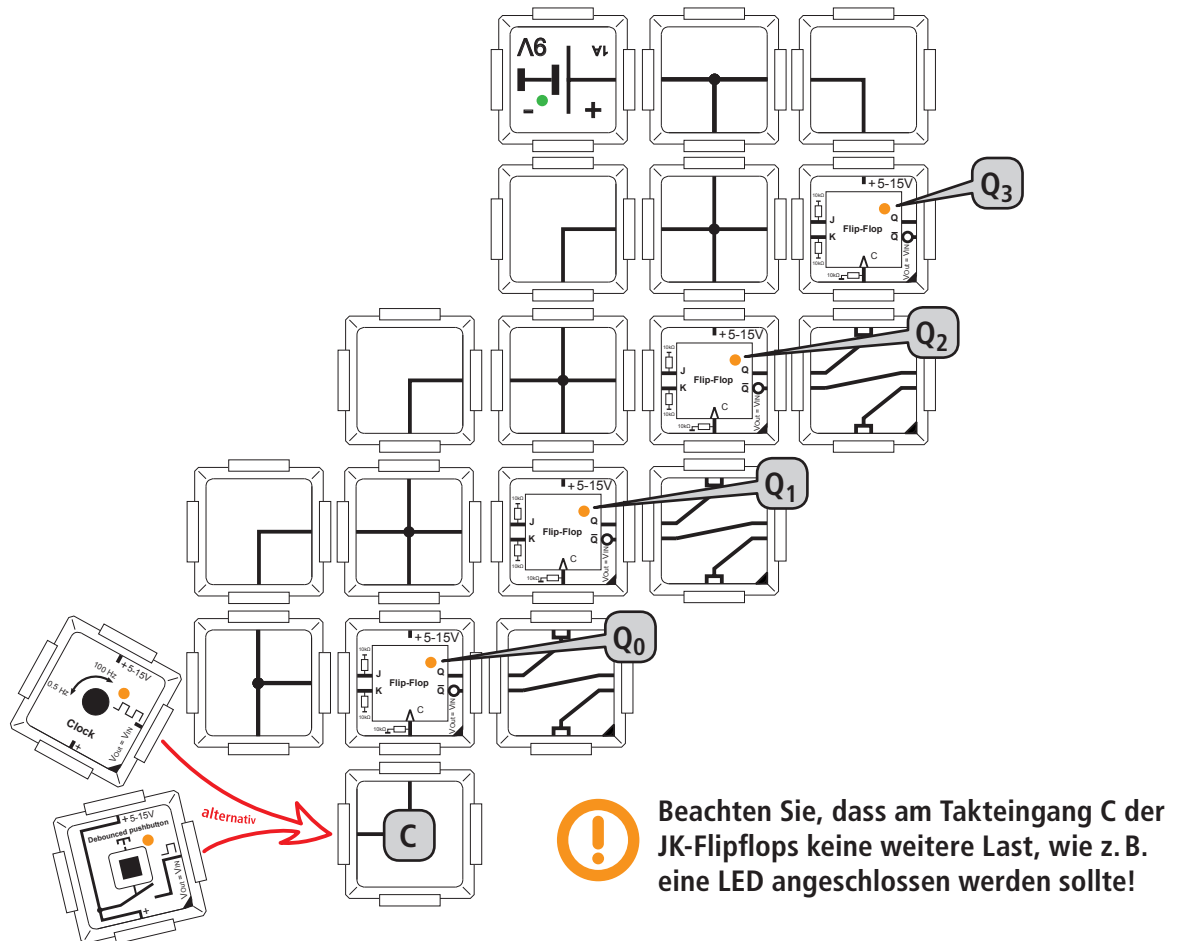


Abb. 49: Asynchroner 4-Bit-Binär-Abwärtszähler aus JK-Flipflops als Brick-Schaltung

Im folgenden Timing-Diagramm sehen Sie einen 4-Bit-Binär-Abwärtszähler, der von 15 bis 0 zählt und danach wieder bei 15 beginnt. Beachten Sie, dass der Takteingang C prellfrei beschaltet werden muss. Bei Verwendung des Taktgeber-Bricks (Clock) ist dies stets gewährleistet. Alternativ können Sie den entprellten Taster (Debounced pushbutton) verwenden um die Zähl Schritte in Ruhe nachzuvollziehen.

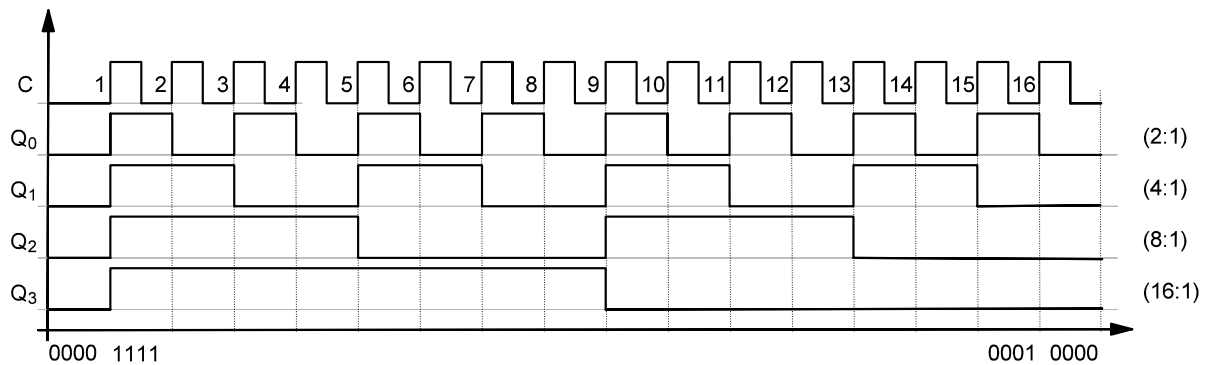


Abb. 50: Timing-Diagramm asynchroner 4-Bit-Binär-Abwärtszähler

5.6.1.3 Frequenzteiler

Frequenzteiler sind Schaltungen, welche die Frequenz eines Signals in einem bestimmten Teilverhältnis herunterteilen. Die meisten Frequenzteiler haben ein festes, ganzzahliges Teilverhältnis. Daneben gibt es noch einstellbare Frequenzteiler, die über zusätzliche Eingänge verfügen, die das Teilverhältnis bestimmen. Man nennt sie programmierbare Frequenzteiler.

Prinzipiell ist jeder Zähler auch als Frequenzteiler verwendbar. Als Schaltungsbeispiel können die im Kapitel 5.6 vorgestellten asynchronen oder synchronen Zähler dienen. Ein einzelnes Flipflop bewirkt bereits eine Frequenzteilung im Verhältnis 2:1 (Ausgang Q_0). Mit zwei Flipflops kann ein Frequenzteiler im Verhältnis 4:1 aufgebaut werden (Ausgang Q_1), usw.

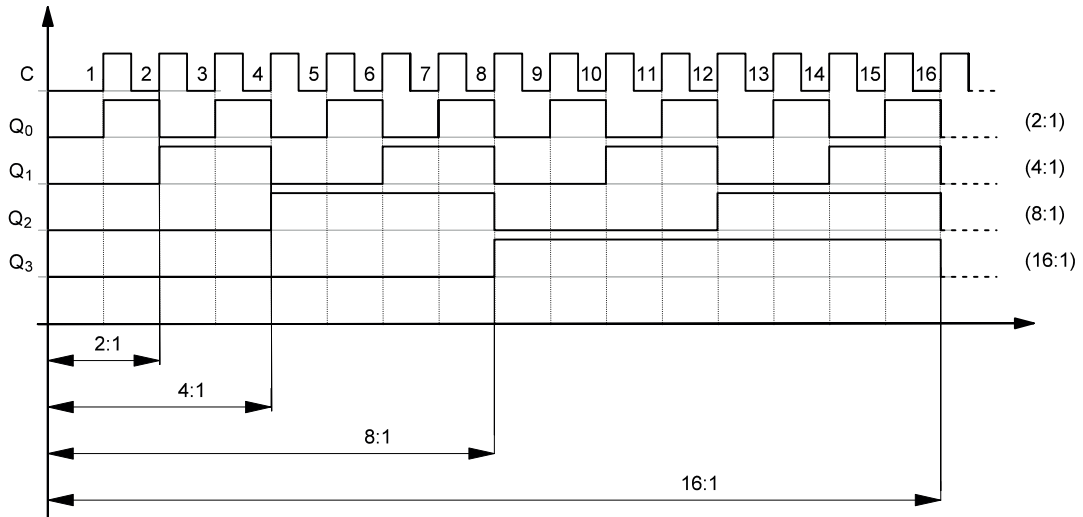


Abb. 51: Timing-Diagramm Frequenzzähler (siehe auch 4-Bit-Binär-Aufwärtszähler in Kap. 5.6.1 auf Seite 46)

Bei einem Takt von 100 Hz (Maximalfrequenz unseres Taktgeber-Bricks) ergeben sich an den Q-Ausgängen folgende Frequenzen:

Teilverhältnis	2:1	4:1	8:1	16:1
Ausgang	Ausgang Q_0	Ausgang Q_1	Ausgang Q_2	Ausgang Q_3
Frequenz	50 Hz	25 Hz	12,5 Hz	6,25 Hz

5.6.2 Asynchroner 4-Bit BCD-Zähler

BCD-Zähler basieren auf einem 4-Bit-Binär-Zähler. Wandelt man Binärzahlen zur Anzeige in Dezimalziffern um, so lassen sich Binär-Zähler auch im Dezimalsystem einsetzen. Für die Zahlen 0 bis 9 jeder Dekade ist ein 4-Bit Binär-Zähler notwendig. Da dieser aber erst nach dem 16. Zähltakt auf 0 zurücksetzt, muss er durch eine Zusatzschaltung vorzeitig nach dem 10. Takt in einen neuen Zyklus gezwungen werden. Binärzähler mit diesen Eigenschaften werden BCD-Zähler (engl. binary coded decimal) genannt. BCD-Zähler gibt es als Vorwärts-, Rückwärts- und umschaltbare Zähler.

Die Grundsaltung des asynchronen BCD-Zählers ist der des asynchronen Binär-Zählers sehr ähnlich. Der BCD-Zähler muss jedoch schaltungstechnisch so erweitert werden, dass nach dem Zählerstand 9 mit 0 weitergezählt wird. Beachte, dass wir hier nur eine Dezimalstelle betrachten. Beim BCD-Zähler darf Q_1 nach dem 10. Eingangstakt nicht auf High schalten. Dazu wird dessen J-Eingang J_B mit dem \bar{Q} -Ausgang von Flipflop D (\bar{Q}_3) verbunden. Das Flipflop D schaltet erst nach dem 8. Takt am \bar{Q} -Ausgang von High auf Low. Bis dahin arbeitet Flipflop B wie ein Toggle-Flipflop und setzt nach dem 8. Takt auf Low zurück. Das Flipflop C arbeitet unverändert als Frequenzteiler.

Flipflop D wird vom \bar{Q} -Ausgang des Flopflops A getaktet. Es kann nur dann auf High gesetzt werden, wenn sein J-Eingang J_B über das AND-Gatter High-Pegel bekommt. Dies ist nach dem 6. Takt der Fall, sobald Q_1 und Q_2 gesetzt sind. Zu diesem Zeitpunkt ist aber der Takt Q_0 für das Flipflop D Low, sodass sein Ausgang nicht gesetzt wird. Mit der fallenden Flanke von Q_0 nach dem 8. Takt wird Q_3 gesetzt. Der Binärwert 1000 nach dem 8. Takt entspricht der Dezimalzahl 8. Zu diesem Zeitpunkt wechselt auch der Pegel am J-Eingang von Flipflop D auf Low. Zum Ende des 9. Takts hat das noch keine Auswirkung, da erst jetzt von Q_0 gesteuert das Flipflop D nach dem 10. Takt auf Low zurückgesetzt wird. Der Binärwert zeigt 0000 an.

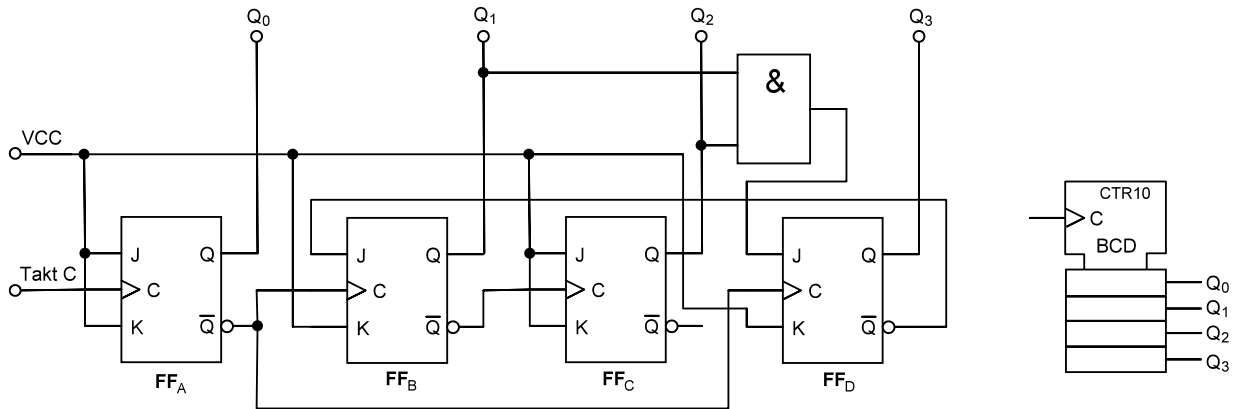
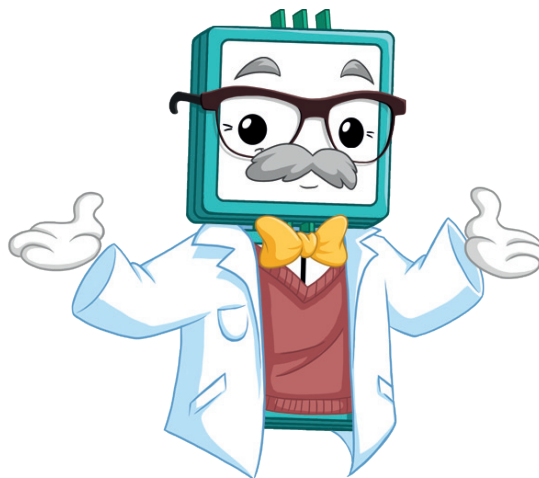


Abb. 52: Asynchroner 4-Bit-BCD-Zähler (links), Schaltzeichen (rechts)



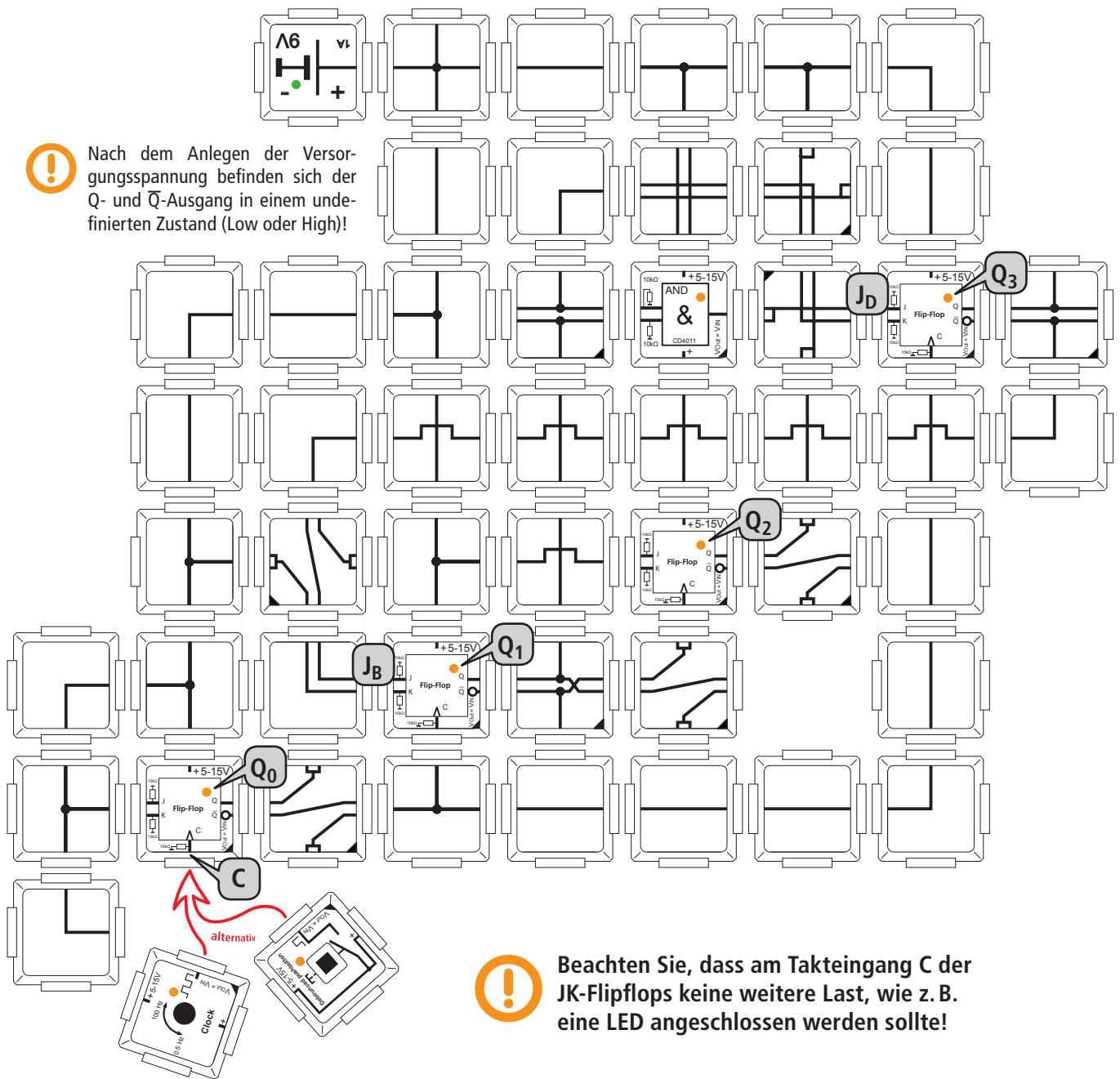


Abb. 53: Asynchroner 4-Bit-BCD-Zähler aus JK-Flipflops als Brick-Schaltung

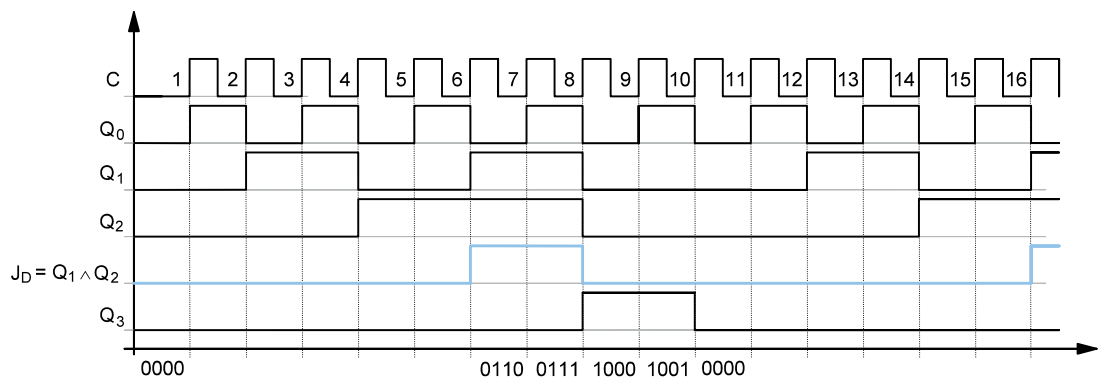


Abb. 54: Timingdiagramm asynchroner 4-Bit BCD-Zähler

5.6.3 Synchroner 3-Bit Binär-Zähler

Synchrone Zähler sind übersichtlicher aufgebaut und leicht erweiterbar. Ihr Schaltnetz ist aufwendiger, dafür treten keine Laufzeitprobleme auf.

Beim Taktwechsel bestimmen die Eingangsspiegel jeder Speicherstufe, ob das Flipflop neu gesetzt wird oder seinen vorherigen Zustand speichert. Da alle Informationen an den Eingängen schon zum Taktbeginn fest anliegen müssen, lassen sich Synchronzähler nicht mit T-Flipflops aufbauen. Auch RS-Flipflops sind nur bedingt tauglich, weil der Schaltungsaufwand wegen der Zusatzbeschaltung zu umfangreich wäre.

Am besten eignen sich JK-Master-Slave-Flipflops, deren J- und K-Eingänge über zusätzliche Gatter gesteuert werden. In unserem Beispiel kommen einflankengesteuerte JK-Flipflops zum Einsatz. Die J- und K-Eingänge des ersten Flipflops liegen dabei auf High-Pegel (entspricht Toggle-Betrieb). Der Ausgang steuert die verbundenen J- und K-Eingänge der Folgestufe. Die J- und K-Eingänge aller weiteren Speicher-Flipflops sind mit einem AND-Gatter verbunden, welches die Ausgangsspiegel der vorhergehenden Speicherstufe auswertet.

Wegen der besseren Übersichtlichkeit wird das folgende Beispiel für den Synchronzähler nur 3-Bit-breit ausgeführt. Damit können wir immerhin von 0 bis 7 zählen.

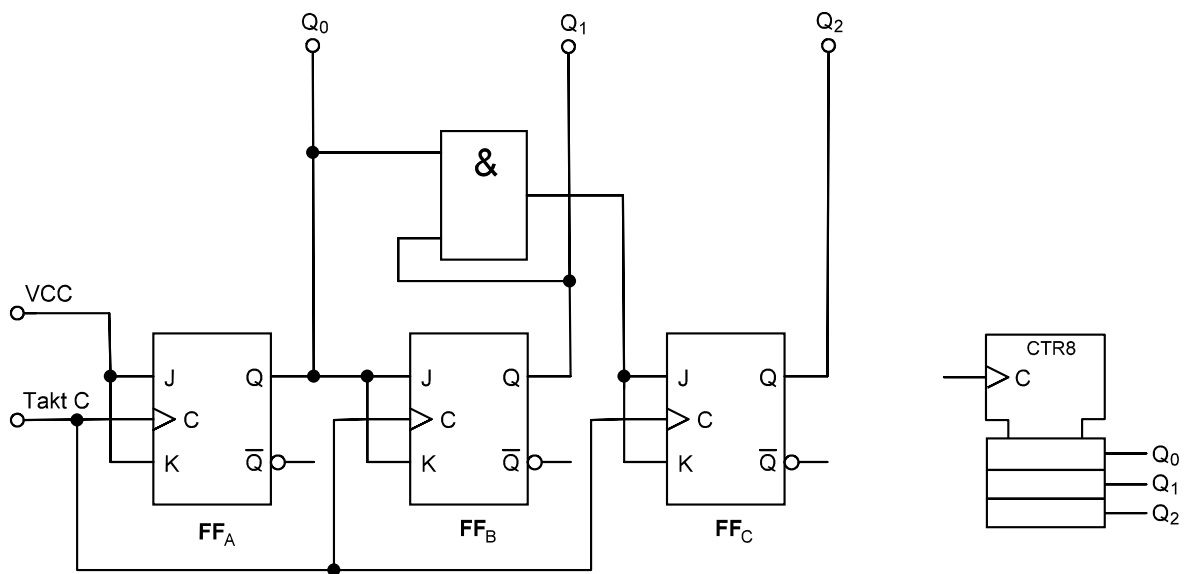


Abb. 55: Synchroner 3-Bit-Binär-Aufwärtszähler (links), Schaltzeichen (rechts)

Zustandstabelle für 3-Bit Synchron-Aufwärtszähler

Takt	Q ₂	Q ₁	Q ₀	Gleichung
1	0	0	0	
2	0	0	1	
3	0	1	0	
4	0	1	1	$Q_0 \wedge Q_1$
5	1	0	0	
6	1	0	1	
7	1	1	0	
8	1	1	1	

Der Zähler hat drei Speicher-Flipflops und kann binär von 000 bis 111 zählen. Mit Hilfe der Zustandstabelle für die Ausgänge Q₀ bis Q₂ kann die Gatterschaltung ermittelt werden. Das Flipflop A schaltet bei jedem Takt um, was einer Halbierung der Taktfrequenz entspricht. Das Flipflop B wird nur dann neu gesetzt, wenn der Ausgang Q₀ high ist, ansonsten wird der vorherige Zustand gespeichert. Das Flipflop C muss gesetzt werden sobald Q₀ und Q₁ gleichzeitig High-Pegel haben.

Die neuen Ausgangsspiegel erscheinen nach der steigenden Taktflanke. Alle Ausgangssignale werden synchron, also gleichzeitig gesetzt. Im Vergleich zu den Asynchronzählern addieren sich die Signallaufzeiten der einzelnen Speicherstufen nicht.



Beachten Sie, dass am Takteingang C der JK-Flipflops keine weitere Last, wie z. B. eine LED angeschlossen werden sollte!

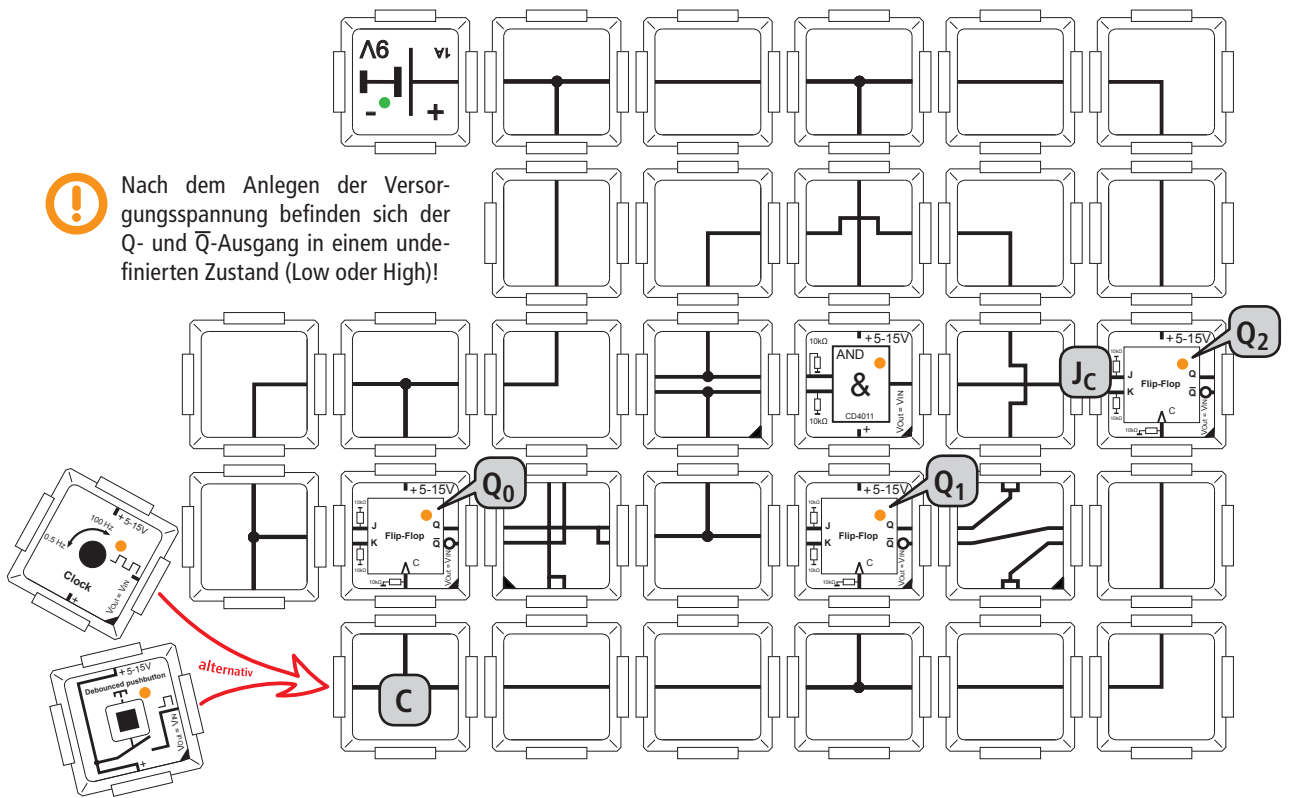


Abb. 56: Synchroner 3-Bit-Binär-Zähler aus JK-Flipflops als Brick-Schaltung

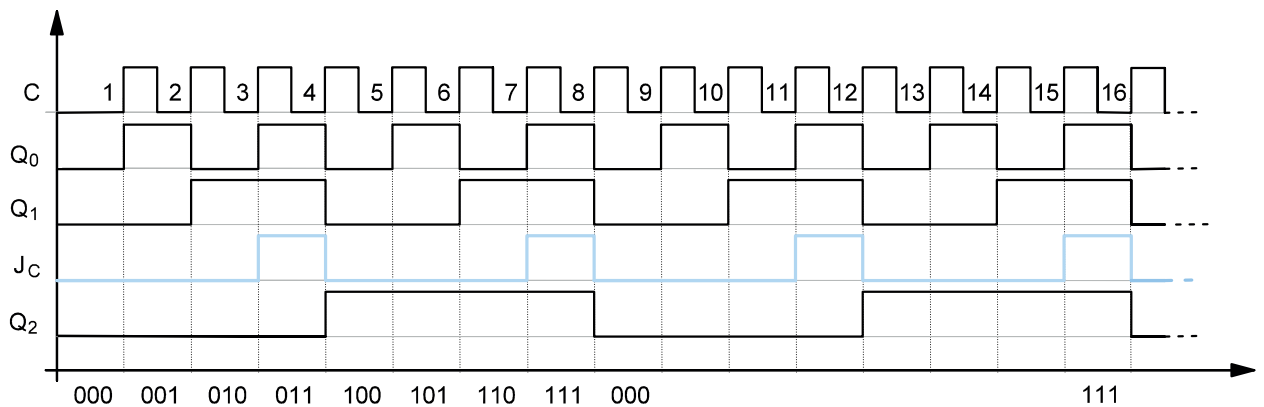


Abb. 57: Timing-Diagramm synchroner 3-Bit Binär-Aufwärtszähler

Variante: Abwärtszähler

Durch Ergänzung von LEDs an den \bar{Q} -Ausgängen, kann der Zähler auch als Abwärtszähler betrachtet werden.

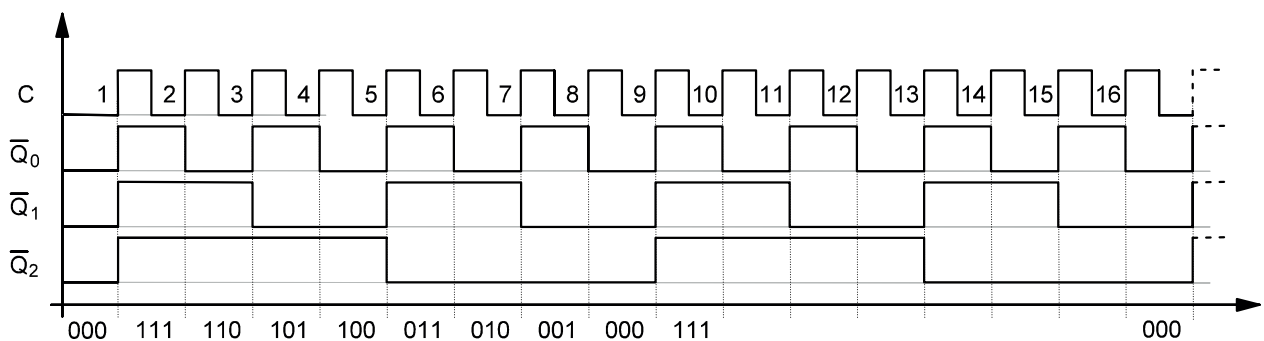


Abb. 58: Timing-Diagramm synchroner 3-Bit Binär-Abwärtszähler

5.6.4 BCD-Zähler mit Übertrag und Reset

In diesem Beispiel kommen die Brick'R'knowledge BCD-Counter-Brick zum Einsatz. Intern sind diese aus zwei synchronen BCD-Zählern vom Typ 4518 aufgebaut, die asynchron mittels Übertrags-Bit verbunden sind. Durch Reihenschaltung mehrerer BCD-Zähler kann man auch über mehrere Dezimalstellen (Dekaden) zählen. Die BCD-zu-7-Segment Decodierung ist im BCD-Counter-Brick bereits integriert.

Die folgende Abbildung zeigt die Verschaltung eines BCD-Zähler-Bricks mit zwei alternativen Möglichkeiten der Taktung. Links eine einfache Variante zur manuellen Erzeugung einzelner prellfreier Takte mit Hilfe des entprellten Tasters (Debounced pushbutton) und rechts die Variante mit dem Taktgeber-Brick (Clock), der einen periodischen Takt zwischen 0,5 Hz und etwa 100 Hz ausgeben kann.

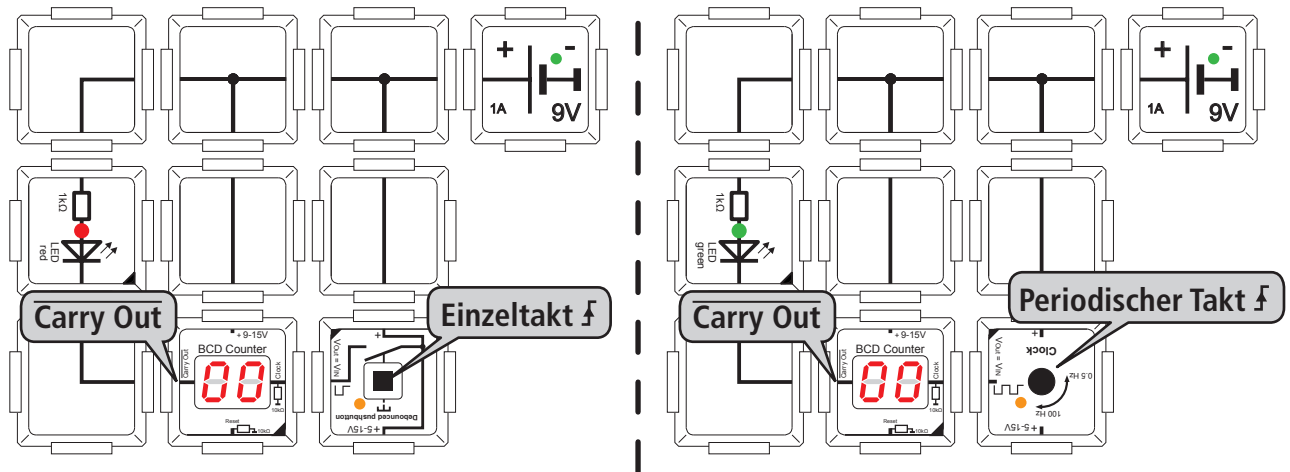


Abb. 59: BCD-Zähler mit entprelltem Handtakt (links) und mit Taktgeber-Brick (rechts)

Sobald die BCD-Zähler-Bricks den Maximalwert von 99 erreicht haben, geben sie einen Übertrag aus. Da dieses Übertragungssignal low-aktiv ist, müssen wir es korrekterweise "Carry Out" schreiben um die Negierung anzuzeigen. Da wir es in der folgenden Schaltung mit zwei Übertragsausgängen zu tun haben, schreiben wir CarryOut1 und CarryOut2 zur Unterscheidung.

In unserem Brick-Beispiel werden wir durch Kaskadierung von zwei BCD-Zähler-Bricks einen insgesamt 4-stelligen BCD-Zähler bauen. Das low-aktive Übertragungssignal Carry Out2 wird mit der grünen LED angezeigt. Die beiden asynchronen Reset-Eingänge sind high-aktiv und können über die Taste angesteuert werden. Das Blockschaltbild für unsere Schaltung schaut folgendermaßen aus:

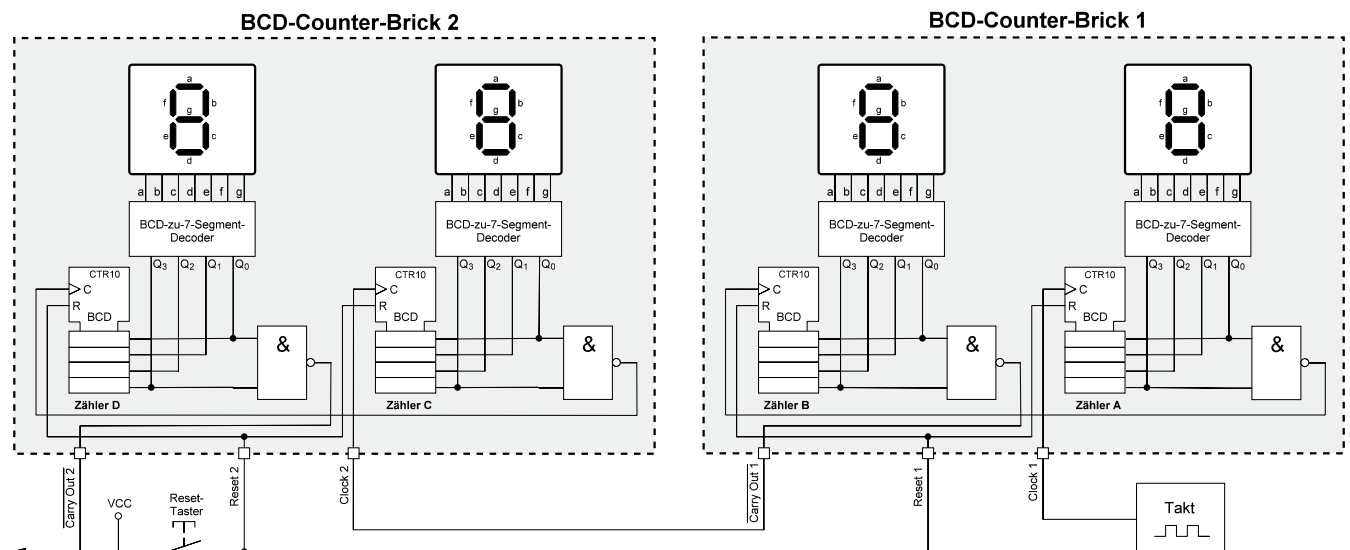


Abb. 60: Blockschaltbild eines 4-stelligen BCD-Zählers

Bedenken Sie, dass die Carry Out2-LED des linken, höherwertigen Zähler-Bricks bei maximaler Frequenz des Taktgebers zum erstenmal nach ca. 1 Minute, 40 Sekunden aktiv wird. D. h. der Carry Out2-Ausgang gibt für kurze Zeit Low-Pegel aus, sodass die LED kurz aufblitzt.

Der Takteingang "Clock" zählt stets mit der positiven Flanke des Taktsignals, also mit dem Übergang von Low- zu High-Pegel.

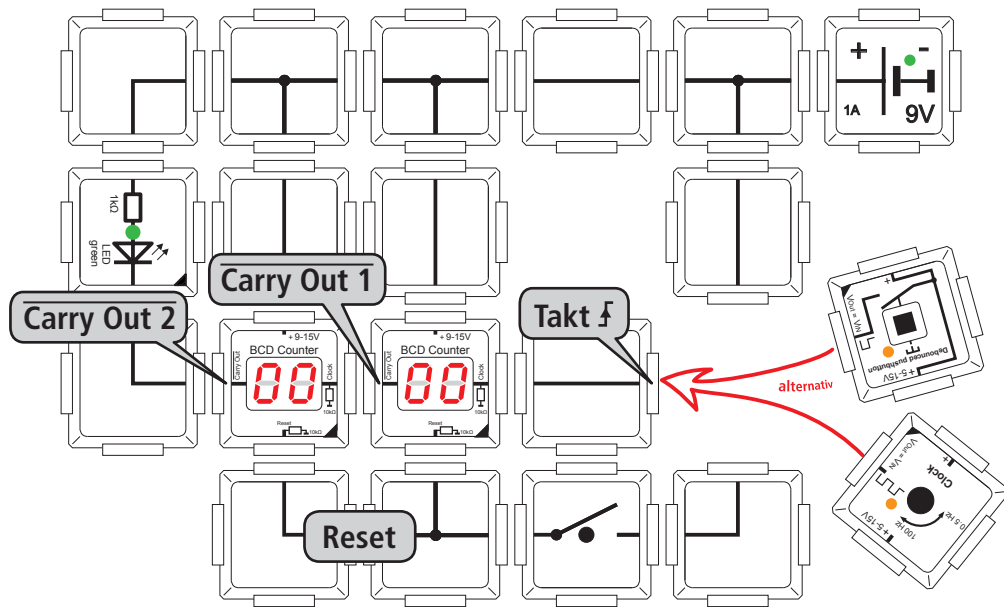


Abb. 61: 4-stelliger BCD-Zähler mit Carry-Ausgängen und Reset-Eingang

TIPP: Experimentieren Sie mit einer erweiterten Schaltung, die es Ihnen erlaubt, zwischen Taktgeber-Brick und dem entprellten Taster (Debounce pushbutton) umzuschalten. Versuchen Sie es mit einem OR-Gatter und überlegen Sie, wie der Taktgeber ganz einfach abgeschaltet werden könnte.

Das Übertragungssignal (Carry Out)

Im Falle unserer BCD-Zähler-Bricks ist das Carry Out-Signal stets 10 Takte lang aktiv, also low. Wie man im folgenden Timing-Diagramm erkennen kann, kündigt sich das Übertragereignis bereits beim Zählerstand von 90 an, indem der Carry-Ausgang auf low geht und beim Übergang von 99 auf 00 wieder high, also inaktiv. Mit dieser positiven Flanke an Carry Out 1 wird in unserem Beispiel der Takt-Eingang (Clock) des höherwertigen Zähler-Bricks (siehe auch Clock 2 an Zähler C im Blockschaltbild, Abb. 60) getaktet.

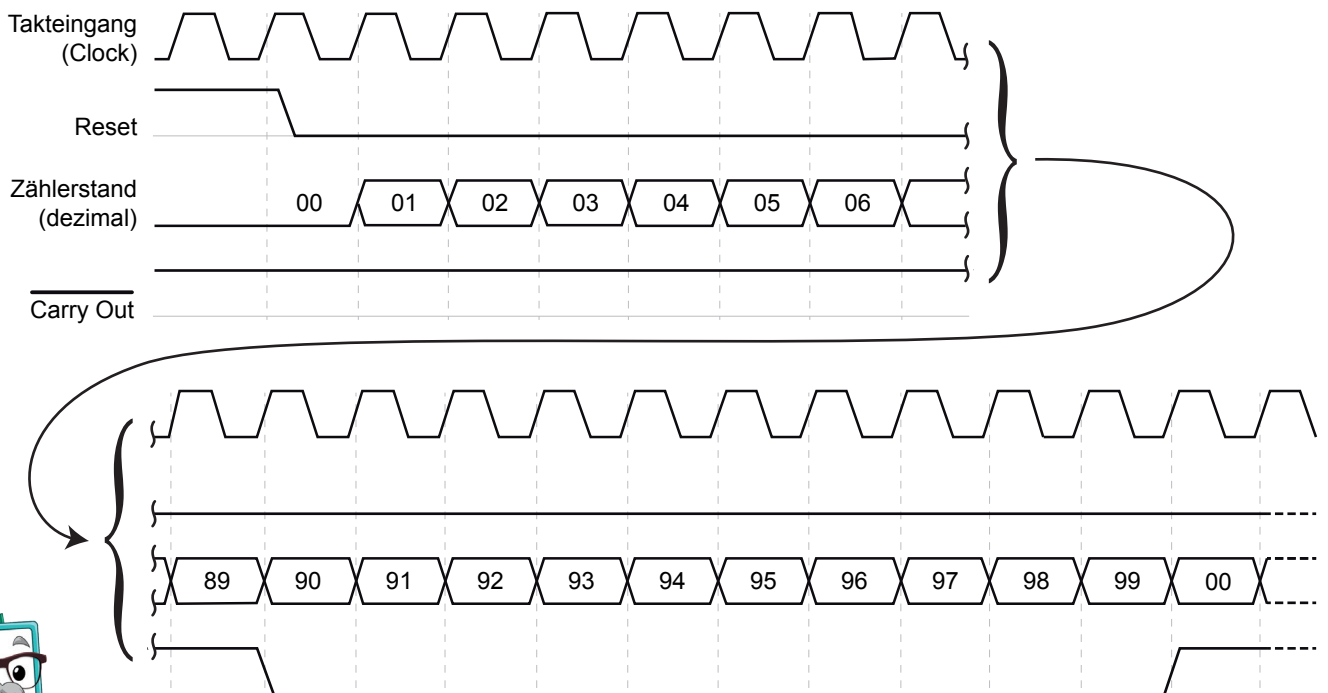


Abb. 62: Timing-Diagramm des BCD-Zähler-Bricks

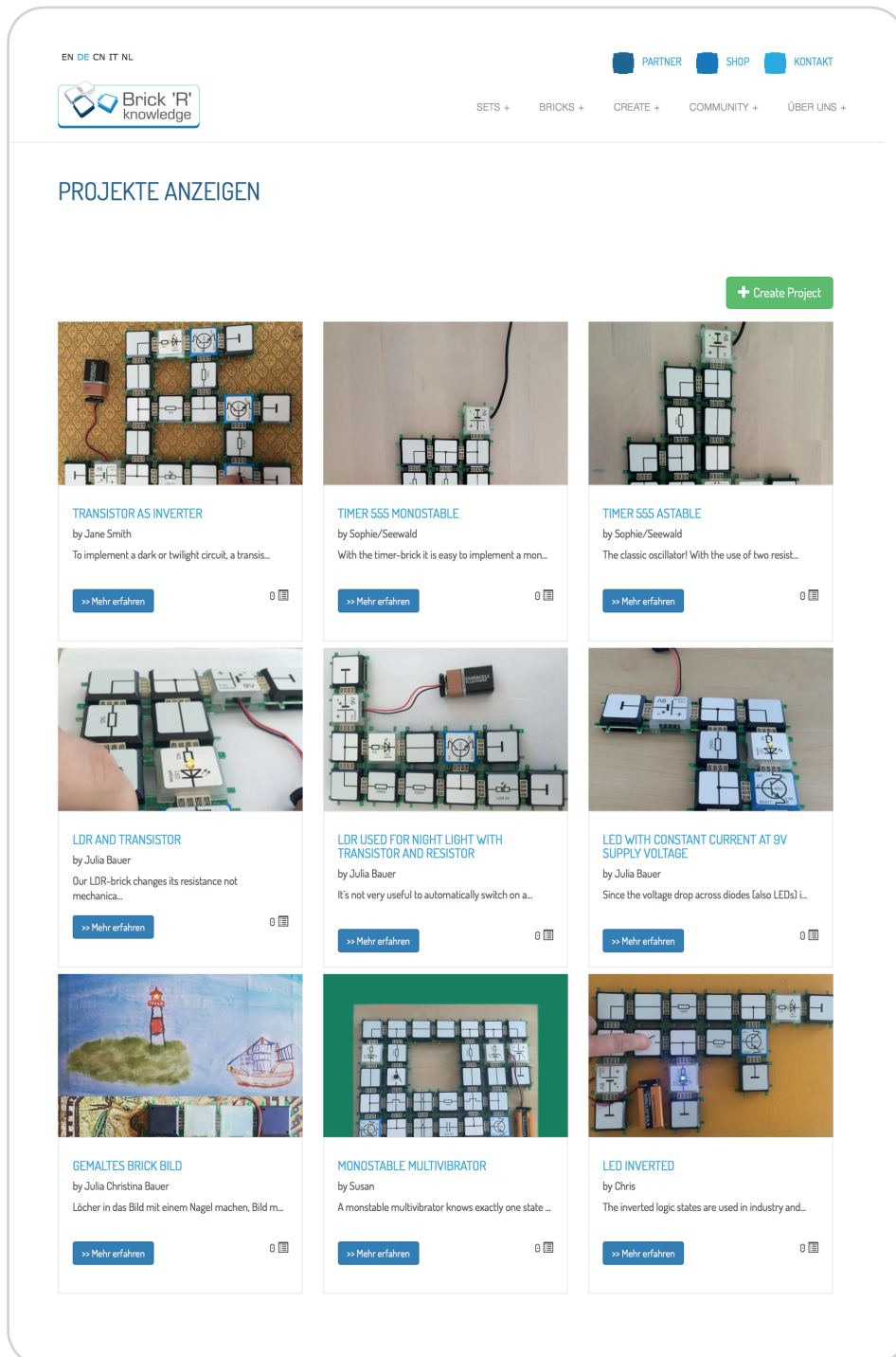


6. Brick Community

Das Brick-Universum dehnt sich aus: Ob auf Messen, auf unserer Website, auf YouTube oder in den sozialen Medien, überall finden Sie weitere Anregungen, Experimente und neue Bricks, mit denen Sie Ihrer Kreativität freien Lauf lassen können!

Mehr Projekte

Im Reiter „Create“ können Sie Projekte und Schaltungen von anderen Community Mitgliedern ausprobieren, nachbauen und verbessern. Natürlich können Sie der Welt auch Ihre eigenen Experimente zeigen.



The screenshot shows the 'Brick 'R' knowledge' website interface. At the top, there are navigation links for 'PARTNER', 'SHOP', and 'KONTAKT', along with a menu for 'SETS +', 'BRICKS +', 'CREATE +', 'COMMUNITY +', and 'ÜBER UNS +'. The main content area is titled 'PROJEKTE ANZEIGEN' and features a grid of project cards. Each card includes a photo of a project, a title, the author's name, a brief description, and a 'Mehr erfahren' button. A '+ Create Project' button is visible in the top right of the grid area.

Project Title	Author	Description
TRANSISTOR AS INVERTER	Jane Smith	To implement a dark or twilight circuit, a transis...
TIMER 555 MONOSTABLE	Sophie/Seewald	With the timer-brick it is easy to implement a mon...
TIMER 555 ASTABLE	Sophie/Seewald	The classic oscillator! With the use of two resist...
LDR AND TRANSISTOR	Julia Bauer	Our LDR-brick changes its resistance not mechanica...
LDR USED FOR NIGHT LIGHT WITH TRANSISTOR AND RESISTOR	Julia Bauer	It's not very useful to automatically switch on a...
LED WITH CONSTANT CURRENT AT 9V SUPPLY VOLTAGE	Julia Bauer	Since the voltage drop across diodes (also LEDs) L...
GEMALTES BRICK BILD	Julia Christina Bauer	Löcher in das Bild mit einem Nagel machen, Bild m...
MONOSTABLE MULTIVIBRATOR	Susan	A monstable multivibrator knows exactly one state ...
LED INVERTED	Chris	The inverted logic states are used in industry and...



Social Media

Im Reiter „Community“ finden Sie unsere Social-Media-Präsenzen und bleiben so immer Up-to-date!

FACEBOOK

Brick 'R' knowledge shared a video 2 days ago

Es gibt ein neues Set! Wir sind gespannt, welche Projekte ihr mit dem Brick'R'knowledge RGB Color Light Set kreiert! <https://www.youtube.com/watch?v=X3pVDZ0XTIA&feature=youtu.be>

Brick 'R' knowledge shared a video 2 days ago

Heute haben wir ein neues Brick- und Arduino.org-Experiment für euch, kreiert von unserem Praktikanten Mattia. Viel Spaß beim Nachbauen! <https://www.youtube.com/embed/5ILiJCt4HRg>

TWITTER

Dank an unseren #Praktikanten Mattia für dieses tolle #Brick- und @ArduinoOrg #Experiment: <https://t.co/eINdCj05b2> <https://t.co/dj0zV0VFu>

Vielen Dank @code_your_life für die tolle Show in der #ufafabrik. Wir waren begeistert! #kids #coding with #bricks <https://t.co/0FFk8z9IYf>

Kids @ #codeyourlife #Brick'R'knowledge <https://t.co/Wtm9llcVeu>

#Brick'R'knowledge an der #TU #Berlin: Wir haben den #Studenten unsere #Bricks vorgestellt <https://t.co/gvJkX6Zug0> <https://t.co/ct0AOxa5qq>

INSTAGRAM

PINTEREST

THANKS TO OUR...

ARDUINO CODING SET

IN CASE YOU MISSED...

CHARGE YOUR MOBILE...

TEST YOUR REACTION...

NEVER BORING WITH...

WWW.MAKER-STORE.DE

THE POWER OF THE ...

Weltweit

Ebenfalls im Reiter „Community“ können Sie sehen, wo es überall schon Brick-Mitglieder gibt, wo wir gerade sind oder mit welchem Wahrzeichen die Bricks schon fotografiert wurden. Hier können Sie uns auch Ihr Brick-Bild zusenden und Sie werden es bald auf der Weltkarte finden!

EN DE CN IT NL

PARTNER
 SHOP
 KONTAKT

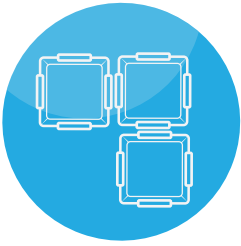
SETS +
BRICKS +
CREATE +
COMMUNITY +
ÜBER UNS +

BRICK'S WORLD

Dein Bild auf der Weltkarte? Einfach eine E-Mail an info@brickrknowledge.de oder Facebook Nachricht mit deinem Vornamen, dem Brick-Bild, Stadt und Land senden und schon bist du ein Teil der Brick World!

WW

Noch mehr Bricks!



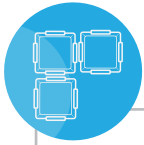
Im Reiter „Bricks“ finden Sie alle verfügbaren Bausteine zum Erweitern Ihrer Schaltungen und Experimente.

Der Brick Blog



Jede Woche gibt es unter „Community“ einen neuen Blog-Post. Sie finden hier Messeberichte, neue Experimente, witzige Geschichten und Informationen über neue Sets und Bricks.

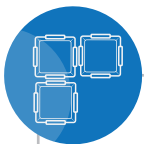
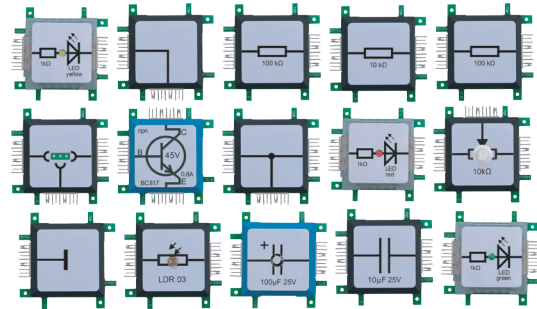
7. Brick Sets im Überblick



Basic Set enthält 19 Bricks

ALL-BRICK-0374

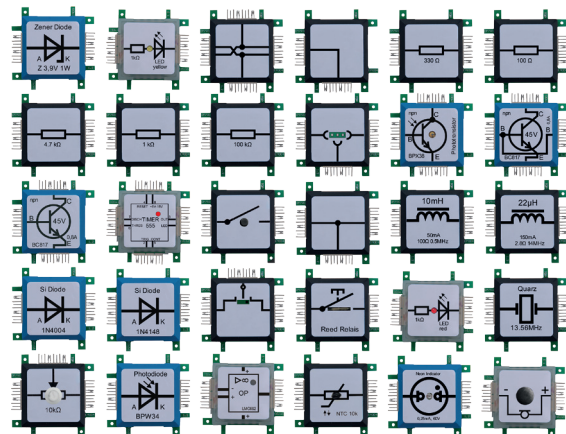
Das Basic Set bietet mit den 19 enthaltenen Bricks einen schnellen Einstieg in die Brick 'R' knowledge Welt und ermöglicht bereits eine Vielzahl von Experimenten. Mit der Basic-Variante können schon junge Entwickler eigene Schaltungen bauen und so ihre ersten physikalischen und technischen Experimente durchführen.



Advanced Set enthält 111 Bricks

ALL-BRICK-0223

Mit 111 Teilen bietet das Advanced Set alles, was zur Veranschaulichung komplexer elektronischer Schaltungen benötigt wird. Unter den über 100 Beispielschaltungen finden sich auch zahlreiche Anwendungen, die wir aus dem Alltag kennen. Das Set wurde so zusammengestellt, dass es auch von Ingenieurbüros zur kostengünstigen Visualisierung im Rahmen von Rapid-Prototyping genutzt werden kann.



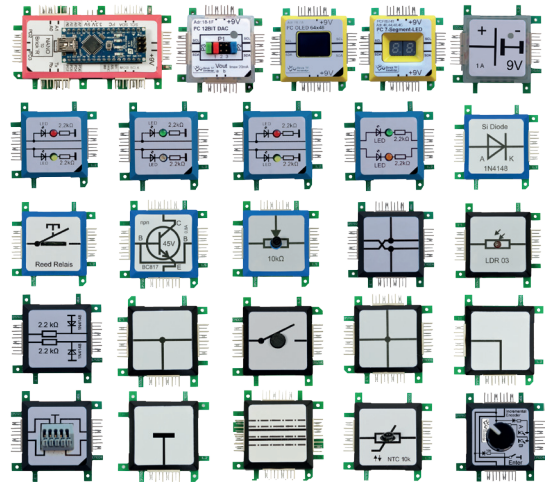


Arduino Coding Set

enthält 44 Bricks

ALL-BRICK-0414

Das Brick'R'knowledge Arduino Coding Set erweitert die Experimente hin zur Digitalelektronik mit der Einführung in die Microcontroller-Programmierung am Beispiel des Arduino Nanos. Neben Bricks für analoge Schaltungen enthält das Set auch Bricks für digitale Anwendungen wie eine 7-Segmentanzeige, ein OLED-Display, einen D/A-Wandler, einen I²C-Brick zur Pin-Erweiterung des Arduino Nanos, einen Arduino Nano Adapter-Brick und natürlich auch den Arduino Nano. Neben der Beschreibung der Experimente werden auch alle Programmierbeispiele zur Verfügung gestellt, um in die Welt der Arduino-Programmierung einsteigen zu können.

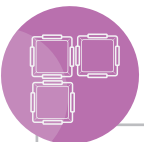
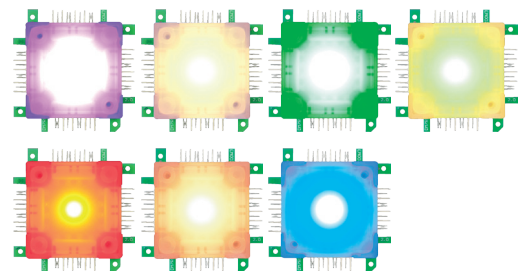


7 Color Light Set

enthält 28 Bricks

ALL-BRICK-0398

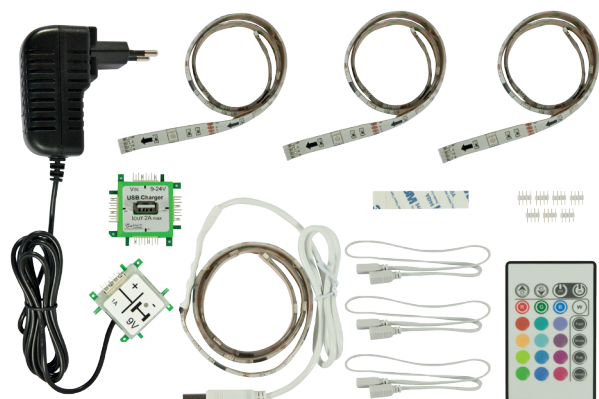
Mit den insgesamt 28 LED-Leucht-Bricks in 7 unterschiedlichen Farben lassen sich beeindruckende Lichtakzente in horizontaler und vertikaler Architektur setzen. Die 1 Watt LEDs in den Farben rot, gelb, blau, orange, violett, grün und warmweiß eignen sich perfekt für individuelle Licht-Figuren oder als mobile Beleuchtungslösung.

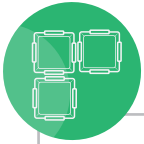


RGB Color Light Set

ALL-BRICK-0619

Erschaffen Sie Ihre eigene Licht-Show! Das RGB Color Light Set enthält vier flexible LED-Streifen mit insgesamt 36 LEDs, die mit einer Infrarot-Fernbedienung angesteuert werden können. Die LED-Streifen können so geklebt, zugeschnitten und verbunden werden, wie Sie es wünschen. Die Infrarot-Fernbedienung hat 16 verschiedene Farbknöpfe und 4 Licht-Programme.



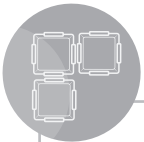
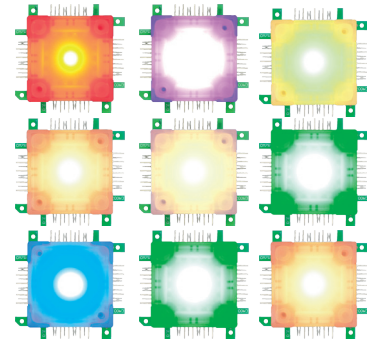


Programmable LED Set

enthält 49 Bricks

ALL-BRICK-0483

Das Set beinhaltet 49 ansteuerbare RGB-LED-Bricks mit zwei oder drei Anschlüssen, sowie einen Anschlussbrick für die Arduino-Steuerung und die Stromversorgung, einen Arduino Adapter-Brick und einen Arduino Nano. Das Set ermöglicht es, eigene LED-Animationen als Farb- oder auch bewegte Bildanimationen zu erstellen und sich spielerisch mit Microcontroller-Programmierung zu befassen. Innovative Lichtinstallationen und individuell leuchtende, blinkende und pulsierende Bilder in unterschiedlichen Farb- und Helligkeitsstufen sind durch das Programmable LED Set wunderbar umsetzbar.

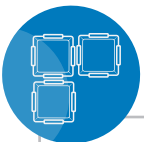
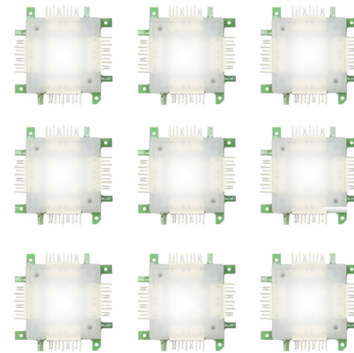


Highpower LED Set

enthält 50 Bricks

ALL-BRICK-0399

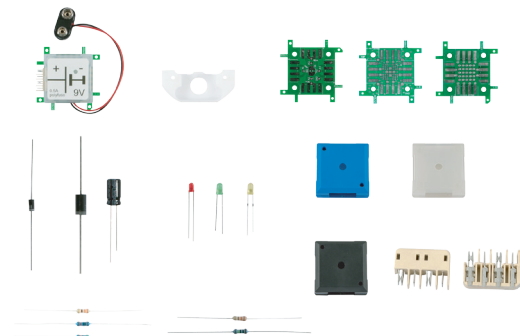
Das strahlende High Power LED Set enthält fünfzig 1 Watt High-Power-Bricks und dazu noch ein 12 Volt, 8 Ampere Netzteil. Die Bricks lassen sich ganz einfach zu individuellen Lösungen zusammenstecken. Zum Beispiel lassen sich aus den Bricks verschiedene Tischlampen bauen, die dann erweiterbar sind. Durch die starke Leuchtkraft bietet dieses ein stilvolles Ambiente und eignet sich perfekt als Nachtlcht. Das High Power LED Set ermöglicht es, sich spielerisch mit Lichtdesign auseinander zu setzen.



DIY Set

ALL-BRICK-0397

Das „Do-it-yourself“ Set ermöglicht es Tüftlern und Entwicklern, ihre eigenen Bricks in Ergänzung zu den bereits vorhandenen zu bauen. Die hier enthaltenen Komponenten bieten einen tiefen Einblick in Aufbau und Architektur der elektronischen Bauelemente. Mit LötKolben und LötZinn können die Tüftler die Standard-Bricks nachbauen oder eigene Bricks für individuelle Spezialanwendungen herstellen und somit sogar eigene Sets entwickeln.

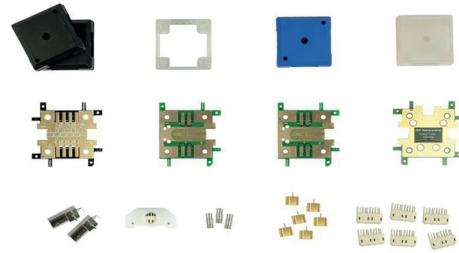




MHz DIY Set

ALL-BRICK-0457

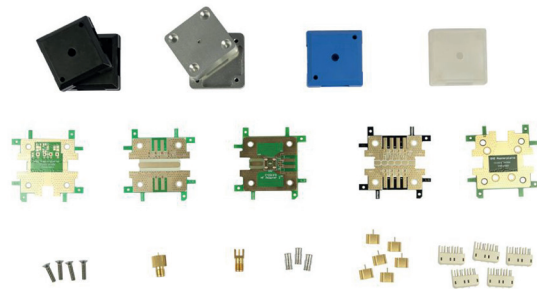
Mit dem MHz DIY Set lassen sich eigene Bausteine für Experimente und Schaltungen im MHz-Bereich realisieren. Das Set enthält drei verschiedene Raster- und Experimentierplatinen, sowie BNC-Buchsen, P-SMP-Stecker und die dazu passenden Verbinder. Außerdem enthält das Set eine Lötlehre für die SMD-Stecker und hermaphrodite Steckverbinder, um Eigenentwicklungen an das Brick-System anzupassen.



GHz DIY Set

ALL-BRICK-0458

Das GHz DIY Set bietet spannende Möglichkeiten zur Entwicklung im Hochfrequenzbereich bis hin zu Gigahertz-Frequenzen. Neben vier verschiedenen Platinen kann das GHz DIY Set auch mit verschiedensten Komponenten, wie liegenden und stehenden SMA- und P-SMP-Koaxialverbindern und den zum Brick-System gehörenden Steckverbindern dienen. So eignet sich das Set besonders für Messtechnik-Fans und Funkamateure.



Solar Set

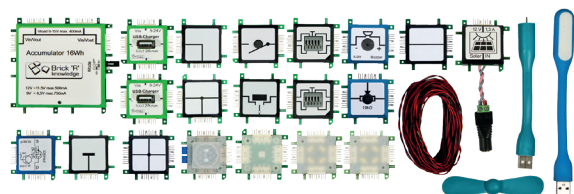
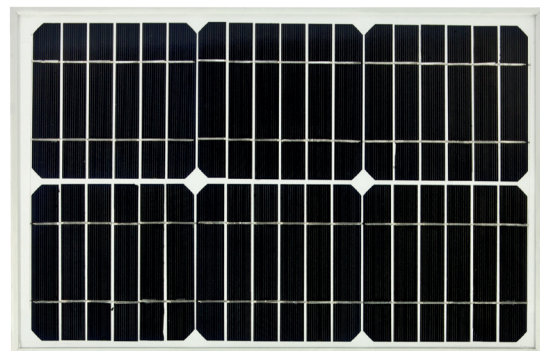
enthält 20 Bricks

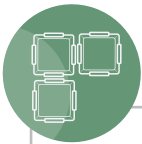
ALL-BRICK-0484

Das Solar Set von Brick'R'knowledge garantiert Experimentierspaß für die ganze Familie und bringt Kindern erneuerbare Energien auf spielerische Art und Weise näher.

- Wie funktioniert eine Solarzelle?
- Wie speichert ein Akku Strom?
- Wie baut man ein Nachtlicht mit Bewegungsmelder?

Auf diese und weitere Fragen gibt das Solar Set Antworten. Mit diesem Set sind Sie und Ihre Kinder offizielle Mitglieder der Maker-Generation.



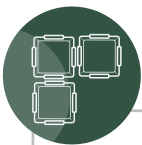
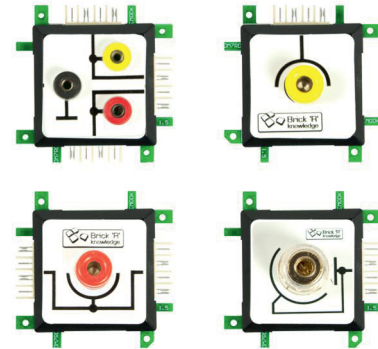


Measurement Set One

enthält 4 Bricks

ALL-BRICK-0637

Das Set ermöglicht es, mit Standardmessgeräten in Brick'R'knowledge Schaltungen Spannung, Stromstärke und andere Messgrößen einfach zu ermitteln. Das Messadapter-Set besteht aus folgenden Bricks: einem Messadapter mit 3 x 2 mm Buchse, einem Messadapter mit 4 mm Closed End GND in schwarz mit zusätzlicher Kabelklemme, einem Messadapter mit 4 mm Endpoint in gelb und einem Messadapter mit 4 mm Inline in rot.

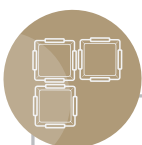
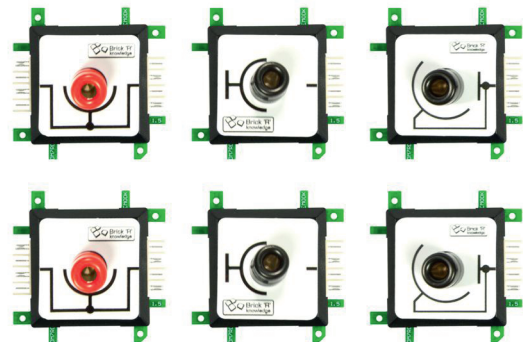


Measurement Set Two

enthält 6 Bricks

ALL-BRICK-0638

Das Set ermöglicht es, mit Standardmessgeräten in Brick'R'knowledge Schaltungen Spannung, Stromstärke und andere Messgrößen einfach zu ermitteln. Das Messadapter-Set besteht aus folgenden Bricks: zwei Messadapter mit 4 mm Closed End GND in schwarz, zwei Messadapter mit 4 mm Inline in rot und zwei Messadapter mit 4 mm Open End GND in schwarz.



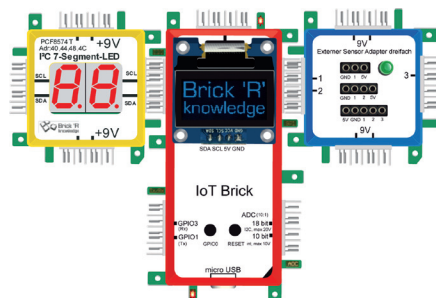
Internet of Things Set

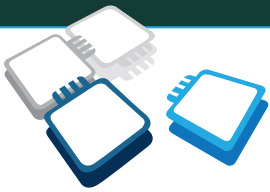
enthält 13 Bricks

ALL-BRICK-0646

Mit dem Internet of Things Set ist es nun möglich, die Bricks via Internet zu kontrollieren. Mit dem zentralen IoT-Brick werden Sie beispielsweise lernen, Ihre erste Website zu bauen und I/O Pins mit Ihrem Smartphone zu steuern. Außerdem enthält das Set einen Temperatur- und Luftfeuchtigkeitssensor, dessen Werte Sie auf einem Display darstellen können: Der erste Schritt zur eigenen Home Automation!

Sie können auch Daten, wie zum Beispiel den Dollar-Kurs aus dem Internet abfragen und sich anzeigen lassen. Um die 7-Segmentanzeige anzusteuern, wird der sogenannte I²C-Bus genutzt, den Sie auch kennenlernen. Das Internet der Dinge wartet darauf, von Ihnen entdeckt zu werden!





Brick 'R' knowledge



ALLNET® GmbH Computersysteme

Maistrasse 2
D-82110 Germering
www.brickrknowledge.com

Telefon: +49 (0)89 894 222 921

Fax: +49 (0)89 894 222 33

info@brickrknowledge.com



Maker Store & Maker Space

Danziger Straße 22
D-10435 Berlin
www.maker-store.de

Telefon: +49 (0)30 473 756 80

service@allknow.de

