

**Grundlegendes Hardware-Table des Inhalts**

<b>Thema</b>	<b>Seite</b>
Inhaltsverzeichnis	1
Arbeiten mit dem Binärsystem	2
Das Binärsystem	2-3
Bits und Bytes	3
Umwandeln binär in Dezimalstrich	3-4
Umwandelnder Dezimalstrich zu binärem	4
Binäre Hinzufügung	4-5
Binärer Abzug	5-6
Einleitung in eine Logikgatter	6-7
Grundlegende LogikGatter	7
Das Gatter AND	7-8
Das Gatter OR	8
Das Gatter NOT	8
Andere LogikGatter	9
Das Gatter NAND	9-10
Das Gatter NOR	10
Das Gatter XOR	10-11
Innere Gatter	11
Das Gatter NOT	12
Das Gatter AND	12
Das Gatter OR	13
Binäre Additionsmaschinen	13
Der Binäre Halbaddierer	13-14
Das Binäre Full-Adder	14-15
Parallele Binäre Additionsmaschine	15-16
Boolesche Algebra	17
Logikgatter und ihre Booleschen Ausdrücke	18
Bilden einer Wahrheitstabelle mit einer Gleichung	19

## Arbeiten mit dem Binärsystem

Wir sind alle mit dem dezimalen System vertraut. Wir verwenden 10 Digits in den unterschiedlichen Kombinationen, um Zahlen darzustellen. Diese Ziffern sind selbstverständlich 0.1.2.3.4.5.6.7.8 und 9. Der Computer benutzt andererseits das binäre System. Dieses System ist viel einfacher als unser vertrautes dezimales System. Anstatt, 10 Digits zu verwenden, verwendet es nur 0 und 1.

Die vier Grundrechenarten (Hinzufügung, Abzug, Vermehrung und Abteilung) können bis zwei verringert werden - Hinzufügung und Abzug. Vermehrung ist wirklich wiederholte Hinzufügung, während Abteilung wirklich wiederholter Abzug ist. Diese Tatsache macht es möglich, diese Operationen in der binären Form anzufassen.

## Das Binärsystem

um eine dezimale Zahl in eine Binärzahl umzuwandeln, müssen wir einige Sachen im Verstand halten. Z.B. im dezimalen System, bedeutet die Nr. 151 wirklich dieses:

Hunderte	10	Maßeinheiten
1	5	1

So kann die Nr. 150 wie auch dargestellt werden  $(1 * 10^2) + (5 * 10^1) + (1 * 10^0)$ . Das 1 ist im Hundertplatz. Die fünf ist im 10platz, und das null ist im Maßeinheitsplatz. Jede Position im dezimalen System steht für einen Wert, der mit einer anderen Energie von 10 multipliziert wird. Deshalb gekennzeichnet das dezimale System manchmal als die **niedrigen 10** System. Die Zahlen  $10^2$ ,  $10^1$  und  $10^0$  stellen die Stellenwerte der Digits die Zahlen dar, mit denen die Digits multipliziert werden. Ist hier eine Tabelle zum Zeigen Ihnen, was dieses Mittel mit der Nr. 150 als Beispiel:

	Hunderte	10	Maßeinheiten
Digit	1	5	1
Bedeutung	$1 * 10^2$	$5 * 10^1$	$1 * 10^0$
Wert	100	50	1

Ähnlich kann das binäre System in der gleichen Weise, ausgenommen genau dargestellt werden, die Stellenwerte sind die Energien von zwei. So wird das binäre System manchmal die niedrigen **zwei benannt** System. Da binäre Dose nur die Digits 1 und 0 haben, verwenden wir die Nr. 110101 als Beispiel:

Digit	1	1	0	1	0	1
Bedeutung	$1 \cdot 2^5$	$1 \cdot 2^4$	$0 \cdot 2^3$	$1 \cdot 2^2$	$0 \cdot 2^1$	$1 \cdot 2^0$
Wert	32	16	0	4	0	1

### Bits und Bytes

Obleich ein Bit das kleinste Formular von Daten ist, die ein Computer, dort Namen für größere Mengen Bits sind. Z.B. wird 8 Bits betrachtet, 1 Byte zu sein. Mit diesen Informationen im Verstand, ist hier eine Tabelle, der Erscheinen Sie die unterschiedlichen Daten die Namen sortieren, die durch den Computer verwendet werden, und wieviele Bits jedes ist:

Name	Abbrivation	Zahl von Bytes (als Energie)	Zahl von Bytes
Byte	-	$2^0$	1
KiloByte	KB	$2^{10}$	1.024
MegaByte	Mb	$2^{20}$	1.048.576
GigaByte	GB	$2^{30}$	1.073.741.824
TeraByte	TB	$2^{40}$	1.099.511.627.776
PetaByte	PB	$2^{50}$	1.125.899.906.842.624
ExaByte	Eb	$2^{60}$	1.152.921.504.606.846.976
ZettaByte	ZB	$2^{70}$	1.180.591.620.717.411.303.424
YottaByte	YB	$2^{80}$	1.208.925.819.614.629.174.706.176

Wie Sie sehen können, geht jeder Jobstep oben durch  $2^{10}$  Anmerkung, müssen Sie nur Byte bis zu GigaByte kennen, weil die anderen Größen nicht noch (Computer haben nicht genug vorgerückt, um zu können, sie zu verwenden), verwendet werden, obgleich es einige harte Laufwerke gibt, die eine Speicherung eines TeraByte oder mehr haben.

### Umwandeln binär in Dezimalstrich

Eine Binärzahl in eine dezimale Zahl umzuwandeln ist einfach. Steht für, um darzustellen aus was diese Zahl, müssen wir Zahlen an jedem Stellenwert der Binärzahl oben hinzufügen. Lassen Sie uns die Binärzahl 110101 als Beispiel verwenden:

$1 * 32 =$	32
$1 * 16 =$	16
$0 * 8 =$	0
$1 * 4 =$	4
$0 * 2 =$	0
$1 * 1 =$	1
<b>Gesamtmenge</b>	<b>53</b>

Antwort - 110101 in binärem ist 53 im Dezimalstrich.

### Umwandelnder Dezimalstrich zu binärem

Eine dezimale Zahl in binäres umzuwandeln ist auch sehr einfach. Es bezieht eine ununterbrochene Abteilung der dezimalen Zahl und Haltenspur des Restes mit ein. Lassen Sie uns die dezimale Nr. 220 als Beispiel verwenden:

Abteilung	Rest
$220 / 2 = 110$	0 Wenig Bedeutendes Bit (Lsb)
$110 / 2 = 55$	0
$55 / 2 = 27$	1
$27 / 2 = 13$	1
$13 / 2 = 6$	1
$6 / 2 = 3$	0
$3 / 2 = 1$	1
$1 / 2 = 0$	1 - Das Meiste Bedeutende Bit (Msb)

Antwort - 220 im Dezimalstrich ist 11011100 in binärem.

Das bedeutendste Bit ist genau was der Name schließt, das bedeutendste. Dies heißt, daß es den größten Wert aus allen Bits heraus hat. Da das MSB das Letzte auf der Tabelle ist, müssen wir das letzte Bit von der Tabelle zuerst plazieren und bearbeiten unsere Weise oben. Folglich ist die dezimale Nr. 220 11011100 im Dezimalstrich, nicht 00111011.

### Binäre Hinzufügung

Nun da wir binäres in Dezimalstrich umwandeln können und umgekehrt, ist es Zeit, zu erlernen, wie man zwei binäre zusammen Zahlen hinzufügt. Binärer Hinzufügung ist die gleiche Weise wie dezimale Hinzufügung erfolgt, die die tragenmethode

verwendet. Ist hier ein Beispiel des Verwendens der tragen Methode, um zwei dezimale Zahlen oben hinzuzufügen:

$$\begin{array}{r} 1 \\ 3248 \\ + 1426 \\ \hline 4674 \end{array}$$

Wie Sie sehen können, als 8 und 6 hinzugefügt wurden, wurden 4 als das Maßeinheitsdigit für die Antwort gespeichert, und 1 wurde zur 10spalte übertragen. Ist hier ein Beispiel von, wie binäre Zahlen hinzugefügt werden.

Für das binäre Hinzufügungs- Beispiel werden die binären Nr. 1001 und 0101 hinzugefügt:

$$\begin{array}{r} 1 \\ 1001 \\ + 0101 \\ \hline 1110 \end{array}$$

Wir fügen zuerst die zwei wenig bedeutenden Bits von den zwei hinzuzufügenden Zahlen, von 1 und von 1 hinzu. Rufen Sie wieder auf, dem binäre Dose nur 0 haben und  $1 + 1$  als 0 in den 2<sup>0</sup> Spalte <sup>gespeichert</sup> würden, und 1 würde bis die folgenden 2 getragen werden müssen <sup>1</sup> Spalte. Wir fügen dann  $0 + 0 +$  tragen von 1 hinzu und erhalten einer Antwort von 1 in den 2<sup>1</sup> Spalte. Zunächst fügen wir das 0 und das 1 hinzu und erhalten eine Antwort von 1 in den 2<sup>2</sup> Schließlich fügen wir oben das 1 und das 0 hinzu und erhalten eine Antwort von 1 in den 2<sup>3</sup> Die abschließende Antwort wird dann aufgedeckt, um 1110 zu sein. Wie Sie sehen können, ist binäre Hinzufügung für Leute sehr einfach, die die größeren dezimalen zusammen hinzufügen können Zahlen.

### Binärer Abzug

Es gibt einige Weisen, binäre Zahlen zu subtrahieren, aber, jedes mögliches Durcheinander zu vermeiden, verwenden wir die borgen Methode, die zu Ihnen bereits vertraut ist. Lassen Sie uns dezimalen Abzug mit einem Beispiel erinnern an:

$$\begin{array}{r} 21 \\ 3248 \\ - 1426 \\ \hline 1822 \end{array}$$

Wenn wir die Zahlen in der Maßeinheitsspalte subtrahieren, sowie die 10spalte, gibt es kein Problem. Wenn wir an die Spalte der 100 gelangen, Begriff, wie die Zahl auf die Oberseite kleiner ist, als die Zahl auf der Unterseite. um mit dem Abzug fortzufahren, borgen wir von der Spalte 1000's und legen sie in die Spalte der 100. Jetzt erhalten wir  $12 - 4$  das 8 ist. Schließlich wird die 3 bis die zwei in der Spalte

1000's wegen borgen und  $2-1 = 1$  geändert und läßt uns mit der abschließenden Antwort von 1822.

Als Beispiel des binären Abzugs, subtrahieren wir 0101 von 1011:

$$\begin{array}{r} 011 \\ 40\ 01 \\ -01\ 01 \\ \hline 01\ 00 \end{array}$$

Im  $2^0$  und  $2^1$  tritt 1 Spalte, Abzug die gleiche Weise auf, die er im dezimalen Abzug wurde. Wenn wir an die  $2^3$  Spalte gelangen, müssen wir borgen. Das 0 in der Spalte  $2^3$  wird bis zwei 1 geändert, weil Sie Spalte 4 borgen müssen <sup>des Formulars 2</sup>, die zweimal soviel wie die Spalte  $2^3$  wert IST. Weil Sie 1 von der Spalte  $2^4$  borgten, ändern Sie sie bis 0. Der Rest des Abzugs ist einfach, und getan der gleichen Weise, die er im normalen dezimalen Abzug erfolgt würde.

### Einleitung in eine Logik- Gatter

In der englischen Sprache " ist " die Bezeichnung " Gatter eine Struktur, die geschwungen werden, gezeichnet werden oder gesenkt werden kann, um einen Eingang oder einen Durchgang zu blocken ". Ein Beispiel ist eine Tür auf einem Zaun, der geöffnet werden kann ließ Leute innen, und geschlossen, um Leute heraus zu halten. In den Computern haben Logikgatter eine ähnliche Anwendung. Ein Logikgatter hat ein oder mehr Input (entweder 1 oder 0), und ausgegebenen einen. Logikgatter werden entworfen, um den Durchgang der Zweipunktsignale durch das Gatter für bestimmte Kombinationen von Input zu erlauben. Diese Gatter konnten einfach ursprünglich, und unbrauchbar scheinen. Obgleich ein Logikgatter alleine fast unbrauchbar ist, kann eine Kombination der Gatter, die zusammen arbeiten, viele Zwecke dienen. Heutige Computer, die sehr komplizierte Arbeit erledigen, bestehen Millionen dieser einfachen Logikgatter. Es gibt drei grundlegende Logikgatter. Sie sind die AND, OR und NOT Gatter. Diese Gatter werden in den unterschiedlichen Kombinationen benutzt, um kompliziertere Gatter wie das XOR zu bilden. Da Sie in einem Moment herausfinden, sind die drei grundlegenden Logik gatter sehr einfach zu verstehen. Aber, bevor wir über Logikgatter erlernen, gibt es einige Bezeichnungen, die wir ungefähr wissen sollten:

1. **Wahrheitstabelle:** Eine Wahrheits- Tabelle ist eine Tabelle von Kombinationen. Sie zeigt alle Inputmöglichkeiten und die Ausgaben, welches Resultat von diesen Möglichkeiten. Z.B. ist hier eine Wahrheits- Tabellenvertretung, was die Produkte ist, wenn die Nr. 2 mit den Zahlen von 1-5 multipliziert wird.

Zahlen	Produkt
2 * 1	2
2 * 2	4
2 * 3	6

2 * 4	8
2 * 5	10

Dieses wird eine Wahrheitstabelle genannt, weil es eine Tabelle ist, die Tatsachen über das Produkt von zwei hat und von 1-5 numeriert. Eine Wahrheitstabelle für Logikgatter wird benutzt um Ihnen alle möglichen Input für die Gatter zu zeigen und den resultierenden Ausgaben für jede Kombination von Input.

2. **Gattersymbole:** Gattersymbole werden in den Diagrammen für die Planung verwendet, bevor Computerschaltkreis aufgebaut wird. Die Gatter nicht wirklich sehen wie die Symbole aus, werden die Symbole verwendet, um die Diagramme einfacher und kompletter zu bilden.

3. **Mehr in binärem auszukennende Sachen:** Von den vorhergehenden Lektionen wissen wir, daß dieses 1 und 0 die zwei Möglichkeiten in binärem sind. In der Computerwelt hören Sie häufig " die HOHE ", die anstelle von 1 verwendet werden, und " das TIEF ", das anstelle von 1 verwendet wird.

4. **Input und Ausgaben:** Die Input und die Ausgaben für Logikgatter werden durch Zeichen dargestellt. Allgemein wird die Ausgabe durch das Zeichen Q dargestellt, oder das Zeichen Y. We wird das Zeichen Y für die Ausgabe benutzen. Die Input werden fast immer durch die ersten Zeichen des Alphabetes dargestellt. A und B sind verwendetes häufig, während C, D und andere Zeichen benutzt werden, wenn die Diagramme komplizierter sind.

### Grundlegende LogikGatter

Dieses hilft Ihnen, zu verstehen wie die drei grundlegende Gatterfunktion. Sie erfahren, daß ihre Namen der Art der Operation ähneln, die sie gebildet wurden, um zu tun. Wenn dieses gesagt ist, gibt es drei grundlegende Gatter. Sie sind das AND Gatter, Gatter OR und das Gatter NOT.

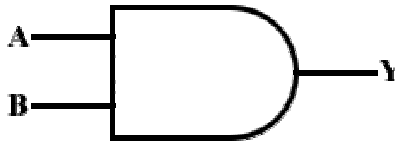
#### Das Gatter AND

Das Gatter AND gibt eine HOHE Ausgabe, nur wenn alle Input HOCH sind. Wenn irgendein der Input NIEDRIG sind, ist die Ausgabe auch NIEDRIG. Diese Informationen über das Gatter können in einer Wahrheitstabelle dargestellt werden. Ist hier die Wahrheitstabelle mit AND Gattern:

INPUT		OUTPUT
A	B	Y
0	0	0
0	1	0
1	0	0

1	1	1
---	---	---

Das Gatter AND kann mehr als zwei Input haben. Eine binäre HÖHE wird nur produziert, wenn alle Input HOCH sind. Sowie alle weiteren Gatter hat dieses Gatter ein Symbol, das verwendet wird, um es in einem Diagramm darzustellen:



**Das Gatter OR**

Das Gatter OR ist ein sehr einfaches Gatter. Es liefert eine HOHE Ausgabe, wenn irgendwelche der Input HOCH sind. Die einzige Weise, daß ein Gatter OR eine NIEDRIGE Ausgabe produzieren würde, ist, wenn alle Input NIEDRIG waren. Wenn dieses gesagt ist, ist hier die Wahrheitstabelle und das Symbol für das Gatter 2-input OR:

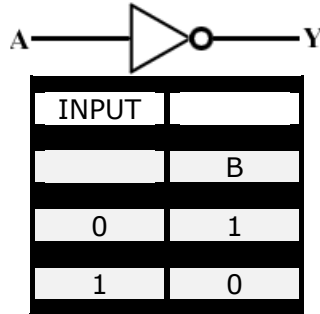


INPUT		OUTPUT
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Gerade wie das Gatter AND, kann das Gatter OR mehr als zwei Input haben. Das Gatter OR würde es ist Logik die gleiche Weise bearbeiten, nur Input einer HOCH würden sein müssen damit die Ausgabe geworden HOCH.

**Das Gatter NOT**

Das einfachste der drei grundlegenden Gatter ist das Gatter NOT. Es wird allgemein einen Inverter genannt. Der Zweck des Gatters NOT ist, den Input zu nehmen, ändert ihn zu seinem binären gegenüber von und gab den umgekehrten Input aus. Anders als die AND oder die OR Gatter kann das Gatter NOT eingegebenes ein nur haben. Ist hier die Wahrheitstabelle und -symbol für das Gatter NOT:



### Andere LogikGatter

Alle von den anderen Gattern werden mit einer Kombination von zwei oder mehr der AND, OR und/oder NOT Gatter erstellt.

### Das Gatter NAND

Das Gatter NAND ist wie das grundlegende Gatter AND gerade. Der einzige Unterschied ist, daß das Gatter NAND die abschließende Ausgabe umkehrt. Ist hier die Wahrheitstabelle für das GATTER NAND:

INPUT		OUTPUT
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

Obleich dieses Gatter ziemlich einfach scheint, ist es komplizierter als irgendwelche der grundlegenden Gatter. Wie Sie durch die Wahrheitß tabelle sehen können, sind die Ausgaben das genaue Entgegengesetzte des Gatters AND. Dieses wird erzielt, indem man eine Kombination von zwei grundlegenden Gattern, von AND und von NOT verwendet. Da es eine Kombination von zwei grundlegenden Gattern verwendet, gilt es nicht als ein grundlegendes Gatter. Ist hier, wie die Bestandteile des Gatters NAND geordnet werden:



Wie Sie sehen können, steigen die Input in ein Gatter AND ein. Die Ausgabe des Gatters AND wird dann durch das Gatter OR umgekehrt um das Resultat des vollständigen Gatters NAND zu erzielen. Das Gatter NAND ist wirklich ein Gatter AND mit einer umgekehrten Ausgabe.

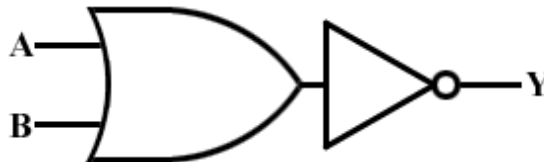
Anstelle von die Kombination der zwei Gatter im oben genannten Diagramm als Symbol für das Gatter NAND verwenden zu müssen, ein vereinfachtes Symbol wird verwendet:



Beachten Sie, daß das Gatter NOT durch einen kleinen Kreis geersetzt worden ist. Dieser kleine Kreis wird allgemein benutzt, um ein vollständiges Gatter NAND zu ersetzen, wenn man Koinzidenzschaltungen entwirft.

### Das Gatter NOR

Das Gatter NOR ist das genaue Entgegengesetzte des Gatters OR. Die Ausgabe ist HOCH, nur wenn alle Input NIEDRIG sind. Wenn irgendwelche der Input HOCH sind, ist die Ausgabe immer NIEDRIG. Das Gatter NOR wird auch mit zwei grundlegenden Gattern konstruiert. Dieses Mal werden das Gatter OR und NOT eingesetzt, um das Gatter NOR zu erstellen. Ist hier ein Diagramm des Gatters NOR zusammen mit der Wahrheitstabelle:



INPUT		OUTPUT
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

Wie Sie sehen können, ist das Gatter NOR dasselbe, das das Gatter NAND, ausgenommen es ein Gatter OR anstatt des Gatters AND benutzt. um Diagramme einfacher zu bilden, ist die gleiche Sache mit dem Gatter NOR erfolgt, wie mit dem Gatter NAND getan wurde:

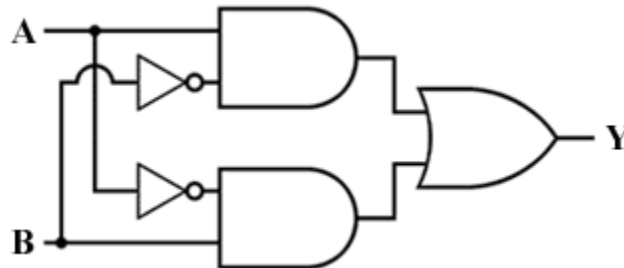


### Das Gatter XOR

Das EXKLUSIVE OR Gatter (XOR wird sehr allgemein in den Rechenschaltungen benutzt. Das Gatter XOR vergleicht die Input. Wenn beide Input (es kann zwei Input für dieses Gatter nur geben), dieselben sind, ist die Ausgabe NIEDRIG. Demgegenüber wenn die Input unterscheiden, dann die Ausgabe ist HOCH. Ist hier die Wahrheitstabelle für das Gatter XOR:

INPUT		OUTPUT
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Das Gatter XOR ist von allen Gattern am kompliziertsten; es benötigt zwei AND Gatter, zwei NOT Gatter, sowie ein Gatter OR. Ist hier ein Diagramm, das zeigt, wie die fünf Gatter geordnet werden, um das Gatter XOR zu erstellen:



Die a- und b-Input sind beide Spalte in zwei zum gleichzeitig Lassen sie die Input für zwei Gatter sein. Die zwei Input, die gerade von der Spalte erstellt wurden, werden umgekehrt. Es gibt zwei AND Gatter. Die zwei ursprünglichen Input gehen zu den unterschiedlichen AND Gattern. Der umgekehrte Input von A wird der zweite Input des Gatters AND, in dem B der andere Input ist. Ebenso wird der umgekehrte Input von B der zweite Input des Gatters AND, in dem A der andere Input ist. Die Ausgaben der AND Gatter werden die Input des Gatters OR. Wenn irgendein der Input HOCH ist, dann ist die Ausgabe des vollständigen Gatters XOR HOCH. Andernfalls ist die Ausgabe des Gatters XOR NIEDRIG.

Gerade wie die zwei anderen Gatter in diesem Kapitel, wird das Gatter XOR in den Diagrammen vereinfacht. Ist hier das Symbol, das verwendet wird, um das Gatter XOR darzustellen:

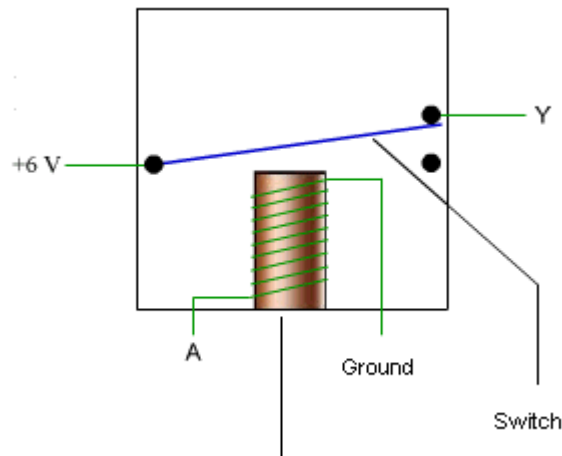


### Innere Gatter

Dieses Kapitel nimmt uns Innere grundlegende Gatter wie die AND, die OR und NOT, um zu sehen, wie sie mechanisch arbeiten.

### Das Gatter NOT

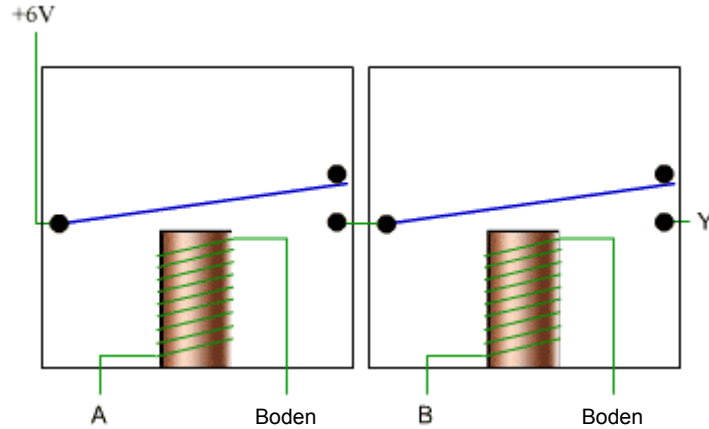
Bevor wir, wie die AND und OR Gatter arbeiten, uns müssen zuerst verstehen wie die grundlegendsten NOT Gatter- Arbeiten verstehen.



Elektromagnet - Ein leitendes Metall mit einer Spule aufgewickelt um sie. Wenn Energie durch die Spule überschreitet, magnetisiert das Metall und zieht den Schalter an.

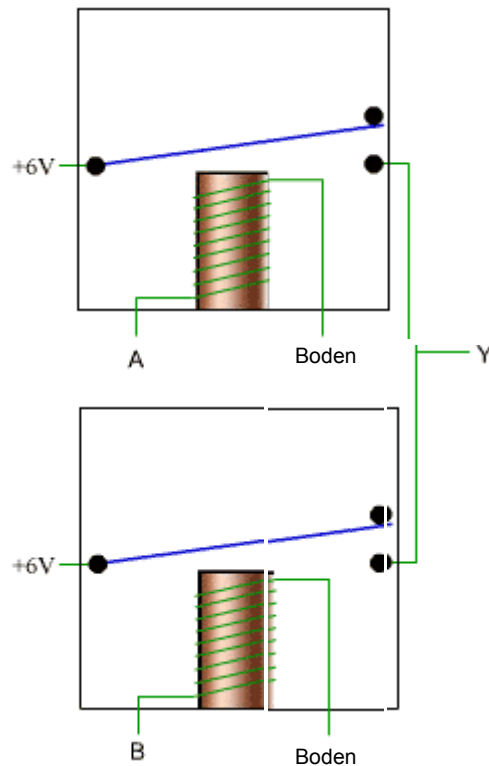
Wenn A bei 0 Volt ist, läßt der Schalter den Strom durch, und die Ausgabe an Y ist 1. Wenn A an + 6 Volt ist, das einen Input von 1 gibt, die Elektromagnetanfänge, den Schalter ziehend und brechen den Fluß vom aktuellen und geben eine Ausgabe von 0.

### Das Gatter AND



Wenn a- und b-Input NIEDRIG sind, werden keine der Elektromagneten angeschaltet, und die Ausgabe (Y) ist NIEDRIG. Wenn B HOCH ist, wird sein Schalter eingestellt, um den Strom durch zu erlauben. Aber der Strom muß durch den ersten Schalter zuerst überschreiten, bevor er die Sekunde durchläuft. Da er nicht kann, bleibt Y NIEDRIG. Wenn A HOCH ist, wird sein Schalter eingestellt, um den Strom durch zu erlauben. Aber der Strom muß durch den zweiten Schalter überschreiten, nachdem er der erste durchläuft. Da er nicht kann, bleibt Y NIEDRIG. Wenn A und B HOCH sind, lassen beide Schalter Strom durch und lassen die Ausgabe an Y, wie HOCH.

### Das Gatter OR



Wenn beide Input (A und B) sind, die Elektromagneten bleiben weg NIEDRIG und der Schalter läßt den Strom nicht zur Ausgabe fließen und gibt Y binäres 0 ausgegeben. Wenn einer der Input NIEDRIG ist und man hoch ist, geht ein Elektromagnet weiter

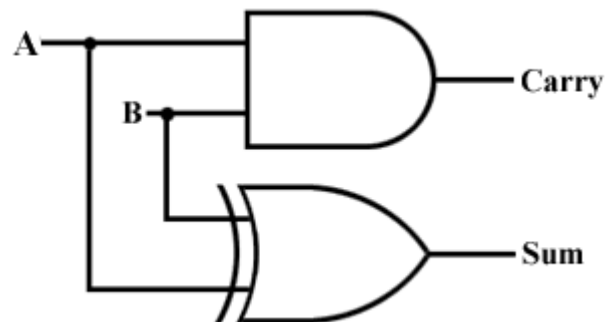
und veranläßt einen Schalter sich zu öffnen. Eins ist alles, das erforderlich ist, damit Y binäres 1 ausgegeben hat. Schließlich wenn beide Input HOCH sind, ist Y noch HOCH, weil es zwei Quellen der Energie gibt und man mindestens für eine HOHE Ausgabe angefordert wird.

### Binäre Additionsmaschinen

Jetzt haben Sie erlernt, wie man binäre Zahlen hinzufügt. Sie haben auch erlernt, wie die 6 allgemeinen Logikgatter funktionieren. Jetzt ist es Zeit, zu erlernen, wie die Logikgatter benutzt werden, um binäre Zahlen hinzuzufügen.

#### Der Binäre Halbaddierer

Sie können überrascht werden, zu sehen, wie einfach er eine Koinzidenzschaltung erstellen soll, die binäre Zahlen hinzufügt. Der binäre Halbaddierer ist für das Hinzufügen der binären Zahlen 1-bit vollkommen. Er wird aus einem AND Gatter und einem Gatter XOR heraus gebildet. Ist hier, wie es in einem Diagramm geordnet wird:



A und B sind die zwei Input. In diesem Fall sind sie die zwei Bits, die hinzugefügt werden sollen. Das Gatter XOR wird benutzt, um aus der Summe darzustellen, während das Gatter AND benutzt wird, um aus tragen darzustellen. Lassen Sie uns mit dem Gatter XOR beginnen. A und B werden eingegeben, und wenn die zwei Bits unterscheiden, ist die Ausgabe des Gatters XOR 1, das bedeutet, daß die Summe 1 ist. Andernfalls ist die Ausgabe des Gatters XOR 0., wenn Sie Hinzufügung der Bits erinnern an,  $0 + 0 = 0$ ,  $0 + 1 = 1$ , während  $1 + 1 = 0$  mit tragen von 1. Dieses ist, wohin das Gatter AND hereinkommt. Wenn A und B 1 sind, dann ist die Ausgabe für das Gatter AND, das tragen darstellt, 1. Andernfalls ist tragen 0. Dieses kann in einer Wahrheitstabelle für den Halbaddierer zusammengefaßt werden:

Input		Ausgabe	
A	B	Summe	Tragen Sie
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

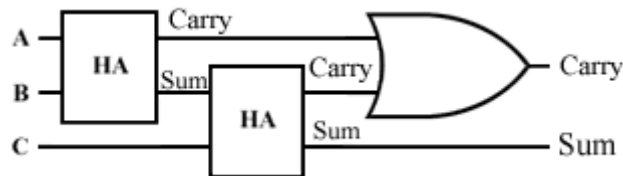
Der Halbaddierer hat sein eigenes Sonderzeichen in den Diagrammen, zwecks sie einfacher zu bilden. Wie dieses ist, was es aussieht:



Nun da wir können, der Computer zwei Digits hintereinander hinzufügt, ist es Zeit, zu erlernen, wie er zwei Bits und tragen gleichzeitig hinzufügt.

### Das Binäre Full-Adder

Die binäre Volladditionsmaschine ist sehr nützlich, weil sie drei Digits hintereinander hinzufügen kann. Wenn Sie von unserem binären Hinzufügungskapitel wiederaufrufen, gibt es trägt manchmal diese Notwendigkeit, zusammen mit den zwei anderen Digits hinzugefügt zu werden. Für dieses benötigt die Hinzufügung von drei Bits hintereinander das ist genau, was die Volladditionsmaschine benutzt wird. Ist hier das Logikdiagramm für die kompliziertere Volladditionsmaschine, die zwei HA's (Halbaddierer) sowie ein Gatter OR einsetzt:



Wie Sie sehen können, gibt es jetzt drei Input: A, B und C. A und B werden normalerweise für die zwei Bits verwendet, die normalerweise hinzugefügt würden, während C normalerweise als tragen von einem niedrigeren Stellenwert verwendet wird. Wie Sie sehen können, gibt A ein und B steigen in einen Halbaddierer ein. Die Summe des Halbaddierers wird als einer der Input gesendet (zusammen mit Input C) zum zweiten Halbaddierer. Die Summe des zweiten Halbaddierers ist die Summe aller drei Input. Wenn es tragen von irgendeinem oder von beiden der Halbaddierer gab, ist tragen, das für die volle Additionsmaschine ausgegeben wird, HOCH. Es gibt 8 mögliche Kombinationen der Input für die Volladditionsmaschine, ganz gespeichert auf der Volladditionsmaschinenwahrheitstabelle:

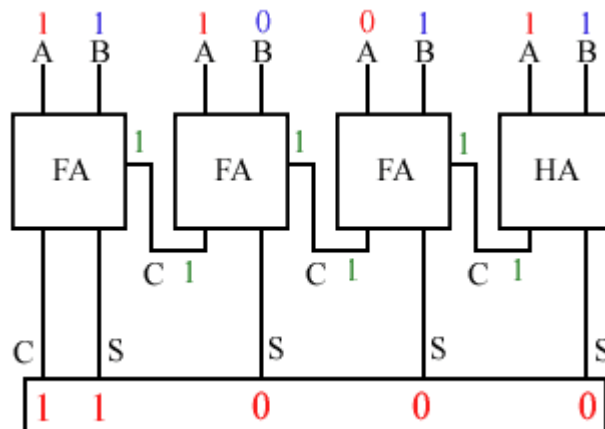
Input			Ausgabe	
A	B	C	Summe	Tragen Sie
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1

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

um dieses Diagramm zu vereinfachen, wird ein Symbol für die Volladditionsmaschine verwendet. Das Symbol ist zum Halbaddierer-Symbol identisch, ausgenommen es das Zeichenfa anstelle vom ha auf ihm trägt.

### Parallele Binäre Additionsmaschine

Der Gebrauch eines Halbaddierers oder einer Volladditionsmaschine alleine ist für zwei binäre Zahlen mit einer Länge von einem Bit pro Stück oben hinzufügen groß, aber was geschieht, wenn der Computer zwei binäre Zahlen mit einer längeren Länge oben hinzufügen muß? Gut gibt es einige Weisen des Tuns dies. Die schnellste Weise soll bei weitem die parallele binäre Additionsmaschine benutzen. Die parallele binäre Additionsmaschine benutzt einen Halbaddierer, zusammen mit einer oder mehr vollen Additionsmaschinen. Die Zahl benötigten den Gesamtadditionsmaschinen hängt von der Länge vom größten der zwei binären Zahlen ab, die hinzugefügt werden sollen. Z.B. wenn wir die binären Nr. 1011 und 1 oben hinzufügen sollten, würden wir vier Additionsmaschinen in der Gesamtmenge benötigen, weil die Länge der größeren Zahl vier ist. Dieses im Verstand halten, ist hier eine Demonstration von, wie eine four-bit parallele binäre Additionsmaschine arbeitet mit 1101 und 1011 als die zwei Zahlen hinzuzufügen:



Gerade wie, wenn wir ohne den Computer, in der parallelen binären Additionsmaschine, den Computer hinzufügen, fügt von rechts nach links hinzu. Ist hier eine schrittweise Liste und zeigt Ihnen, was in der parallelen binären Additionsmaschine geschieht:

1. Im einzigen Halbaddierer geben Input von 1 und 1 uns 0 mit tragen von 1.

2. In der ersten Volladditionsmaschine (von rechts nach links gehend), geben die Input von 1 und von 0 plus tragen von 1 vom Halbaddierer uns 0 mit tragen von 1.

3. In der zweiten vollen Additionsmaschine geben die Input von 0 und 1 plus tragen von 1 von der vorhergehenden Volladditionsmaschine und 0 mit tragen von 1.

4. In der dritten und abschließenden vollen Additionsmaschine geben die Input von 1 und 1 plus tragen von 1 von der vorhergehenden Volladditionsmaschine und 1 mit tragen von 1.

5. Da es keine mehr Zahlen zum oben Hinzufügen gibt und es noch tragen von 1 gibt, wird tragen das bedeutendste Bit

6. Die Summe von 1101 und von 1011 ist 11000.

### Boolesche Algebra

Da die meisten Leute sich gut bewußt fühlen, werden mathematische Ausdrücke verwendet, um andere mathematische Probleme die einfacheren und kleineren zu bilden Übertragen auf Lochkarten und. Boolesche Algebra wird verwendet, um die gleiche Sache zu tun, ausgenommen sie sie mit Koinzidenzschaltungen anstelle von anderen mathematischen Problemen tut.

Die erste Sache, die für Boolesche Algebra bekannt muß, ist die Bedeutung der verschiedenen Zeichen. Es gibt drei von diesen und ganz berechnet von den drei grundlegenden Gattern. Sie sind AND, OR und NOT.

Die Operation AND bekannt auch als Zusammenhang. Sie gibt das Produkt von zwei binären Bits. Mit A und B als Input, würde es als AB oder manchmal  $A \cdot B$  geschrieben. Dieses bedeutet selbstverständlich, daß A von B multipliziert wird, das genau wie AND Gatterfunktionen ist.



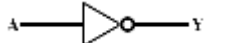



Die Operation OR bekannt auch als Trennung. Diese Operation gibt die Summe von zwei binären Bits. Wieder mit A und B als Input, würde es als  $A + B$  geschrieben, das wie OR Gatterfunktionen ist. 55€, " + " bedeutet notwendigerweise die gleiche Sache, daß es in der normalen Mathematik tut alle, die wir an gewöhntSIND. In der Booleschen Algebra steht es für OR. Z.B. würden  $1 + 1$  bis 1, nicht 10, das binäre Äquivalent von zwei entsprechen. Wenn irgendein der Input 1 ist, ist die Ausgabe 1. Die Operation NOT bekannt auch als Verneinung. Diese Operation gibt das Entgegengesetzte einer einzelnen Bezeichnung. Z.B. wird die Verneinung von A wie geschrieben  $\bar{A}$  Während die Verneinung von AB sein würde  $\overline{AB}$  Wir können die AND, OR und NOT Operationen auf den entsprechenden Gattern in Verbindung stehen:

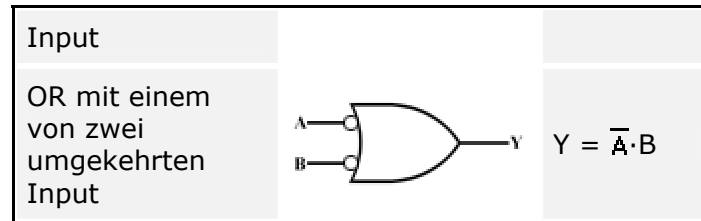
Tabelle Der Wahrheit AND	Tabelle Der Wahrheit OR	Tabelle Der Wahrheit NOT
$A \cdot B = Y$	$A + B = Y$	$\bar{A} = Y$
$0 \cdot 0 = 0$	$0 + 0 = 0$	0 verneint = 1

$0 \cdot 1 = 0$	$0 + 1 = 1$	1 verneint = 0
$1 \cdot 0 = 0$	$1 + 0 = 1$	
$1 \cdot 1 = 1$	$1 + 1 = 1$	

### Logikgatter und ihre Booleschen Ausdrücke

Alle Logikgatter haben einen Booleschen Ausdruck. um Sie zu zeigen, wie Boolesche Algebra weiter arbeitet, seien Sie hier einige Beispiele der Gatter und ihrer Booleschen Ausdrücke:

LogikGatterName	Symbol	Boolescher Ausdruck
AND		$Y = A \cdot B$
OR		$Y = A + B$
NOT		$Y = \bar{A}$
NAND		$Y = \overline{A \cdot B}$
NOR		$Y = \overline{A + B}$
AND mit einem von zwei umgekehrten		$Y = \bar{A} + B$



### Bilden einer Wahrheitstabelle mit einer Gleichung

Wenn es eine Gleichung gegeben wird, ist es einfach, eine Wahrheitstabelle für die Gleichung herzustellen. Ist hier ein Beispiel von, wie es erfolgt ist, mit dem Ausdruck  $Y = \bar{A} \cdot B$ . Wir schließen alle möglichen Werte für A und B an und speichern die Werte von Y:

1. <b>A = 0, B = 0</b> Y = $\bar{A} \cdot B$ Y = 1 · 0 <b>Y = 0</b>	2. <b>A = 0, B = 1</b> Y = $\bar{A} \cdot B$ Y = 1 · 1 <b>Y = 1</b>
3. <b>A = 1, B = 0</b> Y = $\bar{A} \cdot B$ Y = 0 · 0 <b>Y = 0</b>	4. <b>A = 1, B = 1</b> Y = $\bar{A} \cdot B$ Y = 0 · 1 <b>Y = 0</b>

Sobald dieses getan wird, organisieren wir es in einer Wahrheitstabelle:

Input		Ausgabe
A	B	Y
0	0	0
0	1	1

1	0	0
1	1	0

Wir hoffen, daß Sie jetzt ein grösseres Verständnis des Binärsystems und der Booleschen Logik- Gatter haben.