Rechnerarchitektur - 2022

• Wie viele **Zustandskombinationen** gibt es?
- abhängig von der Grundmenge n
- bei n gegebenen Zuständen: 2^n
- n = 3 dann 2^3 = 8 Bitmuster

• **Hexadezimale** Schreibweise:

0000 = 0	0101 = 5	1010 = A
0001 = 1	0110 = 6	1011 = B
0010 = 2	0111 = 7	1100 = C
0011 = 3	1000 = 8	1101 = D
0100 = 4	1001 = 9	1110 = E
		1111 = F

zB. : 0000 0011 = 03
0010 1011 = 2B

• **Umrechen**:
• von Binärsystem ins Dezimalsystem: 101110 = 46

Stellenwert	2^5	2^4	2^3	2^2	2^1	2^0
Ziffer	1	0	1	1	1	0

• von Dezimalsystem ins Binärsystem: 59 = 111011

1. Teile die Zahl mit Rest durch 2 und notiere den Rest.

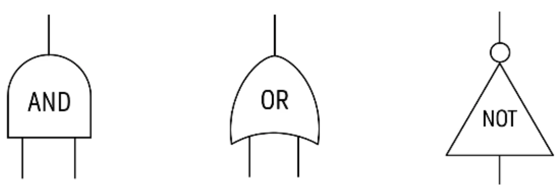
- Teile das Ergebnis wieder durch 2 und notiere den Rest.
- Fahre fort bis dein Ergebnis 0 ist.
- Die gesuchte Binärzahl sind die Ziffern der Reste, wobei man mit dem letzten Rest beginnt.

• **Speichergrößen**:

	:8	:1024	:1024	:1024	:1024	:1024
Bit	Byte	Kilobyte	Megabyte	Gigabyte	Terabyte	Petabyte
*8	*1024	*1024	*1024	*1024	*1024	

• **Gatter**:

AND OR NOT



AND: $A * B$, $A \wedge B$, *A AND B*

OR: $A + B$, $A \vee B$, *A OR B*

NOT: $-A / \bar{A}$, $\neg A$, *NOT A*

AND

A	B	$A * B$
0	0	0
0	1	0
1	0	0
1	1	1

$A * B$
$A \wedge B$
A AND B

OR

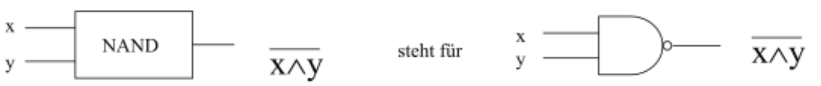
A	B	$A + B$
0	0	0
0	1	1
1	0	1
1	1	1

$A + B$
$A \vee B$
A OR B

NOT

A	-A
0	1
1	0

$-A / \bar{A}$
$\neg A$
NOT A



Bezeichnung: ↑ oder f_{14} oder Shefferscher Strich

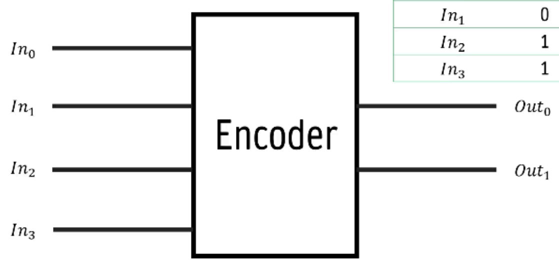
Bezeichnung: ↓ oder f_8 oder Peircescher Pfeil

Logik / Boolsche Algebra

Kommutativgesetze	(1) $a \wedge b = b \wedge a$	(1') $a \vee b = b \vee a$
Assoziativgesetze	(2) $(a \wedge b) \wedge c = a \wedge (b \wedge c)$	(2') $(a \vee b) \vee c = a \vee (b \vee c)$
Idempotenzgesetze	(3) $a \wedge a = a$	(3') $a \vee a = a$
Distributivgesetze	(4) $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$	(4') $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$
Neutralitätsgesetze	(5) $a \wedge 1 = a$	(5') $a \vee 0 = a$
Extremalgesetze	(6) $a \wedge 0 = 0$	(6') $a \vee 1 = 1$
Doppelnegationsgesetz (Involution)	(7) $\neg(\neg a) = a$	
De Morgansche Gesetze	(8) $\neg(a \wedge b) = \neg a \vee \neg b$	(8') $\neg(a \vee b) = \neg a \wedge \neg b$
Komplementärgesetze	(9) $a \wedge \neg a = 0$	(9') $a \vee \neg a = 1$
Dualitätsgesetze	(10) $\neg 0 = 1$	(10') $\neg 1 = 0$
Absorptionsgesetze	(11) $a \vee (a \wedge b) = a$	(11') $a \wedge (a \vee b) = a$

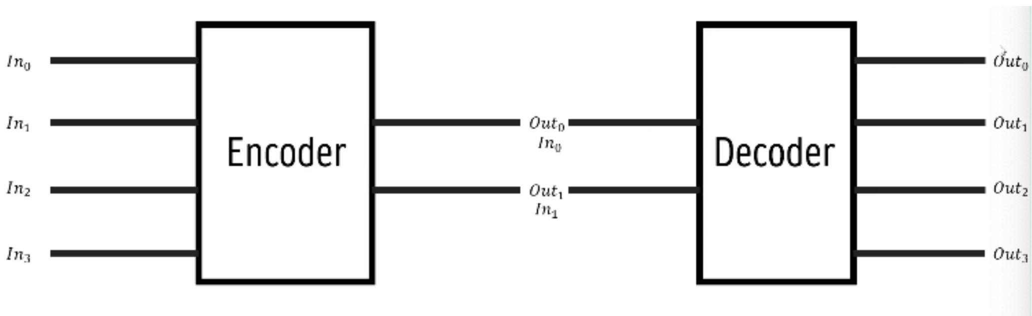
Encoder

Eingangssignal	Out_0	Out_1
In_0	0	0
In_1	0	1
In_2	1	0
In_3	1	1



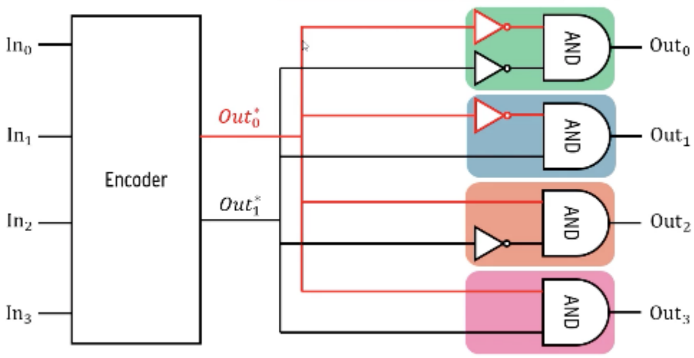
Immer nur eine Eingangsleitung darf an sein!
Hat 2^n Eingänge und produziert einen n-Bit Output.

Encoder & Decoder



Decoder hat n Eingänge und 2^n Ausgänge

In_0/Out_0^*	In_1/Out_1^*	Out_0	Out_1	Out_2	Out_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



• Minterme

$$m_i(x_1, \dots, x_n) = x_1^{i_1} * x_2^{i_2} * \dots * x_n^{i_n}$$

i-ter Minterm (Zeile)

$$x_j^{i_j} =: \begin{cases} x_j \; falls \; i_j = 1 \\ \overline{x_j} \; falls \; i_j = 0 \end{cases}$$

j-te Stelle der booleschen Funktion (Spalte)

i	x_1	x_2	x_3
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

$$m_5(x_1, x_2, x_3) = x_1 * \overline{x_2} * x_3$$

• Maxterme

$$M_i(x_1, \dots, x_n) = x_1^{i_1} + x_2^{i_2} + \dots + x_n^{i_n}$$

i-ter Minterm (Zeile)

$$x_j^{i_j} =: \begin{cases} x_j \; falls \; i_j = 0 \\ \overline{x_j} \; falls \; i_j = 1 \end{cases}$$

j-te Stelle der booleschen Funktion (Spalte)

i	x_1	x_2	x_3
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

$$M_5(x_1, x_2, x_3) = \overline{x_1} + x_2 + \overline{x_3}$$

Der i-te Maxterm ergibt sich aus der Negation des i-ten Minterms

Minterm:
$m_5(x_1, x_2, x_3) = x_1 * \overline{x_2} * x_3$

Maxterm:
$M_5 = \overline{m_5} = \overline{x_1 * \overline{x_2} * x_3} = \overline{x_1} + x_2 + \overline{x_3}$

• Einschlägiger Index

i ist einschlägiger Index, wenn $f(x_1, \dots, x_n) = 1$ gilt

Jede boolesche Funktion $f: B^n \to B$ ist eindeutig darstellbar als

- Summe der Minterme ihrer einschlägigen Indizes oder
- Produkt der Maxterme ihrer nicht einschlägigen Indizes

i	x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

• DNF & KNF

Die Disjunktive Normalform (DNF) erhält man durch die ODER-Verknüpfung aller Minterme der einschlägigen Indizes

Konjunktive Normalform (KNF) erhält man durch die UND-Verknüpfung aller Maxterme der nicht einschlägigen Indizes

Wenn in der Funktionstabelle:

Anzahl einschlägige Indizes < Anzahl nicht einschlägige Indizes → DNF

Anzahl einschlägige Indizes > Anzahl nicht einschlägige Indizes → KNF