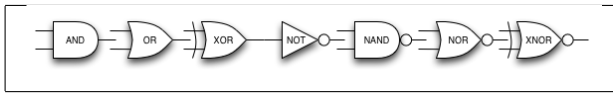


# Logische Verknüpfungen

Für N eingänge hat man  $2^N$  Eingangskombinationen.

Elementare Logische Funktionen: NOT, AND, OR, NAND, NOR, XOR, XNOR



## Zahlensysteme

Dezimal	Binary	Hex
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

LSB: Least Significant Bit - z.B.:  $2^0$   
 Nibble: Gruppe von 4 Bit  
 Byte: Gruppe von 8 Bit (2Nibble)  
 Word: Gruppe von mehr als 8 Bit (Meisstens 16Bit)  
 DWord: Double Word: oft eine Gruppe von 32 Bit

Bsp: 1011 1100 0010 (BIN) = BC2 (HEX)  
 101 111 000 010 (BIN) = 5702 (OCTAL)

Divisionsmethode:  
 $47 : 2 = 23 \text{ r}1$   
 $23 : 2 = 11 \text{ r}1$   
 $11 : 2 = 5 \text{ r}1$   
 $5 : 2 = 2 \text{ r}1$   
 $2 : 2 = 1 \text{ r}0$   
 $1 : 2 = 0 \text{ r}1$

## Schaltalgebra

Funktion	NOR	NAND
NOT	$x \nabla x$	$x \text{ NAND } x$
OR	$(x \nabla y) \nabla (x \nabla y)$	$(x \text{ NAND } x) \text{ NAND } (y \text{ NAND } y)$
AND	$(x \nabla x) \nabla (y \nabla y)$	$(x \text{ NAND } y) \text{ NAND } (x \text{ NAND } y)$

## Vereinfachungen

- Kommutativgesetze

- $X1 \& X2 = X2 \& X1$
- $X1 \# X2 = X2 \# X1$

- Assoziativgesetze

- $(X1 \& X2) \& X3 = X1 \& (X2 \& X3)$
- $(X1 \# X2) \# X3 = X1 \# (X2 \# X3)$

- Distributivgesetze

- $(X1 \# X2) \& X3 = (X1 \& X3) \# (X2 \& X3)$
- $(X1 \& X2) \# X3 = (X1 \# X3) \& (X2 \# X3)$

## Disjunktive Normalform

- OR Verknüpfung von AND Blöcken für  $K=1$
- Jeder AND-Block ist ein MINTERM
- Die DNF  $K$  ist eine OR-Verknüpfung aller guten MINTERME (gut = Wahrheitstabelle 1)

Für die Darstellung mit NAND anstelle von OR:

Das DeMorgan Theorem anwenden:  $K = \neg(\neg K)$  und dann weiter vereinfachen.

## Konjunktive Normalform

- AND Verknüpfung von OR Blöcken
- Herstellen durch DNF von  $K=0$ , dann DeMorgan Theorem anwenden
- Jeder OR-Block ist ein MAXTERM, der einer Zeile in der Wahrheitstabelle entspricht, negiert, wenn in der Wahrheitstabelle =1, direkt falls WT=0.

Multiplexer:

Art von Drehschalter, umschalten zwischen verschiedenen Eingängen

## Vorzeichenlose und Vorzeichenbehaftete Zahlen

Typ	min	-2	-1	0	1	2	max
Unsigned	-	-	-	0000	0001	0010	1111 (15)
One's Complement	1000 (-7)	1101	1110	0000, 1111	0001	0010	0111 (7)
Two's Complement	1000 (-8)	1110	1111	0000	0001	0010	0111 (7)
Sign Magnitude	1111 (-7)	1010	1001	0000, 1000	0001	0010	0111 (7)

CF: Carry Flag: Übertrag beim Addieren

OF: Overflow Flag: Über oder Unterlaufen

## Kombinatorische Schaltungen vs Sequenzielle Schaltungen

- Komb: Reine Schaltnetze, werden nur durch Input beeinflusst, Zeiteinfluss nur von int. Delays
- Seq: Haben "Speicher", Zustände werden auch von den vorhergehenden Inputs beeinflusst (Zustand des Speichers), Zeitlicher Ablauf spielt eine grosse Rolle

### D-Latch (transp. FF)

- $D = \text{Input}$
- $C = \text{Control}$  (solange  $C=1$ , wird der Input durchgeschaltet)
- $Q, \neg Q = \text{Outputs}$
- Rückkopplung ist gefährlich, gespeicherter Zustand zufällig

### D-Flip-Flop (nichttransp. FF)

- Pos/Neg Flanken gesteuert
- gleiche Funktionsweise wie D-Latch
- Rückkopplung halbiert den Takt
- Set 1 und Reset 0 Funktion -> Synchron mit Takt oder Asynchron sofort

Bausteine

- Register: aus FF mit write-enable: zur Datenspeicherung
- asynchrone Schaltungen: D-FF ohne Clock, Staffelung, arbeiten so schnell wie möglich

Finite State Machine

