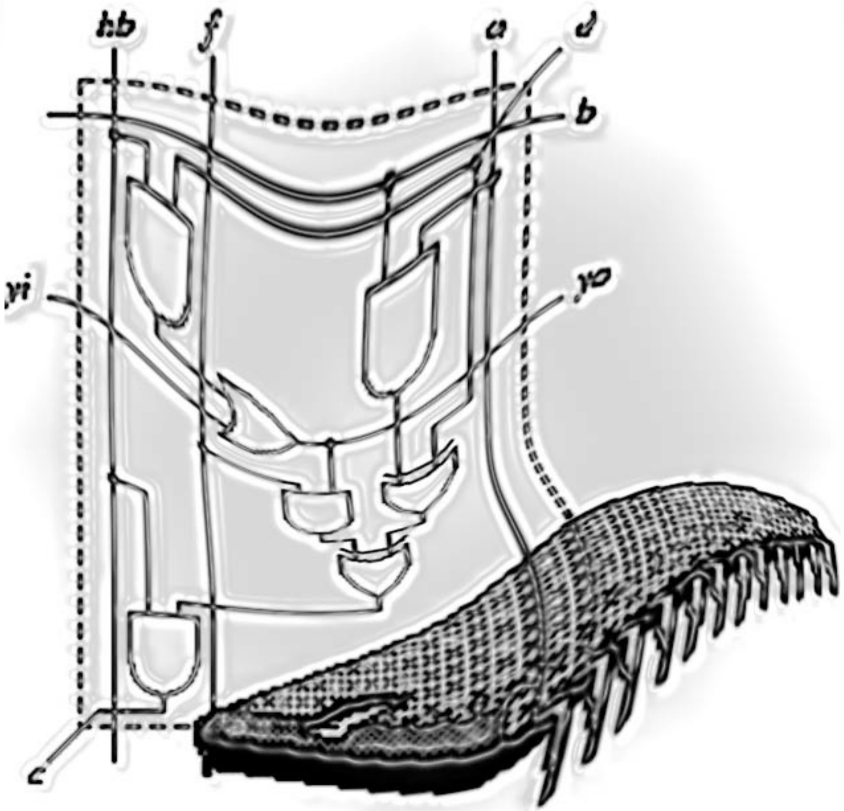
Technische Informatik I



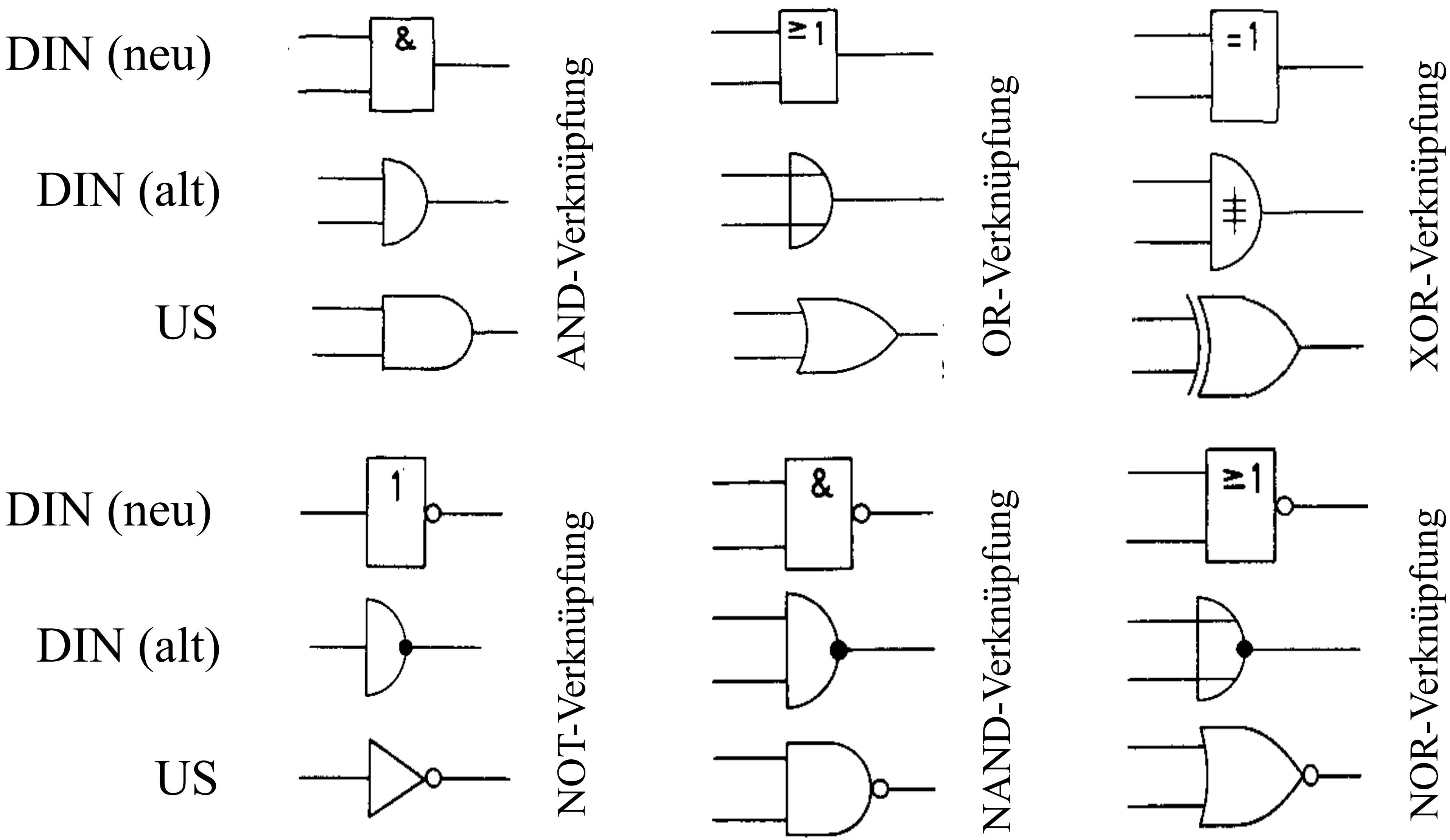
Kapitel 3

Schaltnetze

Prof. Dr. Dirk W. Hoffmann

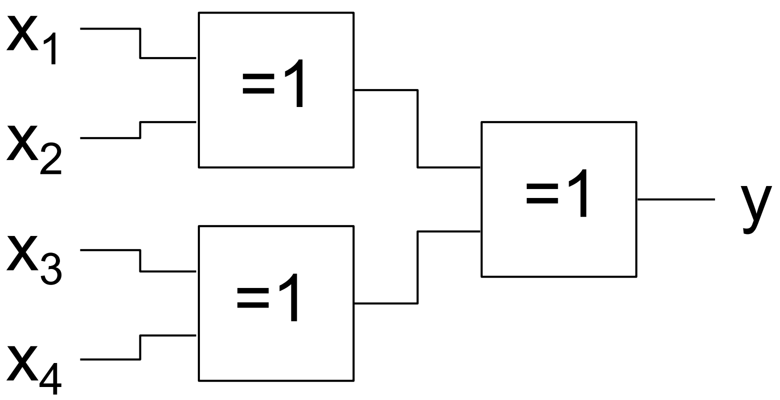
Hochschule Karlsruhe ◆ University of Applied Sciences ◆ Fakultät für Informatik

Logikgatter



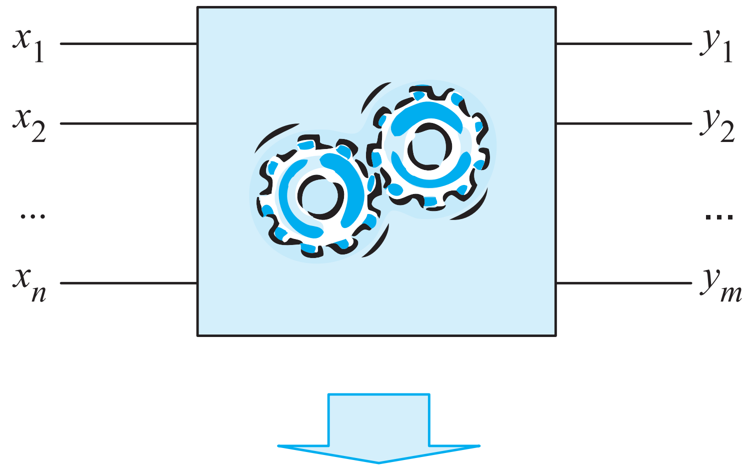
Beispiel

- Beispiel: Paritätsfunktion



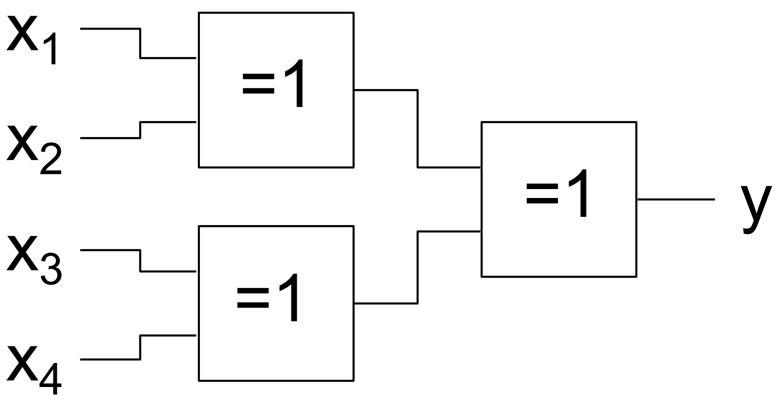
- Eigenschaften
  - Eingänge: x_1, x_2, x_3, x_4
  - Ausgänge: y
  - Gatter: 3 × XOR
  - Stufen: 2

- Allgemeines Schema



$$
\begin{aligned}
y_1 &= f_1(x_1, x_2, \ldots, x_n) \\
y_2 &= f_2(x_1, x_2, \ldots, x_n) \\
&\ldots \\
y_m &= f_m(x_1, x_2, \ldots, x_n)
\end{aligned}
$$

- Beispiel: Paritätsfunktion



- Eigenschaften
  - Eingänge: x_1, x_2, x_3, x_4
  - Ausgänge: y
  - Gatter: $3 \times$ XOR
  - Stufen: 2

- Wahrheitstabelle

	x	x	x	x	y
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	0

Darstellung in VHDL

- Beispiel: Paritätsfunktion

```
with bit_vector'(x4,x3,x2,x1)
   select
      y <=    '0' when "0000",
              '1' when "0001",
              '1' when "0010",
              '0' when "0011",
              '1' when "0100",
              '0' when "0101",
              '0' when "0110",
              '1' when "0111",
              '1' when "1000",
              '0' when "1001",
              '0' when "1010",
              '1' when "1011",
              '0' when "1100",
              '1' when "1101",
              '1' when "1110",
              '0' when "1111";
```

- Wahrheitstabelle

	x	x	x	x	y
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	0

Optimierte Darstellung in VHDL

- Beispiel: Paritätsfunktion

```
entity fktab is
port(x4,x3,x2,x1:in std_logic;
               y:out std_logic);
end;

architecture demo of fktab is
begin

with bit_vector'(x4,x3,x2,x1)
   select
      y <=    '1' when "0001",
              '1' when "0010",
              '1' when "0100",
              '1' when "0111",
              '1' when "1000",
              '1' when "1011",
              '1' when "1101",
              '1' when "1110",
              '0' when others;
end demo;
```

- Wahrheitstabelle

	x	x	x	x	y
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	0

Ausnützen von Redundanz

- Aufgabe
  - Gesucht ist eine Schaltung mit 4 Eingangsleitungen x_1, x_2, x_3, x_4 und einer Ausgangsleitung y. Die Schaltung soll für eine BCD-Ziffer bestimmen, ob diese ungerade (y=1) oder gerade ist (y=0).

```
with bit_vector'(x4,x3,x2,x1)
select
        y <=    '0' when "0000",
                '1' when "0001",
                '0' when "0010",
                '1' when "0011",
                '0' when "0100",
                '1' when "0101",
                '0' when "0110",
                '1' when "0111",
                '0' when "1000",
                '1' when "1001",
                '-' when others;
```

Normalformen (Beispiel)

	x	x	x		y
0	0	0	0		0
1	0	0	1		1
2	0	1	0		1
3	0	1	1		0
4	1	0	0		1
5	1	0	1		0
6	1	1	0		0
7	1	1	1		1

Normalformen (Allgemeine Form)

Gegeben: Boolesche Funktion $f(x_n, ..., x_3, x_2, x_1)$

Kanonische disjunktive Normalform

- Allgemeine Form

$$\bigvee_{e \in E} \text{Minterm}_e$$

  - E = *Einsmenge* von f
  - Jeder *Minterm* hat die Form

    $(L_n \wedge \ldots \wedge L_1) \quad L_i \in \{ x_i, \neg x_i \}$
  - Jedes L_i heißt ein *Literal* von f
- Abkürzungen
  - DNF (Disjunktive Normalform)
  - SOP (Sum of products)

Kanonische konjunktive Normalform

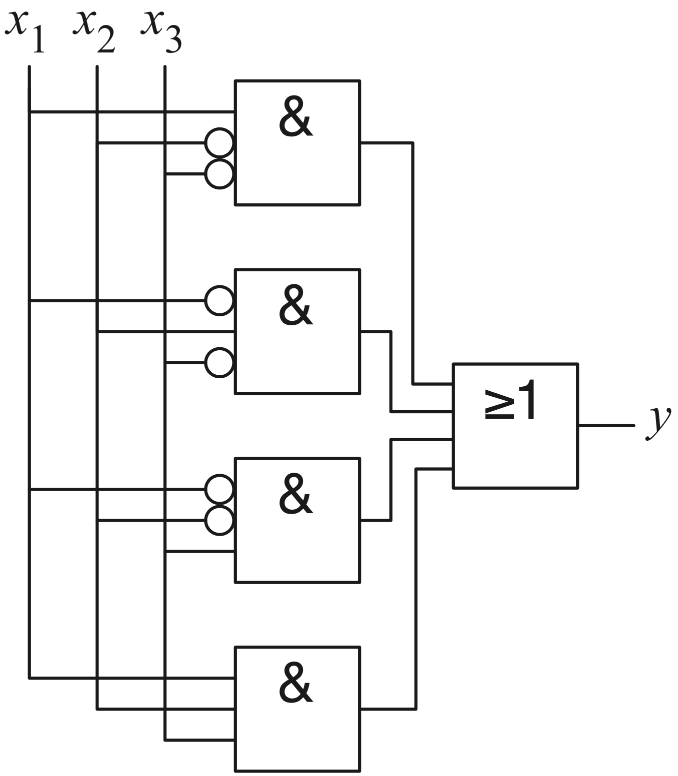
- Allgemeine Form

$$\bigwedge_{n \in N} \text{Maxterm}_n$$

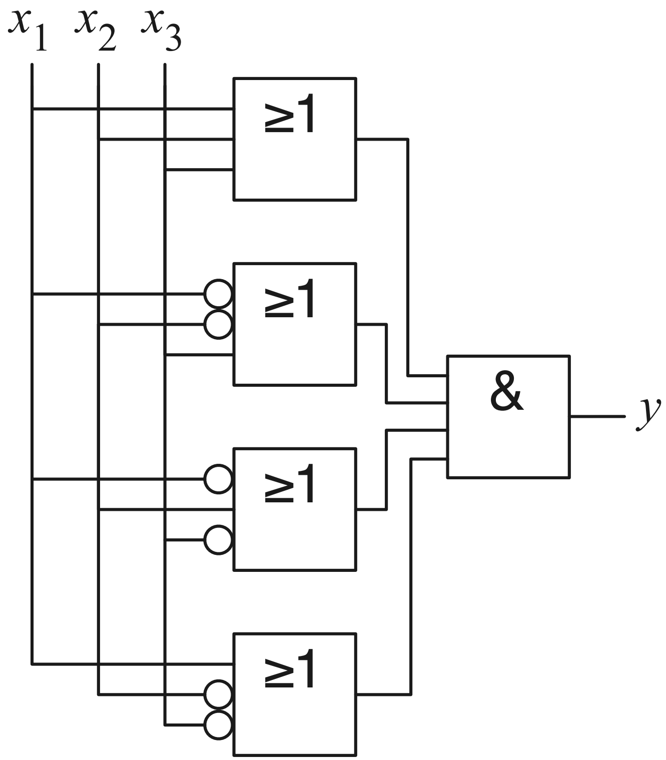
  - N = *Nullmenge* von f
  - Jeder *Maxterm* hat die Form

    $(L_n \vee \ldots \vee L_1) \quad L_i \in \{ x_i, \neg x_i \}$
  - Jedes L_i heißt ein *Literal* von f
-
- Abkürzungen
  - KNF (Konjunktive Normalform)
  - POS (Product of sums)

Jede Gleichung lässt sich 1:1 in Hardware umsetzen



$$y = (x_1 \wedge \neg x_2 \wedge \neg x_3) \vee (\neg x_1 \wedge x_2 \wedge \neg x_3) \vee (\neg x_1 \wedge \neg x_2 \wedge x_3) \vee (x_1 \wedge x_2 \wedge x_3)$$



$$y = (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_3) \wedge (x_1 \vee \neg x_2 \vee \neg x_3)$$