

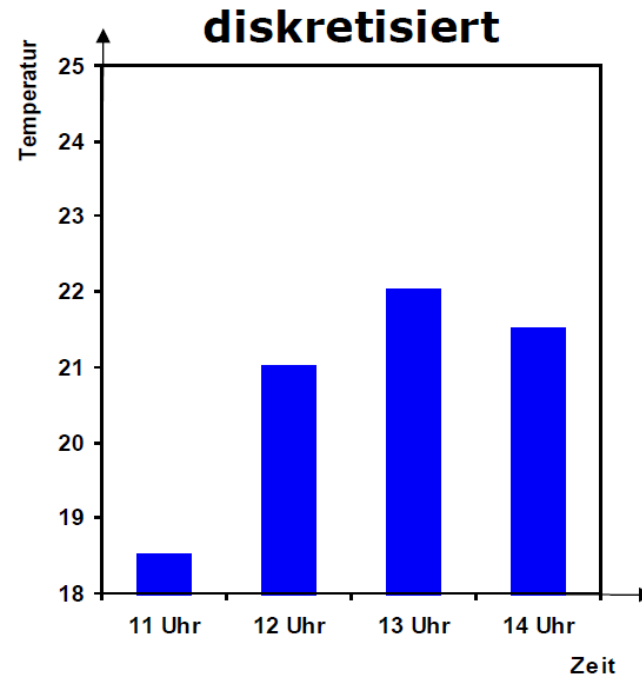
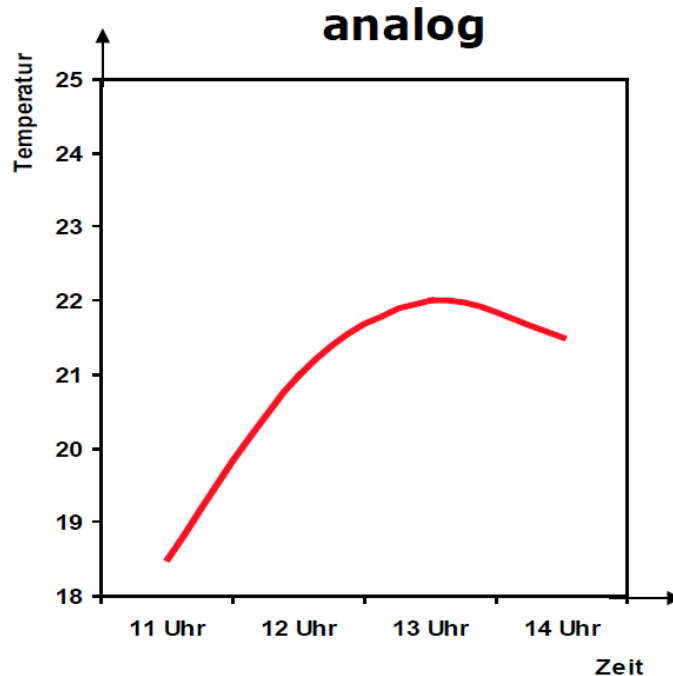
# Darstellung der Informationen im Digital-Computer

- bedingt durch die verwendete Technologie (Halbleiter) können Computer nur zwei Zeichen erkennen und verarbeiten: Strom/kein Strom (bzw. Spannung/keine Spannung)
- zur Verdeutlichung verwendet man eine Menge aus zwei Zeichen (binär System): 0, 1
- In einer Folge von Dualzahlen können beliebige Informationen dargestellt werden.

# Speicherung von Daten

- Alles was zähl- und messbar ist, kann durch binärcodierte Daten ausgedrückt werden
  - Zahlen, Texte
  - physikalische Größen (optische, akustische, ...)
- Dadurch können alle diese Daten durch Digitalrechner verarbeitet werden (Universalität der Daten).
- Die physikalischen Größen werden diskretisiert und digitalisiert, d.h. unendlich viele Werte werden durch endlich viele Werte ersetzt (Diskretisierung oder Rasterung von analogen Daten).
- Die endlich vielen Werte werden digital (durch Zahlen) dargestellt (Digitalisierung).

# Speicherung von Daten



|           |         |
|-----------|---------|
| 11.00 Uhr | 18.5 °C |
| 12.00 Uhr | 21.0 °C |
| 13.00 Uhr | 22.0 °C |
| 14.00 Uhr | 21.5 °C |

**digitalisierte Darstellung**

# Repräsentation von natürlichen Zahlen

Moderne Rechner arbeiten binär (Digitalrechner)

- **Grund**
  - technisch einfacher zu realisieren
  - damit preiswerter zu realisieren
  - bieten wohl-definierte Genauigkeit
  - sind leichter zu programmieren
- **Problem:** Wie repräsentiert man mit nur zwei Grundzuständen:
  - Zahlen?
  - Texte?
  - andere Daten?

# Bit

- Die Einheit zur binären Darstellung von Daten heißt **Bit** (**B**inary dig**it**). Ihr Inhalt wird meist mit 0 bzw. 1 codiert.
- Mit  $n$  Bit lassen sich  $2^n$  Zustände darstellen.
- Die technische Darstellung erfolgt u.a. mit Hilfe von:
  - Elektrische Ladung
    - 0 = ungeladen
    - 1 = geladen
  - Elektrische Spannung
    - 0 = 0 Volt
    - 1 = ca. 6 Volt
  - Magnetisierung
    - 0 = gleichbleibende Magnetisierung
    - 1 = Magnetisierungswechsel
  - Licht
    - 0 = kein Licht
    - 1 = Licht

# Bytes

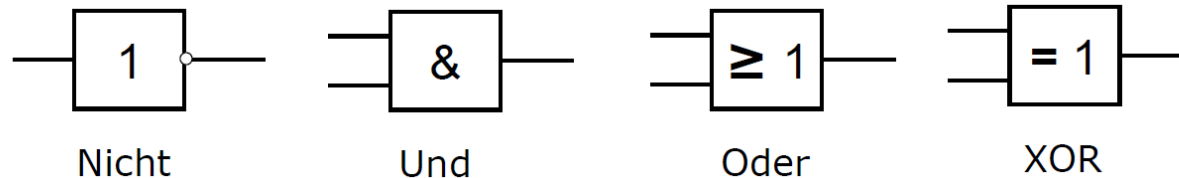
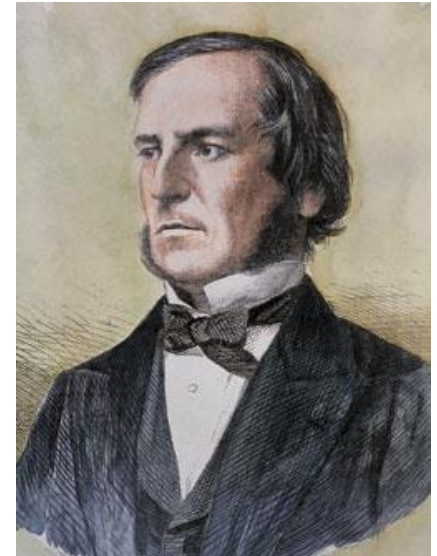
- Wenn ein Rechner Daten nach dem EVA-Prinzip verarbeitet operiert er immer mit Gruppen von 8, 16, 32 oder 64 Bits.
- Deswegen spricht man von 8-Bit-/16-Bit-/32-Bit- oder 64-Bit-Rechnern.
- Meist sind es jedoch Mischformen, z.B. Rechner, die intern 32-Bitfolgen verarbeiten, aber Folgen zu 64 Bits lesen oder schreiben.
- Immer jedoch ist die Länge einer Bitfolge ein Vielfaches von 8.
- Eine Gruppe von 8 Bit heißt Byte, abgekürzt 1B, und lässt sich durch zwei Nibble oder zwei Hexadezimalziffern darstellen.
- Abkürzungen
  - In der Informatik werden die Zahlen 2, 8, 16, 1024 und andere Vielfache der Zahl 2 besonders häufig verwendet, z.B.
    - $8 = 2 \cdot 2 \cdot 2 = 2^3$
    - $1024 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^{10}$

# Speicherung von Daten

- *Bit* (binary digit) ist der kleinstmögliche Speicherplatz
- Eine Folge von acht Bit wird *Byte* genannt (ein halbes Byte wird *Nibble* genannt)
- Byte ist die übliche Maßeinheit für die Speicherkapazität
- Bestimmte 2er-Potenzen werden in der Informatik häufig als Maßzahlen (z.B. für Speichergrößen) verwendet:
  - **1 K** =  $1024 = 2^{10}$  (K = **Kilo**)
  - **1 M** =  $1024 \cdot 1024 = 2^{20}$  (M = **Mega**)
  - **1 G** =  $1024 \cdot 1024 \cdot 1024 = 2^{30}$  (G = **Giga**)
  - **1 T** =  $1024 \cdot 1024 \cdot 1024 \cdot 1024 = 2^{40}$  (T = **Tera**)
  - **1 P** =  $1024 \cdot 1024 \cdot 1024 \cdot 1024 \cdot 1024 = 2^{50}$  (P = **Peta**)
  - **1 E** =  $1024 \cdot 1024 \cdot 1024 \cdot 1024 \cdot 1024 \cdot 1024 = 2^{60}$  (E = **Exa**)
- Für Längenangaben findet man häufig:  
**1" = 1 in = 1 inch = 1 Zoll = 2,54 cm**

# Logische Grundsaltungen

- Schöpfer der Aussagenlogik:  
englischer Mathematiker George Boole (1815 – 1864)
- Elementare logische Operationen
  - Negation (NICHT, NOT,  $\neg$ ,  $\sim$ , Überstrich)
  - Konjunktion (UND, AND,  $\wedge$ ,  $\cdot$ )
  - Disjunktion (ODER, OR,  $\vee$ ,  $+$ )
  - Antivalenz, Exklusiv-ODER (XOR,  $\oplus$ )
- Symbole nach Norm DIN 40 900

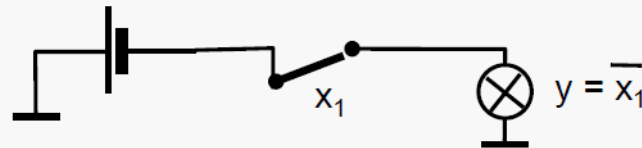




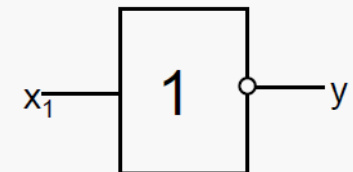
# Grundzüge der Booleschen Algebra

- Realisierung elementarer logischer Schaltungen
- Mit diesen Schaltungen lassen sich alle logischen Verknüpfungen realisieren!

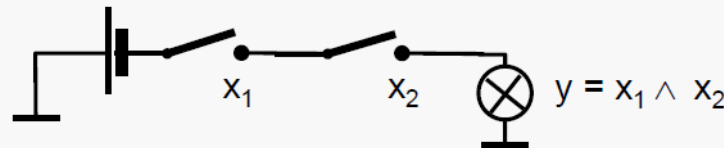
NICHT:  
(NOT)



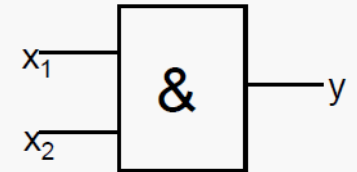
| $x_1$ | $y$ |
|-------|-----|
| 0     | 1   |
| 1     | 0   |



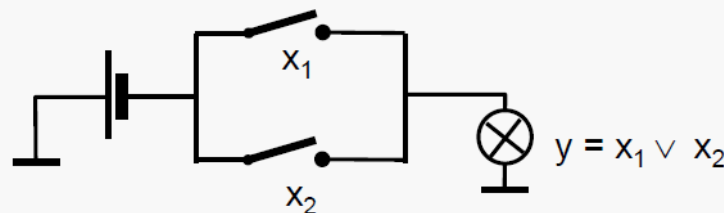
UND:  
(AND)



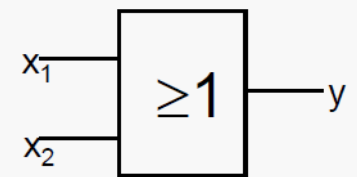
| $x_1$ | $x_2$ | $y$ |
|-------|-------|-----|
| 0     | 0     | 0   |
| 0     | 1     | 0   |
| 1     | 0     | 0   |
| 1     | 1     | 1   |



ODER:  
(OR)



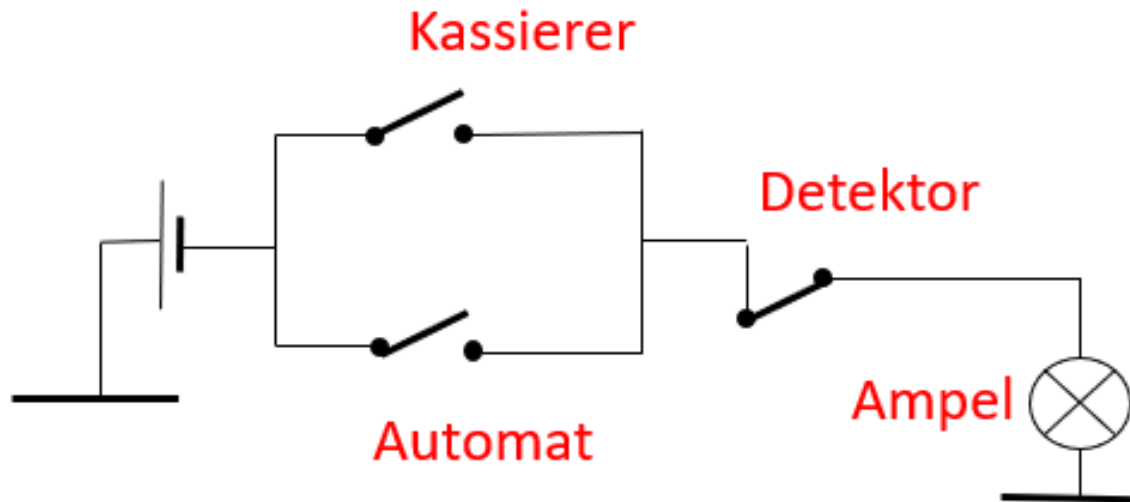
| $x_1$ | $x_2$ | $y$ |
|-------|-------|-----|
| 0     | 0     | 0   |
| 0     | 1     | 1   |
| 1     | 0     | 1   |
| 1     | 1     | 1   |



# Grundzüge der Booleschen Algebra

- Die Ampel an der Einfahrt einer Autowaschanlage soll grün aufleuchten, wenn
  - Der Kassierer per Knopfdruck die Zahlung bestätigt  
ODER
  - Der Zahlungsautomat die Zahlung meldet  
UND
  - Der Detektor im Waschraum kein Waschwasser (vom Vorgänger) mehr meldet

# Grundzüge der Booleschen Algebra

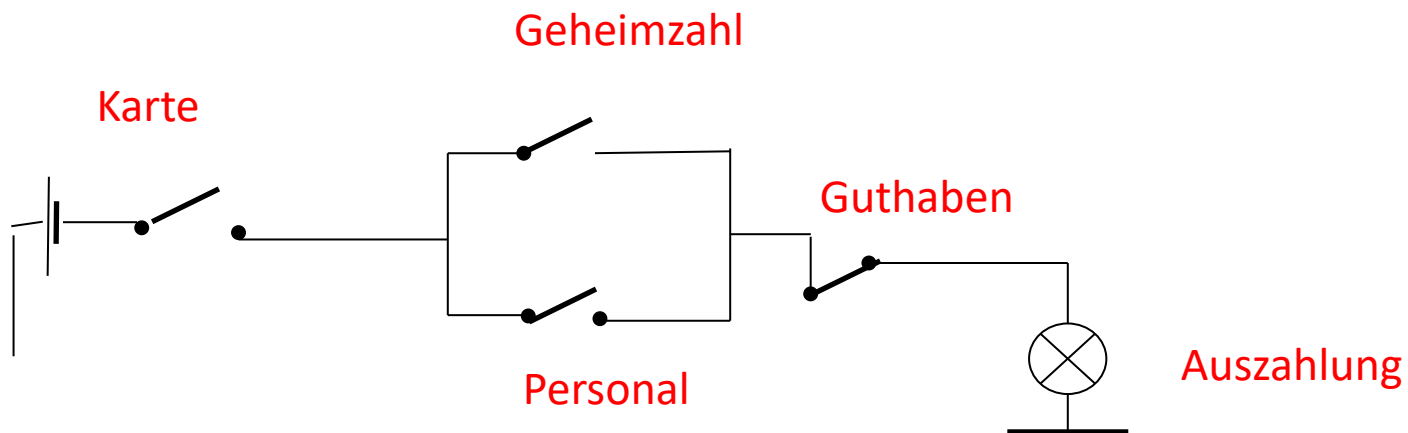


$(\text{Kassierer} \vee \text{Automat}) \wedge \text{Detektor} = \text{Ampelanzeige}$

$(\text{Kassierer ODER Automat}) \text{ UND Detektor} = \text{Ampelanzeige}$

## Weiteres Beispiel

- Eine Bank bietet folgenden Service an:  
nach Vorlage der EC-Karte hat man nunmehr die Wahl, entweder wie bisher am Automat die Geheimzahl einzutippen oder sich bei einem bekannten Mitarbeiter persönlich zu melden; dieser überprüft das vorhandene Guthaben und meldet im Erfolgsfall der Kasse, dass es zur Auszahlung kommen kann.



$\text{Karte} \wedge (\text{Geheimzahl} \vee \text{Personal}) \wedge \text{Guthaben} = \text{Auszahlung}$

Karte UND (Geheimzahl ODER Personal) UND Guthaben = Auszahlung