Informationsdarstellung

Prof. Dr. Aris Christidis

- Signale und Logik
- Grundzüge der Booleschen Algebra
- Signale und Logik (2)
- Grundzüge d. Informationstheorie

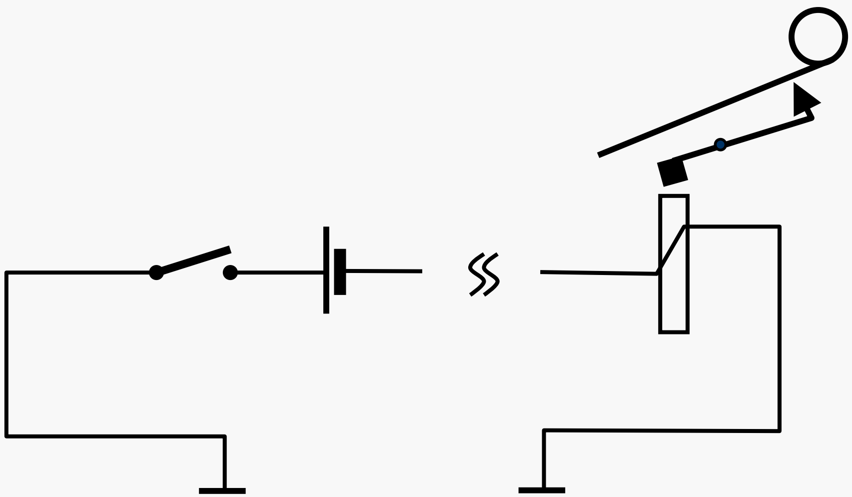
  [Logarithmen-Repetitorium]
- Zahlensysteme und ihre Anwendung
- Signale und Logik (3)
- Rechnen mit Signalen
- Darstellung von Rechenwegen

¬

Signale und Logik

Prof. Dr. Aris Christidis

Erfindung des Telegraphen (Samuel Morse, 1837):
Energieform wird als Signal genutzt (vgl. Feuer)



Signal: veränderliche physikalische Größe, die zur Informations-darstellung u./o. -übertragung eingesetzt wird.

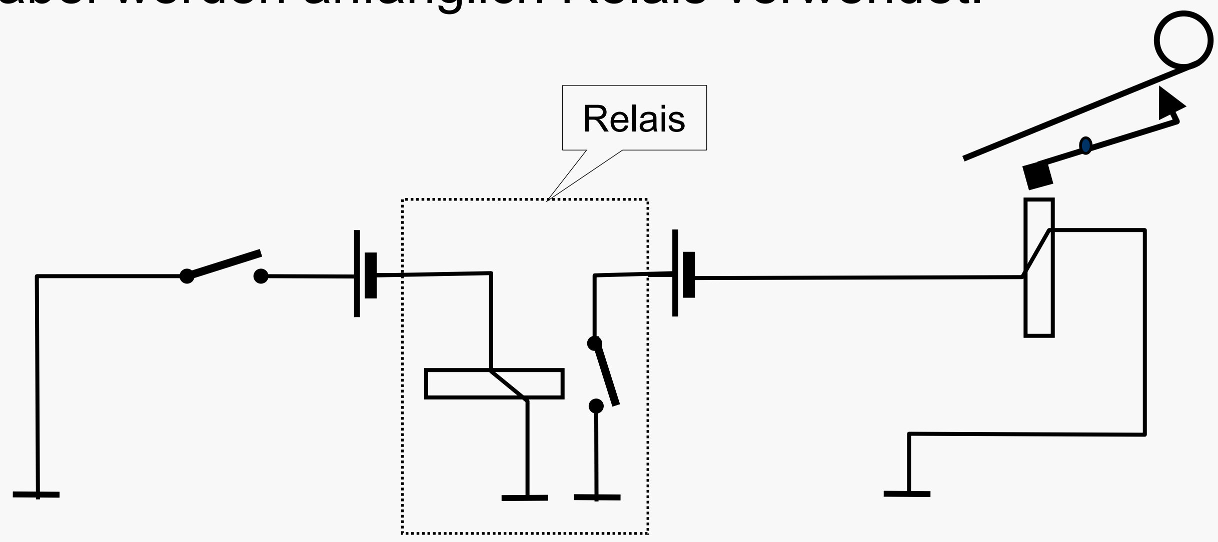
Übertragung nicht von Energie, sondern von (unterscheidbaren) Zuständen: Nutzen unabhängig von eingesetzter Leistung.

Signale und Logik

Prof. Dr. Aris Christidis

Für Übertragungen über große Entfernungen muß das Signal verstärkt werden (Spannungsabfall entlang d. Telegraphenleitung).

Dabei werden anfänglich Relais verwendet.

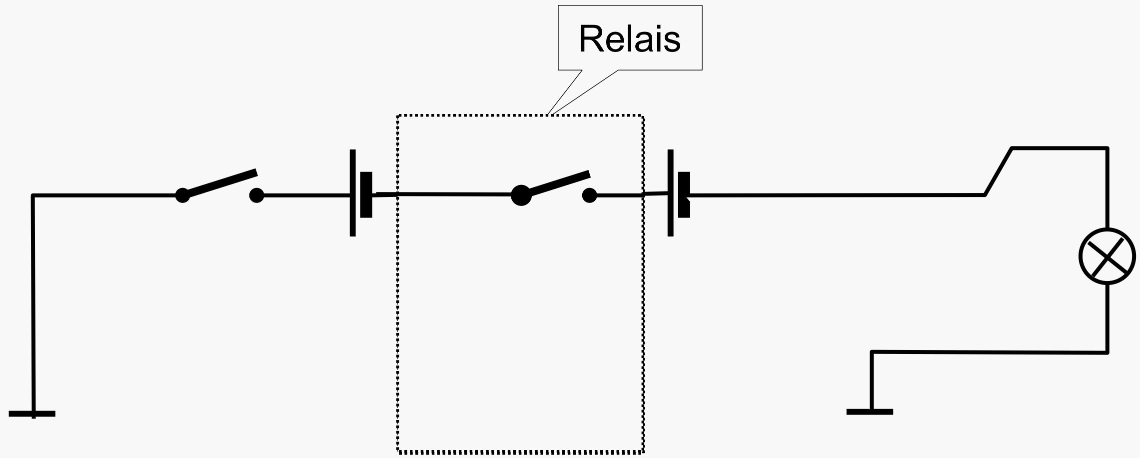


Relais trennen die Stromkreise von Sender u. Empfänger und erleichtern den Zusammenschluß mehrerer Sende- u. Empfangs-stationen.

Signale und Logik

Prof. Dr. Aris Christidis

Unabhängig von der Telegraphie bewirkte der Einsatz von Relais auch eine Trennung von Nutz- u. Schaltsignal.



Anfang des 20. Jh. werden Relais von Elektronenröhren und diese Mitte des Jh. von Transistoren abgelöst.

Allen gemeinsam ist die Eignung zu sog. logischen Schaltungen (im folgenden symbolisch als einfache Schalter dargestellt).

Grundzüge der Booleschen Algebra

Die Boolesche Algebra (Schaltalgebra) bietet Formalismen zur Behandlung logischer Beziehungen und Verknüpfungen, wie sie z.B. in logischen Schaltungen vorkommen.

Sie nutzt die Regeln der Mengenlehre und dient als mathematisches Hilfsmittel

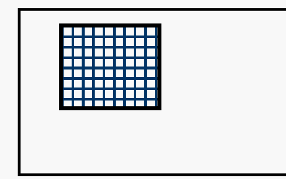
- bei der Beschreibung von Signal-Verknüpfungen und speicherfreien (sog. kombinatorischen) Schaltungen,
- beim Schaltungsentwurf bei vorgeschriebener Funktion,
- bei der Minimierung des technischen Aufwands.

Grundzüge der Booleschen Algebra

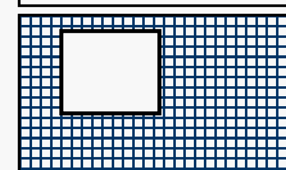
Logische (Boolesche) Variablen x_i (i=1,...) sind **binär** - d.h.: sie nehmen genau einen von zwei Werten (Zuständen) an: $x_i \in \{0,1\}$

Es gibt 4 logische Grundfunktionen. Sie sind beschrieben durch ihre Venn-Diagramme, logischen Schaltungen („Gatter") und Wertetabellen („Zustands"- oder „Wahrheitstabellen").

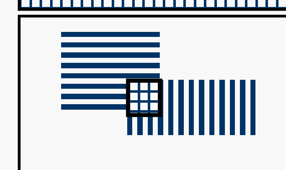
- Identität: $y = x_1$



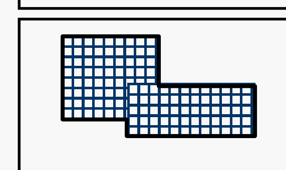
- Negation: $y = \overline{x_1}$ (NICHT x_1 ; x_1 negiert)



- Konjunktion: $y = x_1 \wedge x_2$ (x_1 UND x_2)



- Disjunktion: $y = x_1 \vee x_2$ (x_1 ODER x_2)



Grundzüge der Booleschen Algebra

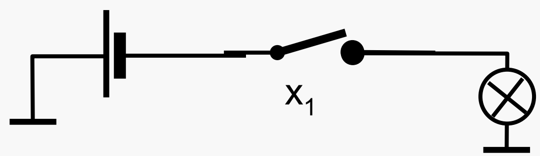
Prof. Dr. Aris Christidis

Realisierung elementarer logischer Schaltungen (Gatter):

[1: Schalter geschlossen, Lampe leuchtet]

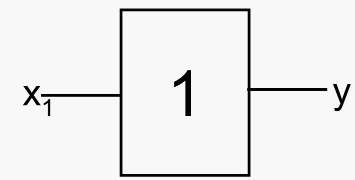
GLEICH:

(EQUAL)



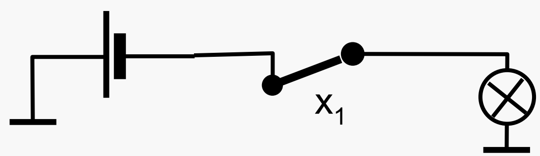
$y = x_1$

x_1	y
0	0
1	1



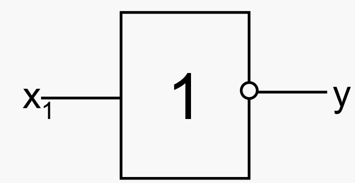
NICHT:

(NOT)



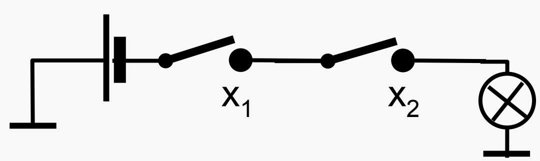
$y = \overline{x_1}$

x_1	y
0	1
1	0



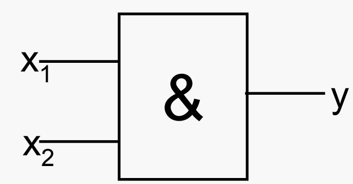
UND:

(AND)



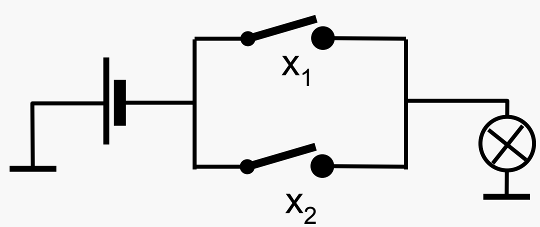
$y = x_1 \wedge x_2$

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1



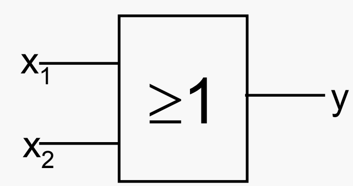
ODER:

(OR)



$y = x_1 \vee x_2$

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1



Mit diesen Schaltungen lassen sich alle logischen Verknüpfungen realisieren!

Grundzüge der Booleschen Algebra

Beispiel:

Die grüne Ampel an der Einfahrt einer Autowaschanlage soll aufleuchten, wenn

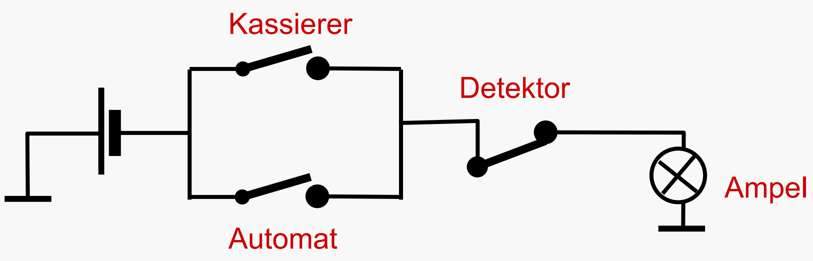
- der Kassierer per Knopfdruck die Zahlung bestätigt

ODER

- der Zahlungsautomat die Zahlung meldet

UND

- der Detektor im Waschraum kein Waschwasser (vom Vorgänger) mehr meldet:



Grundzüge der Booleschen Algebra

Prof. Dr. Aris Christidis

Rechenregeln:

$\overline{1} = 0$; $\overline{0} = 1$

$\overline{\overline{x}} = x$ (Identität als doppelte Negation)

Dualitätsprinzip:

Vertauschen von $\wedge$ durch $\vee$, 0 durch 1 und umgekehrt ergibt immer eine neue Regel!

Konjunktion (AND):

$0 \wedge x = 0$

$1 \wedge x = x$

$x \wedge x = x$

$x \wedge \overline{x} = 0$

Disjunktion (OR):

$0 \vee x = x$

$1 \vee x = 1$

$x \vee x = x$

$x \vee \overline{x} = 1$

Ähnlichkeiten zwischen:

AND u. Multiplikation

OR u. Addition

Grundzüge der Booleschen Algebra

Prof. Dr. Aris Christidis

Rechenregeln:

$\overline{1} = 0$; $\overline{0} = 1$

$\overline{\overline{x}} = x$ (Identität als doppelte Negation)

Dualitätsprinzip:

Vertauschen von $\wedge$ durch $\vee$, 0 durch 1 und umgekehrt ergibt immer eine neue Regel!

<u>Konjunktion (AND):</u>

$0 \bullet x = 0$

$1 \bullet x = x$

$x \bullet x = x$

$x \bullet \overline{x} = 0$

<u>Disjunktion (OR):</u>

$0 + x = x$

$1 + x = 1$

$x + x = x$

$x + \overline{x} = 1$

(Nach Klammern und Negation:)

AND- vor OR-Rechnung!

(vgl. Punkt-/Strichrechnung)

Grundzüge der Booleschen Algebra

Prof. Dr. Aris Christidis

Kommutativgesetze (Vertauschungsgesetze)

$x_1 \wedge x_2 = x_2 \wedge x_1$

$x_1 \vee x_2 = x_2 \vee x_1$

Assoziativgesetze (Verbindungsgesetze)

$x_1 \wedge x_2 \wedge x_3 = x_1 \wedge (x_2 \wedge x_3) = (x_1 \wedge x_2) \wedge x_3$

$x_1 \vee x_2 \vee x_3 = x_1 \vee (x_2 \vee x_3) = (x_1 \vee x_2) \vee x_3$

Distributivgesetze (Verteilungsgesetze)

$$\left.\begin{array}{l} x_1 \wedge (x_2 \vee x_3) = x_1 \wedge x_2 \vee x_1 \wedge x_3 \\ x_1 \vee (x_2 \wedge x_3) = (x_1 \vee x_2) \wedge (x_1 \vee x_3) \; ! \end{array}\right\} \text{Dualitätsprinzip der Booleschen Algebra}$$

Grundzüge der Booleschen Algebra

Absorptionsgesetze:

$$x_1 \vee \; x_1 \wedge x_2 \; = x_1$$

$$x_1 \wedge (x_1 \vee x_2) = x_1$$

$$x_1 \wedge (\overline{x_1} \vee x_2) = x_1 \wedge x_2$$

$$x_1 \vee (\overline{x_1} \wedge x_2) = x_1 \vee x_2$$

Regeln von De Morgan: (Negationsregeln)

$$\overline{x_1 \wedge x_2} = \overline{x_1} \vee \overline{x_2}$$ (NAND = NOT OR NOT)

$$\overline{x_1 \vee x_2} = \overline{x_1} \wedge \overline{x_2}$$ (NOR = NOT AND NOT) $\neg$

Grundzüge der Booleschen Algebra

Prof. Dr. Aris Christidis

Beispiel:

Wenn Gegenstand der Informatik alles ist, was der Erfassung, (oder) Speicherung, (oder) Bearbeitung, (oder) Übertragung, (oder) Umsetzung von Daten dient, was bildet dann <u>keinen</u> Gegenstand dieser Wissenschaft?

$$G_{(I)} = E \vee S \vee B \vee Ü \vee U$$

Antwort:

Alles, was nicht der Erfassung, (und nicht) der Speicherung, (und nicht) der Bearbeitung, (und nicht) der Übertragung (und nicht) der Umsetzung von Daten dient.

$$\overline{G_{(I)}} = \overline{E \vee S \vee B \vee Ü \vee U}$$

$$= \overline{E} \wedge \overline{S} \wedge \overline{B} \wedge \overline{Ü} \wedge \overline{U} \qquad \text{(De Morgan)}$$

¬

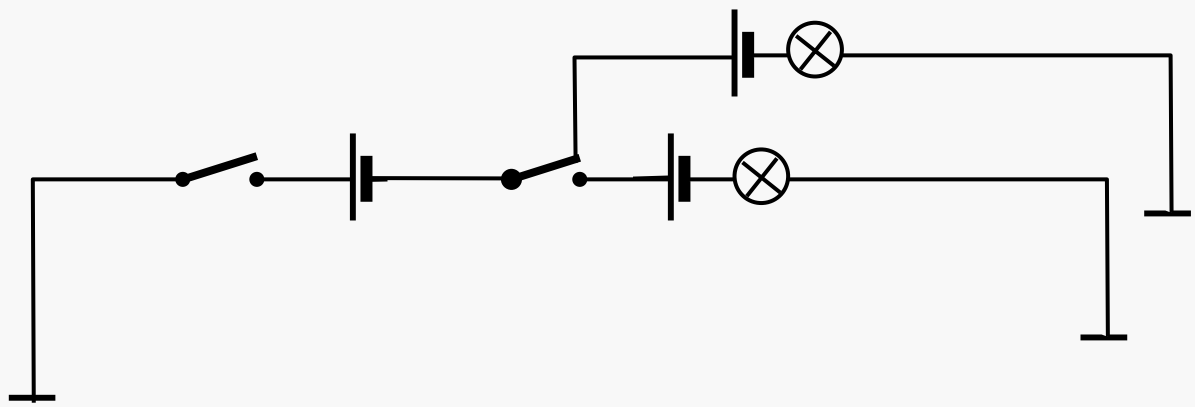
Signale und Logik (2)

Prof. Dr. Aris Christidis

Vereinfachte (Relais-/)Schalterdarstellung:

Trennung von Nutz- u. Schaltsignal macht Adressaten „anwählbar“:

Bei zwei möglichen Empfängern genügt ein Schalter, um einen der beiden zu wählen.

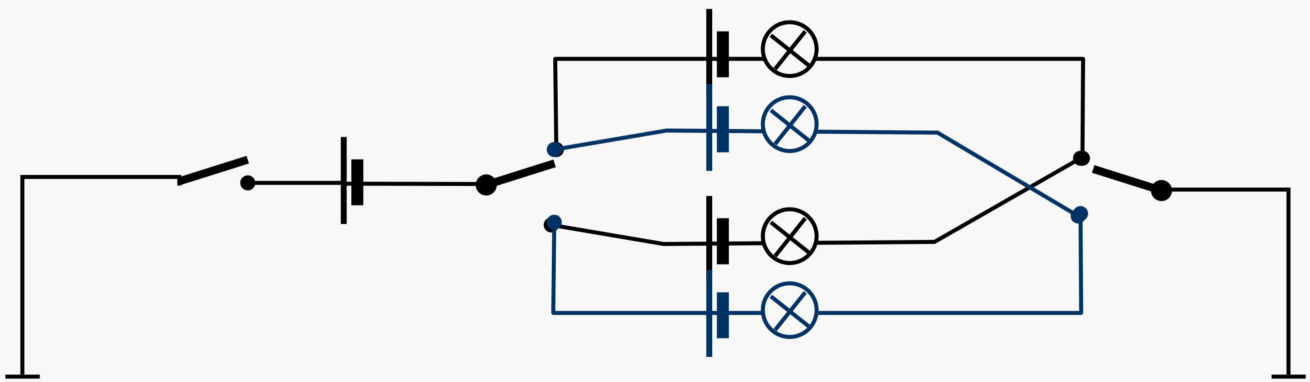


(Hier: Umwidmung des Verstärker-Relais für die Adressierung - in Wirklichkeit benötigen Verstärkung u. Zuschaltung getrennte Relais!)

Signale und Logik (2)

Prof. Dr. Aris Christidis

Zur getrennten Ansteuerung von 4 Empfängern werden 2 Schalter notwendig: (... und weiter?)



Benötigt werden Aussagen darüber, wie eine gegebene Anzahl unterscheidbarer Zustände (hier: Empfängerwahl) mit möglichst wenigen binären Elementen (hier: Schaltern) eindeutig darzustellen -zu „codieren“- ist.

Das ist Gegenstand der Informationstheorie.

Grundzüge der Informationstheorie

Die Informationstheorie untersucht **Darstellung**, **Speicherung** und **Übertragung** von Information.

Anmerkungen:

- Im Vergleich zur (später entstandenen) Informatik überwiegen hier formal-theoretische Aspekte wie Wesen, Erhaltung oder Wiedergewinnung von Information.
- Die Informationstheorie bedient sich meist der Methoden der Wahrscheinlichkeitsrechnung und der mathematischen Statistik.

Grundzüge der Informationstheorie

- Information benötigt zu ihrer Darstellung **Symbole** (=unterscheidbare physikalische Zustände).

  Bsp.: ' * ' (auf Schreibmaschinen- oder Telefontastatur)

  auch: Gesten, Körperhaltungen, Flaggenstellungen

- Ein oder mehrere Symbole werden als **Zeichen** mit eigener Bedeutung definiert (= vereinbart).

  Bsp.: ' • —— ' (‘a‘ bei Morse) oder ' **;** ‘

  auch: Schrift-, Licht-, Hand-, Vogelzeichen

- Ein geschlossen verwendeter, geordneter, verbreiteter Vorrat von mindestens zwei Zeichen bildet ein **Alphabet**.

  Bsp.: Griechisches Alphabet, lateinisches ABC,

  American **S**tandard **C**ode for **I**nformation **I**nterchange (ASCII)¬

Grundzüge der Informationstheorie

Prof. Dr. Aris Christidis

Anmerkungen:

- Es gibt Zeichen, die sich aus Symbolen zusammensetzen: '**?**' und '**!**' bestehen z.B. aus je 2 Symbolen, von denen das jeweils obere allein bedeutungslos ist.
- Flaggenstellungen können Nachrichten durch Flaggenalphabet aber auch Geisteshaltungen u.ä. übermitteln („Halbmast").
- Die Frage, ob ein Alphabet auch geordnet sein muß, wird in der Literatur widersprüchlich behandelt: Gehören die ungeordneten Satzzeichen (, . ;) zum ABC?
- Zur Bezeichnung 'Alphabet' gehört eine gewisse Standardisierung, Verbreitung oder Bekanntheit; insofern kann Code auch die Entsprechungen zwischen willkürlich ausgesuchten Zeichenmengen betreffen (vgl. Verschlüsselung).
- Am Morse-Bsp. wird deutlich, wie wichtig die Code-Wahl ist: 'E' ist im Englischen -wie im Dt.- der häufigste Buchstabe, deshalb im Morse-Alphabet der kürzeste.

Die eindeutige Zuordnung zwischen Alphabeten wird **Code**, ihr Einsatz **Codierung** genannt.

Anmerkungen:

- Codierungen sind nicht immer umkehrbar eindeutig!

  Bsp.: Der Wechsel zwischen dt. Klein- u. Großbuchstaben:

  'Gießen' $\Rightarrow$ 'GIESSEN'

  Bsp.: Der Morse-Code ist umkehrbar eindeutig, weil jedes Zeichen mit einem Leerzeichen abgeschlossen wird:

  'Eis?' $\Leftrightarrow$ '• •• ••• ••– –••'

- Auch die Umsetzung abstrakter Information (Gedanken) in eine Sprache erfolgt durch Codierung; Programmiersprachen sind Gegenstand (u.a.) der Praktischen Informatik, menschliche Sprachen (u.a.) der Linguistik (vgl. Soziolinguistik: restringierter, elaborierter Code) ¬

- Innerhalb eines Codes kann die Anzahl der Zeichen (Symbole) in einem Codewort variieren oder für alle Wörter gleich sein.

  Bsp.: Im Morse-Code sind Zeichen ungleich lang, im ASCII-Code gleich lang:

Schriftzeichen	Morse	ASCII
e	•	`0110 0101`
?	• • — — • •	`0011 1111`

- Codes mit einem Zeichenvorrat von nur zwei Zeichen (z.B. 0 und 1) werden **Binärcodes** genannt. Sie sind technisch interessant, weil sie auch mit einem Schalter (Transistor) realisiert werden können.

  Der Morse-Code ist als Binärcode nicht umkehrbar eindeutig!¬

Grundzüge der Informationstheorie

Prof. Dr. Aris Christidis

- Codes und Codierung bilden einen wichtigen Teil der Informationstheorie; denn sie sind eine grundlegende Voraussetzung für die maschinelle DV.

- **Optimal** werden Codes genannt, die je Zeichen möglichst viel Information verschlüsseln (=codieren). Dazu bedarf es einer Metrik für Information.

Grundzüge der Informationstheorie

- Der **Informationsgehalt** *h* einer Nachricht bzw. eines Zeichens *x* hängt von der Wahrscheinlichkeit *p(x)* seines Auftretens ab: Je unwahrscheinlicher das Auftreten eines Zeichens *x* ist, desto höher ist sein Informationsgehalt:

$$h(x) = f\ (\ 1/p(x)\ )$$

(f: vorerst unbekannte Funktion)

- Für den Informationsgehalt des Empfangs mehrerer voneinander unabhängiger Zeichen folgt daraus:
  - Es muß gelten (Wahrscheinl/keit gleichzeitigen Auftretens):

$$h(xy...) = f\ (\ p(x)^{-1} \cdot p(y)^{-1} \cdot ...)$$

  - Anschauungshalber sollte außerdem gelten:

$$h(xy...) = h(x) + h(y) + ... \qquad \text{d.h.:}$$

$$f\ (\ p(x)^{-1} \cdot p(y)^{-1} \cdot ...) = f\ (\ p(x)^{-1}\ ) + f\ (\ p(y)^{-1}\ ) + ... \neg$$

Grundzüge der Informationstheorie

Prof. Dr. Aris Christidis

Zusammenfassung:
Forderungen an den Informationsgehalt *h* d. Nachrichten *x*, *y*...:

$$h(xy...) = f\,(p(x)^{-1} \cdot p(y)^{-1} \cdot ...) = f\,(p(x)^{-1}) + f\,(p(y)^{-1}) + ... = h(x) + h(y) + ...$$

frei assoziiert:

$$\log\,(x \cdot y \cdot ...) = \log x + \log y + ...$$

Definition (C. Shannon):

Der **Informationsgehalt** *h* eines Zeichens *x* ist definiert als der Logarithmus dualis des Reziprokwertes der Wahrscheinlichkeit, mit der das Zeichen auftritt:

$$\begin{aligned} h(x) &= \mathrm{ld}\,[\,1/p(x)\,] \\ &= -\,\mathrm{ld}\,[\,p(x)\,] \end{aligned}$$

¬

Kurzes Logarithmen-Repetitorium

Prof. Dr. Aris Christidis

Definition: $\log_B Z = L \quad \Leftrightarrow \quad B^{\log_B Z} \equiv B^L = Z$

Speziell:

$\lg Z \equiv \log_{10} Z \qquad \ln Z \equiv \log_e Z \qquad \mathrm{ld}\, Z \equiv \log_2 Z$

$$(\text{Def.}) \Rightarrow Z_1 \cdot Z_2 = B^{L1} \cdot B^{L2} = B^{L1+L2}$$

Interessante Regeln:

$\log (Z_1 \cdot Z_2) = \log (Z_1) + \log (Z_2)$	$\log (Z^E) = E \cdot \log Z$
$\log (Z_1 / Z_2) = \log (Z_1) - \log (Z_2)$	$\log (\sqrt[W]{Z}) = (\log Z) / W$

Kurzes Logarithmen-Repetitorium

Prof. Dr. Aris Christidis

Wechsel der Logarithmenbasis:

$$Z = B_1^{\log_{B1} Z}$$

$$\log_{B2} Z = \log_{B2}(B_1^{\log_{B1} Z})$$

$$= \log_{B1} Z \cdot \log_{B2} B_1$$

$$\boxed{\log_{B1} Z = \log_{B2} Z \,/\, \log_{B2} B_1}$$

Oft benötigte Umrechnungen:

$$\lg 10 = \ln e = \mathrm{ld}\ 2 = 1$$

$$\lg 1 = \ln 1 = \mathrm{ld}\ 1 = 0$$

$$\mathrm{ld}\ (1/x) = -\mathrm{ld}\ x \quad ; \quad \lg (1/x) = -\lg x$$

$$\lg (10^x) = \ln (e^x) = \mathrm{ld}\ (2^x) = x$$

$$10^{\lg x} = e^{\ln x} = 2^{\mathrm{ld}\ x} = x$$

$$\mathrm{ld}\ x = \lg x / \lg 2 = \ln x / \ln 2$$

Grundzüge der Informationstheorie

Prof. Dr. Aris Christidis

Aus der Definition des Informationsgehalts

$$h(x) = -\mathrm{ld}\,[\,p(x)\,]$$

folgt:

Der Informationsgehalt einer aus mehreren (voneinander unabhängigen) Zeichen bestehenden **Sequenz** ist gleich der **Summe** der Informationsgehalte der einzelnen Zeichen:

$$h(xy...) = -\mathrm{ld}\,[\,p(x) \cdot p(y) \cdot ...]$$

$$= -\mathrm{ld}\,[\,p(x)] - \mathrm{ld}\,[\,p(y)\,] - ...$$

$$= h(x) + h(y) + ...$$

ursprüngliche Forderung ¬

Grundzüge der Informationstheorie

Prof. Dr. Aris Christidis

Was bedeutet: „Informationsgehalt von *x* ist eins“?

$$h(x) = -ld\ [\ p(x)\ ] = ld\ [\ 1/p(x)\ ] = 1,$$

woraus folgt:

$$1/p(x) = 2^1 \text{ bzw. } p(x) = ½\ ,$$

d.h.:

- Der Informationsgehalt der Antwort auf eine Frage, die nur zwei (gleich wahrscheinliche) Möglichkeiten zuläßt, ist die Einheit des Informationsgehalts; sie wird **bit** genannt (**b**asic **i**ndissoluble **i**nformation uni**t**).

- Ein bit ist der Informationsgehalt eines Zeichens in einem binären Alphabet mit gleicher Auftretungswahrscheinlichkeit.
  Das binäre Alphabet kann aus den Wertepaaren bestehen: ja/nein; wahr/falsch; schwarz/weiß ; hell/dunkel etc.

Grundzüge der Informationstheorie

Prof. Dr. Aris Christidis

Beispiel: Häufigkeit von Buchstaben im Deutschen

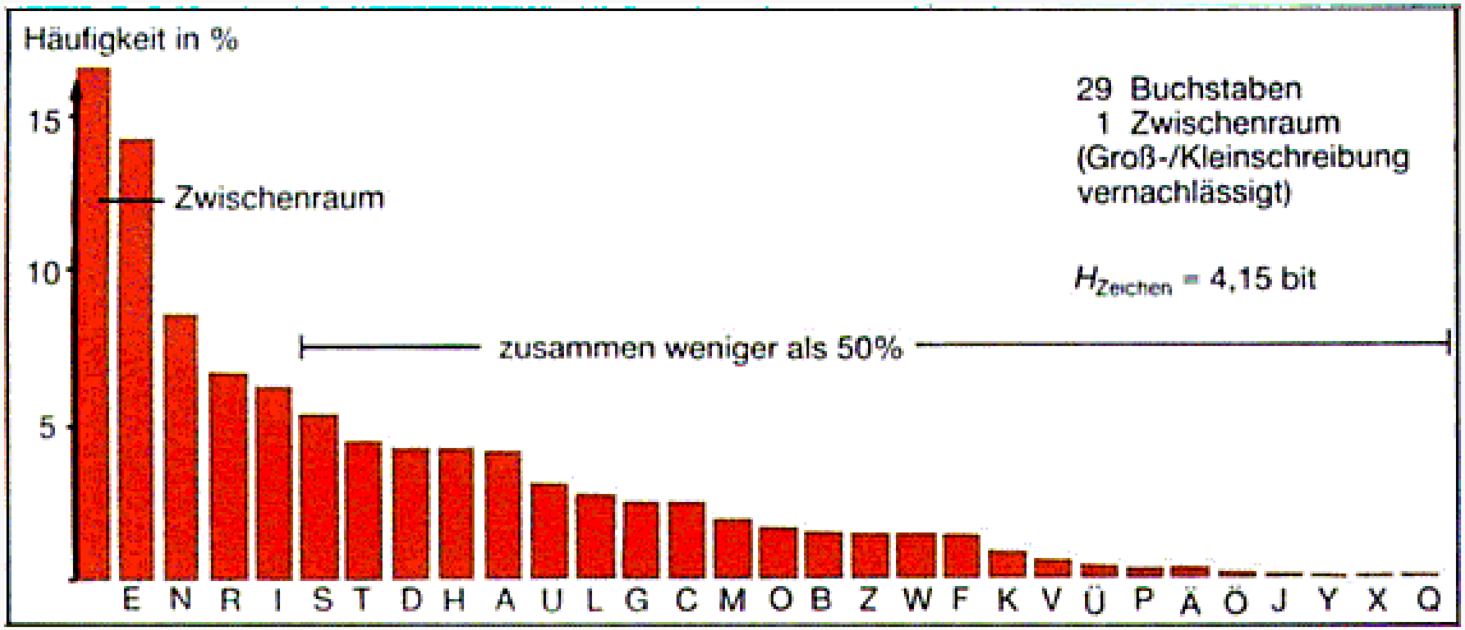


Bild: H.Breuer: „dtv-Atlas zur Informatik", dtv 1995

Ist die Wahrscheinlichkeit für das Auftreten des Buchstabens E: p(E)=0,147 und die für H: p(H)=0,045, so ist der Informationsgehalt der Zeichenkette 'EHE' [1]:

$$-\mathrm{ld}\,(0{,}147 * 0{,}045 * 0{,}147)\text{ bit} \approx [\,-\lg(0{,}066) / \lg 2\,]\text{ bit}$$
$$\approx [\,1{,}18 / 0{,}30\,]\text{ bit} \approx 3{,}9\text{ bit}$$

1) (Zeichenkette gebildet mit zufällig aus einem Text herausgegriffenen Zeichen)

Grundzüge der Informationstheorie

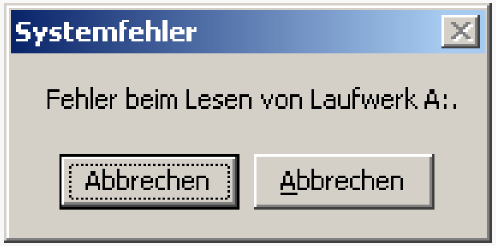
- Die Einheit zur binären Darstellung von Daten heißt **Bit** (**B**inary dig**it**). Ihr Inhalt wird meist mit 0 bzw. 1 codiert.

- Mit *n* Bit lassen sich 2^n Zustände darstellen.
- Es gibt nur ganzzahlige Bit – im Gegensatz zu bit.
- Zur Darstellung von *n* bit benötigt eine elektronische Rechenanlage mindestens *n* Bit.
- Informationsgehalt wird in bit **berechnet**: Es gibt keine Instrumente zu seiner Messung.

Grundzüge der Informationstheorie

Die technische Darstellung erfolgt u.a. mit Hilfe von:

- Ladung
  - 0 = ungeladen
  - 1 = geladen
- Spannung
  - 0 = 0 Volt
  - 1 = ca. 6 Volt
- Magnetisierung
  - 0 = gleichbleibende Magnetisierung
  - 1 = Magnetisierungswechsel
- Licht
  - 0 = kein Licht
  - 1 = Licht
- ...



Grundzüge der Informationstheorie

Prof. Dr. Aris Christidis

- Aus bestimmten technischen Gründen wie:
  - Geschwindigkeit von Lese- und Schreiboperationen
  - Darstellungsmöglichkeit „häufiger“ Zeichen (z.B. Alphabet)
  - Darstellungsmöglichkeiten von Zahlen, etc.

  werden in der Informatik oft Vielfache von 8-Bit-Gruppen verwendet (8Bit, 16Bit, ...)

  Eine 8-Bit-Sequenz heißt ein **Byte.**
- Bestimmte 2er-Potenzen werden in der Informatik häufig als Maßzahlen (z.B. für Speichergrößen) verwendet:
  - 1 kByte = 2^{10} Byte = 1024 Byte (1 Kilobyte)
  - 1 Mbyte = $2^{10} \cdot 2^{10}$ Byte (1 Megabyte)
  - 1 Gbyte = $2^{10} \cdot 2^{10} \cdot 2^{10}$ Byte (1 Gigabyte)
  - 1 Tbyte = $2^{10} \cdot 2^{10} \cdot 2^{10} \cdot 2^{10}$ Byte (1 Terabyte)

Grundzüge der Informationstheorie

Prof. Dr. Aris Christidis

Die Binärcodierung von ***n*** Zuständen benötigt ein Wort der Länge ***l*** (l Bit):

$2^l \geq n \quad \Rightarrow \quad ld(2^l) = l * ld\ 2 \geq ld\ n \quad \Rightarrow \quad \mathbf{l \geq ld\ n} \qquad (ld\ 2=1)$

Beispiel: Codierung der Himmelsrichtungen (N, O, S, W – d.h.: n=4)

$l \geq ld\ 4 = ld\ (2^2) = 2 * ld2 \quad \Rightarrow \quad l \geq 2$ Bit

- Eine mögliche Codierung ist:
  - 00 = Norden
  - 01 = Osten
  - 10 = Süden
  - 11 = Westen
- Obige Codierung läßt sich in zwei Fragen umsetzen:
  - Weht der Wind aus N bzw. O (ja / nein) ?
  - Weht der Wind aus O bzw. W (ja / nein) ?
- Nimmt man noch die Zwischenrichtungen NO, SO, SW und NW hinzu, so können die 8 Zustände mit 3 Bit codiert werden (wie?)

Grundzüge der Informationstheorie

Prof. Dr. Aris Christidis

- **Redundanz** ist ein Maß für die Nicht-Nutzung der Möglichkeiten eines Codes bzw. Maß für den Anteil einer Nachricht, der keine Information enthält.

- Im Falle gleich wahrscheinlicher, gleich lang codierter Wörter (*l* binäre Zeichen [Bit]) ist die Redundanz *r* definiert als Differenz zwischen der Wortlänge und Informationsgehalt *h* [bit]:

$$r = l - h$$

Redundanz wird oft genutzt zur Erstellung prüfbarer u. korrigierbarer Codierungen.

Prof. Dr. Aris Christidis

Ein Beispiel:

Die Binärcodierung eines Tripels (3 Zustände, 3 Spieler,...) benötigt eine Wortlänge von l=2 Bit; denn es ist gefordert:

$$2^l \geq 3 \;\Rightarrow\; \mathrm{ld}(2^l) \geq \mathrm{ld}\ 3 \Rightarrow l \cdot \mathrm{ld}2 \geq \mathrm{ld}\ 3$$

(ld2=1)

$$\Rightarrow \quad l \geq \mathrm{ld}\ 3 = \lg 3 / \lg 2 = 0{,}477/0{,}301 = 1{,}585 \text{ Bit}$$

d.h., die kleinste ganze Zahl Bit, die ausreicht, ist 2.

Diese Codierung hat damit eine Redundanz von:

$$r = l - h = 2 - 1{,}585 = 0{,}415 \text{ bit}$$

(i.d.R.: $r < 1$!?)

Codierungsstandards

Prof. Dr. Aris Christidis

- **ASCII** (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange):
  - 7 Bit / Zeichen;
  - erweiterte Version mit 8 Bit / Zeichen;
  - Breiteste Verwendung (Unix, MS-DOS, Programmiersprachen, ...)
- **ANSI**-Code (**A**merican **N**ational **S**tandards **I**nstitute):
  - 8 Bit (=1 Byte) / Zeichen;
  - Positionen 32-127 wie bei ASCII (Buchstaben, Ziffern, Satzzeichen);
  - Verwendung in Windows95 ff. *
- **Unicode:**
  - 16 Bit (=2 Byte) / Zeichen;
  - Erweiterung des ASCII-Codes: Buchstaben und Symbole aus allen bekannten geschriebenen Sprachen der Welt
  - Verwendung in Windows NT

Codierungsstandards

Prof. Dr. Aris Christidis

American Standard Code for Information Interchange

@	**NUL**	000	T	DC4	020	(	040	<	060	P	080	d	100	x	120
A	SOH	001	U	NAK	021	)	041	=	061	Q	081	e	101	y	121
B	STX	002	V	SYN	022	*	042	>	062	R	082	f	102	z	122
C	ETX	003	W	ETB	023	+	043	?	063	S	083	g	103	{	123
D	EOT	004	X	CAN	024	,	044	@	064	T	084	h	104	_|	124
E	ENQ	005	Y	EM	025	-	045	A	065	U	085	i	105	}	125
F	ACK	006	Z	SUB	026	.	046	B	066	V	086	j	106	~	126
G	BEL	007	[	ESC	027	/	047	C	067	W	087	k	107	DEL	127
H	BS	008	\	FS	028	0	048	D	068	X	088	l	108		
I	HT	009	]	GS	029	1	049	E	069	Y	089	m	109		
J	LF	010	^	RS	030	2	050	F	070	Z	090	n	110		
K	VT	011	_	**US**	031	3	051	G	071	[	091	o	111		
L	FF	012	SP		032	4	052	H	072	\	092	p	112		
M	CR	013	!		033	5	053	I	073	]	093	q	113		
N	SO	014	"		034	6	054	J	074	^	094	r	114		
O	SI	015	#		035	7	055	K	075	_	095	s	115		
P	DLE	016	$		036	8	056	L	076	`	096	t	116		
Q	DC1	017	%		037	9	057	M	077	a	097	u	117		
R	DC2	018	&		038	:	058	N	078	b	098	v	118		
S	DC3	019	'		039	;	059	O	079	c	099	w	119		

Codierungsstandards

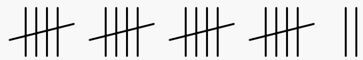
Unicode:

- Aktuelle Version 2.1
- 2 Byte/Zeichen: Buchstaben und Symbole aus allen bekannten geschriebenen Sprachen der Welt
  - Amerika, Europa, Mittlerer Osten, Afrika, Indien, Asien, Pazifik
  - Symbole
  - Satzzeichen
  - Sonderzeichen
- Unterstützt zur Zeit 38.887 Symbole/Zeichen (von 65536 möglichen)
- Genormt in ISO/IEC 10646-1:1993.

Codierungsstandards

- ASCII, ANSI und Unicode belegen übereinstimmend die Positionen 48 bis 57 mit den Ziffern '0' ... '9'.
- 48-57 sind hierbei die Kennziffern von Schriftzeichen.
- Der Wechsel von arithmetischen Schriftzeichen zu mathematischen Größen erfordert die Auseinandersetzung mit Zahlensystemen.

Zahlensysteme und ihre Anwendung

- Nicht systematische Zahlendarstellungen, z.B.:
  - Strichliste: 
  - römische Zahlen: I=1; V=5; X=10; L=50; C=100; D=500; M=1000

- Systematische Zahlendarstellungen Stellenwertsystemen:
  - Jede Zahl N läßt sich als Sequenz von Zeichen a_i darstellen
  - Die Anzahl der notwendigen unterscheidbaren Zeichen ist B
  - $N = \Sigma\, a_i \cdot B^i$

- Dezimalsystem:
  - B=10 unterscheidbare Zeichen: 0,1,2,3,4,5,6,7,8,9
  - $2001_{10} = 2 \cdot 10^3 + 0 \cdot 10^2 + 0 \cdot 10^1 + 1 \cdot 10^0$

Zahlensysteme und ihre Anwendung

Prof. Dr. Aris Christidis

- Dualsysten:
  - B=2 unterscheidbare Zeichen: 0,1
  - $2001_{10} = 1 * 2^{10} + 1 * 2^9 + 1 * 2^8 + 1 * 2^7 + 1 * 2^6 + 1 * 2^4 + 1 * 2^0$
    11111010001_2

<u>Anmerkungen:</u>

- **‚Dual‘**: Dyadisch, zur Basis 2 (vgl. dezimal, dekadisch)
- **‚Binär‘**: Zweiwertig (z.B.: Morse ohne Leerzeichen)

  Das Dualsystem ist das einzige breit verwendete Binärsystem, deshalb werden die Begriffe oft als Synonyme verwendet.

Zahlensysteme und ihre Anwendung

Prof. Dr. Aris Christidis

- Hexadezimalsystem (Sedezimalsystem)
  - Zeichen: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
  - $2001_{10} = 7*16^2 + 13*16^1 + 1*16^0$
    $= 7D1_{16}$ = 0x07D1 = H‘07D1
  - 4 Zeichen einer Binärzahl lassen sich durch eine Hexadezimalziffer darstellen (4 Binärziffern nennt man auch **NIBBLE**)

- Oktalsystem
  - Zeichen: 0,1,2,3,4,5,6,7
  - $2001_{10} = 3*8^3 + 7*8^2 + 2*8^1 + 1*8^0 = 3721_8$
  - 3 Zeichen einer Binärzahl lassen sich durch eine Oktalziffer darstellen

Zahlensysteme und ihre Anwendung

Prof. Dr. Aris Christidis

Übung zur Darstellung ganzer Zahlen:

Kreuzen Sie in der Tabelle alle Zahlensysteme an, die für die entsprechenden Zahlen zutreffen können.

Zahlenwert	Dual	Oktal	Dezimal	Hexadezimal
1992577			x	x
15034820			x	x
102436		x	x	x
120A0110				x
12305		x	x	x
1001101	x	x	x	x

Prof. Dr. Aris Christidis

Wechsel zwischen Zahlensystemen

Beispiel: Ergänzung einer Umwandlungstabelle.

Dezimal	Dual	Oktal
?	**100010**	?
68	?	?
?	?	**153**

Tabelle zeilenweise betrachtet:

- Dual → Dezimal: Berechnung v. Potenzen: $100010_2 = 2^1 + 2^5 = 34_{10}$
- Dual → Oktal: je 3 Stellen zusammenfassen: $100\ 010_2 = 42_8$ ¬

Zahlensysteme und ihre Anwendung

Prof. Dr. Aris Christidis

Beispiel (Forts.)

- Dezimal → Dual: fortlaufende Divisionen:

  68:2=34 R **0**
  34:2=17 R **0**
  17:2= 8 R **1**
  8:2= 4 R **0**
  4:2= 2 R **0**
  2:2= 1 R **0**
  1:2= 0 R **1**, d.h.: $68_{10} = 1000100_2$

- Dezimal → Oktal: fortlaufende Divisionen:

  68:8= 8 R **4**
  8:8= 1 R **0**
  1:8= 0 R **1** , d.h.: $68_{10} = 104_8$

- Dual → Oktal: je 3 Stellen zu einer zusammenfassen: $1\ 0\ 4_8$

- Oktal → Dual: zifferweise übertragen: $153_8 = 1\ 101\ 011_2$
- Oktal → Dezimal: $153_8 = 3 + 5*8 + 1*64 = 107_{10}$

Zahlensysteme und ihre Anwendung

Prof. Dr. Aris Christidis

Beispiel (Forts.)

Ergebnis:

Dezimal	Dual	Oktal
34	**100010**	42
68	1000100	104
107	1101011	**153**

(Hexadezimalsystem ganz entsprechend)

Zahlensysteme und ihre Anwendung

Prof. Dr. Aris Christidis

Technisch ebenfalls relevantes Zahlencodierungsverfahren:

Die BCD (Binary Coded Decimal) - Darstellung von Zahlen ist eine Mischform aus Dezimal- und Binär-Darstellung:

- Jede Ziffer der Dezimalzahl wird binär dargestellt.
- Die Darstellung jeder Ziffer erfolgt mit 4 Bit.
- Die Reihenfolge der Ziffern bleibt erhalten.

Beispiele:

- 7 — 0111
- 53 — 0101 0011
- 1234 — 0001 0010 0011 0100
- 2001 — 0010 0000 0000 0001

0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
Pseudotetraden	1010
	1011
	1100
	1101
	1110
	1111

Darstellung ganzer (positiver u. negativer) Zahlen: *

Darstellung des Vorzeichens im ersten Bit, z.B.

000 = 0	100 = -0
001 = 1	101 = -1
010 = 2	110 = -2
011 = 3	111 = -3

Nachteile:

- Redundanz in der Darstellung der 0
- Probleme beim formalen Addieren:

```
  110        - 2
+ 001        +1
--------     ----
  111        - 3
```

Zahlensysteme und ihre Anwendung

Prof. Dr. Aris Christidis

Zweierkomplementdarstellung (n Bit):

000 = 0	100 = -4
001 = 1	101 = -3
010 = 2	110 = -2
011 = 3	111 = -1

- Darstellung des Vorzeichens im ersten Bit
- Abdeckung der 2^n Zahlen $-2^{n-1}...+(2^{n-1}-1)$: keine Redundanz!
- Keine Probleme beim formalen Addieren:

```
  110        - 2
+ 001        +1
---------    ----
  111        - 1
```

- Betragshohe positive und negative Zahlen hintereinander:

  $(2^{n-1}-1)+1$ ergibt evtl. -2^{n-1}!

Prof. Dr. Aris Christidis

Grundsätzliches zum Komplement:

- Das (B-1)-Komplement einer Zahl im Zahlensystem mit der Basis B wird gebildet, indem jede Ziffer z_i dieser Zahl durch die Ziffer $B-1-z_i$ (*) ersetzt wird.
  - (*) Das ist die Differenz zur höchsten Ziffer des Zahlensystems.

- Das B-Komplement wird gebildet, indem zum (B-1)-Komplement die 1 addiert wird.

Beispiel:

- Das 9er-Komplement der Zahl 02_{10} ist 97_{10},
  das 10er-Komplement der Zahl 02_{10} ist 98_{10} .
- Das Einerkomplement der Zahl 0111_2 ist 1000_2,
  das Zweierkomplement der Zahl 0111_2 ist 1001_2 .

Zahlensysteme und ihre Anwendung

Prof. Dr. Aris Christidis

Der Darstellung negativer Zahlen durch das Komplement liegt die Vorstellung eines Zahlenrads zugrunde (vgl. km-Zähler):

(:)		(:)
0003	[3]	0011
0002	[2]	0010
0001	[1]	0001
0000	[0]	0000
9999	[-1]	1111
9998	[-2]	1110
9997	[-3]	1101
(:)		(:)

- Das 10er-Komplement von 0002 ist (9997+1=) 9998; gleichzeitig entspricht die Position '9998‘ einer ‘-2‘.
- Das Zweierkomplement von 0010 ist (1101+1=)1110; gleichzeitig entspricht die Position ‘1110‘ einer ‘-2‘.
- Negative Zahl durch bitweise Komplementierung u. Addition von 1 (Bsp: 3 = 0011, Zweierkomplement = 1100 + 1 = 1101 = -3)

Zahlensysteme und ihre Anwendung

Darstellung gebrochener Zahlen mit Vor- u. Nachkommateil

- Beispiele:

Gebrochene Dualzahl	Gebrochene Dezimalzahl
0,1	0,5
0,01	0,25
0,001	0,125
111,111	7,875
0,0001 1001 1001 1001	0,1

- Mit n Bit lassen sich 2^n verschiedene Zahlen darstellen.
  - Problem: Extrem große und extrem kleine Zahlen lassen sich nicht mit wenigen Bit darstellen.

    Bei 8 Bit mit 4 Vorkomma- und 4 Nachkommastellen (ohne Vorzeichen):

$$0000.0001 < x < 1111.1111$$

Auflösung des Zahlensystems

$$0,0625 < x < 15,9375$$

Zahlensysteme und ihre Anwendung

Prof. Dr. Aris Christidis

Darstellung von Gleitpunktzahlen **Z**:

$$Z = (-1)^V \cdot M \cdot B^E \text{ , mit:}$$

V: Vorzeichen-Bit (0: positiv, 1: negativ)
B: Basis des zugrundegelegten Zahlensystems (2, 10 oder 16)
M: Mantisse: $1/B \leq M \leq 1$ *
E: Exponent

z.B.: 32 Bit, davon 23 Mantisse (eines von über 20 Formaten)

V	E	Mantisse	Zahlenwert (Dezimal)
+	- 13	01001010110101101111111	+0,0001577567163622..
-	+44	10101011000000000000000	-29343216566272
+	+20	00101101011100010111000	1234711

- Übersetzungs- (dezimal in dual) und Rechenrundungen!

Zahlensysteme und ihre Anwendung

Prof. Dr. Aris Christidis

Addition von Dualzahlen

3 Rechenregeln:

```
0 + 0 = 0
1 + 0 = 1
1 + 1 = 0
```
Übertrag 1

Beispiel: `10+13=23`

```
      001010          10
    + 001101        + 13
    --------        ----
Ü      1
    --------        ----
      010111          23
```

- **Tip:** Binärdarstellung sicherheitshalber um **2** Stellen erweitern: Führende Nullen erleichtern die Erkennung von Vorzeichen und Übertrag bei algebraischen Summen!

Subtraktion als Addition mit dem Zweierkomplement:

- Bildung des **Zweierkomplements** des Subtrahenden (=1er-Kompl.+1 – bei **gleichlang**er Darstellung von Minuend u. Subtrahend, inkl. Vorzeichenstelle!),
- **Addition** und
- Abschneiden entstandener **Überträge** bei Addition von Zahlen ungleichen Vorzeichens (≡ Differenzbildung zwischen Zahlen gleichen Vorzeichens).

Beispiel: `13-10=3`

Aufgabe:

$$\begin{array}{r} 001101 \\ \underline{-001010} \end{array} \quad \Rightarrow$$

Zweierkomplement:

$$\begin{array}{lr} & 001101 \\ + & 110101 \\ + & 1 \\ \hline \text{Ü} & 1\ 111\ 1\ \\ \hline & (1)\,000011 \end{array} \quad \Rightarrow \quad 00011_2 = 3_{10}$$

Zahlensysteme und ihre Anwendung

Subtraktion mit negativem Ergebnis (hier: `10-13=-3`)

```
   001010   (2erK.)      001010
-  001101      ⇒       + 110010
                       +      1
                      Ü     1
                         111101 =-(000010+1)₂ =-3₁₀
```

Addition zweier negativer Zahlen (hier: `-10-13=-23`)

```
  -001010   (2erK.)      110101
              ⇒        +      1
-  001101              + 110010
                       +      1
                     Ü   11 111
                        1101001=-(0010110+1)₂=-23₁₀
```

Signale und Logik (3)

Prof. Dr. Aris Christidis

Zwischenbilanz der bisherigen Erkenntnisse:

- Energieformen (z.B. Elektrizität) können auch als **Signale** (=Informationsträger) genutzt werden (vgl. Telegraph).

  Die **Informationstheorie** liefert das theoretische Fundament zur effektiven **Codierung** (= Darstellung) von Information mit Hilfe von Signalen (z.B. Morse, ASCII).

- Vorrichtungen zur Energieübertragung und -verstärkung (z.B. Relais-Schaltungen) können auch für die **logische Verknüpfung** von Informationen eingesetzt werden (z.B. UND, ODER).

  Die Methoden dazu liefert die **Boolesche Algebra** (Schaltalgebra).

Signale und Logik (3)

Prof. Dr. Aris Christidis

Damit sind prinzipiell Wege zur Realisierung einer technischen DV auf der Grundlage der Signalverarbeitung gefunden.

Als Signal diente bisher d. Auftreten einer physikalischen (elektrischen) Größe (z.B. Zuschaltspannung).

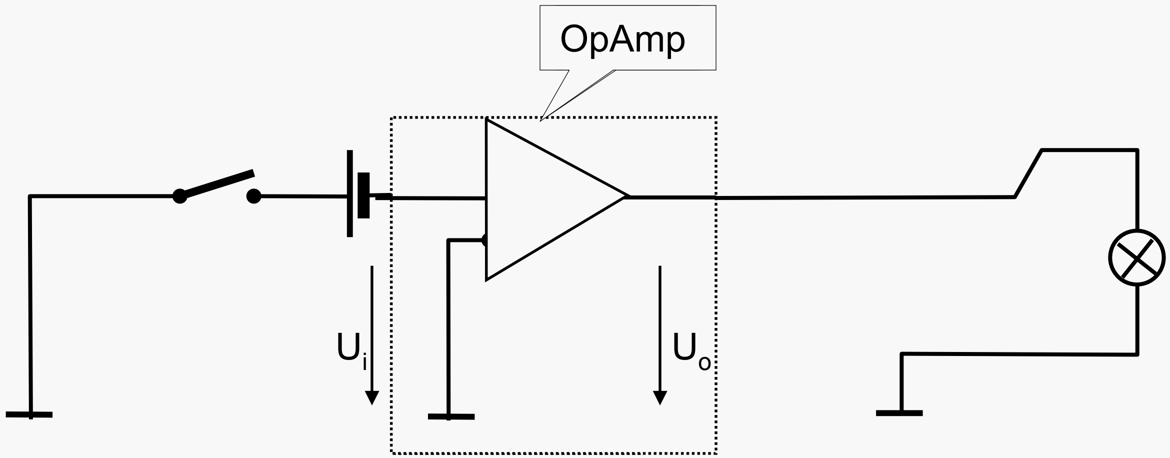
Die große Bedeutung von Rechenvorgängen in der („vor-elektronischen“) DV warf die Frage der Implementierung mathematischer Funktionen auf.

Rechnen mit Signalen

Prof. Dr. Aris Christidis

Ab Ende der 40er werden aus Röhren und Transistoren Operationsverstärker konzipiert und eingesetzt.

Ihr Verstärkungsfaktor liegt bei $U_o : U_i \geq 10^7$ ($U_o \approx 10V$). (*)



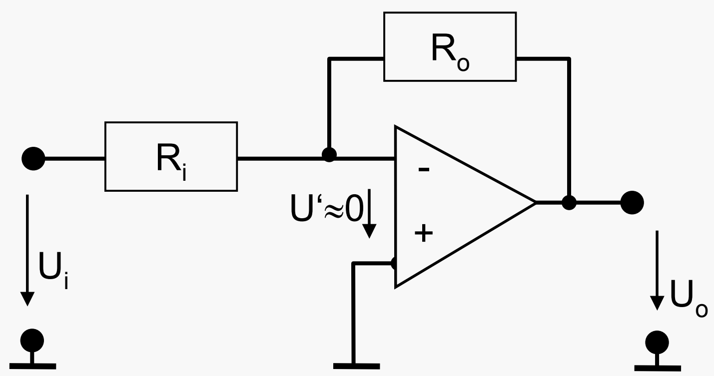
Diese Bauelemente ermöglichen die Simulation mathematischer Operationen.

(*) Das bedeutet Vollaussteuerung einer 10V-Einheit bei max. 100 nV=0,1 µV

Rechnen mit Signalen

Prof. Dr. Aris Christidis

Da der Operationsverstärker eine vernachlässigbar kleine Eingangsspannung und kaum Stromaufnahme hat, können auf einfache Art nützliche Rechenschaltungen entstehen - z.B.:

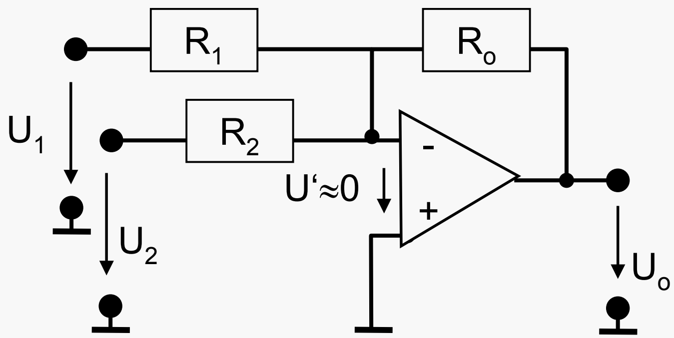


Einfache Verstärkerschaltung

$U_i / R_i \approx -U_o / R_o \approx 0 \quad \Rightarrow \quad \mathbf{U_o = k \cdot U_i} \qquad$ mit $k = -(R_o/R_i)$

Rechnen mit Signalen

Die Variation der Elemente solcher Schaltungen führte zur Implementierung komplexer mathematischer Operationen: - z.B.:



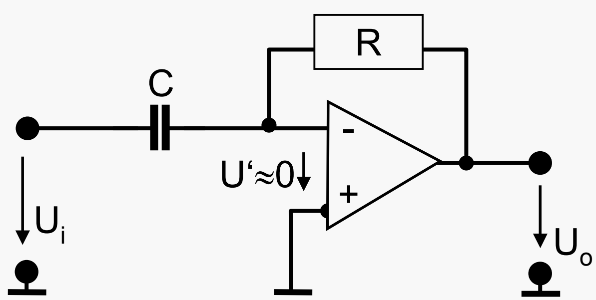
Summierverstärker

$$U_1 / R_1 + U_2 / R_2 + U_o / R_o \approx 0$$

$$-U_o = (R_o/R_1) \cdot U_1 + (R_o/R_2) \cdot U_2$$

$$\Rightarrow \mathbf{-U_o = m \cdot U_1 + n \cdot U_2,}$$

$$m=R_o/R_1 \text{ , } n=R_o/R_2$$



$$U_o = R \cdot C \cdot dU_i/dt$$

Differenzierer

Rechnen mit Signalen

Prof. Dr. Aris Christidis

Schaltungen wie die vorgestellten erzeugen eine **analoge** Ausgangsgröße, die von der <u>Stärke</u> des Eingangssignals abhängt; sie werden auch **analoge Schaltungen** genannt.

> **'Analog'** heißt eine veränderliche (mathematische, physikalische) Größe, wenn sich ihre Werte kontinuierlich verändern können und daher immer zwischen zwei zulässigen Werten auch einen Zwischenwert annehmen können.

Aus analogen Schaltungen entstanden die **Analogrechner**. Sie dienen heute fast ausschließlich zur Darstellung und Lösung mathematischer Funktionen und Gleichungen oder zur Simulation von Prozessen.

Rechnen mit Signalen

Digitale Schaltungen dienen der Erzeugung und Verarbeitung digitaler Signale.

> **'Digital'** heißt eine veränderliche (mathematische, physikalische) Größe, wenn sie abzählbar (meist endlich) viele, diskrete, ganzzahlige Werte annehmen kann.
>
> [digitus (lat.) = Finger; digit (engl.) = Ziffer]

> Sind die verwendeten Signale zudem binär, so sind die Ausgangsgrößen abhängig vom <u>Auftreten</u> bzw. von der <u>Folge</u> von Signalen (u. nicht von deren Intensität).

Einfache digitale Schaltungen lassen sich direkt anhand der Wahrheitstabellen der gewünschten Funktionen herleiten. ¬

Rechnen mit Signalen

Beispiel:

Schaltung zur Addition zweier einstelliger Dualzahlen x, y

$\overline{x}$	$\overline{x}\wedge y$	x	y	Σ	$Ü$	$\overline{y}$	$x\wedge\overline{y}$
1	0	0	0	0	0	1	0
1	1	0	1	1	0	0	0
0	0	1	0	1	0	1	1
0	0	1	1	0	1	0	0

$$\Sigma = (\overline{x}\wedge y) \vee (x\wedge\overline{y})$$ *

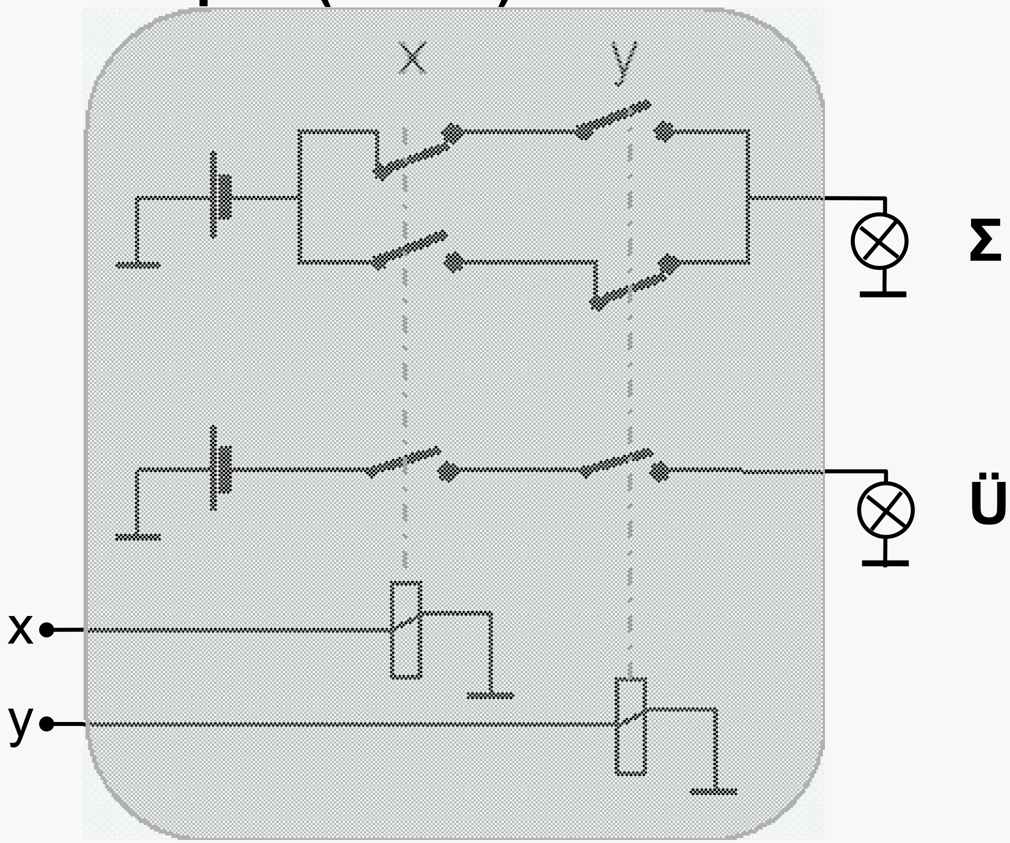
$$Ü = x \wedge y$$

¬

Rechnen mit Signalen

Prof. Dr. Aris Christidis

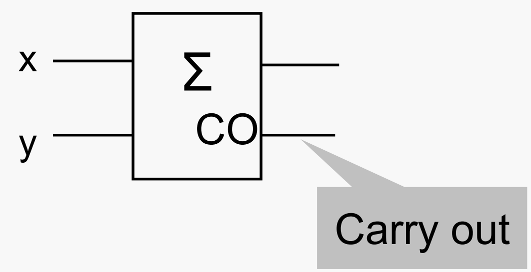
Beispiel (Forts.): Addition



$$\Sigma = (\overline{x} \wedge y) \vee (x \wedge \overline{y})$$

$$Ü = x \wedge y$$

Halbaddierer

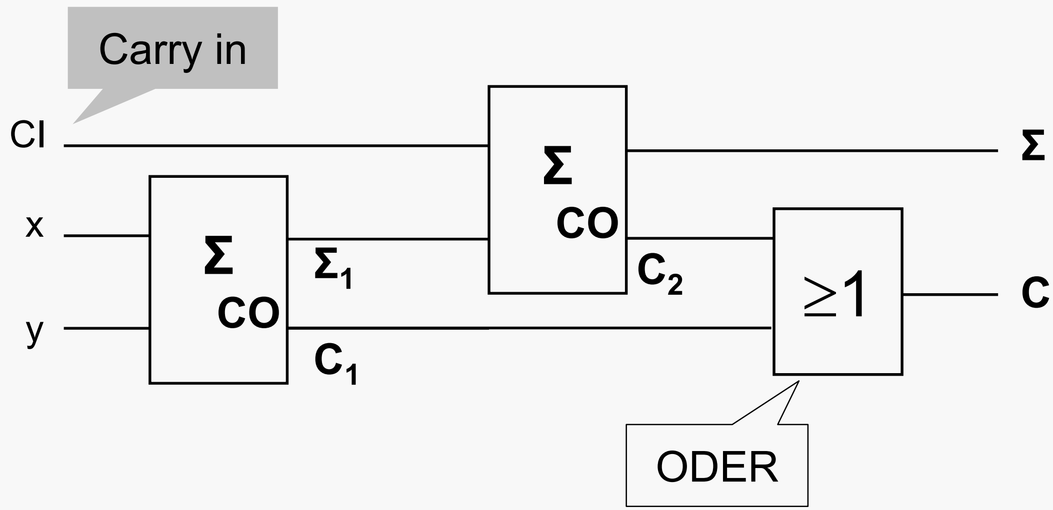


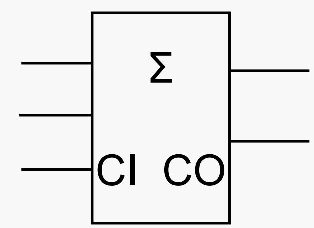
Mit einem Halbaddierer können zwei einstellige Dualzahlen addiert werden.

Rechnen mit Signalen

Prof. Dr. Aris Christidis

Beispiel (Forts.): Addition - Berücksichtigung des Übertrags





1-Bit-Volladdierer

Es kann jeweils nur

- entweder Σ_1=1 oder C_1=1 sein
- entweder C_1=1 oder C_2=1 sein

denn: $C_1=1 \Rightarrow \Sigma_1=0 \Rightarrow C_2=0$,

$C_2=1 \Rightarrow \Sigma_1=1 \Rightarrow C_1=0$

x	y	Σ_1	C_1	CI_{max}	Σ_{max}	C_2	C_1+C_2	$C_1 \vee C_2$
0	0	0	0	1	1	0	0	0
0	1	1	0	1	0	1	1	1
1	0	1	0	1	0	1	1	1
1	1	0	1	1	1	0	1	1

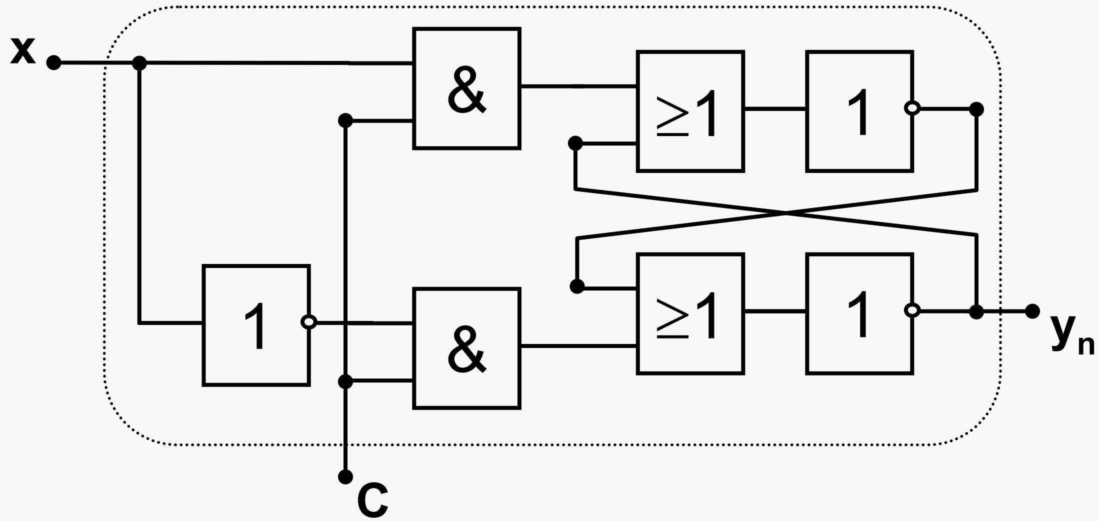
Rechnen mit Signalen

- Zwei Halbaddierer können über ein ODER-Gatter zu einem 1-Bit-Volladdierer für die Addition drei einstelliger Dualzahlen verbunden werden.
- Mit *n* 1-Bit-Volladdierern können *n*-stellige Dualzahlen addiert werden. Dabei ist auf die zeitliche Koordination anfallender Teilergebnisse zu achten.
- Für die Aufnahme von Zwischenergebnissen werden bistabile Kippschaltungen, sog. **Flipflops** eingesetzt.

Rechnen mit Signalen

Prof. Dr. Aris Christidis

Beispiel: Das D-Flipflop (Datenflipflop)



x	C	y_n
0	0	y_{n-1}
0	1	0
1	0	y_{n-1}
1	1	1

- Für C=1 übernimmt y_n den Wert von x
- Für C=0 behält y_n den letzten Wert y_{n-1}
- Der Zeitpunkt der Wert-Zuweisung C=1 bestimmt die Taktung von y_n

Rechnen mit Signalen

Prof. Dr. Aris Christidis

- Flipflops sind schnelle 1-Bit-Speicher; sie eignen sich zur **Speicherung** und **Weiterleitung**, aber auch zur zeitlichen **Koordination** von Zwischenergebnissen.
- Zur Aufnahme von *n*-Bit-Wörtern werden Sätze von *n* Flipflops zu *n*-stelligen **Registern** zusammengefaßt.
- Durch schaltbare Verbindung logischer Schaltungen entstehen **programmierbare** Schaltwerke.
- Schaltungen für **arithmetische** u. **logische** Operationen sowie mehrere **Register** bilden das **Rechenwerk** eines Computers, einen wichtigen Bestandteil der **Hardware** jedes programmierbaren Digitalrechners.

  Sie sind als integrierte Schaltkreise (integrated circuits, ICs, „Chips“) im Mikroprozessor (µP) enthalten.

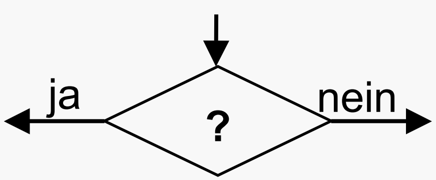
Darstellung von Rechenwegen

Grafische Darstellug von Lösungswegen -allg.: von Abläufen- mit Hilfe sog. (Daten-) **Flußdiagramme** (Flow charts, auch: Ablaufdiagramme o. -pläne). Zeichen nach DIN 66001:

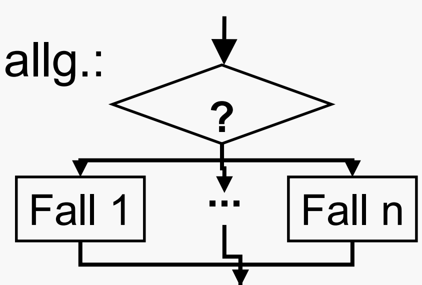
Schnittstelle (Beginn, Ende, Halt)

Aktion, Operation

Ablaufrichtung

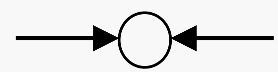


Verzweigung, Abfrage



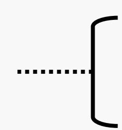
Fallabfrage und -behandlung

Ein- / Ausgabe



Zusammenführung

Hilfsalgorithmus, Unterprogramm

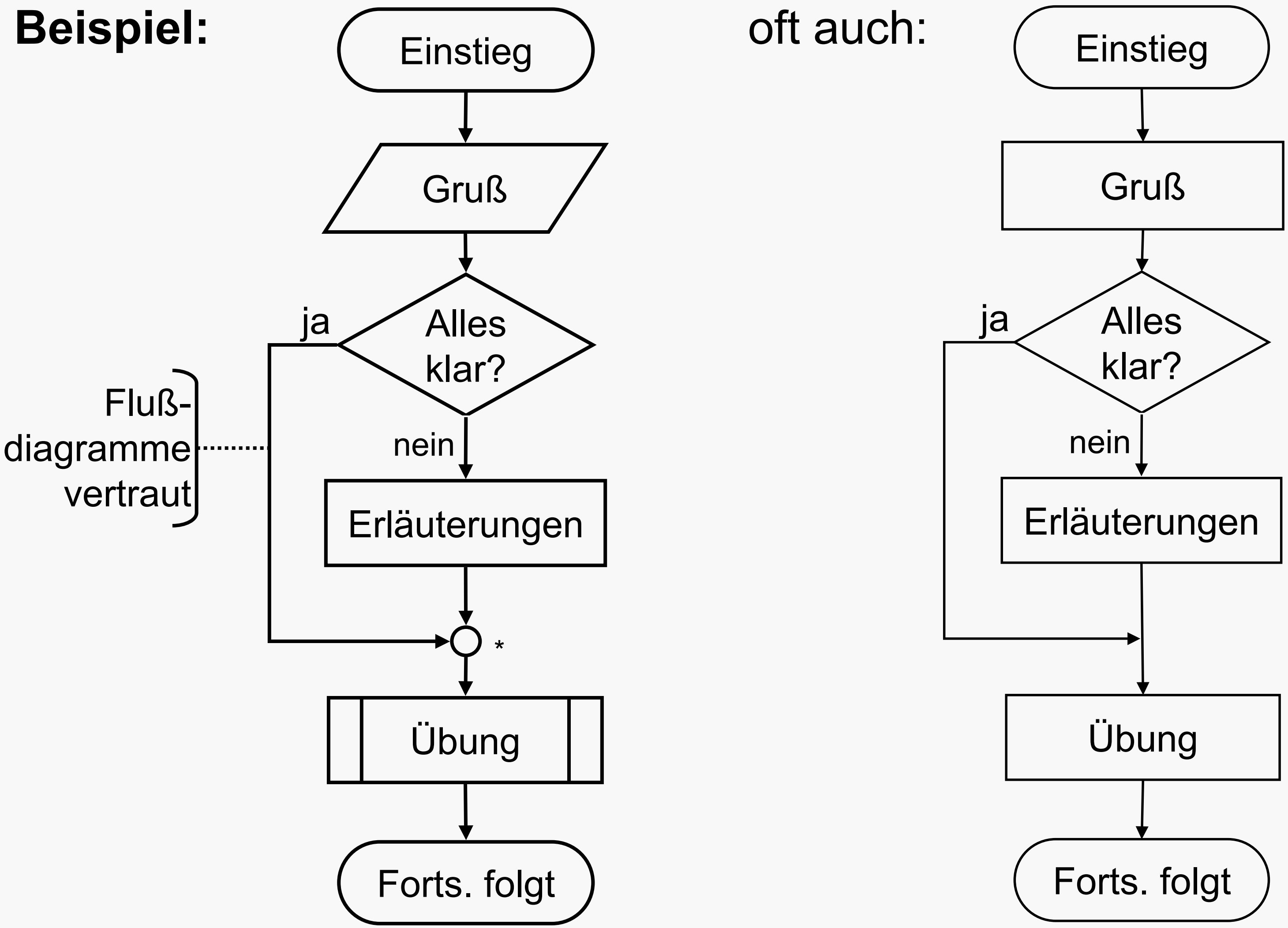


Kommentar

Darstellung von Rechenwegen

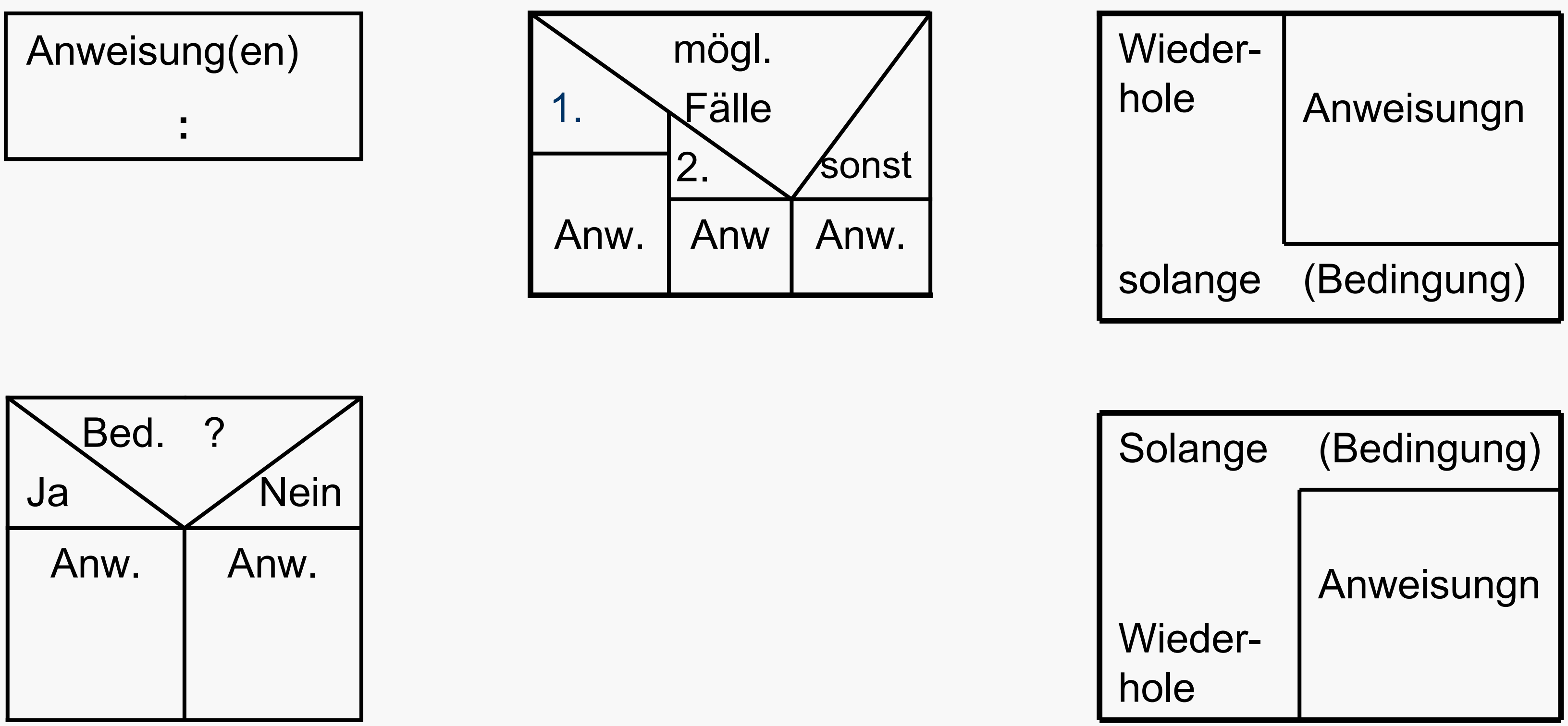
Prof. Dr. Aris Christidis

Beispiel: oft auch:



Darstellung von Rechenwegen

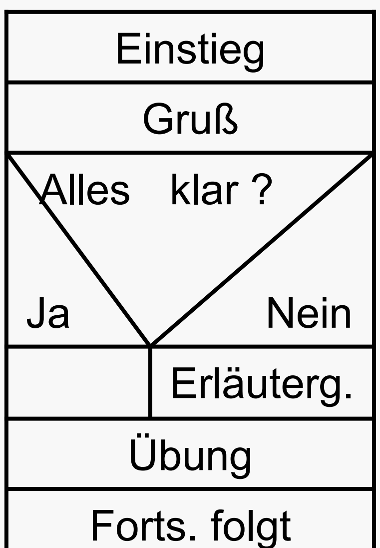
Neben dem Flußdiagramm wird das 1973 von Isaac Nassi u. Ben Shneidermann vorgeschlagene **Struktogramm** o. **Nassi-Shneidermann-Diagramm** verwendet (DIN 66261):



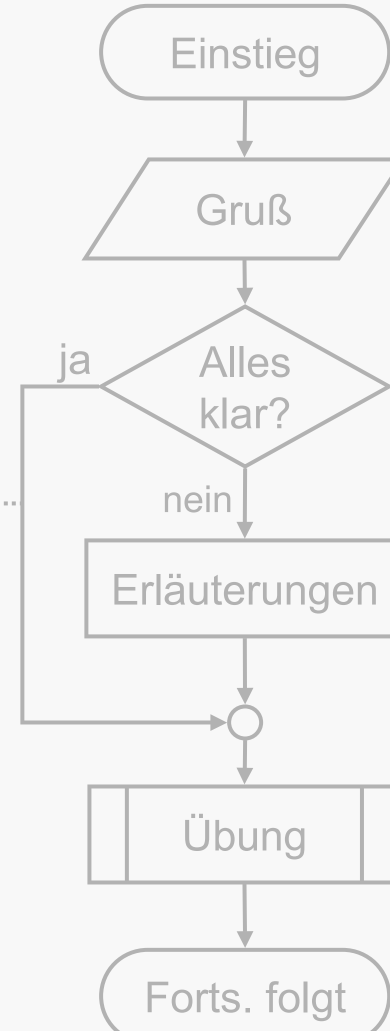
Darstellung von Rechenwegen

Struktogramme sind platz-sparender, oft anschaulicher und können z.T. von Programmier-umgebungen verarbeitet werden.

Beispiel:



Fluß-diagramme vertraut



Darstellung von Rechenwegen

Prof. Dr. Aris Christidis

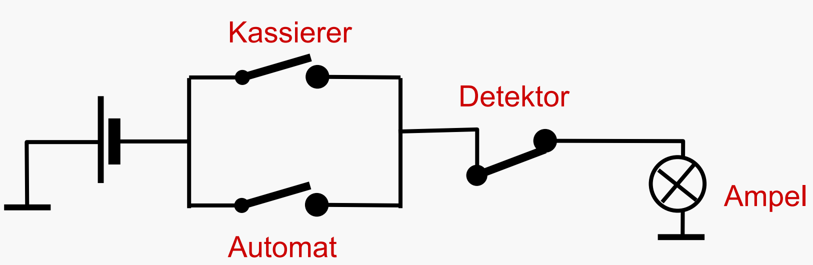
Anwendung auf früheres Beispiel:

Grüne Ampel zur Autowaschanlage soll aufleuchten, wenn

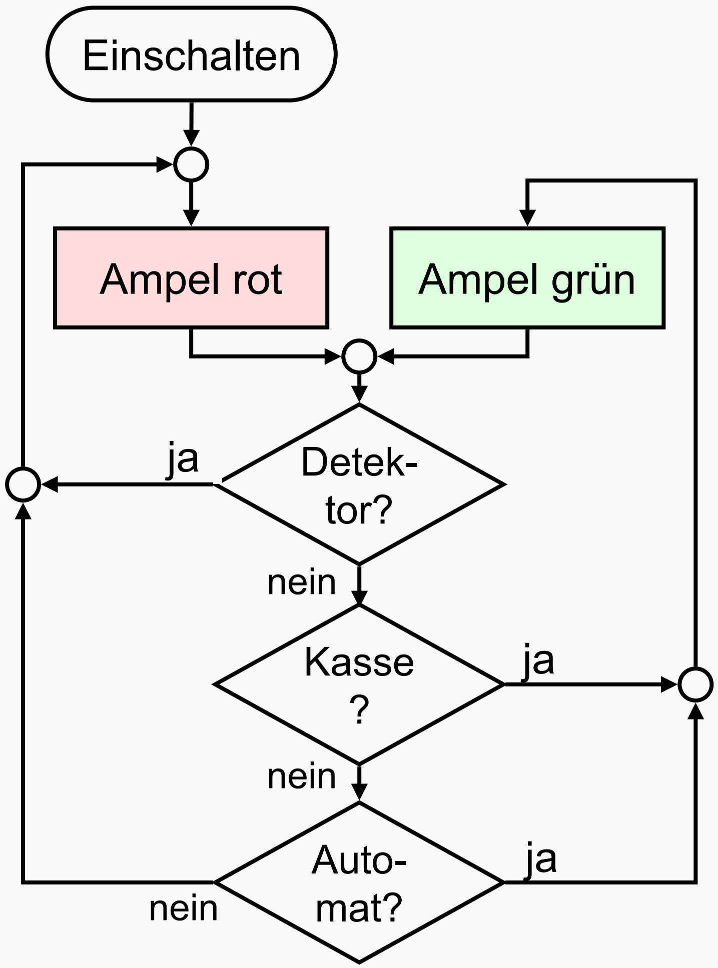
- Kassierer Zahlung bestätigt
  ODER
  Zahlungsautomat Zahlung meldet

UND

- Detektor kein Waschwasser meldet.



Flußdiagramm:



Darstellung von Rechenwegen

Dasselbe Beispiel als Struktogramm:

Grüne Ampel zur Autowasch-
anlage soll aufleuchten, wenn

- Kassierer Zahlung bestätigt

  ODER

  Zahlungsautomat

  Zahlung meldet

UND

- Detektor

  kein Waschwasser meldet