Signale und Logik (3)

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)

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

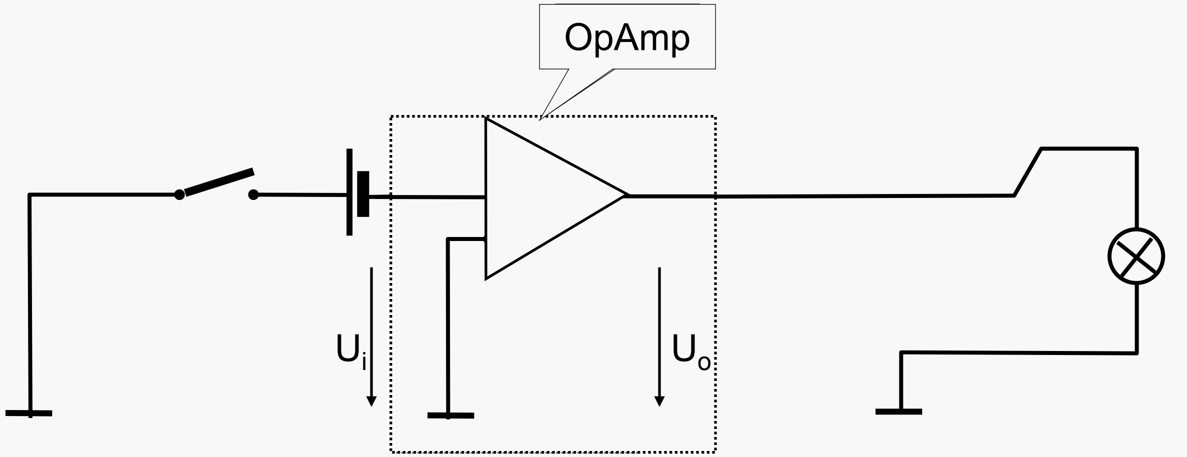
Als Signal diente bisher d. Auftreten einer physikalischen (elektrischen) Größe (z.B. Zuschaltspannung).

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

Rechnen mit Signalen

Ab Ende der 1940er werden nacheinander Röhren/ Transistoren/ Operationsverstärker konzipiert und eingesetzt.

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

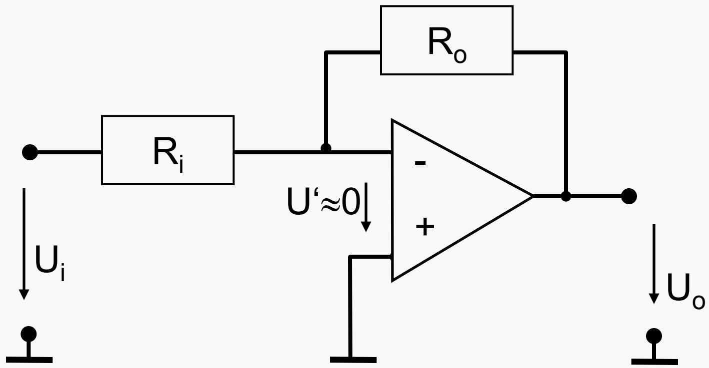


Diese Bauelemente ermöglichen die Simulation mathematischer Operationen.

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

Rechnen mit Signalen

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

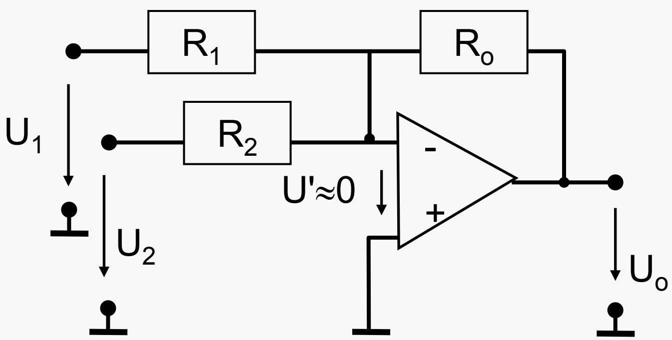


Einfache Verstärkerschaltung

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

Rechnen mit Signalen

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



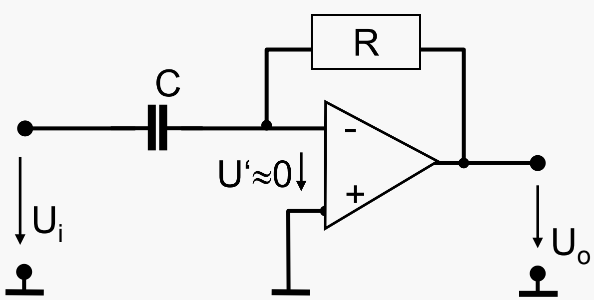
Summierverstärker

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

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

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

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



$$\mathbf{-U_o = R \cdot C \cdot dU_i/dt}$$

Differenzierer

Rechnen mit Signalen

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 Auftreten bzw. von der Folge von Signalen (und nicht von deren Intensität).

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

Beispiel:

Schaltung zur Addition zweier einstelliger Dualzahlen x, y

$\overline{x}$	$\overline{x} \wedge y$	x	y	Σ	$\ddot{U}$	$\overline{y}$	$x \wedge \overline{y}$
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

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

$$\ddot{U} = x \wedge y$$

<u>3 Rechenregeln:</u>

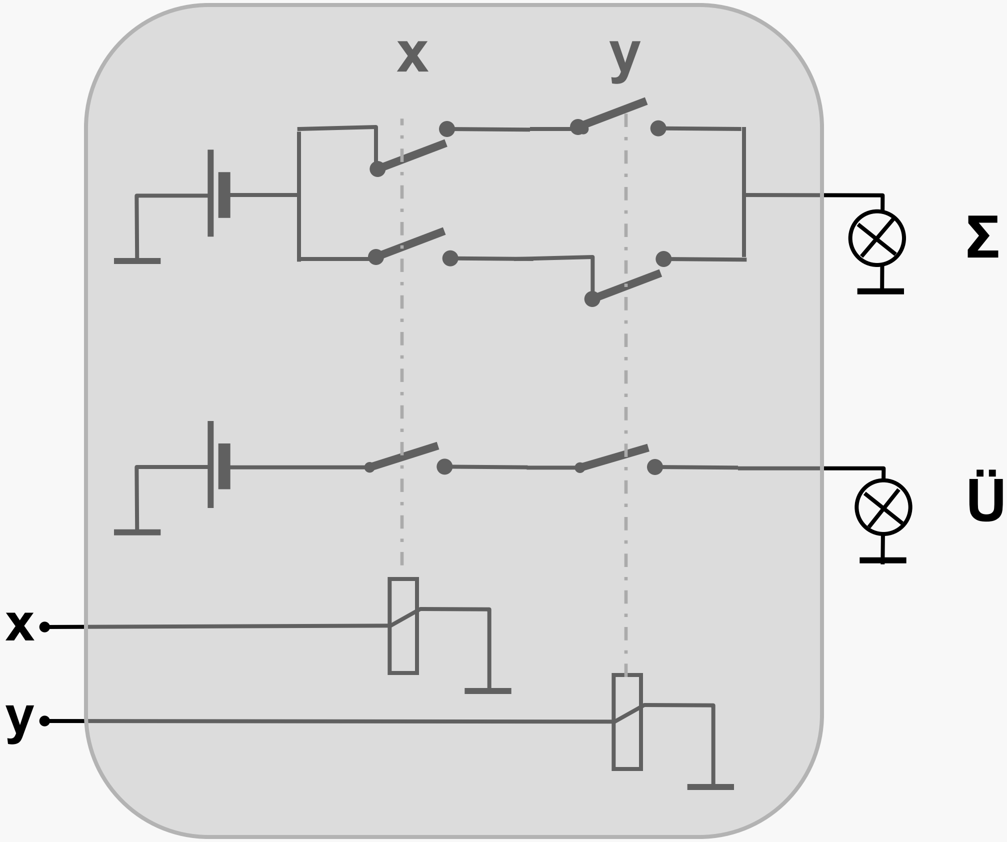
```
0 + 0 = 0
1 + 0 = 1
1 + 1 = 0
```

ODER			UND		
x_1	x_2	y	x_1	x_2	y
0	0	0	0	0	0
0	1	1	0	1	0
1	0	1	1	0	0
1	1	1	1	1	1

Rechnen mit Signalen

Prof. Dr. A. Christidis • SS 2015

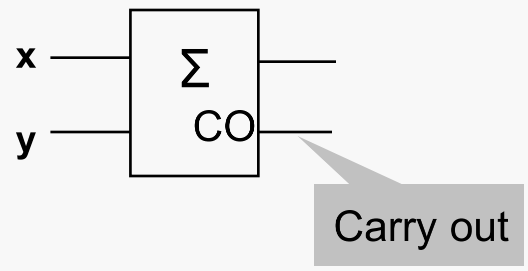
Beispiel (Forts.): Addition



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

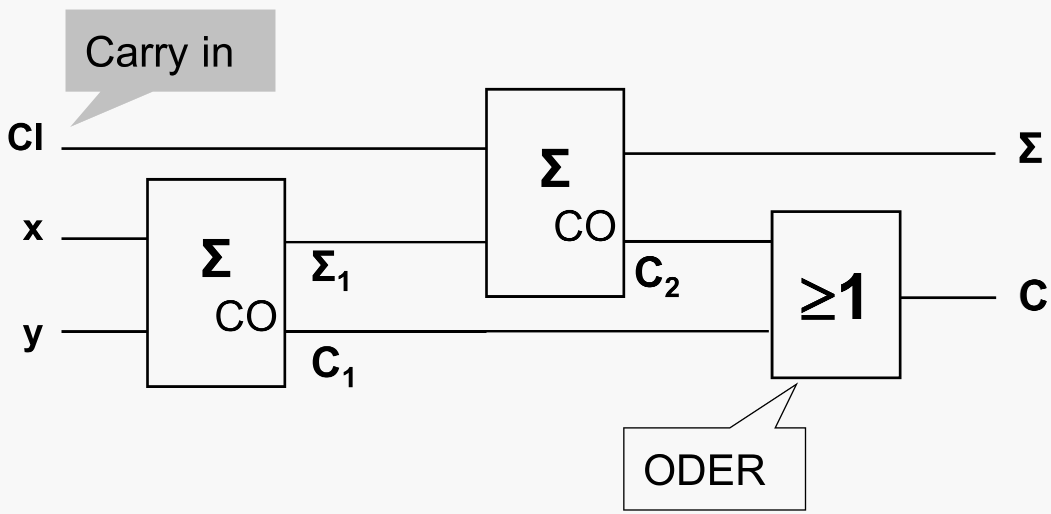
$$Ü = x \wedge y$$

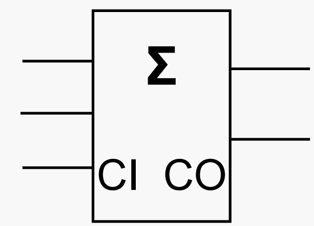
Halbaddierer



Mit einem Halbaddierer können zwei einstellige Dualzahlen addiert werden.

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

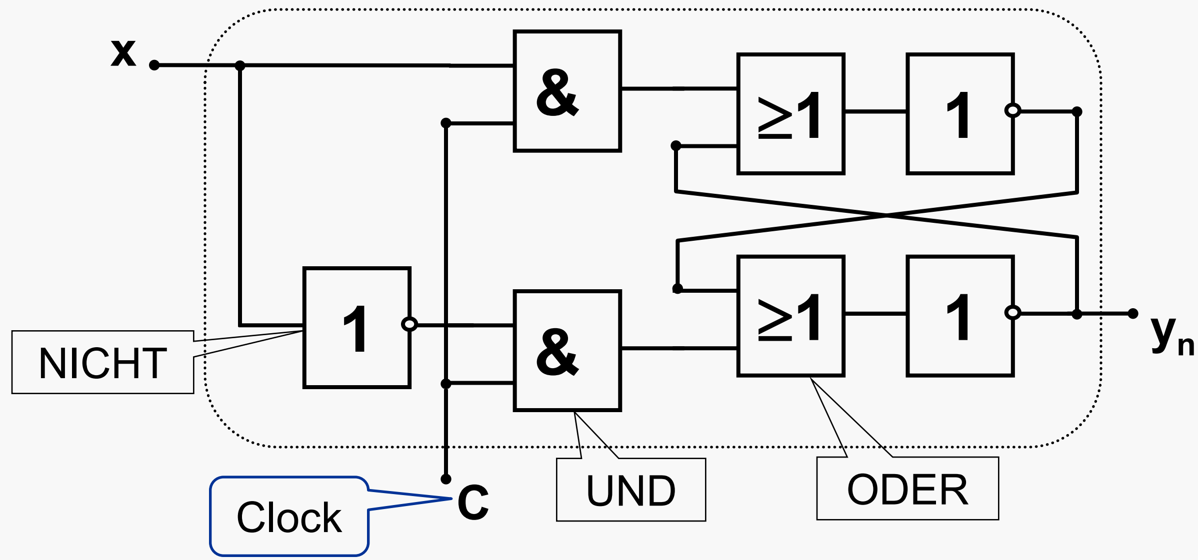
- Zwei Halbaddierer können über ein ODER-Gatter zu einem 1-Bit-Volladdierer für die Addition drei einstelliger Dualzahlen verbunden werden.

- 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

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

- 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.