

4 Informationsverarbeitung in IT-Systemen

4.1 Begriffe der Informationstechnik

4.1.1 Zeichen und Daten

Informationen sind im Sinne der Umgangssprache die Kenntnisse und das Wissen über Sachverhalte, Vorgänge, Zustände, Ereignisse usw. Sie können durch gesprochene und geschriebene Wörter, durch Tabellen und Diagramme oder Grafiken und Bilder dargestellt, gespeichert und verbreitet werden. In der Informations- und Kommunikationstechnik werden Informationen durch **Zeichen** dargestellt.

Ein **Zeichen** ist ein Element aus einer Menge verschiedener Elemente. Die Menge der Elemente wird als **Zeichenvorrat** bezeichnet.

Beispiele für Zeichen sind die Buchstaben des Alphabets, Ziffern, Interpunktionszeichen, Steuerzeichen (Wagenrücklauf, Zeilenvorschub auf der Tastatur usw.).

In der Kommunikationstechnik dient eine Zeichenfolge zur Übertragung einer Information und wird **Nachricht** genannt. In der Informationstechnik werden Zeichenfolgen, die eine Information zum Zweck der Verarbeitung enthalten, als **Daten** bezeichnet.

4

4.1.2 Signalarten

Nachrichten und Daten müssen zur Übertragung oder Verarbeitung in **Signale** umgesetzt werden.

■ **Signale** dienen zur Darstellung von Nachrichten und Daten durch physikalische Größen wie z. B. Spannung, Stromstärke o. Ä.

Zur Verdeutlichung eines Signalverlaufes wird üblicherweise in einem Diagramm der Signalwert in Abhängigkeit von der Zeit dargestellt. Signale können sowohl hinsichtlich des Wertebereiches als auch hinsichtlich des Zeitbereiches **kontinuierlich** (stetig, lückenlos zusammenhängend) oder **diskret** (durch endliche Abstände voneinander getrennt) sein (Bild 4.1).

1. Ein **wert- und zeitkontinuierliches Signal** kann jeden beliebigen Signalwert annehmen; in jedem Zeitpunkt ist ein Signalwert vorhanden.
2. Ein **wertdiskretes zeitkontinuierliches Signal** kann nur bestimmte Werte zwischen einem negativen und einem positiven Höchstwert annehmen; in jedem Zeitpunkt ist ein Signalwert vorhanden.

3. Ein **wertkontinuierliches zeitdiskretes Signal** kann jeden beliebigen Signalwert annehmen, ist aber nur zu bestimmten Zeiten vorhanden.

4. Ein **wert- und zeitdiskretes Signal** kann nur bestimmte Werte annehmen und ist nur zu bestimmten Zeiten vorhanden.

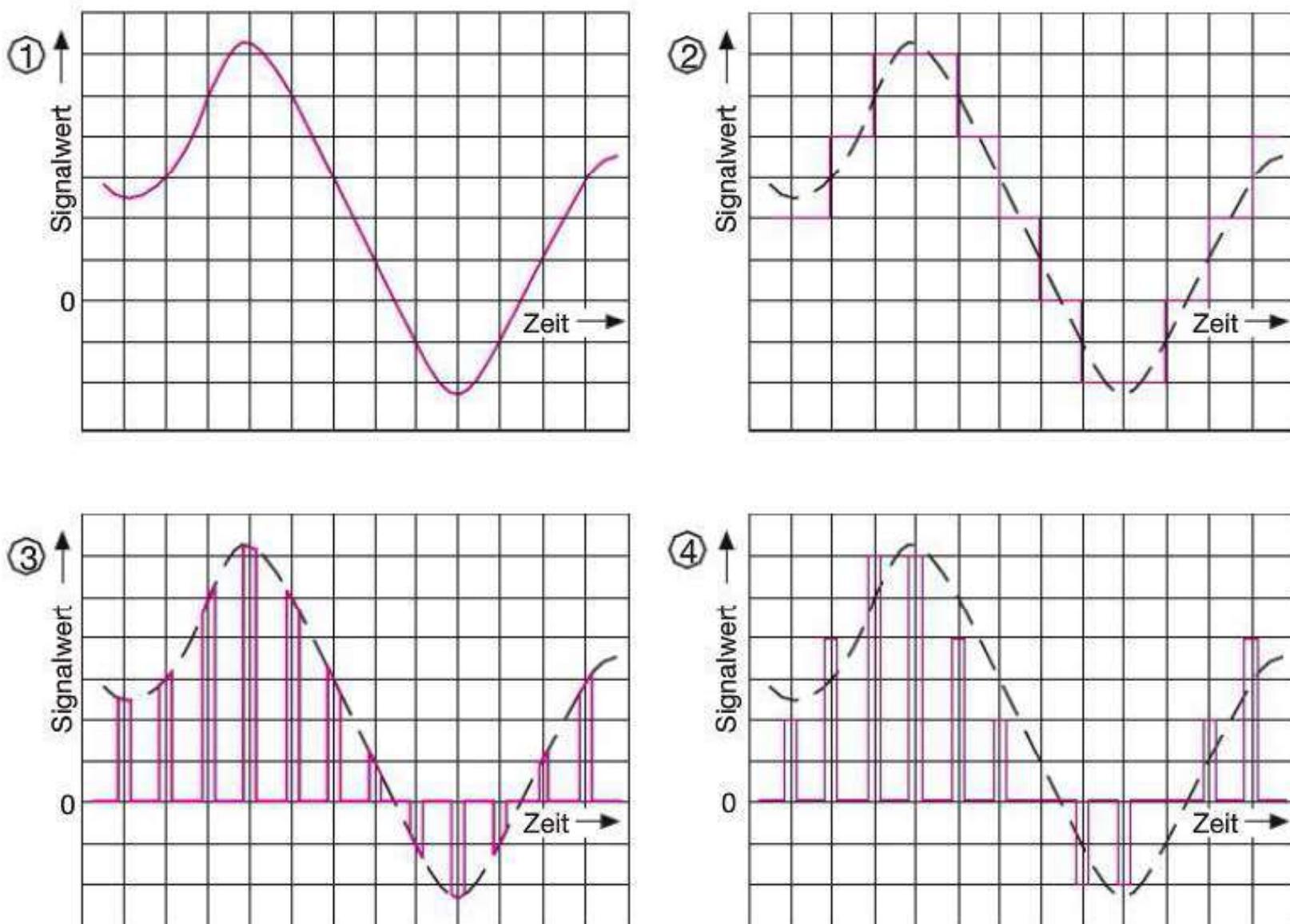


Bild 4.1: Signalarten

In der IT-Technik werden sowohl analoge Signale als auch digitale Signale verarbeitet und übertragen.

Ein Beispiel für ein **analoges Signal** ist die sogenannte Sprechwechselspannung, die in einem Mikrofon durch Umwandlung der auf die Membran auftreffenden Schallwellen erzeugt wird (Bild 4.2).

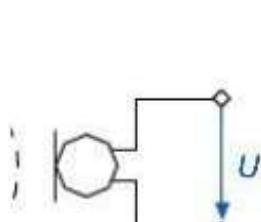


Bild 4.2: Analoges Signal

Bei einem **digitalen Signal** können innerhalb eines begrenzten Wertebereiches nur bestimmte (diskrete) Signalwerte auftreten. Jedem Signalwert kann ein Zeichen zugeordnet werden. So können z.B. die Zeichen von 0 bis 5 jeweils durch einen festen Signalwert dargestellt werden (Bild 4.3).

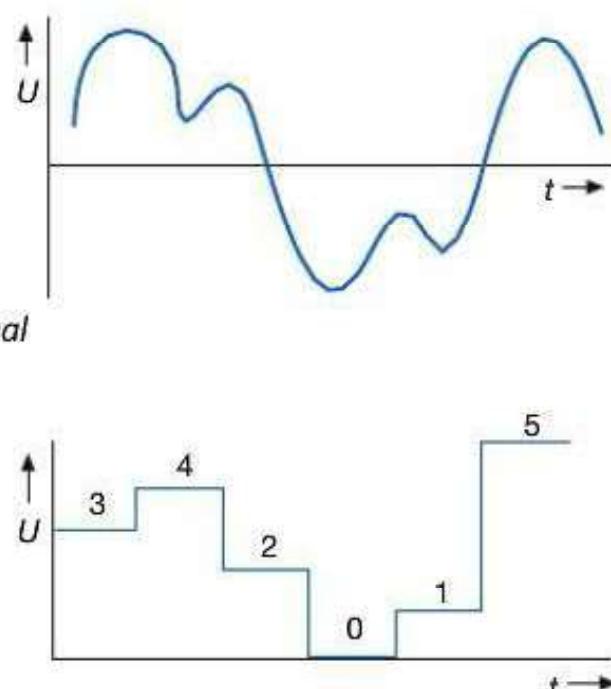


Bild 4.3: Digitales Signal

In IT-Systemen werden – bedingt durch die zwei Schaltzustände elektromechanischer und elektronischer Schaltelemente – fast ausschließlich Digitalsignale verarbeitet, die nur zwei verschiedene Signalwerte annehmen können; man bezeichnet sie als **binäre (zweiwertige) Signale** (Bild 4.4).

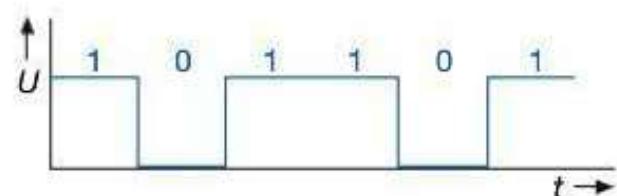


Bild 4.4: Binäres Signal

Als Binärzeichen werden den beiden Signalzuständen die Ziffern 0 und 1 zugeordnet. In der binären Schaltungstechnik verwendet man oft die Buchstaben L (Low Level) und H (High Level), wobei die Wertebereiche für L und H durch die Technologie der Schaltungen bestimmt werden (Bild 4.5).

In einer elektronischen Binärschaltung eines bestimmten Typs werden den Pegeln die folgenden Spannungsbereiche zugeordnet:

- | | |
|---------------|------------------|
| 2,4 V bis 5 V | gilt als H-Pegel |
| 0 V bis 0,4 V | gilt als L-Pegel |

Für die Arbeitssicherheit von Digitalschaltungen ist die Größe des Spannungsabstandes zwischen H-Pegel und L-Pegel wichtig.

Für das Beispiel gilt:

$$\text{Abstand} = 2,4 \text{ V} - 0,4 \text{ V} = 2 \text{ V}$$



Bild 4.5:
Wertebereiche
der Signalpegel
in einer
Binärschaltung

4.1.3 Signalübertragung

4

Den in Bild 4.6 dargestellten binären Zustände 0 und 1 (alternative Schreibweise: log 0 und log 1) werden zur *Signalverarbeitung* im PC vielfach die Spannungswerte 0 V und 5 V zugeordnet. Diese Spannungswerte können auch für die *Signalübertragung* über eine elektrische Leitung verwendet werden. Dies nennt man eine unsymmetrische Übertragung.

Bei einer **unsymmetrischen Übertragung** eines Binärsignals liegt auf dem Hinleiter entsprechend dem logischen Zustand (0 oder 1) die jeweils zugeordnete Spannung (0 V oder 5 V); der Rückleiter liegt hierbei stets fest auf 0 V.

Das Potenzial 0 V wird auch als Bezugspotenzial oder Massepotenzial bezeichnet (Kap. 5.1.1.4).

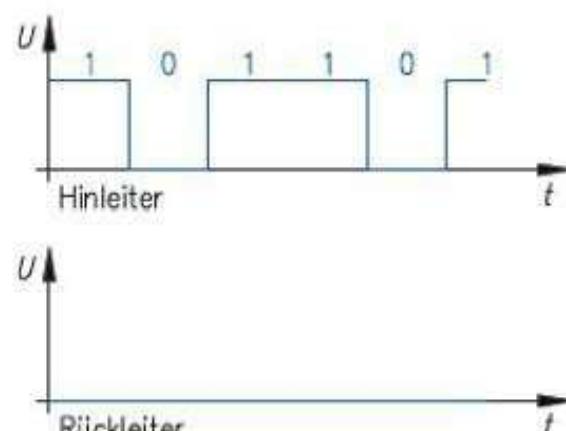
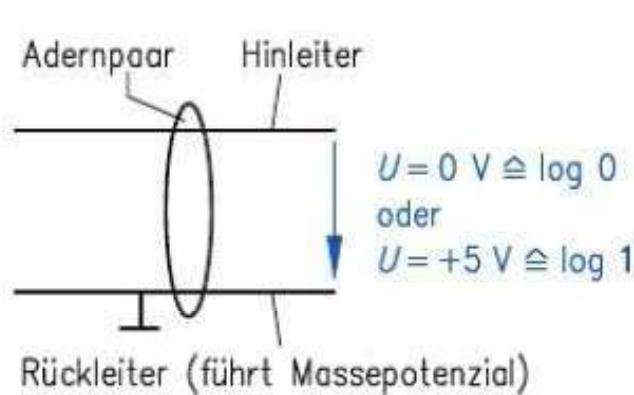


Bild 4.6: Unsymmetrische Übertragung (Grundprinzip)

Diese Art der Signalübertragung ist zwar einfach zu realisieren, hat aber den Nachteil, dass man auf der Empfangsseite den Binärzustand 0 (also 0 V) nicht von einer Leitungsunterbrechung unterscheiden kann.

Aus diesem Grunde werden den binären Zuständen 0 und 1 zur Signalübertragung oftmals andere Spannungswerte zugeordnet. Von besonderer Bedeutung ist hierbei die Übertragung mit differenziellen Signalen.

Als **differenzielles Signal** bezeichnet man die Spannungszuordnung zu einem Binärsignal, bei der sich entsprechend dem logischen Zustand (0 oder 1) sowohl auf dem Hinleiter als auch auf dem Rückleiter der zugehörige Spannungswert ändert.

Die Datenübertragung mithilfe eines differenziellen Signals, bei dem sich der Spannungswert auf den Hin- und den Rückleiter *symmetrisch* zum Nullpotenzial ändert, bezeichnet man als **symmetrische Übertragung**.

Beispiel:

Logisch 0: auf dem Hinleiter -5 V und auf dem Rückleiter $+5\text{ V}$

Logisch 1: auf dem Hinleiter $+5\text{ V}$ und auf dem Rückleiter -5 V

Sowohl bei der Übertragung von logisch 0 als auch bei logisch 1 führt jede Leitung stets ein von 0 V verschiedenes Potenzial. Somit ist eine Leitungsunterbrechung eindeutig erkennbar. Die Bezeichnung *symmetrische Übertragung* resultiert aus der symmetrischen Lage der beiden Leiterpotenziale bezogen auf das Massepotenzial (OV). Der Potenzialunterschied zwischen beiden Leitern wird auch als **Spannungshub** bezeichnet und ist im dargestellten Beispiel mit $\pm 10\text{ V}$ doppelt so groß wie bei der unsymmetrischen Übertragung.

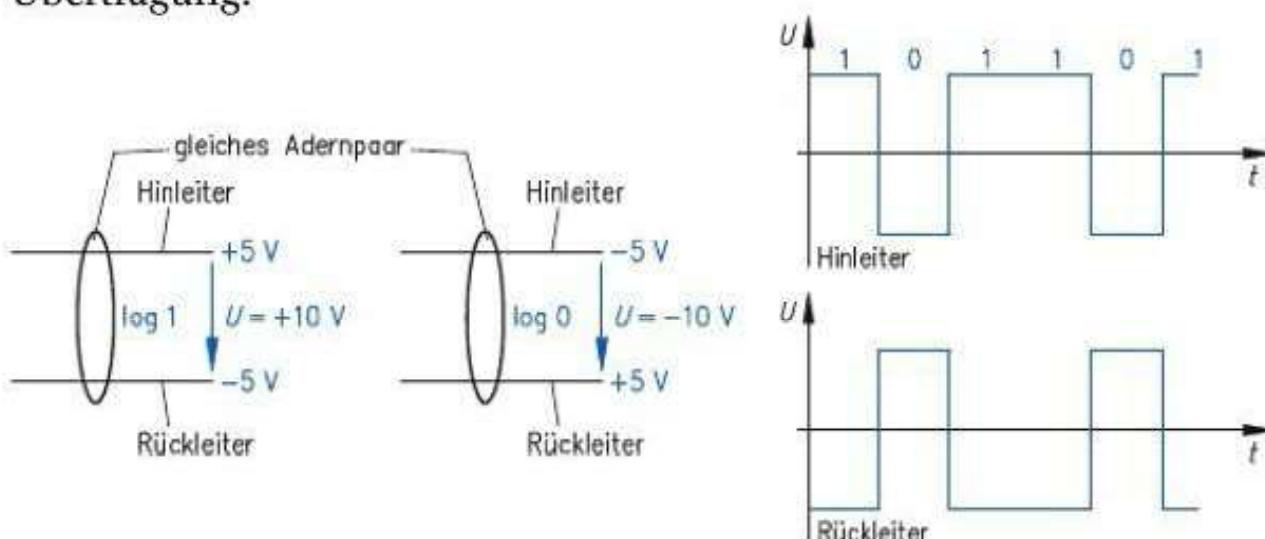


Bild 4.7: Symmetrische Übertragung (Grundprinzip)

Da die symmetrische Übertragung wesentlich weniger störanfällig als die unsymmetrische Übertragung ist, verwendet man in der Praxis auch wesentlich geringere Spannungen für die Datenübertragung (z.B. $\pm 250\text{ mV}$ bei SATA, Kap. 1.7.1).

Ein Verfahren zur **differenziellen Datenübertragung** mit geringem Spannungshub über zwei Leitungen bezeichnet man mit der Abkürzung **LVDS** (Low Voltage Differential Signaling).

Die fortschreitende Miniaturisierung in der Halbleitertechnik führt zur Herabsetzung der Versorgungsspannungen und folglich zur Verkleinerung des Spannungshubs der übertragenen Signale. LVDS arbeitet mit einem Spannungshub von $0,3\text{ V}$ auf beiden Leitungen. Eine „0“ wird dargestellt, indem Leitung A („Hinleiter“) auf $1,1\text{ V}$ und Leitung B („Rückleiter“) auf $1,4\text{ V}$ liegt; eine „1“ wird durch die umgekehrten Spannungswerte angezeigt (Bild 4.8). Die

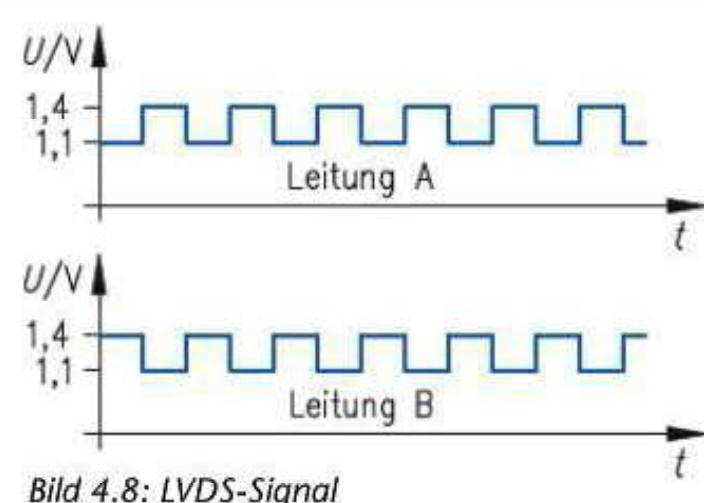


Bild 4.8: LVDS-Signal

Signalpegel auf beiden Leitungen haben also immer entgegengesetzte Werte, sind dabei aber dauernd positiv.

Die erreichbare Datenrate bei LVDS liegt beim derzeitigen Stand der Technik (2012) bei mehreren Gbit/s. Mit Cat-5-Kabel-UTP ist bei einer Leitungslänge von ca. 2 m eine Datenrate von 200 Mbit/s typisch.

LVDS wurde als Übertragungsstandard (ANSI/TIA/EIA-644-1995) zur Hochgeschwindigkeits-Datenübertragung für Platinen und Backplanes entwickelt. Er wird heute zur Übertragung von Daten mit hoher Störsicherheit über Entfernungen von einigen Metern eingesetzt z.B. bei USB, Firewire u. Ä. Ein wichtiges Anwendungsgebiet ist die digitale Ansteuerung von Flachbildschirmen. In der Kfz-Elektronik wird LVDS zur Vernetzung von Kameras, Displays und den jeweiligen Steuer- und Auswertegeräten im Fahrzeug verwendet; neben Radios und Navigationsgeräten gehören auch vermehrt große Farbdisplays zum Standard.

AUFGABEN

1. a) Wozu werden in der Informationstechnik Zeichen verwendet?
b) Nennen Sie einige Beispiele für Zeichen.
2. Was verstehen Sie unter dem Begriff „Signal“?
3. Nennen Sie die charakteristischen Merkmale zur Unterscheidung verschiedener Signalarten.
4. Wodurch unterscheiden sich analoge und digitale Signale?
5. Was verstehen Sie unter
 - a) einer unsymmetrischen Übertragung und
 - b) einer symmetrischen Übertragung?
6. Welche Art von Datenübertragung wird als LVDS bezeichnet?

4

4.2 Zahlensysteme

4.2.1 Dezimalsystem

Im täglichen Leben wird zur Darstellung von Zahlen fast ausschließlich das **Dezimalsystem** (**Zehnersystem**) benutzt. In diesem **Zahlensystem** werden die zehn verschiedenen Ziffern von 0 bis 9 in der sogenannten **Stellenschreibweise** angewendet (Bild 4.9).

| | | | | | | | |
|----------------------|---|---------------|--------------|-------------|----------------|---|-----------------|
| Dezimalzahl | 5 | 4 | 7 | 9 | , | 2 | 6 |
| Stellennummer | 4 | 3 | 2 | 1 | 1 | | 2 |
| Stellenwert | 10^3 | 10^2 | 10^1 | 10^0 | 10^{-1} | | 10^{-2} |
| Potenzwert | $5 \cdot 1000$ | $4 \cdot 100$ | $7 \cdot 10$ | $9 \cdot 1$ | $\frac{2}{10}$ | | $\frac{6}{100}$ |
| Zahlenwert | $5000 + 400 + 70 + 9 + 0,2 + 0,06 = 5479,26_{\text{dez}}$ | | | | | | |

Bild 4.9: Zahlenwert der Dezimalzahl

Aus Bild 4.9 sind die Regeln zu erkennen, nach denen Zahlensysteme aufgebaut sind:

- Die zur Darstellung einer Zahl erforderlichen Ziffern werden von einer Markierung – dem **Komma** – ausgehend nebeneinander geschrieben und nummeriert. Links vom Komma stehen Zahlen ≥ 1 , rechts vom Komma stehen Zahlen < 1 .
- Jede Stelle hat einen eigenen **Stellenwert W**; er berechnet sich aus der **Basis B des Zahlensystems** und der Stellenummer n: Stellenwert vor dem Komma : $W = B^{n-1}$
- Stellenwert nach dem Komma: $W = B^{-n} = \frac{1}{B^n}$
- Die Basis des Zahlensystems ist gleich der Anzahl der verfügbaren Ziffern.
- Der Potenzwert einer Stelle ergibt sich durch Multiplikation der Ziffer mit dem Stellenwert.
- Der Zahlenwert ist die Summe aller Potenzwerte.
- Wird beim Hochzählen in einer Stelle die höchste Ziffer (im Dezimalsystem also die 9) erreicht, so wird im folgenden Schritt ein **Übertrag** von 1 in die nächsthöhere Stelle geschrieben und die hochgezählte Stelle beginnt wieder mit 0 (Bild 4.10).

Nach diesen Regeln können Zahlensysteme mit beliebiger Basis aufgebaut werden. Fragt man jedoch nach dem Wert einer Zifferfolge in einem beliebigen Zahlensystem, so meint man mit der Kurzform „Wert“ immer den Wert dieser Zifferfolge im Dezimalsystem. Das Dezimalsystem ist damit das Bezugssystem für alle anderen Zahlensysteme.

| Hexadezimalsystem | | | | Dezimalsystem | | | | Dualsystem | | | | |
|-------------------|--------|--------|--------|---------------|--------|--------|--------|------------|-------|-------|-------|-------|
| 16^3 | 16^2 | 16^1 | 16^0 | 10^3 | 10^2 | 10^1 | 10^0 | 2^4 | 2^3 | 2^2 | 2^1 | 2^0 |
| 4096 | 256 | 16 | 1 | 1000 | 100 | 10 | 1 | 16 | 8 | 4 | 2 | 1 |
| | | | 0 | | | | 0 | | | | | 0 |
| | | | 1 | | | | 1 | | | | | 1 |
| | | | 2 | | | | 2 | | | | 1 | 0 |
| | | | 3 | | | | 3 | | | | 1 | 1 |
| | | | 4 | | | | 4 | | | 1 | 0 | 0 |
| | | | 5 | | | | 5 | | | 1 | 0 | 1 |
| | | | 6 | | | | 6 | | | 1 | 1 | 0 |
| | | | 7 | | | | 7 | | | 1 | 1 | 1 |
| | | | 8 | | | | 8 | | 1 | 0 | 0 | 0 |
| | | | 9 | | | | 9 | | 1 | 0 | 0 | 1 |
| | | A | | | 1 | 0 | | 1 | 0 | 1 | 0 | |
| | | B | | | 1 | 1 | | 1 | 0 | 1 | 1 | |
| | | C | | | 1 | 2 | | 1 | 1 | 0 | 0 | |
| | | D | | | 1 | 3 | | 1 | 1 | 0 | 1 | |
| | | E | | | 1 | 4 | | 1 | 1 | 1 | 0 | |
| | | F | | | 1 | 5 | | 1 | 1 | 1 | 1 | |
| | 1 | 0 | | | 1 | 6 | 1 | 0 | 0 | 0 | 0 | 0 |
| | 1 | 1 | | | 1 | 7 | 1 | 0 | 0 | 0 | 0 | 1 |

Bild 4.10: Zahlensysteme

4.2.2 Dualsystem

In der IT-Technik werden nur binäre Signale verarbeitet. Daher wird als Zahlensystem das **Dualsystem** (**Zweiersystem**) verwendet, das nur über zwei Ziffern verfügt. Es ist nach der gleichen Gesetzmäßigkeit aufgebaut wie das Dezimalsystem (Bild 4.11).

| Dualzahl | 1 | 0 | 1 | 0 | 1 | 1 | 1 | , | 1 |
|---------------|---|--------------|--------------|-------------|-------------|-------------|-------------|---------------|---------------|
| Stellennummer | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 1 | 2 |
| Stellenwert | 2^6 | 2^5 | 2^4 | 2^3 | 2^2 | 2^1 | 2^0 | 2^{-1} | 2^{-2} |
| Potenzwert | $1 \cdot 64$ | $0 \cdot 32$ | $1 \cdot 16$ | $0 \cdot 8$ | $1 \cdot 4$ | $1 \cdot 2$ | $1 \cdot 1$ | $\frac{1}{2}$ | $\frac{1}{4}$ |
| Zahlenwert | $64 + 0 + 16 + 0 + 4 + 2 + 1 + 0,5 + 0,25 = 87,75_{\text{dez}}$ | | | | | | | | |

Bild 4.11: Zahlenwert der Dualzahl

Im dualen und dezimalen Zahlensystem werden – wie in allen Zahlensystemen – die gleichen Zahlzeichen (Ziffern) verwendet. Um Verwechslungen zu vermeiden, ist es daher notwendig, das jeweils vorliegende Zahlensystem durch einen Index zu kennzeichnen, z. B.:

$$10_{10} = 10_{\text{dez}} = 1010_2 = 1010_{\text{du}}$$

4

Der Vergleich der Zahlen in den verschiedenen Systemen (Bilder 4.10 und 4.11) ergibt:

- Je weniger Ziffern in einem Zahlensystem verfügbar sind, umso mehr Stellen sind zur Darstellung einer Zahl erforderlich.

4.2.3 Hexadezimalsystem

In IT-Systemen werden Dualzahlen mit 8, 16, 32 und mehr Stellen verarbeitet. Für den Menschen sind solche Ziffernkolumnen sehr unübersichtlich. Deshalb ersetzt man vierstellige Dualzahlen durch ein Zahlensystem mit höheren Stellenwerten. Hierfür erweist sich das Dezimalsystem als nicht optimal, denn zur Darstellung einer einstelligen Dezimalzahl ist eine vierstellige Dualzahl erforderlich (Bild 4.10). Andererseits lassen sich mit vierstelligen Dualzahlen 16 verschiedene Zahlzeichen (Ziffern) darstellen. Ein Zahlensystem, das über 16 Ziffern verfügt, ist das **Hexadezimalsystem** (**Sechzehnersystem**, auch Sedenzimalsystem). Als Hexadezimalziffern werden die Dezimalziffern 0 bis 9 und zusätzlich die Ziffern (Buchstaben) A bis F verwendet (Bild 4.10).

Zur Umwandlung einer Dualzahl in eine Hexadezimalzahl werden vom Komma ausgehend jeweils vier Dualstellen zu einer Gruppe zusammengefasst. Jede so entstandene Gruppe wird als vierstellige Dualzahl betrachtet, deren Zahlenwert durch eine einstellige Hexadezimalzahl dargestellt wird (Bild 4.12); ihr Zahlenwert ist wieder als Dezimalzahl angegeben.

| Stellenwert der Dualzahl | 2^{15} | 2^{14} | 2^{13} | 2^{12} | 2^{11} | 2^{10} | 2^9 | 2^8 | 2^7 | 2^6 | 2^5 | 2^4 | 2^3 | 2^2 | 2^1 | 2^0 | 2^{-1} | 2^{-2} | 2^{-3} | 2^{-4} |
|---------------------------------|---|----------------------------------|----------------------------------|----------------------------------|----------------------------------|----------|-------|-------|--------------|-------|-------|-------|--------------|-------|-------|-------|-----------------|----------|----------|----------|
| Dualzahl | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | |
| Hexadezimalzahl | $\underbrace{\quad\quad\quad}_3$ | $\underbrace{\quad\quad\quad}_B$ | $\underbrace{\quad\quad\quad}_7$ | $\underbrace{\quad\quad\quad}_E$ | $\underbrace{\quad\quad\quad}_C$ | | | | | | | | | | | | | | | |
| Stellenwert der Hexadezimalzahl | 16^3 | | | | 16^2 | | | | 16^1 | | | | 16^0 | | | | 16^{-1} | | | |
| Potenzwert der Hexadezimalzahl | $3 \cdot 4096$ | | | | $11 \cdot 256$ | | | | $7 \cdot 16$ | | | | $14 \cdot 1$ | | | | $\frac{12}{16}$ | | | |
| Zahlenwert | $12288 + 2816 + 112 + 14 + 0,75 = 15230,75_{dez}$ | | | | | | | | | | | | | | | | | | | |

Bild 4.12: Umwandlung einer Dualzahl in eine Hexadezimalzahl

Eine Dezimalzahl kann in eine Hexadezimalzahl umgerechnet werden, indem die Dezimalzahl durch den größtmöglichen in ihr enthaltenen Stellenwert des Hexadezimalsystems dividiert wird. Der Rest wird durch den nächstkleineren Stellenwert geteilt usw., bis kein Rest mehr bleibt (Bild 4.13).

Die Umrechnung einer Dezimalzahl in eine Dualzahl erfolgt nach dem gleichen Schema durch fortlaufendes Teilen der Dezimalzahl durch die Stellenwerte des Dualsystems. Um den Rechenvorgang abzukürzen, wandelt man – vor allem bei vielstelligen Dezimalzahlen – diese zunächst in Hexadezimalzahlen und dann in Dualzahlen um.

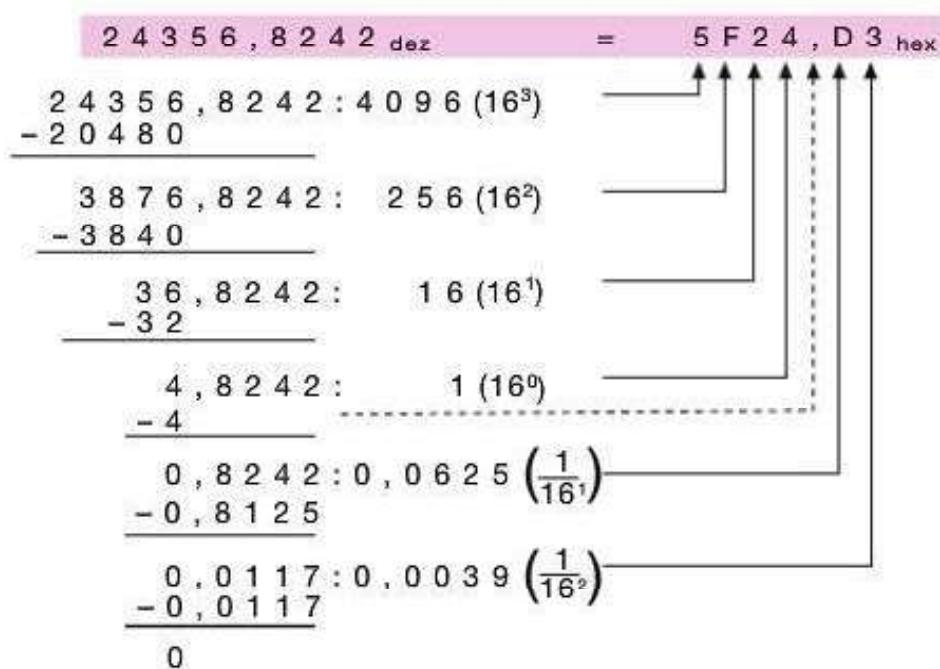


Bild 4.13: Umwandlung einer Dezimalzahl in eine Hexadezimalzahl

AUFGABEN

- Gegeben sind die Zahlen:
 - 4302,1
 - 715,02
 - 302,12
 - 1220,2

In jeder Zahl ist die höchste Ziffer auch gleichzeitig das höchste Zahlzeichen des verwendeten Zahlensystems.

 - Geben Sie an, in welchem Zahlensystem die Zahlen a)–d) dargestellt sind.
 - Ermitteln Sie für jede der vier Zahlen die entsprechende Dezimalzahl.
(Lösungshinweis: Stellen Sie für jede der vier Zahlen eine Tabelle nach dem Beispiel in Kap. 4.2 auf.)
- Wandeln Sie die folgenden Dualzahlen in Dezimalzahlen um:
 - 10110,101
 - 111101,11
 - 10011,011
 - 100010,01

3. Wandeln Sie die folgenden Dezimalzahlen in Dualzahlen um:
 - a) 4273,375
 - b) 97241,5
 - c) 37842,75
 - d) 6224,875
4. Geben Sie für die folgenden Dualzahlen die entsprechenden Hexadezimalzahlen an:
 - a) 10111001010,101
 - b) 10111000,110001
 - c) 11110011011,01
 - d) 100000111101,001
5. Wandeln Sie die folgenden Hexadezimalzahlen in Dualzahlen um:
 - a) 4BF,5
 - b) D4E,9
 - c) C94,7
 - d) 0,4B3
6. Wandeln Sie die folgenden Hexadezimalzahlen in Dezimalzahlen um:
 - a) 5F8C,3A
 - b) 27BE,7D
 - c) 974F,8B
 - d) ABCD,6E
7. Wandeln Sie die folgenden Dezimalzahlen in Hexadezimalzahlen um:
 - a) 698,5
 - b) 4763,6875
 - c) 28359,4375
 - d) 97438,125

4.3 Codes

Zur Darstellung von Informationen werden in der IT-Technik – wie überall in der zwischenmenschlichen Kommunikation – Zeichensätze verwendet. Solche Zeichensätze sind z. B. die Ziffern des Dezimalsystems oder die Buchstaben des Alphabets.

Sollen gleiche Informationen durch verschiedene Zeichensätze dargestellt werden, so müssen dafür bestimmte Vorschriften festgelegt werden. Die Vorschrift, nach der die Zuordnung der Zeichensätze erfolgt, bezeichnet man als **Code**.

Ein Code ist eine Vorschrift für die eindeutige Zuordnung der Zeichen eines Zeichensatzes zu den Zeichen eines anderen Zeichensatzes.

Sollen z. B. die Ziffern des Dezimalsystems durch Buchstabenfolgen des Alphabets dargestellt werden, so muss jeder Ziffer eine feste Buchstabenfolge zugeordnet werden (Bild 4.14). Die Umsetzung geschieht in Geräten, die man Codierer nennt (Abschnitt 4.4.3.2, Codeumsetzer). Beispiele für Codierer sind Tastaturen von Computern, Taschenrechnern, Telefonen usw.

| Zeichensatz „Dezimalziffern“ | Zeichensatz „Alphabet“ |
|------------------------------|------------------------|
| 0 | NULL |
| 1 | EINS |
| 2 | ZWEI |
| 3 | DREI |
| 4 | VIER |

Bild 4.14: Zuordnung von Ziffern und Buchstaben

4.3.1 Code-Arten

Nach ihrem Verwendungszweck unterscheidet man verschiedene Code-Arten (Bild 4.15).

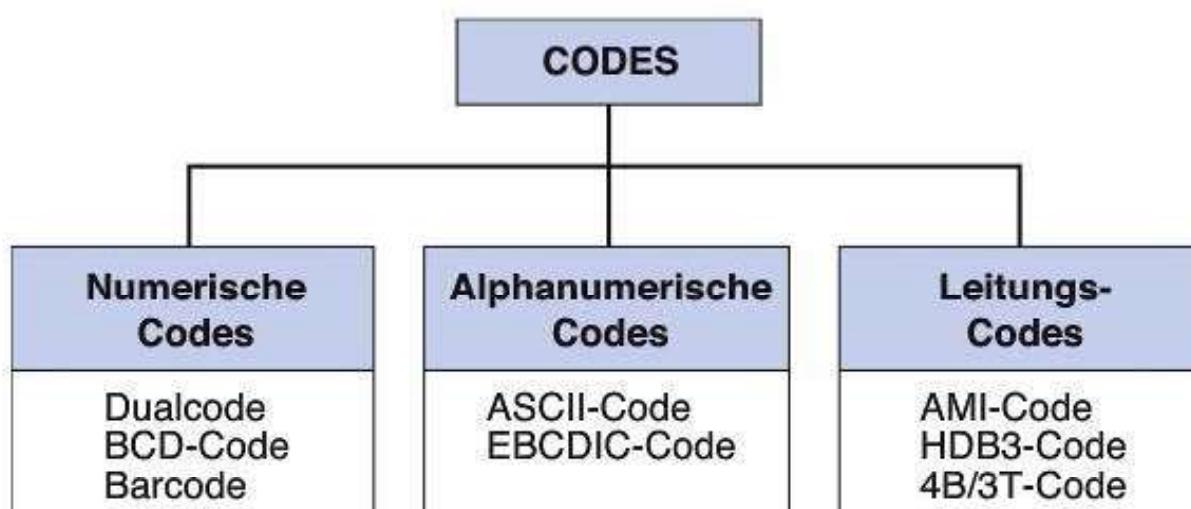


Bild 4.15: Code-Arten

- **Numerische Codes** codieren Ziffern. Angewendet werden sie beim Zählen und Rechnen, zur Codierung von Postleitzahlen oder Artikelnummern in Warenhäusern (Barcode).
- **Alphanumerische Codes** codieren neben Ziffern auch die Buchstaben des Alphabets und Steuerzeichen (siehe Abschnitt 4.3.5).
- **Leitungscodes** dienen zur Umwandlung von binären Signalen in Digitalsignale, die für das Übertragungsmedium (z. B. Kupferleitung, Lichtwellenleiter) am besten geeignet sind. Sie werden im Rahmen der Übertragungstechnik behandelt.

4.3.2 Darstellung von binären Zeichenfolgen

In der IT-Technik werden vielstellige Zeichenfolgen verarbeitet. Jede Stelle, die nur einen von zwei möglichen Werten annehmen kann – z. B. „0“ oder „1“

- wird als **Bit** (Binary Digit) bezeichnet.

Ein Bit (1 bit) ist die kleinste Informationseinheit.

Zur übersichtlichen Darstellung von Daten werden die Bits einer Zeichenfolge nummeriert und zu einem **Byte** zusammengefasst (Bild 4.16).

| 1 Byte = 8 bit | | | | | | | | |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Bit-Nummer | b ₈ | b ₇ | b ₆ | b ₅ | b ₄ | b ₃ | b ₂ | b ₁ |
| Bit-Folge | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |

Bit (1 bit)

Bild 4.16: Darstellung einer Bitfolge

Entsprechend der Stellenzahl einer Zeichenfolge spricht man bei der Informationsverarbeitung von **Datenbreite** oder **Wortlänge**, z. B. 8 bit oder 1 Byte, 16 bit oder 2 Byte usw.

Als Wort bezeichnet man eine Bitfolge, die eine Einheit bildet.

4.3.3 Binär codierte Dualzahlen

Jeder möglichen Bitkombination kann ein Zeichen, z.B. ein Buchstabe oder eine Ziffer, zugeordnet werden (vgl. Kap. 4.3.5). Soll mit einer solchen Bitfolge eine Dualzahl dargestellt werden, so muss jedem Bit ein Stellenwert des Dualsystems zugeordnet werden (Bild 4.17).

Das Bit b_1 ist die niedrigstwertige Stelle der Dualzahl und wird als LSB (Least Significant Bit) bezeichnet; das Bit b_8 als höchstwertige Stelle wird MSB (Most Significant Bit) genannt.

Das Byte kann in ein höherwertiges und ein niederwertiges Halbbyte aufgeteilt werden, dessen vier Stellen jeweils als einstellige Hexadezimalzahl angegeben werden.

Bei einer Wortlänge von 8 bit ergeben sich $2^8 = 256_{\text{dez}}$ verschiedene Bit-Kombinationen. Damit können die Zahlen von 0_{dez} bis $255_{\text{dez}} = 1111\ 1111_{\text{du}} = \text{FF}_{\text{hex}}$ dargestellt werden.

Bei der Codierung von positiven und negativen Dualzahlen werden die Vorzeichen durch Binärzeichen ersetzt:

$$\text{„0“} = „+“ \quad \text{„1“} = „-“$$

| Byte | | | | | | | | |
|-----------------|------------------------|-------|-------|-------|-------------------------|-------|-------|-------|
| Bit-Nummer | Höherwertiges Halbbyte | | | | Niederwertiges Halbbyte | | | |
| | MSB | b_7 | b_6 | b_5 | b_4 | b_3 | b_2 | b_1 |
| Stellenwert | 2^7 | 2^6 | 2^5 | 2^4 | 2^3 | 2^2 | 2^1 | 2^0 |
| Dualzahl | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| Hexadezimalzahl | 9 | | | | D | | | |

Bild 4.17: Darstellung einer Dualzahl

Stehen 8 bit zur Verfügung, so können 7 bit zur Zahlendarstellung genutzt werden, das achte Bit gibt das Vorzeichen an (Bild 4.18).

Vergleicht man die positive Dualzahl mit der negativen, so stellt man fest, dass sich bei der negativen Zahl nicht nur das Vorzeichen ändert, sondern auch die Zahl selbst.

| Byte | | | | | | | | |
|--------------------|-------|-------|-------|-------|-------|-------|-------|---|
| b_8 | b_7 | b_6 | b_5 | b_4 | b_3 | b_2 | b_1 | |
| Vz | 2^6 | 2^5 | 2^4 | 2^3 | 2^2 | 2^1 | 2^0 | |
| +45 _{dez} | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| -45 _{dez} | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |

Bild 4.18: Codierung positiver und negativer Dualzahlen

Negative Dualzahlen werden durch das Zweierkomplement dargestellt.

Mit 8-Bit-Wörtern können Dualzahlen von 1000 0000_{du} (= -128_{dez}) bis 0111 1111_{du} (= +127_{dez}) binär codiert werden.

Das Zweierkomplement einer Dualzahl wird folgendermaßen gebildet:

- Von der positiven Dualzahl wird das Einerkomplement gebildet, indem in jeder Stelle eine „0“ durch eine „1“ und eine „1“ durch eine „0“ ersetzt wird.
- Das Zweierkomplement erhält man durch Addition einer „1“ in der niedrigstwertigen Stelle des Einerkomplements.

In gleicher Weise lässt sich durch Bildung des Zweierkomplements eine negative Dualzahl in eine negative Dezimalzahl umwandeln.

Als Beispiel sind in Bild 4.19 die vierstelligen positiven und negativen Dualzahlen den entsprechenden Dezimalzahlen zugeordnet.

| Positive Dualzahl | | | | Positive Dezimalzahl | | Negative Dualzahl | | | | Negative Dezimalzahl | |
|-------------------|-------|-------|-------|----------------------|--------|-------------------|-------|-------|-------|----------------------|--------|
| Vz | 2^2 | 2^1 | 2^0 | Vz | 10^0 | Vz | 2^2 | 2^1 | 2^0 | Vz | 10^0 |
| 0 | 0 | 0 | 0 | + | 0 | 1 | 0 | 0 | 0 | - | 8 |
| 0 | 0 | 0 | 1 | + | 1 | 1 | 0 | 0 | 1 | - | 7 |
| 0 | 0 | 1 | 0 | + | 2 | 1 | 0 | 1 | 0 | - | 6 |
| 0 | 0 | 1 | 1 | + | 3 | 1 | 0 | 1 | 1 | - | 5 |
| 0 | 1 | 0 | 0 | + | 4 | 1 | 1 | 0 | 0 | - | 4 |
| 0 | 1 | 0 | 1 | + | 5 | 1 | 1 | 0 | 1 | - | 3 |
| 0 | 1 | 1 | 0 | + | 6 | 1 | 1 | 1 | 0 | - | 2 |
| 0 | 1 | 1 | 1 | + | 7 | 1 | 1 | 1 | 1 | - | 1 |

Bild 4.19: Vierstellige Dualzahlen mit Vorzeichen

4.3.4 Binär codierte Dezimalzahlen

Zur Darstellung von Dezimalzahlen mit Binärzeichen sind zwei Verfahren möglich:

- Die Dezimalzahl wird in eine Dualzahl umgewandelt, die wie oben beschrieben codiert wird.
- Jeder Stelle der Dezimalzahl wird ein eigenes 4 bit langes Codewort zugeordnet.

Die Zuordnungsvorschrift von Dezimalziffern zu Binärwörtern wird als **BCD-Code** (Binary Code Decimals) bezeichnet.

In Bild 4.20 sind einige häufig verwendete BCD-Codes mit unterschiedlichen Eigenschaften zum Vergleich nebeneinander aufgelistet.

| Dezimal-ziffer | 1-aus-10-Code | 2-aus-5-Code | 8-4-2-1-Code | 5-4-2-1-Code | 2-4-2-1-Code | Exzess-3-Code | Gray-Code |
|----------------|---------------|--------------|--------------|--------------|--------------|---------------|-----------|
| 0 | 0000000001 | 00011 | 0000 | 0000 | 0000 | 0011 | 0000 |
| 1 | 0000000010 | 00101 | 0001 | 0001 | 0001 | 0100 | 0001 |
| 2 | 0000000100 | 00110 | 0010 | 0010 | 0010 | 0101 | 0011 |
| 3 | 0000001000 | 01001 | 0011 | 0011 | 0011 | 0110 | 0010 |
| 4 | 0000010000 | 01010 | 0100 | 0100 | 0100 | 0111 | 0110 |
| 5 | 0000100000 | 01100 | 0101 | 1000 | 1011 | 1000 | 0111 |
| 6 | 0001000000 | 10001 | 0110 | 1001 | 1100 | 1001 | 0101 |
| 7 | 0010000000 | 10010 | 0111 | 1010 | 1101 | 1010 | 0100 |
| 8 | 0100000000 | 10100 | 1000 | 1011 | 1110 | 1011 | 1100 |
| 9 | 1000000000 | 11000 | 1001 | 1100 | 1111 | 1100 | 1101 |

Bild 4.20: BCD-Codes für Dezimalziffern

- Der 1-aus-10-Code und der 2-aus-5-Code sind **Fehlererkennungs-Codes**. In jedem Codewort sind nur 1 bit bzw. 2 bit mit „1“ besetzt. Ein Bitfehler wird somit erkannt und ergibt kein falsches Codewort.
- Der 8-4-2-1-Code ist ein **gewichteter Code**, d. h. jeder Stelle ist ein fester Stellenwert zugeordnet. Die Codewörter sind mit den Zahlen des Dualsystems identisch.
- Der 5-4-2-1-Code ist ein gewichteter Code. Das vierte Bit der Codewörter für die Ziffern 0 bis 4 ist mit „0“, für die Ziffern 5 bis 9 mit „1“ besetzt. Das erste bis dritte Bit ist jeweils gleich für die Ziffern 0 und 5, 2 und 6, 3 und 7 usw.
- Der 2-4-2-1-Code ist ein gewichteter und **symmetrischer Code**. Die mit „0“ bzw. „1“ besetzten Bits in den Codewörtern für die Ziffern 0 bis 4 sind in den Codewörtern für die Ziffern 5 bis 9 mit „1“ bzw. „0“ besetzt.
- Der Exzess-3-Code ist ein ungewichteter symmetrischer Code.
- Der Gray-Code ist ein **einschrittiger Code**, d. h. beim Zählen ändert sich jeweils nur 1 bit des Codewortes.

4

Die Codewörter der BCD-Codes werden auch als **Tetraden** bezeichnet. Von den 16 möglichen Tetraden werden zur Darstellung der zehn Dezimalziffern jeweils sechs Tetraden nicht verwendet; diese werden **Pseudotetraden** oder **Pseudodezimale** genannt.

Zur Codierung von mehrstelligen Dezimalzahlen wird für jede Stelle ein entsprechendes Codewort des gewählten Codes eingesetzt (Bild 4.21).

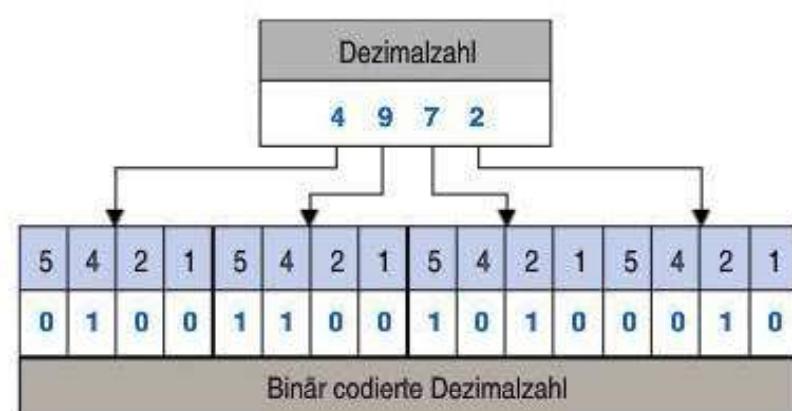


Bild 4.21: Darstellung einer vielstelligen Dezimalzahl im 5-4-2-1-Code

4.3.5 Alphanumerische Codes

Der ASCII-Code (American Standard Code for Information Interchange) ist ein international genormter 7-Bit-Code (Bild 4.22). Er dient zur Ein- und Ausgabe bei Datenverarbeitungsanlagen und zum Austausch digitaler Daten zwischen solchen Anlagen.

| | 7 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
|---------|-------------|---------|-------------|---------|-------------|---------|-------------|---------|-------------|
| | 6 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| | 5 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| Bit-Nr. | Hexadezimal | Zeichen | Hexadezimal | Zeichen | Hexadezimal | Zeichen | Hexadezimal | Zeichen | Hexadezimal |
| 4 3 2 1 | | | | | | | | | |
| 0 0 0 0 | 00 | NUL | 10 | DLE | 20 | SP | 30 | 0 | 40 |
| 0 0 0 1 | 01 | SOH | 11 | DC1 | 21 | ! | 31 | 1 | 41 |
| 0 0 1 0 | 02 | STX | 12 | DC2 | 22 | " | 32 | 2 | 42 |
| 0 0 1 1 | 03 | ETX | 13 | DC3 | 23 | # | 33 | 3 | 43 |
| 0 1 0 0 | 04 | EOT | 14 | DC4 | 24 | \$ | 34 | 4 | 44 |
| 0 1 0 1 | 05 | ENQ | 15 | NAK | 25 | % | 35 | 5 | 45 |
| 0 1 1 0 | 06 | ACK | 16 | SYN | 26 | & | 36 | 6 | 46 |
| 0 1 1 1 | 07 | BEL | 17 | ETB | 27 | ' | 37 | 7 | 47 |
| 1 0 0 0 | 08 | BS | 18 | CAN | 28 | (| 38 | 8 | 48 |
| 1 0 0 1 | 09 | HT | 19 | EM | 29 |) | 39 | 9 | 49 |
| 1 0 1 0 | 0A | LF | 1A | SUB | 2A | * | 3A | : | 4A |
| 1 0 1 1 | 0B | VT | 1B | ESC | 2B | + | 3B | ; | 4B |
| 1 1 0 0 | 0C | FF | 1C | FS | 2C | , | 3C | < | 4C |
| 1 1 0 1 | 0D | CR | 1D | GS | 2D | - | 3D | = | 4D |
| 1 1 1 0 | 0E | SO | 1E | RS | 2E | . | 3E | > | 4E |
| 1 1 1 1 | 0F | SI | 1F | US | 2F | / | 3F | ? | 4F |
| | | | | | | | | | DEL |

Bild 4.22: ASCII-Code

Der ASCII-Zeichensatz umfasst 128 Zeichen. Von diesen sind 94 Schriftzeichen, mit denen die Groß- und Kleinbuchstaben des lateinischen Alphabets, die Dezimalziffern, Interpunktionszeichen und mathematische Zeichen sowie einige Sonderzeichen (z.B. Währungszeichen) dargestellt werden. 34 Zeichen (in Bild 4.22 blau unterlegt) werden als Steuerzeichen genutzt. Ihre Bedeutung ist in Bild 4.23 aufgelistet. Die Zeichen „SP“ und „DEL“ gelten außerdem als nicht abdruckbare Schriftzeichen bzw. als Füllzeichen.

| Zeichen | Bedeutung | Zeichen | Bedeutung |
|---------|---------------------|-----------|---------------------------|
| NUL | NULL | DLE | DATALINK ESCAPE |
| SOH | START OF HEADING | DC1 BIS 4 | DEVICE CONTROL 1 BIS 4 |
| STX | START OF TEXT | NAK | NEGATIVE ACKNOWLEDGE |
| ETX | END OF TEXT | SYN | SYNCHRONOUS IDLE |
| EOT | END OF TRANSMISSION | ETB | END OF TRANSMISSION BLOCK |
| ENQ | ENQUIRY | CAN | CANCEL |
| ACK | ACKNOWLEDGE | EM | END OF MEDIUM |

| Zeichen | Bedeutung | Zeichen | Bedeutung |
|---------|-----------------------|---------|------------------|
| BEL | BELL | SUB | SUBSTITUTE |
| BS | BACKSPACE | ESC | ESCAPE |
| HT | HORIZONTAL TABULATION | FS | FILE SEPARATOR |
| LF | LINE FEED | GS | GROUP SEPARATOR |
| VT | VERTICAL TABULATION | RS | RECORD SEPARATOR |
| FF | FORM FEED | US | UNIT SEPARATOR |
| CR | CARRIAGE RETURN | SP | SPACE |
| SO | SHIFT OUT | DEL | DELETE |
| SI | SHIFT IN | | |

Bild 4.23: Bedeutung der Steuerzeichen im ASCII-Code

Die Schriftzeichen in den grau unterlegten Feldern in Bild 4.22 können durch länderspezifische Schriftzeichen ersetzt werden.

Die in der deutschen Referenz-Version des ASCII-Codes geänderten Schriftzeichen sind in Bild 4.24 dargestellt.

Wegen der üblichen 8-Bit-Wortlänge bei der Darstellung von Zeichen kann das achte Bit als **Prüfbit** zur Erkennung von Übertragsfehlern genutzt werden. In dem Beispiel in Bild 4.25 ist dem Codewort als Prüfbit ein **Paritätsbit** für gerade Parität hinzugefügt. Das Paritätsbit ist „0“, wenn die Anzahl der mit 1 besetzten Stellen des Codewortes gerade ist; es ist „1“, wenn diese Anzahl ungerade ist.

Mit einer erweiterten 8-Bit-Variante (**Extended ASCII**) können 256 Zeichen dargestellt werden. Davon entsprechen die ersten 128 Zeichen dem normalen ASCII-Code und die Zeichen von 128 bis 255 dienen zur Darstellung von weiteren Sonderzeichen (z.B. länderspezifische Zeichen) und Grafiksymbolen. Der **EBCDI-Code** (Extended BCD Interchange Code) ist ein auf ASCII basierender erweiterter Umwandlungscode. Mit diesem 8-Bit-Code sind 256 Zeichen darstellbar. Codiert werden die Dezimalziffern, die Buchstaben des lateinischen Alphabets, Sonder- und Steuerzeichen. Im Gegensatz zum ASCII-Code ist der EBCDI-Code nicht genormt.

| Hexadezimal | Zeichen | Hexadezimal | Zeichen |
|-------------|---------|-------------|---------|
| 5B | Ä | 7B | ä |
| 5C | Ö | 7C | ö |
| 5D | Ü | 7D | ü |
| | | 7E | ß |

Bild 4.24: Deutsche Referenz-Version

| Bit-Nr. | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|--------------|---|---|---|---|---|---|---|---|
| Codewort „F“ | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| Codewort „f“ | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |

Bild 4.25: ASCII-Code mit Prüfbit

4.3.6 Barcodes

Barcodes – auch Strichcodes oder Balkencodes (Bar = Balken) – sind binäre Zeichencodes, die zur Kennzeichnung von Waren im Handel und in der Lagerhaltung sowie zur Codierung von Postleitzahlen angewendet werden. Sie können mit einem Laserabtaster oder einem Lesestift (vgl. Kap. 1.11.4) entschlüsselt werden.

In Bild 4.26 ist eine Artikelkennzeichnung mit dem EAN-Code (European Article Numbering) dargestellt. Dieser Barcode besteht aus zwei Hälften, von denen jede sechs Dezimalziffern enthält. Jede Dezimalziffer wird durch sieben Binärzeichen codiert, die durch Balken („1“) oder Lücken („0“) dargestellt werden. Die beiden Hälften eines Codes werden durch Randzeichen („101“) und Trennzeichen („01010“) begrenzt.

Zur Codierung der Dezimalziffern werden die Zeichensätze A, B und C angewendet; die linke Hälfte in der Folge ABAABB, in der rechten Hälfte alle sechs Ziffern nach Zeichensatz C (Bild 4.26).

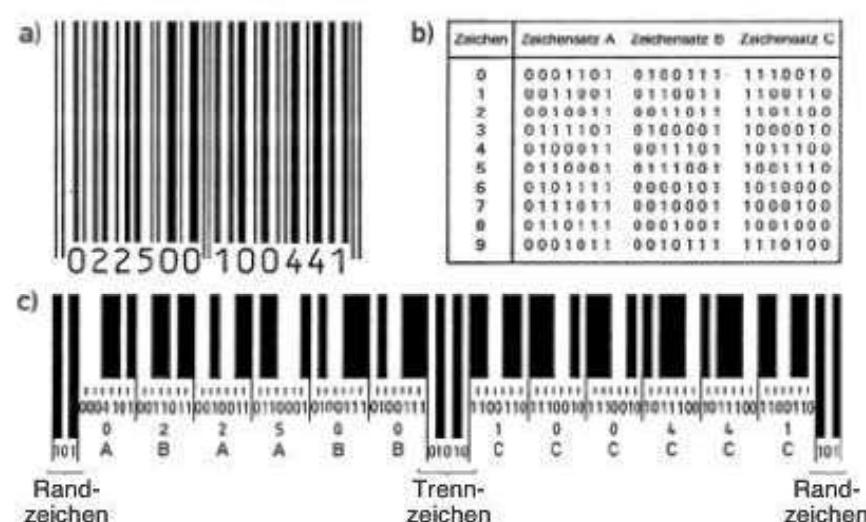


Bild 4.26: a) EAN-Codierung
b) Zeichensätze des EAN-Codes
c) Decodierung eines EAN-Codes

4.3.7 2-D-Codes

Neben den eindimensionalen Strichcodes (1-D-Barcodes) werden zunehmend zweidimensionale Codes (2-D-Codes) verwendet. Aus einer ganzen Reihe verschiedener 2-D-Codes soll hier als Beispiel der Data-Matrix-Code (DMC) kurz erläutert werden.

Den DMC gibt es in verschiedenen Versionen, von denen die aktuelle und sicherste die DMC ECC 200 ist. ECC bedeutet „Error Checking and Correction“ und beschreibt das verwendete Verfahren zur Fehlerkorrektur, bei dem die Rekonstruktion des Dateninhalts selbst dann noch möglich ist, wenn bis zu 25 % des Codes zerstört sind.

Ein DMC besteht aus mehreren Datenregionen. Diese sind aus quadratischen Symbolelementen zusammengesetzt, wovon jedes bis zu 88 numerische oder 64 alphanumerische Zeichen speichern kann. Ein Identifikationsmuster unterteilt die Region in die einzelnen Symbole. Dabei ist das Aussehen der Ränder genau festgelegt. Der linke und der untere Rand bestehen aus einem durchgezogenen schwarzen Balken, am rechten und oberen Rand wechseln sich schwarze und weiße Quadrate ab. Durch dieses Muster kann ein Bildverarbeitungssystem die Größe, die Ausrichtung, die Zahl der Zeilen

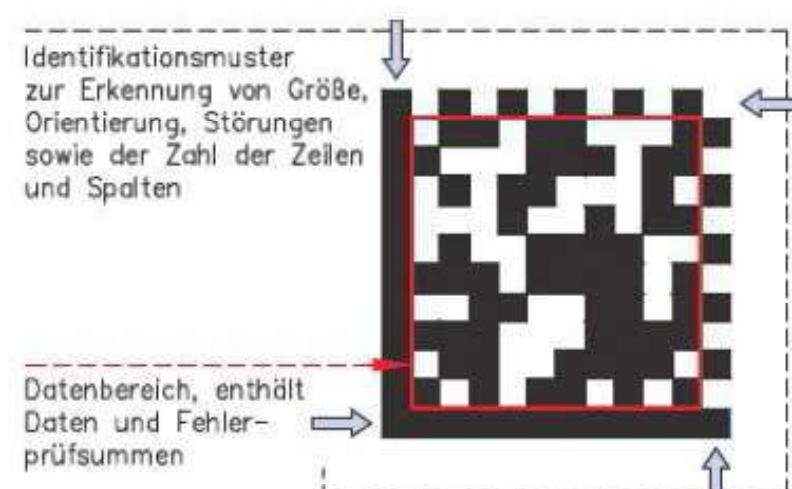


Bild 4.27: Datenregion eines DMC

und Spalten sowie die Zahl der Zeilen und Spalten. Der Datenbereich enthält Daten und Fehlerprüfsummen.

und Spalten sowie die Orientierung des Codes bestimmen. Daher sind 2-D-Codes in jeder 360-Grad-Position lesbar.

Insgesamt ist die Informationsdichte eines DMC wesentlich größer als die eines 2-D-Barcodes. Die Kapazität ist abhängig vom gespeicherten Datentyp und beträgt 1556 Bytes oder 2335 ASCII-Zeichen oder 3116 Ziffern.

Neben dem Einsatz in der Frankier-Software („Stamp IT“) der Deutschen Post wird der ECC 200 in zahlreichen Branchen für die Rückverfolgung von Produkten verwendet.



4.3.8 RFID

RFID (Radio Frequency Identification) bezeichnet eine Technik zur berührlosen automatischen Identifikation von Gegenständen und Objekten durch Funksignale. Sie ist in der ISO 15693 spezifiziert.

Ein RFID-System setzt sich aus einer ortsfesten oder tragbaren Leseeinheit mit Antenne und Decoder sowie einem am zu identifizierenden Objekt anzubringenden Funktransponder mit Antenne und einem Mikrochip zur Datenspeicherung zusammen. Die Datenspeicherung erfolgt in einem integrierten Flash-Speicher (siehe Kap. 1.5.1.2). Über das Lesegerät kann der Mikrochip nicht nur gelesen, sondern auch neu programmiert werden.

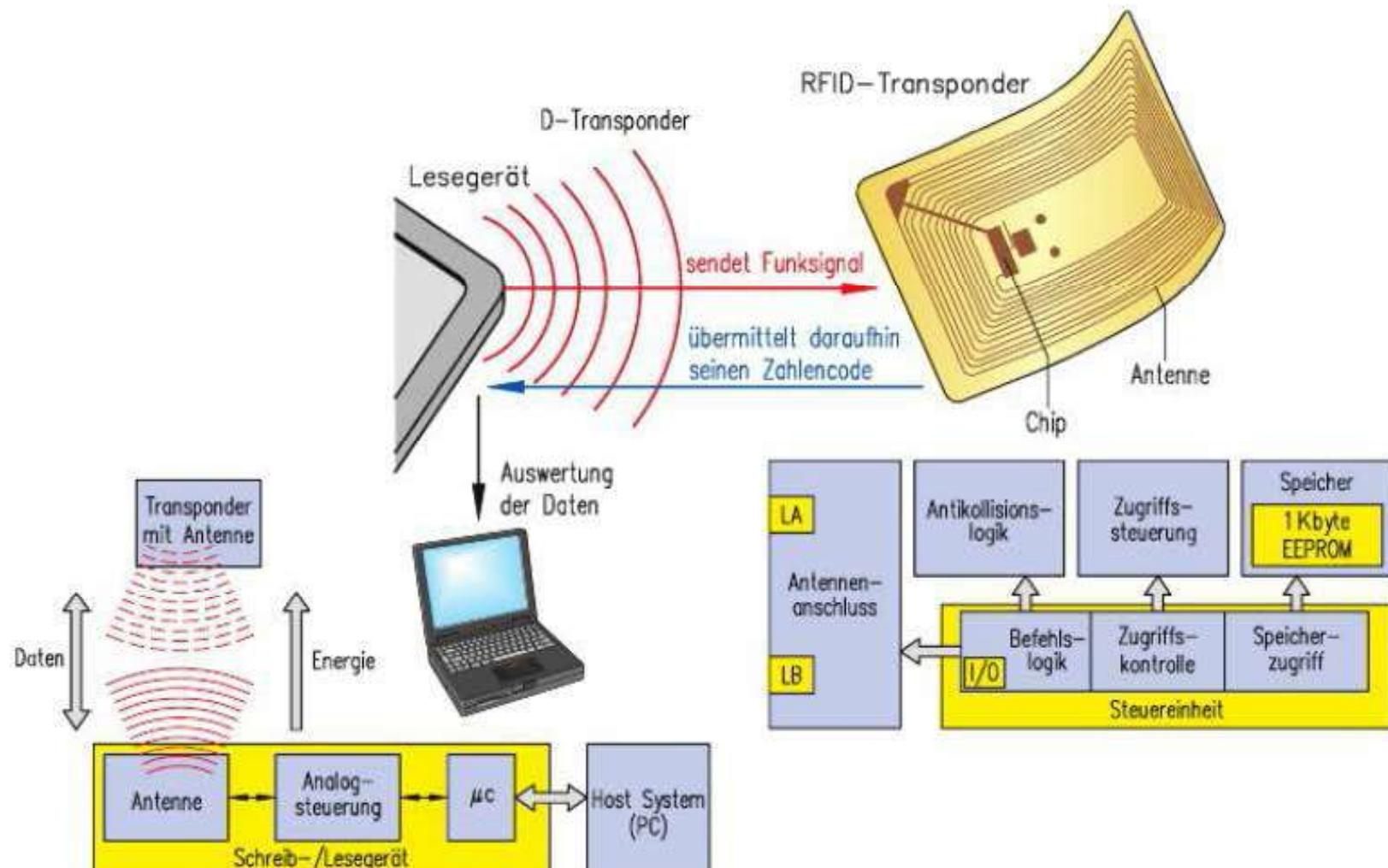


Bild 4.28: Prinzipieller Aufbau eines RFID-Systems

Aktive Transponder besitzen eine Batterie, passive Transponder beziehen die Energie, die zum Auslesen oder Neuprogrammieren erforderlich ist, mittels induktiver Kopplung über die Funkschnittstelle zum Lesegerät. Die Größe eines vollständigen RFID-Labels

(Mikrochip, Transponder, Antenne; auch als RFID-Tag oder Smart-Label bezeichnet) beträgt nur wenige Millimeter und kann beispielsweise nahezu unsichtbar unter einem dünnen Etikett angebracht werden. Die übertragenen Informationen lassen sich auf direktem Wege datentechnisch auswerten. Zur Übertragung der Informationen werden je nach Anwendung die Frequenzbereiche 125 kHz, 13,56 MHz, 860–930 MHz und 2,45 GHz genutzt. Alle Schreib-/Lesevorgänge erfolgen im Millisekundenbereich, sodass auch sich schnell bewegende Objekte erfasst und ausgewertet werden können. Zu den Anwendungsbereichen gehören die gesamte Distributionslogistik, die Handhabung von Gepäck und Paketen sowie die Personenidentifikation.

AUFGABEN

1. Was verstehen Sie unter einem Code?
2. Erklären Sie die Begriffe Bit, Byte und Datenwort.
3. Beschreiben Sie, wie positive und negative Dualzahlen binär codiert werden.
4. Geben Sie die folgenden Dezimalzahlen als binär codierte Dualzahlen mit Vorzeichenbit bei einer Wortlänge von 8 bit an:
 - a) +5 und -5, b) +40 und -40, c) +100 und -100
5. Geben Sie die höchste (positive) und niedrigste (negative) Dezimalzahl an, die mit binär codierten Dualzahlen mit Vorzeichenbit bei einer Wortbreite von 12 bit dargestellt werden kann.
6. Geben Sie für folgende mit Vorzeichenbit codierte Dualzahlen jeweils die entsprechende Dezimalzahl an:
 - a) 0110 0111 b) 1110 0111 c) 0111 1111 d) 1111 1111
7. Die Zeichenfolgen mit unterschiedlichen Wortlängen stellen binär codierte Dualzahlen mit Vorzeichen dar. Geben Sie jeweils die dargestellte Zahl als Dezimalzahl an:
 - a) 1000 0100 b) 10 0100 c) 1 0100 d) 1100
8. a) Wie unterscheidet sich der Dualcode vom 8-4-2-1-BCD-Code?
b) Geben Sie die Zahl Z = 5427dezals Dualzahl und im 8-4-2-1-BCD-Code an.
9. Der ASCII-Code (7-Bit-Code) enthält Steuerzeichen und Schriftzeichen. Die einzelnen Zeichen sind durch die Bitkombinationen in der Reihenfolge b7, b6 ... b1 gekennzeichnet.
 - a) An welcher Bitkombination sind die Steuerzeichen zu erkennen?
 - b) Zwei Zeichen des Codes sind sowohl Steuer- als auch Schriftzeichen. Geben Sie die Bitkombinationen der beiden Zeichen an.
 - c) Vergleichen Sie die Codewörter der Dezimalzahlen. Wie sind diese codiert?
 - d) Wodurch unterscheiden sich bei den Buchstaben die Bitkombinationen für die Groß- und die Kleinschreibung?
 - e) Fügen Sie den Codewörtern für die deutschen Umlaute in Groß- und Kleinschreibung als achtes Bit ein Prüfbit an, wenn eine gerade Parität vereinbart ist.
10. Was wird mit der Abkürzung RFID bezeichnet? Welche Vorteile bietet diese Technik?

4.4 Digitale Signalverarbeitung

4.4.1 Logische Verknüpfungen

4.4.1.1 Schaltalgebra

Die Schaltalgebra dient zur mathematisch exakten Beschreibung der funktionellen Zusammenhänge zwischen den Eingangs- und Ausgangssignalen digitaler Schaltelemente. Sie wurde von dem englischen Mathematiker George Boole (1815–1864) entwickelt.

Die Arbeitsweise aller Informationssysteme beruht auf den Gesetzen der „booleschen Algebra“; diese sind nur auf binäre Schaltvariable anwendbar.

Eine **binäre Schaltvariable** kann nur die zwei Werte „0“ und „1“ annehmen.

Zwischen diesen beiden Werten besteht die Beziehung:

$0 = \bar{1}$ (lies: Null ist gleich Eins nicht)

$1 = \bar{0}$ (lies: Eins ist gleich Null nicht)

Die Abhängigkeit des Ausgangssignals von den Eingangssignalen – die **Schaltfunktion** – kann dargestellt werden:

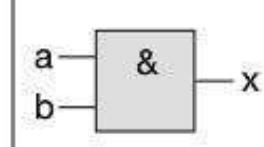
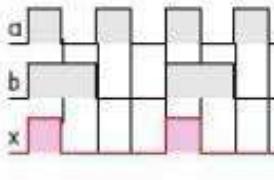
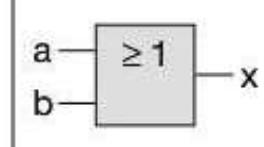
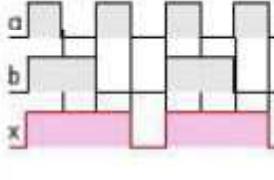
- Durch ein Symbol für die **boolesche Verknüpfung** (Bild 4.29, Spalte 1)
- Durch eine **Wahrheitstabelle** (Bild 4.29 Spalte 2)
- Durch eine **Funktionsgleichung** (Bild 4.29, Spalte 3)
- Durch ein **Zeitablaufdiagramm** (Bild 4.29, Spalte 4)

4

In der Wahrheitstabelle wird für jede mögliche Wertekombination der Eingangsvariablen der Ausgangswert angegeben.

In der Funktionsgleichung wird die Verknüpfungsart der Schaltvariablen durch Zeichen dargestellt:

- \wedge oder \cdot für **UND-Verknüpfung** (Konjunktion)
- \vee oder $+$ für **ODER-Verknüpfung** (Disjunktion)

| Schaltzeichen-Symbol | Wahrheitstabelle | Schaltfunktion, Benennung, Gleichung | Zeitablaufdiagramm | Beschreibung | | | | | | | | | | | | | | | |
|---|---|--------------------------------------|--------------------|--------------|---|---|---|---|---|---|---|---|---|---|---|---|--|---|---|
|  | <table border="1"> <thead> <tr> <th>b</th> <th>a</th> <th>x</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table> | b | a | x | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | UND-Funktion (Konjunktion) $x = a \wedge b$ |  | Der Ausgang nimmt nur dann den 1-Zustand an, wenn sich beide Eingänge im 1-Zustand befinden. |
| b | a | x | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | | | | | | | | | | | | | | | | | |
|  | <table border="1"> <thead> <tr> <th>b</th> <th>a</th> <th>x</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table> | b | a | x | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | ODER-Funktion (Disjunktion) $x = a \vee b$ |  | Der Ausgang nimmt nur dann den 1-Zustand an, wenn sich mindestens ein Eingang im 1-Zustand befindet. |
| b | a | x | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | | | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | | | | | | | | | | | | | | | | | |

| Schaltzeichen-Symbol | Wahrheitstabelle | Schaltfunktion, Benennung, Gleichung | Zeitablaufdiagramm | Beschreibung | | | | | | | | | | | | | | | |
|----------------------|--|--------------------------------------|--------------------|--------------|---|---|---|--|---|---|---|---|---|---|---|---|--|--|--|
| | <table border="1"> <thead> <tr> <th>a</th><th>x</th></tr> </thead> <tbody> <tr> <td>0</td><td>1</td></tr> <tr> <td>1</td><td>0</td></tr> </tbody> </table> | a | x | 0 | 1 | 1 | 0 | NICHT-Funktion (Negation) $x = \bar{a}$ Invertierung | | Der Ausgang nimmt nur dann den 1-Zustand an, wenn sich der Eingang im 0-Zustand befindet. | | | | | | | | | |
| a | x | | | | | | | | | | | | | | | | | | |
| 0 | 1 | | | | | | | | | | | | | | | | | | |
| 1 | 0 | | | | | | | | | | | | | | | | | | |
| | <table border="1"> <thead> <tr> <th>b</th><th>a</th><th>x</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>1</td></tr> <tr> <td>0</td><td>1</td><td>1</td></tr> <tr> <td>1</td><td>0</td><td>1</td></tr> <tr> <td>1</td><td>1</td><td>0</td></tr> </tbody> </table> | b | a | x | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | NAND-Funktion $x = \bar{a} \wedge \bar{b}$ Invertierte AND-Funktion | | Der Ausgang nimmt nur dann den 1-Zustand an, wenn sich mindestens ein Eingang im 0-Zustand befindet. |
| b | a | x | | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | | | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | | | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | | | | | | | | | | | | | | | | | |
| | <table border="1"> <thead> <tr> <th>b</th><th>a</th><th>x</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>1</td></tr> <tr> <td>0</td><td>1</td><td>0</td></tr> <tr> <td>1</td><td>0</td><td>0</td></tr> <tr> <td>1</td><td>1</td><td>0</td></tr> </tbody> </table> | b | a | x | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | NOR-Funktion $x = \bar{a} \vee \bar{b}$ Invertierte OR-Funktion | | Der Ausgang nimmt nur dann den 1-Zustand an, wenn sich beide Eingänge im 0-Zustand befinden. |
| b | a | x | | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | | | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | | | | | | | | | | | | | | | | | |
| | <table border="1"> <thead> <tr> <th>b</th><th>a</th><th>x</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>0</td></tr> <tr> <td>0</td><td>1</td><td>1</td></tr> <tr> <td>1</td><td>0</td><td>1</td></tr> <tr> <td>1</td><td>1</td><td>0</td></tr> </tbody> </table> | b | a | x | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | Antivalenz-Funktion (Exklusiv-ODER) $x = (a \wedge \bar{b}) \vee (\bar{a} \wedge b)$ XOR | | Der Ausgang nimmt nur dann den 1-Zustand an, wenn sich beide Eingänge in unterschiedlichen Zuständen befinden. |
| b | a | x | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | | | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | | | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | | | | | | | | | | | | | | | | | |
| | <table border="1"> <thead> <tr> <th>b</th><th>a</th><th>x</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>1</td></tr> <tr> <td>0</td><td>1</td><td>0</td></tr> <tr> <td>1</td><td>0</td><td>0</td></tr> <tr> <td>1</td><td>1</td><td>1</td></tr> </tbody> </table> | b | a | x | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | Äquivalenz-Funktion (Exklusiv-NOR) $x = (\bar{a} \wedge \bar{b}) \vee (a \wedge b)$ XNOR | | Der Ausgang nimmt nur dann den 1-Zustand an, wenn sich beide Eingänge in demselben Zustand befinden. |
| b | a | x | | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | | | | | | | | | | | | | | | | | |

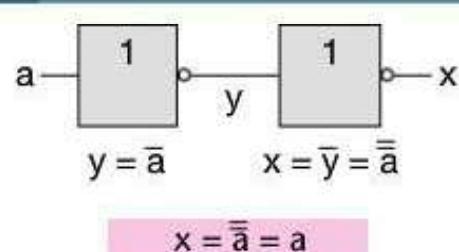
Bild 4.29: Binäre (boolesche) Verknüpfungen

Die Funktionsgleichung lässt sich aus der Wahrheitstabelle herleiten, indem

- die Eingangsvariablen einer Zeile, deren Ausgangswert „1“ ist, UND-verknüpft werden und
- alle Zeilen mit dem Ausgangswert „1“ miteinander ODER-verknüpft werden (Bild 4.29, Antivalenz- und Äquivalenz-Funktion).

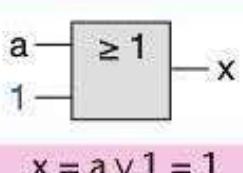
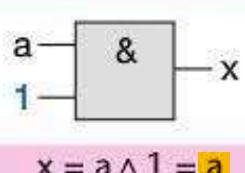
In den nachfolgenden Bildern sind die Regeln der Schaltalgebra tabellarisch zusammengefasst.

1. Negation

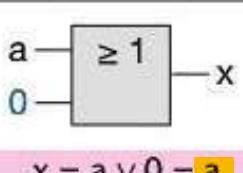
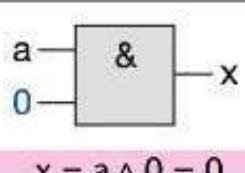


Werden beide Seiten einer Funktionsgleichung negiert, so ändert sich die Schaltfunktion nicht. Eine **doppelte Negierung hebt sich auf**.

2. Verknüpfung einer Variablen mit einer Konstanten k

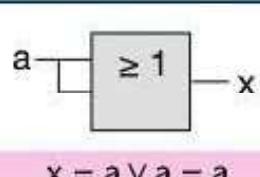
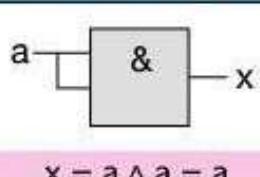


Die UND-Verknüpfung der Variablen a mit der Konstanten $k = 1$ ergibt immer $x = a$, die ODER-Verknüpfung immer $x = 1$.

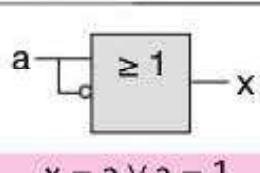
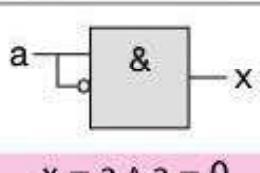


Die UND-Verknüpfung der Variablen a mit der Konstanten $k = 0$ ergibt immer $x = 0$, die ODER-Verknüpfung immer $x = a$.

3. Verknüpfung einer Variablen mit sich selbst



Sowohl die UND- als auch die ODER-Verknüpfung der Variablen a mit sich selbst ergibt immer $x = a$.

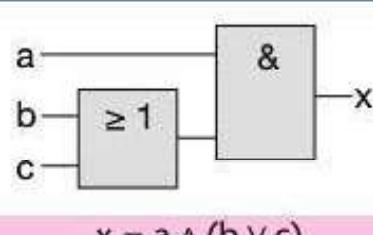
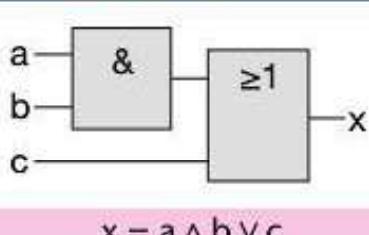


Die UND-Verknüpfung der Variablen a mit ihrer Negation \bar{a} ergibt immer $x = 0$, die ODER-Verknüpfung immer $x = 1$.

4

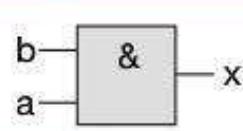
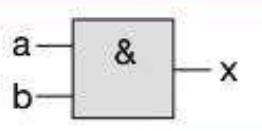
Bild 4.30: Regeln für eine Variable

1. Bindungsregel

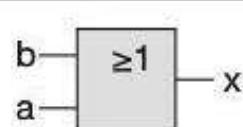
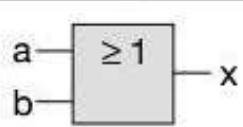


Die **UND**-Verknüpfung bindet stets **stärker** als die **ODER**-Verknüpfung. Eine **Klammer** bindet stets **stärker** als die **UND**-Verknüpfung.

2. Vertauschungsgesetz (Kommutativgesetz)



Die **Reihenfolge** der Variablen in einer UND-Verknüpfung ist **beliebig**.



Die Reihenfolge der Variablen in einer ODER-Verknüpfung ist **beliebig**.

| 3. Verbindungsgesetz (Assoziativgesetz) | | |
|--|--|--|
| | | Treten bei einer UND-Verknüpfung drei oder mehr Variablen auf, so können diese beliebig zu Gruppen zusammengefasst werden. |
| $x = a \wedge b \wedge c = a \wedge (b \wedge c)$ | | |
| 4. Verteilungsgesetz (Distributivgesetz) | | |
| | | Enthalten zwei oder mehr UND-Verknüpfungen, die miteinander ODER-verknüpft sind, eine gemeinsame Variable, so kann diese ausgeklammert werden. |
| $x = (a \wedge b) \vee (a \wedge c) = a \wedge (b \vee c)$ | | |
| | | Enthalten zwei oder mehr ODER-Verknüpfungen, die miteinander UND-verknüpft sind, eine gemeinsame Variable, so kann diese ausgeklammert werden. |
| $x = (a \vee b) \wedge (a \vee c) = a \vee (b \wedge c)$ | | |

Bild 4.31: Regeln für zwei und mehr Variable

Die nach dem englischen Mathematiker De Morgan (1806–1871) benannten Gesetze ermöglichen die Umwandlung negierter Funktionsgleichungen (Bild 4.32).

| | | |
|--|------------------------------|---|
| | | Eine NAND-Funktion kann in eine ODER-Funktion umgewandelt werden, indem die Schaltvariablen negiert werden. |
| $\bar{x} = a \wedge b$ | $x = \bar{a} \vee \bar{b}$ | |
| $x = \overline{a \wedge b} = \bar{a} \vee \bar{b}$ | | |
| | | Eine NOR-Funktion kann in eine UND-Funktion umgewandelt werden, indem die Schaltvariablen negiert werden. |
| $\bar{x} = a \vee b$ | $x = \bar{a} \wedge \bar{b}$ | |
| $x = \overline{a \vee b} = \bar{a} \wedge \bar{b}$ | | |

Bild 4.32: De-Morgansche Gesetze

4.4.1.2 Verknüpfungselemente

Verknüpfungselemente enthalten Schaltungen, die boolesche Verknüpfungen von Schaltvariablen bewirken.

In der IT-Technik werden **integrierte Schaltkreise** (IC = Integrated Circuit) eingesetzt, die mehrere gleiche Verknüpfungselemente enthalten und auf einem Halbleiterchip hergestellt werden.

Der **Funktionszusammenhang**, der zwischen der Ausgangsspannung und den Eingangsspannungen besteht, kann in einer **Arbeitstabelle** dargestellt werden. Bild 4.33 zeigt die Arbeitstabelle für eine Schaltung mit zwei Eingängen, deren Eingangs- und Ausgangsspannungen die Werte +2 V und –3 V annehmen können. Da die absoluten Spannungswerte durch die Technologie des Schaltkreises bestimmt sind, werden in der Arbeitstabelle meist nur die **Pegelwerte** angegeben:

H (High) für das höhere Potenzial
L (Low) für das niedrigere Potenzial

Die Verknüpfungsfunktion ergibt sich durch die Zuordnung der Pegel zu den binären Schaltvariablen:

Bei der **positiven Logik** gilt L = 0 und H = 1,
bei der **negativen Logik** gilt L = 1 und H = 0.

4

(Bild 4.33). Die Wahrheitstabelle zeigt, dass ein und dasselbe Verknüpfungselement abhängig von der gewählten Logik zwei verschiedene boolesche Verknüpfungen durchführen kann: bei positiver Logik eine NAND-Verknüpfung, bei negativer Logik eine NOR-Verknüpfung (Bild 4.33).

| Arbeitstabelle mit Spannungswerten | | | Pegelwerten | | | Wahrheitstabelle und Gleichung für Positive Logik | | | Negative Logik | | |
|------------------------------------|-----|-----|-------------|---|---|---|---|---|----------------|---|---|
| b | a | x | b | a | x | b | a | x | b | a | x |
| –3V | –3V | +2V | L | L | H | 0 | 0 | 1 | 1 | 1 | 0 |
| –3V | +2V | +2V | L | H | H | 0 | 1 | 1 | 1 | 0 | 0 |
| +2V | –3V | +2V | H | L | H | 1 | 0 | 1 | 0 | 1 | 0 |
| +2V | +2V | –3V | H | H | L | 1 | 1 | 0 | 0 | 0 | 1 |

$$x = \overline{a} \vee \overline{b}$$

$$= a \wedge b$$

$$x = \overline{a} \wedge \overline{b}$$

$$= a \vee b$$

Bild 4.33: Positive und negative Logik

Alle booleschen Schaltfunktionen lassen sich auf die Grundfunktionen UND, ODER und NICHT zurückführen. So entsteht eine NAND-Verknüpfung, wenn der Ausgang eines UND-Elementes mit einem NICHT-Element invertiert wird. Ein ODER-Element mit einem nachgeschalteten NICHT-Element bildet eine NOR-Verknüpfung. Das Kurzschie-

ßen der Eingänge eines NAND- oder NOR-Elementes ergibt eine NICHT-Verknüpfung (Bild 4.34).

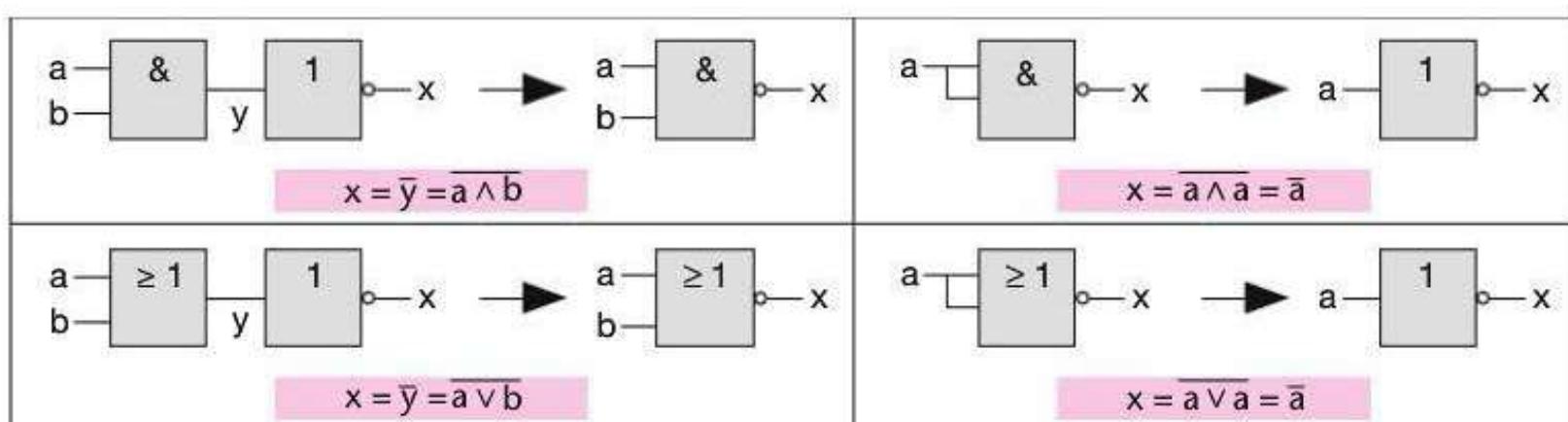


Bild 4.34: Grundverknüpfungen

Für jede boolesche Verknüpfung lassen sich nach den de-Morganschen Gesetzen zwei in der Funktion gleichwertige Gleichungen aufstellen: eine konjunktive und eine disjunktive. Ebenso lässt sich zu jedem Schaltzeichen, das eine oder mehrere Grundverknüpfungen darstellt, ein zweites gleichwertiges Schaltzeichen nach folgenden Regeln zeichnen:

1. Ein UND-Symbol wird durch ein ODER-Symbol ersetzt.
Ein ODER-Symbol wird durch ein UND-Symbol ersetzt.
2. Alle nicht negierten Ein- und Ausgänge werden negiert.
Alle negierten Ein- und Ausgänge werden nicht negiert.

Die Umwandlung einer Konjunktion (UND-Schaltung) in eine Disjunktion (ODER-Schaltung) und umgekehrt ermöglicht die Realisierung beliebiger Verknüpfungsschaltungen sowohl nur mit NAND-Elementen als auch nur mit NOR-Elementen. Dies ist besonders bedeutsam beim Einsatz integrierter Schaltkreise.

Aufgaben zu diesem ersten Teil des Kapitels finden Sie ab Seite 357.

4.4.2 Abhängigkeitsnotation

Durch die **Abhängigkeitsnotation** wird das Zusammenwirken der Anschlüsse eines Schaltzeichenblocks gekennzeichnet. Die Notierungen stehen immer innerhalb des Schaltzeichens.

Für die verschiedenen Arten von Abhängigkeiten sind in der Norm bestimmte große Kennbuchstaben festgelegt. Es werden steuernde und gesteuerte Anschlüsse unterschieden. Je nach Art der Abhängigkeit wird der entsprechende Kennbuchstabe an den steuernden Anschluss geschrieben. Hinter dem Kennbuchstaben steht eine Kennzahl. Diese Kennzahl steht auch an allen Anschläüssen, die mit dem steuernden Anschluss verknüpft sind.

Als Kennbuchstaben sind festgelegt:

- G Der steuernde Anschluss ist mit den gesteuerten Anschläüssen UND-verknüpft.
- V Der steuernde Anschluss ist mit den gesteuerten Anschläüssen ODER-verknüpft.
- N Der steuernde Anschluss ist mit den gesteuerten Anschläüssen EXOR-verknüpft.

C (Steuerabhängigkeit). Der steuernde Anschluss ist mit den gesteuerten Anschlüssen UND-verknüpft. Zusätzlich zur G-Abhängigkeit hat ein H-Pegel an diesem Anschluss eine auslösende Wirkung auf die Funktion des Bausteins (z.B. als Takteingang).

EN (Enable, Freigabeeingang). Der steuernde Anschluss ist mit den gesteuerten Anschlüssen UND-verknüpft. Steht hinter dem EN keine Kennzahl, so wirkt er als Freigabeeingang für den gesamten Baustein und hat damit eine vorbereitende Wirkung, d.h.:

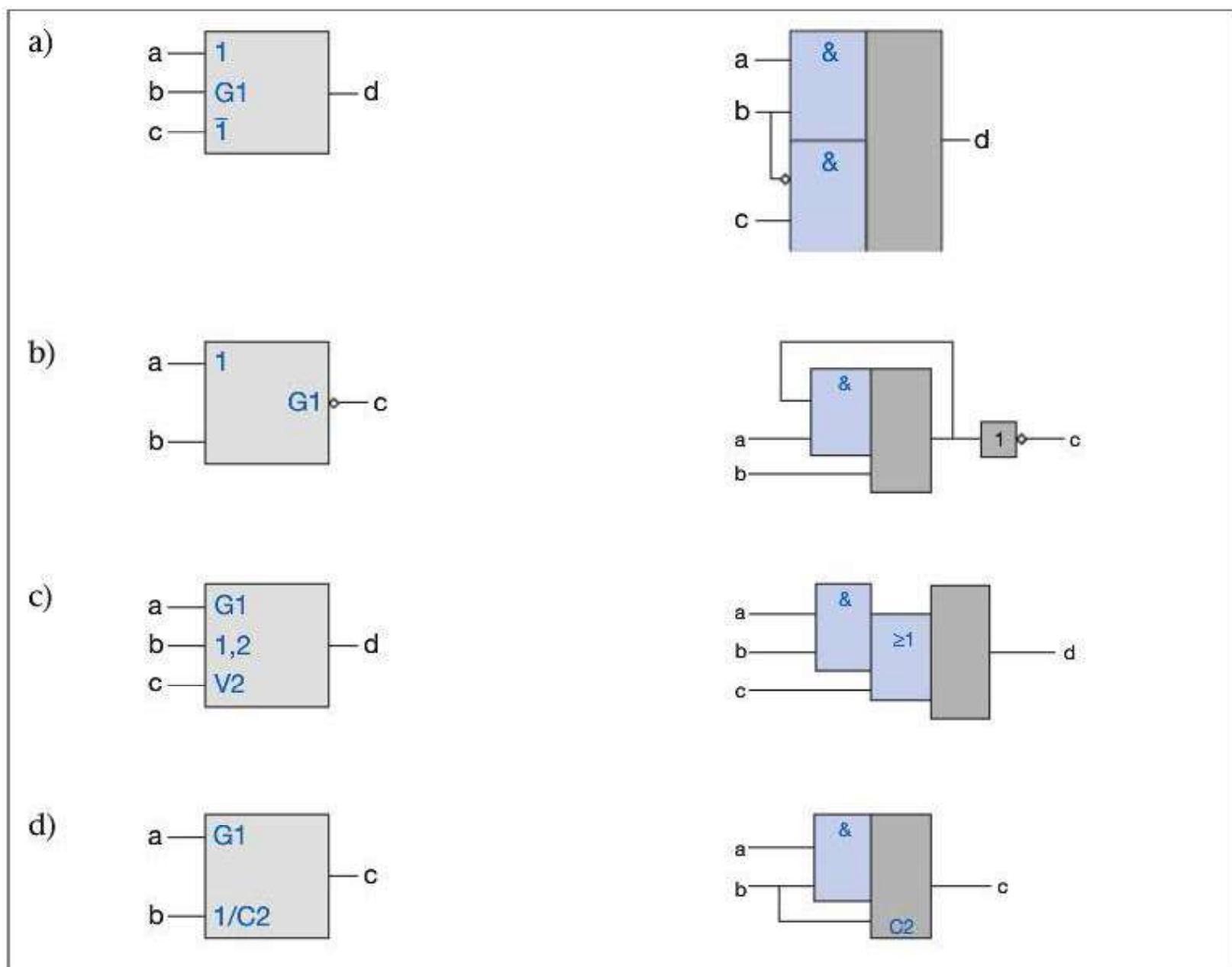
- Führt ein H-aktiver EN-Eingang H-Pegel, so haben alle von ihm gesteuerten Anschlüsse ihre normale Funktion.
- Führt ein H-aktiver EN-Eingang L-Pegel, so haben alle von ihm gesteuerten Eingänge keine Wirkung; alle Tri-state- und Open-Collektor-Ausgänge sind hochohmig und alle übrigen Ausgänge führen L-Pegel.

In Bild 4.35 a) bis e) sind einige Beispiele für zusammengesetzte (komplexe) Verknüpfungsschaltungen als Blockdarstellung mit ihrer Auflösung in Einzelverknüpfungen gezeigt.

Steht über der Kennzahl ein Negationsstrich (Bild 4.35 a, Eingang c), so bedeutet dies, dass der steuernde Anschluss (b) mit seinem negierten Pegel auf den gesteuerten Anschluss (c) wirkt.

Ist ein Anschluss außerhalb des Schaltzeichens negiert (Bild 4.35 b, Ausgang c), so bezieht sich die Abhängigkeitsnotation an diesem Ausgang auf den Pegel an der Umrandung des Schaltzeichens, also von innen gesehen vor der Negation.

4



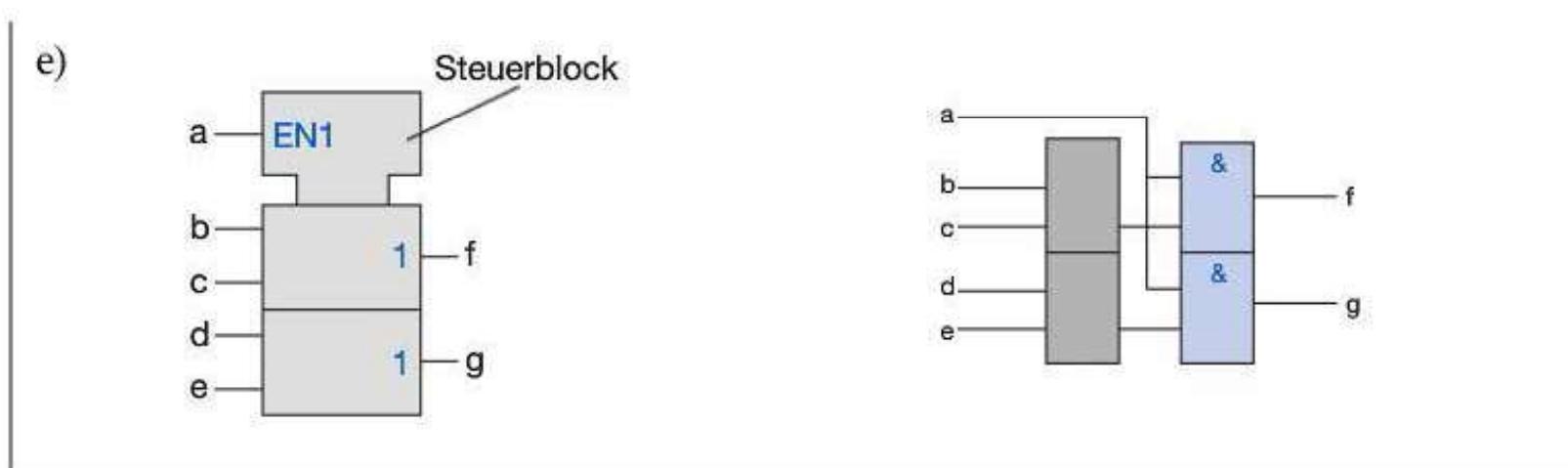


Bild 4.35: Beispiele zur Abhängigkeitsnotation

Ist ein gesteuerter Anschluss von mehreren steuernden Anschlüssen abhängig, so werden die Kennzahlen am gesteuerten Anschluss durch ein Komma getrennt (Bild 4.35 c, Eingang b). Die angegebenen Verknüpfungen sind von links nach rechts in der angegebenen Reihenfolge durchgeführt.

Ist ein Anschluss gleichzeitig gesteuert und steuernd, so werden die Bezeichnungen, die diese Wirkung beschreiben, durch einen Schrägstrich voneinander getrennt (Bild 4.35 d, Eingang b).

Besteht ein Logik-Baustein aus mehreren gleichen Verknüpfungselementen, die von gemeinsamen Steuereingängen beeinflusst werden, so werden diese Steuereingänge in einem Steuerblock zusammengefasst (Bild 4.35 e).

Außerhalb des Schaltzeichenblocks werden die Anschlüsse mit kleinen Buchstaben fortlaufend bezeichnet.

Bezieht sich das Schaltzeichen auf einen bestimmten IC, so stehen außen an den Anschlässen die Zahlen für die Pinbelegung.

4.4.3 Schaltnetze

Schaltnetze sind Verknüpfungsschaltungen, bei denen das Ausgangssignal nur von den anliegenden Eingangssignalen abhängig ist.

Schaltnetze entstehen durch Zusammenschalten von Verknüpfungselementen; sie sind als integrierte Schaltkreise (IC) erhältlich.

4.4.3.1 Addierer

Zur Addition von Dualzahlen wird den Schaltvariablen ein Stellenwert zugeordnet. Dieser ist bei einstelligen Dualzahlen 2^0 . Die Summe von zwei einstelligen Dualzahlen A und B lässt sich bei drei der vier möglichen Kombinationen der beiden Schaltvariablen in einer Stelle mit dem gleichen Stellenwert 2^0 bilden (Bild 4.36). Haben beide Schaltvariablen den Wert 1, so tritt bei der Addition ein Übertrag in den nächsthöheren Stellenwert 2^1 auf.

| Stellenwert | | | |
|-------------|-------|----------|-------|
| 2^0 | 2^0 | 2^0 | 2^1 |
| B | A | Σ | C |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

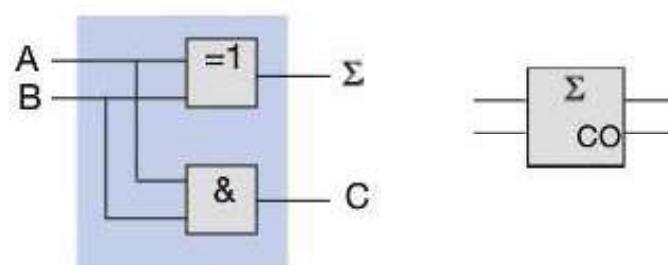


Bild 4.36: Wahrheitstabelle, Verknüpfungsschaltung und Schaltzeichen eines Halbaddierers

Anhand der Tabelle können die Funktionsgleichungen für die Summe Σ und den Übertrag C (Carry, im Schaltzeichen CO = Carry out) aufgestellt werden:

$$\Sigma = (A \wedge \bar{B}) \vee (\bar{A} \wedge B)$$

$$C = A \wedge B$$

Das Schaltnetz zum Addieren von zwei einstelligen Dualzahlen lässt sich mit einem Antivalenzelement und einem UND-Element realisieren (Bild 4.36); es wird als **Halbaddierer** bezeichnet.

Mit einem **Halbaddierer** können zwei einstellige Dualzahlen addiert werden.

4

Sollen mehrstellige Dualzahlen addiert werden, so muss in jeder Stelle noch der Übertrag aus der nächstniederwertigen Stelle addiert werden; hierzu ist ein **Volladdierer** erforderlich.

Mit einem **1-Bit-Volladdierer** können drei einstellige Dualzahlen addiert werden.

Ein Volladdierer kann aus zwei Halbaddierern geschaltet werden (Bild 4.37). Der erste Halbaddierer addiert die beiden Dualzahlen A und B zur Zwischensumme Σ_1 mit dem Übertrag C_1 . Die mit dem zweiten Halbaddierer durchgeführte Addition des Übertrags CI (Carry in) aus der nächstniederwertigen Stelle zu der Zwischensumme Σ_1 ergibt die Endsumme Σ und den Übertrag C_2 . Die Überträge C_1 und C_2 werden zum Übertrag C ODER-verknüpft.

| 1. Halbaddierer | | | | 2. Halbaddierer | | | | Übertrag |
|-----------------|-------|------------|-----------|-----------------|------------|----------|-----------|-----------|
| Stellenwert | | | | Stellenwert | | | | |
| 2^n | 2^n | 2^n | 2^{n+1} | 2^n | 2^n | 2^n | 2^{n+1} | 2^{n+1} |
| B | A | Σ_1 | C_1 | CI | Σ_1 | Σ | C_2 | C |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |

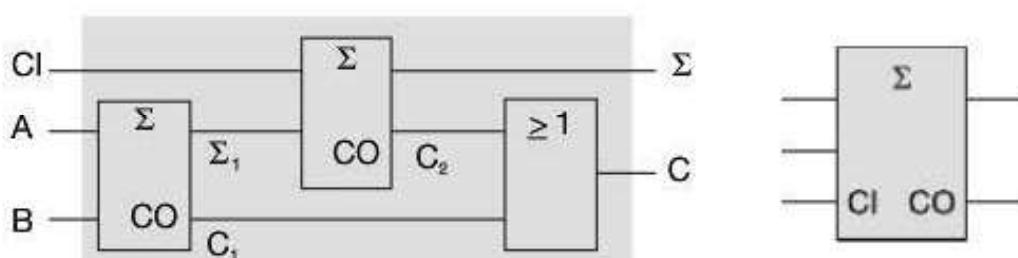


Bild 4.37: Wahrheitstabelle, Schaltung und Schaltzeichen eines 1-Bit-Volladdierers

Mit vier solcher 1-Bit-Volladdierer kann ein 4-Bit-Volladdierer geschaltet werden, der auch als IC erhältlich ist (Bild 4.38). Mit diesen Schaltkreisen lassen sich Addierer für Dualzahlen mit beliebig vielen Stellen aufbauen. Addierer für binär codierte Dezimalzahlen erfordern wegen der möglicherweise auftretenden Pseudodezimalen ein erweitertes Schaltnetz zu deren Korrektur.

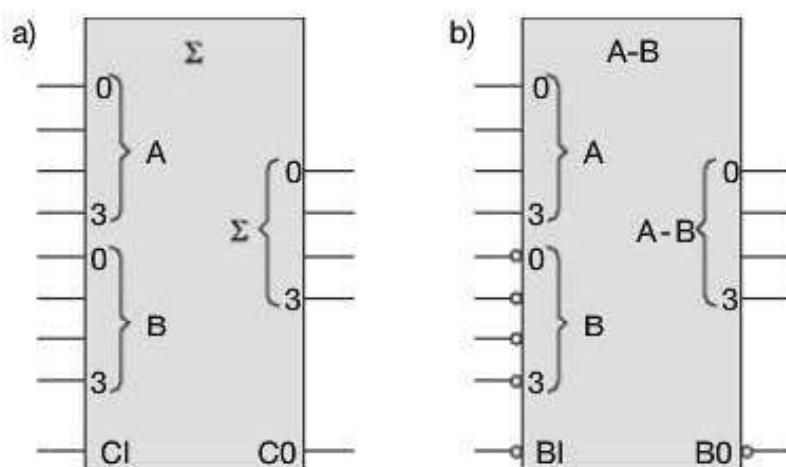


Bild 4.38: 4-Bit-Volladdierer (a) und 4-Bit-Subtrahierer (b)

Mit einem 4-Bit-Volladdierer können auch zwei vierstellige Dualzahlen durch Komplementaddition voneinander subtrahiert werden, indem die Eingänge für den Subtrahenden negiert werden (Bild 4.38). Ein- und Ausgang für den Übertrag werden ebenfalls negiert; sie werden mit BI (Borrow in) und BO (Borrow out) bezeichnet.

Volladdierer und Subtrahierer sind Funktionselemente der Alu eines Prozessors (vgl. Kap. 1.3.1).

4.4.3.2 Code-Umsetzer

Um reale Vorgänge oder Zustände in einem Informationssystem verarbeiten zu können, müssen sie mit einem **Codierer** (Coder) in binäre Signale umgewandelt werden. Nach der Verarbeitung erfolgt in einem **Decodierer** (Decoder) die Umwandlung der binären Signale in einen realen Zustand oder Vorgang.

Ein Coder ist eine Schaltung zur Umsetzung eines realen Vorgangs oder Zustandes in ein binäres Codewort.

Ein Decoder ist eine Schaltung zur Umwandlung eines Codewortes in eine unmittelbar wahrnehmbare optische oder akustische Anzeige.

Häufig erfordert die Verarbeitung binärer Daten einen Wechsel des Binärcodes; hierzu werden Code-Umsetzer (Converter) eingesetzt.

Ein Code-Umsetzer wandelt das Codewort eines Codes in ein entsprechendes Codewort eines anderen Codes um.

Bild 4.39 zeigt eine Anordnung zur optischen Anzeige der Nummern einer Tastatur. Durch Betätigen einer Taste wird die Tastennummer im 1-aus-10-Code codiert. Der Code-

Umsetzer wandelt den 1-aus-10-Code (DEC) in den 7-Segment-Code (7SEG) zur Ansteuerung der optischen Anzeige um.

Im Schaltsymbol für Code-Umsetzer wird die Art der Umsetzung durch die Bezeichnung des Eingangs- und des Ausgangscodes angegeben (Bild 4.40 b, c, d).

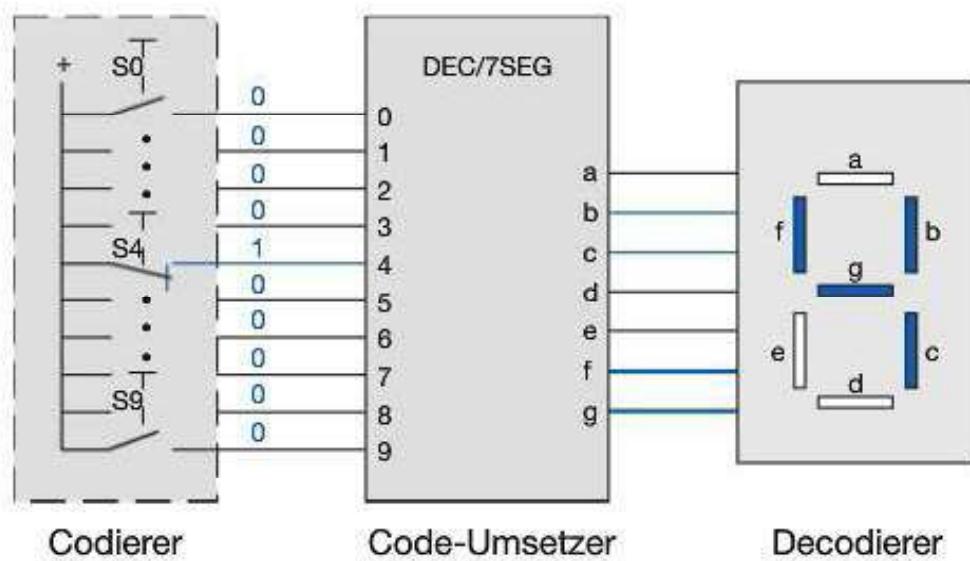


Bild 4.39: Codieren, Umcodieren und Decodieren

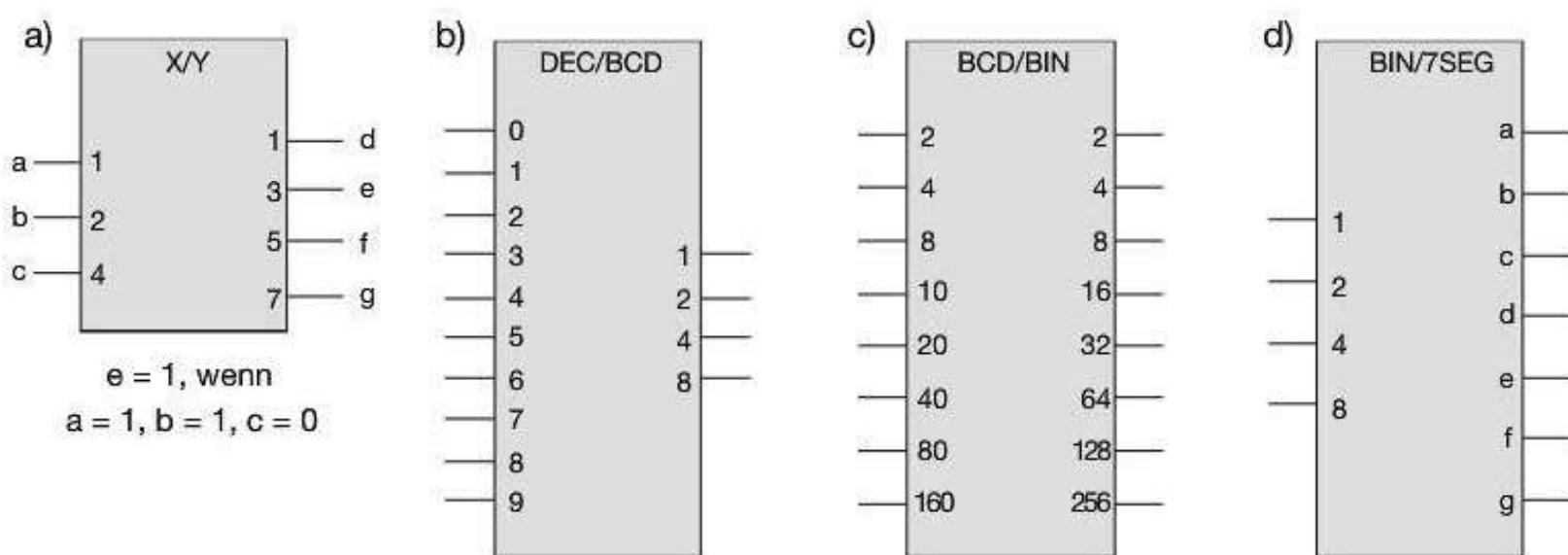


Bild 4.40: Code-Umsetzer

- a) von einem beliebigen Code X in einen Code Y
b) von 1-aus-10-Code auf BCD-Code

- c) von BCD-Code auf Binärkode
d) von Binärkode auf 7-Segment-Code

Beliebige Codes werden mit X und Y bezeichnet, wobei die Art der Umsetzung durch eine Codetabelle oder durch Zahlen an den Eingängen und Ausgängen dargestellt wird. Bei der Kennzeichnung durch Zahlen gilt: Die Summe der Eingangszahlen ergibt eine interne Zahl, die an dem Ausgang einen 1-Zustand bewirkt, der mit dieser Zahl bezeichnet ist (Bild 4.40, a).

4.4.3.3 Multiplexer und Demultiplexer

Zur besseren Ausnutzung von Leitungen sowie zur Übertragung und Anzeige binärer Daten wird die Multiplextechnik angewandt. Hierbei wird durch einen **Multiplexer** aus einer Anzahl von Eingängen jeweils einer auf den Ausgang durchgeschaltet. Die Dateneingänge eines Multiplexers werden durch Steuereingänge „adressiert“, d.h. ausgewählt (Bild 4.41). In einem Code-Umsetzer werden die Signale der beiden Steuereingänge vom Binärkode in den 1-aus-4-Code umgesetzt. Die Bezeichnung G_3^0 bedeutet, dass die vier Ausgänge des Code-Umsetzers mit den Dateneingängen 0 bis 3 UND-verknüpft sind. So wird abhängig von der Bitkombination an den Steuereingängen nur jeweils ein Dateneingang freigegeben.

Ein Multiplexer wählt aus einer Anzahl von Dateneingängen entsprechend der anliegenden Adresse einen Eingang aus, der zum Datenausgang durchgeschaltet wird.

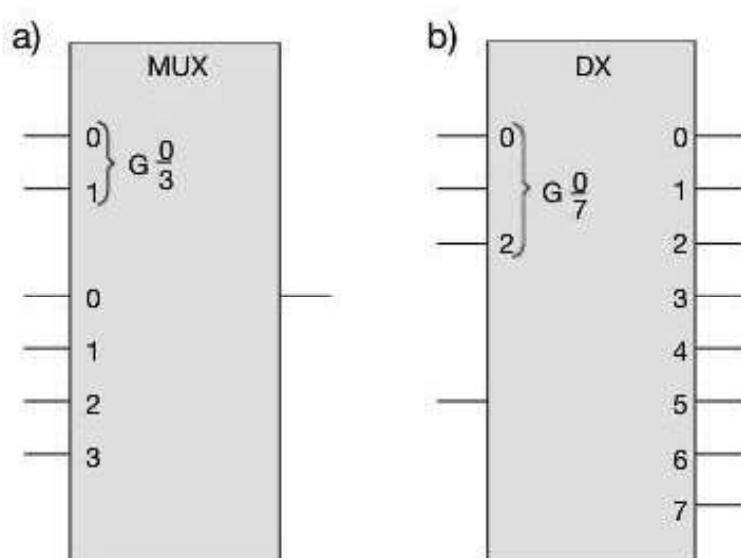


Bild 4.41: 1-aus-4-Multiplexer (a) und 1-auf-8-Demultiplexer (b)

Am Ende der Übertragungsleitung werden die ankommenden Daten durch einen **Demultiplexer** wieder auf eine Anzahl von Leitungen verteilt (Bild 4.41). Die Ausgänge des Demultiplexers werden in gleicher Weise adressiert und freigegeben wie die Eingänge des Multiplexers.

Ein **Demultiplexer** wählt aus einer Anzahl von Datenausgängen entsprechend der anliegenden Adresse einen Ausgang aus, auf den der Dateneingang durchgeschaltet wird.

Ein Multiplexer kann zur Umwandlung von parallelen Daten in serielle Daten eingesetzt werden. Entsprechend wandelt ein Demultiplexer serielle in parallele Daten um.

Aufgaben zu diesem Teil des Kapitels finden Sie ab Seite 357.

4.4.4 Schaltwerke

Schaltwerke sind Verknüpfungsschaltungen, bei denen das Ausgangssignal sowohl von den anliegenden Eingangssignalen als auch von den gespeicherten Signalwerten abhängig ist.

4.4.4.1 Bistabile Elemente

Ein **bistabiles Schaltelement (Flipflop)** hat zwei stabile Schaltzustände; seine beiden Ausgänge führen immer entgegengesetzte Signalpegel.

Das **RS-Flipflop** bildet das Grundelement aller bistabilen Schaltelemente. Es kann aus zwei NOR- oder zwei NAND-Elementen geschaltet werden, indem jeweils der Ausgang des einen Elementes auf den Eingang des anderen zurückgeführt wird (Bild 4.42).

Im Logik-Symbol werden die Eingänge mit S (Setzen) und R (Rücksetzen) bezeichnet. Der stets entgegengesetzte Signalzustand der beiden Ausgänge wird durch das Negationssymbol am Ausgang Q₂ gekennzeichnet. Jeder Eingang steuert den ihm zugeordneten (im Symbol gegenüberliegenden) Ausgang (Bild 4.42):

- S = 0, R = 0 : Das zuletzt eingelesene Signal bleibt gespeichert:
 $Q_1 = Q_n, Q_2 = \bar{Q}_n$
- S = 1, R = 0 : Flipflop wird gesetzt: $Q_1 = 1, Q_2 = 0$
- S = 0, R = 1 : Flipflop wird rückgesetzt: $Q_1 = 0, Q_2 = 1$
- S = 1, R = 1 : Dieser Signalzustand ist zu vermeiden, da er zu einem nicht definierten Signalzustand der Ausgänge führt.

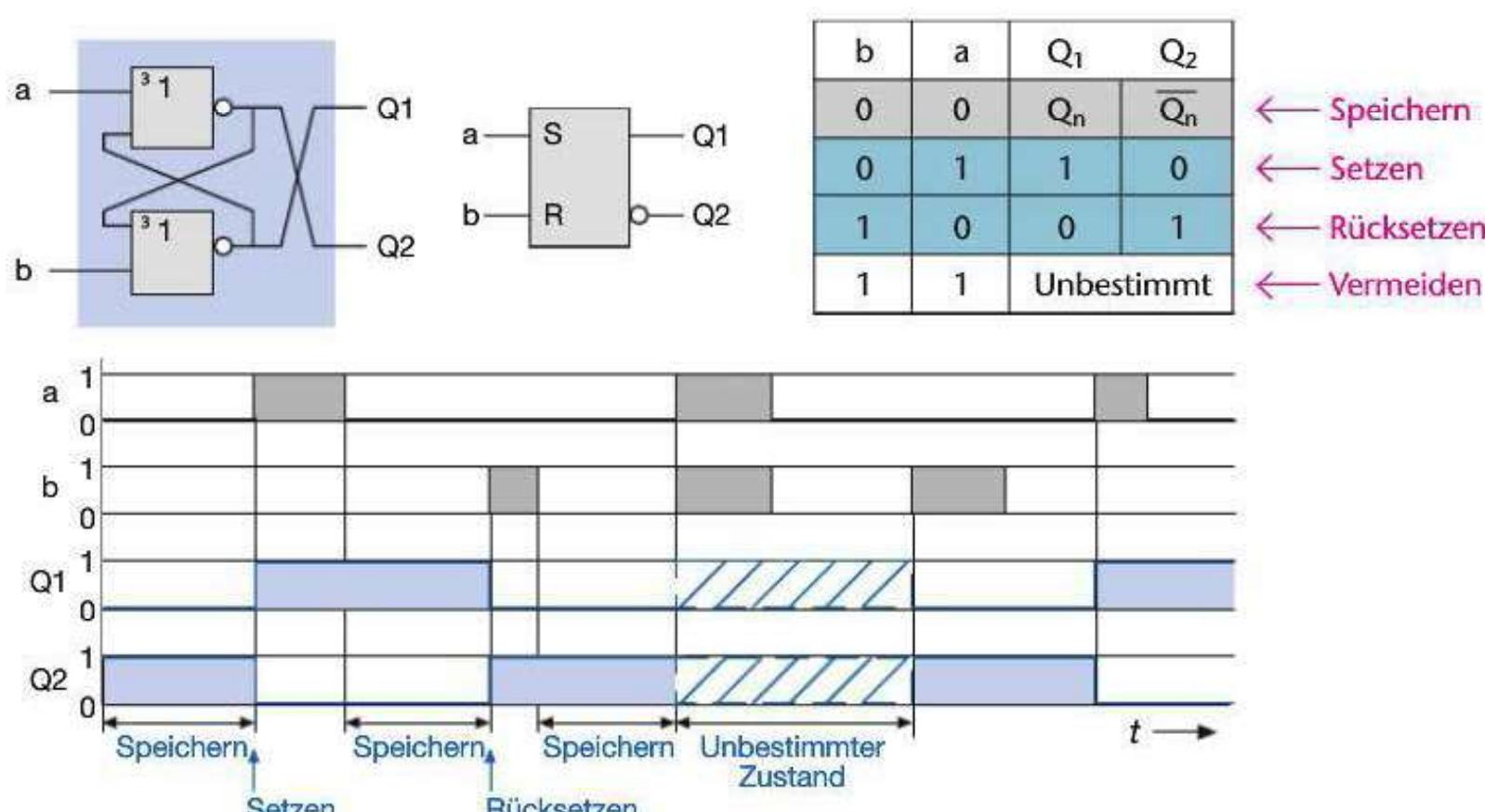


Bild 4.42: RS-Flipflop (Schaltung, Schaltzeichen, Wahrheitstabelle, Zeitablaufdiagramm)

Da bistabile Elemente vorwiegend in sequenziellen (zeitabhängigen) Schaltungen eingesetzt werden, lässt sich ihre Schaltfunktion deutlicher in einem Zeitablaufdiagramm als in einer Wahrheitstabelle darstellen (Bild 4.42).

Oft werden in Logik-Schaltungen Flipflops benötigt, die nur zu einem bestimmten Zeitpunkt die Eingangssignale aufnehmen. Ein solches Flipflop hat einen zusätzlichen Steuereingang (Bild 4.43, a).

4

■ Setz- und Rücksetzeingang eines **einzustandsgesteuerten RS-Flipflops** sind nur dann wirksam, wenn der Steuereingang C1 im internen 1-Zustand ist.

Im Logiksymbol wird der Steuereingang mit dem Buchstaben C (Clock, Takt) und einer nachgestellten Ziffer gekennzeichnet. Die gleiche Kennziffer wird vor die Kennbuchstaben aller gesteuerten Eingänge gesetzt (Abhängigkeitsnotation, vgl. Kap. 4.4.2).

Das **einzustandsgesteuerte RS-Flipflop** ist weniger störanfällig als ein ungetaktetes Flipflop, da ein Störimpuls an einem Eingang nur während des anstehenden Taktsignals wirksam werden kann.

| | | | | | |
|----|----------|-----------|-----|------------|-----------|
| a) | | | b) | | |
| | $a - 1S$ | $c - Q_1$ | | $a - 1S$ | $c - Q_1$ |
| | $c - C1$ | $b - Q_2$ | | $c - >C1$ | $b - Q_2$ |
| | | | | | |
| | a | b | c | a | b |
| | x | x | 0 | x | x |
| | | | 1 | x | x |
| | | | | 1 | 0 |
| | | | | 0 | 1 |
| | | | | 1 | 1 |
| | | | | 1 | 1 |
| | | | | Unbestimmt | |

Bild 4.43: a) Einzustandsgesteuertes und b) einflankengesteuertes RS-Flipflop

Die Störanfälligkeit lässt sich weiter verringern, wenn das Setzen und Rücksetzen des Flipflops nur in der kurzen Zeit möglich ist, in der das Taktsignal seinen Zustand wechselt. Der Wechsel von 0 auf 1 wird als positive Taktflanke, der Wechsel von 1 auf 0 als negative Taktflanke bezeichnet (Bild 4.43 b).

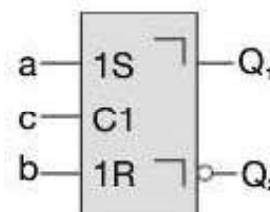
Ein einflankengesteuertes RS-Flipflop kann nur während der ansteigenden (positiven) oder während der abfallenden (negativen) Taktflanke gesetzt oder rückgesetzt werden.

Für viele Anwendungen, wie z.B. bei Schieberegistern (Abschnitt 4.4.4.2), sind Flipflops erforderlich, welche die letzte Information noch speichern, während eine neue Information eingelesen wird; diese Anforderung erfüllt ein zweizustandsgesteuertes RS-Flipflop.

Ein zweizustandsgesteuertes RS-Flipflop übernimmt die Eingangsinformation während des einen Taktsignalzustandes und gibt diese bei dem folgenden Taktsignalzustand aus.

Ein zweizustandsgesteuertes RS-Flipflop wird auch **Master-Slave-Flipflop** genannt; es enthält zwei Speicherelemente: einen Zwischenspeicher (Master) und einen Hauptspeicher (Slave). Die Ausgänge, die das Eingangssignal verzögert ausgeben, werden als **retardierende Ausgänge** bezeichnet, sie werden im Schaltsymbol besonders gekennzeichnet (Bild 4.44).

Im Gegensatz zu einem zweizustands gesteuerten sind bei einem zweiflankengesteuerten RS-Flipflop S- und R-Eingang nur während der Flanken des Taktsignals wirksam.



| c | b | a | Q ₁ | Q ₂ |
|---|---|---|----------------|------------------|
| 0 | x | x | Q _n | $\overline{Q_n}$ |
| 1 | x | x | Q _n | $\overline{Q_n}$ |
| | 0 | 0 | Q _n | $\overline{Q_n}$ |
| | 0 | 1 | 1 | 0 |
| | 1 | 0 | 0 | 1 |
| | 1 | 1 | Unbestimmt | |

Bild 4.44: Zweizustandsgesteuertes RS-Flipflop

Ein zweiflankengesteuertes RS-Flipflop übernimmt ein Eingangssignal bei der einen Taktflanke und gibt es bei der folgenden Taktflanke aus (Bild 4.45).

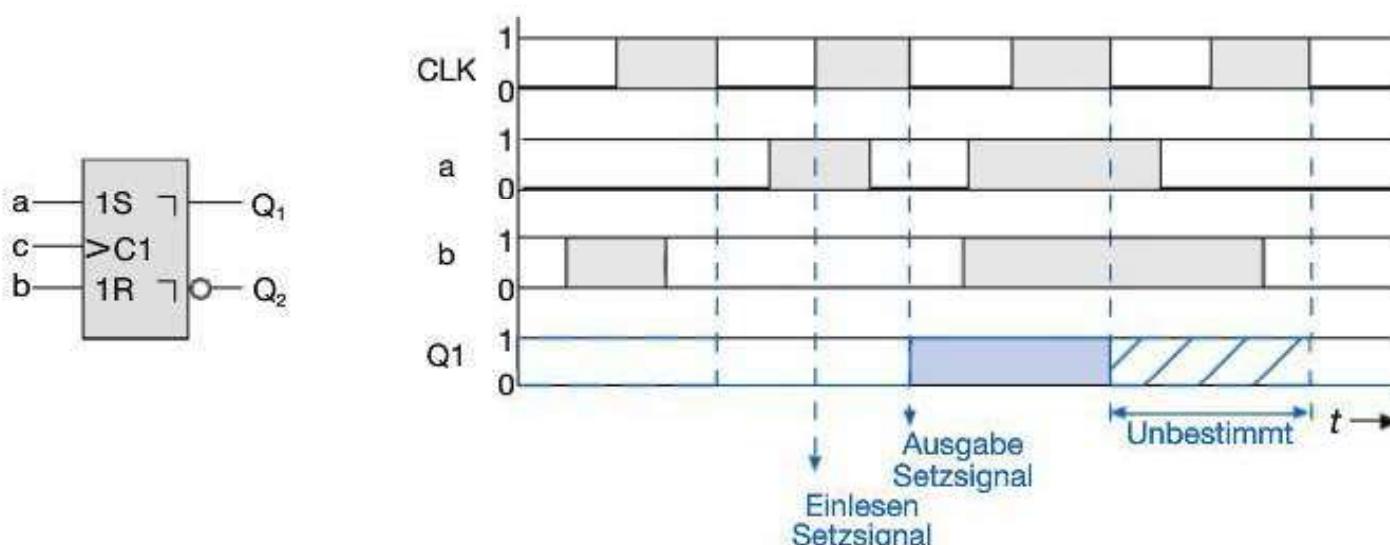


Bild 4.45: Zweiflankengesteuertes RS-Flipflop

Sind die Signaleingänge eines taktgesteuerten RS-Flipflops intern miteinander verknüpft und als gemeinsamer Anschluss nach außen geführt, so entsteht ein Flipflop, das über

nur einen Eingang gesetzt und rückgesetzt werden kann; dieser Eingang wird mit D bezeichnet (Bild 4.46).

Ein D-Flipflop speichert, durch einen Taktimpuls gesteuert, das am Dateneingang anliegende Signal.

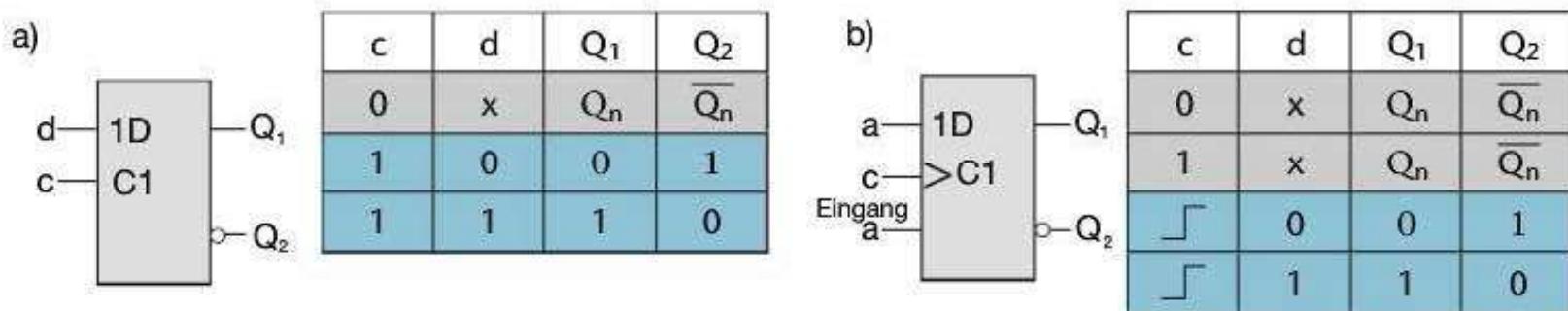


Bild 4.46: a) Einzustandsgesteuertes und b) mit abfallender Flanke gesteuertes D-Flipflop

Bei einem RS-Flipflop führt eine unerlaubte Signalkombination zu einem unbestimmten Ausgangssignal. Dies lässt sich durch eine interne Verriegelung des S- und R-Eingangs vermeiden. Die verriegelten Eingänge werden im Schaltsymbol mit J (Jump) und K (Kill) bezeichnet (Bild 4.47).

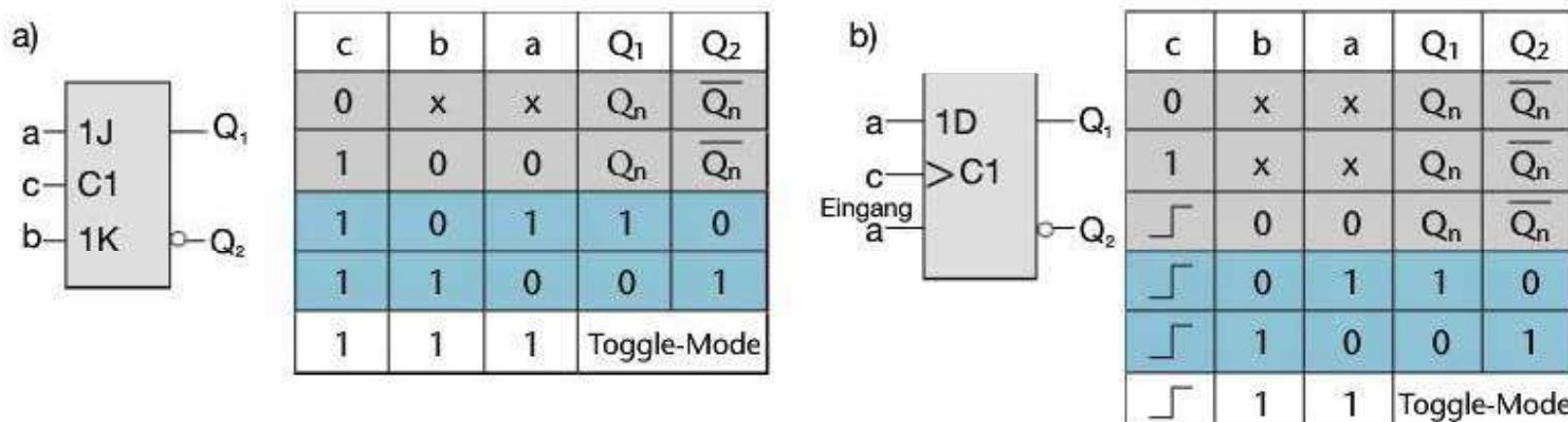


Bild 4.47: a) Einzustandsgesteuertes und b) einflankengesteuertes JK-Flipflop

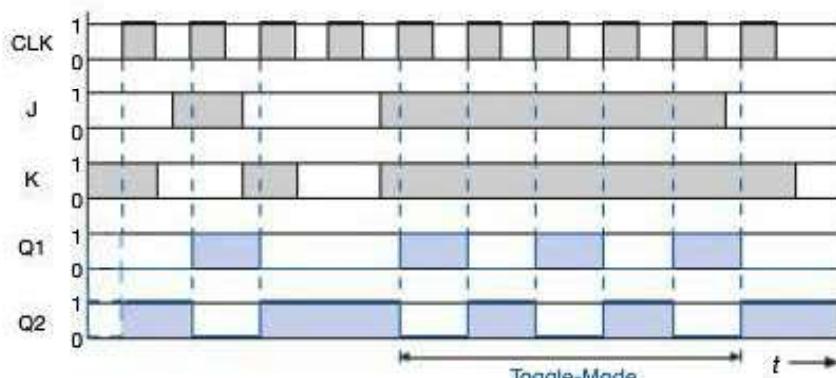


Bild 4.48: c) Zeitablaufdiagramm des JK-Flipflops

Beim **JK-Flipflop** wirkt J als Setzeingang und K als Rücksetzeingang. Führen beide Eingänge 1-Signal, so wechseln mit jedem Takt die Signalzustände der beiden Ausgänge; das Flipflop arbeitet im Toggle-Mode (Bild 4.48).

4.4.4.2 Schieberegister

Register sind kleine Speichereinheiten zur Zwischenspeicherung binärer Signale. Bei einem Schieberegister lassen sich die gespeicherten Signale mit einem Taktimpuls von einer Speicherzelle zur folgenden verschieben.

Ein **Schieberegister** ist ein taktgesteuerter digitaler Speicher, in den seriell anliegende Binärsignale eingelesen, gespeichert und mit jedem Taktimpuls um eine Stelle verschoben werden. Die seriell eingelesenen Signale werden in unveränderter Reihenfolge wieder ausgegeben (Bild 4.49).

Im Schaltsymbol werden Schieberegister mit SRG (Shift Register) gekennzeichnet; die folgende Zahl gibt die Anzahl der Speicherplätze an. Der Takteingang CLK (= Clock) steuert das Einlesen der am seriellen Dateneingang DS anstehenden Signale; der Pfeil weist auf die stellenweise Verschiebung der Signale innerhalb des Registers hin (Bild 4.49).

Wie das Zeitallaufdiagramm zeigt, erscheint bei einem 4-Bit-Schieberegister ein eingegebenes Signal nach vier Taktimpulsen am seriellen Datenausgang QS (Bild 4.49).

Schieberegister werden auch als FIFO-Speicher (First in – First out) bezeichnet, da das zuerst eingegebene Signal auch als erstes wieder ausgegeben wird.

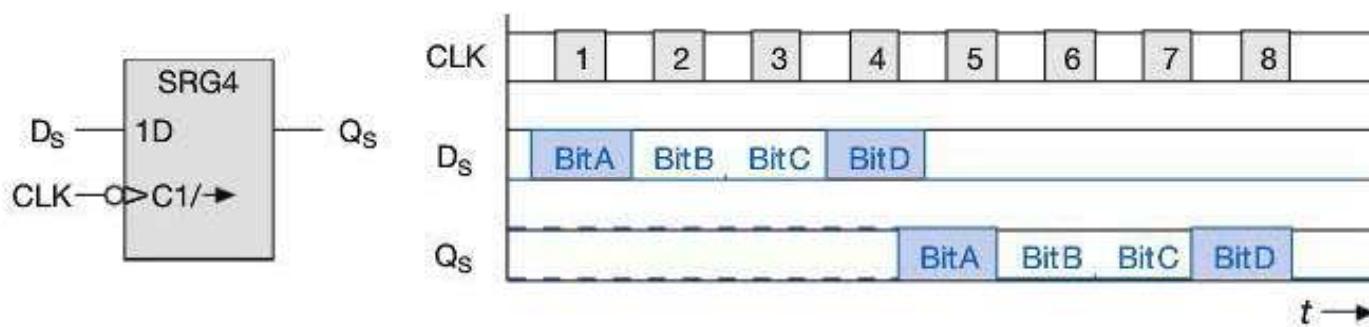


Bild 4.49: 4-Bit-Schieberegister mit serielllem Ein- und Ausgang

Ein typisches Beispiel für den Einsatz von Schieberegistern ist die Übertragung von mehrstelligen Codewörtern über eine einzige Signalleitung (Bild 4.50). Zu Beginn der Übertragung werden die Bits A bis D parallel in das Schieberegister D1 eingelesen. Mit dem Schiebakt werden die gespeicherten Signale seriell ausgegeben, übertragen und seriell in D2 eingelesen. Nach vier Takten stehen alle Bits an den Ausgängen von D2 als paralleles Signal (Codewort) an.

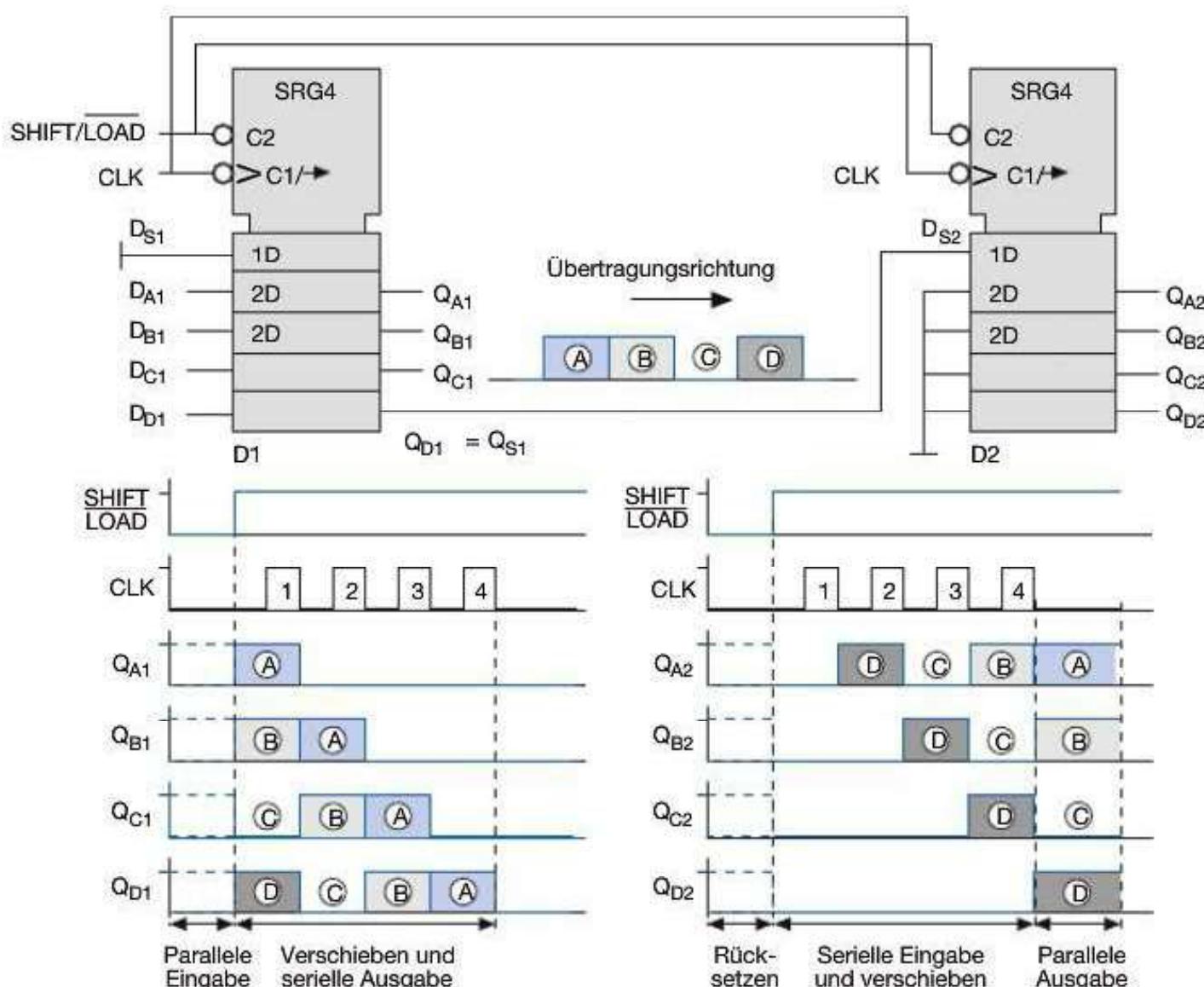


Bild 4.50: Schieberegister als Parallel-Serien-Wandler und als Serien-Parallel-Wandler

4.4.4.3 Zähler und Frequenzteiler

Ein Zähler muss zwei Bedingungen erfüllen (Bild 4.51):

- Er muss, gesteuert durch einen Zählimpuls, eine „1“ zu einer gespeicherten Zahl addieren.
- Das Ergebnis der Addition muss als neue Zahl gespeichert und ausgegeben werden.

Zähler sind Schaltwerke, bei denen ein eindeutiger Zusammenhang zwischen der Anzahl der eingegebenen Zählimpulse und dem Signalzustand der Ausgänge besteht.



Bild 4.51: Prinzip einer Zählschaltung

Ein Zähler kann mit hintereinander geschalteten Flipflops realisiert werden. Die gezählten Impulse werden durch die Signalkombinationen an den Ausgängen der Flipflops dargestellt. Nach der Zuordnung der Signalkombinationen zu Zahlen unterscheidet man:

4

Binärzähler (Dualzähler) mit n hintereinander geschalteten Flipflops zählen maximal bis $2^n - 1$. Nach 2^n Impulsen stehen sie wieder auf null.

Dekadische Zähler (Dezimalzähler) zählen maximal bis neun. Mit dem zehnten Impuls werden sie auf null zurückgesetzt.

Die Arbeitsweise eines Zählers wird bestimmt durch die Schaltungsart der Flipflops. Man unterscheidet:

Asynchrone Zähler; bei ihnen erfolgt die Ansteuerung der Flipflops nacheinander.

Synchrone Zähler; bei ihnen werden alle Flipflops gleichzeitig durch den Zählimpuls gesteuert.

Der Vorteil des asynchronen Zählers liegt in dem geringen Schaltungsaufwand. Wegen der langen Signallaufzeit ist er jedoch im Gegensatz zum synchronen Zähler nur für niedrige Frequenzen geeignet.

Ein Zähler kann durch einen Schaltzeichenblock ohne oder mit Steuerkopf dargestellt werden (Bild 4.52).

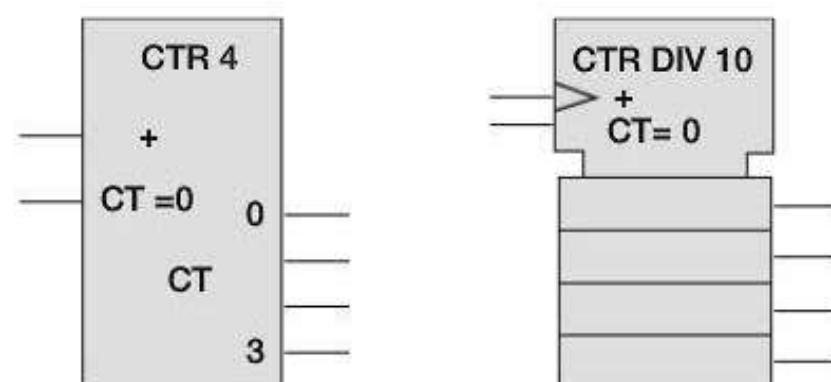


Bild 4.52: Symbole für Zähler

Die einzelnen Bezeichnungen bedeuten:

- CTR 4: Binärzähler (Counter) mit der Zykluslänge 2^4 .
- CTR DIV 10: (Divide by 10). Dekadischer Zähler mit der Zykluslänge 10
- +: Zählereingang für Vorwärtszählen
- CT: Steuereingang, mit dem sich der Zähler auf einen angegebenen Zählerstand setzen lässt

Im Schaltsymbol ohne Steuerkopf werden die Ausgänge entsprechend ihrer Wertigkeit ($0 = 2^0, 1 = 2^1$ usw.) beschriftet, durch eine Klammer zusammengefasst und mit CT (Count) bezeichnet. Bei Schaltzeichen mit Steuerkopf werden die Flipflops der Zählkette durch Rechtecke dargestellt. Die Wertigkeit der Ausgänge wird vom Steuerblock ausgehend gezählt, beginnend mit 2^0 .

In Systemen zur Informationsverarbeitung werden ausschließlich Zählerbausteine eingesetzt, die von den Herstellern als ICs angeboten werden. In Bild 4.53 ist das Schaltsymbol eines universell einsetzbaren Zählerbausteins dargestellt; er enthält einen dekadischen Zweirichtungszähler mit Voreinstellung.

Die Anschlussbezeichnungen haben folgende Bedeutung:

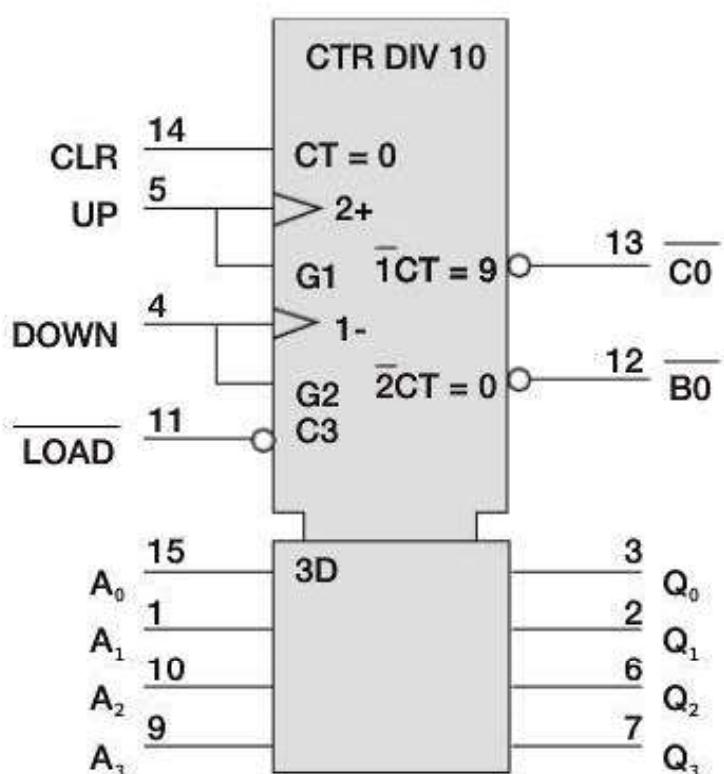


Bild 4.53: Symbol eines dekadischen Zweirichtungszählers mit Voreinstellung

- CLR: (Clear). Steuereingang zur Rückstellung des Zählerstands auf 0
- UP: Takteingang für Vorwärtszählen
- DOWN: Takteingang für Rückwärtszählen
- LOAD: Steuereingang zur Voreinstellung einer Zahl
- A₀ bis A₃: Eingänge zur Eingabe einer Zahl
- Q₀ bis Q₃: Ausgänge zur Ausgabe des Zählergebnisses. Durch Hintereinanderschalten von dekadischen Zählerbausteinen kann eine Zählschaltung für mehrstellige Dezimalzahlen aufgebaut werden. Hierfür dienen die Übertragsausgänge \overline{CO} und \overline{BO} .
- \overline{CO} : Ausgang gibt ein 0-Signal beim Zählerstand „9“ aus.
- \overline{BO} : Ausgang gibt ein 0-Signal beim Zählerstand „0“ aus.

Der Ausgang \overline{CO} wird auf den Eingang UP, der Ausgang \overline{BO} auf den Eingang DOWN der jeweils nächsthöheren Zählerdekade geschaltet.

Zählerbausteine können auch als Frequenzteiler eingesetzt werden. Dabei wird nur der Signalzustand eines Zählerausgangs ausgewertet (Bild 4.54).

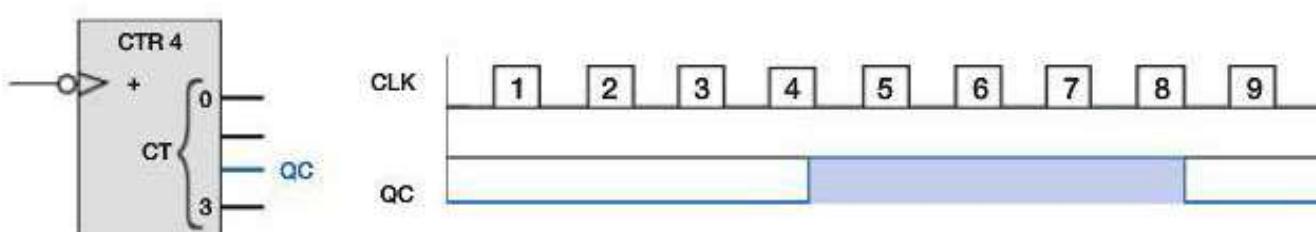


Bild 4.54: Binärzähler als Frequenzteiler 8:1

Das Teilverhältnis eines **Frequenzteilers** ist das Verhältnis der Pulsfrequenz am Eingang zur Pulsfrequenz am Ausgang des Zählers.

Durch Auswerten des Zählerstandes und Rücksetzen auf 0 lässt sich jedes ganzzahlige Teilverhältnis erzielen.

4.4.5 AD- und DA-Umsetzer

Nachrichten (Sprache, Bilder usw.) müssen zur Übertragung und Verarbeitung in elektrische Signale umgewandelt werden. Die Wandler (z. B. Mikrofone) liefern analoge Signale, die in digitale Signale umgesetzt werden. Dadurch ergeben sich wesentliche Vorteile:

- Digitale Signale können in Rechnern verarbeitet werden.
- Digitale Signale können einfacher gespeichert werden als analoge Signale.
- Digitale Signale werden bei der Übertragung weniger verzerrt.
- Die Übertragung digitaler Signale ist weniger störanfällig.

4.4.5.1 Analog-Digital-Umsetzer

Ein analoges Signal kann unendlich viele verschiedene Signalwerte annehmen. Daher kann nicht für jeden analogen Wert ein eigenes Codewort gebildet werden. Vielmehr wird der gesamte Spannungsbereich in einzelne Stufen unterteilt. Diesen ersten Schritt der Analog-Digital-Umsetzung bezeichnet man als Quantisierung.

Quantisierung ist die Einteilung des analogen Spannungsbereichs in Spannungsstufen.

In Bild 4.55 ist der analoge Spannungsbereich von $-U_{END}$ bis $+U_{END}$ in acht gleich große Stufen unterteilt. Diese sogenannten **Quantisierungsintervalle** sind durch Entscheidungswerte abgegrenzt. Ein Signalwert, der einen Entscheidungswert übersteigt, wird dem darüberliegenden Quantisierungsintervall zugeordnet. Im zweiten Schritt der AD-Umsetzung werden die Quantisierungsintervalle codiert.

Durch die **Codierung** wird jedem Quantisierungsintervall ein binäres Codewort zugeordnet.

Zur Codierung von acht Stufen (Bild 4.55) sind 3 bit erforderlich. Das MSB ist das Vorzeichenbit. Eine „1“ kennzeichnet den positiven, eine „0“ den negativen Bereich der analogen Signalspannung. Sowohl im positiven als auch im negativen Bereich werden die Quantisierungsintervalle von null ausgehend aufwärts gezählt und als Dualzahl dargestellt.

Schaltungen oder integrierte Schaltkreise (ICs), die ein analoges Signal in ein binäres Digitalsignal umsetzen, werden als Analog-Digital-Umsetzer oder Analog-Digital-Converter (ADC) bezeichnet.

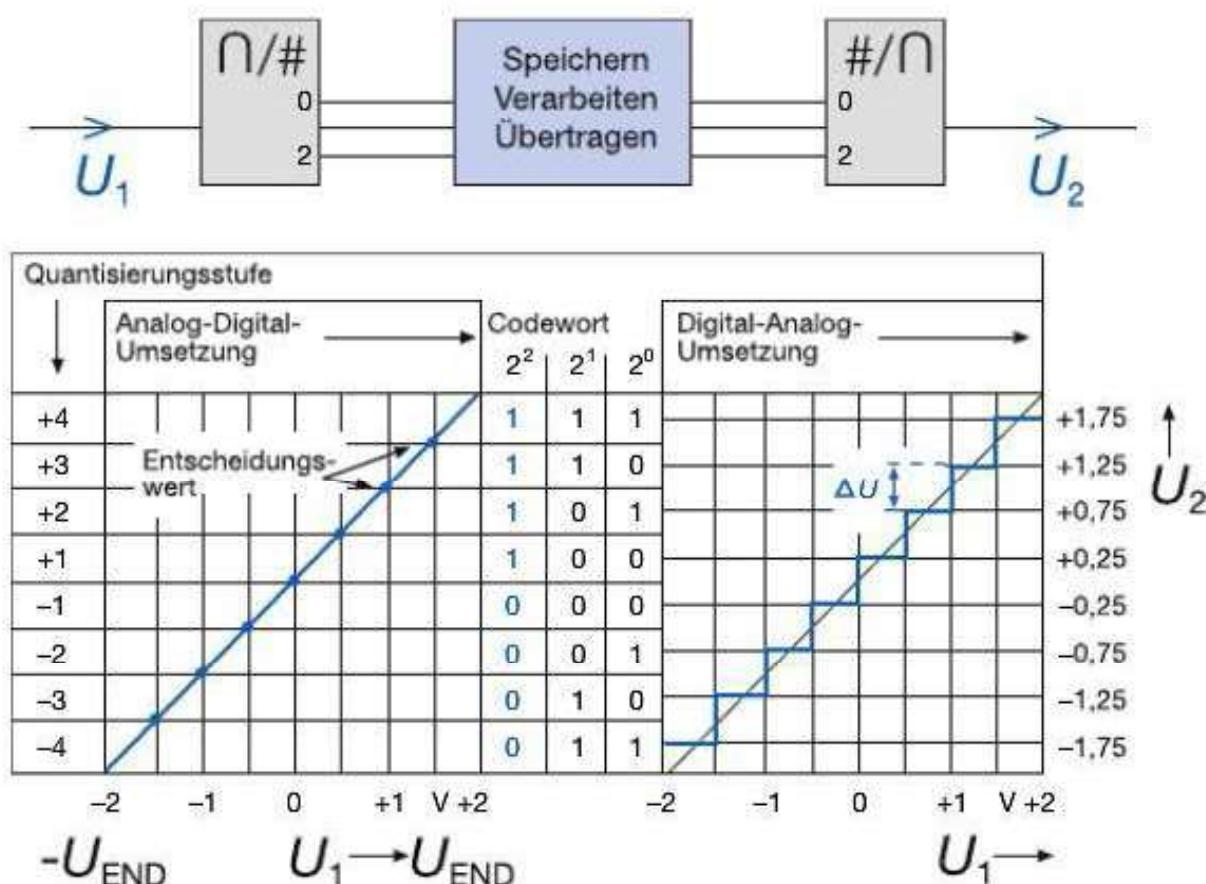


Bild 4.55: Analog-Digital- und Digital-Analog-Umsetzung

Analog-Digital-Umsetzer (ADC)

- stellen den Spannungswert des analogen Eingangssignals fest,
- ordnen diesen Wert dem Quantisierungsintervall zu und
- geben das entsprechende binäre Codewort aus.

4.4.5.2 Digital-Analog-Umsetzer

Um die ursprüngliche Form der Nachricht (Sprache, Bilder usw.) zurückzugewinnen, muss das binäre Codewort in eine analoge Spannung umgesetzt werden. Diese Aufgabe übernehmen Schaltungen, die als Digital-Analog-Umsetzer oder Digital-Analog-Converter (DAC) bezeichnet werden (Bild 4.55).

Digital-Analog-Umsetzer (DAC) setzen das an den Eingängen anliegende binäre Codewort in einen Spannungswert um.

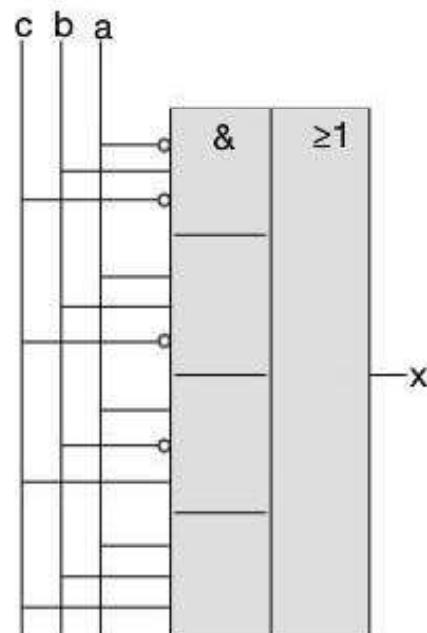
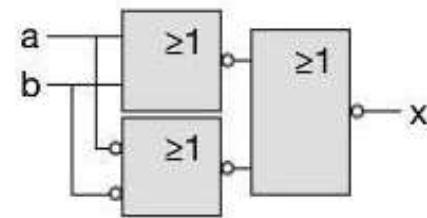
Aus jedem Codewort wird ein Spannungswert zurückgewonnen, der dem Mittelwert des Quantisierungsintervalls entspricht (Bild 4.55). Dadurch entsteht eine Abweichung des zurückgewonnenen von dem ursprünglichen Signalwert, die maximal dem halben Spannungswert einer Stufe entspricht.

Ein DAC liefert nur eine endliche Zahl von Spannungswerten. Das Ausgangssignal ist somit immer ein mehrstufiges Digitalsignal. Die Auflösung einer analogen Signalspannung in einzelne Spannungsstufen ist umso höher, je mehr Bits für die Codierung zur Verfügung stehen.

ADC und DAC werden als ICs mit einer Auflösung von 8 bit bis 24 bit hergestellt.

AUFGABEN

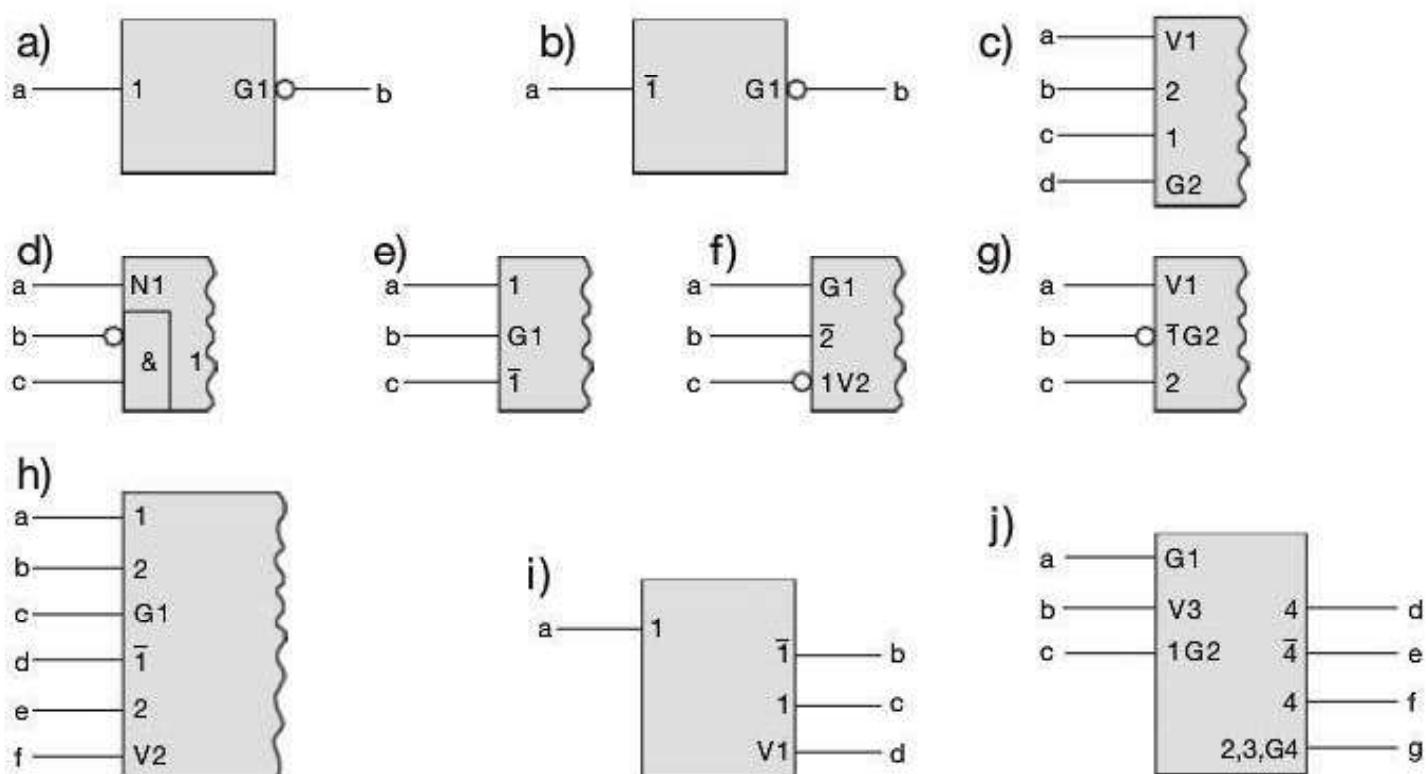
1. a) Stellen Sie für die Verknüpfungsschaltung die Wahrheitstabelle auf.
 b) Welche Funktion erfüllt die Schaltung?
 c) Geben Sie die Funktionsgleichung und das Symbol an.
2. a) Geben Sie die Funktionsgleichung der Schaltung an.
 b) Vereinfachen Sie die Gleichung nach den Regeln der Schaltalgebra.
 c) Überprüfen Sie die Gleichung anhand der Wahrheitstabelle.
3. An einem Verknüpfungselement wurden die in der Arbeitstabelle angegebenen Spannungen gemessen.
 a) Stellen Sie eine Pegeltabelle auf.
 b) Geben Sie die Funktionsgleichung des Elementes bei Anwendung der positiven und der negativen Logik an.
4. Für eine Verriegelungsschaltung wurde die Wahrheitstabelle angegeben.
 a) Stellen Sie die Funktionsgleichung auf.
 b) Vereinfachen Sie die Gleichung mithilfe der Schaltalgebra.
 c) Formen Sie die Gleichung durch Anwendung der Gesetze von De Morgan so um, dass die Verknüpfung ausschließlich mit NAND-Elementen realisiert werden kann.
 d) Formen Sie die Gleichung für eine Realisierung ausschließlich mit NOR-Elementen um.
 e) Zeichnen Sie für c) und d) die Verknüpfungsschaltungen.
5. Vereinfachen Sie die folgenden Logik-Gleichungen mit den Mitteln der Schaltalgebra.
 - a) $X = a \wedge (\bar{a} \vee b) \vee (b \wedge c \wedge \bar{c})$
 - b) $X = (a \wedge b \wedge \bar{c}) \vee (a \wedge b \wedge c)$
 - c) $X = (a \wedge \bar{b} \wedge \bar{c}) \vee (a \wedge \bar{b} \wedge c) \vee (a \wedge b \wedge \bar{c})$
 - d) $X = (a \vee b) \wedge (\bar{a} \vee b) \wedge (a \vee \bar{b})$
 - e) $X = (\bar{a} \wedge b \wedge c) \vee (\bar{a} \wedge \bar{b} \wedge c \wedge d) \vee (a \wedge b \wedge \bar{c} \wedge \bar{d}) \vee (a \wedge b \wedge \bar{c} \wedge d)$



| b | a | x |
|-------|-------|-------|
| 4,2 V | 4,2 V | 0,3 V |
| 4,2 V | 0,3 V | 0,3 V |
| 0,3 V | 4,2 V | 0,3 V |
| 0,3 V | 0,3 V | 4,2 V |

| c | b | a | x |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

6. Die Abbildung zeigt Beispiele für Digitalschaltungen in Blockdarstellung mit Abhängigkeitsnotation. Zeichnen Sie dazu jeweils die in Einzelverknüpfungen aufgelöste Schaltung nach Art von Bild 4.35.



7. Mit handelsüblichen 4-Bit-Volladdierern für Dualzahlen sollen zwei einstellige Dezimalzahlen im 8-4-2-1-BCD-Code addiert werden.

- a) Stellen Sie eine Tabelle mit folgenden Spalten auf:
 1. Die möglichen Summenwerte als Dezimalzahl
 2. Die möglichen Summenwerte im 8-4-2-1-BCD-Code
 3. Die möglichen Summenwerte im Dualcode
 b) Mit welcher Rechenoperation kann – falls erforderlich – die Dualzahl in die BCD-Zahl umgesetzt werden?
 c) Es soll eine Verknüpfungsschaltung entworfen werden, die erkennt, ob eine Ergebniskorrektur erforderlich ist. Stellen Sie die Funktionsgleichung auf.
 d) Zeichnen Sie die vollständige Addierschaltung.

8. Lösen Sie die folgenden Subtraktionsaufgaben im dualen Zahlensystem durch Addition des Zweierkomplements.

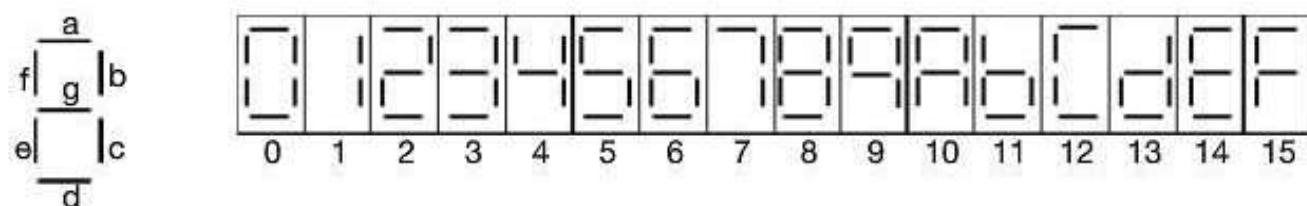
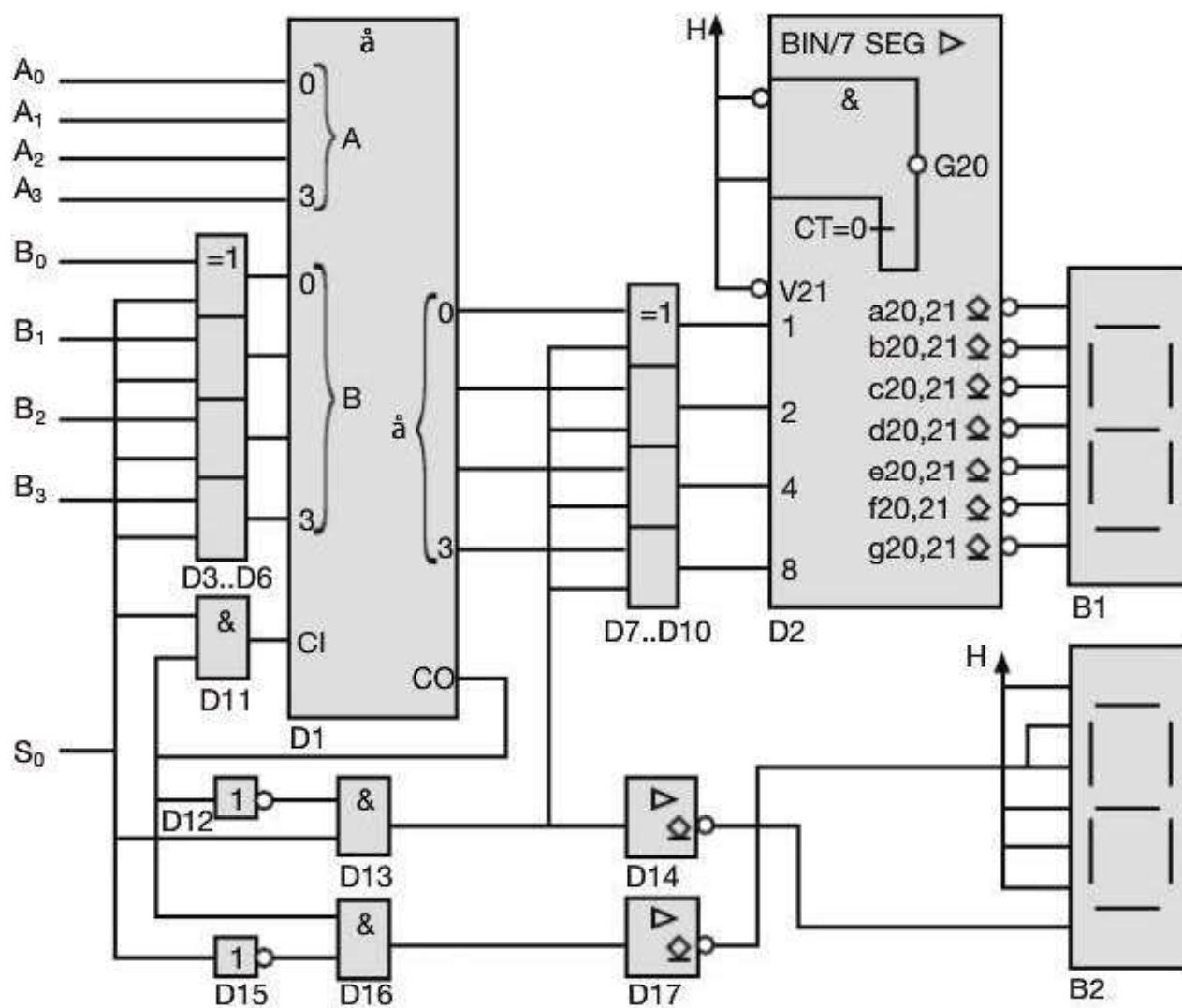
- a) 27 – 18 b) 78 – 36 c) 42 – 54 d) 246 – 139
 e) 45 – 24 f) 19 – 26 g) 63 – 87 h) 139 – 246

| | | BCD/Y |
|---|---|-------------|
| | | 2/3/4/5/8/9 |
| a | 1 | 2/3/4/5/8/9 |
| b | 2 | 3/4/5/6/7/8 |
| c | 4 | 0/4/5/6 |
| d | 8 | 0/1/2/3/4 |

9. Dargestellt ist das Schaltzeichen eines Code-Umsetzers von 8-4-2-1-BCD-Code in einen völlig unbekannten Code, dessen „1-Stellen“ an den Ausgängen angegeben sind.

- a) Stellen Sie die Wahrheitstabelle für diesen Code-Umsetzer auf.
 b) Welche Eigenschaften hat der Ausgangscode (gewichtet, symmetrisch, einschrittig)?

10. Zeichnen Sie einen 1-aus-4-Multiplexer mit Freigabeeingang (EN) mit Verknüpfungselementen.
11. Die unten stehende Schaltung soll anhand der folgenden Fragen analysiert werden.
 - 1) Geben Sie die Funktion der Bausteine D1...D17, B1 und B2 an.
 - 2) Beschreiben Sie die Funktion der Eingänge S₀, A₀ ... A₃ und B₀ ... B₃.
 - 3) Beschreiben Sie die Funktion der 7-Segment-Anzeige B₂.



Legen Sie eine Tabelle mit folgendem Kopf an:

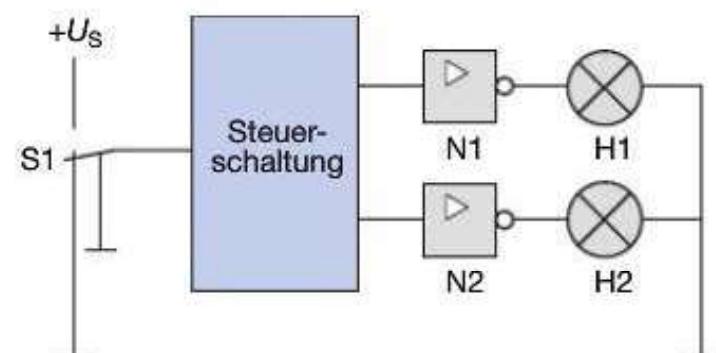
| Operation | Addierer | | | | | | | | | | | | Decodierer | | | | | | | Anzeige | | | | | | | |
|-----------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----|----|----------------|----------------|----------------|----------------|---|---|---|---|---|---------|---|---|---|---|---|----------------|----------------|
| A ± B | A ₃ | A ₂ | A ₁ | A ₀ | B ₃ | B ₂ | B ₁ | B ₀ | Cl | CO | Σ ₃ | Σ ₂ | Σ ₁ | Σ ₀ | 8 | 4 | 2 | 1 | g | f | e | d | c | b | a | B ₂ | B ₁ |

Verfolgen Sie in der Schaltung den Ablauf der folgenden Rechenoperationen und tragen Sie die auftretenden Signalzustände in die Tabelle ein.

- a) 3 + 6
- b) 5 + 5
- c) 9 + 7
- d) 15 + 15
- e) 7 - 5
- f) 15 - 12
- g) 14 - 2
- h) 4 - 8
- i) 9 - 14
- j) 13 - 15

12. Worin unterscheiden sich Schaltnetze und Schaltwerke?
13. Die Eingänge eines RS-Flipflops sollen so miteinander verknüpft werden, dass bei einem 1-Signal an beiden Eingängen das Flipflop rückgesetzt wird.
- Stellen Sie die Wahrheitstabelle auf.
 - Zeichnen Sie die Beschaltung der Eingänge.
14. Mit zweizustandsgesteuerten RS-Flipflops soll eine Steuerschaltung entworfen werden, welche die tabellarisch dargestellte Funktion erfüllt.

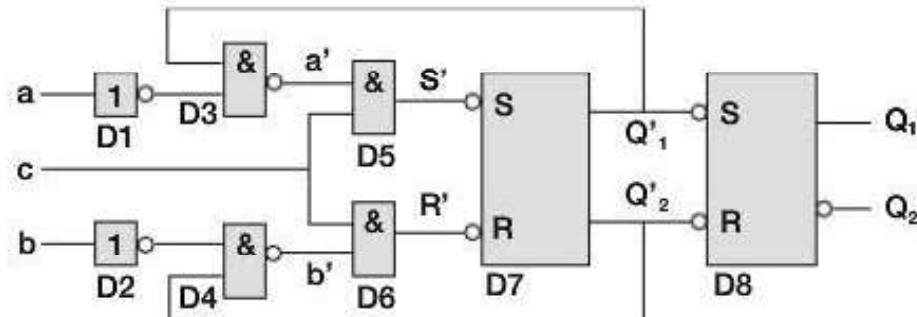
| Betätigung von S1 | Schaltzustand H1 | Schaltzustand H2 |
|-------------------|------------------|------------------|
| einmal | ein | aus |
| zweimal | ein | ein |
| dreimal | aus | ein |
| viermal | aus | aus |



- Zeichnen Sie die Steuerschaltung.
- Zeichnen Sie in ein Zeitablaufdiagramm die Signalzustände an allen Ein- und Ausgängen der Flipflops für vier Tastenbetätigungen (Arbeits- und Ruhelage von S1 = 1 cm).
- Kontrollieren Sie, ob die Steuerschaltung ihre Funktion erfüllt, indem Sie in das Zeitablaufdiagramm auch die Schaltzustände der Lampen eintragen.

15. Was versteht man unter dem Toggle-Mode?

16. Zur Erzielung einer Flankensteuerung werden in integrierten Schaltkreisen die Signallaufzeiten genutzt. Die Schaltung zeigt ein einflankengesteuertes RS-Flipflop. Die Eingänge des Flipflops werden kurze Zeit nach dem Taktwechsel verriegelt. Danach ist eine Änderung der Signale am Setz- und Rücksetzeingang unwirksam. Diese Flipflops bezeichnet man daher als bistabile Elemente mit Data Lockout (Datensperre). Die Funktion der Signalsperre soll anhand eines Zeitablaufdiagrammes analysiert werden.



- a) Tragen Sie in ein Zeitablaufdiagramm (Zeitachse: 0 bis 500 ns, Maßstab: 1 cm \triangleq 20 ns,

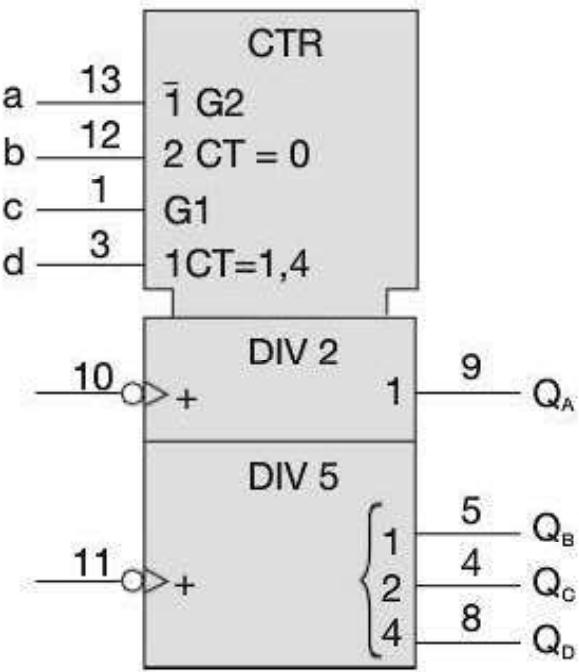
Zeichenblatt in Querlage) folgende Eingangssignale ein:

Eingang a: 1-Signal von $t = 10$ ns bis $t = 250$ ns

Eingang b: 1-Signal von $t = 150$ ns bis $t = 200$ ns
 $t = 300$ ns bis $t = 350$ ns

Eingang c: 1-Signal von $t = 40$ ns bis $t = 420$ ns

- b) Tragen Sie in das Zeitablaufdiagramm die Signale an den Schaltpunkten a' und b' , S' und R' , Q_1' und Q_2' sowie an Q_1 und Q_2 ein. Die Signallaufzeit für die Logikelemente D1 bis D8 beträgt $T_p = 10$ ns. Die Darstellung der Anstiegs- und Abfallzeiten der Signalflanken ist nicht erforderlich. (Lösungshinweis: Bei dem Flipflop D7 wird die zu vermeidende Signalkombination an den Eingängen genutzt. Für $S' = R' = 0$ gilt: $Q_1' = Q_2' = 1$.)
- c) Entnehmen Sie aus dem Diagramm die Zeitspanne, die vom Einschalten des Taktimpulses bis zum Verriegeln der Signaleingänge vergeht.
17. Geben Sie alle Betriebsmöglichkeiten für ein bidirektionales Schieberegister mit serieller und paralleler Ein- und Ausgabe an.
18. Das Schaltsymbol stellt einen vielseitig einsetzbaren Zählerbaustein dar, der zwei Zählerschaltkreise enthält.
- Geben Sie für die in der Funktionstabelle aufgeführten Steuersignale die Signalzustände an den Ausgängen Q_A bis Q_D an ($x = 0$ oder 1).
 - Welche Signalwerte müssen an den Steuereingängen beim Zählen anliegen? Geben Sie die Lösung als Funktionsgleichung an.
 - Geben Sie für beide Zählerschaltkreise den Zählbereich an.
 - Zeichnen Sie für den Zähler DIV 2 das Zeitablaufdiagramm für sechs Zählimpulse.
 - Zeichnen Sie in ein Zeitablaufdiagramm die Signale an den Ausgängen Q_B bis Q_D für 12 Zählimpulse.
 - Der Baustein soll als dekadischer Zähler eingesetzt werden. Wie muss er beschaltet werden und welche Wertigkeit haben die Zählerausgänge?
 - Der Zähler soll als Frequenzteiler mit dem Teilverhältnis 9:1 eingesetzt werden. ga) Geben Sie die Beschaltung des Bausteins an. gb) Zeichnen Sie zur Kontrolle der Beschaltung das Zeitablaufdiagramm für 12 Impulse.



| d | c | b | a |
|---|---|---|---|
| x | 0 | 1 | 1 |
| 1 | 1 | x | x |

19. Ein binärer Zweirichtungszähler (Bild 4.49) soll als programmierbarer Frequenzteiler eingesetzt werden. Hierzu wird er als Rückwärtszähler geschaltet und mit vier Schaltern (S_0 bis S_3) auf eine beliebige Zahl voreingestellt. Bei jedem Nulldurchgang wird die gewählte Zahl erneut geladen. Der BO-Ausgang wird als Teilarausgang genutzt.
- Zeichnen Sie den Schaltplan des Frequenzteilers.
 - Der Zähler wird mit der Zahl 5 geladen. Zeichnen Sie in ein Zeitablaufdiagramm die Signalzustände für die Anschlüsse DOWN, LOAD, BO, A_0 bis A_3 und Q_0 bis Q_3 für 14 Taktimpulse.
 - Entnehmen Sie aus dem Diagramm das Teilverhältnis.
 - Mit welcher Zahl muss der Zähler geladen werden, wenn ein Teilverhältnis von 12:1 erzielt werden soll?
20. Mit einem ADC soll eine analoge Spannung, die einen Bereich von $U = 0 \text{ V}$ bis $U_{\text{END}} = 1,28 \text{ V}$ umfasst, in 8-Bit-Codewörter (Dualcode) umgesetzt werden.
- Geben Sie für die folgenden Codewörter den Spannungsbereich der entsprechenden Quantisierungsstufe an:
 $Z_1 = 0000\ 0000; Z_2 = 0000\ 0001; Z_3 = 0011\ 1011; Z_4 = 0111\ 0101; Z_5 = 1000\ 0000; Z_6 = 1111\ 1111.$
 - Geben Sie die Codewörter der Quantisierungsstufen an, in denen die folgenden analogen Spannungswerte liegen: $U_1 = 2 \text{ mV}; U_2 = 153 \text{ mV}; U_3 = 368 \text{ mV}; U_4 = 996 \text{ mV}; U_5 = 1,101 \text{ V}; U_6 = 1,272 \text{ V}.$
 - Mit einem 8-Bit-DAC sollen die Codewörter wieder in Spannungswerte umgewandelt werden. Die Ausgangsspannung des Umsetzers entspricht jeweils dem mittleren Spannungswert der Quantisierungsstufe. Wie groß ist die maximale Abweichung der zurückgewonnenen Spannung von der ursprünglichen analogen Spannung?
 - Geben Sie für die analogen Spannungswerte $U_1 = 1 \text{ mV}$ und $U_2 = 1,266 \text{ V}$ jeweils das Codewort und den zurückgewonnenen Spannungswert an.
 - Wie groß ist in Aufgabe d) der absolute Spannungsfehler (d.h. die Differenz zwischen dem analogen und dem zurückgewonnenen Spannungswert) und der relative Spannungsfehler (d.h. der absolute Spannungsfehler, bezogen auf den absoluten Spannungswert)?
 - Welche Aussagen können über den absoluten und relativen Spannungsfehler bei sehr kleinen und sehr großen analogen Spannungen gemacht werden?