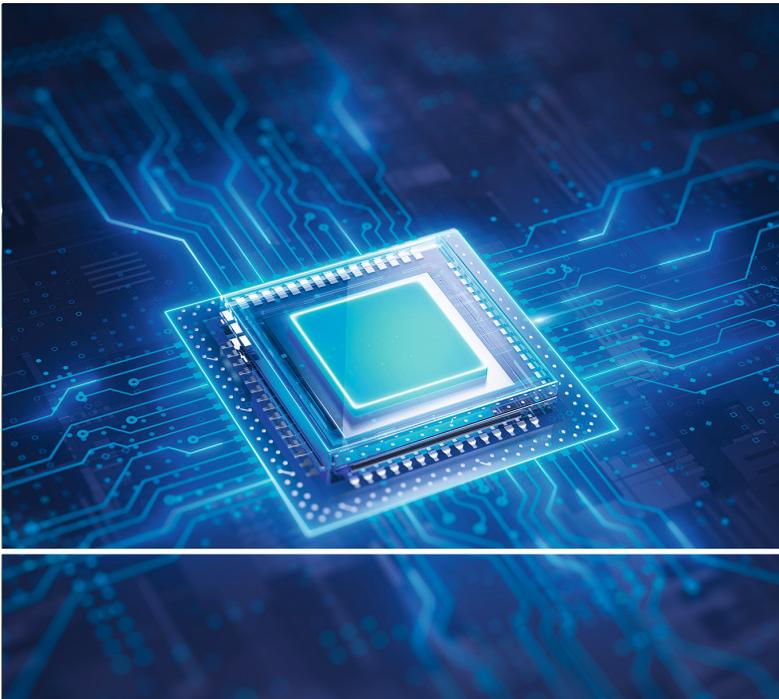


Gerd Wöstenkühler



Grundlagen der Digitaltechnik

Elementare Komponenten,
Funktionen und Steuerungen



3., überarbeitete und erweiterte Auflage

HANSER



Ihr Plus – digitale Zusatzinhalte!

Auf unserem Download-Portal finden Sie zu diesem Titel kostenloses Zusatzmaterial. Geben Sie dazu einfach diesen Code ein:

plus-s6qer-zgtwh

plus.hanser-fachbuch.de



Bleiben Sie auf dem Laufenden!

Hanser Newsletter informieren Sie regelmäßig über neue Bücher und Termine aus den verschiedenen Bereichen der Technik. Profitieren Sie auch von Gewinnspielen und exklusiven Leseproben. Gleich anmelden unter

www.hanser-fachbuch.de/newsletter

Gerd Wöstenkühler

Grundlagen der Digitaltechnik

Elementare Komponenten, Funktionen und
Steuerungen

3., überarbeitete und erweiterte Auflage

HANSER

Über den Autor:

Prof. Dr.-Ing. Gerd Wöstenkühler, Hochschule Harz, Fachbereich Automatisierung und Informatik



Print-ISBN: 978-3-446-47867-1

E-Book-ISBN: 978-3-446-47884-8

Alle in diesem Werk enthaltenen Informationen, Verfahren und Darstellungen wurden zum Zeitpunkt der Veröffentlichung nach bestem Wissen zusammengestellt. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grund sind die im vorliegenden Werk enthaltenen Informationen für Autor:innen, Herausgeber:innen und Verlag mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autor:innen, Herausgeber:innen und Verlag übernehmen infolgedessen keine Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Weise aus der Benutzung dieser Informationen – oder Teilen davon – entsteht. Ebenso wenig übernehmen Autor:innen, Herausgeber:innen und Verlag die Gewähr dafür, dass die beschriebenen Verfahren usw. frei von Schutzrechten Dritter sind. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt also auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benützt werden dürften.

Die endgültige Entscheidung über die Eignung der Informationen für die vorgesehene Verwendung in einer bestimmten Anwendung liegt in der alleinigen Verantwortung des Nutzers.

Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet unter <http://dnb.d-nb.de> abrufbar.

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung des Werkes, oder Teilen daraus, vorbehalten. Kein Teil des Werkes darf ohne schriftliche Einwilligung des Verlages in irgendeiner Form (Fotokopie, Mikrofilm oder einem anderen Verfahren), auch nicht für Zwecke der Unterrichtsgestaltung – mit Ausnahme der in den §§ 53, 54 UrhG genannten Sonderfälle –, reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

© 2024 Carl Hanser Verlag GmbH & Co. KG, München

www.hanserfachbuch.de

Lektorat: Frank Katzenmayer

Herstellung: le-tex publishing services GmbH, Leipzig

Coverkonzept: Marc Müller-Bremer, www.rebranding.de, München

Titelmotiv: © shutterstock/Sashkin

Satz: Eberl & Koesel Studio GmbH, Kempten

Druck: CPI Books GmbH, Leck

Printed in Germany

Vorwort zur dritten Auflage

Unsere Welt wird in zunehmendem Maße digitalisiert. Moderne Verfahren ermöglichen den Einsatz immer komplexerer Bausteine. Hiermit können vorhandene Realisierungen preiswerter erstellt, mit zusätzlichen Funktionen ausgestattet und/oder durch digitale Steuerungen ersetzt werden. Die Werkzeuge zur Erstellung dieser digitalen Funktionen und Steuerungen verwenden immer häufiger Abstraktionen. Die Kenntnisse der elementaren Grundschaltungen ermöglichen eine effektivere Nutzung der eingesetzten Tools.

Dieses Buch vermittelt die Grundlagen digitaler Funktionen und Steuerungen. Es ist nicht das Ziel, komplexe Entwurfswerkzeuge durch manuelle Synthese zu ersetzen, sondern die Kenntnisse über elementare Komponenten und Strukturen zu vermitteln, mit denen moderne Entwurfswerkzeuge effizienter eingesetzt werden können.

Die Inhalte des Buches basieren auf dem Lehrplan der Grundlagen-Vorlesungen „Digitaltechnik“ und „Digitale Systeme“, die ich viele Jahre an der Hochschule Harz in Wernigerode gehalten habe. Zielgruppe sind Studierende elektrotechnischer Bachelor-Studiengänge, aber auch interessierte Studierende anderer technischer Ausbildungsrichtungen. Das Verständnis digitaler Grundlagen wird durch eine schrittweise Vertiefung und umfangreiche Übungsmöglichkeiten vermittelt. Das Buch eignet sich neben der Vorlesungsbegleitung auch zur Prüfungsvorbereitung und zum Selbststudium.

Die dritte Auflage ist eine deutliche Erweiterung der zweiten Auflage. Wesentliche Punkte sind die Fehlererkennungs- und Korrekturverfahren mit Parity- und CRC-Prüfung, eine Erweiterung der Minimierung mit KV-Diagrammen für Schaltnetze mit schaltbarem Inverter am Ausgang (XOR-Verknüpfung), eine Erweiterung der Anwendungsbeispiele bis zur Arithmetisch-logischen Einheit (ALU) und die Unterstützung der Funktionsüberprüfung mithilfe der Simulation mit Logisim (bewährte kostenfreie Simulationssoftware). Für diese Logisim-Anwendungen steht auf *plus.hanser-fachbuch.de* Begleitmaterial zur Verfügung, der Zugangscode ist auf der ersten Seite des Buches abgedruckt.

Mein Dank geht an die Mitarbeiter des Hanser Verlages, Frau Mirja Werner, die dieses Werk initiiert hat, Herrn Dr. Martin Feuchte für die Betreuung während der Abschlussphase der ersten Auflage und Frau Franziska Jacob für die Betreuung sowie Frau Franziska Kaufmann für die technische Realisierung der zweiten Auflage. Den Anstoß zur dritten Auflage verdanke ich Herrn Frank Katzenmayer, der unterstützt von Frau Christina Kubiak verlagsseitig die Überarbeitung betreut hat. Auch bedanke ich mich bei meiner Frau für die Unterstützung bei der Korrektur des Manuskripts und ihr Verständnis, dass ich viel Zeit in dieses Buch investiert habe.

Paderborn, Januar 2024

Gerd Wöstenkühler

Inhalt

Vorwort zur dritten Auflage	V
1 Einleitung	1
1.1 Analoge und digitale Darstellungsformen	1
1.1.1 Analoge Größendarstellung	1
1.1.2 Digitale Größendarstellung	2
1.2 Binäre und logische Zustände	3
1.3 Zahlensysteme	5
1.4 Codierungen	11
1.4.1 Binär Codierte Dezimalziffer	11
1.4.2 (m aus n)-Code	13
1.4.3 Gray-Code	15
1.5 Betrachtung unabhängiger binärer Größen	17
1.6 Fehlererkennende und fehlerkorrigierende Verfahren	21
1.6.1 Fehlermodelle	21
1.6.2 Mehrfachübertragung	22
1.6.3 Parity-Ergänzung	23
1.6.4 Blockcheck	26
1.6.5 Fehlerkorrigierende Bewertung	28
1.6.6 Fehlerkorrigierender Code nach Hamming	29
1.6.7 CRC-Summe	30
1.7 Verarbeitungsgeschwindigkeit	34
1.8 Aufgaben	37

2	Logische Verknüpfungen	39
2.1	Grundfunktionen und Grundglieder	40
2.1.1	UND-Verknüpfung	40
2.1.2	ODER-Verknüpfung	42
2.1.3	NEGATION	43
2.1.4	Verstärker	44
2.2	Zusammengesetzte Elemente	45
2.2.1	NAND-Verknüpfung	45
2.2.2	NOR-Verknüpfung	46
2.2.3	ANTIVALENZ-Verknüpfung	47
2.2.4	ÄQUIVALENZ-Verknüpfung	49
2.2.5	Implikation	50
2.2.6	Inhibition	51
2.2.7	Verknüpfungen bei Gliedern mit zwei Eingängen	51
2.2.8	Wichtige Glieder mit mehr als zwei Eingängen	53
2.3	Schaltungsanalyse	55
2.4	Aufgaben	57
3	Schaltalgebra	59
3.1	Variable und Konstante	59
3.2	Rechenregeln	61
3.2.1	Postulate	61
3.2.2	Theoreme der Schaltalgebra	62
3.2.3	Kommutativgesetz	62
3.2.4	Assoziativgesetz	63
3.2.5	Distributivgesetz	64
3.2.6	De-Morgansche-Gesetze	64
3.2.7	Bindungsregel	65
3.3	Aufgaben	67
4	Schaltungssynthese	69
4.1	Normalformen	69
4.1.1	Disjunktive Normalform	70
4.1.2	Konjunktive Normalform	71

4.2	Schaltungsvereinfachung	72
4.2.1	Algebraische Vereinfachung	73
4.2.2	Grafische Vereinfachung	74
4.2.2.1	KV-Diagramm mit zwei Variablen	76
4.2.2.2	KV-Diagramm mit drei Variablen	77
4.2.2.3	KV-Diagramm mit vier Variablen	78
4.2.2.4	KV-Diagramm mit fünf Variablen	80
4.2.2.5	KV-Diagramm mit sechs und mehr Variablen	81
4.2.2.6	Don't-Care-Felder	83
4.2.2.7	Minimierung mit XOR-Gate	85
4.2.2.8	KV-Diagramm und Gray-Code	86
4.2.3	Algorithmische Umformung (Quine-McClusky)	87
4.3	Aufgaben	91
5	Schaltnetze	95
5.1	Darstellung	96
5.2	Codeumsetzer	98
5.3	Datenselektor	100
5.4	Demultiplexer	103
5.5	Adressdecoder	104
5.6	Addierer	108
5.6.1	Halbaddierer	108
5.6.2	Volladdierer	110
5.7	Komparator	112
5.8	Arithmetisch-logische Einheit	117
5.9	Aufgaben	119
6	Zeitabhängige binäre Schaltungen	123
6.1	Übersicht	123
6.2	RS-Flipflop (RS-FF)	128
6.2.1	Realisierung mit NOR-Gatter	128
6.2.2	Realisierung mit NAND-Gatter	132

6.3	Taktzustandsgesteuerte Flipflops	134
6.3.1	RS-FF (taktzustandsgesteuert)	134
6.3.2	D-FF (taktzustandsgesteuert)	137
6.3.3	JK-FF (taktzustandsgesteuert).	140
6.3.4	JK-MS-FF (taktzustandsgesteuert)	142
6.4	Taktflankengesteuerte Flipflops	144
6.4.1	RS-FF (einflankengesteuert)	145
6.4.2	D-FF (einflankengesteuert)	147
6.4.3	JK-FF (einflankengesteuert)	151
6.4.4	T-FF (einflankengesteuert)	151
6.4.5	RS-MS-FF (zweiflankengesteuert)	152
6.4.6	JK-MS-FF (zweiflankengesteuert)	153
6.5	Charakteristische Gleichung.	154
6.6	Synthese-Tabellen	156
6.7	Monostabile Kippstufen.	157
6.8	Taktgeneratoren	161
6.9	Aufgaben.	162
7	Einfache sequenzielle Schaltungen.	165
7.1	Zählerschaltungen	165
7.1.1	Asynchrone Zähler	166
7.1.2	Synchrone Zähler	172
7.2	Frequenzteiler	182
7.3	Schieberegister.	187
7.4	Aufgaben.	195
8	Getaktete Schaltwerke.	199
8.1	Einführung	199
8.2	Mealy-Automat	206
8.3	Moore-Automat.	215
8.4	Beispiel (ALU-Anwendung)	221
8.5	Aufgaben.	226

9	Hardware	231
9.1	Einführung	231
9.2	Schaltkreisentwicklung	232
9.3	Physikalische Randbedingungen	238
9.3.1	Spannungsbereiche	238
9.3.2	Strombereiche	239
9.3.3	Geschwindigkeitsbereiche	242
9.3.4	Gehäuse	245
9.4	Spezielle Ausgänge	248
9.4.1	Tri-State-Ausgang	249
9.4.2	Open-Kollektor-Ausgang	251
9.5	Spezielle Eingänge	254
9.6	Strukturen komplexer Bausteine	259
10	Musterlösungen	267
10.1	Lösungen zu Kapitel 1	267
10.2	Lösungen zu Kapitel 2	272
10.3	Lösungen zu Kapitel 3	276
10.4	Lösungen zu Kapitel 4	280
10.5	Lösungen zu Kapitel 5	291
10.6	Lösungen zu Kapitel 6	299
10.7	Lösungen zu Kapitel 7	304
10.8	Lösungen zu Kapitel 8	314
	Verzeichnis von Formelzeichen und Abkürzungen	325
	Literatur	333
	Index	335

Für die Logisim-Anwendungen steht auf plus.hanser-fachbuch.de Begleitmaterial und ein Anhang zur Verfügung. Der Zugangscode ist auf der ersten Seite des Buches abgedruckt.

1

Einleitung

Digitaltechnische Realisierungen sind in der heutigen Zeit nicht mehr wegzudenken. Man ist es gewohnt, Handy und Digitalfernseher zu verwenden. Die Funktionsweise ist den meisten Benutzern nicht bekannt und wird für die Benutzung digital arbeitender Geräte auch nicht benötigt. Technikinteressierte möchten jedoch mehr über die Funktionsweise wissen. Hierzu könnte man ein komplexes digitaltechnisches Gerät in seine Funktionsgruppen und weiter in seine Einzelkomponenten auflösen und deren Wirkungsweise vertiefen (**Top-Down Design**). Anschließend weiß man, wie dieses Gerät funktioniert, aber die Anpassung der Signalverarbeitung für weitere Anwendungen wurde nicht trainiert. Dieses Buch verwendet die andere Entwurfsrichtung. Basierend auf den Grundelementen wird deren Zusammenschaltung zu einfachen Funktionsgruppen beschrieben und erläutert, sodass hieraus komplexere Geräte oder deren Komponenten für verschiedene Anwendungen erstellt werden können (**Bottom-Up Design**). Dieses Buch beschreibt die rudimentären Elemente und grundlegenden Funktionsgruppen der Digitaltechnik. In der Einleitung werden Begriffe erläutert.

■ 1.1 Analoge und digitale Darstellungsformen

1.1.1 Analoge Größendarstellung

Viele uns bekannte Größen sind analoger Natur. Hierunter fallen u. a. Abstand, Temperatur, elektrische Spannung etc. Zur Beschreibung dieser Größen wurden analoge Darstellungen verwendet, wie Länge an einem Referenzmaßstab (Metermaß), Flüssigkeitsthermometer, Zeigerinstrumente etc. Die verwendeten Wertangaben in diesem Bereich gehören zu den reellen Zahlen \mathbb{R} . Die Zuordnung zwischen Messwert (z. B. elektrische Spannung) und Darstellungswert (z. B. Winkel der Zeigerposition) ist eine kontinuierliche Funktion. Bild 1.1 zeigt beispielhaft

den Zusammenhang einer analogen Abbildungsfunktion. Theoretisch könnte bei unendlicher Genauigkeit vom Anzeigewert auf den exakten verursachenden Messwert zurückgeschlossen werden. Grenzen sind durch die Ablesegenauigkeit gegeben, die häufig bei der dritten Stelle aufhört. Bei Mess- und Übertragungssystemen gibt es noch den unteren und oberen Grenzwert, die den Wertebereich der Anordnung vorgeben. Vorteil der analogen Darstellung ist u. a. die schnelle Erfassung sowie die Erkennung von Tendenzwerten (z. B. bei analoganzeigenden Messgeräten). Dieser Vorteil wird häufig bei der analogen Ausgabe digitaler Größen verwendet, wie z. B. bei der analoganzeigenden Digitaluhr.

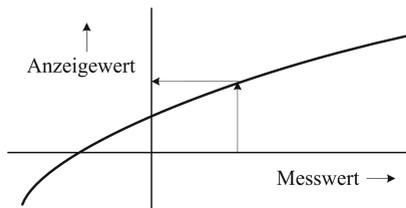


Bild 1.1

Abbildungsfunktion analog \rightarrow analog

1.1.2 Digitale Größendarstellung

Die Bezeichnung „digital“ kommt aus dem Lateinischen (lat.: digitus: Finger (oder Zehe), zählen mit den Fingern). Sie beschreibt eine diskrete Größendarstellung, also eine abzählbare Menge. Einige Größen sind zählbar (z. B. Anzahl von Eiern (Grundmenge ein Ei)), andere werden durch Quantisierung analoger Größen in zählbare digitale Einheiten unterteilt (z. B. Gewichtsklassen von Eiern (Klassen S, M, L oder XL)). Verwendet man sehr viele Klassen oder Stufen (z. B. bei Digitalwaagen), so erhält man bei vernachlässigbarer Klassenbreite bzw. Stufenhöhe einen nahezu kontinuierlichen Verlauf (ähnlich der analogen Abbildung). Bild 1.2 zeigt den Zusammenhang einer digitalen Abbildungsfunktion. Bestimmte Wertebereiche werden einem Digitalwert (einer Stufe) zugeordnet. Die Werteangaben vom Messwert gehören zu den reellen Zahlen \mathbb{R} (kontinuierliche, analoge Größe) und die Angaben vom Anzeigewert zu den ganzen Zahlen \mathbb{Z} (diskrete, digitale Größe). Eine exakte Rekonstruktion des verursachenden Analogwertes ist hierbei prinzipiell nicht möglich, da ein Digitalwert immer einen Bereich der analogen Eingangsgröße abdeckt. Bei der Rekonstruktion wird normalerweise der Mittelwert des Wertebereiches der Stufe als Näherungswert verwendet. Ein anschaulicher Vergleich von analogen und digitalen Größen ist eine Treppe mit einer mitgeführten Fahrradrampe. Die Rampe gibt einen analogen Wert der Höhendifferenz wieder und die Stufen einen digitalen. Zu jeder digitalen Stufe gehört ein Bereich der zugehörigen analogen Höhendifferenz.

Eine häufig zu findende Darstellung entspricht Bild 1.3. Der Messwert in dem Bereich von X_{\min} bis X_{\max} wird in N äquidistante (gleichgroße) Bereiche mit der Breite Q unterteilt. Das **Quant** Q ist normalerweise $Q = (X_{\max} - X_{\min})/N$.

Der Digitalwert wird durch eine Zahl repräsentiert (Zahlendarstellungen werden im nächsten Unterkapitel erläutert), in der digitalen Welt ist dies häufig eine Dualzahl. Die Dualzahl besteht aus J binären Stellen, wodurch sich eine maximal verwendbare Anzahl von $N = 2^J$ Stufen ergibt. In Bild 1.3 ist dieser Zusammenhang mit einem 3-stelligen Dualcode dargestellt.

Durch die ziffernmäßige Darstellung des Digitalwertes ist eine eindeutige Lesbarkeit gegeben (keine Ableserunterschiede zwischen erster und letzter Stelle). Die Auflösung ergibt sich durch die Quantisierungsstufe Q .

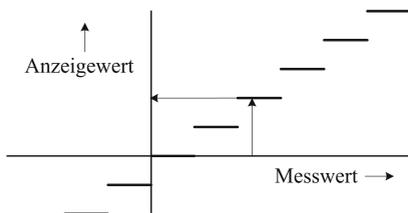
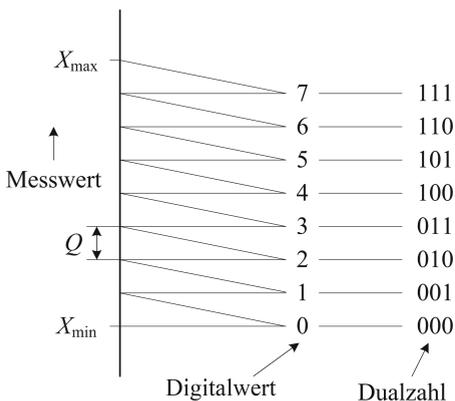
**Bild 1.2**Abbildungsfunktion analog \rightarrow digital**Bild 1.3**

Abbildung einer Messgröße in ein digitales Signal bzw. einen digitalen Signalwert

■ 1.2 Binäre und logische Zustände

Eine digitale Größe besteht aus abzählbaren Elementen (Stufen). Ein digitaler Spannungsverlauf ist somit eine Folge von diskreten Spannungswerten (theoretisch kein kontinuierlicher Übergang). Analoge Spannungswerte repräsentieren hierbei diskrete Werte. Die Anzahl der Stufen ist beliebig, jedoch begrenzt durch die Unterscheidungsmöglichkeit (begrenzte Genauigkeit bei der Spannungsmes-

sung). Bild 1.4 zeigt einen digitalen Spannungsverlauf mit drei Zuständen (0 V, 5 V und 10 V). Problematisch sind bei entsprechender Verwendung die Übergänge zwischen 0 V und 10 V und zwischen 10 V und 0 V. Aufgrund des analogen Verhaltens von Spannungswerten kann dieser Übergang nicht in der Zeit $t=0$ erfolgen. Die Zeit ist ebenfalls eine analoge Größe und dadurch kann nur ein quasi digitales Signal realisiert werden.

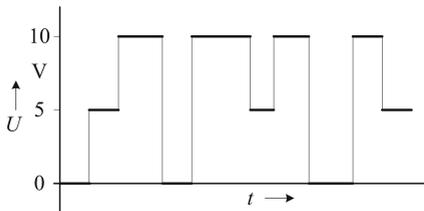


Bild 1.4 Digitales Spannungssignal mit drei Zuständen

Es wird für den Übergang immer ein gewisses, wenn auch sehr kurzes Zeitintervall benötigt. Bei schnellen Signalfolgen (häufiger Wechsel zwischen den Spannungswerten, Zeitspanne für Spannungsänderungen nicht mehr vernachlässigbar) ist der Übergang über die 5 V-Stufe problematisch. Bild 1.5 zeigt einen analogen Übergang zwischen 0 V und 10 V mit der ungewünschten Übergangszeit t_x . Der für die 5 V-Stufe verwendete Spannungsbereich (Messgenauigkeit dieser Stufe etc.) wird durchlaufen und dies könnte zur Detektion dieser nicht im eigentlichen Spannungsverlauf angezeigten Spannungsstufe führen.

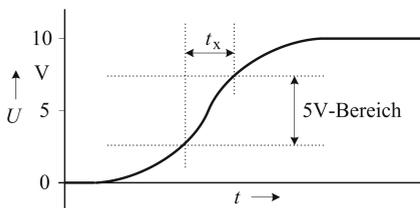


Bild 1.5 Analoges Übergang eines Spannungswechsels von 0 V auf 10 V

Ein digitales Signal mit mehr als zwei Zuständen ist deshalb für einfache Realisierungen nicht geeignet. Zwei Spannungsbereiche lassen sich durch einen Übergang (einen Schwellwert) unterscheiden. Ein zweiwertiges Signal hat zwei mögliche Zustände und wird deswegen binäres Signal genannt (lat.: binär: aus zwei Einheiten bestehend). Die allgemein angewandte Digitaltechnik arbeitet mit binären Signalen.

Vorteil dieser Signale ist, dass sich die zwei Bereiche leicht unterscheiden und, was häufig vorkommt, zur Vermeidung von Störeinflüssen vergleichsweise gut verstärken lassen (ein durch Störeinflüsse „verschwommenes“ Signal wird rekonstruiert, s. auch Abschnitt 9.3.1 Spannungsbereiche und in Abschnitt 9.5 Spezielle Ein-

gänge, Schmitt-Trigger-Eingang). Der negativere Bereich wird mit L (Low) und der positivere Bereich mit H (High) gekennzeichnet. Diesen Bereichen müssen logische Zustände zugeordnet werden. Bei positiven Betriebsspannungen entspricht normalerweise H der logischen 1 (wahr, zutreffend oder ja) und L der logischen 0 (falsch, nicht zutreffend oder nein). Es existiert auch die umgedrehte Zuordnung, die in der Regel mit negativer Betriebsspannung gekoppelt ist. Sie wird als **negative Logik** ($0\triangleq H, 1\triangleq L$) bezeichnet. Diese Zuordnung ist bei den derzeitigen Realisierungen eigentlich nicht mehr anzutreffen und man nimmt stillschweigend die **positive Logik** ($1\triangleq H, 0\triangleq L$) an. Bei der Diskussion logischer Verknüpfungen werden 0 und 1 und bei der hardwaremäßigen Realisierung mit Logikelementen die Bezeichnungen L und H verwendet.



Binäre Systeme sind in fast allen technischen Geräten vorhanden.

Vorteile

- Verarbeitet werden nur zwei Werte (0 und 1 bzw. L und H)
- Impulse (Rechteckimpulse) sind einfach und vollständig regenerierbar
- Einfache Verstärkung (Schalter, Relais, Transistoren im Schaltbetrieb)
- Verknüpfen, verteilen und sortieren von Informationen mittels logischer Schaltungen
- Speicher einfach realisierbar
- Mathematische Probleme werden auf Addition zurückgeführt und gelöst
- Verschiedene mathematische Probleme können mit derselben Schaltung gelöst werden
- Genauigkeit kann durch Vergrößern der Stellenzahl nahezu beliebig erhöht werden

Nachteile

- Prinzipieller Fehler durch Rasterung (Quantisierungsstufe Q bestimmt Fehlergröße)

■ 1.3 Zahlensysteme

Zahlendarstellungen

Im Laufe der Zeit haben sich verschiedene, teilweise anwendungsspezifische Zahlensysteme entwickelt, von denen hier einige exemplarisch vorgestellt werden.

In der **Abzähl Schreibweise** werden gleichwertige Striche verwendet, wie es von Kaffeelisten oder Bierdeckeln bekannt ist. Zur Übersicht gibt es die Bündelung, bei der nach vier Strichen der fünfte über diese vier gezeichnet wird und diese Gruppe (dieses Symbol) eine Fünf repräsentiert. Die Position der Striche ist ohne Bedeutung. Diese Darstellung verwendet den Zahlenraum \mathbb{N} (Natürliche Zahlen) und ist nur für kleine Mengen einsetzbar (nicht bis unendlich).

Das **Römische Zahlensystem** deckt ebenfalls nur einen begrenzten positiven Ganzzahlenbereich aus \mathbb{N} ab und verwendet die Zweier- und Fünfer-Bündelung wie in Tabelle 1.1 angegeben.

Tabelle 1.1 Bündelungen des Römischen Zahlensystems

5	mal	I	wird durch	V	dargestellt	(= 5)
2	mal	V	wird durch	X	dargestellt	(= 10)
5	mal	X	wird durch	L	dargestellt	(= 50)
2	mal	L	wird durch	C	dargestellt	(= 100)
5	mal	C	wird durch	D	dargestellt	(= 500)
2	mal	D	wird durch	M	dargestellt	(= 1000)

Außerdem wird die Stellenschreibweise verwendet. Hierdurch entscheidet die Position über den Wert der Zahl. Die Zahl Vier wird z. B. nicht durch IIII (wie es auf einigen analoganzeigenden Uhren zu finden ist) sondern durch IV ($5 - 1$) dargestellt und unterscheidet sich aufgrund der Stellenschreibweise von der Sechs dargestellt durch VI ($5 + 1$). Die 1 (Symbol I) wird in der linken Position abgezogen und in der rechten Position hinzuaddiert. Das Zahlensystem ist unhandlich und eignet sich nicht für Rechenoperationen.

Bei der **Stellenschreibweise** ergibt sich der Wert nicht nur durch die verwendeten Symbole wie bei den Römischen Zahlen, sondern durch die Position innerhalb der Darstellung. Das Bildungsgesetz zeigt Formel 1.1.

$$N = \left(\sum_{j=0}^{j-1} n_j \cdot W_j \right) \quad (1.1)$$

Hierbei hat jede Stelle j einen Wert W_j . Bei der Angabe 01:02:03:04 muss der Wert jeder Stelle bekannt sein. Hier könnte es Tag:Stunde:Minute:Sekunde sein. Der Wertebereich der linken Stelle mit dem Wert 1 ist der Tag, der mit dem Faktor 24 auf eine Stundenzahl umgerechnet werden kann. Der Wert 2 (Werte von 0 bis 23 sind an dieser Stelle möglich) kann mit dem Faktor 60 auf Minuten umgerechnet werden. Der Wert 3 (Werte von 0 bis 59 sind an dieser Stelle möglich) kann mit dem Faktor 60 auf Sekunden umgerechnet werden. Der angegebene Wert umgerechnet in die kleinste Einheit, in diesem Fall die Sekunde, ergibt:

$$N = (1 \cdot (24 \cdot 60 \cdot 60) + 2 \cdot (60 \cdot 60) + 3 \cdot (60) + 4 \cdot (1)) = 93784 \quad (1.2)$$

Rechenoperationen mit der Stellenschreibweise sind aufgrund der Umrechnungsfaktoren, die z. B. bei der Umrechnung von Monaten in Tage vom jeweiligen Monat abhängen, ungünstig. Innerhalb der jeweiligen Stelle wird der Wert in dem genannten Beispiel bereits mit dem Polyadischen Zahlensystem beschrieben.

Das Bildungsgesetz des **Polyadischen Zahlensystems** für positive Ganzzahlen \mathbb{N}_0 (es existiert auch ein Symbol für die 0) zeigt Formel 1.3. Hierbei ergibt sich die Wertigkeit einer jeweiligen Stelle aus dem Basiswert B mit der Stellenzahl j als Exponent.

$$N = \left(\sum_{j=0}^{J-1} n_j \cdot B^j \right) \quad (1.3)$$

In dem normalerweise verwendeten **Zehnersystem** ist $B=10$ und n ein Element der Menge $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Die Zahl 4711 ist eine 4-stellige Zahl ($J=4$). Die Darstellung des Wertes ist trivial und soll hier zur Verdeutlichung des Rechenweges verwendet werden.

$$N = (4 \cdot 10^3 + 7 \cdot 10^2 + 1 \cdot 10^1 + 1 \cdot 10^0) = 4711 \quad (1.4)$$

Diese Zahlendarstellung kann für verschiedene Basiswerte verwendet werden. In der binären Darstellung erhält man das **Dualsystem**. In diesem Fall ist $B=2$ mit $n \in \{0, 1\}$. Die Dualzahl 1100101_2 (Index 2 kennzeichnet das 2er-System) hat folgenden Wert:

$$N = (1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0) = 101_{10} \quad (1.5)$$

Nach dem gleichen System ist das **Oktalsystem** ($B=8$, $n \in \{0, 1, 2, 3, 4, 5, 6, 7\}$) aufgebaut. Im Vergleich zu dem Dualsystem werden jeweils 3 Bit (Bit \triangleq kleinste Einheit) zusammengefasst. Die Dualzahl 1100101_2 wird mit führenden Nullen auf ein Vielfaches von drei Stellen erweitert zu $001\ 100\ 101$ und jede Dreiergruppe durch das zugehörige Symbol repräsentiert gemäß 145_8 oder 00145 (Null O(ktal) 145).

Gebräuchlicher ist das **Sedezimalsystem** oder **Hexadezimalsystem**. Hierbei ist $B=16$ und $n \in \{0, 1, 2, \dots, 8, 9, A, B, C, D, E, F\}$. Da 16 Zeichen benötigt werden, ergänzt man die gebräuchlichen Ziffern um die ersten sechs Buchstaben des Alphabets. Im Vergleich zu dem Dualsystem werden hier vier Stellen zu einer Hexadezimalziffer zusammengefasst. Die Dualzahl 1100101_2 wird wieder um führende Nullen erweitert zu $0110\ 0101$ und jede Vierergruppe durch das zugehörige Symbol repräsentiert gemäß 65_{16} oder $0X65$ (Hexadezimal 65).

Im Weiteren werden diese Zahlensysteme nicht zum Rechnen verwendet. Sie dienen gegebenenfalls zur verkürzten Schreibweise verwendeter Bitkombinationen.

Werden diese Zahlen übertragen, so muss der Empfänger wissen, in welcher Reihenfolge die Ziffern übertragen werden. Das höchstwertige Bit (in der Regel links) ist das MSB (**M**ost **S**ignificant **B**it) und das niederwertigste Bit (in der Regel rechts) ist das LSB (**L**east **S**ignificant **B**it).

Die Zahlendarstellung kann auch für Nachkommastellen erweitert werden. Eine 5-stellige Zahl mit zwei Vor- und drei Nachkommastellen hat das in Formel 1.6 angegebene Bildungsgesetz. Ein Beispiel im Dualcode zeigt Bild 1.6. Jede Nachkommastelle reduziert hierbei die Quantisierungsstufe Q auf die Hälfte. Bei drei Nachkommastellen ist dies $2^{-3} = 0,125$ bzw. $\pm 0,0625$.

$$N = \left(\sum_{n=-3}^1 n_j \cdot B^j \right) \quad (1.6)$$

Wertigkeit $\left| \begin{array}{c|c|c|c|c} 2^1 & 2^0 & 2^{-1} & 2^{-2} & 2^{-3} \\ \hline 1 & 0,1 & 0,05 & 0,025 & 0,0125 \end{array} \right| = 1 \cdot 2 + 0 \cdot 1 + 1 \cdot 0,5 + 0 \cdot 0,25 + 1 \cdot 0,125 = 2,625$

Bild 1.6 Beispiel einer Dualzahl mit Nachkommastellen

Im **Polyadischen Zahlensystem** können Addition und Subtraktion (Bild 1.7), Multiplikation (Bild 1.8) und Division (Bild 1.9) in gewohnter Weise durchgeführt werden. Die Rechenoperationen mit den Dualwerten (mittlerer Teil) und Dezimalwerten (rechter Teil) werden im Weiteren nicht verwendet. Die Entwicklung einer Schaltung zur Realisierung des in der Tabelle angegebenen Zusammenhangs (linker Teil) jedoch schon.

b	a	$(b+a)$				b	a	$(b-a)$			
0	0	0 0		0 1 0 0	4	0	0	0 0		1 0 1 0	10
0	1	0 1	+ 0 1 1 0	+ 6		0	1	1 1	- 0 1 1 0	- 6	
1	0	0 1	carry \rightarrow 1			1	0	0 1	borrow \rightarrow 1		
1	1	1 0	1 0 1 0	10		1	1	0 0	0 1 0 0	4	

Bild 1.7 Beispiele für Addition und Subtraktion im Dual- und Dezimalsystem

b	a	$b \cdot a$					
0	0	0	1 1 · 0 1 0 1	3 · 0 5			
0	1	0	0 0 ↓ ↓ ↓ ↓	0 ↓			
0	1	0	+ 1 1 ↓ ↓ ↓ ↓	+ 1 5			
1	0	0	+ 0 0 ↓ ↓ ↓ ↓	1 5			
1	1	1	+ 1 1 ↓ ↓ ↓ ↓	1 5			
			0 1 1 1 1				

Bild 1.8 Beispiel für die Multiplikation im Dual- und Dezimalsystem

$$\begin{array}{r} 1111 : 11 = 101 \\ \underline{-11} \\ 01 \\ \underline{-0} \\ 11 \\ \underline{-11} \\ 0 \end{array}$$

$$\begin{array}{r} 15 : 3 = 05 \\ \underline{-0} \\ 15 \\ \underline{-15} \\ 0 \end{array}$$

Bild 1.9

Beispiele für die Division im Dual- und Dezimalsystem

Erweitert man die Darstellung von Natürlichen Zahlen um den negativen Zahlenbereich auf Ganzzahlen (von \mathbb{N}_0 nach \mathbb{Z}), so wird ein Bit für das Vorzeichen benötigt. Dies ist das MSB (Bit mit der höchsten Wertigkeit, bisher 2^{J-1} ; Wertebereich der Zahl von $0, \dots, 2^{J-1}$). Mit den negativen Zahlen wird nun der Zahlenraum von $\pm 2^{J-1} - 1$ abgedeckt, bei vier Bit von -7 bis $+7$. Eine einfache Darstellung besteht aus Vorzeichen und Betrag wie in Tabelle 1.2 in der 3. Spalte zu sehen ist. Diese Darstellung eignet sich für eine Ausgabe eines Wertes mit Vorzeichen (z. B. für die Anzeige bei Messinstrumenten), für Berechnungen ist sie jedoch nicht geeignet.

Die nächste Variante ist das $(b-1)$ -Komplement, auch Einerkomplement genannt. Um aus einem positiven Wert einen negativen zu erhalten, müssen hierbei alle Bits invertiert und das Vorzeichenbit hinzugefügt werden. Aus $+3_{10}$ wird im Dualsystem mit Vorzeichenbit und drei Werte-Bits $0\ 011_2$ für die Zahlendarstellung der positiven 3 aus dem Bereich 0 bis 7. Eine -3_{10} wird mit $1\ 100_2$ dargestellt (die inverse Bitkombination der positiven Zahl). Das Vorzeichenbit ist bei der Betragsdarstellung und dem $(b-1)$ -Komplement gleich, nur die Datenbits sind invertiert (s. Tabelle 1.2 in der 4. Spalte). Eine Umwandlung zwischen diesen beiden Varianten ist dadurch recht einfach. Nachteilig ist, dass in beiden Darstellungsvarianten die Null zweimal existiert.

Für mathematische Berechnungen gibt es eine günstigere Variante, das (b) -Komplement, auch Zweierkomplement genannt. Hierbei wird der negative Zahlenbereich um einen Wert in negativer Richtung verschoben und man erhält den Wertebereich bei vier Bit von $-8, \dots, -1, 0, +1, \dots, +7$ wie in Tabelle 1.2 in der rechten Spalte dargestellt. Wertebereiche typischer Bitlängen zeigt Tabelle 1.3. Bei dieser Zahlendarstellung kann dem Vorzeichenbit wieder ein Wert zugeordnet werden, dieser ist $-(2^{J-1})$. Die Wertigkeiten der Bitstellen bei vier Bit im Dualcode für positive (natürliche) Zahlen von 8, 4, 2, 1 geht für positive und negative Ganzzahlen über in $-8, 4, 2, 1$. Die realisierbaren Werte lassen sich recht gut als Kreis darstellen, um Rechenoperationen wie Addition oder Subtraktion zu erläutern.

Tabelle 1.2 Gegenüberstellung verschiedener Zahlendarstellungen

Wert	Dualzahl	Vorzeichen und Betrag	(b-1)-Komplement 1er-Komplement	(b)-Komplement 2er-Komplement
8	./.	./.	./.	./.
7	111	0 111	0 111	0 111
6	110	0 110	0 110	0 110
...
...
1	001	0 001	0 001	0 001
0	000	0 000	0 000	0 000
-0	./.	1 000	1 111	./.
-1	./.	1 001	1 110	1 111
...
...
-6	./.	1 110	1 001	1 010
-7	./.	1 111	1 000	1 001
-8	./.	./.	./.	1 000
-9	./.	./.	./.	./.

Tabelle 1.3 Wertebereiche typischer Ganzzahldarstellungen

Bitanzahl (j)	Vorzeichenloser Wertebereich (unsigned)	Vorzeichenbehafteter Wertebereich (signed, (b)-Komplement)
4	0, +1, +2, ..., +14, +15	-8, ..., -1, +0, +1, ..., +7
8	0, +1, +2, ..., +254, +255	-128, ..., -1, +0, +1, ..., +127
12	0, +1, +2, ..., +4094, +4095	-2048, ..., -1, +0, +1, ..., +2047
16	0, +1, +2, ..., +65 534, +65 535	-32 768, ..., -1, +0, +1, ..., +32 767

Bild 1.10 stellt die Periodizität der Integerzahlen ohne und mit Vorzeichen dar. Wird durch Addition (Anzahl der Schritte in Richtung +, auch bei der Subtraktion einer negativen Zahl) oder Subtraktion (Anzahl der Schritte in Richtung -, auch bei der Addition einer negativen Zahl) die Bereichsgrenze überschritten, entstehen Fehler (inhärenter (lat.: inhaerens: hängen, kleben, auch: innewohnend, ureigen) Modulo- (Rest einer Ganzzahldivision, wird häufig zur Begrenzung eines Wertebereiches verwendet) Effekt (manchmal ist dieser Effekt auch gewollt). Bei Berechnungen sollte dies überwacht werden. Bild 1.11 zeigt als Beispiel einige Rechenoperationen.

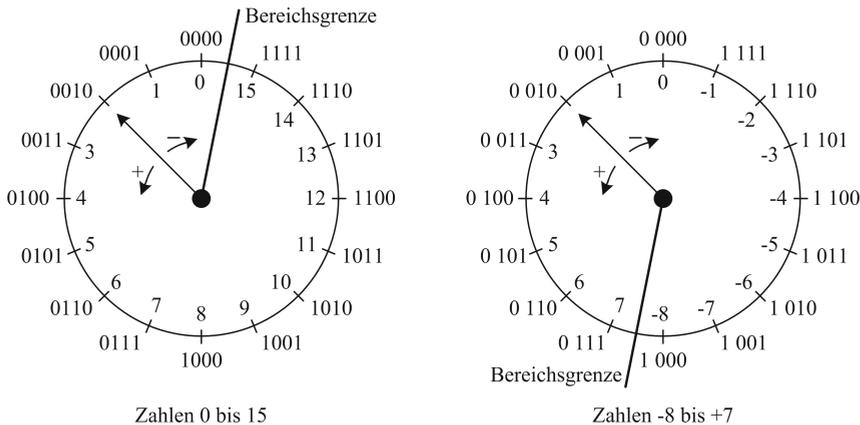


Bild 1.10 Integer Zahlen in umlaufender Darstellung

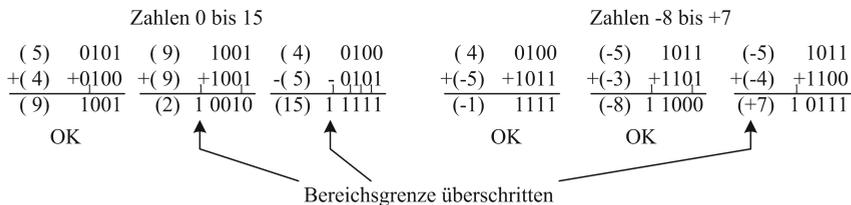


Bild 1.11 Rechenbeispiele mit 4 Bit Integer (unsigned und signed)

■ 1.4 Codierungen

Um die Wertigkeit einer digitalen Bitkombination zu kennzeichnen, können neben den Zahlensystemen auch spezielle Codierungen verwendet werden. In diesem Abschnitt werden gebräuchliche Codierungen und einige damit verbundene Begriffe vorgestellt.

1.4.1 Binär Codierte Dezimalziffer

Für die Darstellung einer mehrstelligen dezimalen Zahl kann jede Ziffer durch einen Dualwert repräsentiert werden. Hierdurch ist die benötigte Hardware zur Umcodierung der übertragenen Bitkombination für alle Stellen gleich. Die Codierung wird **BCD-Code** (Binär Codierte Dezimalziffer) genannt.

Der Code besteht aus vier binären Stellen (4-stellig). Eine Gruppe aus vier Stellen wird **Tetrade**, **Halbbyte** oder **Nibble** genannt. Jeder Spalte kann ein Wert zugeordnet werden, sodass sich der Beitrag des Eintrags in der entsprechenden Spalte aus dem Wert des Symbols (hier 0 oder 1) multipliziert mit dem Wert der Spalte (hier 8, 4, 2, oder 1) ergibt. Aufgrund der Wertigkeiten wird dieser Code auch 8-4-2-1-Code genannt (s. Tabelle 1.4). Unter dem Begriff **Gewicht** versteht man die Anzahl der mit 1 besetzten Stellen. In diesem Fall beträgt das Gewicht je nach Ziffer 0 bis 3 für die jeweils betrachtete Bitkombination. Unter der **Distanz** versteht man die Anzahl der unterschiedlichen Bitstellen verschiedener Codewörter. In diesem Beispiel haben wir die geringste Distanz mit dem Wert 1 z. B. zwischen den Codierungen von 0 und 1, 0 und 2, 0 und 4 als auch zwischen 2 und 6. Die größte Differenz haben wir zwischen den Codierungen 7 und 8, bei der sich alle vier Stellen unterscheiden und wir deshalb hier die Distanz 4 haben. Unter der **Hamming-Distanz** versteht man den kleinsten Distanzwert. Die Hamming-Distanz ist ein Maß für die Erkennbarkeit von Fehlern und deren Korrigierbarkeit. Ein weiterer Begriff ist **stetig**. Die Codierung ist **stetig**, wenn benachbarte Codewörter sich immer um die gleiche Anzahl an Bitstellen unterscheiden. In dem BCD-Code ist das nicht der Fall (Distanz 1 zwischen den Codierungen 0 und 1 und die Distanz 4 zwischen den Codierungen 7 und 8). Die **Redundanz** R ist ein Maß, das die Menge nicht benötigter Kombinationen beschreibt. Die Wahrscheinlichkeit des Auftretens der verwendeten Bitkombinationen wird hierbei als gleichverteilt angenommen. $N_{\text{möglich}}$ ist die Anzahl aller Kombinationsmöglichkeiten (bei 4 Bitstellen sind es 16 Kombinationsmöglichkeiten), N_{genutzt} (in unserem Fall 10 Kombinationsmöglichkeiten) für die 10 Zeichen 0 bis 9 und ld (ist der Logarithmus zur Basis 2). Die Logarithmusfunktion ld ist auf den üblichen Rechnern nicht vorhanden und kann über den Zusammenhang $\text{ld}(x) = \log(x)/\log(2) = \ln(x)/\ln(2)$ entsprechend Formel 1.7 bestimmt werden.

$$R = \text{ld}(N_{\text{möglich}}) - \text{ld}(N_{\text{genutzt}}) = \frac{\log(N_{\text{möglich}})}{\log(2)} - \frac{\log(N_{\text{genutzt}})}{\log(2)} = \frac{\ln(N_{\text{möglich}})}{\ln(2)} - \frac{\ln(N_{\text{genutzt}})}{\ln(2)} \quad (1.7)$$

Für den BCD-Code ist $R = 4 - 3,322 = 0,678$ und kennzeichnet die Menge nicht benötigter Informationseinheiten. Da jedoch nur ganze Bitstellen verwendet werden, ist dieses Übermaß hinzunehmen. Die nicht verwendeten Bitkombinationen nennt man **Pseudotetraden**.

Tabelle 1.4 BCD-Code

Ziffer	Wertigkeit			
	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
-	1	0	1	0
-	1	0	1	1
-	1	1	0	0
-	1	1	0	1
-	1	1	1	0
-	1	1	1	1

1.4.2 (m aus n)-Code

Als Grundlage der Zahlenzuordnung muss nicht immer der Dualcode verwendet werden. Zwei Beispiele sind in Tabelle 1.5 dargestellt. Eines davon ist der (2 aus 5)-Code. Jedes verwendete Bitmuster hat das Gewicht 2, wodurch sich die Möglichkeit der Fehlererkennung (wird in Abschnitt 1.6 vertieft) ergibt. Jeder Einbitfehler führt zu einem nicht verwendeten Bitmuster (Pseudowert). Aufgrund der erhöhten Anzahl an Pseudowerten ist die Wahrscheinlichkeit, bei Mehrbitfehlern einen Pseudowert zu erhalten, deutlich höher als die Entstehung einer verwendeten Bitkombination. Als Vorteil dieser Codierung könnte die Wertigkeit der Bitstellen angegeben werden (ein Teil der Tabelle enthält den Dualcode), die bis auf die linke Spalte für das Bitmuster 11000 nicht zutrifft und als Ausnahme angesehen wird.

Tabelle 1.5 (m aus n)-Code, Beispiele zur Darstellung der Werte 0 bis 9

Ziffer	(2 aus 5)-Code Wertigkeit (7) 4 2 1 0	(1 aus 10)-Code Wertigkeit 9 8 7 6 5 4 3 2 1 0
0	1 1 0 0 0	0 0 0 0 0 0 0 0 0 1
1	0 0 0 1 1	0 0 0 0 0 0 0 0 1 0
2	0 0 1 0 1	0 0 0 0 0 0 0 1 0 0
3	0 0 1 1 0	0 0 0 0 0 0 1 0 0 0
4	0 1 0 0 1	0 0 0 0 0 1 0 0 0 0
5	0 1 0 1 0	0 0 0 0 1 0 0 0 0 0
6	0 1 1 0 0	0 0 0 1 0 0 0 0 0 0
7	1 0 0 0 1	0 0 1 0 0 0 0 0 0 0
8	1 0 0 1 0	0 1 0 0 0 0 0 0 0 0
9	1 0 1 0 0	1 0 0 0 0 0 0 0 0 0

Das zweite Beispiel zeigt einen (1 aus 10)-Code, wird auch als „One Hot“ bezeichnet (die negierte Variante „One Cold“). Das Gewicht der Bitkombinationen ist 1 und Einbitfehler werden immer erkannt, Mehrbitfehler zu einem hohen Prozentsatz. Die Fehlererkennung ist in diesem Fall häufig unbedeutend für die Anwendung. Der Code dient normalerweise zur Ansteuerung von z. B. Anzeigeelementen. Jeder Ausgang hätte z. B. eine LED, die von 0 bis 9 ein Lichtband repräsentiert. Diese Bitkombination sorgt dafür, dass nur eine LED leuchtet und somit den Wert übersichtlich darstellt. Die Bitkombination wird weiterhin als Auswahl schaltung in der Mikroprozessortechnik verwendet, bei der mehrere Bausteine die gleiche physikalische Leitung benutzen, aber nur ein Baustein aktiv sein darf, um einen sicheren Datenaustausch zu ermöglichen. In beiden Anwendungsfällen ist es ein Ausgangsmuster, das z. B. aus einem Dualcode erzeugt wird (Adressdecoder, wird in Abschnitt 5.5 behandelt). Ähnlich könnten auch Tasterstellungen abgefragt werden (s. Bild 1.12). Durch eine sequenzielle Abfrage können die Taster unterschieden werden. Bei der 4×4-Taster-Matrix oder dem -Array wird die abzufragende Zeile über einen (1 aus 4)-Code aktiviert und eine gedrückte Taste über einen eingelesenen (1 aus 4)-Code erkannt. Die Abfrage des gesamten Arrays erfolgt mit einer schnellen permanenten Wiederholung. Veränderungen des Aufbaus, die das fehlerfreie Erkennen von mehreren gleichzeitig gedrückten Tasten ermöglichen, werden an dieser Stelle nicht weiter vertieft.

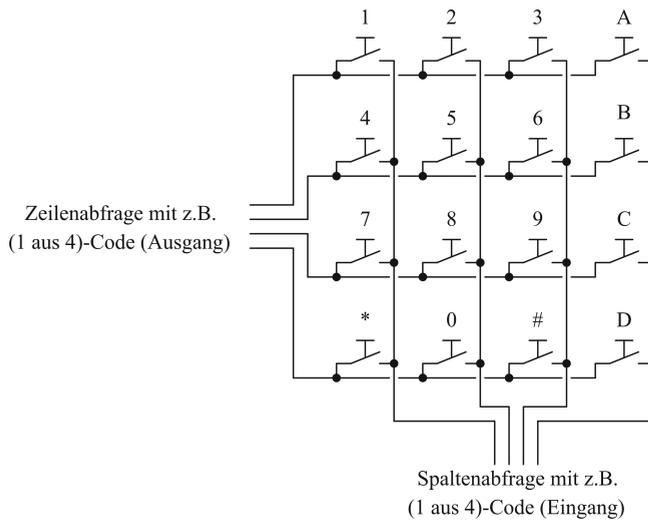
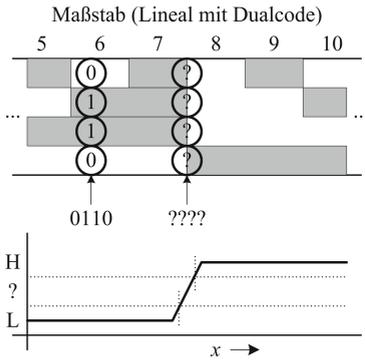


Bild 1.12 Aufbau einer 4×4-Taster-Matrix (vereinfacht)

1.4.3 Gray-Code

Ein Problem des Dualcodes ist die Detektion von Übergängen, z.B. zwischen den Codierungen für 7 und 8. Bild 1.13 zeigt einen Maßstab mit zwei möglichen Positionen eines Lesekopfes. In der linken Position sind die Markierungen auf dem Maßstab eindeutig erkennbar und der Lesekopf zur Erfassung des auf einen Maßstab aufgebracht Bitmusters (Längenmessung oder Winkelmessung) liefert die Bitkombination 0110. In der rechten Position liegt der Lesekopf genau an dem Übergang, bei dem sich alle Bitstellen ändern. Hier macht sich der analoge Charakter der Signale bemerkbar. Exemplarisch ist dies an dem positionsabhängigen Signal der Datenspur 2^3 im unteren Bildteil erkennbar. Es gibt einen Übergang, in dem H und L nicht eindeutig zugeordnet werden können. Dies trifft bei diesem Übergang für jede Datenspur zu. Hierdurch können prinzipiell alle möglichen Bitkombinationen als Ausgangssignal auftreten; Ursache ist der Wechsel mehrerer Bitstellen benachbarter Bitkombinationen. Der Dualcode ist somit für derartige Positionsbestimmungen nicht geeignet. In diesem Beispiel ist das Problem ortsabhängig von x (Längen- oder Winkelmessung), in Bild 1.5 ist die gleiche Problematik zeitabhängig von t dargestellt.

**Bild 1.13**

Signalwechsel zwischen 7 und 8 eines Maßstabes und der Signalverlauf der Datenspur $2^3=8$

Abhilfe schafft eine Codierung, bei der sich benachbarte Bitkombinationen nur um eine Bitstelle unterscheiden. Dies ist der in Tabelle 1.6 angegebene **Gray-Code**.

Tabelle 1.6 Gray-Code

Ziffer	Wertigkeit			
	–	–	–	–
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	0	1	0
4	0	1	1	0
5	0	1	1	1
6	0	1	0	1
7	0	1	0	0
8	1	1	0	0
9	1	1	0	1
10	1	1	1	1
11	1	1	1	0
12	1	0	1	0
13	1	0	1	1
14	1	0	0	1
15	1	0	0	0

Der hier angegebene Gray-Code ist vierstellig. Die jeweiligen Bitstellen sind nicht bewertbar. Es werden alle Bitkombinationen verwendet, dadurch ist die **Hamming-Distanz** eins und die **Redundanz** $R=0$. Besonderes Kennzeichen ist, dass sich benachbarte Bitstellen immer nur um ein Bit unterscheiden, deshalb ist dieser Code **stetig**. Bei den 16 verwendeten Bitkombinationen unterscheidet sich das letzte Codewort von dem ersten Codewort ebenfalls nur um eine Bitstelle. Dadurch ist

diese Codierung **zyklisch** und für die Ermittlung einer Winkelposition in rotierenden Messeinrichtungen geeignet. Der hier angegebene Gray-Code ist die Standardcodierung für absolut messende Längen- und Winkelpositionen.

■ 1.5 Betrachtung unabhängiger binärer Größen

In den bisherigen Abschnitten wurden die Bitkombinationen als Zahl interpretiert, entweder als Summe der mit 1 ausgewählten Wertigkeiten (z. B. Dualcode) oder auch als nicht bewertbare Bitkombination (z. B. Gray-Code). Wenn die Bitkombinationen keinen Wert repräsentieren, sind deren Zustände (0 oder 1) vermutlich unabhängig. Stellen sie sich einen Flur mit mehreren Türen und Lichttastern (z. B. 4 Taster) vor. Zur Ansteuerung des Treppenhausautomaten (Gerät zum zeitlich begrenzten Einschalten der Beleuchtung) werden diese Taster parallel geschaltet. Hierdurch haben alle Taster die gleiche Funktion, sollten aber aufgrund einer geeigneten Zuordnung eindeutig gekennzeichnet sein. Eine Möglichkeit wäre die Verwendung der Bezeichnung des über die Tür erreichbaren Bereiches wie Taster(Wohnzimmer), Taster(Küche), Taster(Esszimmer) und Taster(Hauseingang). Für eine verkürzte Schreibweise kann die Zuordnung zum Taster bzw. Schalter mit s (t ist als Variablenbezeichnung ungeeignet, weil diese als Zeitvariable verwendet wird) oder x gekennzeichnet werden und die vier Taster werden einfach durchnummeriert (z. B. von 0 bis 3). So entstehen die Bits s_3, s_2, s_1, s_0 . Die Zuordnung zu den Tastern sollte natürlich dokumentiert werden. Entsprechend der willkürlichen Zuordnung ergibt sich Tabelle 1.7, in der alle Tasteraktivitätszustände aufgeführt sind.

Es bietet sich an, die Bitkombination als Zahl anzugeben, aber sie hat nicht den Wert einer Zahl. Die Reihenfolge entsteht durch die Zuordnung (z. B. an der Haustür beginnend und links herum oder ...). Somit können die Spalten beliebig vertauscht werden. Nach dem Tausch sollten aus Gründen der Übersichtlichkeit die Zeilen der Tabelle sortiert werden. In unserem Beispiel ergibt sich die gleiche Tabelle nur mit geänderter Reihenfolge der Variablen im Kopf der Tabelle. Die Taster sind unabhängig und haben keine Wertigkeit.

Zahlenwerte aus dem Wertebereich \mathbb{R} werden auf einem Zahlenstrahl dargestellt. Binäre Größen haben nur zwei Werte, 0 und 1. Bei einem Zahlenstrahl oder einer Achse (Koordinatenachse zur Darstellung reeller Werte) sind dies zwei Punkte, einer bei dem Wert 0 und einer bei dem Wert 1.

Tabelle 1.7 Auflistung möglicher Tasterzustände

s_3	s_2	s_1	s_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

Unabhängige Größen werden orthogonal (rechtwinklig) dargestellt. Genauso wird mit der zweiten binären Größe verfahren. Als räumliche Darstellung erhält man durch die zwei Basisvektoren \vec{e}_x und \vec{e}_y (Basisvektoren sind Einheitsvektoren mit der Länge 1 und geben die Richtungen in einem Koordinatensystem an) erst einmal eine Fläche, z. B. Breite und Tiefe mit vier Punkten für die Bitkombinationen 00, 01, 10 und 11. Bei dreistelligen Binärgrößen kommt eine weitere, auf den zwei vorhandenen Richtungen eine ebenfalls rechtwinklige, stehende Richtung hinzu, z. B. die Höhe \vec{e}_z . Hierdurch erhält man eine dreidimensionale Struktur, die normalerweise auf einer zweidimensionalen Tafel- oder Blattoberfläche als Ansicht dargestellt wird. Statt der Einheitsvektoren betrachten wir die Richtungen der Achsen in den Richtungen x , y , und z . Die Werte 0 und 1 sind Punkte auf diesen Achsen, die hier aufgrund der besseren Beschriftungsmöglichkeit durch Kugeln verdeutlicht werden (im Bild als Kreise dargestellt). In unserem binären Fall sind es acht Punkte, die Eckpunkte eines Würfels wie er in Bild 1.14 in der linken Darstellung zu sehen ist. Zur besseren Übersicht sind die Kanten des Würfels als dünne gestrichelte Linien eingezeichnet. Sie verbinden jeden Punkt mit den direkten Nachbarpunkten, die sich nur in einer Bitstelle von dem Ausgangspunkt unterscheiden.

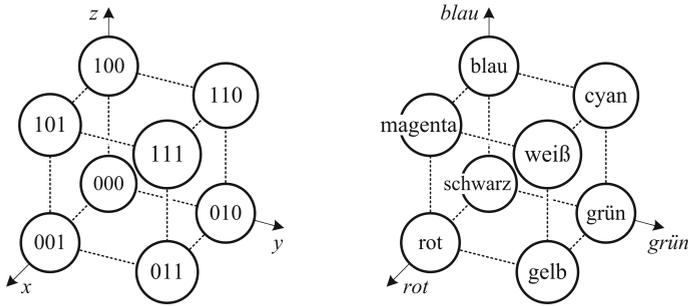
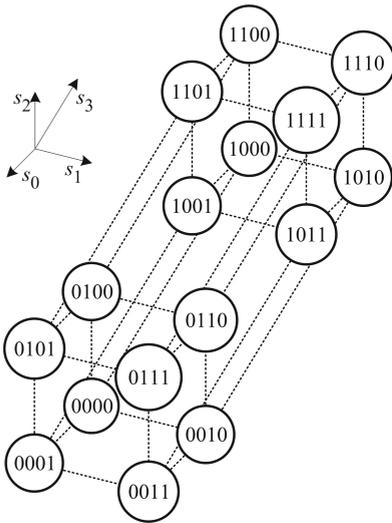


Bild 1.14 Dreidimensionale Darstellung eines Binärcodes mit 3 unabhängigen Bits

Auf der rechten Seite in Bild 1.14 werden die Variablen *rot*, *grün* und *blau* verwendet. Unter der Annahme, dass die Farben nur aus- oder eingeschaltet werden können und sich Weiß ergibt, wenn alle drei Farben eingeschaltet sind (additive Farbmischung), ergeben sich die bezeichneten Farbeindrücke. Gelb ist hierbei nicht die Spektrallinie gelb, sondern die Überlagerung der Spektrallinien Rot und Grün, die den Farbeindruck gelb im Auge entstehen lässt, wie es bei allen RGB-Farbwiedergaben (RGB steht für Rot-Grün-Blau) der Fall ist. Es ist sicher zu erkennen, dass die in der Farbe enthaltenen drei Bit gleichzeitig dargestellt und auch über eine Glasfaser optisch übertragen werden können. Diese Übertragungsart ist ein optisches Multiplexverfahren (lat.: multiplex: vielfach, vielfältig) mit drei unabhängigen Datenkanälen.

Bei einer zusätzlichen Dimension durch ein weiteres Bit scheitert das Farbbeispiel aufgrund der farblichen Wechselwirkungen im Auge (keine eindeutige Farbzusammensetzung erkennbar) als einfaches Beispiel. Das optische (Wellenlängen-)Multiplexverfahren funktioniert weiterhin aufgrund der angepassten Anzahl an Farbfiltern zur Selektion der einzelnen Farbkanäle. Diese Anwendung soll hier nicht weiter vertieft werden.

In der räumlichen Darstellung könnte man die Zeit als vierte Dimension hinzunehmen und hätte die Würfeldarstellung für den Zeitpunkt 0 und für den Zeitpunkt 1. Identische Ecken der beiden Würfel wären gedanklich wieder durch dünne gestrichelte Hilfslinien verbunden. Möglich wäre am Rechner eine GIF-Darstellung (engl.: Graphics Interchange Format, geeignet zur Darstellung kleiner Bildsequenzen), bei der abwechselnd zwischen den Würfeldarstellungen hin und her geschaltet wird (zur Unterscheidung vielleicht mit den Farben rot für 0 und grün für 1). Für eine zweidimensionale Darstellung eignet sich eine seitlich versetzte Ansicht des zweiten Würfels (zu erkennen an der Achse s_3), wie er in Bild 1.15 zu sehen ist.

**Bild 1.15**

Vierdimensionale Darstellung eines Binärcodes mit vier unabhängigen Bits (s_3, s_2, s_1, s_0)

Wie bereits erwähnt verbinden die gestrichelten Linien Bitkombinationen mit der Distanz 1 (unterscheiden sich nur in einem Bit). Der Gray-Code ist ein Weg über alle Knoten entlang der Hilfslinien. Jeder Weg über die Hilfslinien ist ein einschrittiger Code (es müssen nicht alle Knoten verwendet werden); ein geschlossener Weg charakterisiert zusätzlich einen zyklischen Code. Man könnte hierüber einen problemangepassten einschrittigen Code entwerfen (z. B. 10 Bitkombinationen ohne 0000 und ohne 1111, um Unterbrechung und Kurzschluss als gültige Bitkombination auszuschließen); eine übersichtlichere Methode wird in Abschnitt 4 verwendet.

Startet man bei der Bitkombination 1111, so hat das Gewicht den Wert 4 und dies ist ein gerader Wert. Wechselt man entlang einer Hilfslinie zum direkt benachbarten Knoten, so erhält man eine Bitkombination mit ungeradem Gewichtswert und danach wieder eine mit geradem usw. (Bitkombination 0000 hat das Gewicht 0 und zählt als gerade). Diese Betrachtung wird bei der Parity-Bewertung in Abschnitt 1.6.3 vertieft.

Die Beschreibung lässt sich mit beliebig vielen Bits erweitern. In der Mathematik spricht man dann vom n -dimensionalen binären Vektorraum; in unseren Beispielen ist es ein J -dimensionaler binärer Vektorraum (J ist die Anzahl der Bitstellen).

■ 1.6 Fehlererkennende und fehlerkorrigierende Verfahren

1.6.1 Fehlermodelle

Im Wesentlichen geht es hier um Veränderungen gespeicherter und/oder übertragener binärer Daten. Betrachten wir zunächst gespeicherte binäre Daten auf einem Speicherbaustein. Im Idealfall werden alle Daten fehlerfrei wieder ausgelesen. Stellen sie sich vor, sie schreiben z. B. die Tabelle des Gray-Codes handschriftlich auf einem Zettel auf, den sie bei guter Handschrift eindeutig und fehlerfrei lesen können. Könnte es sein, dass eine andere Person ihre Handschrift nicht so gut kennt und er/sie/es die Tabelle nicht fehlerfrei lesen kann? Negative Einflüsse wie Sonnenlicht könnten die Schrift mit der Zeit verblassen oder durch Feuchtflecken unleserlich werden lassen. Bei einer grafischen Darstellung von schwarzen und weißen Feldern könnten Bereiche heller oder dunkler und dadurch unleserlich werden.

Ähnliches Verhalten gibt es auch bei der digitalen Speicherung. Herstellungsbedingt könnte eine Speicherzelle (ein Bit) nicht richtig beschreibbar sein und nach einer gewissen Zeit nach 0 oder 1 driften. Ein sofortiges Auslesen (Schreibtest) würde keinen Fehler anzeigen, weil die ungewünschte Änderung erst nach einer gewissen Zeit auftritt. Je nach Ursache bereits nach wenigen Millisekunden oder erst nach mehreren Jahren. Wenn nur eine Zelle betroffen ist, so spricht man von Einbitfehler, die permanent (der Fehler ist bei jedem Auslesevorgang vorhanden) oder sporadisch (der Fehler tritt manchmal auf, aber nicht immer) auftreten können. Sind beim Auslesen zwei unabhängige Bits betroffen, so spricht man von einem Doppelfehler oder Zweibitfehler, bei drei Bit von einem Dreifachfehler oder Dreibitfehler usw., allgemein von Mehrfachfehlern oder Mehrbitfehlern. Haben die Fehler eine Tendenz, dass z. B. nur Nullen fehlerhaft als Einsen gelesen werden, so spricht man von einseitigen oder gleichseitigen Fehlern mit Tendenz zu 1 (1 dominierend). Gleiches gilt natürlich auch für die andere Richtung, bei der fehlerhafte Einsen als Nullen gelesen werden (0 dominierend). Dies hat unter Umständen Auswirkungen auf auszuwählende Fehlererkennungsverfahren, die aber hier nicht weiter vertieft werden sollen. Im Weiteren werden unabhängige Einzelbitfehler geringer Häufigkeit diskutiert (bei einer hohen Fehlerwahrscheinlichkeit sollte man ein anderes Speichermedium wählen).

Ähnliches gilt auch für die Datenübertragung, bei der einzelne Bits durch Störsignale verfälscht werden. Unter einer Datenübertragung versteht man einen Transport der Bitzustände an einen anderen Ort, die Speicherung von Daten als Datenübertragung in eine spätere Zeit.

Eine andere Fehlerart ist der Ausfall größerer Bereiche zusammengehöriger Bits. Ist zum Beispiel die Leitung kurzzeitig unterbrochen oder kurzgeschlossen, so fällt die Übertragung über mehrere zu übertragende Bits aus und diese werden alle als 1 oder als 0 gelesen. Ähnlich sieht es beim Lesen einer CD aus, wenn die Oberfläche einen Kratzer hat. Für den Lesekopf sind es zusammenhängende Bits und Bytes. Hier kann durch eine Verteilung der Information mit Fehlerkorrekturverfahren entgegengewirkt werden. Diese Fehlerart wird im Weiteren ebenfalls nicht weiter vertieft.

1.6.2 Mehrfachübertragung

Die einfachste Art einer sicheren Übertragung ist die Mehrfachübertragung der gleichen Bitinformation, immer vorausgesetzt wird eine kleine Fehlerrate. Wenn z. B. eine Frage verneint werden soll, kann man das „Nein“ mehrfach wiederholen, in der Hoffnung, dass genügend Wörter in ausreichender Qualität beim Fragesteller (Empfänger der Antwort) ankommen. Dieser könnte die empfangenen Wörter statistisch auswerten, indem er nicht als Antwort erkennbare Laute (unzulässige Bitkombinationen ohne Korrektur) einfach streicht (aber die Problematik des Übertragungskanals erkennen lässt), erkennbare aber nicht ganz korrekte Wörter korrigiert (Fehlererkennung mit Fehlerkorrektur) und eindeutig erkannte Wörter aufschreibt und anschließend eine Art Mehrheitsentscheid durchführt. Hierfür ist mindestens eine Dreifachübertragung notwendig.

Wird jedes Bit dreifach übertragen (Redundanz 1,585), ist die einfachste Form eine Reduzierung der Übertragungsrate um den Faktor drei. Der Empfänger fragt dieses Bit dreimal ab und erwartet, dass alle Abfragen die gleiche Information liefern. Weicht eine der drei Abfragen des um den Faktor drei gedehnten Bits ab, so zeigt sich hierdurch eine unsichere Datenübertragung. Bild 1.16 zeigt oben die ursprüngliche Nachricht mit den Daten, dem führenden Start- und dem abschließenden Stopp-Bit (Start- und Stopp-Bit stellen eine Art Umschlag für die zu übertragenden Daten dar und dienen zur Steuerung der Übertragung). Die mittlere Darstellung zeigt eine Übertragung bei der jedes Bit dreimal ausgegeben oder wichtiger, dreimal gelesen wird.