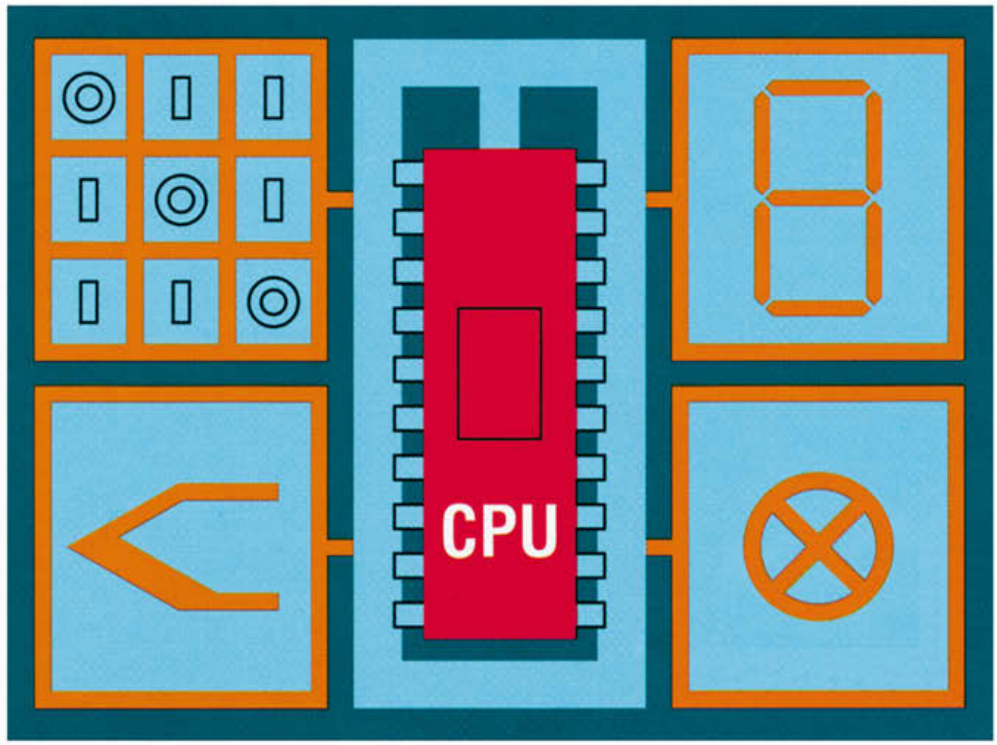
Vogel Fachbuch

Müller/Walz

Mikroprozessor-technik

Elektronik 5



VOGEL

Elektronik 5

Helmut Müller / Lothar Walz

Mikroprozessortechnik

8., bearbeitete Auflage

Vogel Buchverlag

Weitere Informationen:
www.vogel-buchverlag.de


http://twitter.com/vogelbuchverlag


www.facebook.com/vogel.buchverlag


www.vogel-buchverlag.de/rss/buch.rss

ISBN 978-3-8343-3285-1
8. Auflage. 2012

Herstellung: Vogel Business Media GmbH & Co. KG, Würzburg

Vorwort

Mikrocontroller werden immer kleiner, billiger und trotzdem noch leistungsfähiger. Ihr Einsatz ist inzwischen auf allen Gebieten der Steuerungs- und Regelungstechnik, ob im industriellen Bereich, im Verkehr, in der Telekommunikation oder im Haushalt, nicht mehr wegzudenken. Inzwischen sind auch neue Berufe entstanden, die als Hauptlernziel den Erwerb von Kenntnissen über Aufbau und Anwendung von Computern haben. Kernstück dieser stürmischen und noch immer nicht abgeschlossenen Entwicklung ist der Mikroprozessor.

In diesem Band 5 der Fachbuchgruppe *Elektronik* werden der bei allen Computern doch recht ähnliche Aufbau und die gleiche grundsätzliche Arbeitsweise und Programmierung ausführlich erklärt. Aufbau und Verschaltung der zu einem Computer gehörenden Hardware und die Anwendung von externen oder auf dem Chip integrierten Ein-/Ausgabebausteinen werden erläutert, ebenso die zweckmäßige Strukturierung von Programmen und ihre Codierung, sowohl in Assembler als auch in der höheren Programmiersprache C.

Nach unserer Ansicht ist es zweckmäßig, diese allgemeinen Prinzipien anhand eines wirklich existierenden Systems zu erarbeiten. Dazu wird im Hauptteil des Buches ein moderner, sehr weit verbreiteter Mikrocontroller – der 80C537 aus der schon lange existierenden 8051-Familie – verwendet. Um dem Leser eigenständiges Arbeiten zu ermöglichen, ist eine vollständige Entwicklungsumgebung, bestehend aus Editor, Assembler, C-Compiler und Debugger, im Onlineservice InfoClick abrufbar. Die mitgelieferten Beispiele geben Anregungen, wie verschiedene Standardprobleme gelöst werden können. Es können eigene Programme entwickelt und die Funktion mit dem integrierten Debugger getestet werden.

Kurze Einführungen in die speziellen Eigenschaften von Signalprozessoren anhand der Familie ADSP21xx der Firma Analog Devices sowie in einen RISC-Prozessor der PIC-Familie der Firma Microchip sollen einerseits dem Leser einen Überblick über andere moderne Prozessorarchitekturen geben, aber ebenso zeigen, daß auch diese Prozessoren auf dem bereits bekannten Prinzip aufbauen.

Das vorliegende Buch soll auch vermitteln, daß die Abläufe in all diesen Computern im Grunde gleich und eigentlich recht «primitiv» sind. Erst die immer weiter erhöhte Arbeitsgeschwindigkeit und die große Zuverlässigkeit dieser Bausteine machen den Computer heutzutage zu einem wertvollen Werkzeug.

Für das Verständnis dieses Buches sind Grundkenntnisse über binäre Signale und deren Grundverknüpfungen, das duale Zahlensystem und das Prinzip eines Flipflops ausreichend.

Das Buch ist geeignet für alle diejenigen, für die ein genaueres Verständnis der Arbeitsweise und Programmierung von Mikrocomputern wichtig ist, auch als Grundlage für die Aneignung von weiterem Wissen.

Freiburg im Breisgau

Helmut Müller
Lothar Walz

Zur Fachbuchreihe «Elektronik» gehören die Bände:

Klaus Beuth / Olaf Beuth: Elementare Elektronik

Heinz Meister: Elektrotechnische Grundlagen
(Elektronik 1)

Klaus Beuth: Bauelemente
(Elektronik 2)

Klaus Beuth / Wolfgang Schmusch: Grundschaltungen
(Elektronik 3)

Klaus Beuth: Digitaltechnik
(Elektronik 4)

Helmut Müller / Lothar Walz: Mikroprozessortechnik
(Elektronik 5)

Wolfgang Schmusch: Elektronische Meßtechnik
(Elektronik 6)

Klaus Beuth / Richard Hanebuth / Günter Kurz / Christian Lüders: Nachrichtentechnik
(Elektronik 7)

Wolf-Dieter Schmidt: Sensorschaltungstechnik
(Elektronik 8)

Inhaltsverzeichnis

1 Einführung

1.1 Prinzip der Datenverarbeitung

Datenverarbeitung mit Hilfe von Computern ist inzwischen in fast alle Bereiche der industrialisierten Welt eingeführt. Computer helfen bei der Erfassung und Aufbereitung von Meßwerten, der Herstellung von Fertigungsunterlagen, sie steuern Maschinen aller Größen, verwalten Lagerbestände, kalkulieren Angebote, berechnen Gehälter und Steuern, speichern Verkehrssünder und freie Plätze in Flugzeugen. Der Leser könnte diese Liste sicher noch aus eigener Kenntnis ergänzen.

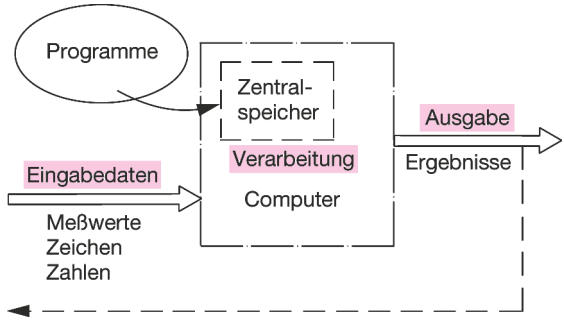
Aber so verschiedenartig diese Anwendungen auf den ersten Blick auch sein mögen, vom Prinzip her ergibt sich immer der gleiche Ablauf einer Datenverarbeitung (Bild 1.1).

Eingabe
Eingabegeräte stellen Daten zur Verarbeitung bereit.

Zu diesen *Eingabegeräten* zählen beispielsweise:

- Meßfühler (Sensoren), etwa für Temperatur, Durchflußmenge, Gehalt an Schwefeldioxid usw.
  Meßfühler besitzen inzwischen oft eine eigene «Intelligenz», mit der die Meßwerte schon aufbereitet werden. Die Übertragung zum «Sammelrechner» erfolgt dann über ein System von Leitungen, ein «Netzwerk», an das viele Baugruppen angeschlossen werden können. Leider gibt es viele konkurrierende Verfahren zur Übertragung der anfallenden Daten, wie Interbus-S, Profibus, I^2C-Bus, CAN-Bus usw.;
- Taster oder ganze Tastaturen, Endschalter,
- Datenspeicher wie Magnetband- oder Diskettenlaufwerke.

Bild 1.1
Prinzip der Datenverarbeitung



Verarbeitung
Diese Daten werden nun von der eigentlichen Verarbeitungseinheit, dem Computer, gelesen und entsprechend einer im Computer vorhandenen Arbeitsanweisung, dem Programm, verarbeitet.

Von den eingelesenen Meßwerten müssen vielleicht ein Mittelwert gebildet, Maximal- oder Minimalwerte überwacht oder Endschalter auf Betätigung überprüft werden. Bei einem Angebot müssen Warenpreise addiert und die Mehrwertsteuer hinzugerechnet werden.

Ausgabe
Die Ergebnisse dieser Verarbeitung müssen ausgegeben werden.

Mittelwerte werden als Diagramm gezeichnet; Motoren werden – abhängig von der Stellung der geprüften Endschalter – gestartet oder angehalten, Meldeleuchten aktiviert, Angebote gedruckt oder Werte über ein «LAN» (Local Area Network) an übergeordnete Steuer- bzw. Verwaltungsrechner weitergegeben.

Der Computer übernimmt dabei nicht nur die Verarbeitung, sondern steuert auch das Einlesen der Daten sowie die Ausgabe der Ergebnisse.

Dieses Arbeitsprinzip gilt unabhängig davon, ob es sich nun um *Mikro-*, *Mini-*, *Midi-* oder *Großcomputer* (Mainframe) handelt. Sie unterscheiden sich durch ihre Verarbeitungsgeschwindigkeit, die Anzahl der anschließbaren Geräte und natürlich durch ihren Preis.

Daten fallen in zwei verschiedenen Formen an:

- *als analoge Werte*, z.B. bei Messungen von Temperatur, Druck, pH-Wert, Spannung am Abgriff eines Potentiometers;
- *in binärer Form*, z.B. bei Eingaben von Tastaturen, beim Lesen von Barcodes auf Warenverpackungen oder bei der Überprüfung von Endschaltern.

1.1.1 Binäre Daten

Heutzutage werden Daten fast ausschließlich in binärer Form verarbeitet, d.h., analoge Daten müssen vor der weiteren Verarbeitung erst mit Hilfe von Analog-Digital-Umsetzern in binäre Signale umgewandelt werden.

Die folgenden Merksätze zeigen die Unterschiede der verschiedenen Signalarten.

Analoge Signale können zwischen zwei Grenzwerten *beliebig viele Zwischenwerte* annehmen.
Digitale Signale können nur eine *begrenzte Anzahl* von Zuständen annehmen, z.B. die Zustände +5 V, 0 V, –5 V.
Binäre *Signale* können nur *zwei Zustände* annehmen.

Ein Schalter ist ein Beispiel für ein binäres Signal; er kann

- ❏ **betätigt** sein, dann ist das davon abgeleitete Signal: **aktiv**, wahr, true, logisch «1» oder in abgekürzter Schreibweise: **1**;
- ❏ **nicht betätigt** sein, dann ist das Signal: **nicht aktiv**, falsch, false, logisch «0» oder in abgekürzter Schreibweise: **0**.

In Computern werden binäre Signale meist durch zwei verschiedene elektrische Potentiale gebildet, die gut voneinander unterschieden werden können.

Bei der Übertragung gilt immer:
- ❏ Das **negativere** der beiden Potentiale wird als **LOW** oder abgekürzt als **L** bezeichnet.
- ❏ Das **positivere** der beiden Potentiale wird als **HIGH** oder abgekürzt als **H** bezeichnet.

Bei dieser Festlegung können auch beide Potentiale positiv bzw. negativ sein; so verwendet man bei der ECL-Technik (Emitter coupled logic) die beiden Potentiale –0,75 V und –1,5 V. In diesem Falle ist –1,5 V das negativ**ere** Potential und wird mit Low bezeichnet.

Nun muß noch eine Festlegung zwischen der logischen Bedeutung eines Signals (**0** oder **1**) und den beiden Potentialen (**L** oder **H**), mit denen es übertragen wird, getroffen werden. Dabei gibt es zwei Möglichkeiten (Tabelle 1.1), was häufig zu Verwirrungen führt.

In einem System werden oft beide Möglichkeiten verwendet. Dann müssen die Signalnamen ein Kennzeichen erhalten, das deutlich macht, welche Zuordnung für dieses Signal gilt (Bild 1.2).

Tabelle 1.1

Positive Logik (Zuordnung)	**Negative Logik (Zuordnung)**
HIGH: Signal ist aktiviert, also logisch 1 LOW: Signal ist nicht aktiviert, also logisch 0	HIGH: Signal ist nicht aktiviert, also logisch 0 LOW: Signal ist aktiviert, also logisch 1

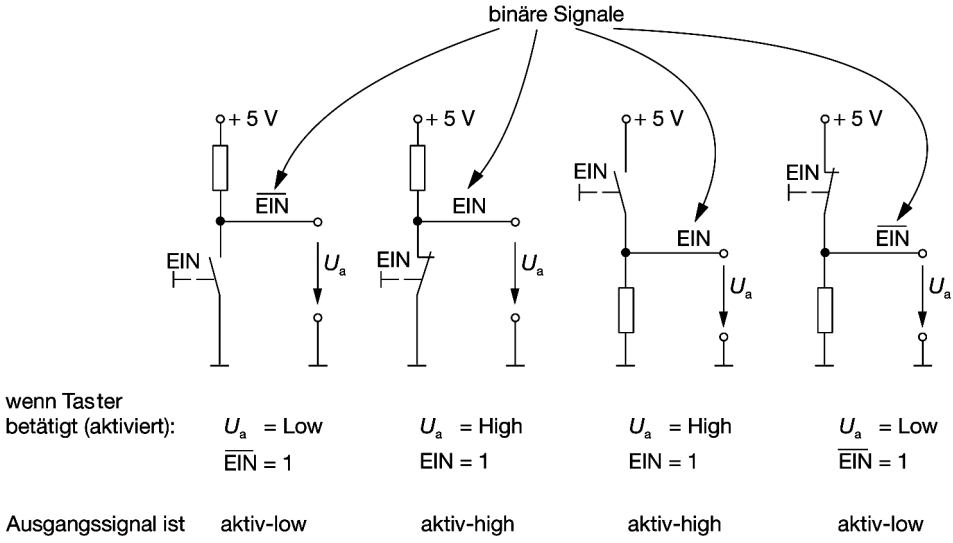


Bild 1.2 Signalkennzeichnung aktiv-high, aktiv-low

Ein Signal, für das die **positive Zuordnung** gelten soll, bezeichnet man als **aktiv-high**, es wird nicht besonders gekennzeichnet; es gilt: **HIGH ⇔ 1** und **LOW ⇔ 0**.

Ein Signal, für das die **negative Zuordnung** gelten soll, bezeichnet man als **aktiv-low**, es erhält im Namen entweder einen Überstrich oder ein voran- bzw. nachgestelltes Doppelkreuz (#) oder einen Schrägstrich (/); es gilt: **LOW ⇔ 1 und HIGH ⇔ 0**.

Beispiele

`A0, A1, SEL`	Positive Zuordnung	**L** bedeutet **0** (Signal inaktiv, false, logisch <0>) **H** bedeutet **1** (Signal aktiv, true, logisch <1>)
`pumpe_ein`	Positive Zuordnung	**L ≅ 0** (Signal inaktiv, Pumpe ist nicht eingeschaltet) **H ≅ 1** (Signal aktiviert, Pumpe ist eingeschaltet)
`#pumpe_ein`, $\overline{\texttt{pumpe\_ein}}$	Negative Zuordnung	**L ≅ 1** (Signal aktiv, Pumpe ist eingeschaltet) **H ≅ 0** (Signal inaktiv, Pumpe ist nicht eingeschaltet)
`#pumpe_aus`	Negative Zuordnung	**L ≅ 1** (Signal aktiv, Pumpe ist ausgeschaltet) **H ≅ 0** (Signal inaktiv, Pumpe ist nicht ausgeschaltet)

Mit einem binären Signal kann **1 Bit** übertragen werden. 1 Bit (0 oder 1) ist die kleinste Einheit der Informationsverarbeitung.

Mikrocomputer sind meist aus Schaltkreisen in MOS-Technologie gefertigt, die mit den folgenden Potentialen arbeiten:

LOW: 0 V...+0,8 V; HIGH: +2 V...+5 V

In diesem Buch sind also immer, wenn von Rechnern, (Mikro-) Computern usw. die Rede ist, Systeme gemeint, die ausschließlich binäre Signale verarbeiten.

Die Technik der sog. *Analogrechner* wird kaum mehr angewendet. Sie eignen sich vor allem für die Untersuchung des zeitlichen (dynamischen) Verhaltens von Systemen, die aus verschiedenen Energiespeichern und Dämpfungen bestehen, also mathematisch durch Differentialgleichungen beschrieben werden können. Ein typisches Beispiel dafür ist die Messung des Einschwingverhaltens eines Pkw-Fahrwerks, bestehend aus trägen Massen, Federn und Stoßdämpfern. Dabei werden diese Kennwerte mit Hilfe von beschalteten Operationsverstärkern elektrisch nachgebildet. Durch Variation der Einstellung der Operationsverstärker können so leicht verschiedene Auslegungen des Fahrwerks *simuliert* werden.

Die digitale Verarbeitung hat sich aus mehreren Gründen durchgesetzt:

- Daten, z.B. analoge Meßwerte, können nur in digitalisierter Form über längere Zeit gespeichert und wieder verarbeitet werden.
- Die Genauigkeit der Verarbeitung läßt sich mit relativ geringem Aufwand praktisch beliebig steigern.
- Digitalisierte Daten sind bei Übertragung störunempfindlicher als analoge Signale. Bei Übertragung analoger Signale addiert sich jede noch so kleine Störung zum Nutzsignal, während sich bei digitaler Übertragung Störungen unterhalb einer bestimmten Schwelle überhaupt nicht bemerkbar machen. Selbst Störungen über dieser Schwelle können bei entsprechendem Aufwand durch Fehlerkorrekturverfahren beseitigt werden.

1.1.2 Verarbeitung binärer Daten

Im Computer – genauer gesagt im Rechenwerk des Computers – befinden sich recht komplizierte Schaltungen, die eingelesene Daten z.B. addieren, subtrahieren, vergleichen oder in ihrem Stellenwert verschieben können. Welche der möglichen Operationen die Daten durchlaufen, wird durch ein Programm festgelegt.

Bei einem Computer handelt es sich nicht um eine Schaltung, die speziell für eine ganz bestimmte Aufgabe konstruiert ist, sondern um eine universelle binäre Schaltung, die eingelesene Daten entsprechend den Befehlen des im Computer gespeicherten Programms bearbeitet.

Dieses Programm kann geändert oder gegen ein anderes ausgetauscht werden.

Das ergibt einen sehr komplizierten, umständlichen und recht langsamen Ablauf. Müssen Mikrocomputer in «Echtzeit» (real time) arbeiten, d.h. etwa bei einem Regler die Menge der anfallenden Meßwerte so schnell verarbeiten, wie sie vom Meßfühler erzeugt werden, haben sie – was die Geschwindigkeit betrifft – deutliche Nachteile gegenüber konventionellen, aus Operationsverstärkern aufgebauten Reglern.

Der große Vorteil der Datenverarbeitung mit binär arbeitenden Computern liegt darin, daß die Schaltung, also die *Hardware*, im Prinzip *immer gleich aufgebaut* werden kann, weil die Anpassung an die spezielle Aufgabe durch das Programm, die *Software*, geschieht.

So lohnt es sich, für die Baugruppen des Computers sehr hochintegrierte Schaltkreise zu entwerfen, die dann in riesigen Stückzahlen hergestellt werden können und dadurch recht preiswert werden.

Bei Änderungen der Aufgabenstellung kann die Software relativ leicht ausgetauscht werden, ohne daß – im Prinzip – Änderungen der Verdrahtung notwendig werden.

Die Kosten verlagern sich dann mehr auf die Herstellung der Software, die bis zu 80% der Gesamtkosten einer Computeranwendung ausmachen kann.

Nur für sehr häufig vorkommende Aufgaben lohnen sich sonst noch die Entwicklungskosten für spezielle integrierte Schaltungen (ICs), z.B. bei Taschenrechnern oder Digitaluhren und in der Telekommunikationstechnik.

Eine Wandlung bahnt sich hier an durch den Einsatz eben dieser Computer beim Entwurf und Testen neuer ICs. Auch bei kleineren Stückzahlen wird es zunehmend wirtschaftlich, ICs speziell für eine ganz bestimmte Aufgabe zu entwerfen, sog. ASICs (Application Specific Integrated Circuits). Diese ASICs arbeiten dann wesentlich effektiver als ein Mikrocomputer.

Inzwischen gibt es programmierbare logische Bausteine (Programmable Logic Devices, PLD), die eine große Anzahl von UND- und ODER-Gattern enthalten, bis hin zu *Sequenzern*, die ganze Befehlsfolgen abarbeiten können und sich immer mehr der Leistungsfähigkeit von Mikroprozessoren annähern.

1.2 Blockschaltbild eines Mikrocomputers

In diesem Abschnitt wird erst der prinzipielle Aufbau eines Mikrocomputers erklärt, genauere Einzelheiten folgen in Kapitel 2.

Ein kompletter Computer kann auf einem einzigen Chip in einem 40poligen Gehäuse integriert, auf einer Leiterplatte im Europaformat 100 × 160 mm untergebracht sein oder – wie bei den heute üblichen Personalcomputern – eine Leiterplatte im DIN-A3-Format einschließlich mehrerer Zusatzplatinen füllen.

Jeder digital arbeitende Computer besteht aus den Baugruppen (Bild 1.3):
- ❏ *Zentraleinheit* (Central Processing Unit, *CPU*),
- ❏ *Zentralspeicher (Memory)*,
- ❏ *Ein-/Ausgabe-Einheiten* (Input/Output, *I/O*).

1.2.1 Zentraleinheit

Die CPU hat zwei Aufgaben:
- ❏ Sie *steuert den gesamten Ablauf*, sowohl innerhalb der CPU als auch im gesamten Computer.
- ❏ Sie *bearbeitet Daten*, kann also arithmetische und logische Operationen ausführen.

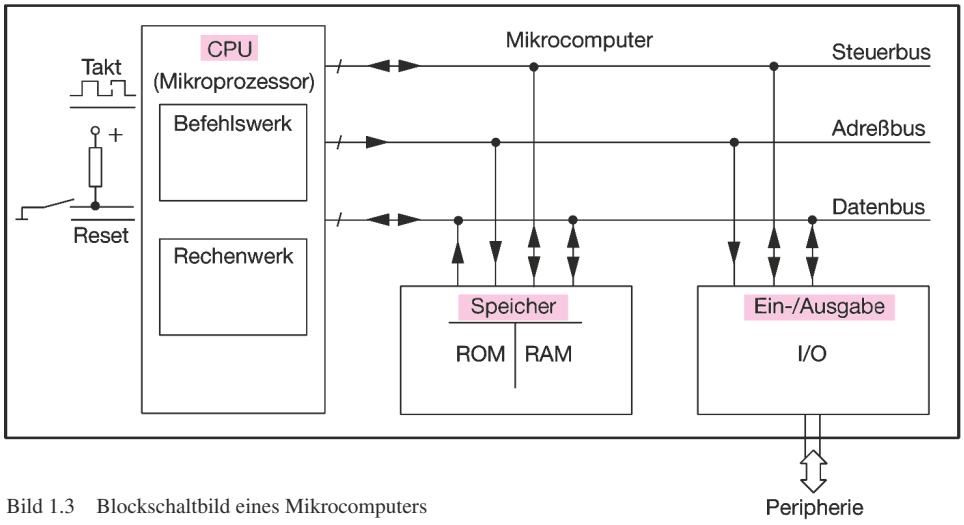


Bild 1.3 Blockschaltbild eines Mikrocomputers

Durch die seit einiger Zeit mögliche Integrationsdichte können diese Funktionen in einem einzigen Chip implementiert werden.

Eine CPU, die auf einem einzigen Chip integriert ist, bezeichnet man als *Mikroprozessor*.

Ein Mikroprozessor ergibt für sich allein noch keinen funktionsfähigen Computer, es gehören noch Speicher und E/A-Bausteine dazu.

Jede CPU hat einen Eingang für ein rechteckförmiges Taktsignal, das meist von einem Quarzgenerator erzeugt wird. Damit werden alle Abläufe innerhalb der CPU und des gesamten Mikrocomputers gesteuert. Solche Mikroprozessoren werden von vielen Firmen angeboten. Sie unterscheiden sich durch ihre Arbeitsgeschwindigkeit, ihren Programmierkomfort und die Anzahl der Bits, die mit einem Befehl verarbeitet werden können (je nach Typ 4, 8, 16, 32 oder 64 Bit).

1.2.2 Zentralspeicher

Zentralspeicher werden von der CPU direkt angesteuert – im Gegensatz zu peripheren Speichern – z.B. Disketten, Festplatten, die über spezielle E/A-Bausteine (Controller) angesteuert werden müssen.

Zentralspeicher werden heute fast ausschließlich in Halbleitertechnologie hergestellt, wobei zwei Gruppen unterschieden werden.

Nichtflüchtige Speicher (Non Volatile Memory) behalten ihre Informationen auch beim Abschalten der Versorgungsspannung. Sie sind geeignet für Programme, Codetabellen usw.

Sie werden, nicht mehr ganz korrekt, als ROM (Read Only Memory = Nurlesespeicher) bezeichnet; es gibt inzwischen Typen, bei denen der Inhalt relativ einfach geändert werden kann, siehe Abschnitt 2.3.

Flüchtige Speicher verlieren ihre Information beim Abschalten der Versorgungsspannung, dafür lassen sie sich aber einfach ändern. Sie werden zum Zwischenspeichern von Eingabedaten und Ergebnissen benötigt.

Sie werden als *RAM* (Random Access Memory) oder *Schreib-Lese-Speicher* bezeichnet.

Grundsätzlich kann man sich einen Zentralspeicher vorstellen als einen Schrank mit vielen untereinander angeordneten Schubladen – fachmännisch «*Speicherzellen*» genannt – die in einem gegebenen System alle die gleiche Anzahl von binären Stellen (Bitstellen) enthalten. Sie ist abgestimmt auf die Anzahl, die die CPU auf einmal verarbeiten kann, z.B. 8 Bitstellen.

Jede Speicherzelle ist durch eine Zahl – fachmännisch «Adresse» genannt – gekennzeichnet. Die CPU kann solche Adressen auf den Adreßbus schalten und damit beliebige Speicherzellen anwählen.

1.2.3 Ein-/Ausgabebausteine

Sie stellen die Verbindung zur «Außenwelt» des Computers, zur *Peripherie*, her. Sie werden oft als Input-Output-Ports oder kurz I/O-Ports (engl.: port = Tor, Türe, Pforte) bezeichnet.

Sie entlasten die CPU von Routineaufgaben und erhöhen dadurch die Arbeitsgeschwindigkeit des Systems. Es gibt inzwischen sehr komplexe Bausteine, die z.B. parallele in serielle Daten in einem selbst erzeugten Zeittakt umwandeln können. Nicht zu vergessen sind die E/A-Bausteine zur Steuerung von Disketten- oder Festplattenlaufwerken. Andere Bausteine verstärken nur die von der CPU gesendeten Signale zum Betrieb von Relais oder Meldeleuchten.

1.2.4 Busleitungen

Alle Baugruppen eines Mikrocomputers sind durch Busleitungen (abgekürzt «Bus») miteinander verbunden.

> Der Bus besteht aus mehreren Leitungen, wobei an jede Leitung parallel mehrere Bausteine angeschlossen sind (Bus von lat.: omnibus = alle).

Das bedeutet, daß eine einzige Leitung alle mit D0 bezeichneten Datenanschlüsse von CPU, Speicher und E/A-Bausteinen miteinander verbindet.

Die beiden anderen Gruppen, *Adreßbus* und *Steuerbus*, benötigt die CPU zur Steuerung dieser Informationstransporte, womit das Problem gelöst werden kann, daß auf dem *Datenbus immer nur ein Baustein senden und ein Baustein empfangen darf.*

1.2.5 Mikrocontroller

Da sich bei ICs die Integrationsdichte immer weiter erhöht hat, können CPU, Zentralspeicher und einige Peripheriebausteine – also ein kompletter Mikrocomputer – auf einem einzigen Chip untergebracht werden. Man bezeichnet solche Computer auf einem Chip als **Mikrocontroller**, **Single Chip Computer** oder **Embedded Computer**. Da sie wenig Platz beanspruchen und – bei niedriger Taktfrequenz – nur wenig Leistung aufnehmen und durch hohe Stückzahlen billig geworden sind, findet man solche Mikrocontroller heute fast in allen Geräten, wo etwas zu messen, steuern, speichern, anzuzeigen oder zu drucken ist. Auch in diesem Buch werden Arbeitsweise, Programmierung und Anwendung von Mikrocomputern anhand eines Mikrocontrollers, des 80C537 aus der weitverzweigten 8051-Familie, erklärt.

1.3 Prinzipielle Arbeitsweise eines Computers

1.3.1 Historische Entwicklung

Der Anfang der Entwicklung, die schließlich zur heutigen Struktur führte, geht auf das Jahr 1833 zurück. Zwar wurden auch schon vorher Rechenmaschinen erdacht und gebaut, aber die waren noch nicht programmgesteuert, entsprachen also eher unseren heutigen Taschenrechnern. 1833 begann der englische Mathematiker CHARLES BABBAGE mit der Entwicklung seiner «Analytical Engine», die bereits aus einem Datenspeicher, Rechenwerk, Steuerwerk und einer Ein-/Ausgabe bestehen sollte. Die Steuerung, also «Programmierung», sollte durch Lochkarten erfolgen, die schon seit 1728 zur Steuerung von Webstühlen verwendet wurden. Die Ausführung scheiterte jedoch an mechanischen Schwierigkeiten.

Erst 1941 verwirklichte KONRAD ZUSE mit seinem Relaisrechner Z3 diese Prinzipien, obwohl er die Ideen von BABBAGE nicht gekannt hatte. Dieser Rechner gilt als der erste programmgesteuerte Rechenautomat, der auch wirklich funktionierte. Das Programm wurde, immer noch extern, auf einem gelochten Kinofilmstreifen gespeichert. ZUSE hat zur Entwicklung der Rechnertechnik zwei neue Ideen beigesteuert: die Verwendung bistabiler, also binär wirkender Schaltelemente und des dualen Zahlensystems, das bereits von LEIBNIZ 1679 in seiner Schrift «De progressione Dyadica» veröffentlicht worden war. LEIBNIZ beschrieb auch, wie einfach die Regeln der vier Grundrechenarten in diesem Zahlensystem werden.

ZUSE erkannte auch, daß sich bei Dualzahlen alle Berechnungen mit den binären Grundoperationen UND, ODER und NICHT durchführen lassen.

Die ersten Rechner von ZUSE waren noch mit mechanischen bistabilen Elementen ausgestattet; der Durchbruch erfolgte erst, als mit der Elektronenröhre ein schnelles, nicht mechanisch bewegtes Bauelement für bistabile Kippstufen zur Verfügung stand. Diese ersten «Elektronenrechner» waren saalfüllende Ungetüme, wie z.B. der 1945 in Amerika in Betrieb genommene ENIAC mit 18 000 Röhren und einer Leistungsaufnahme von 150 kW. Nur etwa in der halben Betriebszeit war der Rechner funktionsfähig, die andere Zeit über mußten defekte Röhren gesucht und ausgetauscht werden.

Die vorerst letzten grundsätzlichen Ideen fügte gegen Ende des 2. Weltkrieges der aus Deutschland ausgewanderte Mathematiker VON NEUMANN hinzu.
Nicht nur Daten, sondern auch die Programme sollten im Zentralspeicher abgelegt werden, sie wurden speicherresident.

Die Verwendung «bedingter Sprungbefehle» ermöglichte es nun – z.B. abhängig von der Größe eines Ergebnisses oder eines Zählregisters – Programmteile beliebig oft zu wiederholen (Schleifen) oder aber zu überspringen, d.h., das Programm konnte selbständig Entscheidungen treffen, ohne daß der Mensch eingreifen mußte.

Damit war die prinzipielle Entwicklung abgeschlossen; was danach noch folgte, war im wesentlichen technologische Weiterentwicklung.

Etwa ab 1955 wurden die leistungsfressenden und relativ unzuverlässigen Röhren durch Transistoren ersetzt, ab 1965 erschienen dann integrierte Schaltkreise.

Hier verzweigt sich die Entwicklung. Zum einen entstehen in bipolarer Technik immer schnellere Schaltkreise mit Durchlaufzeiten unter 1 ns, die in Großrechnern eingesetzt werden. Zum anderen erreicht man in der einfacher aufzubauenden MOS-Technik immer größere Packungsdichten.

Etwa seit 1975 gibt es ICs, bei denen ein komplettes Steuer- und Rechenwerk auf einem Chip integriert ist: eben den Mikroprozessor mit mehr als 1 Million Transistorfunktionen. Dadurch konnten immer kleinere, verlustärmere, dabei noch schnellere und billigere Computer hergestellt werden. Diese Computer werden nicht mehr in klimatisierten Rechenzentren aufgestellt, sondern können überall vor Ort, wo Rechenleistung benötigt wird, eingesetzt werden.

Die grundlegende Arbeitsweise eines Computers ist jedoch seit den Ergänzungen Zuses und Von Neumanns gleichgeblieben.

1.3.2 Prinzipieller Ablauf eines Programms

Das Steuerwerk der CPU ist durch Steuerleitungen mit allen Baugruppen innerhalb der CPU und über die Steuerbusleitungen auch mit Speicher und E/A-Bausteinen verbunden.
Welche Steuerleitungen nun in welcher Reihenfolge aktiviert werden, wird durch *Maschinenbefehle* – das sind bestimmte Kombinationen aus 0 und 1 – festgelegt. Solch ein Maschinenbefehl bewirkt meist nicht sehr viel, z.B. den Transport einer 8stelligen Dualzahl vom Rechenwerk der CPU in eine Speicherzelle oder in einen E/A-Baustein. Andere Maschinenbefehle bewirken die Addition oder Subtraktion von zwei in der CPU gespeicherten 8stelligen Dualzahlen. Auch für scheinbar einfache Probleme, wie Addition längerer Dezimalzahlen, Wurzelziehen oder gar Berechnung einer Sinusfunktion, werden bereits eine ganze Menge dieser Maschinenbefehle benötigt. Befehlswerke von Mikroprozessoren können einige Hundert verschiedene Maschinenbefehle ausführen, von denen allerdings viele sehr ähnlich sind. Für jede CPU liefert der Hersteller *Befehlslisten*, in denen alle ausführbaren Befehle aufgeschrieben sind.

> Aus diesen Befehlslisten muß der Programmierer diejenigen Befehle heraussuchen, die er zur Lösung des Problems benötigt.

Je nach Geschick des Programmierers ergeben sich dabei mehr oder weniger elegante Lösungen.

> Die ausgewählten Befehle müssen nun genau in der Reihenfolge, wie sie vom Befehlswerk ausgeführt werden sollen, im Zentralspeicher in unmittelbar aufeinanderfolgenden Speicherzellen abgelegt werden.

Der erste auszuführende Befehl muß dabei in Speicherzelle 0 stehen, und es dürfen keine Zellen übersprungen werden. Ein solches Programm kann z.B. mit Hilfe eines Programmiergerätes in ein EPROM (siehe Abschnitt 2.3) eingeschrieben und dann in einen für einen Speicherbaustein vorgesehenen Sockel des Mikrocomputers eingesetzt werden.

Jedes Befehlswerk besitzt nun ein «Zählwerk», den *Programmzähler*, der im Prinzip die Nummer oder besser die Adresse des Befehls enthält, der gerade ausgeführt wird. Durch einen Taster, der auf den Reset-Eingang der CPU führt, kann der Programmzähler auf 0 gesetzt werden. Bei den meisten Computern wird dieses Reset-Signal nach dem Einschalten der Spannungsversorgung automatisch für kurze Zeit aktiviert. Anschließend beginnt das Befehlswerk seine Arbeit:

- ❏ Es schaltet den Programmzähler auf den Adreßbus, und der Inhalt der Adresse 0 wird aus dem Speicher über den Datenbus in das Befehlswerk geladen (Bild 1.4).
- ❏ Der Befehl wird nun decodiert und die entsprechenden Steuersignale erzeugt, so daß der Befehl «ausgeführt» wird. Anschließend erhöht das Befehlswerk *automatisch* den Programmzähler um 1 (Bild 1.5).
- ❏ Dieser nächste Befehl wird nun in das Befehlswerk geladen und ausgeführt (Bild 1.6).
- ❏ Nach Ausführung dieses Befehls wird der Zähler wiederum automatisch erhöht, der nächste Befehl geladen usw.

Dieser eigentlich recht sture Ablauf wiederholt sich nun immerzu. Er läßt sich nur beenden durch einen speziellen Befehl, den sog. HALT-Befehl, oder durch Abschalten der Spannungsversorgung.

Spezielle Befehle, die sog. Verzweigungsbefehle (Abschnitt 4.4), bewirken bei ihrer Ausführung, daß der Programmzähler mit einer anderen Zahl geladen wird, die o.g. Prozedur wird dann eben ab dieser Adresse fortgesetzt.

Ein Computer «steht also niemals still», auch wenn «nichts abläuft». So wird z.B. bei einem PC dauernd der Bildschirm neu beschrieben und die Tastatur abgefragt, ob eine Taste gedrückt wurde.

Eines sollte nach dem eben Gesagten klar geworden sein: *Ein Computer kann nicht «denken»*, er kann nur die vom Programmierer eingegebenen Befehle der Reihe nach ausführen. Ein Computer – die Größe spielt dabei keine Rolle – kann nur auf solche Ereignisse reagieren, die der Programmierer in seinem Programm vorausgesehen hat. Ein Beispiel mag das verdeutlichen: Auch Experten können Schwierigkeiten haben, gegen ein gutes Computer-Schachprogramm zu gewinnen. Aber das Schachprogramm wird z.B. von sich aus nie in der Lage sein, etwa die Regeln zu ändern oder gar ein neues Spiel zu erfinden.

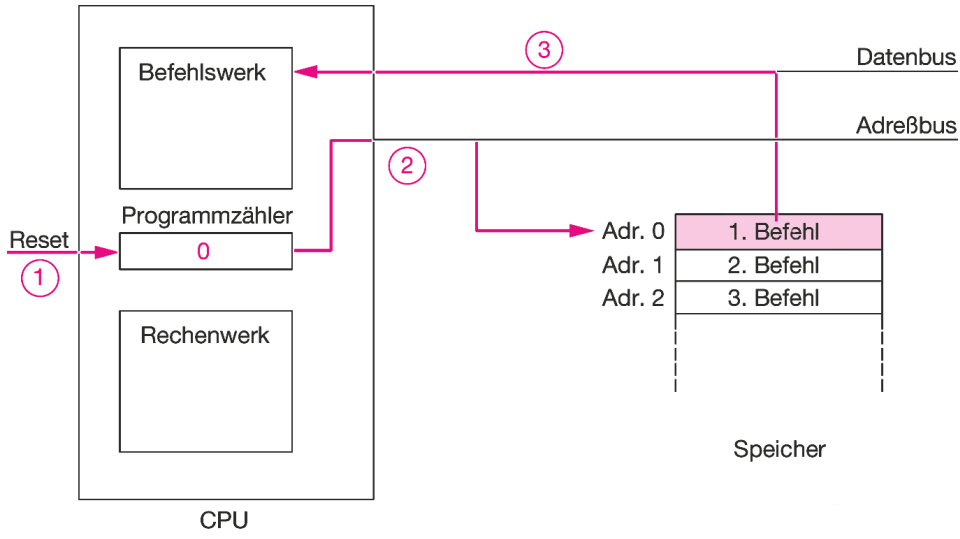


Bild 1.4 Laden des ersten Befehls

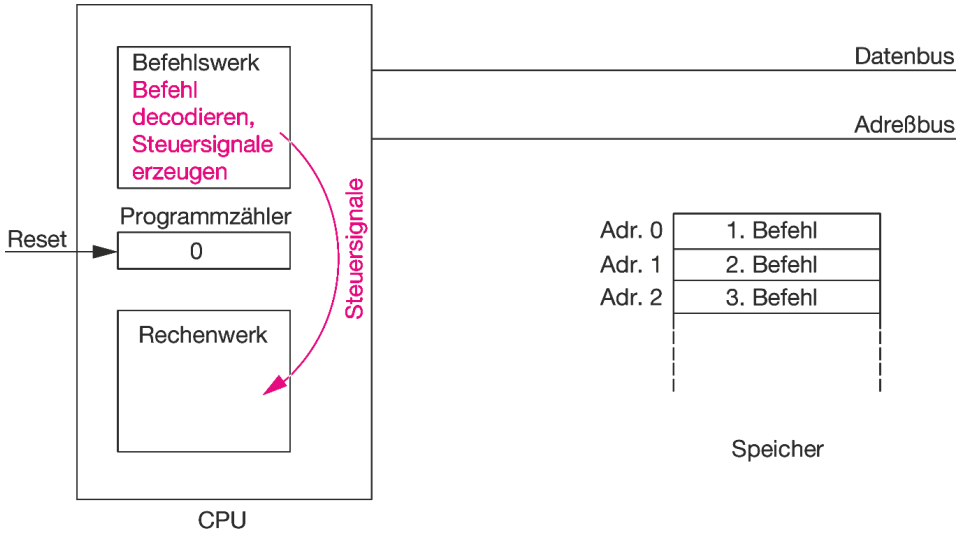


Bild 1.5 Ausführung eines Befehls

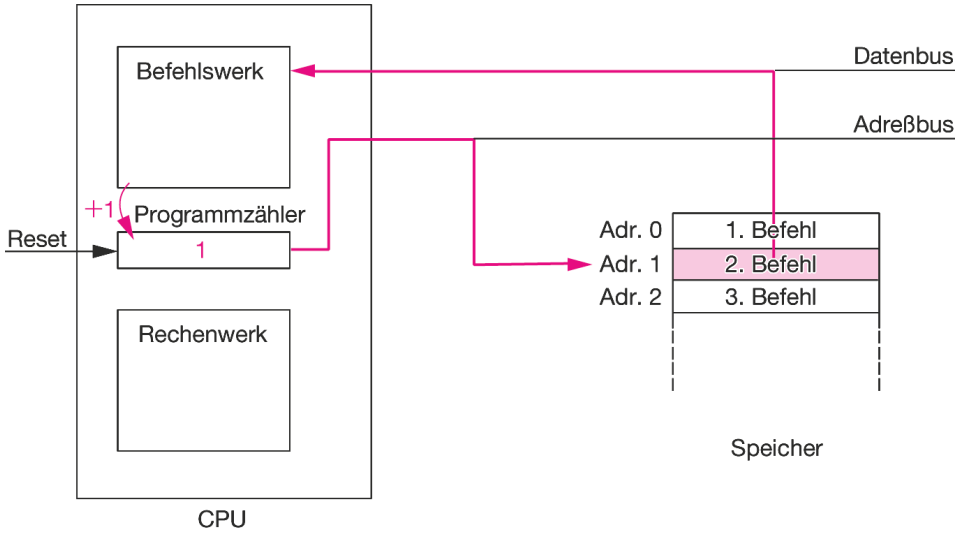


Bild 1.6 Laden des nächsten Befehls

Die Maschine ist dem Menschen eigentlich nur in zwei Dingen überlegen:

- Die Befehlsausführung erfolgt sehr zuverlässig und immer in exakt der gleichen Weise und Zeitdauer.
- Befehle werden sehr schnell ausgeführt. Mikrocontroller, mit denen wir uns noch näher beschäftigen werden, können in der Sekunde etwa 800 000 Befehle ausführen bei einer Taktfrequenz von 12 MHz, andere Mikroprozessoren (siehe Kapitel 10) auch 10mal mehr. Man spricht dann von 0,8 bzw. 8 MIPS (Million Instructions per Second). Auch die weiteren Kapitel dieses Buches werden zeigen, daß ein Computer keineswegs ein denkendes Wesen, sondern nur eine – allerdings recht komplizierte – elektronische Steuerung ist.

2 Baugruppen eines Mikrocomputers

In diesem Kapitel soll die Hardware eines Computers, also CPU, Zentralspeicher, Ein-/Ausgabe-Bausteine und Systembus, etwas eingehender behandelt werden.

2.1 Systembus

Die Busleitungen, die die Baugruppen eines Mikrocomputers – CPU, Zentralspeicher, E/A-Bausteine – miteinander verbinden, werden als *Systembus* bezeichnet.

Es gibt noch andere Busleitungssysteme, z.B. sind in der CPU alle Register durch einen «internen Datenbus» miteinander verbunden. Auch außerhalb eines Computers werden häufig Bussysteme benutzt, um mehrere Peripheriegeräte an eine E/A-Baugruppe anzuschließen; der in der Meßautomatisierung häufig verwendete IEC-Bus ist ein Beispiel dafür. Lokale Netzwerke (LAN, Local Area Networks) werden als Bussystem ausgeführt.

Auch eine Gemeinschaftsantennenanlage ist als Bussystem ausgeführt, wobei ein Sender – der Antennenverstärker – mehrere Empfänger betreibt. Im Prinzip können durch «Anzapfen» Geräte dazugeschaltet werden, ohne daß jedesmal eine neue Leitung vom Dach verlegt werden muß. In einem Computersystem ist die Informationsverteilung jedoch schwieriger, weil mehrere Datensender vorhanden sind. So müssen z.B. Zahlen von verschiedenen Speicherbausteinen zur CPU gesendet werden, damit sie dort bearbeitet werden können. Ebenso müssen Daten von Tastaturen oder Stellungen von Endschaltern über Eingabebausteine zur CPU gesendet und Ergebnisse von der CPU zu Speicher- oder E/A-Bausteinen transportiert werden.

Zur korrekten Abwicklung des Datenflusses werden im Systembus eines Computers entsprechend ihrer Aufgabe drei Gruppen von Busleitungen unterschieden:

- ❏ Datenbus,
- ❏ Adreßbus,
- ❏ Steuerbus.

Um die Steuerung des Datenflusses nicht zu kompliziert zu machen, sind nur *Transporte zwischen CPU und Speicher* oder zwischen *CPU und E/A-Bausteinen* möglich – jeweils in beiden Richtungen.

Da die an einer Datenleitung angeschlossenen Bausteine sowohl senden als auch empfangen können, spricht man hier von einem *bidirektionalen Bus*.

Das Steuerwerk der CPU muß nun zwei Probleme lösen:

- ❏ Auswahl einer bestimmten Speicherzelle oder eines E/A-Bausteins für den Datentransport;
- ❏ Festlegung der Richtung des Transportes.

> Das Steuerwerk sorgt dafür, daß zur selben Zeit bei jeder Datenbusleitung nur ein Sender und ein Empfänger durchgeschaltet sind – entsprechend dem gerade auszuführenden Befehl.

2.1.1 Bausteine an Busleitungen

Zum Anschluß an Datenbusleitungen geeignete Bausteine müssen folgende Voraussetzungen erfüllen:

- ❏ Die Datenbusanschlüsse brauchen abschaltbare Sender bzw. Empfänger, sog. Tristate-Anschlüsse, mit den Zuständen:
  – HIGH,
  – LOW,
  – hochohmig (high impedance, high z).
- ❏ Die Bausteine brauchen Steuereingänge zur Steuerung der Datenflußrichtung und zum Anschalten des Datenbusanschlusses. Meist werden die folgenden Anschlußbezeichnungen verwendet (Bild 2.1):
  – Aktivierung des Bausteins: CE (Chip Enable); CS (Chip Select); DS oder DE (Device Select oder Device Enable),
  – Aktivierung des Dateneingangs: WR oder WE (WRite oder Write Enable),
  – Aktivierung des Datenausgangs: OE oder RD (Output Enable oder ReaD).

Der Übersichtlichkeit halber ist in Bild 2.1 nur einer von 8 Datenbusanschlüssen gezeichnet. Bei nicht aktiviertem Steuereingang $\overline{CS}$ ist der Datenbusanschluß von der Innenschaltung des Bausteins getrennt, der Anschluß ist, von außen gemessen, hochohmig. Der Baustein selbst ist außerdem in eine Art Ruhezustand versetzt, wobei er meist nur ca. 20% des Stromes im aktiven Zustand aufnimmt.

Die Steueranschlüsse sind meist aktiv-low.

> Das Signal für den CS eines Bausteins wird aus den Signalen des Adreßbusses abgeleitet, und zwar so, daß (außer der CPU) immer nur ein Baustein aktiviert wird, siehe nächsten Abschnitt.

Die Signale für OE und WE werden von der CPU ausgegeben, sie geben die Übertragungsrichtung auf dem Datenbus an. Die dabei üblichen Begriffe **Read/Lesen** und **Write/Schreiben** gelten vom Standpunkt der CPU aus (Tabelle 2.1).