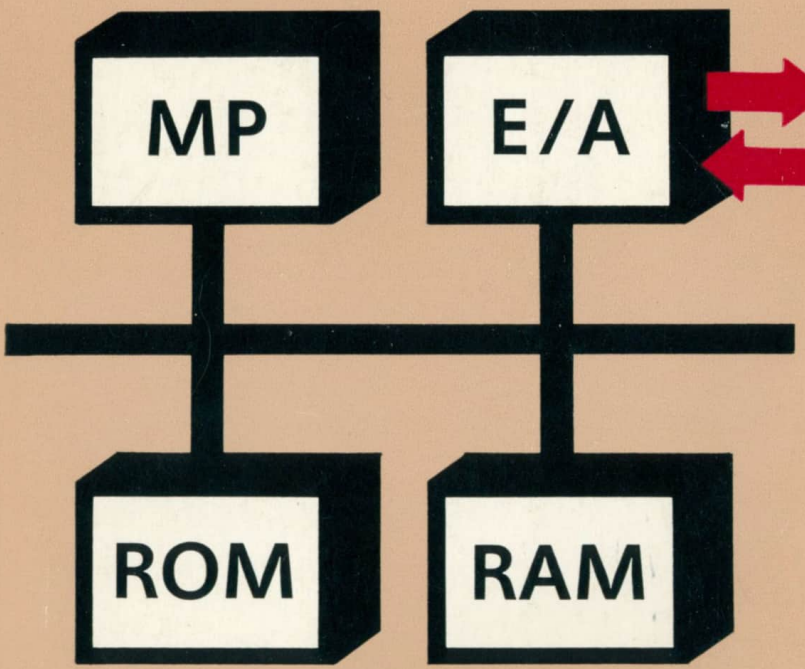
Hopfer

Mikrorechentechnik

allgemeinverständlich

Integrierte Schaltkreise
Mikroprozessoren
Programme



POLYTECHNISCHE BIBLIOTHEK

Integrierte Schaltkreise–Mikroprozessoren–Programme

Dr.-Ing. Reiner Hopfer

Mikrorechentechnik – allgemeinverständlich

Mit 66 Bildern und 9 Tabellen

VEB FACHBUCHVERLAG LEIPZIG

Dieser Band der Polytechnischen Bibliothek
erscheint in Zusammenarbeit mit der
«Urania» Gesellschaft zur Verbreitung
wissenschaftlicher Kenntnisse

Herausgegeben von
Prof. Dr. sc. techn. Lutz-Günther Fleischer, Berlin
Prof. Dr. sc. nat. Horst Wolffgramm, Halle
Doz. Dr. sc. agr. Gerhard Holzapfel, Berlin

1. Auflage
Lizenznummer 114-210/40/85
LSV 3009
Verlagslektor: Angela Szargan
Gesamtgestaltung: Sabine Panster
Printed in GDR
Gesamtherstellung: VEB Druckhaus Köthen
Redaktionsschluß: 15. 1. 1985
Bestellnummer: 5470276
00550

Inhaltsverzeichnis

Vorwort

Die Mikrorechentechnik ist eine verhältnismäßig junge Fachdisziplin. Innerhalb einer relativ kurzen Zeit hat sie jedoch eine Entwicklung erfahren und Bedeutung erlangt, wie sie selten einem technischen Gebiet zuteil wurde. Wegbereiter dieser Fortschritte waren insbesondere die Mikroelektronik sowie eine Reihe anderer Fachgebiete. Der Einsatz der Mikrorechentechnik erstreckt sich heute auf nahezu alle Gebiete der Wirtschaft und des gesellschaftlichen Lebens. Ihre besondere Bedeutung liegt jedoch im Bereich der Automatisierung und Rationalisierung zahlreicher Industrie-, Informations- sowie geistiger Prozesse. Beispiele hierfür sind der Maschinen-, Anlagen- und Produktionsmittelbau, die Steuerungs- und Meßtechnik, die Erzeugung, Verteilung und Umwandlung von Energie sowie alle Arten der Verfahrenstechnik. Außerdem zählen hierzu als wichtige Einsatzgebiete das Verkehrs- und Transport- sowie das Post- und Fernmeldewesen, das Gesundheitswesen sowie alle den genannten Bereichen vor- und nachgelagerten Prozesse, wie Forschung und Entwicklung, Projektierung, Handel und Versorgung und nicht zuletzt auch der Haushalt. An vielen Stellen wird der arbeitende Mensch in Zukunft mehr und mehr direkt oder indirekt mit der Mikrorechentechnik in Berührung kommen. Das wird gegenwärtig schon am Beispiel des Taschenrechners, des Heimcomputers, der Digitaluhr, der elektronisch gesteuerten Näh- und Strickmaschine, der programmierbaren Waschmaschine, des Schachcomputers sowie an einer Vielzahl entsprechender elektronischer Geräte und Einrichtungen am Arbeitsplatz sichtbar.

Die Mikrorechentechnik existiert heute als eigenständiges Fachgebiet. Das theoretische und praktische Fundament dieser Disziplin ist breit gefächert und ruht im wesentlichen auf den Grundlagen der Mikroelektronik, Rechentechnik, Informatik, Mathematik und Physik sowie auf Erkenntnissen einer Reihe anderer Wissensgebiete. Bei vielen Lesern ist sicher der Wunsch entstanden, Näheres über diese Technik zu erfahren. Allerdings sind die in heutigen elektronischen Erzeugnissen ablaufenden Prozesse so kompliziert, daß bestimmte

Kenntnisse über Ursachen und Wirkungsweise notwendig sind. Der modernen Elektronik und auch der Mikrorechentechnik ist ein Prinzip innewohnend, das darin besteht, bestimmte systemtechnische Abstraktionsvorgänge vorzunehmen, die es ermöglichen, auch komplizierte Prozesse oder Systeme auf einem höheren logischen Niveau relativ einfach zu beschreiben. Auf diese Weise kann man Verfahren oder technische Einrichtungen begreifen, überschauen und sich ihrer bedienen, ohne im einzelnen zu wissen, was im «Detail» abläuft. Für eine allgemeinverständliche Darstellung, wie sie in diesem Buch erfolgen soll, sind überdies noch gewisse Vereinfachungen einiger Sachverhalte notwendig. Die darzustellenden Zusammenhänge und Grundprinzipien werden jedoch davon nicht berührt.

Um das Zusammenwirken der einzelnen Funktionsgruppen eines Mikrorechners zu verstehen, muß man sich zunächst mit den physikalischen, technologischen und systemtechnischen Grundlagen vertraut machen, Aufbau und Wirkungsweise der entsprechenden hochintegrierten elektronischen Bauelemente, wie Mikroprozessor und Speicherschaltkreis, begreifen und deren Verhalten und Zusammenwirken in einer «Mikrorechnerstruktur» näher untersuchen. Auf diesem Wissen aufbauend, werden dann die Programmierung sowie geeignete Methoden und Verfahren der inneren Organisation der Mikrorechner und deren vielfältige Einsatzmöglichkeiten verständlich.

Das vorliegende Buch ist für alle diejenigen als Hilfe gedacht, die sich einen Überblick über die Mikrorechentechnik und die entsprechenden, dafür notwendigen Teile der Mikroelektronik verschaffen wollen. An Voraussetzungen werden nur einige wenige Grundkenntnisse in Physik, Elektrotechnik und Mathematik benötigt.

In der Mikrorechentechnik wird mit einer Vielzahl spezifischer Fachbezeichnungen gearbeitet. Eine entsprechende Übersicht der wesentlichsten Begriffe ist hierzu im Anhang enthalten.

Da dieses Buch eine Reihe von Themen nicht oder nur am Rande behandeln kann, werden Hinweise auf weiterführende Literatur gegeben.

An dieser Stelle möchte ich mich besonders bei Herrn Professor Dr. sc. techn. *L.-G. Fleischer* und Herrn Dozent Dr. sc. techn. *D. Werner* für zahlreiche Hinweise und Vorschläge bedanken.

Der Verfasser

1. Entwicklungsgeschichte der Mikrorechentechnik

1.1. Energie und Information

Die zur Herausbildung und Entwicklung der Mikrorechentechnik erforderlichen praktischen und theoretischen Grundlagen haben eine lange und bewegte Geschichte. Ausgangspunkt der elektrotechnischen, elektronischen und später so wichtigen informationstechnischen Zweige und Fachrichtungen war die Physik (Bild 1). Im vorigen Jahrhundert hatte sich die Elektrotechnik als selbständiges Gebiet aus der klassischen Elektrizitätslehre der Physik entwickelt und bedeutende Erfindungen, wie die Telegrafie, Telefonie, das elektrische Licht, den Elektromotor, Generator und Transformator hervorgebracht. Später wurde der elektrische Strom immer mehr dazu benutzt, nicht nur Energie, sondern auch Informationen in größerer Menge und besserer Qualität zu übertragen. Die Nachrichtentechnik bildete sich heraus und gewann wirtschaftliche Bedeutung. Waren es bei der Telegrafie zuerst nur einfache Stromimpulse, so folgten später bereits codierte, d.h. verschlüsselte Stromsignale, die die Buchstaben der Schriftsprache widerspiegelten, sowie das Übertragen von menschlicher Sprache über große Entfernungen und die Verstärkung schwacher elektrischer Signale. Dies verlangte eine eigene Fachdisziplin und eine theoretische Behandlung der neu auftretenden «schwachstromtechnischen» Probleme. In der Tat folgte nunmehr eine allmähliche Aufspaltung der *klassischen Elektrotechnik* in die Gebiete *Starkstromtechnik* (im Zentrum steht hier die Energie) und *Schwachstromtechnik* (zentraler Gegenstand ist die Information). Die Schwachstromtechnik ist insbesondere als die Wiege der Elektronik anzusehen.

Ein wichtiges Instrument und erstes elektronisches Verstärkungselement wurde die Elektronenröhre, deren erste brauchbare Lösung 1907 *R. von Lieben* entwickelte. Es war dann *G. H. Barkhausen*, der das damalige «Wunderwerk Röhre» wissenschaftlich untersuchte und

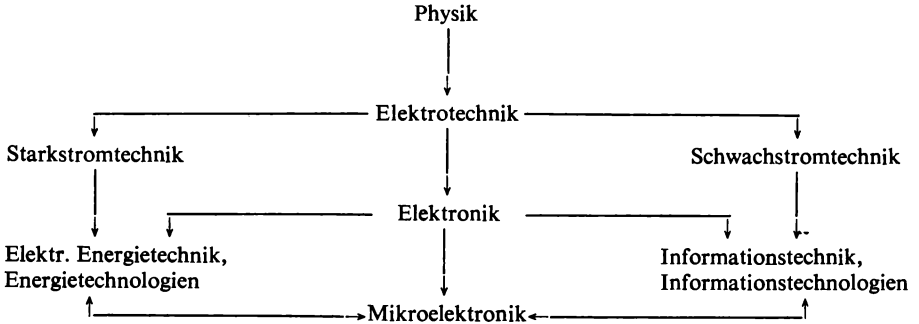


Bild 1. Historische Wurzeln und Einflußgebiete der Mikroelektronik

wichtige theoretische sowie praktische Voraussetzungen für die breite Entfaltung der Schwachstromtechnik schuf.

In der nachfolgenden Zeit wurden weitere elektronische Bauelemente entwickelt, darunter spezielle Widerstände, Spulen, Kondensatoren, Filter, Schalter und Relais. Die Elektronenröhre konnte technisch verbessert und bis zur Großserienfertigung vervollkommnet werden. Diese Bauelemente der sogenannten klassischen Elektronik bildeten in den nächsten Jahrzehnten die Grundlage vieler elektronischer Geräte, Einrichtungen und Systeme, z.B. der Rundfunk- und Fernmeldetechnik, der später an Bedeutung gewinnenden Radar- und Fernsehtechnik sowie der Rechentechnik.

1.2. Von der Elektronik zur Mikroelektronik

Im Jahr 1947 wurde ein neuer Abschnitt der Elektronik eingeleitet, als es *J. Bardeen, W. H. Brattain* und *W. Shockley* gelang, das Phänomen der Halbleitereffekte elektronisch zu nutzen. Mit der Erfindung des *Transistors* schufen sie ein neues und viel kleineres Verstärkungselement als die Röhre (Nobelpreis 1956). Die ersten technischen Nutzungen 1952 eröffneten den Siegeszug des Transistors um die Welt. Bereits Ende der 50er Jahre wurde die Idee verwirklicht, viele dieser Elemente auf kleinstem Raum, auf einem einzigen Baustein – einem Halbleiterplättchen aus kristallinem Material – zu vereinigen. Der «Integrierte Schaltkreis» war entstanden und die Elektronik erhielt ein neues Teilgebiet, die *Mikroelektronik*. Nun folgt eine Entwicklung der Halbleitertechnologien, wie sie in der Technikgeschichte nicht oft zu finden ist. In ununterbrochener Folge werden neue integrierte Schaltkreise entworfen, entwickelt und produziert, die immer leistungsfähiger, zuverlässiger, energieärmer arbeiten, ökonomischer

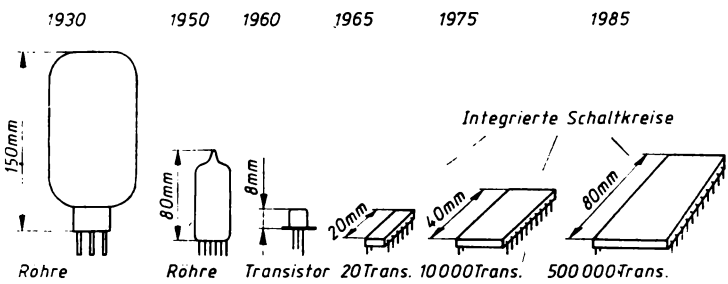


Bild 2. Größenvergleich elektronischer Bauelementegenerationen

herstellbar sind und einen ständig größeren Integrationsgrad aufweisen. Bereits 1970 existieren etwa 30 verschiedene Halbleitertechnologien und -techniken für die Produktion derartiger elektronischer Bauelemente. 1985 sind es schon über 500000 Transistoren, die man auf einem einzigen Schaltkreis integriert (Bild 2).

Ein Ende dieser Entwicklung ist vorerst nicht abzusehen, denn für die Steigerung des Integrationsgrades und der Leistungsfähigkeit der integrierten Schaltkreise sind die physikalischen Grenzen noch nicht erreicht.

1.3. Der Rechenautomat und seine Fortschritte

Verfolgt man die Geschichte der Elektronik sowie die daraus hervorgegangenen technischen Entwicklungen und Leistungen, so stellt man fest, daß sich auf der Basis der jeweils vorhandenen Bauelementetechnologien (z. B. Röhre, Transistor, niedrig integrierter Schaltkreis, hochintegrierter Schaltkreis) bestimmte Etappen oder «Generationen» von elektronischen Geräten und Einrichtungen herausbildeten. Bei jeder neuen Generation wurden wesentliche Verbesserungen ihrer technischen Eigenschaften und Parameter erreicht, möglichst die Vorteile der alten Systeme übernommen und bestimmte Nachteile ausgeschaltet. Besonders eindrucksvoll widerspiegelt sich diese Entwicklung auf dem Gebiet der elektronischen Rechenautomaten, und, wie sich noch zeigen wird, auch in der Mikrorechentechnik.

Die «maschinelle» Berechnung oder – in noch höherer Form – die «automatisch» ablaufende Lösung von numerisch-mathematischen Problemen war schon seit Jahrhunderten ein faszinierender Wunschtraum der Menschen. Die Konstruktion der ersten *Rechenmaschinen* erfolgte bereits im 17. Jh. Damals und auch in der Folgezeit standen nur mechanische Mittel für ihre Realisierung zur Verfügung. *W. Schickardt* und *B. Pascal* waren um 1623 bzw. 1642 die ersten, die die Idee einer mechanischen Rechenmaschine hatten und auch verwirklichten; *G. W. v. Leibniz* konnte bereits 1673 auf einer Ausstellung mit seiner handangetriebenen Rechenmaschine Addition, Subtraktion, Multiplikation und Division vorführen. Diese Maschinen wurden nun stetig verbessert, vervollkommnet und Anfang unseres Jahrhunderts sogar mit elektrischem Antrieb versehen (elektromechanische Rechenmaschine).

Die ersten bahnbrechenden Ideen für einen wirklichen *Rechenautomaten*, d. h. einen programmgesteuerten Rechner, hatte 1832 *Ch. Babbage*. Seine geplante Maschine sollte das menschliche Rechnen nach-

ahmen. Zur Programmspeicherung wollte er das Prinzip des Lochkartenbandes von *J.-M. Jacquard* (dem Erfinder der automatisierten Webmaschine) nutzen. Die in die Zukunft weisende Konstruktion von *Babbage* enthielt bereits die für einen heutigen Rechenautomaten wichtigsten Funktionseinheiten. Eine mechanische Realisierung mußte jedoch zur damaligen Zeit von vornherein ohne sichtbaren Erfolg bleiben, da die entsprechende Technik viel zu kompliziert, aufwendig und störanfällig war.

Die von ihm entworfenen entscheidenden Grundprinzipien blieben lange Zeit vergessen. Sie wurden jedoch wieder aufgegriffen, als die Elektronik einen solchen Stand erreicht hatte, daß das Prinzip eines programmgesteuerten Rechners, das bisher als mechanisch funktionierend gedacht war, sich elektronisch realisieren ließ. Es war *K. Zuse*, der 1934 in Deutschland das erste technische Konzept eines im binären Zahlensystem arbeitenden programmgesteuerten Rechenautomaten vorstellte. Es erwies sich jedoch zunächst wegen der aufwendigen mechanischen Konstruktionen ebenfalls als praktisch undurchführbar (Rechner «ZUSE-1»). Erst auf der Basis von elektromechanischen Relais entstand 1941 ein funktionsfähiger Rechenautomat («ZUSE-3»). Fast zur gleichen Zeit wurde in den USA von *H.-H. Aiken* ein ähnlicher, jedoch beträchtlich größerer, programmgesteuerter Rechner entworfen und 1944 in Betrieb genommen.

Aber auch die theoretischen Grundlagen der «Rechentechnik» konnten geschaffen und erweitert werden. Unter anderem entwarf *A. M. Turing* 1936 das theoretische Konzept einer allgemeingültigen Struktur, Funktion und mathematischen Beschreibung eines «abstrakten» Rechenautomaten, die sogenannte Turingmaschine. Dabei gewann er grundlegende Erkenntnisse über die «Berechenbarkeit» von mathematischen Problemstellungen. Mit der Entwicklung der Rechentechnik hatte in der Folgezeit zweifellos auch die numerische Mathematik ihren großen Aufstieg. Sie lieferte Grundprinzipien und Regeln für die verschiedenartigsten numerischen Verfahren, wie Reihenentwicklungen, Integrationen, Iterationen usw., zur Lösung mathematischer Aufgaben, die für einen Rechner aufbereitet werden sollten.

Die ersten praktisch realisierten Rechenautomaten waren infolge ihrer elektromechanischen Lösung im Grunde genommen nur «halbelektronische» Rechner. Sie arbeiteten zwar nach dem Prinzip einer Programmsteuerung, aber relativ langsam und schwerfällig, zudem waren sie wegen der vielen Relaiskontakte sehr störanfällig. Erst die verbesserte Technik der Elektronenröhren ermöglichte die Realisierung wirklich praktisch verwendbarer elektronischer Rechenauto-

maten. Tatsächlich wurden bereits kurz nach dem zweiten Weltkrieg die ersten Röhrenrechner gebaut und in Betrieb genommen. Weitere Verbesserungen betrafen die innere Organisation der elektronischen Rechner. *J. v. Neumann* entwarf 1946 das Konzept der gesteuerten Interpretation von Binärzahlen, d.h. den sogenannten speicherprogrammierten *Digitalrechner*. Nach diesem Prinzip arbeiten heute alle elektronischen Digitalrechner, einschließlich der Mikrorechner. Das Zeitalter der elektronischen Rechenanlagen war angebrochen.

Die nachfolgend auf dieser technischen und theoretischen Basis entwickelten elektronischen *Rechner der 1. Generation* ermöglichten bereits recht umfangreiche numerische Berechnungen. Sie waren jedoch nur sehr umständlich zu programmieren, arbeiteten unzuverlässig und hatten einen voluminösen Aufbau sowie einen riesigen Energieverbrauch (Bild 3).

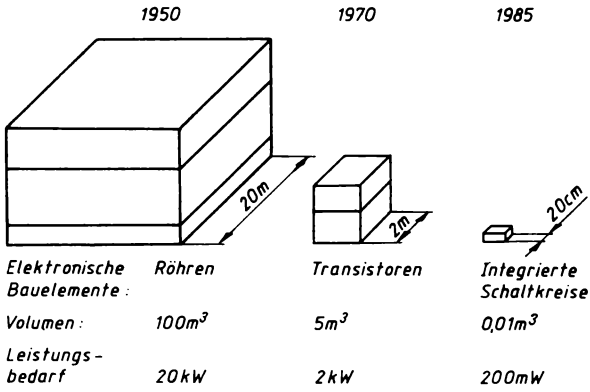


Bild 3. Raum- und Leistungsbedarf von Digitalrechnern verschiedener Generationen (bei etwa gleicher Rechenleistung)

Die *2. Generation* elektronischer Rechenautomaten mit wesentlich verbesserten Eigenschaften erreichte ihren Höhepunkt auf der Grundlage von Transistoren, die, erstmals auf Leiterplatten angeordnet, eine neue, viel produktivere Herstellungstechnologie und höhere Zuverlässigkeit der Rechner ermöglichten. Die Programmierung wurde wesentlich erleichtert. Erste höhere *Programmiersprachen* (FORTRAN, ALGOL) entstanden und erste leistungsfähige Steuerprogramme, sogenannte Betriebssysteme, ermöglichten eine effektivere Nutzung der Rechenanlagen.

Die *3. Generation* der elektronischen Rechner schließlich war gekennzeichnet vom Einsatz integrierter Schaltkreise, vorerst niedrig integriert, später durch die technische Weiterentwicklung mit höherem Integrationsgrad. Die Leistungsfähigkeit dieser Rechner wuchs stetig auf der Grundlage neuerer Verarbeitungskonzepte und Nutzungsformen sowie anderer technischer Verbesserungen. Der Teilnehmer- und Teilhaberbetrieb (mehrere Nutzer teilen sich den Rechner), der Dialogbetrieb (Mensch-Rechner-Kommunikation), die elektronische Datenfernverarbeitung (Verbindung Rechen- und Nachrichtentechnik), der Betrieb von Rechnernetzen sowie der Aufbau komplexer Daten- und Informationsbasen (Datenbanken) in Verbindung mit modernen Kommunikationstechnologien sind wichtige Merkmale der heutigen Rechentechnik.

1.4. Herausbildung der Mikrorechentechnik

Führende Fachleute erwarteten während dieser Weiterentwicklungen eine vierte Rechnergeneration mit neuen, noch unbekannten Qualitäten und riesigen zentralisierten Rechenleistungen. Wie nicht selten in der Geschichte der Technik, verlief die Entwicklung in eine ganz andere Richtung: nicht «noch größer» und «noch leistungsfähiger», sondern vorerst einmal «kleiner», sogar «viel kleiner» sollten die Rechner werden. Ende der 60er Jahre gelang es auf der Grundlage neuer Bauelementetechnologien der Mikroelektronik, den Integrationsgrad von integrierten Schaltkreisen so weit zu erhöhen, daß es technisch möglich wurde, die zentrale Steuer- und Recheneinheit, das Herzstück eines Digitalrechners (auch Prozessor genannt), sowie einen Programmspeicher auf je einem einzigen integrierten Schaltkreis von wenigen Quadratmillimetern Fläche unterzubringen. 1971 stellte die amerikanische Firma «Intel» den ersten Mikroprozessor und das Konzept eines Mikrorechners vor; beide zwar mit nur minimalem Funktionsumfang und im Vergleich zu einem konventionellen Digitalrechner mit sehr «bescheidenen» Leistungsmerkmalen, aber mit vollkommen neuer Technologie in der Herstellung. Volumen und Energieverbrauch waren sehr klein, Einsatz- und Entwicklungsmöglichkeiten aber äußerst breit. In den nächsten Jahren erfaßte eine stürmische Entwicklung nahezu alle Halbleitertechnologien und -techniken. Mikroprozessoren und Halbleiterspeicher wurden vervollkommnet, neue Realisierungstechniken eingeführt, die verschiedenartigsten Typen entstanden, und es gelang sogar, nicht nur den Mikroprozessor, sondern einen kompletten Digitalrechner auf einem ein-

zigen integrierten Schaltkreis unterzubringen (Einchiprechner). 5 Jahre später existierten auf der Welt etwa 50 verschiedene Mikroprozessor- bzw. Mikrorechnertypen. Nach weiteren 10 Jahren war die Typenvielfalt im internationalen Maßstab für einen einzelnen Entwickler oder Nutzer fast nicht mehr auswertbar.

Überblickt man die relativ kurze historische Entwicklung der Mikrorechentechnik bis zum heutigen Stand, so kann man zwar kürzere, aber ähnliche Etappen oder «Generationen» wie bei konventionellen elektronischen Rechenanlagen feststellen. Die einzelnen zeitlichen Entwicklungsabschnitte der Mikrorechentechnik sind hierbei durch folgende Merkmale gekennzeichnet:

1. Generation (ab 1970)

Einfache Halbleitertechnologie, etwa 1000 Transistoren für jeden integrierten Schaltkreis, geringer Funktionsumfang, wenig leistungsfähige Programmierung, geringe Verarbeitungsgeschwindigkeit, Umfang des Programmspeichers etwa 10000 Befehlseinheiten.

2. Generation (ab 1975)

Verbesserte Halbleitertechnologien, bis 10000 Transistoren je Schaltkreis, Verarbeitungsleistung etwa 10mal größer, verbesserte Programmiermöglichkeiten, effektive Programmiersysteme, Herausbildung verschiedenartiger Betriebssysteme, Umfang des Programmspeichers etwa 100000 Befehlseinheiten.

3. Generation (ab 1980)

Sehr hoch integrierende Technologie mit 10000 bis über 100000 Transistoren je Schaltkreis, Mikroprozessorbauart und -funktion (Architektur) der Programmierung angepaßt, mehrere effektive höhere Programmiersprachen, Schaltkreisreihen für Multiprozessor- bzw. Multirechnersysteme, Mikrorechnernetze, Umfang des direkt ansprechbaren Befehlsspeichers mehrere Millionen Befehlseinheiten.

Die realisierten Mikrorechner der einzelnen Entwicklungsabschnitte unterscheiden sich um Größenordnungen in ihrer Leistungsfähigkeit. Ein besonderer Umstand liegt dadurch vor, daß Geräte und Einrichtungen, die auf den Mikrorechnern der einzelnen Zeitabschnitte basieren, gegenwärtig und sicher auch in der nächsten Zeit nebeneinander bestehen werden. Für den völligen Verschleiß der Geräte aus den ersten Entwicklungsabschnitten waren die bisher vergangenen Zeiträume ganz einfach viel zu kurz.

Heute arbeiten Forschung und Entwicklung an Problemen der weiteren Erhöhung des Integrationsgrades, der noch größeren Komplexität der Mikrorechner-Schaltkreise und an neuen Architekturen von Mikroprozessoren und Mikrorechnern. Die Anforderungen der Mikrorechentechnik und deren hoher technologischer Stand haben auch Auswirkungen auf die übrigen Bereiche der Elektronik. Hier ergibt sich die Notwendigkeit, Methoden und Verfahren der Mikro-Miniaturisierung zu verallgemeinern. Insbesondere gilt dies für jene elektronischen Bauelemente, die einer Integration bisher weniger oder nicht zugänglich waren.

1.5. Von der Aufgabe zum Programm

Die vorangegangenen Ausführungen haben gezeigt, mit welcher Dynamik und welchem Tempo sich die Elektronik, insbesondere die Mikroelektronik und auf ihrer Basis die Mikrorechentechnik historisch entwickelt haben. In der Mikrorechentechnik wie auch in der konventionellen Rechentechnik gibt es jedoch noch Prozesse, auf die bisher nur wenig eingegangen wurde, nämlich die Programmierung und alle damit zusammenhängenden Fragen der Programmherstellung. Hier liegen jedoch andere «Methoden» und «Technologien» zugrunde als beispielsweise in der Produktion elektronischer Bauelemente sowie der daraus gefertigten Baueinheiten eines Mikrorechners. Da sich beide Arbeiten wesentlich im Arbeitsgegenstand unterscheiden, bezeichnet man die Gerätetechnik, die elektronischen Einrichtungen, die elektronischen Bauelemente, d.h. die Gesamtheit der systemtechnischen Elemente eines Rechners, auch als Hardware und alle Produkte, die mit der Programmierung zusammenhängen und meist «auf Papier» und «am Schreibtisch» entstehen, als Software.

Während die theoretischen und praktischen Voraussetzungen der Hardware der Mikrorechner sehr schnell einen relativ hohen Stand erreicht haben, gilt dies für die Mittel, Methoden und Verfahren der Softwareherstellung nicht in dem Maße. Mit der Mikrorechentechnik hat sich auch dieses Gebiet weiterentwickelt, wobei zusätzlich auf die bei konventionellen Rechnern vorhandenen Erfahrungen zurückgegriffen werden konnte. Es wurden auch neue Programmiersprachen sowie effektivere Programmiersysteme entwickelt und zudem leistungsfähigere Methoden, Verfahren und Technologien im Programmentwurf (Software engineering) eingeführt. Insgesamt nimmt jedoch der Wirkungsgrad der Software weit weniger zu als der der

Hardware. Eine Folge daraus ist, daß heute beim praktischen Einsatz von Mikroprozessoren und Mikrorechnern die Kosten für Programmentwicklungen gegenüber den Aufwendungen für die elektronisch-technischen Einrichtungen überwiegen (Bild 4).

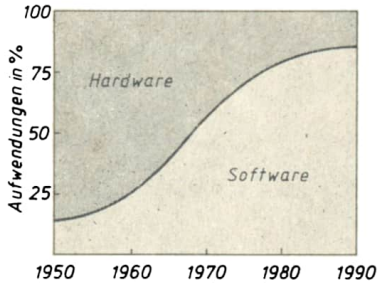


Bild 4. Zeitliche Entwicklung der Hard- und Software-Aufwendungen bei Digitalrechnern

Wie noch gezeigt wird, kann man einen Mikrorechner aus wenigen hochintegrierten Schaltkreisen und einigen Zusatzeinrichtungen aufbauen, aber damit ist er noch nicht arbeitsfähig. Erst ein der jeweiligen Aufgabenstellung entsprechendes Programm, das erdacht, aufgeschrieben, gespeichert, übersetzt, korrigiert, getestet werden muß, bevor es zu benutzen ist, macht den Mikrorechner zu dem, was er sein soll, zu einem «intelligenten» elektronischen System mit eigenem «Leben». Häufig sind auch bestimmte Algorithmen zu entwerfen. Die Entwicklung eines solchen Programms ist dann ein schöpferischer Vorgang, und Rationalisierungen derartiger Prozesse sind nun einmal nur sehr schwierig durchzuführen. Dieses «Mißverhältnis» zwischen Hardware und Software bedingt ein spezielles Herangehen bei der Lösung von Aufgabenstellungen in der Mikrorechentechnik, das darin besteht, den Problemen der Softwareherstellung größte Aufmerksamkeit zu widmen. In den nächsten Jahren, vielleicht sogar Jahrzehnten, wird die Notwendigkeit bestehen, auf dem gesamten Gebiet der Programmierung Fortschritte in Theorie und Praxis zu erreichen, um die Programmiereffektivität zu erhöhen, neue Erkenntnisse, Mittel, Methoden und Verfahren der Softwareentwicklung zu erarbeiten sowie die entsprechenden mathematischen Grundlagen zu vertiefen und zu vervollkommnen.

Nachdem wir uns einen groben Überblick über die Wurzeln der Mikrorechentechnik und die benachbarten Fachgebiete verschafft

haben, sollen im nachfolgenden Abschnitt die mikroelektronischen Grundlagen behandelt werden, die für die Mikrorechentechnik von Interesse sind. Das ist jedoch nur ein, wenn auch wichtiger Teil der Mikroelektronik.

Auf weitere Gebiete wird im Literaturverzeichnis verwiesen [1, 2, 3, 4].

2. Grundlagen der Mikroelektronik

2.1. Begriffsbestimmungen

Zunächst sollen einige Begriffe erläutert werden, um eine fundierte Ausgangsposition zu schaffen. Unter den *mikroelektronischen Grundlagen* der Mikrorechentechnik sind die Techniken und Verfahren zur Herstellung der Elementarbausteine sowie Funktionsgruppen des Mikrorechners zu verstehen. Hierzu gehören die physikalischen Grundlagen der elektronischen Bausteine, das Gebiet der Halbleitertechnologien und die Gerätetechnologien.

Die *Elektronik* selbst befaßt sich – etwas allgemeiner formuliert – mit der Bewegung und Steuerung von geladenen Teilchen in Festkörpern, Flüssigkeiten, Gasen und im Vakuum sowie den entsprechenden Anordnungen, Einrichtungen, Schaltungen und Systemen, in denen diese Effekte ausgenutzt werden. Die *Mikroelektronik* ist ein Teilgebiet der Elektronik, dessen Ziel darin besteht, diese Anordnungen, Einrichtungen, Schaltungen usw. zu miniaturisieren (verkleinern), zu integrieren (zusammenzufassen), deren Energieverbrauch, Volumen und Gewicht zu vermindern, die Leistungsfähigkeit, Zuverlässigkeit zu erhöhen sowie eine ökonomische Herstellung und einen flexiblen Einsatz zu garantieren. Als Elementarbausteine der Elektronik bezeichnet man die *elektronischen Bauelemente*. Hierbei handelt es sich um nach bestimmten Gesichtspunkten in sich abgeschlossene und in der Regel nicht weiter zerlegbare Anordnungen oder Elemente mit ausgewählten elektrischen und elektronischen Eigenschaften. Mehrere solcher elektronischen Bauelemente, miteinander verknüpft, ergeben eine *elektronische Schaltung*. Der Begriff *Funktionselement* charakterisiert die schaltungstechnische Funktion oder eine bestimmte herausragende schaltungstechnische Wirkung eines elektronischen Elements bzw. einer elektronischen Baugruppe in seiner räumlichen Umgebung. Mehrere Funktionselemente zusammengefaßt ergeben eine *Funktionsgruppe*. Die Bezeichnungen Bauelement, Funktionselement bzw. Schaltung und Funktionsgruppe

werden häufig in der Elektronik inhaltlich unterschiedlich, z.T. aber auch sinnverwandt gebraucht.

Eine Einteilung der Elektronik in einzelne Gebiete wird oft nach anwendungstechnischen Gesichtspunkten vorgenommen. Steht bei elektronischen Bauelementen, Schaltungen oder komplexen Systemen der informationelle Aspekt, d.h. die Erzeugung, Gewinnung, Übertragung, Speicherung, Verarbeitung oder Verwertung von Informationen, im Mittelpunkt, dann wird eine Zuordnung zur sogenannten *Informationselektronik* getroffen. Steht demgegenüber der energetische Aspekt, d.h. die Erzeugung, Übertragung, Verteilung von Elektroenergie und damit zusammenhängende Steuerungsprozesse, im Vordergrund, dann handelt es sich um Elemente und Systeme der *Leistungselektronik*. Die Mikrorechentechnik gehört demnach zur Informationselektronik.

Eine Klassifizierung in diese beiden wichtigen Gebiete ist sehr häufig. Es lassen sich jedoch noch eine Reihe anderer Möglichkeiten der Einteilung anführen. Meist sind die Grenzen zwischen den einzelnen Gebieten nur schwer zu ziehen. Beispielsweise werden bestimmte Gebiete der Physik oder physikalische Vorgänge zur Bezeichnung herangezogen, wie dies an den Gebieten der «Optoelektronik», «Akustoelektronik», «Magnetoelektronik» und «Kryoelektronik» sichtbar wird. Auch das betreffende Einsatzfeld wird oft als Klassifizierungsmerkmal benutzt, wie die Begriffe «KFZ-Elektronik», «Heimelektronik», «Raumfahrtelektronik», «Medizinische Elektronik», «Rechenelektronik» u.a. verdeutlichen. Für die Gebiete sind dann häufig bestimmte elektronische Verfahren und Zielstellungen typisch.

2.2. Das elektronische Bauelement

In der Elektronik unterscheidet man zwei große Gruppen von Bauelementen: *aktive* und *passive*. Aktive elektronische Bauelemente benötigen zu ihrer Funktion die Zuführung eines bestimmten Energiebetrages (Hilfsenergie), passive dagegen nicht. Wie der Name bereits sagt, beeinflussen die Bauelemente das «elektronische Geschehen» in einer Schaltung entweder «aktivierend» (verstärkend) oder «passiv». Aktive Bauelemente sind z.B. Elektronenröhren, Transistoren und Thyristoren, passive Bauelemente sind Widerstände, Spulen, Kondensatoren u.a.

Außer diesen beiden Arten gibt es in der Elektronik noch eine Reihe von «Hilfs»-Bauelementen, die man, je nach Funktionsweise, zu den aktiven oder passiven rechnen kann. Hierzu gehören u.a. Fas-

sungen oder Befestigungselemente für die entsprechenden elektronischen Bauteile, Trägermaterialien (z.B. Leiterplatten) sowie alle Arten von Steckern, Steckverbindern, Schalter. Auch Anzeigeelemente, z.B. aus der Optoelektronik, Sensoren (Meßfühler) und Einrichtungen zur Energieversorgung sind wichtige Bauelemente der Elektronik, die oft wieder selbst recht komplizierte elektronische Systeme darstellen.

2.3. Elektronische Schaltungen und Systeme

Werden mehrere gleiche oder verschiedenartige elektronische Bauelemente zu Strukturen, Netzen oder – allgemeingültiger – zu Systemen elektrisch verbunden, so entsteht eine elektronische Schaltung bzw. ein elektronisches System. Der Schaltungsentwurf erfolgt nach bestimmten Gesichtspunkten, die sich aus der praktischen Anwendung und Zielstellung ableiten. Entwurfsmethoden und entsprechende Theorien sind Inhalt der *Schaltungstechnik*. Seit dem Aufkommen der Mikroelektronik sind jedoch die Grenzen zwischen «Bauelement» und «Schaltung» verwischt. Zwischen beiden existiert eine Beziehung, auf die wir noch näher eingehen werden. Entstammen die Elementarbausteine der Mikroelektronik, so spricht man auch von «mikroelektronischen Bauelementen» bzw. «mikroelektronischen Schaltungen».

Jede Schaltung, oder allgemein, jedes elektronische System, setzt sich, den Regeln der Elektronik entsprechend, aus funktionellen Elementargliedern zusammen. In der klassischen Elektronik (Makroelektronik) entspricht jedes dieser Elementarglieder (bis auf wenige Ausnahmen der verteilten Systeme, z.B. Leitungen) einem diskreten und «anfaßbaren» elektronischen Bauelement mit einer bestimmten elektrischen bzw. elektronischen Eigenschaft. Als Beispiele seien der Transistor oder ein diskreter Widerstand genannt (Bild 5). Demgegenüber liegen in der Mikroelektronik die Verhältnisse anders, da hier eine große Zahl von Funktionselementen, d.h. Bauelemente im Sinne der klassischen Elektronik, in einem Integrationsprozeß zu *einem* mikroelektronischen Bauelement vereinigt werden. Das Prinzip der Integration geht davon aus, Teile einer elektronischen Schaltung oder eine komplette Schaltung auf einem einzigen Bauelementeträger, meist einem Halbleitermaterial, anzuordnen und als Ganzes – also nicht mehr Trennbares – herzustellen (Bild 6). Der Vorgang wird als «Schaltungsintegration», das Ergebnis als *Integrierte Schaltung* oder *Integrierter Schaltkreis* bezeichnet. Ein solcher Schaltkreis ist wieder elektronisches Bauelement, d.h. Elementarbaustein der Elektronik

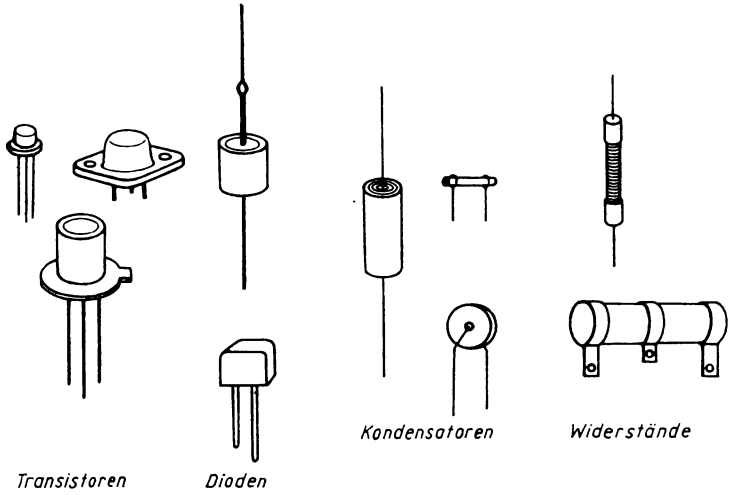


Bild 5. Beispiele diskreter elektronischer Bauelemente

bzw. elektronischer Schaltungen, und die oben erwähnte Beziehung zwischen Bauelement und Schaltung wird sichtbar.

Bei der Frage nach der elektronischen Eigenschaft einer integrierten Schaltung geht man grundsätzlich nur noch von der Gesamtwirkung des Bauelements aus (Methode der Black Box). Das bedeutet, daß das elektronische Bauelement als ein nach innen nicht zugängliches, «geschlossenes» System, also als «schwarzer Kasten» aufgefaßt wird.

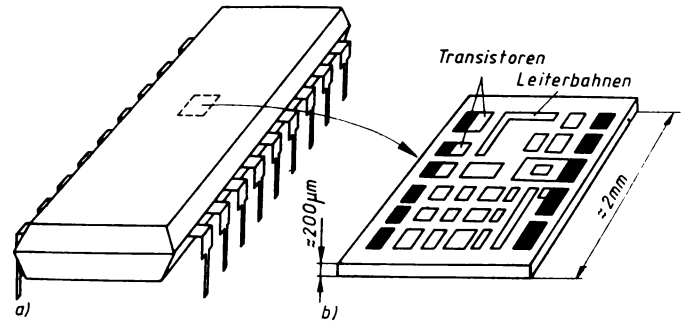


Bild 6. Integrierter Schaltkreis
a) komplettes Bauelement b) Halbleiterscheibe (Chip)

Für die praktische Anwendung genügt daher eine «äußere» und allgemeine Funktionsbeschreibung. Eine solche Darstellung der elektrischen, elektronischen und mechanischen Parameter von integrierten Bauelementen erfolgt in der Regel durch die vom Hersteller mitgelieferten Kenn- und Datenblätter. Die innere Wirkungsweise und der Aufbau der integrierten Schaltkreise sowie die dabei ablaufenden Prozesse spielen für diesen Fall beim Anwender eine untergeordnete Rolle. Insgesamt gesehen schließt demnach die Funktionsbeschreibung eines integrierten Schaltkreises nach schaltungstechnischen Gesichtspunkten einen vollzogenen Abstraktionsvorgang ein.

Solche allgemein beschriebenen Bauelemente der Elektronik lassen sich zu noch komplexeren elektronischen Schaltungen und Systemen vereinigen. Dieses «Verknüpfen» von Schaltkreisen – nun auf wesentlich «höherer Ebene» – ist viel einfacher zu realisieren und formalisieren als ein Entwurf des gleichen Systems mit nicht integrierten Bauelementen. Dies ist einer der entscheidenden Vorteile von mikroelektronischen Bauelementen und Schaltungen.

Die Schaltungsintegration wird in der Praxis je nach gewünschtem technischem und ökonomischem Ziel mit sehr verschiedenem Integrationsgrad durchgeführt (Tabelle 1). Unter *Integrationsgrad* ist die Anzahl der Funktionselemente je Volumen oder Fläche, also die «Schaltungsdichte», «Strukturfeinheit» oder die «Kleinheit» einer bestimmten Schaltungsfunktion des integrierten Schaltkreises zu verstehen. Historisch gesehen sind die niedrig integrierten Schaltkreise als erste entstanden, danach folgte die Entwicklung zu stetig höherem Integrationsgrad. Heute liegen in der Elektronik Bauelemente aller

Tabelle 1. Aufstellung der in der Mikroelektronik verwendeten Integrationsstufen

SSI	Kleinintegration (small scale integration): bis etwa 100 Transistoren je Chip; Chipgröße: 1 bis 10 mm²
MSI	Mittelintegration (medium scale integration): bis etwa 1000 Transistoren je Chip; Chipgröße: bis 20 mm²
LSI	Großintegration (large scale integration): bis etwa 10000 Transistoren je Chip; Chipgröße: bis 50 mm²
VLSI	Höchstintegration (very LSI): um 100000 und mehr Transistoren je Chip; Chipgröße: 10 bis über 100 mm²

Integrationsstufen vor. Die Auswahl des Einsatzes erfolgt nach ökonomischen und schaltungstechnischen Gesichtspunkten, so daß prinzipiell eine elektronische Schaltung oder ein komplexes elektronisches System Bauelemente mit sehr unterschiedlichem Integrationsgrad enthalten können. In der Mikrorechentechnik werden jedoch vorwiegend hochintegrierte Schaltkreise eingesetzt.

2.4. Analoge und digitale Technik

Eine wichtige Unterteilung der elektronischen Schaltungen ergibt sich aus ihren verschiedenen inneren Arbeitsweisen und den dabei verwendeten Signalarten [5, 6]. Man unterscheidet

- Analogschaltungen
- Digitalschaltungen
- Analog/Digitalschaltungen.

Die *Analogschaltungen* arbeiten mit analogen (stetig veränderlichen) elektrischen Signalen, d.h., die zur Darstellung der Informationsparameter benötigte physikalische Größe (Strom oder Spannung) durchläuft kontinuierlich den Wertebereich. Bei Digitalschaltungen werden digitale (ziffernmäßige) Signale und dabei vor allem binäre, also aus zwei Einheiten bestehende, benutzt, um die Informationen darzustellen. Hier nehmen Strom und Spannung im praktischen Betrieb nur bestimmte, diskrete Amplitudenwerte an. Ein besonderer Typ von Schaltungen, die Analog-Digital-Wandler und Digital-Analog-Wandler, erlaubt einen Übertritt zwischen beiden Schaltungsarten. Digitalschaltungen lassen sich besonders günstig in integrierter Form herstellen. Sie enthalten im Unterschied zu Analogschaltungen in der Regel keine besonderen Widerstände, Kondensatoren und Induktivitäten, die sich nur sehr aufwendig, schwierig bzw. überhaupt nicht in einen Schaltkreis integrieren lassen. In der Elektronik und deren Anwendungsbereichen besteht daher heute die Tendenz, alle Prozesse und Systeme in digitaler Technik zu verwirklichen. In der Mikrorechentechnik werden fast ausschließlich digitale Schaltungen benutzt.

Die Einteilung in «analog» und «digital» ist in der Elektronik sehr weitreichend und hat zu einer Reihe von Begriffsbildungen geführt. Sie bezieht sich nicht nur auf Signale und elektronische Bauelemente, Schaltungen und alle Arten elektronischer Systeme, sondern auch auf Entwurfsverfahren und Schaltungstechniken sowie Einsatzgebiete. Wir unterscheiden beispielsweise analoge und digitale integrierte

Schaltkreise, analoge und digitale Systeme, analoge und digitale Schaltungstechnik, analoge und digitale Meßgeräte usw. Auch in der elektronischen Rechentechnik kam es zur Einteilung in Analogrechner und Digitalrechner. Auf diese Problemkreise wird im Abschnitt 3.2. eingegangen. Zunächst wollen wir Aufbau, Eigenschaften und Herstellung der in der Mikrorechentechnik verwendeten integrierten Schaltkreise und deren Grundbestandteile etwas genauer betrachten.

2.5. Der Transistor

Das grundlegende elektronische Bauelement ist der Transistor. Er kann einzeln und auch als Grundbaustein in einem Halbleiterblock zum Aufbau einer elektronischen Schaltung aus einer großen Zahl von Transistoren in einem integrierten Schaltkreis Einsatz finden. In der Mikrorechentechnik ist nur der zweite Fall von Interesse. Die dort verwendeten integrierten Schaltkreise bestehen, schaltungstechnisch betrachtet, fast ausschließlich aus vielen Transistoren.

Der Rohstoff
Transistoren und die entsprechenden integrierten Schaltkreise werden aus Halbleitermaterialien wie Silicium, Germanium oder Galliumarsenid hergestellt, die äußerst hohe Reinheit und nahezu fehlerfreie Kristallstruktur aufweisen müssen. Größte Bedeutung in der Auswahl der Halbleitermaterialien hat das Silicium. Es weist außerordentlich günstige technologische sowie elektronische Eigenschaften auf und steht außerdem in der Natur in ausreichendem Maß in gebundener Form als Siliciumdioxid (Quarz, Quarzsand) und Silikat (Feldspat, Kaolin, Ton, Glimmer) zur Verfügung. Mit verschiedenartigsten – allerdings recht aufwendigen – technologischen Verfahren ist man heute in der Lage, Silicium ultrarein, d.h. mit sehr hohen Reinheitsgraden herzustellen.

Das Modell des Transistors
Der Grundaufbau eines Transistors entspricht einer Anordnung aus Halbleitermaterialien mit unterschiedlichen Leitfähigkeiten bzw. Leitfähigkeitstypen. Bemerkenswert ist, daß die Leitfähigkeit in einem bestimmten Bereich des Halbleitermaterials mit einer von außen anliegenden elektrischen Spannung oder einem elektrischen Strom verändert und damit gesteuert werden kann. Das Modell des klassischen Transistors ist im Bild 7 dargestellt. Der Halbleiterblock besteht aus verschiedenen Leitfähigkeitsbereichen (im Bild mit n, p, n gekenn-

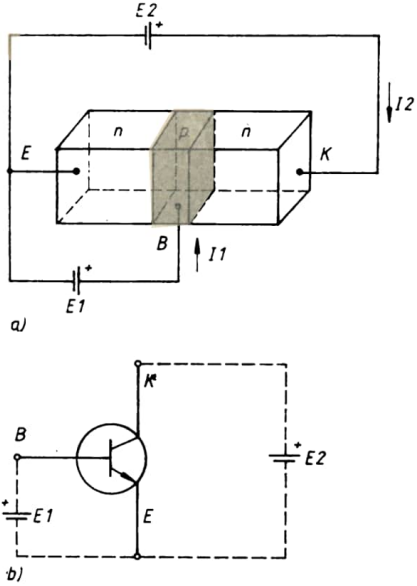


Bild 7. Modell des Transistors
a) Geometrie b) standardisiertes Schaltsymbol

zeichnet). Der mittlere Teil des Transistors ist hierbei der erwähnte Bereich, in dem die Steuerung der Leitfähigkeit bzw. des Stromflusses stattfindet. Das Funktionsprinzip des Transistors ist folgendes: Die an den äußeren Klemmen des Transistors angelegten Spannungsquellen E_1 und E_2 entsprechen der Energieversorgung und bewirken einen Stromfluß I_1 und I_2 im Halbleitermaterial. Für die Funktion des Transistors ist es wesentlich, daß von den zwei fließenden Strömen einer der Steuerstrom bzw. die steuernde Größe ist (z.B. I_1) und jeweils der andere als gesteuerte Größe (z.B. I_2) fungiert. Auf diese Weise wird, ausgehend vom Steuerstrom, nahezu trägheitslos und mit nur sehr kleinen erforderlichen Steuerenergien, eine Verstärkung oder das Ein- und Ausschalten des gesteuerten Stroms erreicht. Der Transistor wirkt dadurch als elektronisches Verstärker- und Schaltelement.

Der Leitmechanismus

Um die physikalischen Vorgänge im Transistor besser zu verstehen, müssen wir den Leitmechanismus im Halbleiterkristall etwas näher

betrachten. Das Halbleitermaterial wird als kristallin vorausgesetzt. Zum Verständnis des Vorgangs muß man jedoch zunächst die Eigenschaften des reinen Halbleiters (Eigenhalbleiter) kennen [4]. Die elektrische Leitfähigkeit in einem superreinen Halbleitermaterial ist von der Temperatur abhängig; bei hohen Temperaturen ist es ein elektrischer Leiter, bei sehr niedrigen Temperaturen ein Nichtleiter oder Isolator. Letzteres trifft praktisch auch bei Zimmertemperatur zu.

Der Stromfluß in einem beliebigen Halbleiter kann im allgemeinen auf zweierlei Art und Weise stattfinden. Zum einen gibt es die sogenannte «Elektronenleitung». Im Halbleitermaterial frei bewegliche Elektronen ermöglichen den Stromfluß: einen Strom negativer Ladungsträger. Eine andere Form der Stromleitung ist die «Löcherleitung». Hier übernehmen sogenannte Defektelektronen oder «Löcher» die Funktion von Ladungsträgern. Defektelektronen muß man sich als Lücken im Elektronenbestand des Halbleiterkristalls vorstellen. Diese Lücken bewegen sich im Halbleiter infolge außen angelegter Spannungsquellen ähnlich den Elektronen, aber in entgegengesetzter Richtung und bewirken damit einen Strom scheinbar positiver «Ladungsträger». In Wirklichkeit verbirgt sich dahinter natürlich ebenfalls eine Elektronenbewegung. Wichtig ist festzuhalten, daß es sich – technisch betrachtet – einmal um einen Strom negativer und zum anderen um einen Strom positiver Ladungsträger handelt.

Um in einem sehr reinen und praktisch nichtleitenden Halbleiter gewünschte Leitungseffekte der genannten Art hervorzurufen, wird das Halbleitermaterial gezielt mit anderen Stoffen (vornehmlich Phosphor und Bor beim Halbleiter Silicium), nach ausgewählten technologischen Verfahren künstlich verunreinigt. Die Technik des Einbringens einer sehr kleinen Menge dieser sogenannten Störatome wird als *Dotierung* des Halbleiterkristalls bezeichnet. Zwei Arten von Störatomen sind hierbei von Interesse: die *Donatoren* und die *Akzeptoren*. Erstere bewirken nach der Dotierung einen bestimmten Überschuß an freien Elektronen (z.B. Phosphor in Silicium), während im zweiten Fall (z.B. Bor in Silicium), ein gezielter «Mangel» an Elektronen bzw. Fehlstellen (Lücken oder Löcher) im Halbleiter entstehen. Die so dotierten Halbleitermaterialien werden entsprechend als *n-leitend* (n negativ) oder *p-leitend* (p positiv) bezeichnet (Bild 8).

Ein Transistor oder ein entsprechender Halbleiterblock eines integrierten Schaltkreises setzt sich stets aus solchen verschiedenartig dotierten Halbleiterzonen oder -gebieten zusammen. Sie sind die entscheidenden Voraussetzungen für die Funktion von Transistoren und der darauf aufbauenden integrierten Schaltkreise. Mit Hilfe der angelegten Spannungsquellen wird die Dichte der Elektronen bzw. der

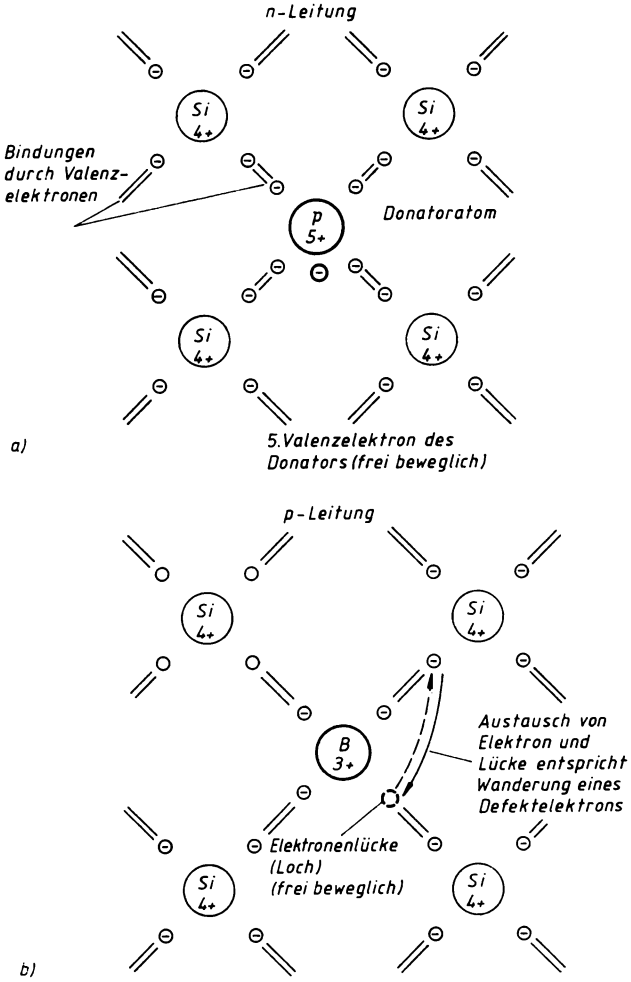


Bild 8. Störstellenleitung im Silicium-Halbleiterkristall
a) Elektronenleitung b) Löcherleitung
(Si Silicium, P Phosphor, B Bor)

Löcher in ausgewählten Bereichen des Kristalls gezielt verändert und damit die bereits erwähnte Steuerung des Stromflusses im Halbleiter bewirkt.

Unipolare und bipolare Technik

Auf der Basis der beiden möglichen Leitungsmechanismen Elektronenleitung und Löcherleitung wird auch eine wichtige Einteilung aller Transistorarten und der darauf aufbauenden Schaltkreistechnologien vorgenommen. Sorgt nämlich in dem betreffenden Gebiet des Transistors, in dem die Steuerung des Stromflusses vor sich geht, nur eine der beiden Leitungsarten – Löcher- *oder* Elektronenleitung – für den Stromfluß, so spricht man von *unipolaren* Transistoren bzw. Schaltkreisen. Tragen jedoch beide Leitungsarten – Löcher- *und* Elektronenleitung – zum Stromfluß bei, dann liegen *bipolare* Transistoren oder Schaltkreise vor. Es sei betont, daß nur der Stromfluß in der Steuerzone gemeint ist und nicht die statisch vorliegenden Dotierungsarten in den anderen Gebieten des Transistors. Die noch zu erläuternden Basistechnologien zur Herstellung der Transistoren und integrierten Schaltkreise ordnet man entsprechend dieser Klassifikation in Unipolartechniken oder Bipolartechniken ein.

Der wichtigste Vertreter der unipolaren Transistoren ist der Feldeffekttransistor, für bipolare der Diffusions- und Drifttransistor. Die Bezeichnungen widerspiegeln jeweils das innere Wirkungsprinzip. Feldeffekttransistoren werden im allgemeinen auch kurz MOS-Transistoren oder MOSFET (**m**etal-**o**xid-**s**emiconductor-**f**ield-**e**ffect-**t**ransistor) genannt, obwohl diese Bezeichnung ursprünglich für eine spezielle Variante der Feldeffekttransistoren gedacht war. Bipolare und MOS-Transistoren unterscheiden sich wesentlich in der Wirkungsweise, insbesondere in der Art der Steuerung des Stromflusses im Halbleiter. Bei Bipolartransistoren sorgt ein verschieden starkes Einströmen oder die «Diffusion» von Ladungsträgern in die aktive Steuerzone des Transistors für eine Steuerung (Bild 9a), während

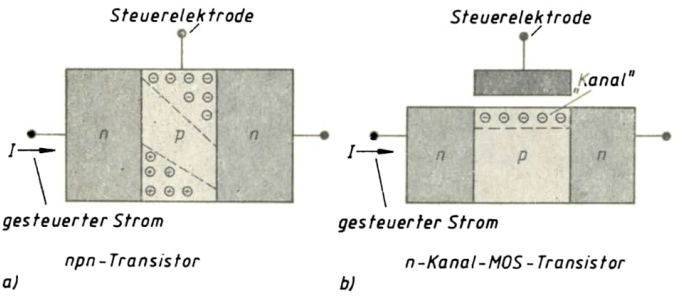


Bild 9. Vergleich bipolarer Transistor (a) und Feldeffekttransistor (b) (eindimensionales Modell)

dies bei Feldeffekttransistoren durch Influenz von Ladungsträgern (d.h. durch Ladungsverschiebung) über eine Verbreiterung oder Einschnürung eines leitenden «Kanals» im Halbleitermaterial bewirkt wird (Bild 9b). Hinsichtlich der Einzelheiten von Steuer- und Verstärkungsvorgängen in Transistoren sei auf die Literatur verwiesen [4, 5].

MOS-Transistoren arbeiten leistungsärmer als Bipolartransistoren, jedoch im allgemeinen auch wesentlich langsamer. Dies hat seine Ursache in den größeren Verzögerungszeiten für die inneren Ladungsträgertransporte. Beide Arten finden bei der Herstellung von integrierten Schaltkreisen vielfache Anwendung. MOS-Transistoren können in integrierter Form technologisch einfacher und kleiner hergestellt werden. Sie ermöglichen infolge ihrer hohen Packungsdichte, der niedrigeren Verlustleistungen sowie der relativ einfachen Fertigung, der damit verbundenen besseren Ausbeute und geringeren Kosten eine günstige Realisierung von sehr hoch integrierten Schaltkreisen. Bipolare Transistoren werden dagegen eingesetzt, wenn höhere Arbeitsgeschwindigkeiten der integrierten Schaltkreise gefordert sind.

2.6. Der integrierte Schaltkreis

Heute werden integrierte Schaltkreise vorwiegend in sogenannter Halbleiterblocktechnik gefertigt. Hierbei erfüllt eine Grundsubstanz der Bauelemente – ein kristalliner Halbleiterblock – zugleich die mechanische (Bauelementeträger) und die elektronische (Schaltung) Funktion. Bei dieser Technik wird die gesamte Schaltung, also die einzelnen Funktionselemente sowie alle Verbindungen, in einem Halbleiterkristall (einem Ausschnitt aus einem Einkristall) realisiert. Während der Herstellung wird der Kristall durch verschiedene Techniken physikalisch oder chemisch bearbeitet und präpariert. Dabei entstehen leitende Strompfade, Isolierschichten, Widerstände und aktive elektronische Elemente, wie Transistoren. Man bezeichnet diese Schaltungen auch als Festkörperschaltkreise oder *monolithische* integrierte Schaltungen und ihre Herstellung als monolithische Technik.

Realisierungstechniken

Für die Herstellung von integrierten Festkörperschaltkreisen haben bestimmte Technologien der Unipolar- und Bipolartechnik Bedeutung erlangt. Die wichtigsten der Unipolartechnik sind die N-MOS-, P-MOS- und CMOS-Technologie, in der Bipolartechnik die Stan-

Tabelle 2. Vergleich der wichtigsten Schaltkreistechnologien bei Mikrorechnern
(N-MOS: Negativ-MOS-Technologie; P-MOS: Positiv-MOS-Technologie; CMOS: Complementär-MOS-Technologie; Standard: Standard-Bipolartechnologie; I^2L = IIL: Integrierte Injektions-Logik; MOS – metal oxid semiconductor: Feldeffekttransistor)

Halbleitertechnik	unipolar			bipolar	
Schaltkreistechnologie	N-MOS	P-MOS	CMOS	Standard	I^2L
Integrationsgrad	hoch	mittel	hoch	niedrig	hoch
Arbeitsgeschwindigkeit	mittel	sehr langsam	mittel	sehr schnell	schnell
Leistungsverbrauch	gering	gering	sehr gering	hoch	sehr gering
Vorteile	einfache Herstellung	einfachste Herstellung	äußerst kleine Verlustleistungen	viele vorhandene Standardschaltkreise	vereinigt Vorteile von TTL- und MOS-Technik
Nachteile	–	komplizierte Stromversorgung	Herstellung verschiedener Typen von Transistoren	aufwendige Herstellung	schwierige Herstellung

dard- und I^2L-Technologie (Tabelle 2). Alle auf diese Weise hergestellten integrierten Schaltkreise finden breite Anwendung in der Mikrorechentechnik [2, 3, 5].

Außer der monolithischen Technik unterscheidet man in der Herstellung integrierter Schaltkreise noch die Schicht- und Hybridtechnik [1, 2]. Beide gehen von einzelnen, getrennt vorliegenden elektronischen Komponenten einer integrierten Schaltung aus, die in mehreren technologischen Schritten auf einem Bauelementeträger vereinigt werden. In der Mikrorechentechnik haben sie jedoch nur wenig Bedeutung.

Monolithische Schaltkreise werden ausschließlich nach der sogenannten Planartechnologie hergestellt (Bild 10). Dies bedeutet, daß der Halbleiterblock – in Wirklichkeit ein hauchdünnes Plättchen – nur von einer Seitenfläche aus technologisch bearbeitet wird (im Bild entspricht dies der oberen Seite). Ein Vorteil dieses Herstellungsverfahrens besteht darin, daß man gemeinsam alle Funktionselemente der integrierten Schaltung, wie Transistoren, Verbindungsleitungen

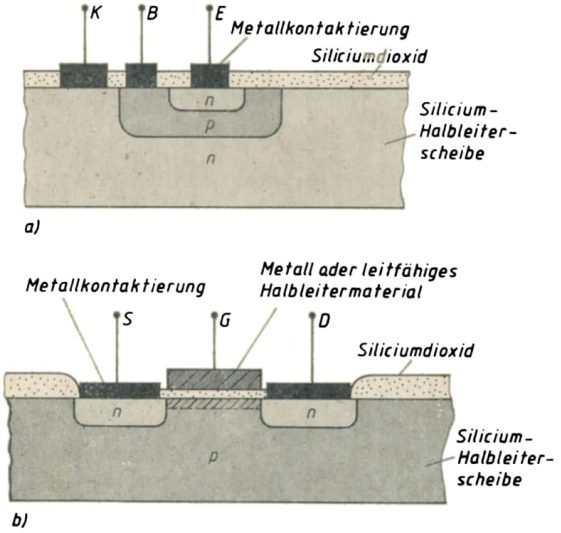


Bild 10. Aufbau eines Bipolartransistors (a) und Feldeffekttransistors (b) eines integrierten Schaltkreises bei Planartechnologie (Querschnitt, vereinfacht dargestellt; Anschlüsse: K Kollektor, B Basis, E Emitter, S Source, G Gate, D Drain)

usw., geometrisch auf der Oberfläche verteilt anordnen kann. Durch eine Folge von relativ einfachen technologischen Prozeßschritten werden sie, von der betreffenden Oberfläche der Halbleiterscheibe aus in die Tiefe des Materials vordringend, realisiert. Hierbei ist es üblich, nicht nur einen einzigen Schaltkreis, sondern eine größere Anzahl (einige 100) identischer Schaltungen auf einer Halbleiterscheibe unterzubringen und diese «kollektiv» herzustellen. In Verbindung mit der Planartechnik haben sich besondere Dotierungsverfahren und Methoden der Strukturierung sowie Präparation von Oberflächenschichten herausgebildet. Das wird im folgenden sichtbar.

Herstellungsschritte

Bei der Produktion von integrierten Festkörperschaltkreisen auf Siliciumbasis hat man wichtige technologische Prozesse zu unterscheiden (Bild 11).

– Herstellung des Halbleitermaterials:

Als Ausgangsrohstoff dient Quarz, aus dem durch Reduktion mit Kohlenstoff (Koks) das Rohsilicium (Reinheit etwa 96%) entsteht. Anschließende verfahrenstechnische Prozesse wandeln es in chemisch reines polykristallines Silicium um. In dieser Form ist es jedoch noch nicht als Halbleitermaterial verwendbar. Hierzu wird es in einen größeren Einkristall umgeschmolzen und die Reinheit in sogenannten Ziehprozessen [1] in sehr hohem Maß gesteigert.

Das Ergebnis der Ziehprozesse ist ein Silicium-Einkristall mit einem Durchmesser von 50 bis 150 mm und einer Länge bis zu einem Meter.

– Scheibenherstellung:

Vom Silicium-Einkristall werden etwa 300 µm dicke Scheiben abgesägt und einer umfangreichen Oberflächenbehandlung (Schleifen, Polieren, Läppen, Ätzen) unterzogen. So entstehen Silicium-Einkristallscheiben, sogenannte Wafer, mit einer Dicke von 100 bis 200 µm, die eine Kristalloberfläche mit weitgehend störungsfreiem und regelmäßigem Gitteraufbau aufweisen.

– Scheibenaufteilung:

Jede Scheibe wird in viele kleine Rechtecke eingeteilt. Diese Segmente, *Chip* oder *Mikrochip* genannt, tragen später eine integrierte Schaltung (Bild 12). Die Anzahl der integrierten Schaltungen je Scheibe ist die Losgröße. Infolge kleiner Oberflächenfehler der Kristallscheiben und Prozeßfehler verschiedenster Ursachen in den nachfolgenden Schritten wird jedoch immer ein gewisser Anteil (20 bis 80%) von Chips unbrauchbar, die später auszusondern sind. Bei der Produktion von integrierten Schaltkreisen ist ein solcher Verlust stets

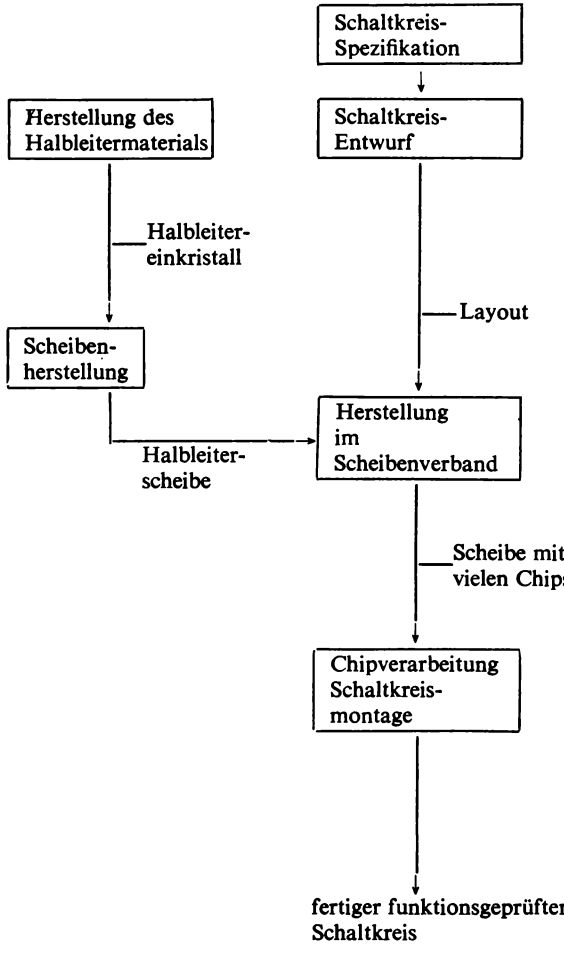


Bild 11. Prozeßgrundschritte bei der Herstellung von integrierten monolithischen Schaltkreisen

mit einzuplanen. Seine Größe ist auch wesentlich von der Losgröße, d.h. von der Anzahl der Segmente je Scheibe und damit von der geometrischen Größe der einzelnen Chips, abhängig. Bei relativ großen Chips ist infolge der größeren Wahrscheinlichkeit eines Fehlers die Ausbeute geringer.

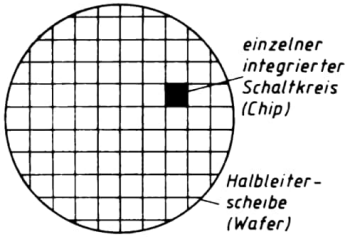


Bild 12. Integrierte Schaltkreise im Scheibenverband

Aus schaltungstechnischen und technologischen Gründen wird ein Chip in bestimmte Bereiche eingeteilt (Bild 13).

Zentral liegen die hochintegrierten Schaltungsteile, am Rand befinden sich die niedriger integrierten und die Kontaktierungsbereiche.

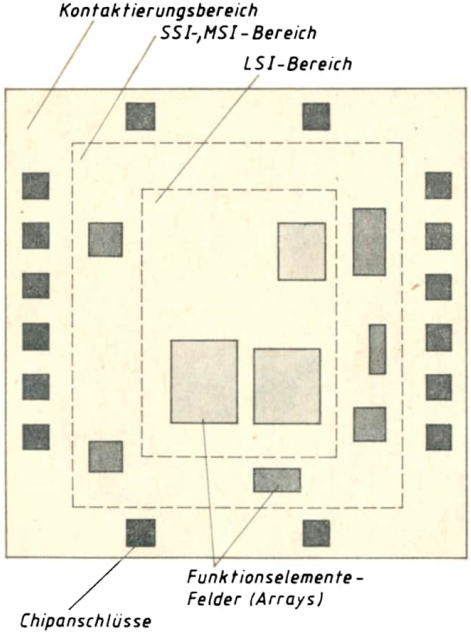


Bild 13. Chipaufteilung eines LSI-Schaltkreises

– Schaltungsentwurf:

Bevor die Scheibenprozesse in Angriff genommen werden können, müssen in einem Entwurfsprozeß die elektronische Schaltung, die Eigenschaften der einzelnen Funktionselemente sowie die daraus folgenden «topologischen» Grob- und Feinstrukturen des integrierten Schaltkreises bestimmt werden. Je nach seiner Aufgabenstellung und der zu realisierenden Logikfunktion werden im Schaltungsentwurf die elektronischen Funktionselemente, wie MOS- oder bipolare Transistoren und Verbindungsleitungen, miteinander verknüpft. Sie bilden ein «Schaltbild» des elektronischen Netzwerkes. Mit Hilfe von Simulationen, also der Nachbildung der Schaltung auf einem größeren Rechner, wird die Schaltung überprüft, verbessert und deren richtige Arbeitsweise nachgewiesen. Im folgenden Schritt wandelt man die Schaltung in geometrische Figuren für die technologische Feinstrukturierung des Halbleiterchips um, was dem sogenannten Layoutentwurf entspricht (Bild 14). Das *Layout* eines integrierten Schaltkreises kann man sich etwa als die «Landkarte» der Oberfläche sowie der darunter «vergrabenen» Halbleiterstrukturen des Chips vorstellen. Es ist die geometrische Darstellung der elektronischen Schaltung. Das Ergebnis des Schaltungsentwurfs eines integrierten Schaltkreises sind Schablonen oder Datenträger, auf denen die gesamte Geometrie und Struktur des Schaltkreises digital gespeichert sind.

– Bearbeitung der Halbleiterscheiben:

In den folgenden Prozeßschritten wird mit Hilfe einer ausgewählten Basistechnologie die Halbleiterscheibe strukturiert. Unter einer Basis-

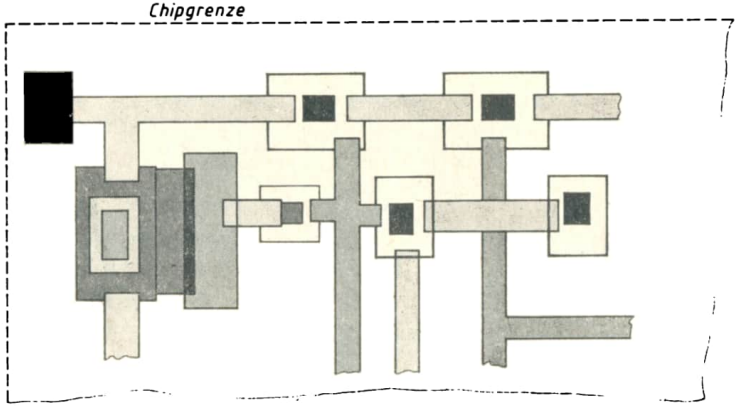


Bild 14. Ausschnitt aus dem Layout eines integrierten Schaltkreises

technologie versteht man hierbei eine bestimmte Abfolge in der technologischen Bearbeitung, Behandlung und Veränderung des Halbleitermaterials. Im Verlauf dieser Arbeitsschritte entstehen dotierte Gebiete, Isolier- und Halbleiterschichten, Metallschichten sowie Verbindungsstrukturen. Die Art und Reihenfolge der Prozeßschritte und die dabei einzustellenden Parameter unterscheiden die Basistechnologien. Bei der Strukturierung der Halbleiterscheibe sind die Prozesse Mustervorgabe, Strukturübertragung und Strukturerzeugung von Bedeutung.

Mustervorgabe
Betrachtet man den Aufbau eines integrierten Schaltkreises im Querschnitt (z. B. Bild 10), so erkennt man, daß die Halbleiterscheibe nicht nur in horizontaler (lateral), sondern auch in vertikaler Richtung strukturiert ist. Um eine solche Struktur auf das Halbleitermaterial zu übertragen und letztlich zu erzeugen, sind mehrere «horizontale» Strukturvorgaben oder Muster notwendig. Sie liegen in Form von Schablonen oder Datensätzen aus den bereits erläuterten Prozessen der Entwurfsphasen des Schaltkreises vor.

Mikrolithografie
In den Prozessen der Strukturübertragung spielen sogenannte lithografische Verfahren eine große Rolle. Bedeutsam ist zur Zeit die Lichtlithografie. Im Prinzip handelt es sich um einen lichtoptischen Schattenwurf der gewünschten Struktur auf eine mit lichtempfindlichem Lack bedeckte Halbleiterscheibe. Durch eine vorgefertigte Maske, die Fotoschablone, ist die Halbleiterscheibe mit Licht zu bestrahlen. Das Muster der zu übertragenden Struktur wird ebenfalls auf lithografischem Wege eingraviert. Danach lassen sich belichteter oder unbelichteter Lack chemisch entfernen. Die freigelegten Stellen können in verschiedenster Art und Weise, durch Ätzen, Dotieren usw., behandelt werden (Bild 15). Mit der Lichtlithografie sind Strukturen von 2 bis 3 μm (Kontaktlithografie) realisierbar, bei Musteraddition, d. h. einer stückweisen Belichtung der Halbleiterscheibe (step-and-repeat-Verfahren), und Verwendung von ultraviolettem Licht sogar bis etwa 1 μm (kontaktlose Projektionslithografie). Die natürliche Grenze der Lichtlithografie bildet die Wellenlänge des Lichtes, die im sichtbaren Bereich etwa 0,4 bis 0,8 μm beträgt und dann zu Beugungserscheinungen führt. Werden noch feinere Strukturauflösungen für höchstintegrierte Schaltkreise benötigt, findet die Elektronenstrahl- oder Röntgenstrahllithografie Anwendung. Hierbei lassen sich Strukturen bis 0,1 μm erzeugen.

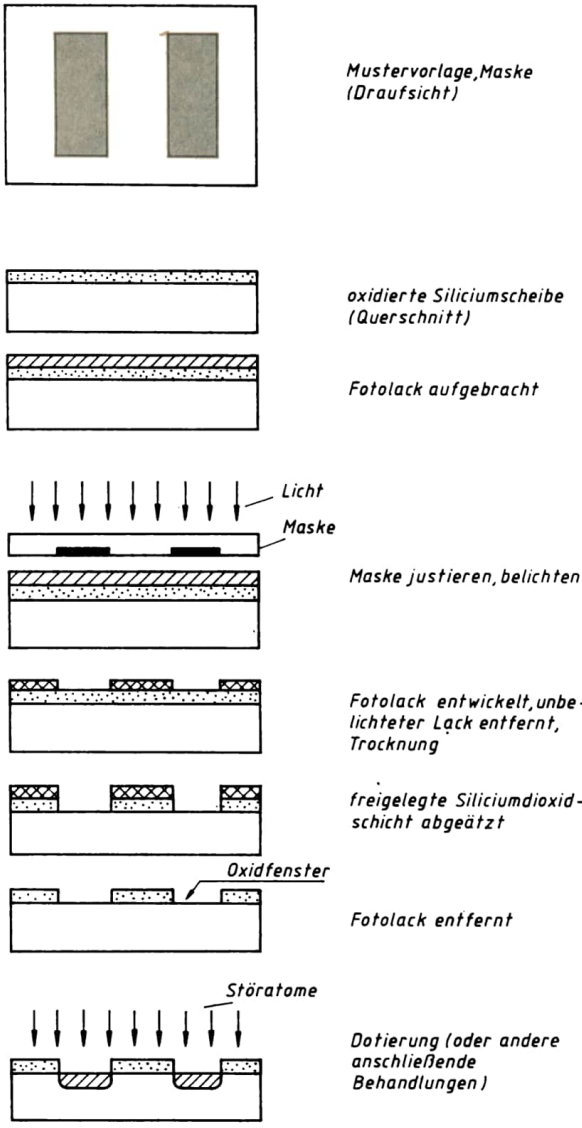


Bild 15. Stationen des fotolithografischen Prozesses

Die Schablonenherstellung und die Übertragung des Schablonenbildes auf die Halbleiterscheibe erfordern eine ganze Kette spezieller Geräte und hochpräziser Einrichtungen des optischen Gerätebaus, wie Bildgeneratoren, Projektionsanlagen, automatische und rechnergesteuerte Kreuztische sowie Justier- und Belichtungseinrichtungen.

Strukturerzeugung

Maßgebliche Prozesse der Strukturerzeugung auf bzw. in der Halbleiterscheibe sind u.a. Dotieren (s. 2.5.), Ätzen (Abtragen von Material), Beschichten (Auftragen von lichtempfindlichen Lacken oder Metall), Epitaxie (Aufwachsen von Halbleitermaterial), Oxydieren (Umwandeln von Si in SiO_2) und Maskierung (Abdeckung von Oberflächenteilen) [1, 3]. Die Strukturerzeugung auf der Halbleiterscheibe hat zum Ziel, die integrierten elektronischen Funktionselemente in Form von dünnen, vertikal und horizontal abgegrenzten Bereichen im Halbleiterkristall zu bilden. Ein Beispiel für eine einfache Struktur ist im Bild 15 gezeigt; in der Praxis liegen allerdings noch viel kompliziertere Strukturen vor.

– Chipverarbeitung, Montageprozeß:

Nachdem die Siliciumscheiben vollständig bearbeitet sind, werden die einzelnen integrierten Schaltkreise voneinander getrennt, indem die Scheibe geritzt und danach gebrochen wird. Die entstehenden Chips haben je nach Komplexitäts- und Integrationsgrad eine Kantenlänge von einem bis zu mehreren Millimetern. Die winzigen Halbleiterplättchen werden einem Montageprozeß zugeleitet (Bild 16), beim «Verkappen» in ein Gehäuse eingehüllt und zur «Kontaktierung» mit Verbindungsdrähten zwischen Chipanschlüssen und Gehäuseanschlüssen versehen. Nach der Montage erfolgt die Schlußprüfung der fertigen Schaltkreise mit entsprechenden Prüfautomaten.

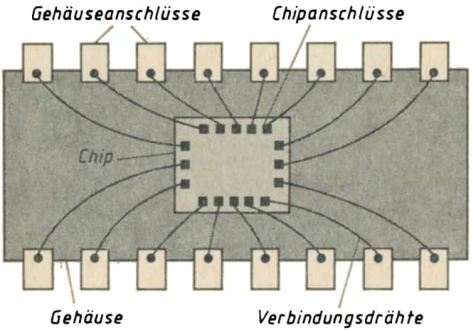


Bild 16. Fertig montierter integrierter Schaltkreis (Innenansicht)

Bei all diesen Arbeiten der Entwicklung und Herstellung hochintegrierter Schaltkreise, beginnend mit dem Logikentwurf über den Layoutentwurf, die Layoutverarbeitung, die Scheibenbearbeitung bis zur Endprüfung, ist der Einsatz von Rechenanlagen unumgänglich. Einerseits sorgen sogenannte CAD-Systeme (computer aided design – Rechnerunterstützter Entwurf) auf der Basis von umfangreichen Datenbanken und speziellen Kommunikationssystemen während des Entwurfs für eine effektive Bearbeitung und Speicherung der beträchtlichen Datenmengen.

Andererseits finden heute neben den CAD-Systemen zunehmend auch sogenannte CAM-Systeme (computer aided manufacturing – Rechnerunterstützte Herstellung) oder CAD/CAM-Systeme Eingang. Dies bedeutet, daß eine durchgehende rechnergestützte oder rechnergeführte Entwicklung und Fertigung der integrierten Schaltkreise vorliegt.

Für alle Bearbeitungsschritte zur Herstellung der Schaltkreise ist außerdem der Einsatz komplizierter und aufwendiger Einrichtungen sowie Fertigungsautomaten erforderlich. Diese Investitionen sind einer der Gründe für den hohen ökonomischen Aufwand einer Schaltkreisentwicklung und der entsprechenden Vorbereitung der Produktion. Läuft jedoch die Fertigung eines Schaltkreistyps zufriedenstellend bei ausreichender Chipausbeute, dann ist der Aufwand für die Herstellung eines einzelnen integrierten Schaltkreises äußerst klein. Aus diesem Grund wird die Produktion um so ökonomischer, je höher die Gesamtstückzahl eines herzustellenden Typs ist. Da bei Mikrorechnern vereinheitlichte hochintegrierte Schaltkreise eingesetzt werden und die Flexibilität der Schaltung bzw. der Geräte erst durch nachgelagerte Programmierprozesse erreicht wird, ist die geforderte Bedingung nach hohen Produktionsstückzahlen eines Typs gegeben.

3. Grundlagen der Mikrorechentechnik

3.1. Information – Signal – System

Wir hatten bereits festgestellt, daß in der Mikrorechentechnik, wie in allen informationstechnischen Anlagen, das Gewinnen, Darstellen, Übertragen und Verarbeiten von Informationen eine zentrale Rolle spielen. Typisch für all diese Prozesse ist, daß es sich dabei immer um ein charakteristisches Zusammenspiel von verschiedenartigsten Signalen und Systemen handelt. Entscheidend ist hierbei die «Steuerung», die eine «zielgerichtete Beeinflussung» eines Systems kennzeichnet. Dazu sollen zunächst einige begriffliche Erläuterungen gegeben werden.

In der Technik sind drei verschiedene Betrachtungsweisen der benutzten Objekte (Maschinen, Geräte, Schaltungen usw.) möglich, die sich meist sinnvoll ergänzen: Es handelt sich um den physikalischen, den technologischen und den systemtechnischen Aspekt. Zu den ersten beiden Gesichtspunkten wurden im Hinblick auf die Mikrorechentechnik im vorhergehenden Abschnitt einige Ausführungen gemacht. Jetzt wollen wir den systemtechnischen Aspekt etwas näher beleuchten. Mit Hilfe von Abstraktionsvorgängen und Modellbildungen liefert die systemtechnische Betrachtungsweise allgemeingültige Aussagen für eine Reihe von Vorgängen, die physikalisch völlig verschieden sind. Damit ist es möglich, für unterschiedliche Sachverhalte eine einheitliche theoretische und mathematische Beschreibung zu verwenden. Bezogen auf die Elektronik – oder, in unserem Fall, auf die Mikrorechentechnik – ist es dann beispielsweise gleichgültig, ob eine bestimmte elektronische Schaltung aus Röhren, Transistoren, hochintegrierten Schaltkreisen oder gar aus noch nicht existierenden elektronischen Bauelementen zu realisieren ist. Bestimmte funktionelle Eigenschaften, entsprechende allgemeingültige Aussagen und, davon abgeleitet, bestimmte technische Prinzipien, sind immer die gleichen. Zentrale Begriffe in dieser Betrachtungsweise sind *Information*, *Signal* und *System* [7, 8, 9].

Information

Der Begriff Information hat im allgemeinen Sprachgebrauch die Bedeutung von Nachricht, Mitteilung, Neuigkeit oder Auskunft über bestimmte Prozesse in Natur, Technik oder Gesellschaft. Der technische Aspekt einer Information stellt die Erfassung, Speicherung, Übertragung und Verarbeitung mit technischen Hilfsmitteln in den Mittelpunkt. Bei diesen Prozessen wird ein materieller «Träger» bzw. «Behälter» benötigt, der die Informationen enthält. Dieser muß unterscheidbare Merkmale aufweisen, z.B. veränderbare physikalische Größen wie Helligkeit, Temperatur, Druck, elektrische Spannung oder elektrische Stromstärke und Struktur- bzw. Musterbildung (beispielsweise Schrift) ermöglichen. Wird ein solcher «Träger» oder «Behälter» für die Übertragung von Informationen benutzt, bezeichnet man ihn als *Signal*, wird er dagegen für die Aufbewahrung von Informationen verwendet, so wird er *Speicher* genannt. Während der Übertragung kann ein Träger in vielfältiger Weise, gewollt oder ungewollt Wandlungen unterliegen. Über die verschiedenen Stationen – auch bei Änderung der Trägerart – bleibt jedoch ein Anteil, nämlich die Information, unverändert erhalten. Etwas anders liegen die Verhältnisse, wenn Informationen verarbeitet werden. Hier finden in bestimmten Systemen Prozesse statt, die für eine Wandlung von Informationen sorgen oder die Informationen in Wirkungen umsetzen.

Es hat sich herausgestellt, daß der Begriff Information so weitreichend und umfassend ist, daß bisher eine exakte und allseitige Bestimmung nicht möglich ist. Bis jetzt existieren nur eingeschränkte Festlegungen auf speziellen Gebieten [8].

Auch zwischen Mikrorechner und Umwelt sowie im Mikrorechner selbst werden verschiedenartigste Informationen ausgetauscht. Wir wollen sie für die Belange der Mikrorechentechnik als eine zielgerichtete, für den Empfänger verständliche Mitteilung auffassen. Als «Empfänger» wird sowohl die Maschine, der Automat, das Gerät, das elektronische System usw. als auch der Mensch bezeichnet. Als «Mitteilung» sollen Werte von allen möglichen Größen gelten. Derartige Informationen lassen sich mit elektrischen Signalen übertragen sowie mit elektronischen Systemen speichern und verarbeiten. Sie sind außerdem geeignet, bestimmte technische Systeme zu steuern.

Signale

Wie wir bereits festgestellt haben, dienen Signale zur Informationsübertragung. Grundsätzlich kann jede physikalische Größe als Träger von Informationen benutzt werden. Solche Signale enthalten einen oder auch mehrere *Informationsparameter*, die beeinflußbar sind und

Informationen «tragen». Kennzeichnend für ein Signal ist ein zeitlicher Verlauf der entsprechenden physikalischen Größe. In der Praxis gibt es sehr viele Arten. So liegen in der Mikrorechentechnik *elektrische Signale* vor, bei denen die physikalische Größe der elektrische Strom oder die elektrische Spannung der entsprechenden Informationsparameter ist. Wollen wir eine Einteilung nach dem Wertevorrat des Informationsparameters vornehmen, dann können unterschieden werden:

- analoge Signale, bei denen Strom oder Spannung beliebig viele Werte (im Grenzfall unendlich viele) innerhalb des Wertebereichs annehmen können;
- diskrete Signale, wo der Strom oder die Spannung nur bestimmte, diskrete Werte aus einer abzählbaren Wertemenge erhalten.

Die diskreten Signale lassen sich weiterhin unterteilen nach der maximalen Anzahl der möglichen Werte des Informationsparameters in binäre (zweiwertige) und mehrwertige sowie digitale Signale, in denen Informationen in sogenannten Worten dargestellt werden. Anhand einiger typischer Beispiele aus der Elektronik wird in Bild 17 gezeigt, wie die Signale aufgebaut sind. In der Mikrorechentechnik haben die binären und digitalen Signale die größte Bedeutung. Auf diese Signalarten wird zu einem späteren Zeitpunkt näher eingegangen.

Systeme

Unter einem System versteht man im allgemeinen Sprachgebrauch eine Gesamtheit von verbundenen (wechselwirkenden) Einzelbestandteilen. Das können beispielsweise eine technische Anlage oder Einrichtung, in der Mikrorechentechnik vornehmlich elektronische Schaltungen, Baugruppen oder Geräte sein. Man interpretiert hierbei als System eine abgegrenzte Menge beliebiger Elemente, die als ein zusammenhängendes Ganzes, als eine «Welt» für sich mit eigener «Ordnung» und eigenen «Gesetzmäßigkeiten» betrachtet werden. In den bisherigen Ausführungen wurde der Begriff *System* in der genannten Art und Weise auch verwendet. In diesem Sinne ist es also eine sehr weit gefaßte Bezeichnung für alle Arten von Objekten aus der uns umgebenden Realität. Man unterscheidet u.a. physikalische, biologische, ökonomische und technische Systeme. Wir beschränken uns im weiteren stets auf technische Systeme.

Der Begriff System wird jedoch noch in einer weiteren Bedeutung gebraucht, nämlich dann, wenn unter einem System ein *Modell* verstanden wird, das in sehr allgemeiner Weise zur Beschreibung und Untersuchung gewisser technischer Gebilde, elektronischer Schaltun-

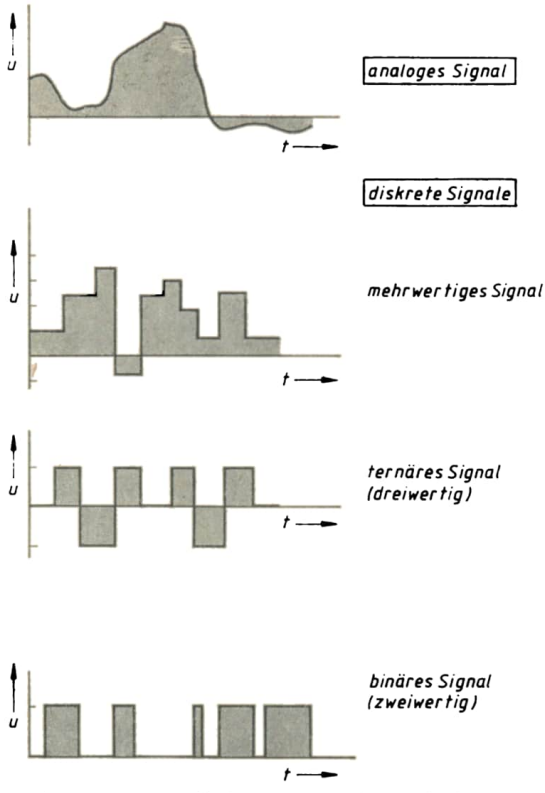


Bild 17. Signalarten in der Elektronik (U elektrische Spannung, t Zeit)

gen, Netzwerke oder ganzer Rechnersysteme dient [10]. In diesem Fall werden oft sogenannte Blockbilder zur grafischen Darstellung der entsprechenden Systeme benutzt. Ein System ist von der Umwelt abgegrenzt und kann von ihr «Eingangsgrößen» aufnehmen bzw. an sie «Ausgangsgrößen» abgeben (Bild 18). Um jedoch Informationen weiterzuleiten, muß ein Informationsträger vorliegen. Diese Aufgabe übernehmen die bereits erläuterten Signale. Laufen im Inneren von Systemen Vorgänge oder Prozesse ab, so gelangt eine Nachricht darüber durch die Signale an die Umwelt. Umgekehrt kann eine Beeinflussung des Systems, also eine Steuerung, über Signale von außen erfolgen.

Die Elemente oder *Glieder* eines Systems sind Teilsysteme, bei denen die Eingangsgrößen in bestimmter Weise die Ausgangsgrößen

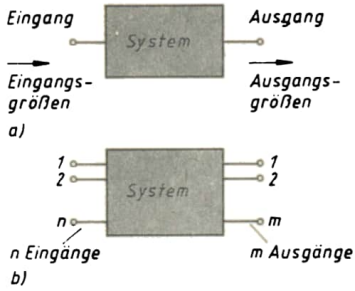


Bild 18. Blockdarstellung eines Systems mit einem Ein- und Ausgang (a) sowie mehreren Ein- und Ausgängen (b)

beeinflussen. Sie sind gewissermaßen die Elementarbausteine, die in der betreffenden Betrachtungsweise nicht weiter unterteilt werden sollen. Wir bemerken jedoch, daß bei einem System mit Modellcharakter immer das Verhältnis des Standpunktes gegenüber dem System zu beachten ist. Eine gleiche Menge von Elementen kann in einem Fall, bei Lösung bestimmter Aufgaben, als Teil eines größeren Systems betrachtet werden. Ebenso ist es möglich, ein Glied eines Systems bei einer anderen Untersuchung als ein ganzes System aufzufassen.

Die *Systemstruktur* ist die Menge und Art der Kopplungen zwischen den Gliedern oder Elementen (Bild 19). Jedes System kann seiner Struktur und Funktionsweise entsprechend, verschiedene «innere» Zustände einnehmen oder durchlaufen. Die maximale Anzahl der möglichen Zustände wird in Abhängigkeit vom Abstraktionsgrad

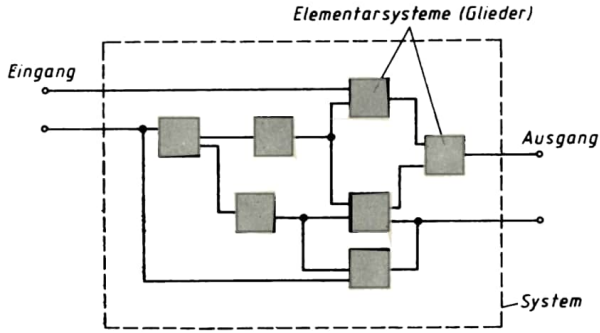


Bild 19. Beispiel einer Systemstruktur

und der Systemart sehr klein (z. B. zwei Zustände), relativ groß oder auch unendlich groß sein. Die Kennzeichnung des jeweiligen Zustands eines Systems erfolgt mit Hilfe von «Zustandsgrößen» [11].

Die *Systemart* wird wesentlich von den Eigenschaften der Ein- und Ausgangs- sowie Zustandsgrößen des Systems gekennzeichnet. Sind dies analoge Größen, so liegt ein analoges System vor, sind sie diskreter Natur, so spricht man von einem diskreten System. Eine analoge Eingangs-, Ausgangs- oder Zustandsgröße kann – wie bei Signalen – innerhalb eines (begrenzten oder unbegrenzten) Bereiches jeden beliebigen Wert annehmen, eine diskrete Größe demgegenüber nur Werte aus einer abzählbaren Menge von Werten. Bei elektronischen Systemen, wie Geräten, Schaltungen, integrierten Schaltkreisen, kann man sowohl analoge als auch diskrete Systeme unterscheiden (Bild 20). Letztere werden weiter unterteilt in binäre und digitale.

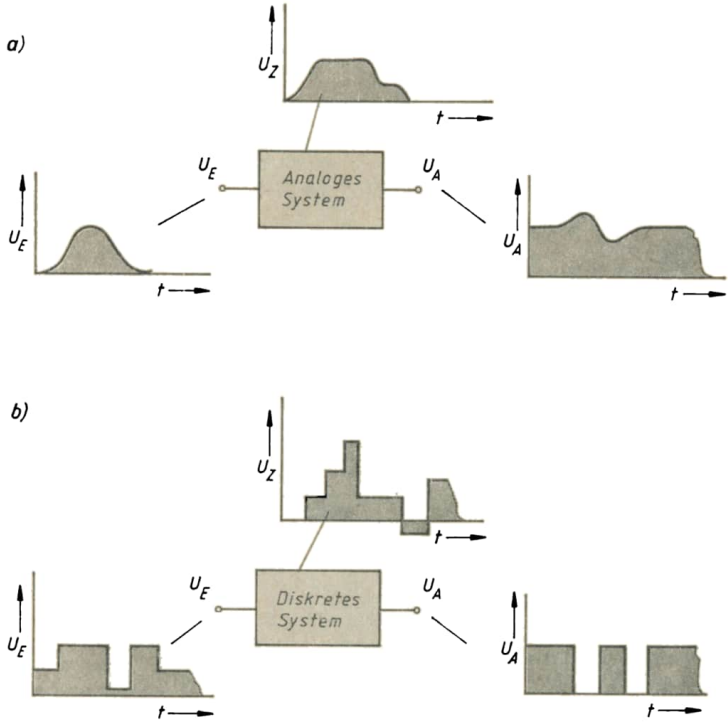


Bild 20. Vergleich zwischen analogen (a) und diskreten Systemen (b) (U_E Eingangsgröße, U_A Ausgangsgröße, U_Z Zustandsgröße)

3.2. Analogrechner – Digitalrechner

Auf der Grundlage der gegebenen Systemmerkmale hat man bei elektronischen Rechnern zwei wichtige Arten zu unterscheiden: Analogrechner und Digitalrechner. Sie unterscheiden sich wesentlich in Aufbau, Wirkungsweise und Verwendungszweck bzw. Aufgabenspektrum. Nach klassischer Auffassung arbeiten Analogrechner nach dem Prinzip des Messens und Vergleichens, während dem Digitalrechner das Prinzip des Zählens zugrunde liegt.

Der *Analogrechner* ist, wie sein Name sagt, ein analoges System und verarbeitet analoge Größen. Interne Signale sowie Ein- und Ausgabegrößen haben daher einen kontinuierlichen Werteverlauf. Die zu verarbeitenden Zahlen oder Funktionen werden im Rechner mittels physikalischer Größen, im allgemeinen über die elektrische Spannung, in Abhängigkeit von der Zeit dargestellt. Der Analogrechner verarbeitet die veränderlichen Größen und Funktionen mit Hilfe von bestimmten Rechenelementen, wie Additoren, Subtraktoren, Integratoren oder Multiplikatoren. Diese Grundelemente sind ebenfalls analoge Systeme. Die Programmierung erfolgt nach einem zu entwerfenden Verschaltungs- oder Koppelplan, der einer bildlichen Darstellung der Zusammenschaltung verschiedener Rechenelemente entspricht. Analogrechner haben vor allem für die Untersuchung, Berechnung sowie Darstellung von Bewegungsgleichungen (Differentialgleichungen) oder allgemeiner Prozeßgleichungen und -zusammenhänge aus Natur und Technik Bedeutung.

Praktisch gewichtiger sind jedoch die Digitalrechner. Sie verarbeiten digitale Größen, die intern mittels binärer Signale übertragen werden. Auch hier sind die Werte, Funktionen oder Daten durch die physikalische Größe elektrische Spannung (oder Strom), allerdings digital codiert, darzustellen. Der klassische Digitalrechner verarbeitet diese Größen in einer Recheneinheit, dem Rechenwerk. Alle im Rechner ablaufenden Vorgänge koordiniert das Steuerwerk, eine zentrale Steuereinheit. Die Programmierung des Digitalrechners erfolgt durch Vorgabe eines zu entwerfenden Programms, welches einer diskreten Folge der auszuführenden Bearbeitungsschritte des Rechners entspricht. Die Genauigkeit ist in der Regel um Größenordnungen höher als beim Analogrechner. Sie läßt sich praktisch bei erhöhtem Aufwand beliebig steigern, während beim Analogrechner relativ schnell physikalische Grenzen gesetzt sind. Sollen analoge Größen in einem Digitalrechner verarbeitet werden, ist eine Umformung dieser Eingabegrößen in digitale und andererseits der digitalen Ausgabegrößen in analoge notwendig. Auf diese Weise wird ein Digitalrechner, von

«außen» betrachtet, zum «Analogrechner». Dieses Prinzip wird in modernen Digital- und Mikrorechnern sehr häufig angewendet, falls die Notwendigkeit der Verarbeitung analoger Größen besteht.

3.3. Großrechner – Kleinstrechner

Eine Einteilung der in der Praxis vorkommenden Digitalrechner in Großrechner, Rechner mittlerer Größe und Kleinstrechner ist zweckmäßig. Kriterien für die Einordnung sind z. B. Volumen, Gewicht und Leistungsfähigkeit. So entspricht die Datenverarbeitungsanlage eines größeren Rechenzentrums in einem Betrieb einem Großrechner, während der heute übliche Taschenrechner ein Kleinstrechner ist. Aber auch diese Einteilung läßt sich noch verfeinern. So kann man die Kleinstrechner weiter unterteilen, etwa in die Größen einer Schreibmaschine (Tischrechner), einer Briefmappe (Personalcomputer) oder einer Zigarettenschachtel (Taschenrechner).

Die Klassifizierung macht es dem Anwender möglich, die mit dem Rechner zu lösenden Aufgabenklassen sowie die Nutzungsmöglichkeiten abzugrenzen. Aber auch für den Techniker und Hersteller ist eine solche Einteilung in Leistungsklassen von Interesse, da je nach Klasse der Rechner auch bestimmte technische und ökonomische Parameter erreicht werden müssen und bestimmte Einsatzbedingungen vorliegen. Beispielsweise wird eine kommerzielle Großrechenanlage eines Rechenzentrums in klimatisierten und staubgeschützten Räumen untergebracht, während ein Tischrechner unter normalen Raumbedingungen und ein in einer Werkzeugmaschine eingebauter kleiner Steuerrechner unter den relativ harten Bedingungen der Produktion funktionieren müssen. Dies soll verdeutlichen, daß der Digitalrechner heute in sehr verschiedenen Größen, Leistungen und auf unterschiedlichen Gebieten zum Einsatz kommt.

3.4. Mikrorechner

Während bei den oben gegebenen Einteilungen der Rechner entweder das grundsätzliche Funktionsprinzip (analog, digital) oder die Leistungsmerkmale und äußeren Kennzeichen (Kleinstrechner bis Großrechner) im Mittelpunkt stehen, wird bei der Charakterisierung eines Mikrorechners oder Mikrocomputers von inneren Realisierungs- und Funktionsprinzipien eines Digitalrechners ausgegangen [12, 13, 14,

15]. Ein Mikrorechner stellt im weitesten Sinne ein digitales Informationsverarbeitungs- und Steuerungssystem dar. Seine Merkmale sind:

- Aufbau aus einem oder wenigen hochintegrierten Schaltkreisen;
- Verwirklichung des Prinzips des programmgesteuerten Digitalrechners;
- relativ kleine geometrische Abmessungen, kleine Masse und geringer Energieverbrauch sowie hohe Zuverlässigkeit;
- Möglichkeit der Verwendung eines Mikrorechners als elektronisches Bauelement.

Die Herstellung eines solchen Digitalrechners mit konventionellen Mitteln der Elektronik war – wie bereits erläutert – nicht möglich; erst die Mikroelektronik schaffte die Voraussetzungen.

Aus der Sicht der Rechentechnik kann man demnach, unter Berücksichtigung der Realisierungsmöglichkeiten für Digitalrechner, zwei Arten unterscheiden:

- konventionelle Rechner (Makrorechner);
- hochintegrierte Rechner (Mikrorechner).

Anstelle von «Makrorechner» ist es besser, einfach «Rechner» zu setzen. Hieraus leiten sich auch die Begriffe «Rechentechnik» und «Mikrorechentechnik» ab. Der Mikrorechner unterscheidet sich von einem konventionellen Digitalrechner dadurch, daß er eine «integrierte Form» des Digitalrechners darstellt und dadurch die einzelnen Funktionsgruppen des Rechners nicht mehr zugänglich sind. Dies ist der gleiche Sachverhalt, wie er auch bei einer üblichen integrierten Schaltung auf die einzelnen, nicht mehr zugänglichen elektronischen Bauelemente zutrifft.

Aus der Sicht der Elektronik entspricht der Mikrorechner einer der Realisierungsmöglichkeiten mikroelektronischer Schaltungen oder Systeme. Daher umfaßt der Gegenstand der Mikrorechentechnik eine Teilmenge elektronischer Schaltungen und Systeme, nämlich diejenigen, die nach bestimmten Vorschriften programmierbar sind. Hierbei ist «programmierbar» im Sinne der Programmierung eines Digitalrechners zu verstehen. Mit anderen Worten: Die Mikrorechentechnik ist in der Betrachtungsweise der Elektronik der Teil, in dem in den mikroelektronischen Einheiten als Funktionsgrundlage das Prinzip des programmgesteuerten Rechenautomaten sowie die Mittel und Methoden der automatisierten Verarbeitung von codierten Informationen verwendet werden. Wenn wir in dieser Weise den Mikrorechner als eine spezielle Schaltung oder als ein spezielles elektronisches Bauelement auffassen, dann gelten natürlich auch alle für elektronische Schaltungen sowie Systeme üblichen Regeln und Methoden

bezüglich des Entwurfs und des Aufbaus für Systeme mit Mikrorechnern.

Die Programmierung der Mikrorechner erfolgt im Prinzip nach gleichen oder ähnlichen Regeln, Methoden und Verfahren wie bei konventionellen Digitalrechnern. Einige beachtenswerte Besonderheiten resultieren aus der sehr großen Breite in der Anwendung sowie aus den technologischen Besonderheiten der Halbleiterschaltkreise. Die Fragen der Programmierung von Mikrorechnern werden in einem besonderen Abschnitt eingehender behandelt.

Aufbau eines Mikrorechners

Der prinzipielle Aufbau eines Mikrorechners ist im Bild 21 dargestellt. Seine wichtigsten Funktionseinheiten sind:

- Mikroprozessor
- Halbleiter-Hauptspeicher
- Ein- und Ausgabesystem
- Bussystem.

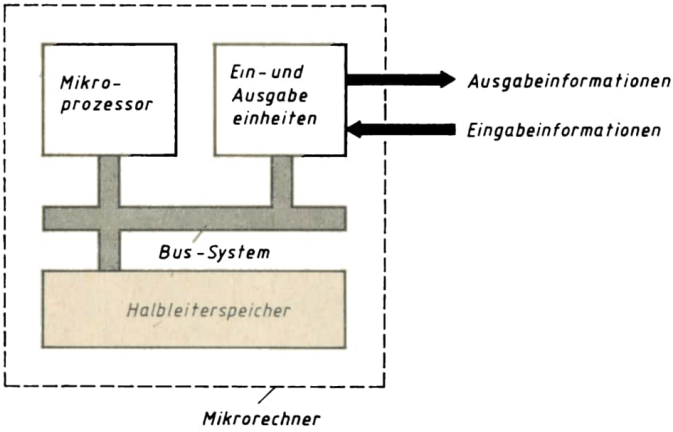


Bild 21. Grundstruktur eines Mikrorechners

Der Mikroprozessor, die Halbleiterspeicher und meist auch die Ein- und Ausgabeeinheiten liegen als hochintegrierte Schaltkreise vor und sind schaltungstechnisch durch ein sogenanntes Bussystem (ein System von speziell betriebenen Verbindungsleitungen) miteinander verbunden. In besonders hochintegrierter Form können auch alle Komponenten auf einem einzigen Mikrochip untergebracht sein und bilden dann einen sogenannten Einchip-Mikrorechner.

Ein funktionstüchtiger Mikrorechner enthält in seinem Hauptspeicher ein Programm, die Instruktions- oder Befehlsfolge. Es entspricht einer bestimmten Folge von Anweisungen, die der Mikrorechner im Verlauf seines Betriebs abarbeitet. Dabei wird ihm mitgeteilt, welche Funktionen er unter ganz bestimmten Bedingungen auszuführen hat. Der Mikrorechner kann auf diese Weise Vorgänge, Geräte oder ganze Einrichtungen steuern, Berechnungen vornehmen oder numerische und nichtnumerische Daten (z.B. Texte) bearbeiten, wobei in ihm komplexe Informationsprozesse ablaufen. Die hierfür notwendige Verbindung zur «Außenwelt» wird mit Hilfe der Ein- und Ausgabeeinheiten erreicht. Es findet dabei ein wechselseitiger Austausch von verschiedensten Informationen zwischen Mikrorechner und gesteuertem Objekt statt (Bild 22).

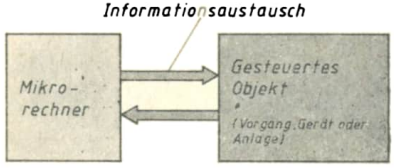


Bild 22. Mikrorechner als steuerndes System

Das Herzstück eines Mikrorechners ist der *Mikroprozessor*. Er koordiniert und steuert alle Vorgänge im Mikrorechner und kontrolliert alle ein- und auslaufenden Informationen. Als wichtigste Funktionskomponenten sind die «Steuerzentrale» und das «Verarbeitungszentrum» in ihm enthalten. In der Terminologie der klassischen Rechentechnik entspricht der Mikroprozessor der zentralen Verarbeitungseinheit eines Digitalrechners oder CPU (central-processing-unit). Er besteht aus einem, gegebenenfalls auch aus mehreren, hochintegrierten Schaltkreisen oder – im Fall des Einchip-Rechners – nur aus einem Teil des Mikrochips. Der Mikroprozessor kann eine Vielzahl von verschiedenen Funktionen ausführen, wenn ihm die Befehlsfolge zugeführt wird. In ähnlicher Weise wie der gesamte Mikrorechner stellt er ein komplexes digitales Informationsverarbeitungssystem dar und ist einer der wichtigsten Bausteine eines Mikrorechners. Er ist gewissermaßen sein «Steuer- und Rechenelement». In der Praxis gibt es eine sehr große Zahl verschiedener Mikroprozessor-Typen, die sich erheblich in Aufbau, Wirkungsweise und Leistungsfähigkeit unterscheiden.

3.5. Autonomer Mikrorechner und Mikroprozeßrechner

Wie in den vorhergehenden Abschnitten erläutert, wird eine Verbindung des Mikrorechners zur Umwelt mit Hilfe der Ein- und Ausgabeeinheiten geschaffen. Über diese Einrichtungen findet, unter «Leitung» des Mikroprozessors, ein wechselseitiger Informationsaustausch zwischen Mikrorechner und den zu steuernden Objekten statt. Die außerordentlich große Bedeutung der Mikrorechentechnik liegt darin begründet, daß faktisch jede technische Einrichtung oder jeder steuerbare Vorgang als zu steuerndes Objekt in Frage kommen kann. Hierbei hat man zwei Fälle zu unterscheiden: Zum einen können die gesteuerten Einrichtungen periphere Geräte zur Ein- und Ausgabe von Daten eines Nutzers sein. Dieser Betriebsfall ist der eines «normalen» elektronischen Rechners. Etwas anderes ist es, wenn der Mikrorechner für die Steuerung eines Systems sorgt, in dem Prozesse stattfinden, die in ununterbrochener Folge Informationen erzeugen bzw. für deren Ablauf ununterbrochen Informationen benötigt werden. Hierbei muß der Mikrorechner das System oder den Vorgang ständig «beobachten», rechnend verfolgen und entsprechende Reaktionen einleiten. Beispiele mit praktischer Bedeutung sind die Steuerung einer Werkzeugmaschine, eines Industrieroboters, eines chemischen Prozesses oder eines Flugzeugs. Dieser Betriebsfall wird auch als *Prozeßsteuerung* und der dabei eingesetzte Rechner als *Prozeßrechner* bzw. *Mikroprozeßrechner* bezeichnet.

Um den Unterschied der beiden Betriebsweisen zu verdeutlichen, seien zwei Beispiele angegeben. Im ersten (Bild 23) handelt es sich um eine autonome Arbeit eines Mikrorechners. Über ein Eingabe-

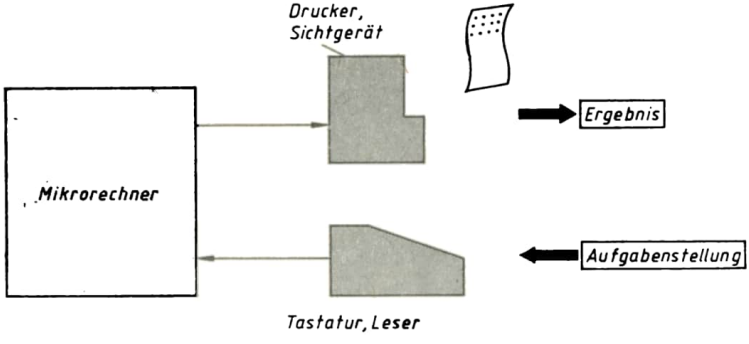


Bild 23. Autonome Betriebsweise des Mikrorechners

gerät wird eine fest umrissene Aufgabenstellung der Datenverarbeitung, beispielsweise die Lösung einer Gleichung oder die Berechnung des Lohnes eines Arbeiters, mitgeteilt. Nach dem Start der Berechnung werden ein oder mehrere Programme durchlaufen – der Rechner arbeitet eine bestimmte Zeit – um nach Bereitstellung der Ergebnisse, gegebenenfalls über einen Drucker auf Papier, die Arbeit an dieser Aufgabe wieder einzustellen. Die Aufgabe ist gelöst oder – was auch vorkommen kann – es trat ein Fehler auf, und die Aufgabe wurde abgebrochen.

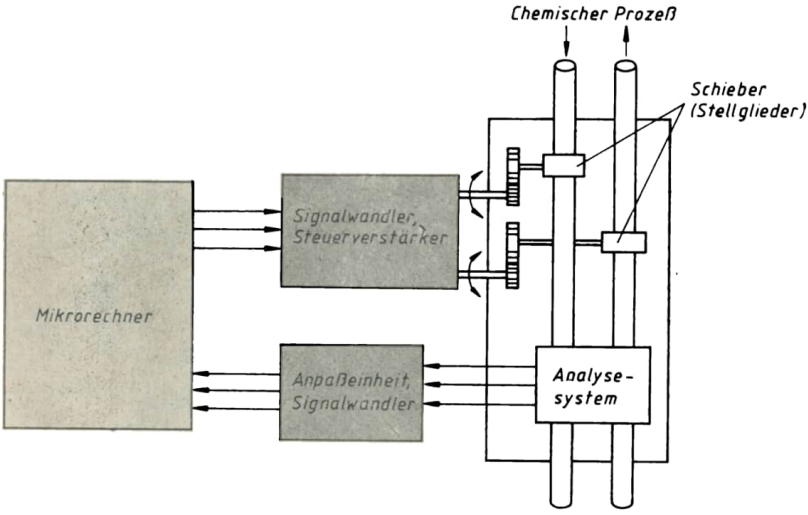


Bild 24. Mikrorechner als Prozeßrechner

Im zweiten Beispiel (Bild 24) steuert ein Mikrorechner das Rohrleitungssystem einer chemischen Anlage. Er erhält von einem Meßsystem laufend Informationen über das strömende Medium, die Temperatur, den Druck, die Strömungsgeschwindigkeit. Mit Hilfe der in ihm vorhandenen Programme ermittelt er auf der Grundlage der erhaltenen Informationen die jeweilige, von der Zeit abhängige Schieberstellung der Rohre, um über die verschiedenen Steuerverstärker eine entsprechende Veränderung des Flusses in den Rohren zu bewirken. Beide Rohre sind natürlich in einer hier nicht dargestellten Art und Weise über den chemischen Prozeß miteinander verbunden. In dieser Betriebsweise besteht im Mikrorechner eine ununterbro-

chene, vom äußeren Prozeß laufend beeinflußte Informationsverarbeitung; der Rechner läuft im sogenannten *Echtzeitbetrieb*. Dies bedeutet, daß er zeitlich mit seinen im Innern ablaufenden Vorgängen im Verhältnis zu den äußeren zu steuernden Prozessen Schritt hält. Er ist dann in der Regel Bestandteil des Steuer- oder Regelsystems. Bei seinem etwaigen Ausfall tritt in den meisten Fällen eine Störung im Prozeßablauf ein. Daher müssen Mikroprozeßrechner eine möglichst hohe Zuverlässigkeit aufweisen.

3.6. Vergleich zwischen programmierbarem Element und Mikrorechner

Ein besonders interessanter Fall liegt vor, wenn das Prinzip des Digitalrechners, also eine programmgesteuerte Funktionsweise, für die innere Organisation oder zur Steuerung von bestimmten Funktionsgruppen im Rechner selbst benutzt wird. Auf diese Weise lassen sich hierarchisch aufgebaute Systeme aus «Elementar»-Digitalrechnern entwerfen. Die Anwendung des Prinzips der Programmsteuerung kann dabei in vielfältiger Weise erfolgen, wie Organisation der Befehlsabarbeitung (Befehlsprozessor), Speicherorganisation (Speicherprozessor), Koordinierung von Transportaufgaben im Rechner (BUS-Prozessor) oder die Organisation von Ein- und Ausgabeprozessen (Peripherieprozessoren).

Der Grundgedanke einer solchen Steuerung mit einem Digitalrechner stammt von *M. v. Wilkes*. Er führte bereits 1951 das Konzept der sogenannten *Mikroprogrammsteuerung*, die im Prinzip einem Digitalrechner entspricht, ein. Mit Hilfe eines «kleinen», intern eingebauten Rechners werden die einzelnen Befehle des «großen» Rechners dargestellt und jeder Befehl während des Betriebs schrittweise abgearbeitet. Das Programm des «kleinen» Rechners sorgt gewissermaßen für das Erzeugen der einzelnen Befehle des «großen» Rechners. Eine Veränderung seines Programms bringt in einfacher Weise auch eine gewünschte Veränderung des Funktionsinhalts der einzelnen Befehle des «großen» Rechners mit sich. In der Folgezeit wurde dann dieses Prinzip ebenso für die Ein- und Ausgabesteuerung bei Großrechnern eingesetzt, was sich in den selbständig arbeitenden «Kanälen» heutiger Rechner widerspiegelt. Das Konzept der Mikroprogrammsteuerung wird auch in der Mikrorechentechnik angewendet. Beispielsweise haben viele Mikroprozessoren auf ihrem Chip eine Mikroprogrammsteuerung.

Die herausragende Bedeutung des Prinzips von *Wilkes* besteht jedoch darin, daß es als Ausgangspunkt für die Entwicklung von neuartigen Rechnerarchitekturen diente und die Strukturierung von Rechnersystemen in bestimmte Hierarchieebenen bezüglich der Programmsteuerung einleitete. In der klassischen Mikroprogrammsteuerung werden nur zwei solcher Ebenen eingeführt: die Mikroprogramm- und die Maschinenprogrammebene (Bild 25). Heute wird in der Mikrorechentechnik dieses Prinzip auf mehreren Ebenen verallgemeinert angewendet und damit der aus der Systemtheorie erläuterte Sachverhalt der Relativität zwischen «System» und «Elementen» auch auf «programmgesteuerte Systeme» und «programmgesteuerte Elemente» eines Systems erweitert.

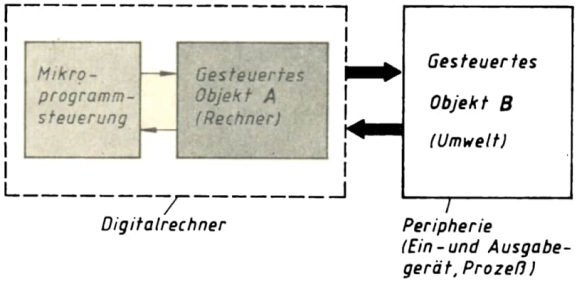


Bild 25. Prinzip der Mikroprogrammsteuerung

Um dies zu verdeutlichen, sei noch einmal auf den Systembegriff zurückgegriffen. Wir hatten festgestellt, daß sich ein System aus Elementargliedern zusammensetzt. Dabei ist es vom Standpunkt des Betrachters abhängig, ob es sich jeweils um ein Glied oder System handelt. Ein praktisches Beispiel dieser Betrachtungsweise ist die bereits angeführte Relativität zwischen «elektronischem Bauelement» und «elektronischer Schaltung». Die Verallgemeinerung des Konzepts der Mikroprogrammsteuerung ist nun in der Weise zu verstehen, daß ein aus programmgesteuerten Elementen entworfenes Gesamtsystem wieder als programmgesteuertes Element aufzufassen ist und als Grundlage für den Aufbau von noch komplexeren Systemen dienen kann. Dieser Vorgang der Abstraktion läßt sich theoretisch beliebig oft fortsetzen, und man erreicht damit Hierarchieebenen immer höheren Komplexitätsgrades. Das hierbei verfolgte Ziel besteht darin, auch sehr umfangreiche und komplexe programmgesteu-

erte Systeme aus einer relativ geringen Anzahl von «Elementarsystemen» aufzubauen und den Entwurf, die Architektur, die Funktion sowie die Programmierung des Gesamtsystems überschaubar zu erhalten. Auf jeder Hierarchieebene ist dann ein ähnliches Herangehen im Entwurfsprozeß oder bei Systemuntersuchungen möglich, wobei immer auf die in den darunterliegenden Ebenen durchgeführten Abstraktionsvorgänge aufgebaut wird.

4. Informationen und Prozesse im Mikrorechner

4.1. Binärdarstellung

Nachdem in den vorangegangenen Abschnitten die mikroelektronischen und systemtechnischen Grundlagen der Mikrorechentechnik im Mittelpunkt standen, soll nun auf die Funktionsweise des Mikrorechners näher eingegangen werden. Dabei ist es zunächst erforderlich, die im Mikrorechner auftretenden Informationsarten voneinander abzugrenzen sowie die Codierung und die physikalische Darstellung der Informationen zu betrachten. Diese Kenntnisse sind für das Verständnis der Funktionsweise eines Mikrorechners wichtig, denn sie bilden das theoretische Fundament.

Wie bereits ausgeführt, ist der Mikrorechner ein digitales System, in dem komplizierte programmgesteuerte Prozesse ablaufen. Nach außen liefert er auf Grund von ausgewählten Eingangsgrößen bestimmte Ergebnisgrößen. Demnach muß zunächst einmal zwischen Eingabeinformationen, Informationsverarbeitungsprozessen und Ausgabeinformationen unterschieden werden. Während der Informationsverarbeitungsprozesse in einem programmgesteuerten System lassen sich folgende elementaren Vorgänge unterscheiden: Die *Verknüpfung* von bestimmten Informationen, den vorliegenden Verarbeitungsvorschriften entsprechend, die *Informationsübertragung*, als Transport von Informationen nach bestimmten Transportvorschriften, und die *Informationsspeicherung*, als die Fixierung der Informationszustände nach Speichervorschriften. Darauf wird im einzelnen noch eingegangen. Daß die Informationen bei all diesen Prozessen in einer besonderen Form auftreten, nämlich binär codiert und digitalen Signalen aufgeprägt, ist besonders hervorzuheben. Eine solche Darstellung ist jedoch nicht mit beliebigen Informationen aus allen unseren Lebensbereichen möglich, sondern nur bei wertmäßig erfaßbaren Erscheinungen, Vorgängen oder Gegenständen mit Hilfe von vereinbarten Zeichen oder Zuständen physikalischer Größen. Im Zusammenhang mit rechentechnischen Vorgängen spricht man hierbei

auch von *Daten*. Sie werden durch Zeichenfolgen repräsentiert, die nach bestimmten Gesetzmäßigkeiten aus Elementarzeichen oder Symbolen zusammengesetzt sind. Nur solche eindeutig darstellbaren Informationen sind maschinell mit Hilfe technischer Einrichtungen zu verarbeiten. Davon rührt z.B. auch die, anstelle von Digitalrechnern, häufig benutzte Bezeichnung *Datenverarbeitungsanlage* her.

Die einfachste Form von Daten sind Zahlen, wie wir sie aus dem täglichen Leben gewohnt sind. Sie können Werte von verschiedensten Größen bezeichnen. Geläufig sind uns die Dezimalzahlen. In technischen Einrichtungen sind jedoch gerade die Dezimalzahlen sehr schwierig zu handhaben, da man zur Darstellung von einer *Ziffer* zehn Unterscheidungsmerkmale (z.B. zehn Spannungsstufen) heran-

Tabelle 3. Darstellung von Dual- und Dezimalzahlen

Dualzahl Z_b	Dezimalzahl Z_d
0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15
10000	16
10001	17
10010	18
10011	19
10100	20
⋮	⋮

$$Z_{\text{dual}} = b_n \cdot 2^n + b_{n-1}\, 2^{n-1} + \ldots + b_1 \cdot 2^1 + b_0 2^0$$
$$b_i = \{0, 1\}$$

$$Z_{\text{dezimal}} = d_n \cdot 10^n + d_{n-1}\, 10^{n-1} + \ldots + d_1\, 10^1 + d_0\, 10^0$$
$$d_i = \{0, 1, 2, \ldots, 9\}$$

ziehen muß. Am günstigsten erweisen sich hierbei die Dualzahlen (auch die Bezeichnung Binärzahlen ist üblich). Während eine Dezimalzahl aus den einzelnen Ziffern 0, 1, 2, ..., 9 zusammengesetzt ist, wird eine Dualzahl nur aus den Ziffern 0 und 1 (auch andere Symbole, z.B. 0 und L sind möglich) aufgebaut. Dem dezimalen Wert «12» entspricht beispielsweise die duale Darstellung «1100». Die Gegenüberstellung von dezimalen und dualen Zahlen sowie ihre Konstruktion (Zahlendarstellung) sind in Tabelle 3 wiedergegeben. Die Größe des darstellbaren Wertes einer Dezimal- bzw. Dualzahl ist theoretisch unbegrenzt. Je größer der Wert, um so mehr Stellen in der Zahlendarstellung sind allerdings erforderlich.

4.2. Wortbildung

In Digitalrechnern, einschließlich Mikrorechnern, ist jedoch aus technischen Gründen niemals eine beliebig große Zahl darstellbar, immer muß eine Begrenzung auf eine bestimmte maximale Stellenzahl vorgenommen werden. Eine solche Vorgehensweise wird in der Rechentechnik als Wortbildung, das Ergebnis als Wortdarstellung und die entsprechende Zahl als *Wort* bezeichnet. Der Wert des Wortes ist dessen *Belegung*. Hat man duale Zahlen zugrunde gelegt, so sind dies *Binärworte*. Sie sind das wichtigste Mittel zur Codierung von Informationen im Mikrorechner. Im Bild 26a ist ein Binärwort mit 5 Stellen und der Binärbelegung von «01101» aufgeführt. Um die Begrenzung auf die maximale Stellenzahl zu verdeutlichen, wurde eine symbolische Kästchendarstellung verwendet; sie wird auch in den weite-

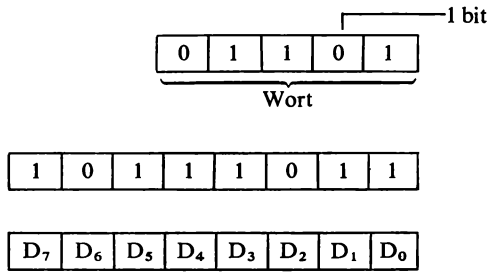


Bild 26. Wortdarstellungen
a) 5-bit-Wort mit konstanter Belegung
b) 8-bit-Wort mit konstanter Belegung
c) 8-bit-Wort mit variabler Belegung

ren Ausführungen stets gewählt. Jede Stelle des Binärwortes entspricht einem Kästchen und jedes ist mit dem *Binärsymbol* 0 oder 1 – man sagt hierzu auch kurz *Bit* – zu belegen (*Bit* – als Begriff; *bit* – als Einheit). Eine Binärstelle in einem Wort heißt deshalb auch oft Bitstelle. Im Beispiel handelt es sich also um ein 5-bit-Wort, im Bild 26b um ein 8-bit-Wort. Grundsätzlich hat man zwei Wortdarstellungen zu unterscheiden: Worte mit fester Belegung (Konstanten) und solche mit veränderlicher Belegung (Variablen). Wie in der Mathematik einer festen Zahl ihr Wert und einer noch nicht festgelegten oder unbekannten Zahl z. B. das allgemeine Symbol a oder x zugeordnet wird, so hat man auch bei Binärworten die Möglichkeit, ein Wort mit einer festen Binärbelegung zu versehen (Bild 26a, b) oder die allgemeine Darstellung zu wählen (Bild 26c). Im ersten Fall wird jede Bitstelle mit einem bestimmten Binärsymbol (0 oder 1) besetzt, im zweiten mit einem Buchstaben (im Bild D_i), der je nach Belegung dann später den Wert 0 oder 1 annehmen kann. Besonders interessant ist nun die Frage, wieviel verschiedene Binärbelegungen ein Wort in allgemeiner Darstellung zuläßt. Die Antwort hierauf gibt die wichtige Beziehung

$$N = 2^n$$

Der Buchstabe N symbolisiert hierbei die Anzahl der verschiedenen Binärbelegungen und n die Stellenzahl des Wortes. Ein einfaches Beispiel für $n = 3$ soll dies verdeutlichen. In diesem Fall sind entsprechend der Formel acht verschiedene 3-bit-Worte aufstellbar. Die dabei möglichen Bitbelegungen sind im Bild 27 gezeigt. Sie lassen sich durch binäres Zählen ermitteln. In Tabelle 4 ist für verschiedene Stellenzahlen in einer zusammenfassenden Aufstellung die jeweilige

Wortnummer	D_2	D_1	D_0	Dezimaläquivalent
1	0	0	0	0
2	0	0	1	1
3	0	1	0	2
4	0	1	1	3
5	1	0	0	4
6	1	0	1	5
7	1	1	0	6
8	1	1	1	7

Bild 27. Mögliche Bitbelegungen eines 3-bit-Wortes (Bildmitte)

maximale Anzahl der Binärworte angegeben. Wie aus der Formel ersichtlich, wächst mit der Stellenzahl n die Anzahl N der möglichen Binärworte exponentiell, also sehr schnell an.

Tabelle 4. Anzahl N der möglichen Bitbelegungen in Abhängigkeit von der Stellenzahl n des Wortes

Stellenzahl n	Anzahl N der Binärbelegungen
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024
12	4096
14	16384
16	65538
⋮	⋮
$N = 2^n$	

4.3. Wortformate

In der Mikrorechentechnik finden Wortdarstellungen mit sehr verschiedenen Stellenzahlen Verwendung. Dies zeigt sich in unterschiedlichen Wortformaten. Eine besondere Bedeutung hat das 8-bit-Format, das entsprechende Wort wird auch als *Byte* (sprich «bait») bezeichnet. Auf seiner Grundlage werden die Mehrbyteworte gebildet, deren Aufbau im Bild 28 veranschaulicht ist. Man hat dann außer dem 1-byte-Wort auch 2-byte-, 3-byte-Worte usw. vorliegen. Sie alle werden zur Informationsdarstellung im Mikrorechner benutzt.

4.4. Physikalische Darstellung der Binärworte

Für einen Techniker, der einen Mikrorechner entwickelt oder aufbaut, ist die Frage wichtig, wie die einzelnen «Bits» oder «Bytes» sowie die daraus abgeleiteten Binärworte im Mikrorechner zu verwirklichen

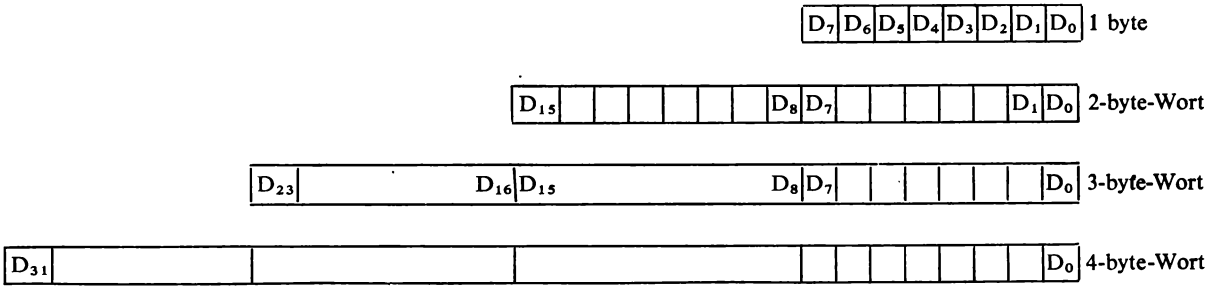


Bild 28. Byteorientierte Wortformate im Mikrorechner

sind. Zu diesem Zweck definiert man eine Zuordnung zwischen den binären Symbolen 0 und 1 sowie den im Mikrorechner intern benutzten Signalzuständen in der folgenden Weise (Tabelle 5): Einer binären «0» entspricht der Signalzustand «L» (low – niedriger Pegel), der binären «1» der Signalzustand «H» (high – hoher Pegel) der binären Signale. Im Bild 29 ist dieses Vorgehen am Beispiel der Zeitfunktion eines binären Signals gezeigt. Die allgemein bezeichneten Signalzustände L und H lassen sich im Mikrorechner «physisch» realisieren, indem ihnen zwei konkrete Werte einer physikalischen

Tabelle 5. Zuordnung zwischen logischem Wert und Signalpegel

Symbol (logischer Wert)	Signalpegel (Signalzustand)
1	H (high)
0	L (low)

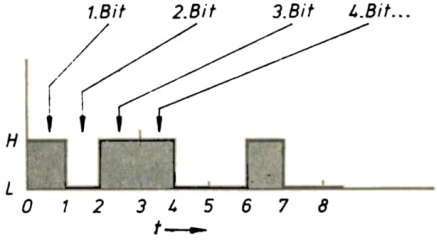


Bild 29. Binäres Signal (H (high) hoher Signalpegel, L (low) niedriger Signalpegel; *t* Zeit in Schritten der Bitteilung)

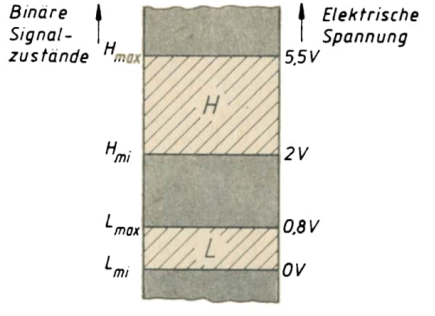


Bild 30. TTL-Logikpegelsystem (schraffiert zulässiger, unschraffiert verbotener Bereich für Binärsignal; H (high) Bereich für hohen Signalpegel, L (low) Bereich für niedrigen Signalpegel)

Größe, meist der elektrischen Spannung, zugeordnet werden. Größte Bedeutung hat hierbei die Zuordnung nach Bild 30, die auch als *TTL-Logikpegelsystem* bekannt ist. Diese Bezeichnung geht zurück auf ein Pegelsystem, das ursprünglich nur in einer bestimmten Schaltkreisfamilie, nämlich der Transistor-Transistor-Logik (TTL-System), verwendet wird. Heute wird dieses Pegelsystem in vielen integrierten Schaltkreisen realisiert und ist Standard in fast allen Mikrorechnern.

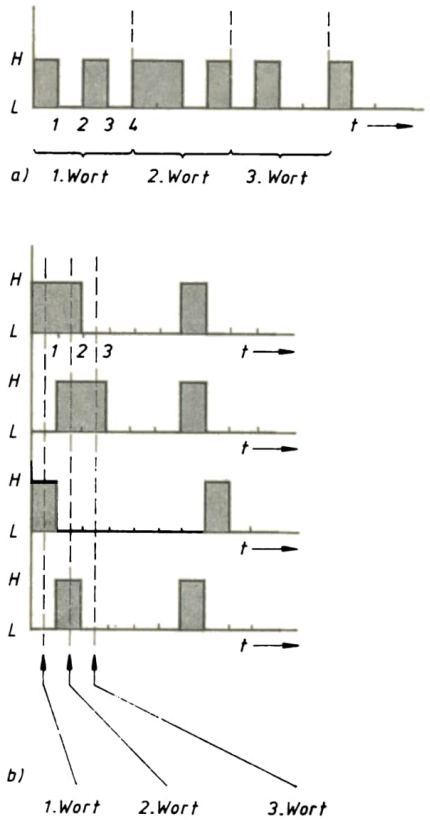


Bild 31. Wortbildung am binären Signal (Beispiel: 1 Wort = 4 bit; H, L Pegel des Signals, *t* Zeit in Schritten der Bitteilung).
a) wortserielle Darstellung des binären Digitalsignals
b) wortparallele Darstellung des binären Digitalsignals

Damit ist gezeigt, wie eine einzelne Binärstelle, also ein Bit, im Mikrorechner technisch abgebildet wird. Nun ist noch zu erläutern, wie dies für ein ganzes Byte oder für ein Wort aus mehreren Bytes geschieht. Hierzu gibt es zwei Möglichkeiten. Die erste besteht darin, daß man die einzelnen Bits eines Wortes zeitlich parallel führt und für jede Bitstelle eine Signalleitung vorsieht (*bitparallele* Darstellung, Bild 31a). Die zweite Möglichkeit geht davon aus, die Bits eines Wortes zeitlich nacheinander auf einer Signalleitung zu übergeben und jedem Bit ein bestimmtes Zeitintervall eines binären Signals zuzuordnen (*bitserielle* Darstellung, Bild 31b). Signale, in denen eine dieser Wortbildungen vorgenommen wird, sind digitale Signale. Erst mit ihrer Hilfe ist ein Funktionieren des Digitalrechners möglich. In Wirklichkeit sind alle Prozesse, die im Rechner oder in einem elementaren digitalen System ablaufen, «Operationen» mit derartigen Signalen (Bild 32).

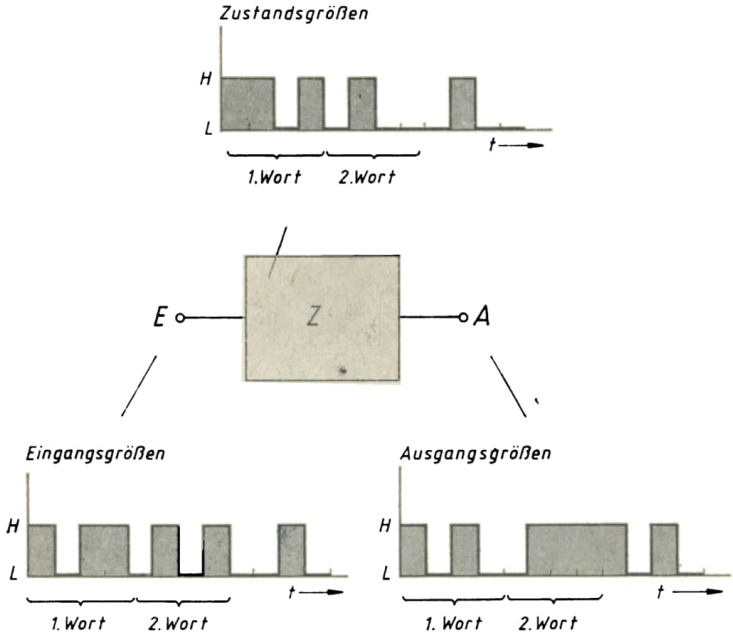


Bild 32. Beispiele der Eingangs-, Zustands- und Ausgangsgrößen eines elementaren Digitalsystems (1 Wort = 4 bit; E Eingang, A Ausgang, Z Zustand des Systems)

4.5. Elementarprozesse im Mikrorechner

Im Mikrorechner lassen sich die folgenden Vorgänge abgrenzen, die mit Hilfe der gebildeten Binärworte ablaufen:

Informationsübertragungen
Die Binärworte mit den darin enthaltenen Informationen werden im Rechner von einem Punkt zum anderen übertragen, indem die Binärbelegungen des Wortes, parallel oder seriell gestaffelt, den Übertragungssignalen aufgeprägt werden (Bild 33). Die Übertragung selbst läuft vornehmlich über das BUS-System des Mikrorechners ab. Eine

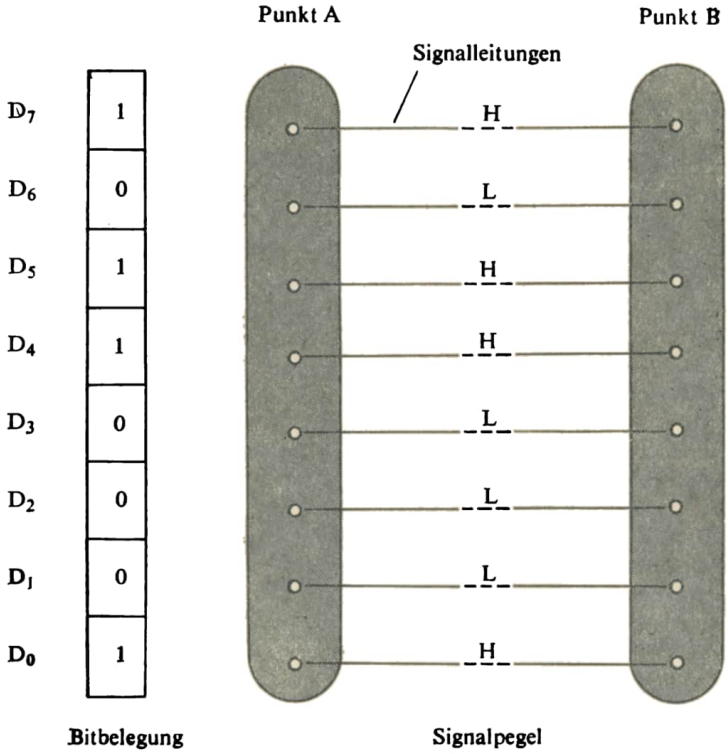


Bild 33. Schematische Darstellung der bitparallelen Übertragung von 1 byte (Belegung 10110001) zwischen Punkt *A* und *B*

elektrische Übertragung erfolgt über Drahtverbindungen, eine optoelektronische über Glasfaserverbindungen.

Informationsübertragungen im Wortformat finden nicht nur innerhalb des Mikrorechners, sondern auch nach außen über die Ein- und Ausgabeeinheiten statt.

Informationsspeicherung
Die Binärworte werden gespeichert, indem der elektrische Zustand von binären Elementarspeichern übernommen und festgehalten wird. Ein solches Speicherelement ist in der Lage, genau 1 bit zu speichern; sein logischer Pegel kann daher «L» oder «H» betragen. Da jedoch ganze Worte zu speichern sind, wird ein Speicher immer byteweise eingeteilt. Hierzu sind jeweils acht Elementarspeicher zu 1 byte zusammengefaßt. Das «Entspeichern» von Informationen, also das Lesen des Speichers, erfolgt, indem jedes Speicherelement wortweise abgefragt und der entsprechende Zustand H oder L einem digitalen Signal aufgeprägt wird.

Informationsverknüpfung
Im Verlauf der Arbeit eines Mikrorechners werden die binären Worte mit den darin enthaltenen Informationen nach bestimmten Regeln beeinflußt (manipuliert) oder mehrere Worte miteinander verknüpft. Auch hier wird – ähnlich wie bei der Speicherung – im Byteformat oder einem Vielfachen davon gearbeitet. Für die Verknüpfung von Informationen sind im Mikroprozessor oder Mikrorechner meist spezielle Einheiten (Verarbeitungs- oder Recheneinheiten) vorgesehen.

4.6. Informationsarten im Mikrorechner

Im Zusammenhang mit den im Mikrorechner ablaufenden Elementarprozessen Übertragen, Speichern und Verknüpfen kann man verschiedene Arten von Informationen feststellen, die, Binärworten aufgeprägt, zwischen den einzelnen Funktionseinheiten des Rechners ausgetauscht werden:

Steuerinformationen
Sie sorgen zum einen für die exakte Funktion und Zusammenarbeit der einzelnen Funktionseinheiten des Mikrorechners. Außerdem enthalten sie «Arbeitsvorschriften» über durchzuführende Operationen und ablaufende Prozesse im Mikrorechner.

Daten
Das sind die eigentlich im Mikrorechner zu bearbeitenden Größen. Sie liegen als gespeicherte Informationen und als Ein- oder Ausgabegrößen vor. In der Praxis hat man sehr viele Datenarten zu unterscheiden. Die Auswahl erfolgt in Abhängigkeit vom konkreten Verwendungszweck des Mikrorechners.

Adreßinformationen
Sie kennzeichnen die Speicherplätze im Mikrorechner, in denen Informationen oder Teilinformationen, die in den ersten beiden Punkten genannt wurden, gespeichert sind oder zu denen solche Informationen hin- bzw. wegtransportiert werden sollen. Sie sind vergleichbar mit dem «Ort» oder der «Nummer» eines Speicherfaches, wobei eine Adresse angibt, wo man dieses Speicherfach im Mikrorechner erreicht.

Die aufgeführten Arten der Informationen müssen vom Mikrorechner exakt unterschieden werden. Wie dies erfolgt, wird im nächsten Abschnitt gezeigt.

4.7. Codierung der Informationen

Mit Hilfe der Wortbildung lassen sich im Prinzip nur Binärbelegungen erzeugen. Der Mikrorechner arbeitet im Grunde genommen auch nur mit solchen Größen. In der praktischen Anwendung der Mikrorechentechnik müssen jedoch eine Reihe von Größen, wie verschiedene Zahlentypen, Zeichen und Schriftsymbole sowie Steuer- und Adreßinformationen dargestellt werden. Um alle diese Größen und Informationen bearbeiten zu können, findet eine Verschlüsselung oder Codierung Anwendung. Dazu werden die entsprechenden Informationen in diskrete Einzelinformationen zerlegt und ausgewählten Binärworten zugeordnet. Man erreicht auf diese Weise eine sogenannte Abbildung aller Informationen auf die Menge der Binärworte. Eine solche Zuordnungsvorschrift wird auch als *Code* oder *Codetabelle* und die entsprechenden Schlüsselworte als *Codeworte* bezeichnet. Im Falle der Codierung mit Hilfe n-stelliger Binärworte spricht man von einem *Binärcode* bzw. von Binärcodeworten.

Eine Codetabelle hat folgenden Aufbau: In einer Spalte der Tabelle stehen die zu codierenden Einzelinformationen, z. B. Operationen, verschiedenartige Zahlen oder Symbole, und in einer zweiten Spalte die entsprechenden binären Codeworte (Bild 34).

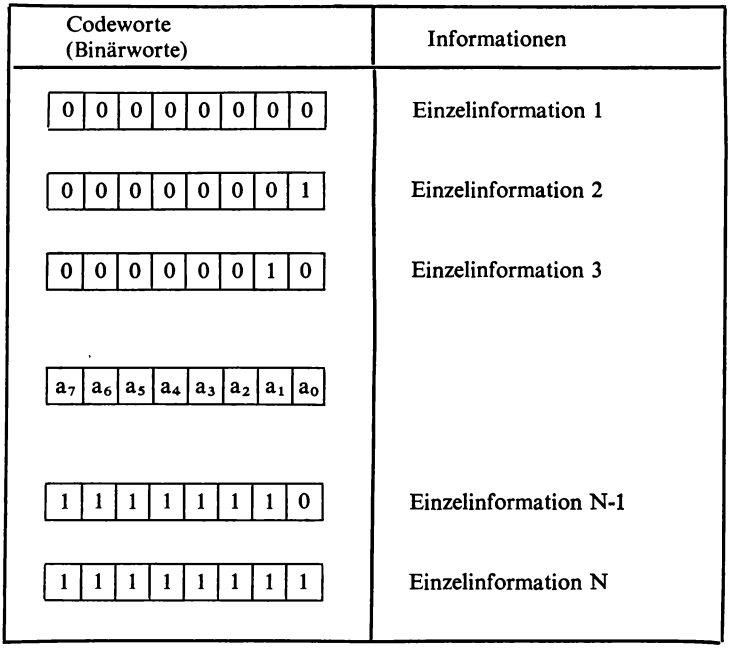


Bild 34. Beispiel einer Codetabelle für $N = 2^n = 2^8 = 256$ Einzelinformationen

Besonders wichtig ist die Frage, wieviel solcher Einzelinformationen den Binärworten mit einer bestimmten Stellenzahl eindeutig zugeordnet werden können. Darüber gibt das Grundgesetz der Codierungstheorie Auskunft. Es besagt, daß maximal soviel Einzelinformationen codiert werden können, wie verschiedene Bitbelegungen oder Bitkombinationen eines Wortes zur Verfügung stehen. Also gilt für die Anzahl E der codierbaren Einzelinformationen die Beziehung

$$E \leqq N = 2^n$$

wobei n die Stellenzahl der Binärworte ist. Liegen beispielsweise wie im Bild 34 Binärworte im Format zu 1 byte vor, so sind maximal $2^8 = 256$ verschiedene Binärbelegungen aufzustellen und damit genausoviel Einzelinformationen zu codieren. Hat man demgegenüber 2-byte-Worte, so können bereits $2^{16} = 65536$ verschiedene Binärworte gebildet und auch entsprechend so viele Einzelinformationen codiert werden.

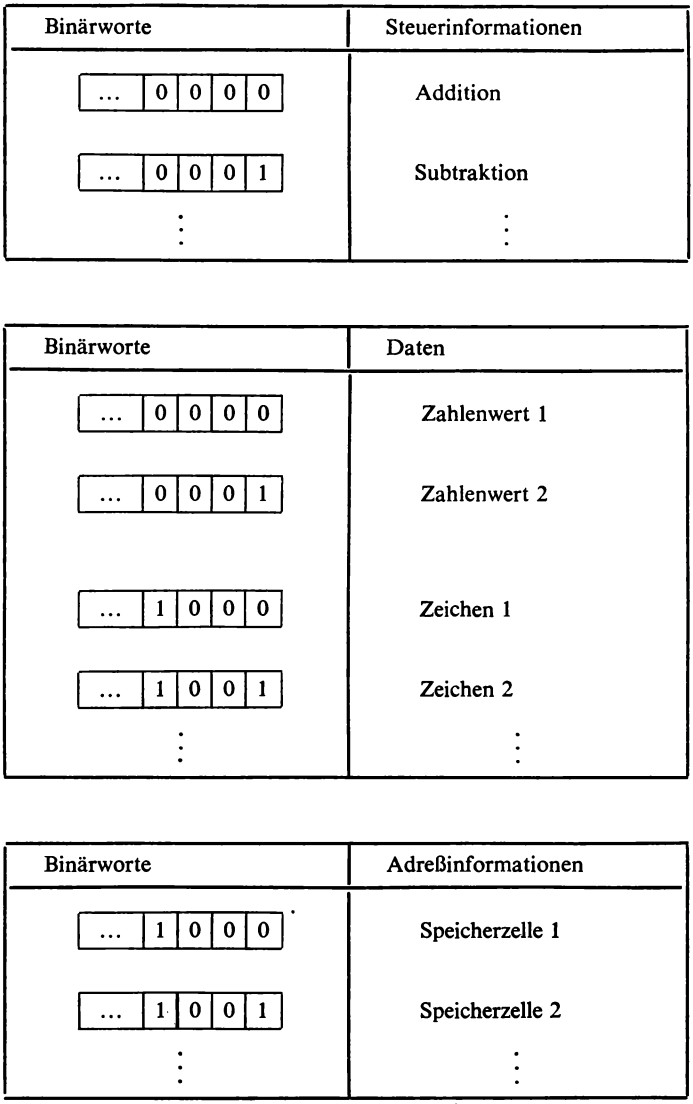


Bild 35. Möglichkeiten der Codierung von Informationen im Mikrorechner; a) Codierung der Steuerinformation (Befehlssatz), b) Codierung der zu verarbeitenden Daten, c) Codierung der Adreßinformationen

Um die verschiedensten Informationen im Mikrorechner abzubilden, existiert für jede Informationsart eine derartige Zuordnungsvorschrift. Da sich die Binärzahlen in den einzelnen Codetabellen jedoch nicht unterscheiden, wird zusätzlich vom Mikrorechner eine bestimmte Auslegung oder *Interpretation* der Binärworte, z.B. je nach Zeitpunkt des «Betrachtens», vorgenommen sowie ein und derselben Binärzahl eine unterschiedliche Bedeutung zugeordnet. Folgende In-

Tabelle 6. Standardisierter rechnerinterner Code und Übertragungscode zwischen rechentechnischen Einrichtungen zur Darstellung von Zeichenketten; Auszug von gemeinsamen Zeichen aus internationalen Standards

Zeichen	Codewort	Zeichen	Codewort
A	1000001	0	0110000
B	1000010	1	0110001
C	1000011	2	0110010
D	1000100	3	0110011
E	1000101	4	0110100
F	1000110	5	0110101
G	1000111	6	0110110
H	1001000	7	0110111
I	1001001	8	0111000
J	1001010	9	0111001
K	1001011	:	0111010
L	1001100	;	0111011
M	1001101	<	0111100
N	1001110	=	0111101
O	1001111	>	0111110
P	1010000	?	0111111
Q	1010001	!	0100001
R	1010010	"	0100010
S	1010011	%	0100101
T	1010100	'	0100111
U	1010101	(	0101000
V	1010110	)	0101001
W	1010111	*	0101010
X	1011000	+	0101011
Y	1011001	,	0101100
Z	1011010	–	0101101
		.	0101110
		/	0101111

terpretationen von Binärworten sind im Mikrorechner üblich:
- als Steuerinformation (z.B. Befehle);
- als Daten (z.B. numerische Werte und Zeichen, Buchstaben, Ziffern, Sonderzeichen, also Punkt, Komma, usw.);
- als Adressen (z.B. Speicherzellen des Hauptspeichers).

Der schematische Aufbau der entsprechenden Codetabellen ist im Bild 35 gezeigt. Wichtige Codes sind bereits standardisiert, z.B. Zeichencodes (s. Tabelle 6).

5. Struktur und Funktionsprinzipien bei Mikrorechnern

5.1. Das Modell eines Mikrorechners

Nachdem untersucht wurde, welche Informationen im Mikrorechner auftreten und wie sie sich codieren lassen, sollen im folgenden die Struktur und die funktionellen Eigenschaften eines Mikrorechners näher betrachtet werden. Derartige Problemkreise sind unter anderem auch Gegenstand der Fachdisziplin *Rechnerarchitektur*. Sie befaßt sich mit der modellmäßigen Darstellung, der Bewertung und der Beschreibung des zweckbezogenen Aufbaus von Rechnersystemen. Als kleinste Strukturelemente dienen Prozessoren, Speichereinheiten, Ein-/Ausgabeeinheiten und Verbindungseinheiten. Der Inhalt der Elemente wird nicht im Detail betrachtet, sondern mehr ihre Wirkungsweise.

Zunächst analysiert man jedoch die Gesamtstruktur und Funktionsweise des Mikrorechners an Hand einer Modelldarstellung, um dann besser die Eigenschaften und Wirkungsweise der einzelnen Funktionseinheiten sowie das Zusammenwirken im Gesamtsystem zu verstehen.

Als Ausgangspunkt soll ein Blockbild des Mikrorechners dienen, das man erhält, wenn insbesondere steuerungstechnische Gesichtspunkte zu berücksichtigen sind (Bild 36). Es ist ein relativ einfaches Modell, zeigt aber deutlich das Prinzip eines mehrfach gegliederten Aufbaus der wichtigsten Steuerfunktionen. Hierbei wird folgende Funktionsweise sichtbar: Der Mikroprozessor «holt» sich seine für die Arbeit notwendigen Instruktionen, die Befehle (Steuersignale B) aus dem Programmspeicher, indem er bestimmte Steuerinformationen und Adressen (Steuersignale A) an den Speicher aussendet. Er erhält dadurch die einzelnen Anweisungen und führt diese der Reihe nach aus. Dabei werden bestimmte Funktionseinheiten im Mikroprozessor und Mikrorechner in Gang gesetzt, die den Austausch von Informationen zwischen den einzelnen Einheiten sowie die Verarbeitungsprozesse im Mikrorechner steuern (Steuersignale A'). Außerdem

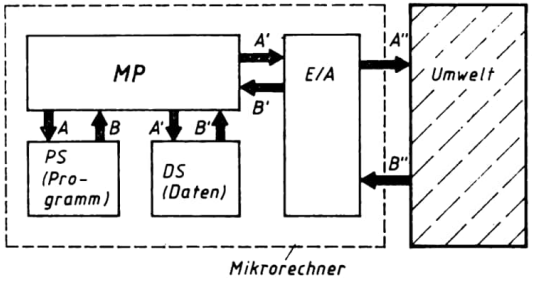


Bild 36. Steuerfluß im Mikrorechner (MP Mikroprozessor, E/A Ein- und Ausgabeeinheiten, PS Programmspeicher, DS Datenspeicher, A, B Steuerinformationen)

erhält der Mikroprozessor Rückinformationen (Steuersignale B') über den vorschriftsmäßigen Ablauf aller Vorgänge. Die gleichen Betrachtungen gelten auch für den Mikrorechner, wenn man ihn als ein Gesamtsystem auffaßt. Er steuert die Peripherie, also außerhalb stattfindende Prozesse (Steuersignale A'') und erhält von dort Rückinformationen (Steuersignale B'') als Bestätigung.

Eine andere Modellbeschreibung ergibt sich, wenn der physikalische Aufbau – man sagt auch die physische Struktur – in den Mittel-

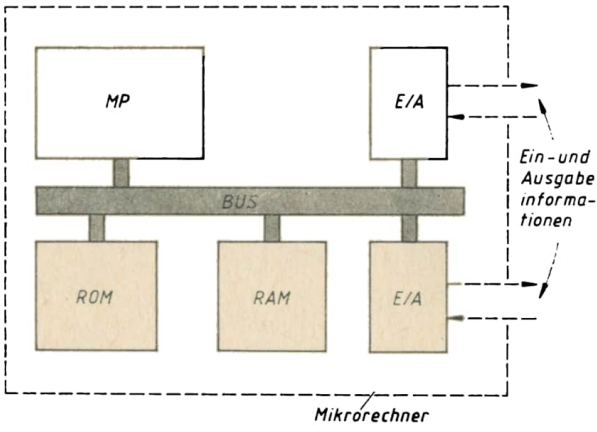


Bild 37. Grundstruktur eines Mikrorechners (MP Mikroprozessor, ROM Nur-Lese-Speicher [read only memory], RAM Schreib-Lese-Speicher [read acces memory], E/A Ein- und Ausgabeeinheiten, BUS Verbindungs- und Transportsystem)

punkt gerückt wird. Hierbei wird mehr die wirkliche Gestalt des Mikrorechners, also die Aufteilung in Baugruppen oder in die entsprechenden integrierten Schaltkreise sichtbar (Bild 37). Diese Darstellung soll in den weiteren Ausführungen als Grundlage dienen.

5.2. Struktur und Eigenschaften des Mikroprozessors

Der Mikroprozessor ist die komplizierteste und vom Verständnis her schwierigste Einheit des Mikrorechners. Während des Betriebs findet zwischen dem Mikroprozessor und den übrigen Einheiten des Mikrorechners ein Austausch von verschiedensten Informationen statt. Die wichtigsten sind Steuer- und Adreßinformationen sowie Daten (Bild 38). Außerdem erhält der Mikroprozessor von einem stabilen Taktgenerator den exakten Zeitmaßstab (Takt). Dieser ist gewissermaßen die «innere Uhr», nach der sich der Mikroprozessor zu richten hat.

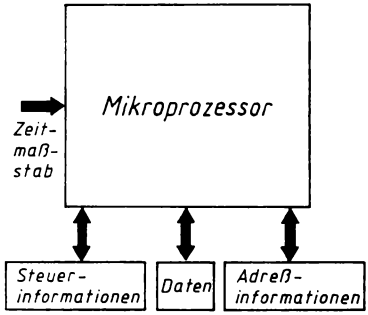


Bild 38. Der Mikroprozessor mit seinen ein- und auslaufenden Informationen

Der Mikroprozessor als hochintegrierter Schaltkreis tauscht sämtliche Informationen über seine Chipanschlüsse mit der Umgebung aus. Die Anzahl der Anschlüsse ist technologisch begrenzt; häufig beträgt sie 40 oder 64. Anzahl, Aufteilung (Anschlußschema) und bestimmte elektrische Kenngrößen der Anschlüsse sowie bestimmte funktionelle Eigenschaften der Mikroprozessorschaltkreise sind oft standardisiert, so daß Mikroprozessoren gleichen Typs, aber von verschiedenen Herstellern, gegenseitig ersetzt werden können [16, 17, 18, 19, 20].

Das Strukturmodell eines Mikroprozessors ist im Bild 39 wiedergegeben. Die *zentrale Steuereinheit* koordiniert alle im Mikroprozessor ablaufenden Prozesse. Dazu erhält sie Informationen von allen Einheiten des Mikroprozessors, insbesondere aber von der *Befehlsinterpretationseinheit*. Letztere erkennt die in den Mikroprozessor einlaufenden Befehle, decodiert und löst sie in einzelne Arbeitsschritte auf. Auf der Grundlage der entschlüsselten Informationen werden damit die Funktionen der Registereinheit, der Verarbeitungseinheit und der Datenaustausch zwischen Mikroprozessor und Mikrorechner gesteuert.

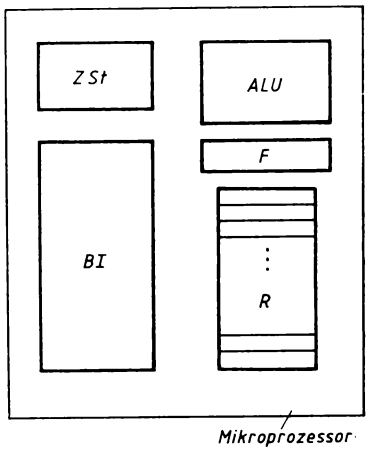


Bild 39. Schematischer Aufbau eines Mikroprozessors (ZSt Zentrale Steuereinheit, BI Befehlsdecodier- und -interpretationseinheit, ALU Arithmetik-Logik-Einheit, F Anzeigefeld, R Registereinheit)

Die *Registereinheit* (Registersatz) ist eine Art Schnellspeicher zur kurzzeitigen Zwischenspeicherung von Daten und anderen wichtigen Informationen. Sie ist ein byte- und wortorientierter Arbeitsspeicher mit einer Speicherkapazität von etwa 10 bis einigen 100 byte und relativ hoher Arbeitsgeschwindigkeit. Die Speichereinheit des Registersatzes ist ein einzelnes Register. Dieses kann im allgemeinen ein Byte oder ein Wort zwischenspeichern.

Die Verarbeitungseinheit ist in der Lage, Daten nach arithmetischen oder logischen Gesetzen miteinander zu verknüpfen, daher auch die Bezeichnung «Arithmetik-Logik-Einheit» (ALU – arithmetic logic unit). Das Ergebnis einer Operation steht in der Regel in einem be-

sonders dafür eingerichteten Zwischenspeicher, dem Akkumulator (Rechenregister), der Bestandteil des Registersatzes ist. Häufig bieten auch mehrere oder alle Register des Mikroprozessors diese Möglichkeit des Rechenregisters. Besondere Mitteilungen über das Ergebnis der Operation, z.B. «Ergebnis gleich null» oder «Ergebnis ist negativ», werden in einem besonderen Anzeigefeld (als «Flag»-Register bezeichnet) codiert abgespeichert. Sie stehen dort zur Weiterverarbeitung oder Auswertung zur Verfügung.

Außer der Funktion des Rechenregisters können bestimmte Register auch noch andere Aufgaben übernehmen, etwa Adressierungsfunktionen als Befehlszähler, Index- oder Zeigerregister. In diesem Fall wird der Inhalt eines Registers als ein Adreßwert aufgefaßt, der auch entsprechend manipuliert werden kann.

Ein Mikroprozessor ist in der Lage, eine Vielzahl verschiedener Funktionen oder Operationen auszuführen. Der Umfang aller möglichen Operationen wird vom Hersteller des Mikroprozessors festgelegt. Er stellt dem Anwender für jeden Typ eine sogenannte Befehlstabelle oder *Befehlsliste* zur Verfügung. Sie ist eine Codetabelle, in der die binären Steuerworte, die Befehle und die entsprechenden, vom Mikroprozessor ausführbaren Operationen gegenübergestellt sind. Die Menge aller möglichen Befehlszuordnungen heißt auch *Befehlssatz*. Neben der ebenfalls vom Hersteller festgelegten maximalen *Operationsgeschwindigkeit*, d.h. der Anzahl der ausführbaren Operationen je Zeiteinheit, ist der Befehlssatz die wichtigste Kenngröße eines Mikroprozessors und bestimmt dessen Leistungsfähigkeit.

Der *Funktionsablauf* im Mikroprozessor ist durch einen schrittweisen Betrieb gekennzeichnet. Den Grundschritt bildet das vom Taktgenerator gelieferte Taktsignal (Bild 40). Im Zeitraster dieses

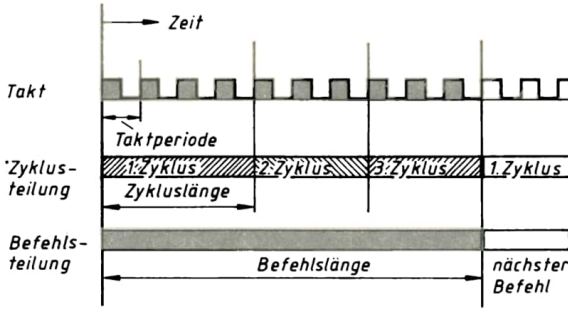


Bild 40. Zeitlicher Aufbau eines Befehls

Taktes sendet der Mikroprozessor schrittweise Steuer- und Adressiersignale aus, empfängt ebenso Befehle und führt diese Schritt für Schritt aus. Diese Betriebsweise ist für den gesamten Mikrorechner typisch. Jeder Befehl, den der Mikroprozessor abarbeitet, ist in eine bestimmte Anzahl von Zyklen unterteilt. Jeder Zyklus setzt sich wiederum aus einer bestimmten Anzahl von Taktimpulsen zusammen. Der erste Zyklus jedes Befehls ist der sogenannte *Befehlsabruf*, d.h. das «Holen» des Befehls vom Hauptspeicher. Dann folgen ein oder mehrere Zyklen zur Durchführung der Operation, z.B. ersten Operand holen, zweiten Operand holen, beide Operanden verknüpfen. Ist ein Befehl beendet, folgt automatisch der Abruf des nächsten Befehls.

5.3. Speichersystem

Die neben dem Mikroprozessor wichtigste Funktionseinheit im Mikrorechner ist der Halbleiter-Hauptspeicher. In ihm sind die Befehlsfolge und die für deren Abarbeitung notwendigen Daten enthalten. Die zu speichernden Informationen werden in vorher bestimmte Speicherplätze «eingeschrieben» und verbleiben dort solange, bis man sie zur Aufnahme neuer Informationen überschreibt oder löscht. Liegen die Informationen gespeichert vor, dann können sie beliebig oft, ohne Zerstörung «gelesen» werden. Da im Speicher des Mikrorechners alle Informationen binär darzustellen sind, müssen die Speicherelemente auch nur die Binärbelegung 0 oder 1, also jeweils 1 bit, aufbewahren können. Diese Tatsache trägt sehr stark zum Vergrößern der Sicherheit und Vereinfachen aller Speichervorgänge bei.

Im Bild 41 ist die Grundstruktur des Hauptspeichers dargestellt. In der Regel ist er byteorientiert, d.h., jeder Speicherplatz enthält acht Elementarspeicherzellen. Jedes Byte ist mit einer Speicherplatznummer, der Adresse «versehen». Bei Lese- oder Schreibvorgängen wird jeweils das gesamte Byte der betreffenden Adresse angesprochen, d.h. abgespeichert oder ausgelesen. Die Adressen selbst sind ebenfalls binär codiert. Dazu ordnet man jedem Speicherplatz, der durch eine bestimmte natürliche Zahl gekennzeichnet ist, ein Binärwort zu. Infolge dieser Wortdarstellung der Adressen ist der Adreßumfang eines Speichers, also die Anzahl der ansprechbaren Speicherplätze, begrenzt und kann maximal nach der Beziehung

$$S = 2^k$$

betragen. S ist die Anzahl der Speicherplätze und k die Adreßbreite, d.h. die Anzahl der Bitstellen im binären Adreßwort. Mit einer

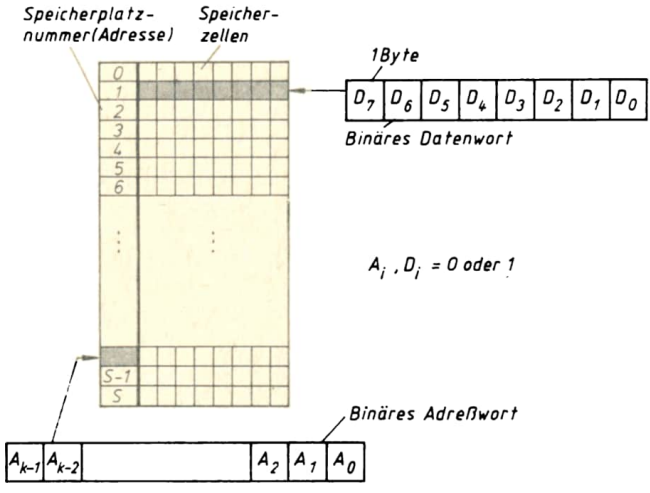


Bild 41. Grundstruktur eines Speichers (Anzahl der Speicherplätze $S = 2^k$; k Adreßbreite)

Adreßbreite von beispielsweise 8 bit ist es demnach möglich, einen Speicher mit 256 Speicherplätzen zu adressieren, mit 9 bit bereits 512 usw. In der Praxis der Mikrorechentechnik beträgt die Adreßbreite in der Regel 16 bis maximal 24 bit. Die entsprechende Anzahl der damit adressierbaren Speicherplätze ist in Tabelle 7 aufgeführt. Die Größe eines Speichers wird im allgemeinen in Anzahl der Byte angegeben. Eine Größe von 1024 byte (entsprechend einer Adreßbreite von 10 bit) wird auch mit 1 Kbyte (sprich «kilobait») abgekürzt. Daß

Tabelle 7. Zusammenhang zwischen Breite des Adreßwortes und maximaler Speicherkapazität bei einem byteorientierten Speicher

Adreßbreite (Anzahl der Bits)	Speicherkapazität (Anzahl der Bytes)
8	256 byte
10	1 Kbyte (1024 bit)
16	64 Kbyte
18	256 Kbyte
20	1 Megabyte (1024 Kbyte)
22	4 Mbyte
24	16 Mbyte

hier «kilo» nicht, wie allgemein üblich, für 1000 steht, sondern K = 1024 gesetzt ist, hat seine Ursache im binären Zahlensystem.

Außerdem ist es auch möglich, die Größe eines Speichers in der Anzahl der speicherbaren Bits anzugeben, wobei hier die Abkürzung 1 Kbit = 1024 bit verwendet wird. Ein Speicher von beispielsweise 256 byte hat die Größe von 8×256 bit = 2048 bit = 2 Kbit. Diese Einteilung wird häufig bei integrierten Speicherschaltkreisen zur Kennzeichnung der Speicherkapazität angewendet.

Bei der technischen Realisierung eines Speichers werden die Adressen der Speicherplätze in Form von binären Worten über das BUS-System zugeführt und in der Adreßcodierung gemäß der erwähnten Zuordnung Binärwort zu Speicherplatznummer entschlüsselt. Die zu speichernden oder auszulesenden Informationen lassen sich über einen Datenbus zu- bzw. abführen (Bild 42).

Für die Praxis ist ein Speicher mit möglichst kleiner *Zugriffszeit* wünschenswert. Darunter versteht man die Zeit, die vergeht, bis ein an den Speicher gerichteter Schreib- oder Lesebefehl ausgeführt ist. Sie ist wesentlich vom Aufbau und von der Art des verwendeten Speichers abhängig. Im allgemeinen sind Speicher mit sehr kleiner Zugriffszeit jedoch aufwendiger und teurer.

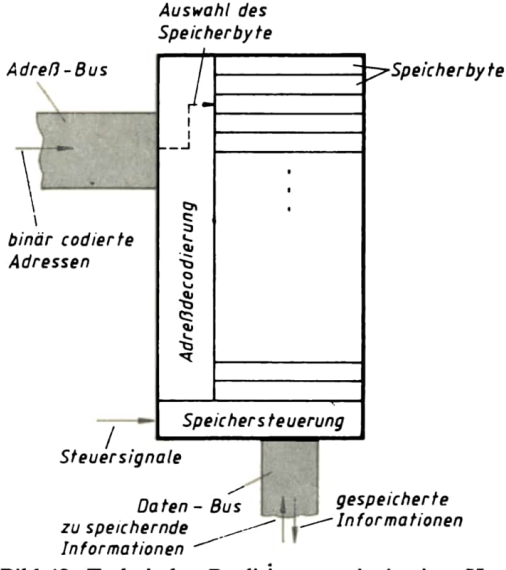


Bild 42. Technisches Realisierungsprinzip eines Hauptspeichers

Bei Mikrorechnern ist der Hauptspeicher aus technologischen Gründen meistens in zwei Bereiche gegliedert:
- *ROM-Speicher* (read only memory): Nur-Lese-Speicher;
- *RAM-Speicher* (random acces memory): Schreib-Lese-Speicher.

Beide bestehen aus verschiedenartigen integrierten Schaltkreisen. Im *ROM-Bereich* ist es nicht möglich, die gespeicherten Binärworte zu verändern, daher auch die Bezeichnung *Festwertspeicher*. Im *RAM-Bereich* können beliebig oft binäre Informationen gespeichert und auch ohne Löschung der Information wieder ausgelesen werden. Allerdings verliert der RAM beim Abschalten der Betriebsspannung alle gespeicherten Informationen.

ROM-Speicherschaltkreise sind einfacher aufgebaut als RAM-Schaltkreise und damit auch ökonomischer herzustellen. Um jedoch in einem Festwert-Speicherschaltkreis Informationen einzuschreiben, muß in einem dem Einsatz vorhergehenden Prozeß der Schaltkreis programmiert werden. Hierbei hat man zu unterscheiden, ob dies gleich während des Herstellungsprozesses geschieht (maskenprogrammierter ROM) oder ob der Anwender das Programm mit Hilfe eines besonderen Programmiergeräts selbst einschreibt (programmierbarer

Tabelle 8. Eigenschaften von Halbleiterspeichern

Speicherart	Eigenschaften
ROM – *Nur-Lese-Speicher* (read only memory)	Programmierung während des Herstellungsprozesses; Änderung der Information nicht möglich, Speichervolumen 2048 bit bis 256 Kbit je integrierter Schaltkreis; hohe Zuverlässigkeit.
PROM – *Programmierbarer* Nur-Lese-Speicher (programmable ROM)	Durch Anwender einmalig programmierbar, danach Information nicht mehr änderbar; Speichervolumen 2048 bit bis 128 Kbit je Schaltkreis.
EPROM – *Mehrmals* programmierbarer ROM (erasable PROM)	Durch Anwender *n*-malig programmierbar, löschbar durch UV-Bestrahlung; Speichervolumen 2048 bit bis 64 Kbit je Schaltkreis.
RAM – *Schreib-Lese-Speicher* (read acces memory)	Informationen können beliebig oft eingeschrieben oder gelesen werden, Umschalten auf Schreiben bzw. Lesen durch Steuersignal; Informationsverlust beim Abschalten der Betriebsspannung; Speichervolumen 256 bit bis 256 Kbit je Schaltkreis.

ROM = PROM). Bestimmte PROM-Typen können auch nachträglich wieder gelöscht und erneut programmiert werden. Weitere Unterscheidungsmerkmale der Halbleiterspeicher sind in Tabelle 8 aufgeführt.

An einen Mikrorechner können zusätzlich auch sogenannte *externe Speicher* angeschlossen werden. Das sind Speicher mit relativ hoher Speicherkapazität, jedoch auch größerer Zugriffszeit. Sie arbeiten also langsamer. Praktische Bedeutung haben bei Mikrorechnern besonders Kassetten-Magnetbandgeräte, Folienspeichergeräte, sogenannte «Floppy Disk»-Einheiten und Lochbandgeräte. Die entsprechende Speicherhierarchie eines Mikrorechners zeigt Bild 43. Hier sind die Speichermedien nach ihrer Arbeitsgeschwindigkeit und Speicherkapazität angeordnet.

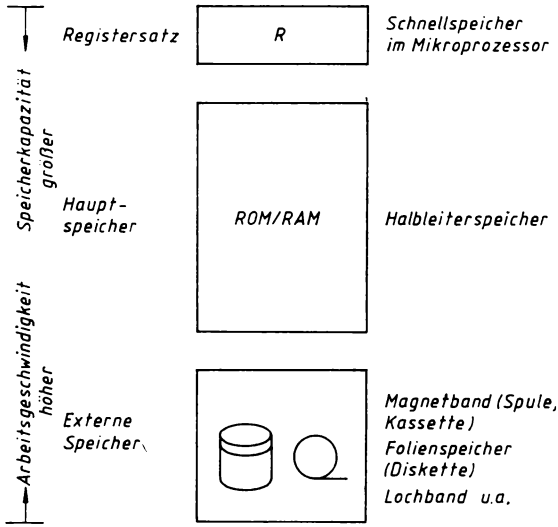


Bild 43. Schematische Darstellung der Speicherarten eines Mikrorechners

5.4. Ein- und Ausgabesystem

Um den Informationsaustausch mit der Umwelt, also den peripheren Einrichtungen, in geeigneter Weise durchzuführen, werden besondere Ein- und Ausgabeeinheiten im Mikrorechner vorgesehen. Da die zu steuernden Vorgänge, Einrichtungen oder Geräte von sehr unterschiedlicher Art sein können, kommt diesen Einheiten vor allem die

Funktion der Wandlung und Anpassung von Signal- und Informationsdarstellungen sowie elektrischer Kenngrößen zu. Außerdem haben sie die Aufgabe, den Informationsaustausch zwischen Mikrorechner und Peripherie zu steuern (Bild 44). Häufigste Wandlungs-

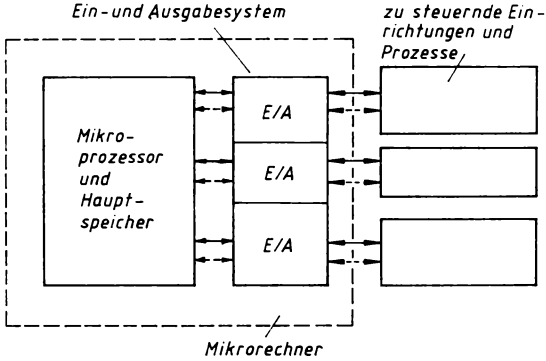


Bild 44. Ein- und Ausgabesystem im Mikrorechner (↔ Datenfluß, ↔ Steuerfluß)

funktion ist die Veränderung der Übertragungsgeschwindigkeit von Informationen. Während im Mikrorechner sehr hohe Geschwindigkeiten möglich sind, ist dies in den externen Einrichtungen meist nicht der Fall. Oft tritt auch das Problem der Umwandlung von digitalen Signalen in analoge, bzw. umgekehrt auf. Hier sind sogenannte Digital/Analog- und Analog/Digital-Wandler erforderlich (Bild 45).

Bei Prozeßsteuerungen im Echtzeitbetrieb ist es häufig notwendig, im Mikrorechner schnelle Reaktionen auf besondere externe Ereignisse einzuleiten. Dies wird dem Rechner in der Regel durch sogenannte Alarm- oder Interruptsignale mitgeteilt. Sie bewirken eine sofortige Unterbrechung seiner Arbeit an der bisherigen Aufgabe und das Anlaufen eines anderen, des sogenannten Unterbrechungs-Behandlungsprogramms. Erst wenn die einzuleitenden Sondermaßnah-

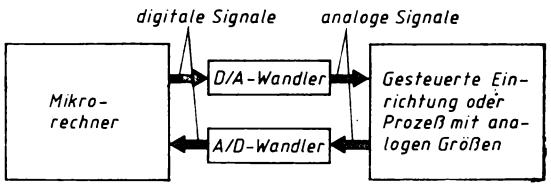


Bild 45. Steuerung von analog arbeitenden Systemen und Prozessen

men beendet sind, nimmt der Mikrorechner die Arbeit an der davorliegenden Aufgabe wieder auf. Auch eine solche Unterbrechungssteuerung läuft über die Ein- und Ausgabeeinheiten ab.

Aus der Vielzahl der Realisierungsmöglichkeiten von externen Einrichtungen ergibt sich natürlich ein relativ hoher Aufwand an technischen Mitteln, da im Prinzip jede zu steuernde Einrichtung ihren «eigenen» Anschluß am Rechner haben müßte. Um dem zu begegnen, wurden für die Peripherie von datenverarbeitungsspezifischen Geräten und ausgewählten Prozeßsteuereinrichtungen sogenannte *Ein- und Ausgabe-* (E/A-) *Schnittstellen* geschaffen. Das sind in der Regel national oder auch international standardisierte Anschlüsse und Leitungen (Anschlußbilder) mit ganz bestimmten elektrischen Kenngrößen oder vorgeschriebene Übertragungsabläufe (Übertragungsprozeduren und -protokolle), die am Mikrorechner bereitgestellt werden. Die Einführung von E/A-Schnittstellen bewirkt eine Trennung, einen «Schnitt» des gesamten Ein- und Ausgabesystems in sogenannte *E/A-Steuereinheiten*, die Bestandteil des Mikrorechners sind, und in spezielle *E/A-Geräte* bzw. den zu steuernden Prozeß, die außerhalb des Mikrorechners liegen (Bild 46). So ist es möglich, an einem Mikrorechner die unterschiedlichsten Geräte ohne technische Veränderungen anzuschließen, auch wenn sie beispielsweise von Herstellern verschiedener Länder stammen. Bedingung ist nur, daß sie die entsprechende Schnittstelle aufweisen.

Die technische Realisierung einer E/A-Steuereinheit kann sehr unterschiedlich erfolgen. Im einfachsten Fall hat man für die Zwischenspeicherung der Informationen einige Steuer- und Speicherregister, die die Ein- und Ausgabe übernehmen. Häufig werden jedoch spezielle programmierbare E/A-Schaltkreise oder noch leistungsfähigere E/A-Prozessoren, in Form hochintegrierter Schaltungen, angewendet. Im letzteren Fall befindet sich in der E/A-Steuereinheit

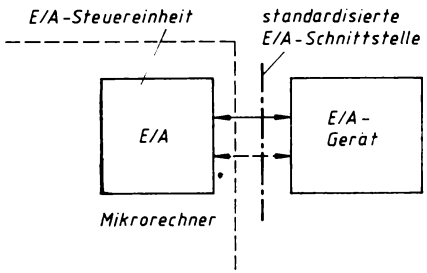


Bild 46. E/A-Schnittstellen am Mikrorechner

selbst wieder ein «kleiner» Mikroprozessor mit Programmspeicher, der die gesamte Abwicklung der Ein- und Ausgabe des «großen» Mikrorechners übernimmt.

5.5. BUS-System

Das BUS-System bildet die technische Grundlage der innerhalb des Mikrorechners ablaufenden Informationsübertragungen. Es besteht aus einem oder mehreren in Zeitteilung betriebenen Leitungsbündeln, dem eigentlichen BUS, und aus den in den einzelnen Teilkomponenten des Mikrorechners, wie Mikroprozessor, Speicher- sowie Ein- und Ausgabeeinheiten, enthaltenen BUS-Anschaltesteuerungen. Zeitteilung heißt dabei, daß zwei oder mehrere solcher Teilsysteme in einem bestimmten Zeitabschnitt den BUS für ihre Übertragung benutzen und zu einer anderen Zeit wieder andere Teilsysteme dies tun, sich also den BUS zeitlich «aufteilen».

Die Struktur eines BUS-Systems ist meist byteorientiert, d.h., die Anzahl der Übertragungsleitungen beträgt acht oder ein Vielfaches davon, und auch die Übertragung selbst erfolgt im byteorientierten Wortformat. Im Mikrorechner kann man im allgemeinen drei Einzelbusse unterscheiden, die entsprechend den auf ihnen zu übertragenden Informationen benannt sind: *Steuerbus*, *Datenbus* und *Adreßbus* (Bild 47). Die Anschaltung an den BUS sowie der Transport der In-

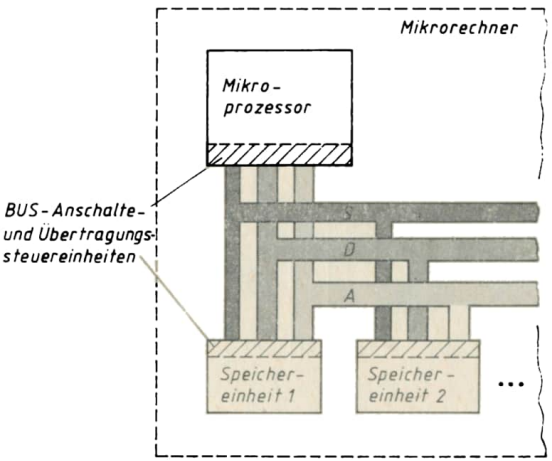


Bild 47. Struktur des BUS-Systems (S Steuerbus, D Datenbus, A Adreßbus)

formationen über ihn erfolgen auf der Grundlage von sogenannten BUS-Zugriffs- und Übertragungsprotokollen. In ihnen sind die Zuschaltung, der Ablauf und die dabei einzuhaltenden Regeln der Übertragung vereinbart.

5.6. Befehlsabarbeitung

Im folgenden sollen die Vorgänge betrachtet werden, die bei der Abarbeitung eines Programms oder, noch genauer, bei der Abarbeitung eines Befehls im Mikrorechner ablaufen. Da die Programmierung erst im Abschnitt 6. behandelt wird, sei zunächst der Fall angenommen, daß ein Programm bereits entworfen ist und im Hauptspeicher bereit steht.

Nach dem Start des Mikrorechners folgt als erste Reaktion des Mikroprozessors das Aussenden einer Adresse aus einem speziellen Register, dem Befehlszähler (auch: Programm- oder Befehlsadreßzähler), über den Adreßbus an den Hauptspeicher. Diese Adresse – auch Startadresse genannt – kennzeichnet jenen Speicherplatz im Hauptspeicher, von dem der erste Befehl zu holen ist. In einem weiteren Schritt wird dann der ausgewählte Befehl über den Datenbus vom entsprechenden Speicherplatz zum Mikroprozessor transportiert (Bild 48). Schließlich folgt die Ausführung des Befehls. Der Ab-

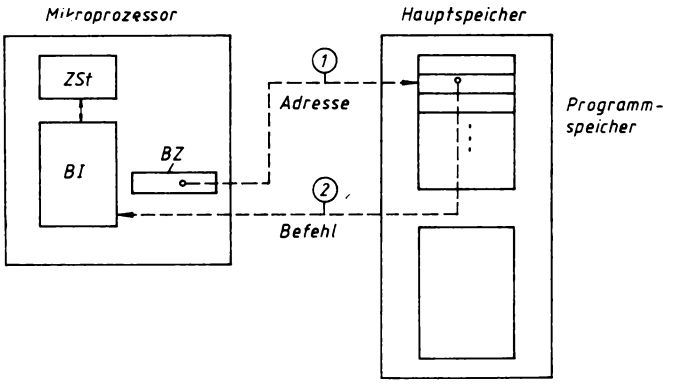


Bild 48. Befehlsabruf (① Aussenden der Adresse, ② Transport des Befehls zum Mikroprozessor; BZ Befehlszähler, ZSt Zentrale Steuereinheit, BI Befehlsinterpretationseinheit)

lauf dieser drei Phasen nennt sich auch Befehlszyklus (Bild 49). Mit dem nächsten Befehl wiederholt sich der Zyklus. Der Mikroprozessor sendet wiederum die im Befehlszähler gespeicherte Adresse aus, die jedoch zwischenzeitlich um eine Befehlseinheit erhöht wurde. Anschließend wird der neue Befehl geholt und ausgeführt. Dies setzt sich in der Weise fort, bis das gesamte Programm abgearbeitet ist.

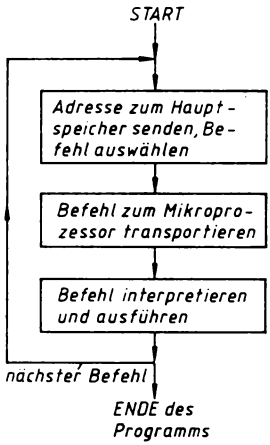


Bild 49. Befehlszyklus des Mikroprozessors

In den ersten beiden Phasen eines Befehlszyklus laufen im Mikrorechner fast immer die gleichen Vorgänge ab. In der Ausführphase finden jedoch, je nach der Art des abzuarbeitenden Befehls, sehr unterschiedliche Aktionen statt, wie:

- Daten einlesen, ausgeben (*Ein- und Ausgabe-Befehle*);
- Daten transportieren und speichern (*Transport-Befehle*);
- Daten verknüpfen (*arithmetisch-logische Befehle*);
- Programmsteuerungen realisieren (*Programmsteuer-Befehle*).

Jede dieser Aktionen wird im Mikroprozessor bzw. Mikrorechner in mehreren Teilschritten abgearbeitet, d.h., die Befehlsausführphase ist weiter untergliedert.

Zunächst sei der *Datentransport* betrachtet (Ein-/Ausgabe- und Transportbefehle). Dieser ist möglich im Mikroprozessor von Register zu Register sowie im Mikrorechner zwischen Mikroprozessor und Hauptspeicher, zwischen Mikroprozessor und E/A-Steuereinheiten sowie zwischen Hauptspeicher und E/A-Steuereinheiten. Der Transport innerhalb des Mikroprozessors erfolgt relativ problemlos.

Alle anderen Datentransporte laufen jedoch unter Benutzung des BUS-Systems in ähnlicher Weise ab wie der Befehlstransport während der Befehlsabrufphase. Auch hier sendet der Mikroprozessor zuerst eine Adresse über den Adreßbus an den Datenspeicher, und anschließend erfolgt über den Datenbus der Transport der Daten, z.B. vom Hauptspeicher zum Mikroprozessor oder umgekehrt (Bild 50).

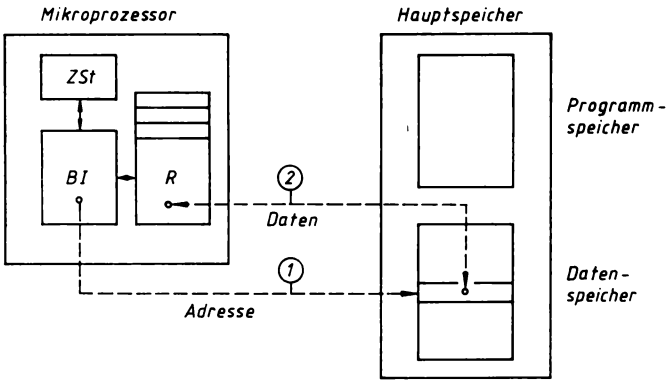


Bild 50. Ausführen eines Datentransports zwischen Hauptspeicher und Mikroprozessor (① Aussenden der Adresse, ② Transport der Daten vom Datenspeicher zum Mikroprozessor bzw. vom Mikroprozessor zum Speicher; ZSt Zentrale Steuereinheit, BI Befehlsinterpretationseinheit, R Registereinheit)

Die *Ein- und Ausgabe* von Daten unterscheidet sich prinzipiell nicht vom Transport innerhalb des Rechners. Anstelle der Hauptspeicheradressen treten die der E/A-Einheiten auf. In ähnlicher Weise erfolgt zunächst eine Adressierung der einzelnen E/A-Einheiten und anschließend der entsprechende Informationstransport.

Das *Manipulieren* und *Verknüpfen* von Daten wird von der Verarbeitungseinheit (ALU) im Mikroprozessor auf der Grundlage von arithmetisch-logischen Befehlen durchgeführt. In diesem Zusammenhang laufen ebenfalls Übertragungsvorgänge ab. So werden zu bearbeitende Daten während der Befehlsausführphase in die Verarbeitungseinheit transportiert und erst anschließend – entsprechend der im Befehl angegebenen Operation – miteinander verknüpft (Bild 51). Das Ergebnis steht dann in einem der Rechenregister. Es gibt auch leistungsfähige Verarbeitungsbefehle, wo man während der Befehls-

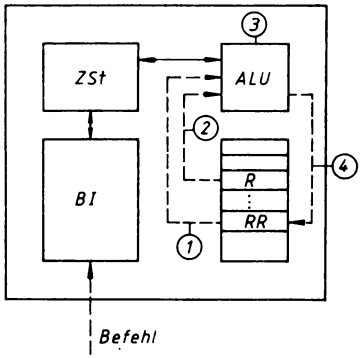


Bild 51. Ausführen eines Verarbeitungsbefehls im Mikroprozessor (① Operand 1 aus Rechenregister RR holen, ② Operand 2 aus einem Register R holen, ③ beide Operanden in der Arithmetik-Logik-Einheit ALU verknüpfen, ④ Ergebnis in Rechenregister RR eintragen, Operand 1 wird dabei «überschrieben»)

ausführung die Operanden aus dem Hauptspeicher holen, verknüpfen und wieder in den Hauptspeicher bringen kann. Welche Vorgänge dabei in einem Mikrorechner ablaufen, demonstriert das Beispiel im Bild 52.

Werden *Programmsteuer-Befehle* abgearbeitet, dann wird der Mikroprozessor während der Befehlsabarbeitungsphase in seiner grundsätzlichen Arbeit beeinflußt, z.B. der Befehlsstrom vom Hauptspeicher sprungartig geändert (Sprungbefehle) oder die Abarbeitung der Befehlsfolge unterbrochen sowie gegebenenfalls ein anderes Programm gestartet (Interruptbefehle, Haltbefehle).

Um alle bei der Bearbeitung der einzelnen Befehle ablaufenden Vorgänge zu steuern und zu kontrollieren, findet ein ununterbrochener Austausch von Steuer-, Status- und Bestätigungssignalen zwischen den einzelnen Einheiten des Mikrorechners statt. Auf diese Weise wird auch eine exakte Zusammenarbeit (Synchronisation) dieser Einheiten während aller Bearbeitungsphasen gewährleistet.

Die Geschwindigkeit aller hier dargestellten Prozesse hängt wesentlich von den technischen Kenngrößen und Eigenschaften der elektronischen Bauelemente des Mikrorechners ab. Grenzgröße aller Übertragungsgeschwindigkeiten im Rechner ist jedoch die Lichtgeschwindigkeit. Ihr Einfluß macht sich heute bereits bei schnellen Mikrorechnern störend bemerkbar.

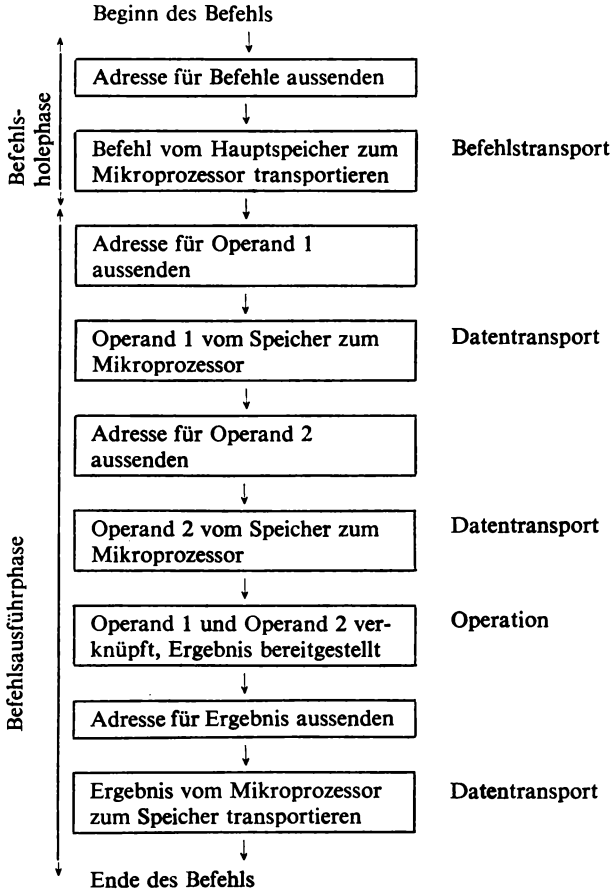


Bild 52. Beispiel für das Ablaufschema eines Arithmetik/Logik-Befehls mit Transport der Operanden vom bzw. zum Hauptspeicher

5.7. Mikrorechnerverbundsysteme

5.7.1. Multisysteme

In den bisherigen Ausführungen wurden Mikrorechner mit nur einem Mikroprozessor, sogenannte Monosysteme, vorausgesetzt. In der Praxis sind jedoch häufig in einem Mikrorechner mehrere Mikropro-

zessoren miteinander zu einem Multiprozessor- oder Mehrprozessorsystem verbunden. Eine weitere Möglichkeit besteht darin, mehrere Mikrorechner (mit einem oder mehreren Mikroprozessoren) gemeinsam als ein *Multimikrorechner-* oder *Mehrrechnersystem* zu betreiben. Derartige Verbundsysteme werden eingesetzt, wenn Forderungen nach Rechnersystemen mit z.B. höherer Leistungsfähigkeit und Rechengeschwindigkeit (Lastverbund), größerem Funktionsumfang (Funktionsverbund), größerer Zuverlässigkeit (Sicherheitsverbund) oder räumlich gegliedertem Zugriff zum Rechnersystem (verteilte Systeme) bestehen.

Verbundsysteme sind besonders bedeutungsvoll geworden, seitdem es möglich ist, hochintegrierte Mikroprozessor- und Mikrorechner-Schaltkreise in sehr großen Stückzahlen sowie mit stark sinkendem ökonomischem Aufwand herzustellen. Jeder Schaltkreis hat zwar für sich genommen nur eine relativ kleine Leistungsfähigkeit. Werden jedoch mehrere zu größeren, komplexen Systemen verbunden, erreicht man je nach Anzahl der Einzelkomponenten und gewünschtem Ausrüstungsgrad eine optimale Leistungsfähigkeit sowie gleichzeitig eine ökonomisch günstige Lösung für das Gesamtsystem. Zentrales Problem in Verbundsystemen ist jedoch die Koordinierung und Organisation der *Parallelarbeit* und die abgestimmte Funktion der Einzelkomponenten. Diese Aufgaben sind relativ aufwendig, sie binden häufig einen beträchtlichen Teil der Rechenleistung eines Multisystems.

5.7.2. Multiprozessorsysteme

In einem Multimikroprozessorsystem sind mehrere (2 bis über 1000) Mikroprozessoren über ein BUS-System miteinander verbunden. Sie benutzen gemeinsam den Hauptspeicher sowie die Ein- und Ausgabeeinheiten und arbeiten auch zusammen an einem oder mehreren Programmen (Bild 53). Das Kennzeichen eines Multiprozessorsystems ist die starke system- und programmtechnische Kopplung sowie Verflechtung der Einzelkomponenten des Multisystems. Dadurch lassen sich bestimmte Programme (parallelisierbare Programme) äußerst schnell und effektiv bearbeiten. Bei einer größeren Anzahl von im Verbund betriebenen Mikroprozessoren ist jedoch ein erhöhter Aufwand zur Koordinierung des Benutzens der gemeinsamen Einheiten des Rechnersystems notwendig. Der durch den Zusammenschluß der Mikroprozessoren gegebene Leistungszuwachs wird dann z.T. durch «innere Organisationsarbeiten» wieder zunichte gemacht.

Als Sonderform des Multiprozessorsystems gibt es den *Feldrechner*. In ihm sind mehrere gleiche Prozessoren matrixartig angeordnet und starr miteinander verkoppelt (Bild 54).

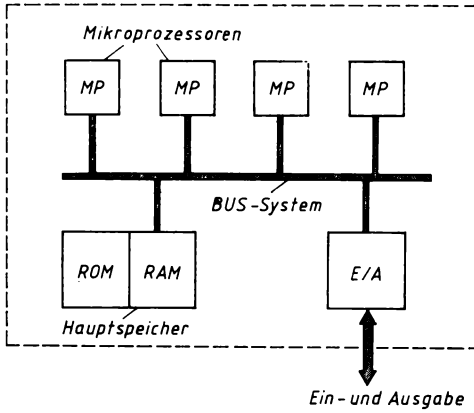


Bild 53. Struktur eines Multiprozessorsystems

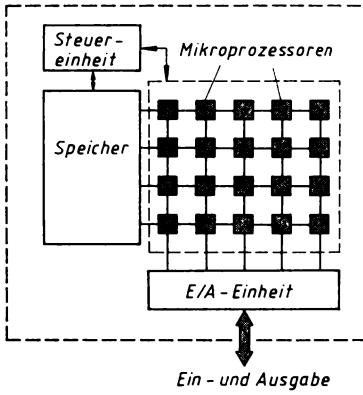


Bild 54. Schematischer Aufbau eines Feldrechners

5.7.3. Multirechnersysteme

Bei einem Multimikrorechnersystem sind mehrere Mikrorechner über ein geeignetes Verbundsystem miteinander verknüpft (Bild 55). Jeder einzelne enthält seine eigenen (privaten) Betriebsmittel, wie Mikroprozessor, Hauptspeicher, Ein-/Ausgabeeinheiten, allerdings können auch gemeinsam verwendete Komponenten, z.B. externe Speicher, vorliegen. Das Verbundsystem eines Mehrrechnersystems kann sehr unterschiedliche Strukturen aufweisen, etwa Stern-, Ring-, Baum-

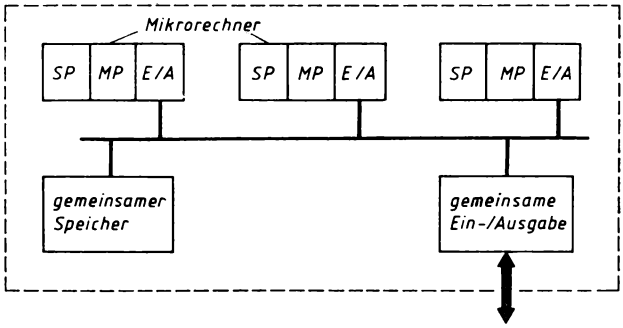


Bild 55. Struktur eines Multirechnersystems (Sp Speicher, MP Mikroprozessor, E/A Ein- und Ausgabeeinheit)

oder Maschenstruktur (Bild 56). Am meisten wird jedoch die BUS- oder Baum-Verbindungsstruktur verwendet.

Bei einem Multirechnersystem ist der Kopplungsgrad zwischen den Mikrorechnern weitaus geringer als zwischen den Mikroprozessoren eines Multiprozessorsystems. Jeder Mikrorechner arbeitet in bestimmten Zeitabschnitten weitgehend selbständig und verfügt über

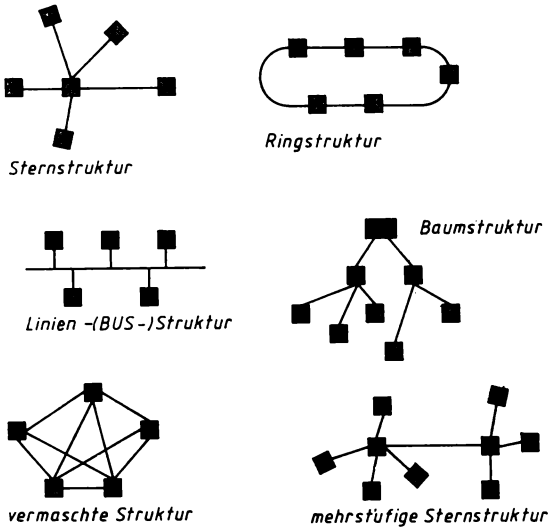


Bild 56. Verbundstrukturen bei Multisystemen

ein «eigenes» Programm. Es gibt aber auch Fälle, wo gemeinsame Programme bearbeitet werden. Oft haben Multirechnersysteme die Eigenschaft, daß die einzelnen Rechner auf bestimmte Aufgaben «spezialisiert» sind. Eine sinnvolle Aufgabenteilung wird dann meist von einem Leitrechner beim Bearbeiten der Programme vorgenommen.

5.7.4. Mikrorechnernetze

Sie sind dadurch gekennzeichnet, daß mehrere Mikrorechner an einem *Kommunikationssystem* angekoppelt sind. Dieses bietet sogenannte Kommunikations-«Dienste», die die Mikrorechner für die Herstellung von Kommunikationsverbindungen in Anspruch nehmen. In der Regel kann in einem Mikrorechnernetz jeder Rechner mit jedem kommunizieren. Ein solches Netz kann sich über kleinere räumliche Distanzen erstrecken, z.B. in einer Maschine, über eine Werkhalle oder ein Gebäude (lokale Netze). Es lassen sich aber auch größere Entfernungen, meist unter Zwischenschaltung nachrichtentechnischer Einrichtungen, überbrücken, beispielsweise in Netzen über ein räumlich ausgedehntes Gebiet einer Stadt (territoriale Netze).

In einem Mikrorechnernetz übt das Kommunikationssystem die zentrale Funktion aus (Bild 57). Es enthält Kommunikationsknoten (Netzknoten) sowie Kommunikationswege (Übertragungskanäle) und kann ähnliche unterschiedliche Verbindungsstrukturen (Topologie) wie in den Mehrrechnersystemen aufweisen. Besondere Bedeutung hat auch hier die Stern-, Ring-, Baum- und BUS-Struktur. Im allgemeinen werden die genannten Knoten ebenfalls mit Hilfe von Mikrorechnern realisiert. Als Kommunikationswege dienen elektrische Lei-

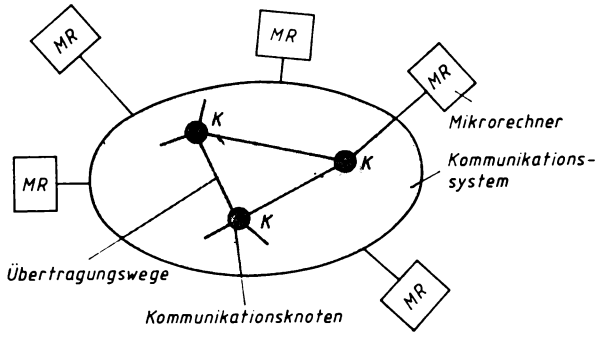


Bild 57. Schematischer Aufbau eines Mikrorechnernetzes

tungen oder optoelektronische Verbindungen (Glasfaser-Leitungen). Da das Kommunikationssystem auch dazu benutzt wird, den Nutzerzugriff zum Rechnersystem zu erleichtern, erfolgt oft eine logische Trennung des gesamten Mikrorechnernetzes in Kommunikationssystem, Zugriff- und Verarbeitungssysteme (Bild 58).

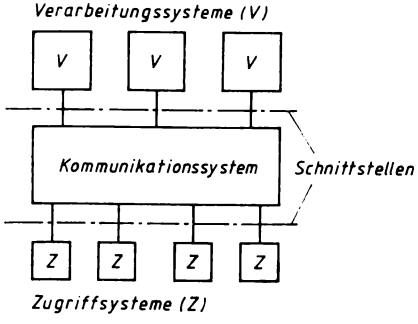


Bild 58. Trennung eines Mehrrechnerverbundsystems in Kommunikationssystem, Verarbeitungs- und Zugriffsysteme

6. Programmierung von Mikrorechnern

6.1. Abgrenzung Hard- und Software

Es wurde bereits betont, daß dem Einsatz eines Mikrorechners das Programmieren als ein wichtiger Arbeitsprozeß vorausgeht. Alle dabei entstehenden Produkte, wie Programme, Programmsysteme, die programmtechnischen Mittel zur Unterstützung und Verbesserung dieser Arbeiten sowie die entsprechenden Dokumentationen werden mit dem Begriff *Software* umfaßt. Im Unterschied dazu bezeichnet die *Hardware* die Gesamtheit aller technischen und elektronischen Elemente, Systeme und Einrichtungen eines Rechners. Eine absolute Grenze zwischen Hard- und Software kann jedoch bei Mikrorechnern nicht gezogen werden. Sie verschiebt sich vielmehr von Anwendungsfall zu Anwendungsfall. Im Laufe der technischen Weiterentwicklung sind zudem wesentliche Veränderungen festzustellen. Heute ist es möglich, für bestimmte Hardwaresysteme auch ökonomisch günstigere Softwarelösungen bereitzustellen. Andererseits tritt häufig der Fall auf, daß bestimmte Softwaresysteme in Form von leistungsfähigen Hardwarelösungen, z.B. als fertig programmierte hochintegrierte Schaltkreise (sogenannte «Software in Silicium») zur Verfügung stehen. Allgemein gilt der Grundsatz: «Was heute noch Software ist, kann morgen bereits Hardware sein – und umgekehrt.»

6.2. Das Programm

Ein Mikrorechner ist seiner Anlage nach zur Lösung zahlreicher Problemtypen befähigt. Allerdings muß dem Mikrorechner zunächst die Arbeitsvorschrift, das Programm eingegeben werden, mit dem ihm das spezielle Lösungsverfahren oder ein entsprechendes Reaktionsverhalten vorgeschrieben wird. Ein Programm ist die maschinell verarbeitbare Darstellungsform eines bestimmten Algorithmus. Dieser wiederum wird durch die entsprechende jeweilige Aufgabenstel-

lung bestimmt und ist im Prozeß der Programmentwicklung zu entwerfen.

Die Vielfalt der Programme – hinsichtlich Darstellungsart, Leistungsumfang und Einsatzzweck – ist für die Mikrorechentechnik charakteristisch. Das Spektrum reicht von kleinen wissenschaftlich-technischen Aufgaben und einfachen Steuerprogrammen bis zu komplexen Softwaresystemen mit höchsten Sicherheitsanforderungen.

6.3. Merkmale der Programmierung

Bis heute kann ein Digitalrechner, einschließlich aller Mikrorechner, mangels der Fähigkeit zu lernen und schöpferisch zu reagieren, nicht selbsttätig sein Programm aufstellen und abarbeiten. Er kann keine brauchbare Arbeit durchführen, wenn er nicht mit einer vom Menschen geschaffenen Arbeitsvorschrift versehen ist. Daher ist es von jeher für die Entwicklung von Rechnern besonders wichtig und vorrangig anzustreben, daß sie problemlos für die verschiedensten Aufgabenstellungen programmiert werden können.

Programmieren heißt – einfach ausgedrückt – die Arbeitsanweisungen für den Mikrorechner aufstellen. In Wirklichkeit gehört jedoch hierzu, wie sich noch zeigen wird, viel mehr. Im wesentlichen ist es die Übertragung oder Umsetzung eines technischen, ökonomischen oder mathematischen Sachverhalts in die «Sprache» und «Welt» des Rechners. Dies ist jedoch einfacher gesagt als getan. Alle heutigen Rechner können, wie bereits bemerkt, nicht schöpferisch reagieren. Sie sind zudem (noch) nicht in der Lage, mit Anweisungen – man sagt auch Befehle oder Instruktionen – direkt zu arbeiten, wenn diese der normalen sprachlichen, technischen oder mathematischen Aussageform entsprechen. Hierfür sind erst noch (unter Umständen recht komplizierte) Umformungen in der «Art» des Beschreibens der Problemstellung nötig.

Um mit dem Rechner eine Aufgabe zu lösen, muß diese in einer besonderen Sprache, der Computersprache, formuliert werden. Diese künstliche Sprache, die *Programmiersprache*, kann in ihren Beschreibungsmöglichkeiten sowie in ihrer Leistungsfähigkeit sehr unterschiedlich sein. Grundlage dafür ist, daß sich jeder rechentechnische Vorgang in eine Folge von Grundoperationen auflösen läßt, den «Wörtern» und «Sätzen» der Programmiersprache. Auf diese Weise wird es möglich, die Problemstellung rechentechnisch zu «formulieren» und damit als Aufgabe dem Rechner zu stellen.

Die allerersten Digitalrechner wurden in der sogenannten Maschinensprache programmiert. Wie sich noch zeigen wird, ist dies im Vergleich zu anderen Programmiersprachen jedoch sehr umständlich und unhandlich, da der Programmierer direkt im Bereich der binären Steuerworte des Rechners «denken» muß und alle Daten in binärer Form darzustellen sind. Für die ersten Digitalrechner gab es überhaupt nur diese Möglichkeit der Programmierung. Aber bereits damals gab es erste Gedanken und theoretische Untersuchungen, wie eine «wirkliche» Programmiersprache aussehen müßte. Anfang der 50er Jahre begann die systematische Entwicklung durch eine kleine Gruppe von Wissenschaftlern. Das Ziel war zunächst ein Vereinfachen von mathematischen Formeln und Algorithmen. Es folgten dann Jahre, in denen die Programmiersprachen einen gewaltigen Aufschwung nahmen und Grundlagen für eine eigene Fachdisziplin entstanden. Heute hat dieses Fachgebiet längst einen festen Platz im Gebäude der rechentechnischen Wissenschaften sowie ein hohes Niveau in Theorie und Praxis erreicht. Eine stetige Weiterentwicklung zu noch leistungsfähigeren und verbesserten Sprachen ist dennoch unverkennbar.

Außer Programmiersprachen waren jedoch noch weitere Voraussetzungen für eine effektivere Programmierung zu entwickeln. Ist nämlich eine Aufgabe formuliert, so ist der Rechner in der Regel damit noch nicht arbeitsfähig. Erst nach bestimmten Sprachübersetzungen, entsprechenden Programmkorrekturen, Testläufen, Umformungen und dem Transport des Programms in den Speicher kann er diese Aufgabe vorschriftsmäßig erledigen. Alle diese vorbereitenden Arbeiten wurden ebenfalls dem Rechner übertragen und so angelegt, daß sie automatisch, nur unter Beobachtung des Menschen ablaufen. Zu diesem Zweck entstanden zusätzliche umfangreiche Softwarekomponenten, «Programmiersysteme», die die Programmierzeit auf Bruchteile des früheren Aufwands verminderten. Auch war in der Folgezeit die Effektivität des Betriebs von Rechnern zu verbessern. Es wurden sogenannte Betriebssysteme entwickelt. Das sind Programme, die den inneren Ablauf sowie die Abarbeitung der Anwenderprogramme im Rechner organisieren. Alle diese Mittel, die man dem Nutzer in die Hand gab, bewirkten nicht nur eine bessere und leistungsfähigere Programmierung, sondern verschafften der Rechentechnik auch Zugang zu Bereichen, in denen sie vordem, aus Gründen des Aufwands und der Kompliziertheit der Probleme, nicht angewendet werden konnte.

Mit der weiteren Entwicklung der Rechentechnik und der Zunahme des Umfangs der Informationsverarbeitungsprozesse wurden die An-

wendungsprogramme sowie die für Programmierung und Rechnerbetrieb notwendigen Softwaresysteme immer umfassender, komplexer und damit auch immer unübersichtlicher. Die Folge war, daß sich unerkannte Fehler in Anwender- und Systemprogrammen einschlichen. Die Programmierarbeiten ließen sich immer schwieriger planen und die Zuverlässigkeit der Software verringerte sich insgesamt. Auf diese Weise stiegen die Aufwendungen für die Herstellung der Software stetig, obwohl im Laufe der weiteren Entwicklung die Hardware an Leistungsfähigkeit gewann und leichter zu programmieren war. So entstand der Begriff der «Softwarekrise». Er sollte ausdrükken, daß in der Softwareherstellung eine Ausnahmesituation eingetreten war. Mit der weiteren Entwicklung der Mikroelektronik wurde die Produktion der Hardware noch ökonomischer und das Verhältnis zwischen den Aufwendungen der Hard- und Software verschob sich weiter zuungunsten der Software. Diese Entwicklung hält auch heute noch an.

Die Hauptursachen der genannten Schwierigkeiten liegen in der Besonderheit der Software begründet. Mit Hilfe von Programmen ist es prinzipiell möglich, beliebig komplizierte Vorgänge abzubilden oder sehr komplexe logische Systeme zu entwerfen. Beispielsweise kann bereits ein einzelner Mensch so verwickelte und umfangreiche Softwaresysteme aufbauen, daß er sie – wenn er nicht besondere Maßnahmen trifft – nach einiger Zeit nicht mehr zu überblicken vermag und ihm das Programm «über den Kopf wächst». Bereits nach einigen Wochen oder Monaten der Programmiertätigkeit kennt er Einzelheiten nicht mehr, die er zu Beginn der Arbeiten selbst programmiert hat. Die wirkliche Ursache der Krise liegt in den Methoden des Entwurfs komplexer Systeme. Um diesen Schwierigkeiten und Problemen bei der Programmentwicklung zu begegnen, hat *E. W. Dijkstra* bereits in den 60er Jahren folgende Forderungen gestellt:

- Methodische Durchdringung der Software hinsichtlich ihrer Funktion und Wirkungsweise sowie ingenieurmäßiger Umgang mit ihr sowie
- Auswahl von angemessenen und übersichtlichen Programmstrukturen sowie bewußte Einschränkung der «Strukturierungsmittel».

Diese Gedanken führten in der Folgezeit zu einem grundsätzlich neuen Herangehen in der Softwareherstellung (z.B. Strukturierte Programmierung) und beeinflußten auch die Entwicklung neuer, geeigneter Programmiersprachen. Sie waren die Basis der heute vorhandenen und sich stetig weiterentwickelnden Methoden der Softwareherstellung.

6.4. Programmiermethodik

6.4.1. Der industrielle Herstellungsprozeß eines Programms

In den weiteren Abschnitten wird der Weg eines Programms von der Aufgabenstellung bis zu seiner Abarbeitung auf dem Mikrorechner im Überblick verfolgt. In diesem Zusammenhang sollen auch die bei den entsprechenden Prozessen anzutreffenden wichtigsten Begriffe der Software eingeführt und in das Gesamtbild eingeordnet werden.

Die Arbeit eines *Programmentwicklers*, also jenes Menschen, der Rechner-Programme entwirft, testet, korrigiert, verbessert und dokumentiert, läuft unter recht verschiedenartigen betrieblichen Bedingungen ab. Besonderen Einfluß haben die Komplexität der Aufgabenstellung, materielle Bedingungen und Voraussetzungen, Mittel und «Werkzeuge», die zur Verfügung stehen sowie deren Qualität und technischer Stand. Diese Einflußfaktoren sind in der Praxis der Programmierung herkömmlicher Rechner und Datenverarbeitungsanlagen von Fall zu Fall sehr differenziert, in der Mikrorechentechnik aber noch weit unterschiedlicher. Ein wesentlicher Grund dafür ist zum einen die große Breite der Einsatzmöglichkeiten der Mikrorechentechnik und zum zweiten die Ökonomie, die bestimmte Bedingungen erzwingt. Beispielsweise wird eine Einrichtung, die nur einmal im Jahr ein kleineres Mikrorechner-Steuerprogramm entwickelt, mit viel «bescheideneren» Programmiermöglichkeiten aufwarten, als ein Betrieb, der sehr komplizierte Programme entwirft oder/und auf Grund seiner Erzeugnispalette tagtäglich solche Arbeiten durchzuführen hat. Im letzteren Fall lohnt sich auch die Anschaffung aufwendiger und hochproduktiver Einrichtungen zur Programmentwicklung.

In der heutigen Zeit ist man zur Erkenntnis gelangt, daß ein Programm, ein Softwareprodukt, in ähnlicher Weise zu «produzieren» ist wie ein Erzeugnis der materiellen Produktion. Die hohen Aufwendungen und Mittel, die insgesamt zur Softwareherstellung notwendig sind, zwingen einfach dazu, solche organisatorischen Formen einzusetzen. Bei näherer Betrachtung stellt sich jedoch heraus, daß bei der Softwareherstellung – im Unterschied zum materiellen Produktionsprozeß – die größten Aufwendungen nicht in der Phase der Produktion selbst liegen, also in der «Vervielfältigung» des Erzeugnisses, sondern in der Programm-«Entwicklung», der Programm-«Konstruktion». Dies sind Etappen, die in der materiellen Produktion mit der Entwicklung, Projektierung oder Überleitung eines Erzeugnisses vergleichbar sind und dort zum Teil ähnliche Schwierigkeiten bereiten.

Eine weitere Besonderheit liegt darin, daß bei der Produktion von Programmen ökonomische Gesetzmäßigkeiten herrschen, die denen bei der Herstellung eines Buches oder eines integrierten Schaltkreises vergleichbar sind. Ist nämlich der aufwendige Weg bis zu den Druckformen eines Buches oder bis zum Layout und zu den Masken eines Schaltkreises zurückgelegt, so sind die bei einer Massenfertigung noch für die Herstellung eines einzelnen «Exemplars» erforderlichen Aufwendungen, im Vergleich zu den bisherigen, vernachlässigbar klein. Ein Programm kann ebenfalls, wenn es erst einmal vorliegt, sehr einfach vervielfältigt und in einer beliebig großen Zahl von Rechnern eingesetzt werden.

Es ist also sinnvoll, die in der materiellen Produktion bewährten Prinzipien zur effektiven Gestaltung des Herstellungsprozesses zu Rate zu ziehen. Dort gilt folgender Grundsatz: In einer ersten Etappe wird das Erzeugnis in seiner Gesamtheit sowie in seinen Einzelheiten entworfen, entwickelt und konstruiert (Bereich angewandte Forschung und Entwicklung); anschließend (gegebenenfalls auch zeitlich verzahnt) wird eine geeignete «Technologie» für den Prozeß der Herstellung geschaffen oder, falls diese bereits vorhanden ist, darauf zurückgegriffen. Der erste Schritt entspricht der Beantwortung der Frage «WAS ist herzustellen?», während die Technologie die Antwort auf «WIE ist es herzustellen?» gibt. Auch in der Softwareproduktion wird eine solche Zweiteilung vorgenommen.

6.4.2. Beschreibung und Struktur eines Softwaresystems

Ein Programm läßt sich als System, als Softwaresystem, mit den Mitteln und Methoden der Systemtheorie darstellen und beschreiben. Wie bereits ausgeführt, setzt sich ein System aus Elementen zusammen, die durch bestimmte «Einzel»-Eigenschaften und -Merkmale gekennzeichnet sind. In Softwaresystemen sind das Bytes, Wörter, Befehle, komplexe Anweisungen, Programmteile, logische Funktionen, Einzeldaten und Datenfelder. Die Gesamtheit aller dieser Elemente sowie die zwischen ihnen bestehenden Beziehungen bestimmen Struktur und Eigenschaften eines Softwaresystems. Es umfaßt in der Regel zwei wichtige Bestandteile: Verarbeitungsfolgen sowie zu bearbeitende Größen, die Daten. Beide bestehen aus einer Vielzahl der genannten elementaren Bausteine. Alle in einem Programmsystem verwendeten Verarbeitungsbausteine und ihre Verknüpfungen bezeichnet man als *Verarbeitungsstruktur*, alle Datenbausteine und ihre Verknüpfungen als *Datenstruktur*.

Der Umfang eines Systems wird im allgemeinen durch seine Grenzen markiert. Jedes arbeitsfähige System tauscht über diese mit der

Umwelt irgendwelche Informationen aus. Bei Softwaresystemen wird die Grenzziehung durch Einführung sogenannter *Softwareschnittstellen* vorgenommen. Sie ermöglichen den Zugriff des Nutzers auf ein solches System oder die Kopplung verschiedener Softwaresysteme untereinander. Eine solche Schnittstelle beschreibt Art und Eigenschaften der Ein- und Ausgabegrößen des Systems sowie die Regeln, nach denen der Austausch von Informationen zwischen System und Umwelt zu erfolgen hat. Außerdem wird damit gleichzeitig der Funktionsumfang, also der Zweck des Programms, festgelegt.

In der Praxis sind noch weitere Eigenschaften des Softwaresystems sowie gewisse Regeln für dessen Gestaltung von Bedeutung. Sie alle werden durch die *Architektur* des Programms bzw. des Softwaresystems gekennzeichnet. Sie umfaßt:

- Darstellung, systemtheoretische Beschreibung und Bewertung eines Softwaremodells;
- Kriterien für übersichtliche und zweckmäßige Gestaltung des Softwareprodukts;
- Kriterien zur geeigneten Einordnung (Integration) in die «Umwelt» des Systems;
- Aufzeigen der Grundprinzipien für Aufbau und Realisierung des Softwaresystems («Konstruktionsprinzipien»).

Je nachdem, wie diese Kriterien vom Programmentwickler erfüllt werden, erhält man Programme mit guten oder weniger guten Architektureigenschaften. Günstige Gestaltungsregeln und Entwurfsprinzipien reduzieren den Aufwand bei der späteren Realisierung, und zweckmäßige Regeln für die Integration in eine Systemumgebung ermöglichen eine flexible und breite Anwendung des Programms.

6.4.3. Technologie der Programmierung

Unter Technologie der Programmentwicklung soll die effektive Gestaltung des Herstellungsprozesses eines Softwareprodukts unter industriellen Bedingungen verstanden werden. Das Ziel besteht darin, die Programme in guter Qualität, hoher Zuverlässigkeit und unter möglichst geringem personellem und materiellem Aufwand herzustellen. Für die Softwareproduktion gilt insbesondere, daß eine Technologie – wenn sie sich für einen bestimmten Aufgabenbereich einmal bewährt hat – wiederholt anwendbar sein sollte. Dies bewirkt, daß die Menschen, die in der Programmentwicklung und deren Umfeld tätig sind, sich mehr und mehr eine Technologie «zu eigen machen», Routine erlangen und auf diese Weise die Effektivität der Softwareproduktion stetig steigt.

Eine Softwaretechnologie ist durch ihre Methode, Werkzeuge und Organisationsformen gekennzeichnet.

Die *Methode* zeigt den grundsätzlich zu beschreitenden Weg auf und berücksichtigt dabei die zur Verfügung stehenden Theorien, technische Verfahren sowie Werkzeuge.

Die *Werkzeuge* umfassen manuelle und maschinelle Hilfsmittel, die dem Programmentwickler bei seiner Arbeit zur Verfügung stehen. Dies sind zum einen Instrumente für Darstellung und Kommunikation, wie Zeichnungen, Ablaufplan, sprachliche Texte, Programmiersprachen, und zum anderen technische Hilfsmittel für ablaufende Prozesse, wie Programmiersysteme, Programmierarbeitsplätze, Testsysteme, Dokumentations- und Informationssysteme, deren Grundlage in jedem Fall ein Rechner ist. Die *Organisationsformen* schließlich geben den Rahmen an, in dem die Methode mit den entsprechenden Hilfsmitteln unter Berücksichtigung der betrieblichen Möglichkeiten und Besonderheiten wirksam wird.

6.4.4. Strukturierte Programmierung

Als günstigste und effektivste Softwaretechnologie hat sich bisher die Strukturierte Programmierung – auch als «Systematische Programmierung» bezeichnet – erwiesen [21]. Sie ist durch folgende Merkmale charakterisiert:

- Strukturierung der Softwareherstellung in definierte Arbeitsetappen mit vorgegebenem Inhalt und Leistungsumfang;
- wiederholte funktionelle Zergliederung und strukturelle Verfeinerung des Informationsverarbeitungs-Algorithmus und des entsprechenden Programms in sich abgeschlossener Bausteine;
- Verwendung weniger, übersichtlicher (standardisierter) Algorithmen- und Funktionsbausteine.

Strukturierung und funktionelle Verfeinerung bewirken, daß sogenannte wohl-strukturierte Programme entstehen, die übersichtlich aufgebaut sowie gut dokumentierbar sind, und daß bereits in der Herstellungsphase eine weitgehende Fehlerfreiheit der Programme erreicht wird. Außerdem werden Änderungen oder Erweiterungen der Programme sowie die Fehlersuche wesentlich erleichtert, wie überhaupt bei allen Arbeiten ein «systematisches» Vorgehen gewährleistet ist. Die Verwendung nur weniger Algorithmen- und Funktionselemente bedeutet zwar eine Einschränkung der Vielfalt der Möglichkeiten, bringt aber einen entscheidenden Gewinn in der Überschaubarkeit der Programme.

Arbeitsetappen
In der Programmentwicklung werden zweckmäßig folgende Schritte gewählt:
- *Spezifizierung* (Aufgabenstellung des Programms);
- *Entwurf* (Programmentwurf, Verfeinerung);
- *Implementierung* (Codierung in Programmiersprache);
- *Verifizierung* (Programmtest, Fehler- und Mängelanalyse);
- *Integration* (Einführung in bestehendes System, Nutzung).

In jeder Arbeitsetappe entstehen entsprechende Programmdokumentationen, die am Schluß zusammenzufassen sind. Die einzelnen Schritte werden bei der Erarbeitung von kleineren, überschaubaren Programmen bzw. Teilprogrammen in zeitlicher Folge durchlaufen. Sind jedoch größere Softwareprodukte in kollektiver Arbeit herzustellen, wird zweckmäßig erst eine Zerlegung in Teilaufgaben vorgenommen, von denen dann jede entsprechend dem Schema, gegebenenfalls auch zeitlich nebeneinander, zu bearbeiten ist.

Aufgabenstellung:
Eine Programmentwicklung beginnt in der Regel mit Untersuchungen am bestehenden Problem (Problemanalyse) und der «Spezifizierung», d.h. der Festlegung der gewünschten Lösung. Mit ihr liegt eine möglichst klare und vollständige Beschreibung des zu erarbeitenden Programms vor. Die wesentlichsten Eigenschaften und Funktionen sind aufgezeigt sowie die Anforderungen an das zukünftige Softwareprodukt aus der Sicht des Nutzers niedergelegt. Vergleichbar ist die Spezifizierung mit dem Pflichtenheft eines Erzeugnisses aus dem Bereich der materiellen Produktion. In beiden Fällen werden außer technischen Parametern und Kenngrößen auch Zuverlässigkeit, Qualität sowie ökonomischer Aufwand festgehalten. Bei einer geplanten Programmentwicklung spricht man häufig darüber, *was* das zukünftige Programm zu leisten hat und *wie* es zu realisieren ist. In der Spezifikation geht es demnach um die Beschreibung des «WAS».

Einer sorgfältigen Aufgabenstellung kommt eine besondere Bedeutung zu. Ist das geplante Programm nur unzulänglich festgelegt, wird auch das fertige Produkt unzulänglich sein, unabhängig davon, wie «gut» die anderen Arbeitsschritte der Softwareentwicklung ausgeführt werden. Internationale Untersuchungen zeigen, daß die in den nachfolgenden Etappen entstehenden Fehler bis zu einem Drittel ihre Ursache in einer unzureichenden Aufgabenstellung haben.

Programmentwurf:
Die nächste Etappe in der Softwareentwicklung ist der Entwurf des Programms in mehreren Verfeinerungsschritten. Der Entwurf ist ein

Realisierungsplan der Software, eine Beschreibung des «WIE». In ihm wird die Problemlösung in ihren wichtigsten Zügen aufgezeigt. Bevor man jedoch an den Entwurf geht, sind die Bedingungen auszuwählen oder zu ergründen, unter denen das Programm arbeiten soll; also Mikrorechnertyp, mögliche Betriebsarten des Rechners, Größe der Speicher, Art der Ein- und Ausgabegeräte sowie Art, Umfang und Leistung bereits vorhandener und eventuell mit benutzbarer Programme bzw. Programmsysteme. Alle Punkte beeinflussen den Programmentwurf und das geplante Ergebnis.

Zunächst wird ein erster, noch relativ grober Algorithmus für die Lösung der Aufgabe entworfen. Ein *Algorithmus* ist die eindeutige und formalisiert dargestellte Vorschrift zur schrittweisen Lösung der Aufgabe, in unserem Fall der Informationsverarbeitungsaufgabe [20, 22]. Die Aufstellung des «ersten» Algorithmus ist der schwierigste Teil des Entwurfs, da alle wesentlichen Funktionen des Programms bereits enthalten sein müssen. In den weiteren Teilschritten folgt dann eine Verfeinerung der Lösung, d.h. ein stufenweise immer größerer Detaillisierungsgrad. Die Hauptaufgaben des geplanten Programms werden in entsprechende Teilaufgaben überführt und bilden wieder in sich abgeschlossene Einheiten. Dieser Prozeß wird mehrfach wiederholt, bis ein Feinentwurf vorliegt. Hierbei ist man bestrebt, so spät wie möglich die Besonderheiten des Mikrorechners zu berücksichtigen, um eine möglichst allgemeingültige Darstellung und damit eine breite Verwendungsfähigkeit der Lösung zu sichern. In der industriellen Softwareentwicklung haben sich für die ersten drei Verfeinerungsschritte folgende Zerlegungen bewährt: Im *Systementwurf* wird die Gesamtstruktur- und -funktion behandelt, im *Programmentwurf* in sich abgeschlossene Programmbausteine und schließlich im *Modulentwurf* (Modul entspricht einem Teilprogramm) entsprechende Teilfunktionen. Die Verfeinerung wird dann weiter fortgesetzt, bis man letztlich zu algorithmischen Grundstrukturen oder Elementarbausteinen kommt, die sich nicht weiter zerlegen lassen. Sind nur «kleine» Programme zu realisieren, kann man auch mit weniger Verfeinerungsschritten auskommen. Hauptproblem und «schöpferischer Akt» der einzelnen Entwurfsschritte sind die Formulierung der gewünschten Funktionen bzw. Aufgaben in geeignete Abläufe, die sich maschinell abarbeiten lassen, also das Aufstellen der Algorithmen und Teilalgorithmen. Theoretische Untersuchungen zu dieser Problematik haben aber gezeigt, daß alle überhaupt denkbaren Algorithmen aus nur wenigen Grundstrukturen, sogenannten Algorithmenbausteinen, realisierbar sind. Indem man sich bewußt auf die Verwendung nur der notwendigsten Grundelemente beschränkt und auch die Ge-

staltungsfreiheiten der Algorithmen einengt, ist es möglich, die Algorithmierung zu systematisieren sowie die Überschaubarkeit der daraus hervorgehenden Programme zu erhöhen. Diese Erkenntnisse werden auch von modernen Entwurfsverfahren berücksichtigt. Folgende algorithmische Grundstrukturen haben besondere Bedeutung:

- *Sequenz* (Reihenfolge von Verarbeitungsschritten);
- *Selektion* (Entscheidung; Auswahl von Verarbeitungsschritten);
- *Zyklus* (Schleife; Wiederholung von Verarbeitungsschritten).

Mit Hilfe dieser Grundstrukturen kann man die Verarbeitungsbausteine miteinander zu algorithmischen Abläufen verknüpfen (Bild 59). Ein Baustein symbolisiert dabei einen in sich abgeschlossenen algorithmischen Verarbeitungsvorgang oder -schritt. Ein solcher Baustein enthält nur einen Eingang und Ausgang (Bild 59a, 59c). Zur grafischen Darstellung werden häufig Flußablaufdiagramme (Bild 59b) oder Struktogramme (Bild 59d) benutzt. Letztere unterstützen insbesondere die Strukturierte Programmierung.

Codierung:
In dieser Arbeitsetappe wird schließlich die im Entwurf endgültig verfeinerte Lösung mit den Mitteln einer Programmiersprache codiert und in eine maschinell verarbeitbare Form umgesetzt. Um auch in dieser Etappe ein vom speziellen Rechnertyp unabhängiges Programm sowie eine effektive Programmierung zu erreichen, sollte eine Programmiersprache ausgewählt werden, die mehr dem zu lösenden Problem und weniger dem Mikrorechner angepaßt ist. Auf diesen wichtigen Sachverhalt wird später noch eingegangen. Die Hilfsmittel und Werkzeuge in der Phase der Codierung sind außer der Programmiersprache grafische und tabellarische Darstellungen, wie Funktionspläne, Programmablaufpläne, Struktogramme sowie maschinelle Mittel, etwa Übersetzungs-, Test- und Korrektursysteme. Gewöhnlich stehen dafür in der Praxis Programmentwicklungssysteme oder Programmentwicklungsplätze (s. 6.8.) auf der Grundlage eines Rechners zur Verfügung.

Softwareprüfung:
Wie jedes Erzeugnis muß auch ein Programm bestimmte Forderungen hinsichtlich Qualität und Zuverlässigkeit erfüllen. Um diese in der Softwaretechnik nicht ganz einfachen Nachweise zu erbringen, sind Programmtests und Qualitätsprüfungen sowie (nach Abschluß einzelner Entwicklungsabschnitte) Fehler- und Mängelanalysen durchzuführen. Wichtigstes Ziel ist hierbei, die «Richtigkeit» eines Programms, also die vorschriftsmäßige Funktion gemäß der Aufgaben-

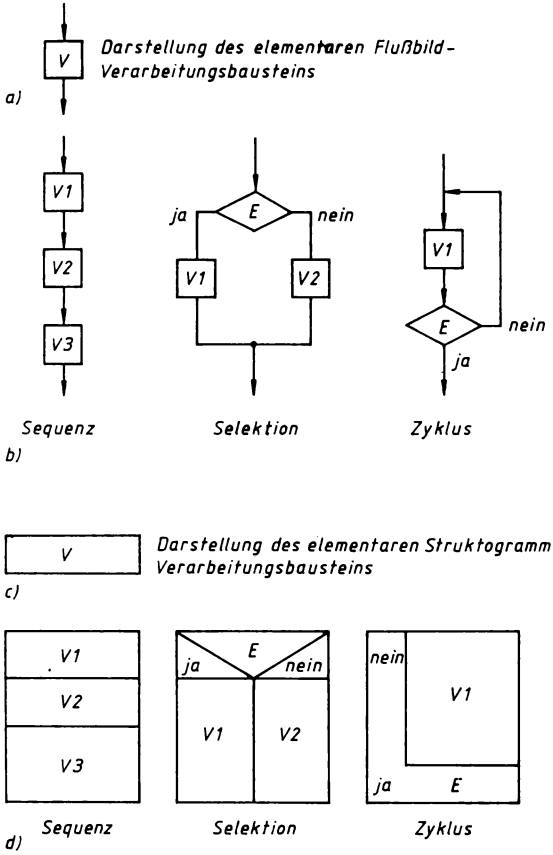


Bild 59. Algorithmische Grundstrukturen und ihre grafische Darstellung a) und b) Flußbilddarstellung; c) und d) Struktogrammdarstellung (V Verarbeitungsbaustein, E Entscheidungsbaustein)

stellung zu erreichen. Forschungsarbeiten auf diesem Gebiet haben allerdings gezeigt, daß der mathematisch strenge Beweis der Richtigkeit von Programmen in voller Allgemeinheit nicht zum Ziel führt. Man kann zwar feststellen, daß ein Programm fehlerhaft ist, niemals jedoch eine Fehlerfreiheit exakt nachweisen. Daher kommt den systematischen Untersuchungen, Testläufen und bestimmten Prüfstrategien besondere Bedeutung zu. Sie sorgen während der gesamten Entwicklungsphasen eines Programms als begleitender Prozeß dafür, daß

möglichst frühzeitig alle Fehler aufgedeckt und von Anfang an fehlerfreie Programme realisiert werden können. Allgemein gilt, je früher sich ein unerkannter Fehler in die Software einschleicht, um so größer sind am Ende die Auswirkungen und um so schwieriger ist seine Feststellung und Korrektur.

Einführung und Nutzung:
Wenn alle Prüfungen positiv verlaufen sind und das Programm den gestellten Anforderungen gerecht wird, erfolgt die Einführung in ein vorhandenes Mikrorechnersystem. Nach einem eventuell noch notwendigen Probebetrieb kann dann zur routinemäßigen Nutzung und gegebenenfalls zur Vervielfältigung des Programms übergegangen werden. In der Einführphase werden alle Teilprogramme zum Gesamtsystem zusammengefaßt und die Korrektheit als Ganzes in seiner hard- und softwaremäßigen Umgebung getestet. Häufig geht es dabei auch um Fragen der zweckmäßigen Eingliederung des fertigen Programms in ein bestehendes Softwaresystem. Dabei spielen Anpaßfähigkeit, Flexibilität sowie die geeignete Wahl der Übergangsstellen (Softwareschnittstellen) eine große Rolle.

Jedes Programm ist infolge der technischen Weiterentwicklung häufig Änderungen und Verbesserungen unterworfen. Hier wirkt sich die bereits in der Phase des Entwurfs eingeführte Strukturierung und klare Trennung in voneinander unabhängige Programmbausteine, d.h. die *geplante* leichte Änderungsmöglichkeit, vorteilhaft aus.

Dokumentation
Außer der Anweisungsfolge für den Rechner gehört zum Programm als industriellem Produkt noch eine entsprechende *Dokumentation.* Sie umfaßt die wichtigsten Informationen über Aufbau und Funktionsweise des Programms. Außerdem kennzeichnet eine Dokumentation den Leistungsumfang und enthält die «Betriebsvorschrift» sowie die «Bedienanleitung». Da ein Softwareprodukt in der Regel ebenso wie ein materielles Produkt gehandelt und vertrieben wird, sind derartige Dokumentationen unabdingbare Forderungen des Nutzers.

Softwarezyklus
Alle aufgeführten Arbeitsschritte werden in der Praxis meist nicht in der vorgestellten linearen Folge durchlaufen, sondern in einem sogenannten Softwarezyklus (Bild 60). Auf dem Weg von der Problembeschreibung bis zur maschinellen Lösung folgen – insbesondere bei größeren Aufgaben – ein verzweigter und mehrfach durchschrittener «Rückweg» von Kontrollen des halbfertigen Programms mit den in

der Spezifikation vorhandenen Forderungen sowie der Nachweis der Fehlerfreiheit der Ergebnisse aller Arbeitsetappen. Auf diese Weise gewinnt das Programm die entsprechende Reife und Qualität und erfüllt am Ende dann auch die Erwartungen und Wünsche, die an das zu entwickelnde Softwareprodukt gestellt werden.

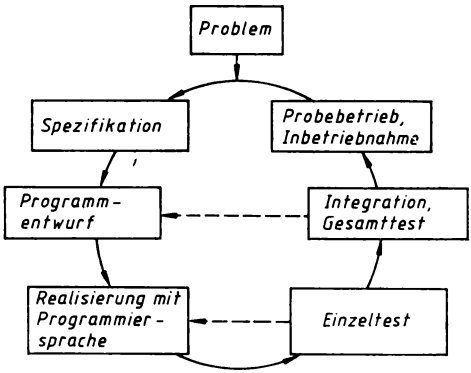


Bild 60. Softwarezyklus

6.5. Programmiersprachen

6.5.1. Kennzeichen einer Programmiersprache

Wesentlichen Einfluß auf die Art und Weise sowie Effektivität der Programmierung übt die verwendete Programmiersprache aus. Im Unterschied zu natürlichen Sprachen, die historisch in einem langen Entwicklungsprozeß und unter verschiedensten Einflußfaktoren entstanden sind, werden Programmiersprachen von Menschen künstlich geschaffen. Ihr Aufbau unterliegt strengen, wissenschaftlich begründeten Regeln. Künstliche Sprachen – auch als formale Sprachen bezeichnet – dienen zur Darstellung und Beschreibung von Vorgängen, Prozessen und Strukturen in vielen Fachgebieten, darunter in der Mathematik, Physik, Chemie. Die Programmiersprachen sind speziell entwickelt worden, um Daten und deren Verarbeitungsvorgänge in einem Digitalrechner symbolisch zu beschreiben.

Allen Sprachen ist gemeinsam, daß sie eine gegebene Menge von Grundzeichen – das *Alphabet* – enthalten, aus denen die sprachlichen Gebilde aufgebaut werden. Das Alphabet einer Programmiersprache enthält im allgemeinen neben den Buchstaben A, B, C, ..., Z auch

die Ziffern 0, 1, 2, ..., 9 sowie Sonderzeichen, z.B. Komma, Punkt, Klammern, Leerzeichen, usw., die alle als *Zeichen* oder *Symbole* benannt werden. Die sprachlichen «Gebilde», die man in einer natürlichen Sprache als «Wörter» und «Sätze» vorfindet, sind auch in einer Programmiersprache vorhanden. Allerdings haben sie, je nach Typ der Sprache, eine sehr verschiedenartige Gestalt und unterschiedliche Bezeichnungen. Die allgemeine Form eines programmiersprachlichen Gebildes ist die *Zeichenkette* oder *Symbolfolge*, d.h., die Aneinanderreihung gleicher oder verschiedenartiger Zeichen aus dem Alphabet der Programmiersprache. Erst die sinnvolle Kombination von Symbolen oder die Kombination von Symbolfolgen untereinander machen es möglich, bestimmte Bedeutungen in ein Sprachgebilde «hineinzulegen».

Die *Grammatik* einer Sprache legt fest, wie die «Wörter» und «Sätze» der Sprache zu konstruieren, welche Regeln dabei zu beachten sind. Diese werden in einer Programmiersprache auch als *Syntax* der Sprache bezeichnet. Für Programmiersprachen ist die Grammatik der Umgangssprache nicht geeignet, da sie zu unbestimmt und vieldeutig ist. Aus diesem Grund werden mathematisch formulierte, sogenannte *formale Grammatiken*, verwendet. Außer den syntaktischen Regeln ist jedoch auch Sinn und Bedeutung der einzelnen Symbolfolgen festzulegen. Dies wird mittels der *Semantik* der Programmiersprache beschrieben. Die Bedeutung der einzelnen sprachlichen Bestandteile widerspiegelt hierbei die im Rechner ablaufenden Aktionen. Beispielsweise könnte festgelegt sein, daß das Zeichen «+» eine Addition oder die Symbolfolge «MOVE» (engl. to move – transportieren) einen Datentransport im Rechner bewirken. Die Definition einer Programmiersprache erfolgt durch Angabe der Syntax *und* Semantik. Dies muß so geschehen, daß der Nutzer die Sprache möglichst leicht erlernen und eindeutig, d.h. ohne Hineinlegen verschiedener Bedeutungen, anwenden kann.

Ein wichtiges Kriterium der Qualität einer Programmiersprache für Mikrorechner ist zum einen die Leistungsfähigkeit, d.h. das Vermögen zur Beschreibung von Vorgängen und Daten, verbunden mit der effektiven Umsetzung auf den Rechner, und zum zweiten der Gesichtspunkt der Austauschbarkeit von Programmen zwischen verschiedenen Rechnertypen. Letzteres hat besondere Bedeutung. Sind beispielsweise Programme ohne wesentliche Änderungen von Rechner zu Rechner zu übertragen, wird teure Doppel- oder Mehrfachentwicklung vermieden und der Zugang zur Software aus anderen Quellen erleichtert. Aus diesem Grund wurden und werden auch internationale *Sprachstandards* erarbeitet und zur Anwendung empfohlen.

Bei Mikrorechnern unterscheidet man verschiedene Programmierniveaus. Sie sind abhängig von der Art der verwendeten Programmiersprache. Diese teilt man ein in:
- Maschinensprachen
- maschinenorientierte Sprachen
- höhere Programmiersprachen.

Jede Art der Sprachen hat ihre Vor- und Nachteile sowie spezielle Einsatzgebiete.

6.5.2. Maschinensprache

Dieser Begriff stammt aus der klassischen Rechentechnik und bedeutet soviel wie das Arbeiten auf dem direkten Niveau der Maschinenbefehle und der inneren Verarbeitungsgrößen des Rechners. Da es sich früher bei Digitalrechnern mit der Leistungsfähigkeit heutiger Mikrorechner wirklich um «Maschinen» mit großem Volumen und Energieverbrauch handelte, waren auch Bezeichnungen dieser Art gerechtfertigt. Heute sind die damit zusammenhängenden Begriffe zwar in der Mikrorechentechnik nicht mehr ganz zutreffend, sie werden jedoch weiter verwendet. Programmieren in Maschinensprache heißt demnach soviel wie Aufstellen eines Programms im Binärcode des Maschinenbefehlssatzes eines entsprechenden Mikroprozessors. Das Ergebnis ist ein Maschinenprogramm. Die elementaren sprachlichen Gebilde – Maschinenbefehle und Daten – bestehen hierbei aus Binärworten zu 1 byte oder einem Vielfachen davon und diese selbst nur aus den Symbolen 0 und 1 (Bild 61). Nachteile der Programmierung in Maschinensprache bestehen darin, daß ein Maschinenprogramm sehr aufwendig niederzuschreiben sowie unübersichtlich ist, daß leicht Programmierfehler entstehen und Korrekturen schwierig durchzuführen sind. Als Vorteil ist zu sehen, daß ein sofort abarbeitungsfähiges Programm in einem Schritt entsteht. Dies ist bei den anderen noch zu behandelnden Programmiersprachen nicht der Fall, hier muß erst noch ein «Übersetzen» des Programms eingeschoben werden. Um die Übersichtlichkeit eines Maschinenprogramms doch etwas zu erhöhen, werden die Befehle und die zu bearbeitenden Daten häufig in einer oktalen oder hexadezimalen Zahlendarstellung (Tabelle 9) wiedergegeben. Eine Programmierung in Maschinensprache wird in der Praxis nur für Testarbeiten unmittelbar an der Hardware, bei Reparaturarbeiten oder für kleinere Programme verwendet. Für die routinemäßige Programmierung ist diese Sprache ungeeignet. In den maschinenorientierten und höheren Programmiersprachen ist die Maschinensprache jedoch Zielsprache; d.h., Endergebnis der dort notwendigen Übersetzungsvorgänge ist ein Maschinenprogramm.

Speicheradresse	Befehl	Erläuterung	
⋮	⋮		
00010000	10010001	Subtraktion *C* von *A*	
00010001	01000111	Transport *A* nach *B*	(a)
00010010	10000010	Addition von *D* und *A*	
00010011	01001111	Transport *A* nach *C*	
00010100	11000101	*B*, *C* in Kellerspeicher	
00000101	11001101	Aufruf eines Unterprogramms	
00000110	00010000		
00000111	00010000		
00001000	01001111	Transport *A* nach *C*	
00001001			
⋮			

Speicheradresse	Befehl	Erläuterung	
⋮	⋮		
10	91		
11	67	wie a)	
12	82		
13	6F		
14	C5		(b)
15	CD		
16	10		
17	10		
18	6F		
19	⋮		
1A			
1B			

Bild 61. Ausschnitt aus einem Maschinenprogramm in binärer (a) und hexadezimaler (b) Darstellung

Tabelle 9. Codierung von Binärworten in Oktal- und Hexadezimalzahlen

Binär-Oktal-Code		Binär-Hexadezimal-Code	
000	0	0000	0
001	1	0001	1
010	2	0010	2
011	3	0011	3
100	4	0100	4
101	5	0101	5
110	6	0110	6
111	7	0111	7
		1000	8
		1001	9
		1010	A
		1011	B
		1100	C
		1101	D
		1110	E
		1111	F

Codierung für ein Byte

D_7D_6	$D_5D_4D_3$	$D_2D_1D_0$	$D_7D_6D_5D_4$	$D_3D_2D_1D_0$
1.	2.	3. Oktal-ziffer	1.	2. Hexadezimal-ziffer

6.5.3. Maschinenorientierte Programmiersprachen

Da die Maschinensprache für die praktische Programmierarbeit viel zu unbequem ist, hat man weitere Programmiersprachen entwickelt, in denen die Binärworte eine andere Form haben. Eine solche maschinennahe Sprache, auch als maschinenorientierte oder *Assembler*-Sprache (engl. to assemble – zusammensetzen) bezeichnet, bedient sich symbolischer Zeichen und Ausdrücke. Sowohl Maschinenbefehle als auch Speicheradressen und Verarbeitungsgrößen werden durch sinnvoll gewählte und leicht einprägsame Symbole oder umgangssprachliche Kurzformen (meist englische Abkürzungen) ausgedrückt (Bild 62). Die Darstellung ist damit an einen bestimmten Rechnertyp gebunden, daher auch die Bezeichnung maschinenorientierte Sprachen. Dennoch unterscheiden sich die Assembler-Sprachen verschiedener Mikrorechner im Grundsätzlichen voneinander nur wenig. Der Befehlssatz einer solchen Sprache ist stets umfangreicher als der entsprechende Maschinenbefehlssatz des Mikrorechners, da sogenannte Pseudobefehle, die für die Erleichterung der Programmierarbeiten

symbolische Speicheradresse	symbolischer Befehl		Erläuterung (Kommentar)
⋮	⋮	⋮	⋮
MARKE 2:	SUB	*A, C*	Subtraktion *C* von *A*
	LD	*B, A*	Transport von *A* nach *B*
	ADD	*A, D*	Addition von *D* und *A*
	LD	*C, A*	Transport von *A* nach *C*
	PUSH	*B*	*B, C* in Kellerspeicher
MARKE 3:	CALL	UP1	Aufruf eines Unterprogramms
	LD	*C, A*	
⋮	⋮	⋮	⋮

Bild 62. Ausschnitt aus einem Assemblerprogramm

und der Steuerung des Programmaufbaus sorgen, hinzukommen. Das Umwandeln von Programmen aus der Assembler-Sprache in die Maschinensprache der Rechner nennt man *Übersetzen* oder *Assemblieren*. Dies ist ein schematisches Verfahren, das ein Rechner selbst durchführen kann. Das hierfür erforderliche Programm wird Übersetzungsprogramm, kurz *Übersetzer* oder *Assembler* genannt. Hierbei werden den symbolischen Bezeichnungen und Ausdrücken die entsprechenden binär codierten Maschinenbefehle, Adressen und Daten zugeordnet sowie die entsprechende Maschinenbefehlsfolge automatisch zusammengestellt.

Der Vorteil der Programmierung in einer maschinenorientierten Sprache besteht darin, daß in der Phase der Programmentwicklung die Befehlsfolgen einfach formuliert und Tests sowie Korrekturen sehr gut durchgeführt werden können. Das Ergebnis ist außerdem ein rechenzeit- und speicherplatzgünstiges Programm. Ein weiterer Vorteil ergibt sich dadurch, daß der Rechner während der Übersetzung eine ständige Prüfung auf Verletzung der Sprachregeln – auf sogenannte syntaktische Fehler – vornimmt und der Programmierer Mitteilung erhält, ob er derartige Programmfehler gemacht hat und welcher Art sie sind.

Mit maschinenorientierten Programmiersprachen kann man zur Zeit für Mikrorechner die effektivsten Programme erzeugen. Allerdings ist der Aufwand doch noch relativ hoch, und die Programme sind häufig nur für einen Mikrorechnertyp verwendbar. Ein weiterer Nachteil besteht darin, daß eine maschinenorientierte Sprache schwierig zu erlernen ist und außerdem gewisse Kenntnisse über die innere

Wirkungsweise des verwendeten Mikrorechners, also über die Hardware, notwendig sind.

Die maschinenorientierten Sprachen werden dann vorteilhaft eingesetzt, wenn vielbenutzte System- und Betriebsprogramme für Mikrorechner oder zeitkritische Steuerprogramme zu erstellen sind. Außerdem wird mit solchen Sprachen gearbeitet, wenn ein und dasselbe Programm in einer großen Zahl von Mikrorechnern (z.B. als Einbaurechner) fest einzuprogrammieren ist.

6.5.4. Höhere Programmiersprachen

Die bisher behandelten Programmiersprachen setzen alle mehr oder minder den benutzten Mikrorechner in den Mittelpunkt der Programmierung. Alle entstehenden Programme sind an einen Mikrorechnertyp gebunden; bei Beginn der Programmierung müssen eindeutig der verwendete Rechner sowie dessen wesentlichste Eigenschaften bekannt sein. Bei Verwendung einer höheren Programmiersprache («hoch» bedeutet «weit weg» vom Rechner) existiert demgegenüber keine direkte Beziehung zu einem bestimmten Rechner. Eine solche Sprache ist allgemein, d.h. für jeden Mikrorechner, in der Regel sogar für jeden Digitalrechner geeignet. Wichtig ist jedoch, daß der betreffende Rechner das entsprechende Übersetzungsprogramm, *Compiler* genannt, besitzt, das es ihm ermöglicht, die in der höheren Programmiersprache vorliegenden Programme in solche seiner «eigenen» Sprache umzuwandeln.

Im Laufe der Entwicklung der Rechentechnik wurden sehr viele höhere Programmiersprachen für unterschiedliche Problemstellungen, bestimmte technische Verfahren oder ausgewählte fachliche Einsatzfälle entwickelt. Bezeichnungen, wie «problemorientierte Sprachen», «verfahrensorientierte Sprachen» oder «Fachsprachen» stammen daher. Die Anzahl der auf der Welt zur Zeit vorhandenen höheren Programmiersprachen wird auf weit über 1000 geschätzt. Die erste symbolische problemorientierte und maschinell übersetzbare Programmiersprache war FORTRAN (formula translation – Formelübersetzung). Sie wurde in der Zeit von 1953 bis 1957 geschaffen. Bestimmt war sie zur Bearbeitung von wissenschaftlich-technischen Aufgaben. FORTRAN ist heute die älteste und in der Rechentechnik immer noch eine der verbreitetsten höheren Programmiersprachen. Auch die Sprache ALGOL (algorithmic language – Algorithmische Sprache) gehört zur gleichen Gruppe. Bei ihrer Entwicklung in den Jahren 1957 bis 1961 galt ALGOL als universelle Programmiersprache. Auf ihrer Grundlage entstand eine Reihe von neuen und fruchtbaren Ideen, die wichtige Impulse zur Entwicklung neuer Programmiersprachen gaben.

Der Vorteil einer höheren Sprache besteht darin, daß sich ein komplizierter Sachverhalt mit den Mitteln der Sprache einfach und effektiv formulieren läßt. Dafür stehen eine Reihe von leicht einprägbaren Sprachelementen, meist aus dem englischen umgangssprachlichen Wortschatz, zur Verfügung (Bild 63). Ein weiteres Plus einer höheren Sprache ist ihre leichte Erlernbarkeit. Außerdem braucht der Programmierer keine speziellen Kenntnisse über Aufbau und Wirkungsweise des verwendeten Rechners. Der Nachteil ist darin zu sehen, daß die entstehenden Programme gegenüber denen, die mit Maschinen- oder Assembler-Sprachen geschrieben wurden, weniger effektiv sind, d.h. einen höheren Speicherplatzbedarf sowie längere Rechenzeiten aufweisen. Dies hat seine Ursache zum einen in der schematischen Arbeitsweise des Compilers während des Übersetzungsvorgangs, und zum anderen ist es die direkte Folge der Abkehr von einer Berücksichtigung der konkreten Besonderheiten des jeweiligen Rechners.

```
CONST  PI = 3.14157;
VAR    X:INTEGER;
       A.B,C.VOLUMEN.DICHTE.GEWICHT REAL;
BEGIN
READ (A, B.C.DICHTE);
FOR   X: = 1 TO 10 DO
      BEGIN
      VOLUMEN: = (A*X+B*X+X*X*PI)*C;
      GEWICHT:   VOLUMEN*DICHTE;
      WRITELN (X.VOLUMEN.GEWICHT)
      END
END;
```

Bild 63. Ausschnitt aus einem Programm in höherer Programmiersprache

Höhere Programmiersprachen werden in der Mikrorechentechnik sehr vielseitig angewendet, da auf diese Weise eine effektive und fehlerarme Programmierung erreicht wird. Prinzipiell sind alle in der Rechentechnik bewährten höheren Sprachen zur Programmierung von Mikrorechnern einsetzbar. Praktisch werden jedoch gegenwärtig nur die eingesetzt, deren Sprachumfang nicht allzu groß ist. Im folgenden werden die für die Mikrorechentechnik wichtigsten höheren Programmiersprachen aufgeführt.

PASCAL

Die Programmiersprache PASCAL ist eine der modernsten algorithmischen Sprachen und wurde von *N. Wirth* in der Zeit von 1968 bis 1973 entwickelt. Sie erhielt ihren Namen zu Ehren des bereits erwähnten französischen Physikers und Mathematikers *B. Pascal.* Die

Sprache PASCAL berücksichtigt nahezu alle bisherigen Erkenntnisse aus der Entwicklung höherer Programmiersprachen entsprechend dem Stand zum Entwicklungszeitpunkt. Geschaffen wurde sie ursprünglich für die Lehre, sie ist aber genauso gut in der Praxis anwendbar. Die wichtigsten Ziele einer höheren Programmiersprache, nämlich leichte Erlernbarkeit und Klarheit der Sprache, gute Lesbarkeit und Struktur der Programme für den Benutzer sowie leistungsfähige Übersetzung durch den Rechner, werden überzeugend realisiert. PASCAL bietet ein reichhaltiges Angebot an Datenstrukturen. Zur Steuerung des Programmablaufs gibt es alle notwendigen Anweisungstypen, die für eine gute Strukturierung der Programme gebraucht werden. Die Sprache PASCAL ist international standardisiert, was die Austauschbarkeit der Programme zwischen verschiedenen Rechnern erleichtert [23].

BASIC

Die Programmiersprache BASIC (**b**eginners **a**ll purpose symbolic **i**nstruction **c**ode – symbolische Allzweicksprache für Anfänger) gehört ebenfalls zur Gruppe der modernen höheren Sprachen. Entwikkelt wurde sie in England in der Zeit von 1963 bis 1964. BASIC ist eine «einfache» Sprache und tatsächlich für Programmieranfänger oder für Programmierer, die nur hin und wieder ein kleines Programm zu schreiben haben, sehr gut geeignet; denn sie ist leicht erlernbar und verständlich. Leider unterstützt sie aber nicht einen strukturierten Programmentwurf. BASIC besteht aus einfachen, international gebrauchten englischen Wörtern und algebraischen Zeichen. Wie jede Programmiersprache hat sie auch grammatische Regeln, die jedoch äußerst einfach gehalten sind. BASIC ist außerdem auf sehr kleinen und weniger leistungsfähigen Mikrorechnern einsetzbar. Besonders geeignet ist die Sprache für die Programmierung im direkten Dialog mit dem Rechner (Dialogbetrieb).

Von BASIC gibt es zur Echtzeitprogrammierung ebenfalls eine Spracherweiterung (Real-Time-BASIC), die mit gutem Erfolg in der Programmierung von Mikrorechner-Maschinensteuerungen sowie Steuerungen in Experimentalaufbauten und Meßgeräten eingesetzt wird.

PLM

Die erste für Mikrorechner verwendete höhere Programmiersprache war PLM (**p**rogram **l**anguage for **m**icrocomputers – Programmiersprache für Mikrorechner; Firmenbezeichnung). Sie wurde aus der höheren Programmiersprache PL/1, die man bei Großrechnern viel-

fach einsetzt, als Untermenge «abgeleitet». Zusätzlich sind spezielle, die Mikrorechnerprogrammierung unterstützende sprachliche Erweiterungen sowie einige maschinenorientierte Komponenten enthalten, die Besonderheiten von Mikroprozessoren berücksichtigen. Die vorhandenen sprachlichen Elemente gestatten damit eine effektive Programmierung.

Die Sprache PLM ist günstig einzusetzen bei der Programmierung von System- und Betriebsprogrammen (Systemsoftware) sowie für Echtzeitaufgaben, beispielsweise für Mikrorechnersteuerungen. Mit Erfolg ist auch die Formulierung von Datenverwaltungs- und Textverarbeitungsprogrammen in dieser Sprache möglich [24].

PLZ

Auch diese Sprache wurde für die Programmierung von Mikrorechnern geschaffen. Mit PLZ (program language Zilog; Firmenbezeichnung) wurde versucht, die Vorteile einer höheren Sprache – leistungsfähige und einfache Programmierung – mit denen einer maschinenorientierten Sprache – die Erzeugung effektiver Programme – zu verknüpfen. Dies erreicht man, indem das zu programmierende Problem in zwei Komponenten gegliedert wird: einen Teil, der vom Typ des Mikrorechners unabhängig ist und einen anderen, der die konkreten Besonderheiten des verwendeten Mikrorechners, insbesondere die Ein- und Ausgabeoperationen, berücksichtigt. Die erste Teilaufgabe wird durch eine Programmiersprache beschrieben, die weitgehend der Sprache PASCAL angelehnt ist, während man für die zweite Teilaufgabe die Assembler-Sprache zugrunde legt.

Damit ist PLZ eine höhere Sprache, die ebenfalls für System- und Betriebsprogramme eines Mikrorechners sowie für Mikrorechnersteuerungen eingesetzt werden kann. Außerdem wird der systematische Programmentwurf durch Bereitstellen von entsprechenden Anweisungstypen unterstützt. Der Nachteil der Sprache PLZ besteht darin, daß bestimmte Programmteile nun doch rechnerspezifisch ausfallen, die den Programmaustausch zwischen verschiedenen Mikrorechnern erschweren.

6.6. Maschinelle Stationen eines Programms

Auf dem Weg von der Aufgabenstellung eines Programms bis zur wirklichen Abarbeitung auf dem Mikrorechner ist die Etappe der Programmentwicklung das längste und aufwendigste Stück. Darauf wurde bereits eingegangen. Als Ergebnis entsteht zunächst ein fertiges Programm, das sogenannte *Quellprogramm.* Es ist in einer Programmier-

sprache geschrieben, die zum einen dem Programmierer verständlich sowie geläufig ist und zum anderen der Mikrorechner «versteht», also verarbeiten kann. «Verarbeiten» bedeutet nicht unbedingt, daß dem Rechner das Programm bereits als Steuer- oder Arbeitsvorschrift dient, sondern in der Regel folgen erst bestimmte Wandlungen in der Darstellungsform des Programms. Diese Umformungen sind gewöhnlich Prozesse, die in Minuten oder Sekunden mit Hilfe eines Rechners (Rechner als «Werkzeug») durchzuführen sind. Hierbei durchläuft das Programm maschinelle Stationen, die im folgenden näher betrachtet werden sollen.

Übersetzer-Binde-Lade-Systeme

Zunächst wird das in einer Programmiersprache geschriebene Programm mit Hilfe eines Rechners und vorliegender Übersetzungsprogramme (*Assembler*, *Compiler*) in das *Objektprogramm* umgeformt. Dieses ist in einer dem Programmierer meist nicht zugänglichen «Zwischensprache» dargestellt. Das *Übersetzen* kann sich unter Umständen auch über mehrere Zwischenschritte fortsetzen, bis letztlich ein Programm entsteht, das aus binären Befehls- und Datenworten aufgebaut ist und das der Mikroprozessor direkt «interpretieren», also abarbeiten kann. Letzteres ist das *Zielprogramm*. Handelt es sich um ein größeres Maschinenprogramm, das sich aus verschiedenen Teilen zusammensetzt, dann wird vielfach auch noch ein maschineller Prozeß des «Zusammenfassens», das *Binden* einzelner Komponenten zu einem Gesamtprogramm, eingeschoben. In einem anschließenden Schritt wird das so aufbereitete Maschinenprogramm in den Hauptspeicher des Mikrorechners transportiert und an die richtige «Stelle» im Speicher plaziert. Diesen Prozeß bezeichnet man auch als *Laden* des Programms. Erst nach dem Laden und Start ist es dem Mikrorechner möglich, das Programm Befehl für Befehl auszuführen und seinen *Lauf* zu organisieren.

Die aufgeführten maschinellen Stationen eines Programms sind während der Programmentwicklung wiederholt durchzugehen, falls während der Übersetzung, des Bindens oder Laufs Fehler festgestellt werden sollten (Bild 64). Sie führen zu einer Korrektur oder Änderung des Quellprogramms. Für solche Arbeiten steht dem Programmentwickler ebenfalls ein geeignetes Standardprogramm, der *Editor*, zur Verfügung.

In der praktischen Anwendung der Mikrorechentechnik müssen nicht immer alle aufgeführten Stationen unbedingt durchlaufen werden. Dies ist nur notwendig, wenn ein neues Programm in maschinenorientierter oder höherer Programmsprache entwickelt werden soll.

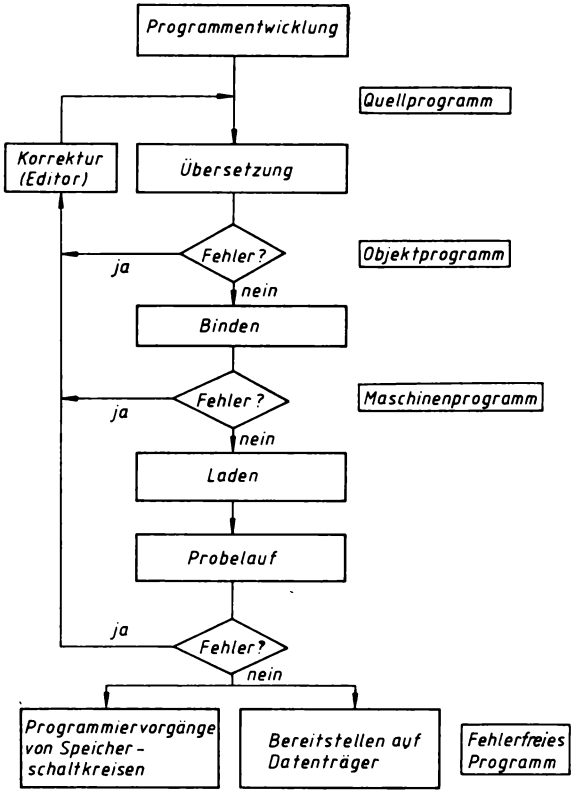


Bild 64. Maschinelle Stationen eines Programms bei einem Übersetzer-Binde-Lade-System

Interpretersysteme

In der bisherigen Betrachtung wurde davon ausgegangen, daß ein Programm die erwähnten Stationen als eine geschlossene Einheit durchläuft. In diesem Fall wird ein Quellprogramm komplett übersetzt. Einen grundsätzlich anderen Weg beschreitet man, wenn die Übersetzungs- und Umwandlungsprozesse nicht auf das ganze Programm ausgedehnt sind, sondern jeweils nur auf einzelne Anweisungen. Eine derartige Arbeitsweise ist bei interpretierenden Programmsystemen, den *Interpretern*, zu finden. Sie übersetzen das in einer Programmiersprache geschriebene Programm schrittweise, Anwei-

sung für Anweisung, in Maschinenbefehle und führen nach jedem Schritt die übersetzte Anweisung auch sogleich aus (Bild 65). Ein Interpreter enthält daher außer Übersetzungsprogrammen und der Möglichkeit der entsprechenden Fehleranzeigen auch ein «Laufsystem», das die automatische Abarbeitung des zu interpretierenden Programms organisiert. Der Nachteil dieser Arbeitsweise besteht darin, daß oft die Rechenzeit sehr hoch ist und sich nicht jede Programmiersprache dafür eignet, ein zu interpretierendes Programm zu beschreiben. Als besonderer Vorteil ist die einfache Handhabung des Systems während der Programmentwicklung zu werten, da vom Interpreter ein Programm ohne störende Zwischenschritte abgearbeitet und bei auftretenden Fehlern leicht korrigiert werden kann.

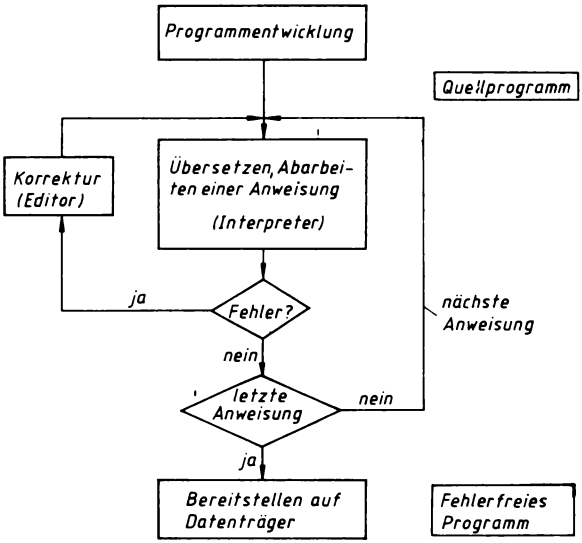


Bild 65. Bearbeitung eines Programms bei einem Interpretersystem

Direktverarbeitende Systeme

Moderne Mikrorechnerkonzeptionen gehen davon aus, die aufgezeigten Umwandlungsprozesse mit Hilfe von Software ganz zu vermeiden. Ein in einer leistungsfähigen Programmiersprache geschriebenes Programm wird *direkt* auf dem Mikrorechner abgearbeitet (high level microcomputer – Hochsprachen-Mikrorechner) oder diese

Vorgänge laufen für den Anwender «unsichtbar» in spezieller Soft- und Hardware im Mikrorechner ab (Mikrorechner mit direkter Hochsprachenverarbeitung). Derartige Systeme sind besonders einfach zu handhaben und sehr effektiv zu programmieren.

6.7. Physische Darstellung der Programme

In der Mikrorechentechnik hat man mehrere Arten der Speicherung bzw. Zwischenspeicherung von Programmen zu unterscheiden. Im allgemeinen erfolgt außerhalb des Mikrorechners ein Speichern von Quell-, Objekt- und Maschinenprogrammen auf Datenträgern. Diese können z.B. Lochbänder, Magnetbänder, Magnetbandkassetten oder Disketten sein. Sie alle sind im Bestand des externen Speichers des Mikrorechners einfügbar. Soll der Mikrorechner jedoch ein Maschinenprogramm abarbeiten, dann muß es im Hauptspeicher bereitgestellt werden. Hierfür gibt es die drei folgenden Verfahren.

- RAM-Speicherung:

Das Programm wird nach Inbetriebnahme des Mikrorechners oder bei jeder neuen Programmbenutzung von einem externen Datenträger in den RAM-Speicher transportiert (Laden des Programms) und gelangt dort zum Bearbeiten. Ein nicht mehr benötigtes Programm wird von einem neuen Programm überschrieben. Diese Vorgehensweise ist für die klassische Datenverarbeitungsanlage typisch, findet aber auch oft bei Mikrorechnern Verwendung.

- PROM-Speicherung:

Die Programme sind in programmierbare ROM-Schaltkreise (PROM) zu speichern, die der Anwender selbst zu programmieren hat. Das entsprechende Maschinenprogramm wird in einem besonderen Arbeitsgang mit Hilfe eines Programmiergeräts in die betreffenden integrierten Schaltkreise eingeschrieben oder «eingebrannt». Obwohl dieser Prozeß als «Programmierung» der Schaltkreise bezeichnet wird, ist es aus der Sicht des Mikrorechners ein Ladevorgang. Mit dem Einstecken des programmierten Schaltkreises in eine Schaltkreisfassung des Mikrorechner-Hauptspeichers findet tatsächlich ein «Laden» des Programms statt.

- ROM-Speicherung:

In diesem Fall ist das Maschinenprogramm bereits während der Herstellung der integrierten Speicherschaltkreise zur Verfügung zu stellen. Es wird in mehreren technologischen Schritten als geometrisches Abbild in den entsprechenden Schaltkreis-Chip eingeprägt. Das Programm muß dabei fehlerfrei vorliegen, da eine nachträgliche Ände-

rung nicht mehr möglich ist. Dieses Verfahren ist nur dann ökonomisch, wenn sehr große Stückzahlen des Schaltkreises mit dem gleichen Programm benötigt werden. Bei Einbaurechnern ist das oft der Fall.

Die PROM- und ROM-Programmspeicherung wird vielfach bei Mikrorechnersystemen mit festen, unveränderlichen Aufgabenstellungen, z.B. bei Mikrorechnersteuerungen, oft aber auch in Kombination mit der RAM-Programmspeicherung verwendet.

6.8. Programmentwicklungs- und Testsysteme

Das wichtigste Werkzeug für die Programmentwicklung und alle damit zusammenhängenden Prozesse ist der Rechner selbst. Damit er diese Funktion ausüben kann, müssen jedoch eine Reihe von zusätzlichen Arbeitsmitteln, wie geeignete Softwaresysteme (Projektierungssysteme, Übersetzer, Testmonitore, Simulatoren, Korrektursysteme) und entsprechende Hardwareeinrichtungen (zusätzliche Speichereinheiten, Anschlüsse für Testperipherie und Prozeß) zur Verfügung stehen. Ein derartig erweiterter Rechner, der speziell für Entwicklungsarbeiten ausgerüstet ist, wird auch als *Programmentwicklungssystem* bezeichnet. In der Praxis der Mikrorechnerprogrammierung hat man hierbei zwei Möglichkeiten. Entweder man benutzt denselben Mikrorechner zur Programmentwicklung, der auch später das zu erstellende Programm abarbeiten soll, oder aber man wählt ein – meist leistungsfähigeres und komfortableres – Rechnersystem (Hostcomputer – Wirtsrechner) aus, welches nur der möglichst effektiven Programmentwicklung dient. Außerdem hat man noch zu unterscheiden, ob der Wirtsrechner vom gleichen Typ ist – also den gleichen Mikroprozessor enthält – wie der Mikrorechner (Zielrechner), für den das Programm bestimmt ist oder nicht. Ist dies nicht der Fall, so wird eine «Abbildung» des entsprechenden Mikrorechners im Wirtsrechner vorgenommen. Ein solches Vorgehen wird auch als Cross-Verfahren (engl. cross – Kreuzung, Mischung) bezeichnet, und die entsprechenden Programme nennt man Cross-Software.

Ein Programmentwicklungssystem hat einen Aufbau, der für einen «Experimentier»-Rechner typisch ist. Da ein solches System in einer Einrichtung gewöhnlich nur in sehr geringer Stückzahl – meist nur einmal – zur Verfügung zu stehen braucht, jedoch für eine größere Zahl von Zielrechnern eingesetzt werden kann, ist es auch ökonomisch gerechtfertigt, eine umfangreiche technische Ausstattung vorzusehen. Programmentwicklungssysteme haben daher einen großen

Arbeitsspeicher, viele periphere Geräte, eine Einrichtung zur Programmierung von PROM-Schaltkreisen sowie mehrere Anschlußmöglichkeiten zur Steuerung von externen Vorgängen. Außerdem sorgen geeignete Betriebsprogramme (Betriebssystem) für die Organisation aller Arbeiten, und eine Reihe von «Dienstprogrammen» steht für Routineaufgaben sowie für die Prüfung und Kontrolle (z. B. Fehlerverfolgungsprogramme) zur Verfügung.

Besteht die Möglichkeit, daß mehrere Nutzer zur gleichen Zeit über Terminals, sogenannte *Programmierarbeitsplätze*, Zugriff zum Entwicklungssystem haben, dann kann jeder Anwender direkt am Arbeitsplatz sein Programm entwerfen, testen und dokumentieren. Das ist allerdings nur bei sehr leistungsfähigen Anlagen der Fall.

7. Anwendung der Mikrorechentechnik

7.1. Einsatzgebiete

Im folgenden werden einige typische Anwendungsfälle der Mikrorechner dargestellt. Zwei wichtige Einsatzformen lassen sich unterscheiden. Zum einen kann ein Mikrorechner als selbständiges Gerät – gegebenenfalls mit Zusatzeinrichtungen ausgerüstet – Verwendung finden. Im anderen Fall ist er Bestandteil einer komplexen Einrichtung oder einer Maschine und hat dort die Funktion eines eingebauten Bauelements oder einer Baugruppe. Im letzteren Fall spricht man vielfach auch von «eingebetteten» Mikrorechnern. In beiden Nutzungsformen werden Aufgaben der Informationserfassung, -verarbeitung, -übertragung und -auswertung sowie Steuerungsaufgaben übernommen. Wichtige Einsatzgebiete sind

- Kleinstdatenverarbeitung
- Informations-, Auskunfts- und Recherchesysteme
- Kleinstdatenbanken
- Steuer- und Prozeßrechner
- Automatisierungsgeräte und -anlagen
- Meßwerterfassungs- und -verarbeitungssysteme.

Mit Hilfe von verschiedenen Programmen (Softwarevariabilität) und wählbarem technischem Ausrüstungsgrad (Hardwarevariabilität) ist ein Mikrorechner für jeden der Einsatzfälle einzurichten. Um hierbei auch günstige ökonomische Lösungen zu erreichen, ist es allerdings notwendig, sowohl im Bereich der Hardware als auch Software einen baustein- oder modulartigen Systemaufbau und die Möglichkeiten der Systemerweiterung mit «gestuftem» Leistungsumfang vorzusehen. Dies ist bei modernen Mikrorechner-Schaltkreisfamilien sowie den entsprechenden Programmsystemen auch der Fall.

7.2. Anwendungsbeispiele

7.2.1. Kleinstdatenverarbeitungsanlagen

Eine breite Verwendung findet der Mikrorechner als Kleinstrechner in Büro und Haushalt. Diese Kleinstdatenverarbeitungsanlagen sind als programmierbare Taschenrechner, Aktentaschenrechner (Personencomputer), Tischrechner und Bürocomputer ausgeführt. Auch Hobby-, Lern- und Heimcomputer gehören zu dieser Kategorie. Oft enthalten derartige Geräte kleinere Ein- und Ausgabeeinrichtungen, wie Miniaturtastatur, Miniaturdrucker und Kleinstdisplay (Anzeigeeinrichtung). Fast jeder Rechner wird mit entsprechenden Betriebsprogrammen geliefert, die für eine leichte Programmierbarkeit und Handhabung sorgen. Vielfach kann auch ein Kassettenrecorder für die Aufzeichnung bzw. Speicherung von Anwenderprogrammen sowie ein herkömmliches Fernsehgerät als komfortable alphanumerische oder grafische Anzeige angeschlossen werden.

7.2.2. Hochleistungsrechner und verteilte Rechnersysteme

Da die Leistungsfähigkeit eines einzelnen Mikroprozessors oder Mikrorechners im Verhältnis zu einem kommerziellen Großrechner relativ klein ist, können *Hochleistungsrechner* nur als Verbund einer großen Zahl von Mikroprozessoren oder Mikrorechnern aufgebaut werden (s. 5.7.). Sind diese unmittelbar zu einem Großrechner verknüpft, arbeiten sie gewöhnlich *gemeinsam* an einer oder mehreren zu lösenden Aufgaben. Beispiele sind die Aufbereitung von Wetterkarten, Wettervorhersage, Berechnungen komplizierter statischer und dynamischer Lastfälle in Bauwesen und Aerodynamik sowie Aufgaben zur Dynamik von Maschinenbauerzeugnissen.

Sind die verkoppelten Mikrorechner zu einem *verteilten* System räumlich getrennt angeordnet, so sind in den meisten Fällen die einzelnen Rechner zur Lösung der unmittelbar am jeweiligen Standort auftretenden Probleme selbst in der Lage («verteilte Intelligenz»). Nur bei besonders umfangreichen Aufgabenstellungen sowie speziellen Problemen erfolgt deren Lösung mit Hilfe anderer Mikrorechner. Falls dies nicht ausreicht, kann ein zentral angeordneter Hochleistungsrechner in Anspruch genommen werden. Schwierigkeiten bereiten bei solchen komplexen Systemen jedoch häufig die Herstellung der Software sowie die Koordinierung aller im Gesamtsystem ablaufenden Prozesse.

7.2.3. Mikroprozeßrechner

Sehr umfangreiche und wichtige Anwendungsbereiche ergeben sich, wenn ein Einsatz als Mikroprozeßrechner konzipiert wird. In der

Praxis sind die zu steuernden Vorgänge und Einrichtungen sehr vielgestaltig. Sie reichen von der Forderung nach sehr schneller Verarbeitung relativ weniger Daten über Einsatzfälle mit einem sehr großen Umfang der zu verarbeitenden Daten bei längeren Reaktionszeiten bis hin zu Aufgaben, wo eine schnelle Reaktion und die Verarbeitung sehr vieler Daten notwendig sind. Beispiele aus der Praxis sind Prozeßsteuerungen chemischer Anlagen (viele Daten, aber relativ langsam), KfZ-Steuerung (sehr schnelle Verarbeitung), Kraftwerks- und Energieverteilungs-Steuerungen (hohe Zuverlässigkeit), Steuerungen von Vorgängen im Transport- und Verkehrswesen, wie Luftverkehr (schnelle Verarbeitung *und* hohe Zuverlässigkeit). Auch im Haushalt lassen sich Anwendungsbeispiele finden, darunter programmgesteuerte Waschmaschinen und elektronische Strickmaschinen (wenig Daten).

Die Vielseitigkeit der Anwendung von Mikroprozeßrechnern bringt es mit sich, daß nahezu für jeden Einsatzfall auch spezielle Echtzeitprogramme zu schaffen sind. Viele Hersteller von Mikrorechnern für den Prozeßeinsatz sind daher bemüht, geeignete *Grundsoftware*, sogenannte Systemkerne für Echtzeitbetrieb sowie oft benutzte «Standardprogramme» für ein möglichst breites Einsatzgebiet zur Verfügung zu stellen. Der Anwender kann dann mit Hilfe einiger Zusatzprogramme seine speziellen Aufgaben lösen.

7.2.4. Steuerungs- und Automatisierungssysteme

Um der Variantenvielfalt der Hard- und Softwarestrukturen beim Mikroprozeßrechnereinsatz zu begegnen, werden Steuerungssysteme entwickelt, die jeweils eine ganze Klasse von Problem- und Aufgabenstellungen lösen können. In solch eine Steuerung werden ein oder mehrere Mikrorechner eingebettet, und die Gesamtstruktur ist so konzipiert, daß sie flexibel einsetzbar, modulartig aufzubauen und zu programmieren sind. Solche «Standard»-Systeme sind bei Industrieroboter-Steuerungen, Steuerungen für CNC-Werkzeugmaschinen (CNC: computerized numerical control – vom Rechner numerisch gesteuert), für Fertigungsautomatisierung und feinmechanisch-optische Steuerungen sowie bei Steuerungen auf vielen Gebieten der Automatisierungstechnik anzutreffen. Der Anwender kann mit Hilfe solcher problembezogener Steuerungssysteme und einiger leicht herzustellender Nutzerprogramme unterschiedliche Aufgabenstellungen in seiner unmittelbaren Umgebung lösen. Die im Steuerungssystem vorhandenen Programmierhilfsmittel unterstützen ihn bei der Programmierung, und die enthaltenen Betriebsprogramme entlasten ihn von Arbeiten der Prozeßorganisation sowie Systemprogrammierung.

7.3. Betriebssysteme

7.3.1. Verwaltung der Betriebsmittel

Um alle in einem Mikrorechner ablaufenden Prozesse zu koordinieren, zu überwachen, zu automatisieren und um einen effektiven Betrieb des Mikrorechners unter den aufgezeigten Einsatzfällen zu gewährleisten, sind besondere Betriebs- und Systemprogramme erforderlich. Diese haben einen allgemeingültigen Charakter, da sie nicht zur Lösung einer einzigen Aufgabe dienen, sondern Voraussetzungen zur Lösung aller auftretenden Problemstellungen am jeweiligen Rechnersystem schaffen. Sie werden in einem besonderen Programmsystem, dem *Betriebssystem*, zusammengefaßt. Ein Mikrorechner-Betriebssystem besteht demzufolge aus allen denjenigen Softwarekomponenten, die auf der Grundlage der Hardware die Basis der möglichen Betriebsarten des Mikrorechners bilden. Das Betriebssystem übernimmt insbesondere die Verwaltung und Kontrolle aller sogenannten *Betriebsmittel* des Rechners. Als wichtigste Betriebsmittel eines Mikrorechnersystems zählen:

- Mikroprozessoren
- Speichereinheiten
- Ein- und Ausgabeeinheiten
- Programme (in einer Bibliothek)
- Datenbestände (in einer Datei).

Im Bild 66 ist der schematische Aufbau eines Betriebssystems dargestellt. Wichtigste und zentrale Komponente ist der sogenannte *Be-*

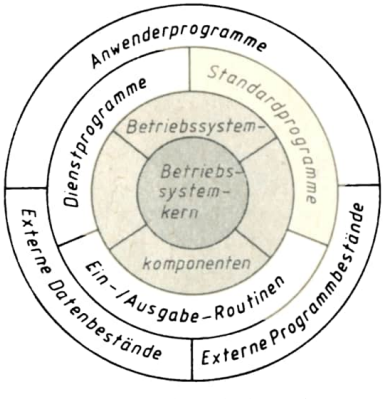


Bild 66. Schematischer Aufbau eines Betriebssystems

triebssystemkern. Sein Aufbau und seine Funktionsweise charakterisieren Art und Eigenschaften des gesamten Betriebssystems. Um ihn «herum» sind weitere Betriebssystemkomponenten und die zu verwaltenden Betriebsmittel einschließlich der aktiven Anwenderprogramme angeordnet.

Ein Betriebssystem bietet dem Nutzer beträchtliche Unterstützung bei der Lösung seiner Aufgaben in Form des vereinfachten Gebrauchs des Mikrorechners sowie Entlastung von Routineaufgaben. Insbesondere wird das Programmieren, Übersetzen und Testen von Programmen verbessert und die gleichzeitige oder aufeinanderfolgende Bearbeitung von mehreren Anwenderprogrammen automatisiert und überwacht. Damit ist für eine Erhöhung des Wirkungsgrades sowie für eine Steigerung des Preis-Leistungs-Verhältnisses des Mikrorechners gesorgt.

7.3.2. Betriebssystemarten

In der praktischen Arbeit mit einem Betriebssystem kommt ein Nutzer nicht mehr unmittelbar mit dem Rechner in Berührung. Für ihn ist das Betriebssystem jetzt der «Partner», an den er seine «Aufträge» erteilt. Im Normalfall wird es sich bei einem solchen Auftrag um das Bearbeiten eines Programms handeln. In der Praxis hat man jedoch verschiedene Arten der «Auftragserteilung» und «Programmbearbeitung» zu unterscheiden. Aus der Sicht des Anwenders ergeben sich folgende Möglichkeiten.

– *Geschlossene Verarbeitung* (closed processing): Der Auftrag oder die Aufgabe wird vollständig formuliert und als geschlossene Einheit an das Betriebssystem, bzw. den Rechner übergeben. In der Fertigstellung der Aufgabe bestehen keine strengen zeitlichen Forderungen. Nach Abschluß der Aufgabe wird die Lösung dem Nutzer als ein abgeschlossener Komplex übergeben. Ein typisches Beispiel wäre die Berechnung einer Gleichung.

– *Interaktive oder «unterbrochene» Verarbeitung, Dialogbetrieb* (interactive processing): Der Nutzer stellt aus der Gesamtaufgabe eine Folge kurzer Teilaufgaben und erwartet nach jeder Teilaufgabe eine schnelle Bearbeitung bzw. «Antwort» vom Betriebssystem. Die «Reaktionszeit» des Rechners liegt in der Regel zwischen Bruchteilen einer Sekunde bis zu einigen Sekunden. Der Nutzer nimmt es jedoch auch in Kauf, wenn hin und wieder, infolge unterschiedlicher Bearbeitungszeiten, eine etwas größere Verzögerung auftritt. Die «Denkzeit» des Nutzers, also die Zeit, die vom Erhalt des letzten Ergebnisses bis zum erneuten Stellen einer Teilaufgabe vergeht, ist jedoch wesent-

lich größer (z.B. 20 Sek.) als die Antwortzeit des Rechners. Ein typisches Beispiel wäre hier der Dialog mit dem Rechner während der rechnergestützten Projektierung eines Erzeugnisses.

– *Echtzeitverarbeitung* (real time processing): Die an das Betriebssystem des Mikrorechners übergebene Aufgabe ist «ununterbrochen» zu bearbeiten und wird von einer Maschine, Anlage oder von einem zu steuernden Prozeß «gestellt». Auch hier kann die Aufgabe in kurze Teilaufgaben zerfallen. Die Zeitforderungen hinsichtlich ihrer Bearbeitung sind jedoch sehr hoch und müssen präzise in vorgegebenen Grenzen gehalten werden. Überschreiten die «Antwortzeiten» des Rechners die Grenzen, führt dies im allgemeinen zu Störungen und Ausfall der Maschine, Anlage oder zum Abbruch des zu steuernden Prozesses.

In diese Betrachtungen wurde bisher nur *ein* an das Betriebssystem gestellter Auftrag einbezogen. Moderne Betriebssysteme sind jedoch in der Lage, die gleichzeitige Erledigung mehrerer Aufträge zu organisieren. Für den ersten Fall bedeutet dies, daß das Rechnersystem eine Reihe («Stapel») von Aufträgen erhält, und das Betriebssystem versucht, diese optimal zu erledigen (batch processing – *Stapelbetrieb*).

Im zweiten Fall stellt eine größere Zahl von Nutzern nahezu gleichzeitig ihre kurzen Teilaufträge, und das Betriebssystem koordiniert alle Aufträge in der Weise, daß für jeden Nutzer der Eindruck entsteht, er arbeite allein am Rechner (interactive time sharing processing – *interaktiver Mehrnutzerbetrieb*).

Schließlich werden im letzten Fall mehrere Teilaufgaben gleichzeitig von einer Maschine, Anlage oder mehreren Maschinen bzw. Anlagen oder vom steuernden Prozeß erteilt, die den Parallellauf mehrerer Echtzeitaufträge (*Tasks*) im Mikrorechner unter der «Leitung» des Betriebssystems bewirken (multitasking real time processing – *Echtzeitbetrieb mehrerer Programme*).

Jeder der hier dargestellten Betriebsfälle (auch Kombinationen sind möglich) wird durch eine entsprechende Betriebssystemart repräsentiert.

Vielfach wird auch in der Praxis nicht nur die Art des Betriebssystems allein, sondern die Gesamtheit Betriebssystem *und* alle Betriebsmittel auf ein bestimmtes Anwendungsgebiet bezogen. Hierbei hat man beispielsweise Universal- oder Spezialrechnersysteme zu unterscheiden. Eine weitere Klassifizierung unterteilt in Stapelsysteme, Interaktive Systeme, Dialog- und Realzeitsysteme. Im allgemeinen gehört zu jeder speziellen Mikrorechneranwendung, z.B. für Entwicklungs- und Projektierungssysteme, Informations-, Datenbank-, Kommunikationssysteme usw., ein ausgewähltes, dem jewei-

ligen Anwendungsfall angepaßtes Betriebssystem. Der für seine jeweilige Realisierung erforderliche programmtechnische Aufwand ist jedoch beträchtlich. Außerdem wird von einem Betriebssystem auch ein gewisser Teil der Rechenleistung des Mikrorechners verbraucht. Diese Nachteile werden jedoch durch die sich bei der Nutzung des Mikrorechners ergebenden Vorteile bei weitem aufgewogen.

8. Zuverlässigkeit

8.1. Kenngrößen der Zuverlässigkeit

Neben technischen Kenngrößen, wie Funktionsumfang und Leistungsfähigkeit, wird ein Mikrorechnersystem noch von weiteren Parametern, z.B. Zuverlässigkeit und Ausfallsicherheit gekennzeichnet [25, 26]. Wie bei jeder technischen Einrichtung kann auch in Mikrorechnern während des Betriebs ein fehlerhaftes Arbeiten oder ein vollständiger Ausfall auftreten. Außerdem gibt es Fehler, die bereits in der Herstellungsphase sichtbar werden und zu einem von Anfang an fehlerhaften Erzeugnis führen. Daher unterscheidet man zwei Arten der Zuverlässigkeit: als Eigenschaft des Herstellungsprozesses und als dynamische Kenngröße des hergestellten Systems. Eine hohe Zuverlässigkeit in der Herstellung sichert die Produktion fehlerfreier Erzeugnisse. Eine hohe «Betriebs»-Zuverlässigkeit eines Geräts bzw. Systems charakterisiert dagegen die Sicherheit gegen Ausfall des Systems. Für den Nutzer ist besonders letztere Kenngröße von Bedeutung; nur sie wird im weiteren behandelt.

Fehler in einem System äußern sich im allgemeinen in einer Abweichung von der Systemfunktion. Bei Mikrorechnern können sie in der Hard- und Software auftreten. Während die theoretischen Grundlagen zu ihrer Beschreibung, Behandlung und Bewertung sowie die entsprechenden Fragen der Zuverlässigkeit für die Hardware von Rechnern einen relativ hohen Stand erreicht haben, ist dies für die Software bisher weniger der Fall.

Zuverlässigkeit

Die Zuverlässigkeit eines Systems ist eine Kenngröße, die erst während des Betriebs sichtbar wird. Vereinfacht gesprochen, versteht man darunter die Fähigkeit, daß ein Mikrorechner oder ein ausgewähltes Teilsystem davon, innerhalb vorgegebener Grenzen, den Anforderungen bei einer bestimmten Belastung genügt. Um jedoch die Zuverlässigkeit eines Rechners exakt anzugeben, braucht man ein Maß,

das mathematische Kenngrößen beschreibt. Dies allerdings ist mit erheblichen Schwierigkeiten verbunden. Vielfach werden die Methoden der gut ausgebauten Zuverlässigkeitstheorie benutzt. Sie haben jedoch nur einen bewertenden Charakter. Die entsprechenden Kenngrößen charakterisieren zwar die Funktionstüchtigkeit eines technischen Systems, geben aber weniger darüber Auskunft, wie und mit welchen Mitteln z. B. die Zuverlässigkeit eines Systems erhöht werden kann. Die Festlegung der Zuverlässigkeit als solch ein Wahrscheinlichkeitsmaß besagt: *Zuverlässigkeit* ist die Wahrscheinlichkeit dafür, daß bis zu einem bestimmten Zeitpunkt der Mikrorechner bei definierter Beanspruchung funktionstüchtig bleibt.

Mittlere Lebensdauer | ausfallfreie Betriebszeit
Hierbei handelt es sich ebenfalls um eine zufällige Größe; es ist die Zeitspanne zwischen Inbetriebnahme und Ausfall des Systems. Die mittlere Lebensdauer bzw. ausfallfreie Betriebszeit ist bei Mikrorechnern sowie deren Teilkomponenten, etwa hochintegrierten Schaltkreisen, eine sehr wichtige Kenngröße. Sie liegt z. B. bei Mikroprozessoren und Speicherschaltkreisen in der Größenordnung von 10000 bis zu mehreren Millionen Stunden. Ihr Wert wird vom Hersteller der integrierten Schaltkreise als technische Kenngröße angegeben.

Verfügbarkeit
Für praktische Anwendungen von Mikrorechnern ist ein weiteres Zuverlässigkeitsmaß besonders geeignet, die *mittlere Verfügbarkeit* eines Systems. Sie gibt die Wahrscheinlichkeit dafür an, den Mikrorechner zu einem bestimmten Zeitpunkt in einem funktionstüchtigen Zustand anzutreffen. Beträgt die Verfügbarkeit eines Rechners beispielsweise 98 % (für Mikrorechner ein sehr schlechter Wert), so würde dies bedeuten, daß der Rechner in einem Zeitraum von 100 Stunden im Mittel 2 Stunden defekt ist. Ebenso ist es möglich, daß der Rechner in einem Zeitbereich von 10 Stunden insgesamt 0,2 Stunden ausfällt. Daher ist die Einführung einer weiteren Kenngröße, die der *mittleren Ausfallzeit*, erforderlich. Sie ist die Zeit, die zwischen dem Ausfall des Rechners bis zu dem Augenblick verstreicht, wo der Fehler erkannt, lokalisiert und behoben ist. Besonders hohe Verfügbarkeit bzw. kleine Ausfallzeiten werden erreicht, wenn im Rechnersystem Fehlerermittlung und -abstellung automatisch erfolgen.

8.2. Möglichkeiten der Erhöhung der Zuverlässigkeit

Hohe Zuverlässigkeit eines Mikrorechners gewinnt man nur, wenn die Komponenten möglichst funktionssicher sind. Es gilt: Je höher

die Zuverlässigkeit der einzelnen Glieder, um so höher ist in der Regel auch die Zuverlässigkeit des Gesamtsystems. Umgekehrt ist aber zu beachten: Je mehr Einzelsysteme ausfallen können, um so niedriger ist die Zuverlässigkeit des Gesamtsystems. Genügt die sich ergebende Zuverlässigkeit eines entworfenen Gesamtsystems nicht mehr, was bei relativ komplexen Systemen (z.B. Mikrorechnerverbundsysteme) schnell erreicht ist, müssen zusätzliche Vorkehrungen getroffen werden. Hierbei hat man Maßnahmen zu unterscheiden, die die Zuverlässigkeit und solche, die die Verfügbarkeit erhöhen. In der Regel wird in der Praxis das Einfügen von sogenannten redundanten, d.h. «überflüssigen» Ersatzeinheiten oder Systemfunktionen vorgenommen. Sie benötigt man für die eigentliche Funktion des Rechners nicht.

Statische Redundanz
Es erfolgt der Einbau zusätzlicher Ersatzkomponenten, die beim Auftreten eines Fehlers die Funktion dieser Einheit übernehmen. Zum Beispiel werden wichtige zentrale Komponenten eines Rechners doppelt oder dreifach ausgeführt. Statische Redundanz erhöht die Zuverlässigkeit.

Dynamische Redundanz
Im System sind zusätzliche Systemeigenschaften enthalten, die ihm die Möglichkeit geben, das Fehlverhalten selbst zu erkennen, diesem entgegenzuwirken und gegebenenfalls abzustellen. Dynamische Redundanz erhöht die Verfügbarkeit des Rechners.

Systeme mit dynamischer Redundanz nutzen die zusätzlichen Mittel und Funktionen mit größerem Wirkungsgrad. In den meisten Fällen werden jedoch die Möglichkeiten der statischen und dynamischen Redundanz gemeinsam verwendet.

8.3. Fehlertolerante Systeme

In der Mikrorechentechnik, vor allem bei Mikrorechner-Verbundsystemen, haben besonders selbstprüfende oder fehlertolerante Systeme Bedeutung erlangt. Ihre besondere Fähigkeit ist – wie der Name sagt –, Fehler zu tolerieren, also in Kauf zu nehmen. Dies wird erreicht, indem man dynamische Redundanz in das Rechnersystem einfügt. Beim Auftreten eines Fehlers bleibt es in der Regel arbeitsfähig, hat dann aber möglicherweise eine verminderte Leistungsfähigkeit. Das System erkennt und beseitigt den Fehler, es sorgt für das

Weiterarbeiten unter neuen Bedingungen. Hierbei wird nicht unbedingt die wirkliche Ursache (z.B. ein defekter Schaltkreis) des Fehlers beseitigt, sondern eine Adaption, also Anpassen an den Fehler erreicht. Durch Strukturveränderungen oder Veränderung der Systemeigenschaften (z.B. Wechsel von Hard- oder Softwarekomponenten) ist eine Weiterarbeit des Rechners möglich und nur ein sogenannter «sanfter» Leistungsabfall bei zunehmender Zahl von Fehlerzuständen spürbar. Vielfach werden auch fehlerhafte Komponenten vom Rechnersystem abgekoppelt, so daß die verbleibenden deren Arbeit mit übernehmen müssen. Um jedoch die dynamische Redundanz gezielt einzusetzen und z.B. fehlertolerante Maßnahmen einzuleiten, ist ein umfassendes automatisches Erkennen, Bewerten und Behandeln (Diagnose und Prophylaxe) von Fehlerzuständen notwendig.

8.4. Technische Diagnose

Jeder Arzt weiß, wie bedeutsam eine gute, schnelle und zutreffende Diagnose für den Erfolg eines Heilprozesses sein kann. Jedoch ist es mitunter sehr schwierig, die richtige Diagnose zu stellen. Auch in der Technik spricht man von Diagnostik. Hier ist ihr Gegenstand nicht das «lebende System», sondern ein künstliches, technisches System. Wenn bisher weniger von technischer Diagnose gesprochen wurde, so lag dies daran, daß man im Falle eines Fehlers immer die Möglichkeit hatte, die Einrichtungen, Anlagen oder Geräte außer Betrieb zu setzen und sie bis in alle Einzelheiten zu zerlegen. Man brauchte also nicht zu diagnostizieren, sondern konnte «nachschauen», messen und überprüfen. Dies änderte sich naturgemäß, als komplexe technische Systeme entstanden, die nicht demontierbar sind oder solche, die ununterbrochen in Betrieb sein müssen. Als Beispiel kann ein hochintegrierter Mikrorechnerschaltkreis dienen. Ist dieser einmal hergestellt, so kann man nur noch über seine Anschlußklemmen und mit Hilfe komplizierter Meßeinrichtungen Auskunft über seinen inneren Funktionszustand sowie über eventuell vorhandene Fehlerzustände erhalten. Müssen diese Untersuchungen sogar während des normalen Betriebs eines Mikrorechners durchgeführt werden, dann wird die Problematik noch wesentlich komplizierter.

Bei einfachen elektronischen Systemen wird auch heute noch die «Prüftechnologie» verwendet, eine Methode, bei der alle möglichen Zustände durchprobiert und nach Fehlern abgefragt werden. Das Ergebnis ist die exakte Aussage: «fehlerfrei» oder «fehlerhaft». Bei

komplexen digitalen Systemen ist es jedoch überhaupt nicht möglich, alle denkbaren Zustände zu durchlaufen, da deren Zahl zu groß ist. Beispielsweise kann bereits ein sehr kleiner und einfacher Mikrorechner mit etwa 10 Kbyte Hauptspeicherkapazität theoretisch ungefähr 10^{3000} (eine Zahl mit dreitausend Nullen!) Zustände einnehmen. Hier setzt die technische Diagnose ein. Sie umfaßt die Methoden sowie Verfahren der Fehlererkennung und -benennung bei Systemen komplexer Natur. Es werden in der Regel nicht einzelne Fehler bzw. Fehlerzustände gesucht, sondern sogenannte *Fehlerkomplexe*. Zu ihrer Erkennung ist es erforderlich, das zu diagnostizierende System zu *testen*, die Merkmale zu *beurteilen* und auf dieser Grundlage die *Diagnose* zu stellen. Allerdings ist man im Ergebnis nie ganz sicher, ob ein einzelner Fehler überhaupt oder richtig erkannt wurde. Man bekommt nur die Aussage: System mit «großer Wahrscheinlichkeit fehlerfrei» oder «fehlerhaft».

In der Praxis gibt es zwei Diagnoseformen: die *Fremddiagnose* und die *Selbstdiagnose*. Bei ersterer wird das zu untersuchende System von einem anderen System, das z. B. im Fehlerfall anzuschließen ist, oder vom Menschen selbst diagnostiziert. Demgegenüber erfolgt im zweiten Fall eine selbsttätige, automatische Funktionsüberwachung und -analyse im eigenen System. Nur in Rechnersystemen mit dynamischer Redundanz ist eine Selbstdiagnose möglich. Sie ist die entscheidende Voraussetzung für Konstruktion und Aufbau fehlertoleranter Mikrorechner bzw. Mikrorechnerverbundsysteme.

8.5. Fehlerbehandlung und Instandsetzung

Ist ein Fehler erkannt und bewertet, folgt seine Behandlung, die primär zum Ziel hat, eine abweichende Systemfunktion zu normalisieren. In der praktischen Anwendung der Mikrorechentechnik geht man hierbei zwei Wege: Fehlertoleranz oder Instandsetzung. Bei der Fehlertoleranz wird das System in seinen Eigenschaften so verändert, daß bestimmte Fehler nicht mehr zur Auswirkung kommen. In der Instandsetzung handelt es sich dagegen um alle Maßnahmen der wirklichen Beseitigung der Fehlerzustände bzw. der fehlerhaften Komponenten. Hierzu werden für moderne Rechnersysteme effektive *Instandsetzungsregime* bereits bei Entwurf und Konstruktion des Rechners konzipiert. Derartige Verfahren ermöglichen vielfach die Schadenbeseitigung im normalen Betrieb des Rechners. Aber auch zur Methode der Fehlertoleranz gehört in gewissen zeitlichen Abständen eine Phase der Instandsetzung.

8.6. Technische Prophylaxe

Sie beinhaltet die Gesamtheit der Maßnahmen zur Verbesserung und Vorbeugung von Fehlerzuständen. Das Ziel besteht zum einen darin, sie ganz zu verhindern und zum anderen in einer frühstmöglichen Erkennung sowie Behandlung von Fehlern. Zur Früherkennung und -behandlung werden die Gesetzmäßigkeiten der *Fehlerprognostik* ausgenutzt. Sie haben Maßnahmen zur Bestimmung des voraussichtlichen Verlaufs und die Voraussage der zu erwartenden Fehlererscheinungen zum Inhalt.

Um Fehlerzustände überhaupt zu vermeiden, wird in der technischen Prophylaxe auch die sogenannte vorbeugende *Wartung und Instandsetzung* angewendet. Sie kann bei Rechnern periodisch, planmäßig oder auch situationsbezogen durchgeführt werden. Das Ziel aller dieser Maßnahmen ist die Erhöhung der Zuverlässigkeit und Verfügbarkeit des Rechnersystems.

9. Perspektiven der Mikrorechentechnik

In den vorhergehenden Abschnitten wurde versucht, die historischen und technologischen Wurzeln der Mikrorechentechnik aufzuzeigen, ihren gegenwärtigen Stand zu charakterisieren sowie insbesondere Aufbau, Funktionsweise und Programmierung eines Mikrorechners zu veranschaulichen. Dies war nicht ohne Bezug auf die klassische Rechentechnik möglich. Dadurch wurde sichtbar, daß die Mikrorechentechnik sich in das Gesamtgebäude der Theorie und Praxis informationsverarbeitender Anlagen und Einrichtungen folgerichtig einordnet und damit auf allen Erkenntnissen und Erfahrungen aus diesen Disziplinen aufbauen kann. Die weitere Entwicklung der Mikrorechentechnik wird wesentlich vom zukünftigen Geschehen auf dem Gebiet der Mikroelektronik abhängen. Alles spricht dafür, daß zunächst der Weg zu noch höherem Integrationsgrad sowie zu größeren Speicherdichten weiterbeschritten wird. Heute ist es bereits möglich, Mikroprozessoren zu produzieren, die einen Halbleiterspeicher von rund 50 Mbyte direkt ansprechen (adressieren) können. In naher Zukunft wird man noch riesigere Speichermengen zur Verfügung haben. Außerdem werden effektivere Herstellungstechnologien für hochintegrierte Schaltkreise entwickelt, und damit kann man die Produktion derartiger elektronischer Bauelemente noch ökonomischer gestalten. Der zukünftige Digitalrechner wird daher ein System aus sehr hochintegrierten Funktionsbausteinen, also im Prinzip ein Mikrorechner oder ein Mikrorechnerverbundsystem, sein.

Vielfach ordnet man die gegenwärtigen Entwicklungsetappen, wie Schaffung der Mikroprozessoren und Mikrorechner, den Übergang von zentralisierten informationsverarbeitenden Anlagen zu verteilten Systemen, den Aufbau von Rechnerverbundsystemen sowie das Verschmelzen von Rechentechnik und Nachrichtentechnik zu modernen Text-, Bild- und Sprachkommunikationssystemen den Merkmalen der *4. Rechnergeneration* zu. Ob diese Bezeichnung gerechtfertigt ist, werden die Menschen erst zu einem späteren Zeitpunkt entscheiden. Schon heute spricht man allerdings von einer *5. Generation* der infor-

mationsverarbeitenden Anlagen, die etwa ab 1990/1995 wirksam werden könnte. Sie wird gekennzeichnet sein durch vollkommen neue Wirkprinzipien der Erfassung, Verarbeitung und Darstellung von Informationen sowie von neuen Kommunikationsformen mit dem Rechner. Computer der 5. Generation werden «intelligenter» sein als die jetzigen Rechnersysteme. Der Mensch wird seinen Sinnen entsprechende Kommunikationsmöglichkeiten vorfinden, z.B. einen akustischen, bildlichen oder «fühlenden» (sensorischen) Informationsaustausch. Der Rechner wird also dem Menschen mehr «angepaßt». Geschriebene oder gesprochene Sprache, Bilder, Grafik lassen sich dann ebenso erfassen, speichern und manipulieren, wie heute alphanumerische Informationen. Zukünftige Rechner werden nicht mehr wie bisher Datenverarbeitungsanlagen, sondern «Wissensverarbeitungsanlagen» sein, und anstelle der heutigen Informationsbasis oder Datenbank wird eine «Wissensbank» für den Rechner die Grundlage seiner Entscheidungen und Problemlösungsstrategien bilden. Dem Menschen werden Maschinen mit «künstlicher Intelligenz» zur Seite stehen, die beispielsweise eine automatische Produktion von Gütern gestatten und bestimmte Entscheidungsfindungen selbst übernehmen. Während heute den Rechnern noch alle Aktionen exakt vorgegeben werden müssen, werden sie zukünftig nur noch ein grobes Gerüst der Aufgabenstellung benötigen, und ihre Intelligenz wird diese so vervollkommnen, daß sie exakt ausgeführt werden kann. Hierzu sind jedoch noch eine ganze Reihe von theoretischen und praktischen Erkenntnissen erforderlich, um die dabei ablaufenden Informationsprozesse besser verstehen und gestalten zu können.

Wichtige Fachbegriffe der Mikrorechentechnik

Addierer (adder) – Logische Grundschaltung zur Addition von binär codierten Zahlen

Adresse (adress) – Eindeutige Bezeichnung für Quelle, Zielort, Übertragungskanal oder Ort der Speicherung einer Information; auch binäres Wort oder Symbol zur Kennzeichnung eines Speicherplatzes, -bereiches oder einer Funktionseinheit des Rechners

Adreßbus – Leitungssystem zur Übertragung von Adressen im Mikrorechner

Akkumulator (accumulator) – Zentrales Register eines Digitalrechners; dient als Arbeits- oder Zwischenspeicher für arithmetische und logische Operationen

Aktion (action) – Allgemeiner Ausdruck für einen Vorgang oder einen Verarbeitungsschritt im Rechner

Aktive Bauelemente (active elements) – Elektronische Bauelemente mit der Eigenschaft zur Strom-, Spannungs- oder Leistungsverstärkung; benötigen Energiezufuhr (z.B. Röhren, Transistoren, Thyristoren, integrierte Schaltkreise)

Akzeptor (acceptor) – Chemische Elemente, die in das reine Halbleitermaterial eingebracht werden und dort «Löcher» (fiktive positive Ladungen) erzeugen. Erhöhung der Leitfähigkeit des Halbleitermaterials; es wird p-leitend

Algorithmus – Rechenvorschrift zur schrittweisen und eindeutigen Lösung eines Problems

Alphabet – Liste von vereinbarten Zeichen oder Symbolen

Alphanumerische Darstellung – Darstellung von Informationen durch Ziffern und Buchstaben

Analoges Signal – Signal, dessen Informationsparameter innerhalb festgelegter Grenzen jeden beliebigen Wert annehmen können

Anweisung – siehe Befehl

Arbeitsspeicher – Kurzzeitzwischenspeicher für Daten und Adressen

Arithmetik-Logik-Einheit (ALU – arithmetic logic unit) – Verarbeitungseinheit im Mikrorechner bzw. Mikroprozessor für die arithmetische und logische Verknüpfung der Daten

Assembler-Sprachen – Maschinenorientierte Programmiersprachen für Datenverarbeitungsanlagen bzw. Mikrorechner

Assemblierung – Übersetzung eines in Assembler-Sprache geschriebenen Programms in ein Maschinenprogramm mit Hilfe eines Rechners.

Assoziativspeicher (associative memory) – Speicher, in dem auf die Informationen nicht durch Angabe einer Adresse (Ort), sondern des Inhalts der gespeicherten Information zurückgegriffen wird
Ätzprozeß (etching) – Materialabtrag durch chemische Behandlung; technologischer Schritt bei der Herstellung von integrierten Schaltkreisen. Das Ätzen kann ganzflächig oder selektiv erfolgen, letzteres wird durch Masken gesteuert
Aufdampfprozeß (evaporation) – Technologischer Prozeßschritt bei der Herstellung integrierter Schaltkreise; dient zum Erzeugen von dünnen Schichten durch Verdampfen im Hochvakuum, entweder ganzflächig oder selektiv durch abdeckende Masken
Automat – Einrichtung, bei der Eingabe, Umformung, Transport und Ausgabe von Stoffen, Energien oder Informationen ohne unmittelbaren Einfluß des Menschen erfolgen

Baud – Einheit für die Schrittgeschwindigkeit bei der Informationsübertragung (**1 Baud** (Bd) = 1 Schritt/Sekunde). Im Falle eines binären Signals gilt 1 Bd = 1 bit/s
BCD-Code (binary coded decimal representation) – Binärcode bei Rechnern für die Darstellung von Dezimalziffern. Jede Ziffer 0 bis 9 einer Dezimalzahl wird 4stellig binär verschlüsselt
Befehl (instruction)– Anweisung; Instruktion; Information in Form einer elementaren Arbeitsanweisung für einen Rechner
Befehlszyklus (instruction cycle) – Arbeitsperiode des Mikroprozessors; umfaßt Holen, Decodieren und Ausführen eines Maschinenbefehls
Betriebssystem (operating system) – Zusammenfassung aller für die Steuerung der inneren Abläufe und Nutzung eines Rechners notwendigen Programmkomponenten zu einem zentralen Programmsystem
Bibliothek – Kurzform für Programmbibliothek; enthält die für den Rechner verfügbaren Programme
Binärcode (binary code) – Code, in welchem die Codeworte nur aus zwei unterschiedlichen Elementen (z.B. «0» oder «1») zusammengesetzt sind
Binärsignal (binary signal) – Signal, dessen Informationsparameter nur zwei Werte annehmen kann
Bipolare Schaltkreise – Integrierte Schaltkreise, die mit bipolaren Transistoren aufgebaut sind
Bipolartechnologie (bipolar technology) – Herstellungstechnologie von integrierten Schaltkreisen, deren Schaltungen vorwiegend bipolare Transistoren enthalten
Bipolartransistor (bipolar transistor) – Elektronisches Halbleiterbauelement, in dem am Stromfluß in den aktiven Zonen sowohl Löcher als auch Elektronen beteiligt sind
Bit, bit (binary digit) – Kurzform für die Bezeichnung eines Binärzeichens (Bit) sowie als Maßeinheit für die Informationsmenge (bit). Als Binärzeichen kann ein Bit nur zwei mögliche Werte (z.B. 0 und 1) annehmen. Als Maßeinheit entspricht 1 bit in der Informationstheorie der kleinsten Informationsmenge

BUS – Leitungsbündel (elektrische oder optische Leitungen) zur gleichzeitigen Übertragung von mehreren Informationsarten
BUS-System – BUS mit angeschlossenen Sende- und Empfangseinrichtungen und Regeln der Informationsübertragung
Byte, byte – Bezeichnung für die Zusammenfassung von 8 Binärstellen (8 bit) zu einem Binärwort (Byte). Üblicherweise ist bei Mikrorechnern 1 byte die kleinste adressierbare und verarbeitbare Informationseinheit

CAD (computer aided design) – Rechnergestützter Entwurf, Projektierung, Entwicklung, Konstruktion von technischen Systemen
CAM (computer aided manufacturing) – Rechnergestützte, rechnergeführte oder rechnergesteuerte Produktion, einschließlich Vorfertigung, Lagerhaltung u.ä.
Chip (chip) – Einzelne vollständig integrierte Schaltung bzw. einzelnes Halbleiterscheibchen aus einem Scheibenverband
CMOS (complementary MOS) – Herstellungstechnologie von integrierten Schaltkreisen, die sowohl p-Kanal- als auch n-Kanal-Transistoren enthalten
CNC-Maschine (computerized numerical control machine) – Numerische Steuerung einer Maschine mit Hilfe eines Rechners
Code (code) – Zuordnungsvorschrift zwischen den Elementen einer Menge von Zeichen (Codeworte) und den zu codierenden Einzelinformationen, oder Zuordnung zwischen zwei Mengen von Zeichen oder Symbolen. Wichtigster Code in der Mikrorechentechnik ist der Binärcode
Codierung – Verschlüsselung; Darstellung von Informationen durch digitale Größen oder digitale Signale
Compiler – Übersetzer bzw. Übersetzungsprogramm von einer höheren Programmiersprache in eine Maschinen- oder maschinennahe Sprache
Computer – Rechenautomat; Digitalrechner
CPU (central processing unit) – Mikroprozessor

D/A-Wandler (D/A-converter) – System oder Schaltkreis zum Umsetzen von digitalen Signalen oder Werten in analoge
Datei – Kurzbezeichnung für eine Menge von Einzeldaten oder Datenbeständen; enthält die für den Rechner verfügbaren Daten, Datenfelder u.ä.
Daten – Numerische oder alphanumerische Angaben über Sachverhalte, Vorgänge, Dinge oder Verarbeitungsgrößen im Digitalrechner; maschinell darstellbare Größen
Datenbank – Logisch geordnete und nach systematischen Regeln abgespeicherte Informationen
Datenbus – Leitungssystem zur Übertragung von Datenworten im Mikrorechner
Datenflußplan – Grafische Darstellung der zeitlichen oder logischen Folge einzelner Transport- und Verarbeitungsschritte der Daten in einem Rechner bzw. von einem Programm
Datenregister – Register zum Zwischenspeichern von Daten
Decodierung – Umkehrung der Codierung

Digitales Signal (digital signal) – Diskretes Signal, in dem die Werte des Informationsparameters Worten entsprechen
Digitaltechnik – Technische Einrichtungen oder Verfahren, die auf der Grundlage digitaler Signale und Systeme arbeiten
Diskretes Signal – Signal, dessen Informationsparameter nur bestimmte, «diskrete» Werte einer endlichen Menge annehmen können

Editor – Kurzbezeichnung für Korrekturprogramm; dient dem Eingeben, Ausgeben, Ändern, Modifizieren von Programmen bzw. Programmteilen, Daten und Texten
Empfänger – Einrichtung, die das Empfangssignal in das ursprüngliche Sendesignal oder in eine gewünschte Informationsart umwandelt
EPROM (erasable programmable read-only memory) – Bezeichnung für löschbaren und mehrfach programmierbaren Nur-Lese-Speicher bzw. entsprechenden Speicherschaltkreis
Externer Speicher – Speichereinheiten eines Rechners, auf die nicht direkt, sondern über Ein- oder Ausgabeoperationen zugegriffen werden kann

Fehlertoleranz – Eigenschaft eines Programms, Rechners oder allgemein eines Systems; sie bleiben bei auftretenden Fehlerzuständen arbeitsfähig
Feldeffekttransistor (field effect transistor) – Halbleiterbauelement, dessen Wirkungsprinzip auf einem Feldeffekt, also durch Influenz von Ladungsträgern beruht; gewöhnlich auch als MOS-Transistor bezeichnet
Flip-Flop – Zweiwertiges (binäres) Speicherelement zum Abspeichern von 1 bit
File (file) – Datei, Bibliothek oder deren Elemente
Firmware (firmware) – Vom Hersteller eines Mikrorechnersystems mitgelieferte Software
Folienspeicher – Externer Speicher mit auswechselbarem Datenträger (Diskette) auf magnetischer Basis, auch mit Floppy disk bezeichnet

Glied – Teil- oder Elementarsystem; Objekt in einem Abschnitt des Wirkungswegs eines Systems; kann im allgemeinen Sinn wieder als System aufgefaßt werden

Halbleiterspeicher – Programm- oder Datenspeicher, der mit Hilfe von hochintegrierten Halbleiterschaltkreisen realisiert ist
Hardware – Bezeichnung für alle technischen und elektronischen Teile sowie Einrichtungen eines Rechners

Indexregister – Register, in dem ein Adreßwert oder Adreßdifferenzwert zwischengespeichert oder aufbereitet werden kann
Informatik – Lehre von der Verarbeitung, Speicherung und Darstellung von Informationen sowie den dazu erforderlichen Verfahren und Einrichtungen
Informationsparameter – Parameter eines Signals, der Informationen enthält

Informationstheorie – Lehre von den Informationen, deren mathematischer Beschreibung und Messung
Instruktion – siehe Befehl
Intelligentes Terminal – Terminal, in dem zusätzlich Verarbeitungsprozesse von Daten ablaufen
Interaktive Betriebsweise – Dialog mit dem Rechner; Möglichkeit der unmittelbaren Beeinflussung des Rechners bzw. der Programme durch den Nutzer
Interrupt – Unterbrechungssignal von externen Ereignissen für den Rechner; unterbricht in der Regel ein laufendes Programm

Kellerprinzip – Verfahren zur Abspeicherung von Größen in einem Kellerspeicher; hierbei werden die zuletzt abgespeicherten («eingekellerten») Größen bei einer Entspeicherung wieder zuerst ausgelesen («entkellert»)
Kellerspeicher (stack) – Speicherbereich in einem Schreib/Lese-Speicher, in dem Daten oder Adressen nach dem Kellerprinzip zwischengespeichert oder «gerettet» werden können

Magnetbandspeicher – Externe Speichereinheit mit auswechselbaren Datenträgern (Magnetbandspule oder Magnetbandkassette)
Makroassembler – Ein Übersetzungsprogramm, das Makrobefehle verarbeiten kann
Makrobefehl (macroinstruction) – Gruppe von Befehlen, die zu einem neuen Befehl zusammengefaßt werden; ein Makrobefehl (Makro) wird wie ein einzelner Befehl behandelt
Maschinenbefehl – Elementare Steueranweisung für einen Prozessor oder Mikroprozessor; besteht im allgemeinen aus Instruktions- (Befehls- oder Operationscode) und Adreßteil (Adreßcode)
Maschinenprogramm – Programm in Maschinensprache; besteht aus den binären Befehlen und binär dargestellten Daten und kann vom Rechner direkt abgearbeitet werden
Mikrocomputer – siehe Mikrorechner
Mikroprogramm – Programm einer Mikroprogrammsteuerung
Mikroprogrammsteuerung – Programmierbare Ablaufsteuerung in einem Rechner; arbeitet nach dem Prinzip eines Digitalrechners
Mikroprozessor (central processing unit) – Hochintegrierter digitaler Schaltkreis, der zentrale Verarbeitungseinheit enthält. Wichtige Komponenten sind Befehlsinterpretationssystem (Befehlsdecodierung), Verarbeitungssystem (Arithmetik-Logik-Einheit) und Zwischenspeicher (Registersatz)
Mikrorechner (microcomputer) – Digitalrechner aus einem oder mehreren hochintegrierten digitalen Schaltkreisen
MOS-Technik (metall-oxid-semiconductor technology) – Herstellungstechnologie von integrierten Schaltkreisen, deren Schaltungen vorwiegend MOS-Transistoren enthalten
MOSFET – MOS-Feldeffekt-Transistor
MOS-Transistor (metall-oxid-semiconductor transistor) – Unipolärer Transistor, der nach dem Prinzip des Feldeffekttransistors arbeitet

Multisystem – Zusammenschluß mehrerer Systeme, z.B. Rechnerverbundsystem

NC-Maschine (numerical control machine) – Numerisch gesteuerte Maschine, z.B. bei Werkzeugmaschinen

OEM-Einheiten (original equipment manufactures) – Vom Halbleiterbauelementehersteller oder Rechnerhersteller gelieferte Halbfertigprodukte, z.B. bestückte Leiterplatten, Rechnereinheiten; sind für den Einbau in anderen Erzeugnissen gedacht
Operation – Bezeichnung für Befehlsabarbeitung oder Verknüpfungsvorschrift von Größen

Programm – Geordnete Menge von Befehlen, (Instruktionen, Anweisungen) zur Abarbeitung eines Algorithmus
Programmablaufplan – Grafische Darstellung aller möglichen Ablaufwege und der dabei durchzuführenden Operationen
Programmiersprachen – Künstliche Sprachen zur Formulierung von Programmen; man unterscheidet Maschinen-, maschinenorientierte und höhere Programmiersprachen
PROM (programmable read only memory) – Bezeichnung für programmierbaren Nur-Lese-Speicher bzw. entsprechenden Speicherschaltkreis
Prozeß – Vorgang der Umformung oder Transport von Stoffen, Energien oder Informationen; auch im Rechner ablaufender Vorgang

RAM (random-access memory) – Bezeichnung für Schreib-Lese-Speicher oder entsprechenden Speicherschaltkreis
Retteverfahren – Zwischenspeicherung von Daten, Adressen in einem Rettebereich (meist ein Kellerspeicher) bei Auftreten einer Unterbrechung (Interruptsignal) eines laufenden Programms
ROM (read only memory) – Bezeichnung für Nur-Lese-Speicher oder entsprechenden Speicherschaltkreis, auch Festwertspeicher

Sender – Einrichtung, die ein zu übertragendes Signal an den Übertragungskanal anpaßt und in diesen einspeist
Signal – Darstellung von Informationen durch physikalische Größen; dient zur Informationsübertragung
Software – Bezeichnung für alle programmtechnischen Teile, Informationen und Dokumentationen eines Rechners
Stack – siehe Kellerspeicher
Steuerbus – Leitungssystem zur Übertragung von binären Steuerworten im Mikrorechner
Steuerung – Vorgang der zielgerichteten Beeinflussung von Größen eines Systems oder einer Einrichtung sowie von Prozessen
System – Eine Menge von Gliedern, die selbst Systemcharakter tragen können und durch Kopplungen miteinander verbunden sind

Systemanalyse – Theorie zur Beschreibung von bekannten oder angenommenen Systemen (Modellen) durch Kennfunktionen
Systemsynthese – Theorie zum Entwurf von Systemen auf der Grundlage von vorgegebenen Eigenschaften, Forderungen und aufgestellten mathematischen Beschreibungen
Systemtheorie – Lehre vom Verhalten der Systeme, deren mathematische Beschreibung und entsprechende Untersuchungsmethoden

Terminal – Ein- und Ausgabeeinheit am Rechner, Rechnerverbundsystem oder Rechnernetz für den Nutzer, gewöhnlich auf der Grundlage eines Bildschirms

Übertragungssystem – Gesamtheit aller Mittel, die eine Übertragung des Sendesignals bewirken, u.a. Sender, Übertragungskanal und Empfänger
Unipolare Schaltkreise – Integrierte Schaltkreise, die mit unipolaren (MOS-) Transistoren aufgebaut sind
Unipolartechnik (unipolar technology) – Herstellungstechnik von integrierten Schaltkreisen, deren Schaltungen vorwiegend Unipolartransistoren (MOS-Transistoren) enthalten, auch als MOS-Technik bezeichnet
Unipolartransistor (unipolar transistor) – Halbleiterbauelement, in dem der Stromfluß in den aktiven Zonen durch eine Ladungsträgerart (Löcher oder Elektronen) erfolgt; auch als MOS-Transistor oder Feldeffekttransistor bezeichnet

Wort (word) – Geordnete Folge von Zeichen eines Alphabets, die in ihrer Gesamtheit eine Information enthält

Zeigerregister – Register im Mikroprozessor, in dem ein Adreßwert eines bestimmten Speicherplatzes, die Zeigeradresse, zwischengespeichert oder aufbereitet werden kann
Zustand – Gesamtheit der Werte von Größen, die ein System zu einem bestimmten Zeitpunkt enthält

Sachwortverzeichnis

Literaturverzeichnis

[1] Paul, R.: Mikroelektronik – Eine Übersicht. – Berlin: Verl. Technik, 1981
[2] Mikroelektronik – Stand und Entwicklung. – Berlin: Akademie-Verl., 1981
[3] Möschwitzer, A.: Integration elektronischer Schaltungen. – Berlin: Verl. Technik, 1974
[4] Möschwitzer, A.; Lunze, K.: Halbleiterelektronik. – Berlin: Verl. Technik, 1979
[5] Rumpf, K.-H.; Pulvers, M.: Transistorelektronik. – Berlin: Verl. Technik, 1979
[6] Leonhardt, E.: Grundlagen der Digitaltechnik. – Berlin: Verl. Technik, 1976
[7] Völz, H.: Information I, II. – Berlin: Akademie-Verl., 1982
[8] Fey, P.: Informationstheorie. – Berlin: Akademie-Verl., 1963

[9] Woschni, E.-G.: Informationstechnik – Signal, System, Information. – Berlin: Verl. Technik, 1981
[10] Lerner, A. J.: Grundzüge der Kybernetik. – Berlin: Verl. Technik, 1970
[11] Unbehauen, R.: Systemtheorie. – Berlin: Akademie-Verl., 1970
[12] Matschke, J.: Von der einfachen Logikschaltung zum Mikrorechner. – Berlin: Verl. Technik, 1981
[13] Nührmann, D.: Schlüssel zum Mikrocomputer. – München: Franzis-Verl., 1980
[14] Jugel, A.: Mikroprozessorsysteme. – Berlin: Verl. Technik, 1978
[15] Osborne, A.: Einführung in die Mikrocomputertechnik. – Berlin: Verl. Technik, 1983
[16] Schwarz, W.; Meyer, G.; Eckhardt, D.: Mikrorechner – Wirkungsweise, Programmierung, Applikation. – Berlin: Verl. Technik, 1980
[17] Paulin, G.: Kleines Lexikon der Mikrorechentechnik. Reihe Automatisierungstechnik, Nr. 206. – Berlin: Verl. Technik, 1983
[18] Barthold, H.; Bäurich, H.: Mikroprozessoren – Mikroelektronische Schaltkreise und ihre Anwendung. Amateurreihe «elektronika» Nr. 186, 187, 188. – Berlin: Militärverl., 1980
[19] Kieser, H.; Meder, M.: Mikroprozessortechnik. – Berlin: Verl. Technik, 1982
[20] Lampe, B.; Jorke, G.; Wengel, N.: Algorithmen der Mikrorechentechnik. – Berlin: Verl. Technik, 1983
[21] Herrlich, O.; Lindner, U.: Strukturierte Programmierung. – Leipzig: BSB B. G. Teubner Verlagsgesellschaft, 1981
[22] Gluschkow, W. M.; Zeitlin, G. J.; Justschenko, J. L.: Algebra, Sprachen, Programmierung. – Berlin: Akademie-Verl., 1980
[23] Paulin, G.; Schiemangk, H.: Programmieren mit PASCAL. – Berlin: Akademie-Verl., 1981
[24] Werner, D.: Programmierung von Mikrorechnern. – Berlin: Verl. Technik, 1983
[25] Reinert, D.: Prüftheorie diskreter Systeme. – Berlin: Verl. Technik, 1979
[26] Reinert, D.: Entwurf und Diagnose komplexer digitaler Systeme. – Berlin: Verl. Technik, 1983

VEB Fachbuchverlag Leipzig

Automatische Prozesse in Produktion und Verwaltung, mikrorechentechnische Durchdringung ganzer Industriekomplexe, Taschenrechner, Heim- und Lerncomputer, programmierbare Haushaltgeräte – das alles sind technische Errungenschaften, mit denen wir zunehmend täglich konfrontiert werden.

Wie ist ein Mikrorechner aufgebaut? Welche Prozesse laufen in ihm ab? Wie wird ein Programm erarbeitet? Was sind Hardware und Software? Welche Anwendungsbereiche findet die Mikrorechentechnik?

Diese und zahlreiche andere Fragen beantwortet der Autor für alle interessierten Leser. Wichtige Fachbegriffe zur Mikrorechentechnik erläutert er in einem Anhang.

Gut handhabbar und allgemeinverständlich geschrieben, ermöglicht dieses Buch einen schnellen Überblick zu dieser hochaktuellen Thematik.