

Datenverarbeitungssysteme

Table of Contents

<u>Einführung Datenverarbeitungssysteme</u>	1
<u>Inhalt</u>	1
1. Einführung.....	1
2. Aufbau und Arbeitsweise einer Datenverarbeitungsanlage.....	1
3. Informationsdarstellung in Rechenanlagen.....	1
4. Rechenwerk.....	1
5. Befehle (Maschinensprache).....	1
6. Leitwerk.....	1
7. Speicherwerk (Arbeitsspeicher).....	2
8. Rechnerperipherie.....	2
Skript der früheren Vorlesung Rechnerperipherie (Prof. Dr. Vollmann).....	2
Download des gesamten Skripts.....	2
Literatur:.....	3
Links zum Thema PC-Hardware.....	3
Einführung Datenverarbeitungssysteme.....	4
<u>1. Einführung</u>	4
1.1 Grundbegriffe.....	4
DVS (Datenverarbeitungssystem):.....	4
Computer.....	5
RISC "Reduced Instruction Set Computer".....	6
Informationseinheiten.....	6
1.2 Überblick der Entwicklungsgeschichte der Datenverarbeitung.....	8
Buchtips:.....	8
Einführung Datenverarbeitungssysteme.....	9
<u>2. Aufbau und Arbeitsweise einer Datenverarbeitungsanlage</u>	9
2.1 Hardware & Software.....	9
2.2 "Klassische" Rechnerarchitektur.....	11
Operationsprinzip.....	12
2.3 Innovative Rechnerarchitekturen.....	12
Erhöhung der Leistungsfähigkeit des v. Neumann-Rechners.....	13
2.4 Befehlszyklus.....	13
Ablauf des Befehlszyklus.....	15
2.5 Programm-Unterbrechungen.....	16
Die HTML-Fassung entstand unter Mitwirkung von Volker Arndt Copyright © FH	
München, FB 04, Prof. Jürgen Plate.....	16
Einführung Datenverarbeitungssysteme.....	17
<u>3. Informationsdarstellung in Rechenanlagen</u>	17
3.1 Grundlagen der Zahlensysteme.....	17
Definitionen.....	17
3.2 Stellenwertsysteme.....	17
Ganze Zahlen:.....	18
Dezimalzahlen:.....	18
Dualzahlen.....	18
Oktalzahlen.....	18
Sedezimalzahlen (= Hexadezimalzahlen).....	18
3.3 Umrechnung zwischen den Stellenwertsystemen.....	18
Umwandlung Dezimalsystem anderes Stellenwertsystem.....	20
Umwandlung anderes Stellenwertsystem Dezimalsystem.....	21

Table of Contents

3. Informationsdarstellung in Rechenanlagen	
<u>Umwandlung Dual - Oktal - Hexadezimal</u>	21
3.4. <u>Arithmetik im Dualsystem</u>	22
<u>Arithmetische Grundoperationen</u>	22
<u>Darstellung negativer Zahlen</u>	24
<u>Komplementbildung im Dualsystem</u>	26
<u>Multiplikation</u>	27
<u>Division</u>	28
3.5 <u>Festpunkt- und Gleitpunktdarstellung</u>	28
<u>Festpunktzahlen</u>	28
<u>Gleitpunktzahlen</u>	30
<u>Gleitpunkt-Arithmetik</u>	31
<u>Beispiel für Gleitpunkt-Formate</u>	31
3.6 <u>Informationsdarstellung in Rechenanlagen</u>	31
<u>Integer-Zahlen</u>	31
<u>Real-Zahlen</u>	31
<u>BCD - Zahlen</u>	32
<u>Zeichen</u>	33
<u>Logische Werte</u>	34
<u>Einführung Datenverarbeitungssysteme</u>	35
4. Rechenwerk	35
<u>Binäre Addierschaltung</u>	35
4.1 <u>Arithmetisch-Logische Einheit (ALU)</u>	36
<u>Einfache ALU</u>	37
4.2 <u>ALU mit Registern</u>	38
<u>Blockschaltung einer Register-ALU</u>	39
4.3 <u>Realisierung von Multiplikation und Division</u>	39
<u>Schema der Multiplikation</u>	39
4.4 <u>Ergänzungen zur RALU</u>	40
<u>Die HTML-Fassung entstand unter Mitwirkung von Volker Arndt Copyright © FH</u>	
<u>München, FB 04, Prof. Jürgen Plate</u>	40
<u>Einführung Datenverarbeitungssysteme</u>	41
5. Befehle (Maschinensprache)	41
5.1. <u>Befehlsaufbau und Befehlsformate</u>	41
<u>Vereinfachung der binären Darstellung</u>	42
5.2 <u>Befehlsklassen, Befehlslisten</u>	42
<u>Typische Befehlsklassen</u>	43
5.3 <u>Adressierungsarten</u>	43
<u>Programmiermodell</u>	44
<u>Absolute Adressierung</u>	44
<u>Seitenbezogene Adressierung</u>	45
<u>Indizierte Adressierung</u>	45
<u>Indirekte Adressierung</u>	46
<u>Unmittelbare Adressierung (Immediate Addressing)</u>	46
<u>Implizite Adressierung (Inherent Addressing)</u>	47
<u>Einführung Datenverarbeitungssysteme</u>	48
6. Leitwerk	49
6.1 <u>Ablaufsteuerung und Befehlsdecoder</u>	51
6.2 <u>Unterbrechungs-Behandlung</u>	55

Table of Contents

6. Leitwerk

6.3 Bus-System und Bussteuerung.....	57
Anhang: Einfache Beispiele zur Ablaufsteuerung.....	58
1. Abfüllanlage.....	60
Einführung Datenverarbeitungssysteme.....	61

7. Speicherwerk (Arbeitsspeicher).....61

7.1 Speicherhierarchie.....	62
7.2 Speicherarten.....	63
Speicherzellen sind meist als Matrix angeordnet.....	65
7.3 Arbeitsspeichermedien.....	66
Halbleiter-Schreib-Lese-Speicher.....	69
7.3.3 Halbleiter-Festwertspeicher.....	73
7.4 Organisation des Speicherwerks.....	73
7.4.1 Blockweise Organisation des Arbeitsspeichers.....	74
7.4.2 Speicherverschränkung = Adressverschränkung = "memory interleave".....	75
7.5 Pufferspeicher (Cache).....	76
7.6 Neue Speichermedien.....	76
Memory Stick.....	77
Einführung Datenverarbeitungssysteme.....	78

8. Rechnerperipherie.....78

8.1 Methoden des Datentransfers.....	78
8.1.1 Programmierter E/A-Transfer.....	80
8.1.2 Direktspeicherzugriff (DMA = Direct Memory Access).....	82
8.2. Periphere Speicher (Massenspeicher).....	82
8.2.1 Speicherprinzip magnetomotorischer Speicher.....	83
8.2.2 Magnetplattenspeicher.....	88
8.2.3 Magnetbandspeicher.....	88
8.2.4 Optische Speicherplatten.....	93
8.3 Periphere Geräte.....	93
8.3.1 Datensichtgerät und Tastatur.....	96
8.3.2 Drucker.....	101
8.3.3 Plotter.....	102
8.3.4 Lochkarten- und Lochstreifengeräte.....	102
8.3.5 Joysticks (grafische Eingabe).....	103
8.3.6 Trackball und Maus.....	103
8.3.7 Digitalisierer.....	104
8.3.8 Scanner (Abtaster).....	107
8.4 Audiotechnik im Computer und MIDI.....	107
8.4.1 Elektroakustik.....	110
8.4.2 Klang- und Tonerzeugung.....	115
8.4.3 Digitalisierung von Tönen.....	118
Hardware zur Aufnahme und Wiedergabe von Tönen.....	118
8.4.4 MIDI.....	title

Inhalt

1. Einführung

1. Grundbegriffe
2. Überblick der Entwicklungsgeschichte der Datenverarbeitung

2. Aufbau und Arbeitsweise einer Datenverarbeitungsanlage

1. Hardware und Software
2. "Klassische" Rechnerarchitektur
3. Innovative Rechnerarchitekturen
4. Befehlszyklus
5. Programmunterbrechung

3. Informationsdarstellung in Rechenanlagen

1. Grundlagen der Zahlensysteme
2. Stellenwertsysteme
3. Umrechnung zwischen den Stellenwertsystemen
4. Arithmetik im Dualsystem
5. Festpunkt- und Gleitpunktdarstellung
6. Informationsdarstellung

4. Rechenwerk

1. Arithmetisch-Logische Einheit (ALU)
2. ALU mit Registern
3. Realisierung von Multiplikation und Division
4. Ergänzungen zur RALU

5. Befehle (Maschinensprache)

1. Befehlsaufbau und Befehlsformate
2. Befehlsklassen, Befehlslisten
3. Adressierungsarten

6. Leitwerk

1. Ablaufsteuerung und Befehlsdecoder
2. Unterbrechungs-Behandlung
3. Bus-System und Bussteuerung
4. Anhang: Einfache Beispiele zur Ablaufsteuerung

7. Speicherwerk (Arbeitsspeicher)

1. Speicherhierarchie
2. Speicherarten
3. Arbeitsspeichermedien
4. Organisation des Speicherwerks
5. Pufferspeicher (Cache)

8. Rechnerperipherie

1. Methoden des Datentransfers
2. Periphere Speicher (Massenspeicher)
3. Periphere Geräte
4. Audiotechnik im Computer und MIDI

Skript der früheren Vorlesung Rechnerperipherie (Prof. Dr. Vollmann)

1. Teil 1: Datenpuffer, PCI, SCSI, USB
2. Teil 2: Plattenspeicher, RAID
3. Teil 3: Warteschlangen

Download des gesamten Skripts

Literatur:

Ulrich Appel:
Mikrocomputer und Minicomputer
Oldenbourg Verlag

Rechenberg/Promberger:
Informatik-Handbuch
Hanser Verlag

K.-U. Witt
Elemente des Rechneraufbaus
Hanser Verlag

Rainer Kelch:
Rechnergrundlagen - Von der Binärlogik zum Schaltwerk
Fachbuchverlag Leipzig

Rainer Kelch:
Rechnergrundlagen - Vom Rechenwerk zum Universalrechner
Fachbuchverlag Leipzig

H.J.Tafel/A.Kohl:
Ein- und Ausgabegeräte der Datentechnik
Hanser Verlag

Messmer:
PC-Hardware
Verlag Addison-Weley

H. Bähring:
Mikrorechner-Systeme
Springer-Verlag

W. Ameling:
Digitalrechner, Grundlagen und Anwendungen
Vieweg Verlag

B. Bundschuh und P. Sokolowsky:
Rechnerstrukturen und Rechnerarchitekturen
Vieweg-Verlag

W. Oberschelp, G. Vossen:
Rechneraufbau und Rechnerstrukturen
Oldenbourg Verlag

Links zum Thema PC-Hardware

<http://www.nickles.de>
<http://www.pctechguide.com>
<http://www.epanorama.net/links/pc/>
<http://users.erols.com/chare/hardware.htm>
<http://www.karbosguide.com/>
http://dir.yahoo.com/Computers_and_Internet/Hardware/

Die HTML-Fassung entstand unter Mitwirkung von Volker Arndt

Copyright © FH München, FB 04, Prof. Jürgen Plate

Letzte Aktualisierung: 15. Oct 2007



Einführung Datenverarbeitungssysteme

von Prof. Jürgen Plate

1. Einführung

1.1 Grundbegriffe

DVS (Datenverarbeitungssystem):

Programmgesteuerte digitale Systeme zur numerischen und nichtnumerischen Datenverarbeitung = "Computersysteme". Hier: nur Systeme mit flexibler Programmsteuerung, d.h. programmierbare Systeme = Universalrechner.

Im Zusammenhang mit Datenverarbeitung versteht man unter einem "System" praktisch immer die Kombination aus "Hardware" (DV-Anlage, Geräte) und "Software" (darauf ablaufende Programme)!

Computer

Die Wirkungsweise der Rechenmaschine besteht darin, daß die mechan. Addition oder Subtraktion zweier Zahlen mittels einer einzigen Drehung einer Handkurbel bewirkt wird, indem eine Anzahl von Scheiben um je einen den Ziffern der Rechnung entsprechenden Winkel gedreht werden; der Mechanismus ist derart eingerichtet, daß, wenn die Scheiben die Lagen 0—9 oder 9—0 überschreiten, ein Weiterdrehen der diesen letztern Scheiben folgenden (höhern) stattfindet. Dieses Princip lag schon den sinnreichen ältern Konstruktionen zu Grunde, an deren Vervollkommenung berühmte Gelehrte, wie Pascal, Leibniz, Poleni, Leupold, gearbeitet haben. Neuere Systeme sind die R. von Hahn, Müller, Thomas, Roth, Scheuch, Diehschold, Selling, Gutbier, Odhner.

Rechnen heißt, gegebene Größen nach gewissen Regeln miteinander verbinden oder voneinander trennen, um eine unbekannte Größe zu finden.

Brockhaus' Konversations-Lexikon. 14. Auflage 1894

Computer

Nach Abmessung, Preis und (bedingt) Leistungsfähigkeit unterscheidet man zwischen:

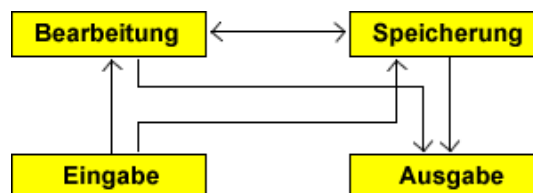
- **Groß-Computer (DV-Anlagen, "Mainframe"):** Rechner für große wissenschaftliche und kommerzielle Rechenzentren, die in der Regel in klimatisierten Räumen stehen müssen und eine eigene Stromversorgung besitzen.
- **Mini-Computer:** Laborrechner, Prozessrechner für wissenschaftliche und technische Anwender (z.B. auch NC-Steuerungen) oder Systeme der mittleren Datentechnik (MDT) im kommerziellen Bereich.
- **Mikro-Computer:** Sehr kompakte Systeme auf der Basis integrierter Steuer- und Verarbeitungsbausteine (Mikroprozessoren) und Speicher (VLSI und ULSI-Bausteine) für allgemeine und spezielle DV-Anwendungen (z. B. Steuerung, Regelung):
 - ♦ Tisch-/Personal-Computer (PC),
 - ♦ Homecomputer,
 - ♦ Einplatinen-Computer,
 - ♦ Ein-Chip-Computer.

Die Grenzen sind nicht eindeutig, sondern unscharf und ständigen Veränderungen unterworfen. Mikrocomputer reichen in der Leistung immer mehr an Minis heran und diese an die Großrechner ("Superminis").

Die Grundprinzipien der Struktur und die Arbeitsweise sind bei allen DV-Systemen weitgehend ähnlich, zum Teil sogar gleich. Auch komplexe Strukturen der Großrechner werden zunehmend auf Mini- und Mikrocomputer übertragen. Gleichzeitig entwickelt sich eine neue Klasse oberhalb der Großrechner, die "Supercomputer" (z.B. Cray, Suprenum).

Dieses Skript beschränkt sich auf DVS mit "klassischer" Struktur (*siehe später*). Auf Systeme mit neuen komplexeren Strukturen (Feldrechner, Mehrprozessor-Systeme, Transputer, Rechnernetze, etc.) kann nur andeutungsweise eingegangen werden.

Ursprünglich dienten Computer als Hilfsmittel zum Rechnen, wurden also in der numerischen DV eingesetzt (selbst der Name "Computer" stammt von einer Berufsbezeichnung her; "Computer" waren Leute, die z.B. in der Astronomie, bei Volkszählungen oder in Versicherungen Berechnungen durchführten). Inzwischen hat der Einsatz von nichtnumerischen Daten fast höhere Bedeutung erlangt (z.B. Textverarbeitung, Datenbanken ® Handhaben von Datenstrukturen). Durch geeignete Codierung lassen sich nichtnumerische Informationen genau wie Zahlen auf binäre Werte (Bitfolgen) abbilden. ® Erweiterung der "Rechner" auf universelle DV-Systeme. Die Grundfunktionen eines Computers (= DVS) sind



von Informationen. Je nach Anwendung stehen eine oder mehrere dieser Aufgaben im Vordergrund.

Grundsätzlich gilt: Auch bei der nichtnumerischen DV wird gerechnet (ausführen von logischen und arithmetischen Operationen) und auch bei der numerischen DV gibt es Datenstrukturen. Die Bereiche sind nicht zu trennen.

RISC "Reduced Instruction Set Computer"

(Computer mit eingeschränktem Befehlssatz)

Hat man anfangs versucht, beim Design neuer Prozessor-Chips immer mehr und immer leistungsfähigere Befehle zu integrieren, so wird beim RISC-Prozessor der umgekehrte Weg beschritten.

Für jeden Befehl des Prozessors gibt es eine fest verdrahtete Folge von Ablaufschritten im Chip, das Mikroprogramm. Je komplexer und mächtiger ein Befehl ist, desto mehr Einzelschritte muss das Mikroprogramm auf den Chip durchlaufen und desto mehr Taktzyklen sind notwendig, bis der Prozessor den nächsten Befehl verarbeiten kann. So liegt z.B. beim Prozessor 8088 die Zahl der Taktzyklen für einen Befehl zwischen 2 und 190.

RISC-Prozessoren haben einen kleinen Befehlssatz, bei dem aber fast alle Befehle innerhalb eines einzigen Taktzyklus ausgeführt werden können. Das bedeutet auch eine Vereinfachung des Chips und erreicht auf diese Weise Taktfrequenzen von mehr als 60 MHz.

Informationseinheiten

- **Bit (binary digit):**

(Dualziffer, Binärziffer)

Bit ist die Kurzform für Binärziffer und wird auch als Kurzform für Dualziffer verwendet. Ein

Bit (groß geschrieben) ist die kleinste Darstellungseinheit für Binärcodes und binäre Daten (Binärzeichen).

Bit ist die Maßeinheit für die Anzahl der Binärentscheidungen. In der Informationstheorie werden alle logarithmisch definierten Größen, wie Entscheidungsgehalt, Informationsgehalt, Redundanz in **bit** (klein geschrieben!) ermittelt, wenn der Logarithmus zur Basis 2 (Logarithmus Dualis, Kurzform *ld*) genommen wird.

- **Byte:**

Eine Folge von zusammen betrachteten acht Binärziffern, die zur Darstellung eines Zeichens im Rechner verwendet werden können. Die Codierung der Zeichen kann auf verschiedene Weise erfolgen.

Die Arbeitsspeicher der heutigen Rechner (vor allem der Mikrocomputer) sind meist byteweise organisiert, d.h. Transporte vom und zum Speicher erfolgen byteweise parallel oder parallel in Vielfachen eines Bytes (16 oder 32 Bit @ Speicherwort).

- **Wort:**

Eine Folge von Zeichen, die in einem bestimmten Zusammenhang als eine Einheit betrachtet wird. Ein wichtiger Zusammenhang, in dem dieser Begriff verwendet wird, ist der des Speicherwortes (Speicherstelle). Enthält das Speicherwort einen Befehl, so spricht man von einem Befehlswort.

1.2 Überblick der Entwicklungsgeschichte der Datenverarbeitung

Viele der heutigen Entwicklungen lassen sich auf den entwicklungsgeschichtlichen Prozess zurückführen. So sind viele der heutigen Rechnerarchitekturen aus der Geschichte leichter zu verstehen.

ab 350 v. Chr.	Erste vollständige Zahlensysteme (Ägypten, Babylonien)
um 300 v. Chr.	Erste bezeugte Verwendung des Rechenbretts (Abakus)
600-800 n. Chr.	Entstehung des heutigen Dezimalsystems in Indien
1518	Adam Riese (1492-1559): "Rechnen auff den Linihen"
1623	Wilhelm Schickard (1592-1635) baut erste mechanische Rechenmaschine (4-spezies, automatischer Zehnerübertrag)
1642	Blaise Pascal (1623-1662) entwirft mechanische Rechenmaschine für Addition und Subtraktion (Komplementaddition)
1673	Gottfried Wilhelm v. Leibnitz (1646-1716) entwickelt eine mechanische Rechenmaschine (4-spezies, Staffelwalze); nicht funktionsfähig.
1679	Schaffung des dualen Zahlensystems durch Leibnitz.

Datenverarbeitungssysteme

1709	Johannes Polenius baut eine mechanische Rechenmaschine mit Sprossenrad (nicht funktionsfähig).
1727	Antonius Braun (1685-1728) baut die erste funktionsfähige 4-spezies Sprossenrad-Rechenmaschine.
1747	Phillip Matthäus Hahn (1739-1790) baut erste funktionsfähige 4-spezies Staffelwalzen-Rechenmaschine.
1805	Joseph-Marie Jaquard (1753-1834) baut Webstuhl mit Lochkarten-Steuerung.
ab 1821	Serienmäßige Fabrikation mechanischer Rechenmaschine
1833	Charles Babbage (1792-1871) entwickelt das Konzept eines programmgesteuerten Rechenautomaten.
1886	Herrmann Hollerith (1860-1929) baut elektromechanische Sortier- und Zählmaschinen für Lochkarten.
1903	Percy E. Ludgate führt bedingte Sprünge und Dreiadressbefehle ein.
1941	Konrad Zuse (geb. 1910) baut den ersten programmgesteuerten Rechenautomaten der Welt, die "Z3" (elektromechanisch)
1943	In England wird unter Mitarbeit von Alan. M. Turing die Rechenmaschine "Colossus" zur Entschlüsselung deutscher Funksprüche gebaut.
1944	Howard Hathaway Aiken (1910-1973) baut den ersten programmgesteuerten Rechenautomaten der USA ("Harvard MARK I", elektromechanisch).
1944	John v. Neumann (1903-1957) konzipiert den elektronischen Rechenautomaten "EDVAC" mit interner Speicherung des Programms. Inbetriebnahme 1952.
1946	Die erste vollelektronische Großrechenanlage der Welt wird in Betrieb genommen ("ENIAC", Elektronenröhren, von Eckert, Mauchly und Goldstine). Beginn der 1. <i>Computergeneration</i> .
1947	John Bardeen und Walter H. Brattain entdecken den Transistoreffekt.
1948	"SSEG" von IBM als erster betriebsbereiter Rechner mit Speicherprogrammierung. Claude E. Shannon begründet die Informationstheorie.
1949	Ultraschall-Laufzeitspeicher; Trommelspeicher.
ab 1951	Serienfertigung von Rechnern für wissenschaftliche und kommerzielle Aufgaben (z.B. "UNIVAC" von Remington Rand Corp.)
1952	Unter Leitung von Robert Piloty wird an der Technischen Hochschule München die "PERM" gebaut.
1954	John W. Backus entwickelt die Programmiersprache FORTRAN (FORmula TRANslator).
1955	"TRADIC", der erste transistorbestückte Rechner der Welt. Beginn der 2. <i>Computergeneration</i> .
1958	Jack Kilby entwickelt bei Texas Instruments die erste integrierte Schaltung.
ab 1962	Einsatz von Hybridschaltungen in Computern. Beginn der 3. <i>Computergeneration</i> . DEC entwickelt die PDP1.
ab 1965	Entwicklung von Minicomputern.

ab 1968	Monolithisch-integrierte Schaltungen, Beginn der 4. Computergeneration, Mehrbenutzer-Dialogbetrieb.
1971	Erste Mikroprozessoren (Intel 4004, 8008), Taschenrechner HP-35.
ab 1973	Serienfertigung von elektronischen Taschenrechnern.
1976	Erster Einchip-Mikrocomputer Intel 8048, erste 16-Bit-Mikroprozessoren (Texas 9900).
1977	Commodore "PET", der erste vollständige Heimcomputer/ Personal Computer.
1981	IBM-PC (4,77 MHz Taktfrequenz), erste 32-Bit-Mikroprozessoren (Intel, HP).
1983	HP entwickelt VLSI-Chip mit 450 000 Transistorfunktionen (50 mm ²)
1984	ULSI-Chips mit ca. 1 600 000 Transistorfunktionen.
1988	RISC-Architektur (Reduced Instruction Set Computer), Transputer
1989	Megabit-Speicherchip, 64-Bit-RISC-Prozessoren.
2000	Pentium 3 mit 1 GHz Taktfrequenz

Buchtips:

Edgar P. Vorndran:
"Entwicklungsgeschichte des Computers"
VDE-Verlag

Rolf Oberliesen:
"Information, Daten und Signale"
rororo Sachbuch

Lindner/ Wohak/ Zeltwanger:
"Planen, Entscheiden, Herrschen"
rororo Sachbuch

Dr. Michael Schöttner
"Systemprogrammierung"
pi3.informatik.uni-mannheim.de/~schiele/sysprog/



[Zum vorhergehenden Abschnitt](#)



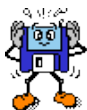
[Zum Inhaltsverzeichnis](#)



[Zum nächsten Abschnitt](#)

*Die HTML-Fassung entstand unter Mitwirkung von Volker Arndt
Copyright © FH München, FB 04, Prof. Jürgen Plate*

Letzte Aktualisierung: 12. Jul 2011



Einführung Datenverarbeitungssysteme

von Prof. Jürgen Plate

2. Aufbau und Arbeitsweise einer Datenverarbeitungsanlage

2.1 Hardware & Software

Eine DVS besteht aus einzelnen untereinander verbundenen Bauteilen, Schaltungen und Geräten, die fest verdrahtet sind → im wesentlichen digitale Schaltungen

→ Hardware

→ DV-Anlage. Die Hardware bildet noch kein programmierbares DVS (abgesehen von festverdrahteten Spezialrechnern für bestimmte Aufgaben, z.B. Taschenrechner). Die Hardware (HW) eines Universalrechners ist problemunabhängig.

Zum Arbeiten, d.h. zum Verarbeiten von Daten, sind noch Programme erforderlich.

→ Programme sind die Bearbeitungsvorschriften für die Daten. Ein Programm ist die Beschreibung eines Arbeitsablaufs mit folgenden Eigenschaften:

- Eindeutigkeit,
- Ausführbarkeit,
- endliche Länge,
- jeder Schritt legt fest, was womit zu tun ist.

Die Gesamtheit der Programme nennt man **Software** (SW). Die Software ist veränderbar. Das jeweilige Programm bestimmt, welche Aufgabe das DVS bearbeiten kann. Bei der Software unterscheidet man:

- **Systemsoftware:**
Sie entlastet den Benutzer von Verwaltungsaufgaben (z.B. das Anordnen der Daten auf einem Speichermedium) und ermöglicht eine (mehr oder weniger) komfortable Handhabung der DVS und der Anwenderprogramme.
- **Anwendersoftware:**
Die Gesamtheit der Anwenderprogramme bilden die Anwendersoftware, z.B. Textverarbeitung, Datenbanken, Buchhaltung, Kalkulation.

Die Grenze zwischen Hardware und Software kann bei verschiedenen DVS verschieden sein (Funktionen entweder als SW oder HW realisiert, z.B. programmierte Gleitpunktarithmetik oder Arithmetikprozessor).

Es gibt Teile der Systemsoftware, die zwar als Programm realisiert sind, jedoch in einem Festwertspeicher abgelegt werden. Der Speicher kann nur gelesen werden, ist wie die HW nicht ohne weiteres veränderbar und für den Benutzer nicht von der HW unterscheidbar, z.B. Mikroprogramme, Betriebssystemteile, Systemsoftware, SW von Einplatinen-Computern (Steuerung/ Regelung) → Software on Silicon → Firmware.

2.2 "Klassische" Rechnerarchitektur

Die meisten Computer, die heute verwendet werden, sind nach dem Konzept organisiert, das John v. Neumann (zusammen mit Burks und Goldstine) in den Jahren 1944-1947 entwickelt hat:

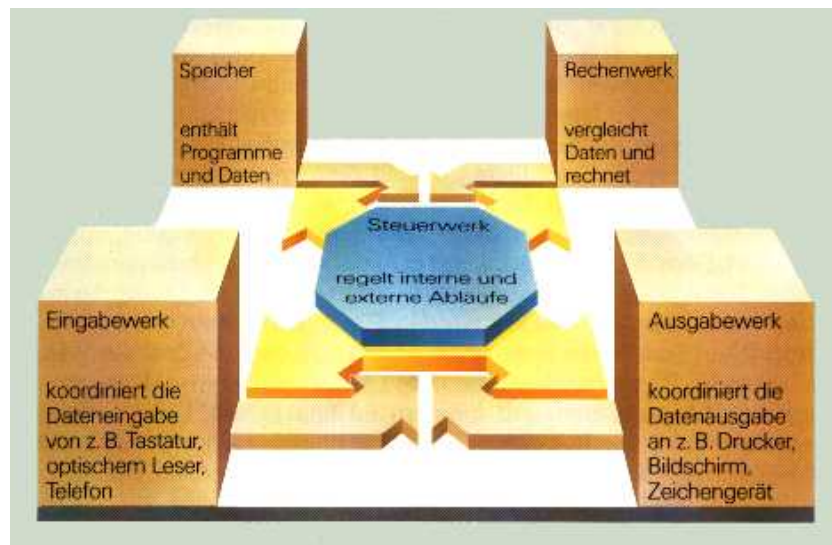
→ problemunabhängige Struktur

→ Universalrechner mit interner Programmspeicherung

Datenverarbeitungssysteme

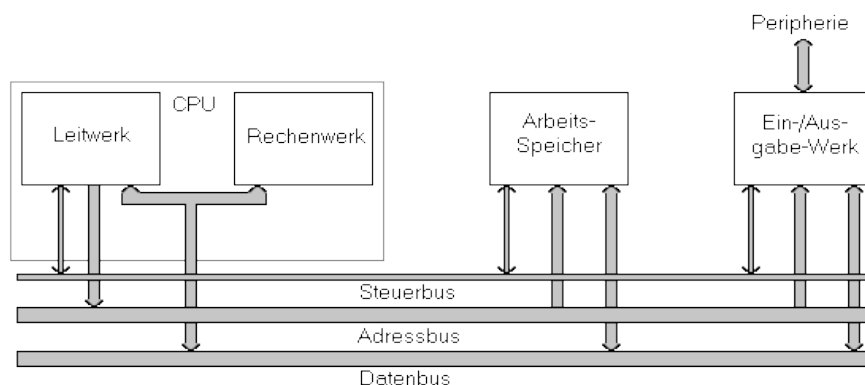
→ v. Neumann-Architektur
(→ "klassischer" Rechner)

Der **v. Neumann-Rechner** besteht aus 4 Funktionsgruppen:



- **Speicher**, der sowohl Programmbefehle als auch Daten aufnimmt (Hauptspeicher, Arbeitsspeicher, Main Memory). Die Daten/Befehle liegen binär verschlüsselt, also als 0/1-Folgen vor. Prinzipiell besteht kein Unterschied zwischen Daten und Befehlen. Unterteilung in Speicherplätze (Speicherzellen, Speicherworte), die über Adressen angesprochen werden.
- **Steuerwerk, Leitwerk** (Befehlsprozessor), das den Programmablauf steuert. Die Befehle werden interpretiert und deren Ausführung veranlasst, gesteuert und überwacht.
- **Rechenwerk** (Datenprozessor), das die zu bearbeitenden Daten verknüpft und verändert. Leitwerk und Rechenwerk bilden die zentrale Verarbeitungseinheit (CPU, Prozessor).
- **Ein- und Ausgabewerk** (E/A-Prozessoren), welche die Schnittstelle zur Außenwelt (Peripherie) bilden. Zusammen mit den E/A-Geräten besorgen sie die Kommunikation mit der realen Umwelt.

Der Informationsaustausch (Steuersignale, Daten/Befehle, Adressen) zwischen diesen Funktionsgruppen erfolgt über interne Datenwege. Diese können realisiert werden als direkte Verbindung der einzelnen Funktionsgruppen oder als gemeinsame Datenschiene → Busstruktur. Bei Mikros und Minis (z. B. Workstation, PC) wird im allgemeinen die Busstruktur verwendet.



Die Funktionsgruppen Speicher, Leitwerk, Rechenwerk, E/A-Werk bilden zusammen mit den

Verbindungswegen die Zentraleinheit (ZE, CU, CPU) eines Computers. Die ZE wird häufig schon als "Computer" bezeichnet. Zu einer vollständigen DVS gehören jedoch noch die Peripheriegeräte. Dies sind:

- Dateneingabe-Geräte
Lochkartenleser, Lochstreifenleser, Belegleser, Messfühler (Sensoren), Messgeräte, A/D-Wandler, Digitalisierer
- Datenausgabe-Geräte
Lochkarten-Stanzer, Lochstreifen-Stanzer, Drucker, Plotter, D/A-Wandler, Stellglieder
- Kommunikations- (Dialog-) Geräte
Tastensfeld, Datensichtgerät (Terminal), Modem, Maus, Lichtgriffel, Sprach-E/A
- externe Speicher
Magnet-Trommel, Magnet-Band, Magnet-Platte, optische Platte

Operationsprinzip

Die CPU kann nur elementare, (fest verdrahtete) Befehle verarbeiten (→ Maschinenbefehle). Jedes Programm besteht also aus einer Folge elementarer Befehle. Da im Speicher nicht zwischen Daten und Befehlen unterschieden wird, muss das Leitwerk entscheiden, ob der Inhalt einer Speicherzelle als Befehl oder Datum aufzufassen ist. Es gibt somit zwei Zustände des Rechners: Befehl holen (Interpretieren als Befehl) und Befehl ausführen (Interpretation als Datum). Entsprechend gibt es zwei Phasen der Programmabarbeitung:

- 1. Phase: Der (durch den sogenannten Befehlszähler) referierte Speicherplatzinhalt wird geholt und als Befehl interpretiert (Befehlsholphase, instruction fetch)
- 2. Phase: Der Speicherplatzinhalt der durch den Befehl spezifizierten Adresse wird geholt, als Datum interpretiert und dem Befehl entsprechend verarbeitet (Befehlsausführungsphase, instruction execution)

Dieses Zweiphasenschema erfordert eine streng sequentielle Ausführung eines Programms, d.h. es sind zwar Sprünge möglich, jedoch keine parallele Bearbeitung mehrerer Befehle. Ein v. Neumann-Rechner bearbeitet zu jedem Zeitpunkt immer nur einen Befehl, der immer eine Datenoperation im Rechenwerk bewirkt. → Typ SISD (single instruction, single data)

Vorteile der v. Neumann-Architektur:

(Dies ist der Hauptgrund für ihre Langlebigkeit)

- Einfachheit (übersichtlich, minimaler HW-Aufwand)
- maximale Flexibilität (bei genügend elementaren Befehlen)

Nachteile der v. Neumann-Architektur:

- nur ein Prozessor
- nur ein Verbindungsweg zwischen CPU und Speicher (zwischen CPU und Speicher wird immer nur ein Wort transportiert)
- sequentielle Verarbeitung von Befehl und Datum → v. Neumann-Flaschenhals

Durch neue Rechnerarchitekturen kann man die Nachteile des v. Neumann-Rechners beseitigen, was aber erst durch die stürmische Entwicklung auf dem Hardware-Sektor (sinkende Preise, höhere Integrationsdichte, schnellere Chips, höhere Zuverlässigkeit der Komponenten) möglich wurde. Die neuen Architekturen werden (jedenfalls zur Zeit) den v. Neumann-Rechner nicht ablösen, sondern nur in den Gebieten sinnvoll ergänzen, in denen sie wirkliche Vorteile bringen. Ein Weg ist z.B. der RISC (Reduced Instruction Set Computer), der durch den vereinfachten Befehlssatz wesentlich schneller

arbeitet. Ein Teil der neuen Architekturen ist bereits in kommerziell vertriebenen DVS vorhanden, ein Teil befindet sich im Experimentierstadium und ein letzter Teil ist noch in der Entwicklungsphase.

2.3 Innovative Rechnerarchitekturen

Erhöhung der Leistungsfähigkeit des v. Neumann-Rechners

- Verlagerung der Prozessorfunktionen bei der Ein-/Ausgabe auf das E/A-Werk ("intelligente" Schnittstellen, E/A-Prozessoren, Vor-Rechner, Front-End-Rechner). Dies vor allem bei Großrechnern, aber auch immer häufiger bei Minis und Mikros!
- Weiterentwicklung des Architekturprinzips durch Steuerung der einzelnen Funktionseinheiten über eigene Prozessoren, wodurch eine teilweise Parallelarbeit möglich wird.
- Bearbeiten einzelner, spezieller Befehle durch Spezialprozessoren, die abwechselnd mit der "normalen" CPU arbeiten (z. B. Arithmetik-Koprozessor).

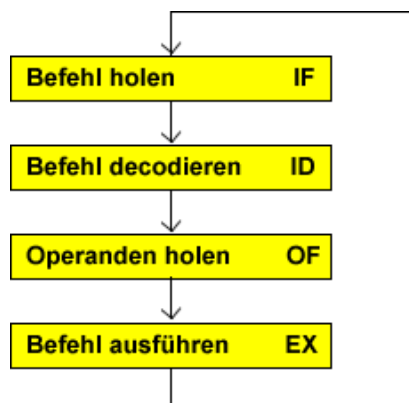
Vermeiden oder Mildern des v. Neumann-Flaschenhalses

- Neue Rechnerarchitekturen durchbrechen den streng sequentiellen Ablauf des v. Neumann-Rechners durch Parallelisierung verschiedener Abläufe im Computer.
- Feldrechner bestehen aus mehreren parallelen Rechenwerken, mit gemeinsamen Steuerwerk. So kann die gleiche Operation auf vielen verschiedenen Datenwerten (Feldern) gleichzeitig ausgeführt werden. → SIMD (single instr., multiple data)

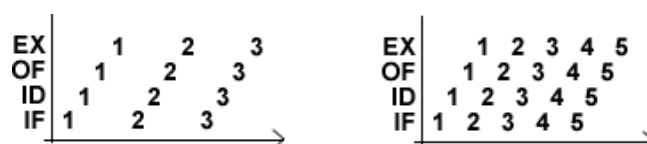
Asynchrone Trennung von Befehlshol- und -ausführungsphase

→ Warteschlange

- Pipeline-Rechner: Entweder mehrere hintereinandergeschaltete Rechenwerke, die verschiedenen Aufgaben erfüllen, aber gemeinsam gesteuert werden oder Aufbau eines Prozessors mit hintereinandergeschalteten spezialisierten Teilelementen. Die einzelnen Befehle werden in Teilaufgaben zerlegt, die nach einander von den hintereinandergeschalteten Teilelementen bearbeitet werden.



Die nächste Operation kann bereits begonnen werden, wenn die vorhergehende in die nächste "Station" gelangt ist.



gleichzeitige Bearbeitung mehrerer Befehle (schematisch dargestellt)

Zur selben Zeit führt jedes Verarbeitungselement verschiedene Operationen auf unterschiedlichen Daten durch.

→ überlappende Befehlszyklen

→ MIMD (multiple instr., multiple data)

- Vektorrechner: Kombination mehrerer Pipelines
- Multiprozessorsysteme: Koppelung mehrerer Prozessoren, die auf den gleichen Speicher (bzw. Speicherbereich) zugreifen und weitgehend unabhängig voneinander arbeiten. Jede CPU kann ein eigenes Programm bearbeiten → MIMD.
- Polyprozessorsysteme: Multiprozessorsystem mit verteilter Kontrolle, bei dem die einzelnen CPUs autonom arbeiten, aber miteinander kommunizieren und kooperieren → MIMD.

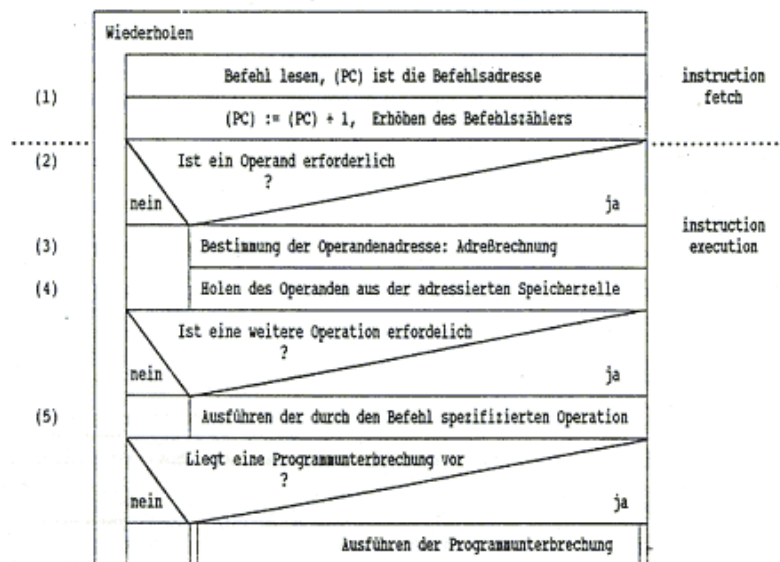
Meist besitzt jeder Prozessor einen eigenen Speicher, z.B. Transputer. Fehlertolerante Mehrrechnersysteme eng/lose gekoppelt (z. B. in der Raumfahrt).

2.4 Befehlszyklus

Der Befehlszyklus wird von der CPU ständig durchlaufen:

- die Befehle stehen im Speicher.
- das Leitwerk "weiß" jederzeit, welcher Befehl als nächster auszuführen ist.
- die Adresse (= Nummer der Speicherzelle) des nächsten auszuführenden Befehls steht in einem speziellen Register des Leitwerks, dem Befehlszähler (Program Counter, PC, BZ, Instruction Address Register, IAR, Instruction Pointer, IP).
- üblicherweise stehen aufeinanderfolgende Befehle in aufeinander folgenden Speicherzellen, der zuerst auszuführende Befehl hat die niedrigste Adresse.
- zu Beginn des Programms wird der BZ mit dessen Startadresse geladen.

Ablauf des Befehlszyklus



In der Befehlsholphase erfolgt ein Speicherzugriff (1a) auf die vom Befehlszähler (BZ) angezeigte Adresse. Der entsprechende Befehl wird in das Befehlsregister (IR) des Leitwerks gebracht (1b). Anschließend wird der BZ um 1 erhöht. Er zeigt damit auf den nächsten Programmbefehl. Besteht ein Befehl aus mehreren Speicherworten, setzt sich diese Phase auch aus mehreren Speicherzugriffen zusammen (BZ wird jedes Mal erhöht), bis der Befehl vollständig im IR steht. Es erfolgt hier in der Befehlsholphase bereits eine Teilauswertung des Operations-Codes (s. Grafik). Das Befehlsregister besteht hier aus Op-Code-Register (OR, Befehlsregister) und Adress-Register (AR).

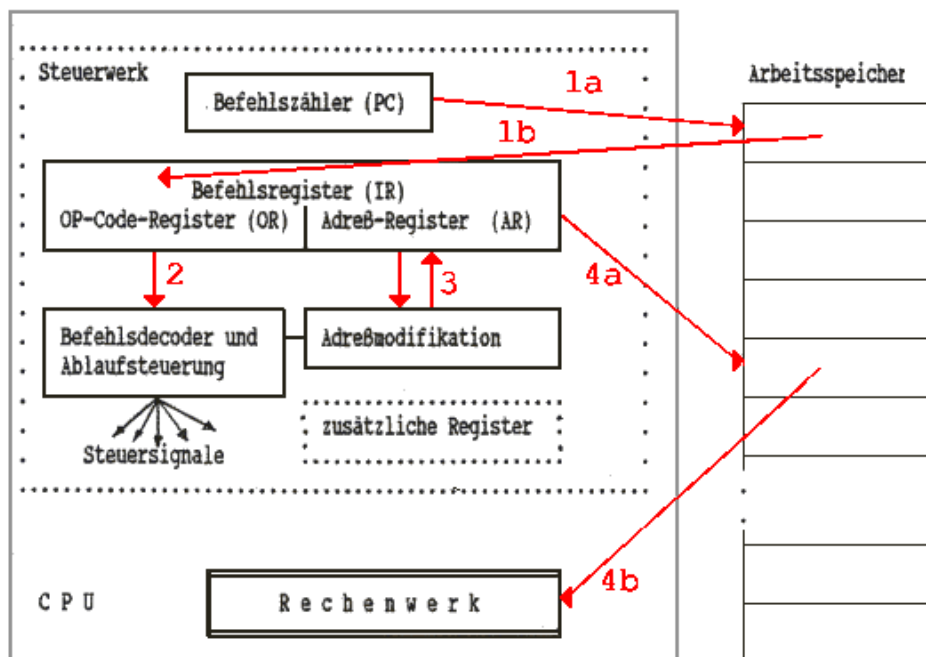
Datenverarbeitungssysteme

Der Befehl im OR wird nun decodiert (Befehlsdecoder) und der Ablaufsteuerung zugeführt. Diese ist in der Regel als Mikroprogramm realisiert. Die Ablaufsteuerung erzeugt nun die nötigen Steuersignale (siehe Kapitel 8).

Benötigt der Befehl Operanden, so wird deren Adresse aus dem Inhalt des AR ermittelt. Häufig ist im Befehl nicht die tatsächliche Operandenadresse, sondern nur eine Teilinformation enthalten, die noch geeignet ergänzt werden muss (→ Adressrechnung, s. *später*).

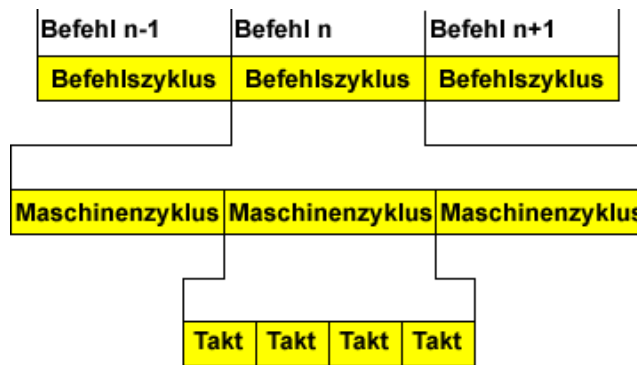
Nun erfolgt ein Speicherzugriff (4a) auf die so festgestellten Operanden-Adresse. Der Operand wird in das vom Op-Code spezifizierte Register oder in das Rechenwerk (4b) oder in die ausgewählte Speicherzelle gebracht.

Falls durch den Op-Code weitere Teiloperationen gefordert sind, werden diese nun ausgeführt. Dabei kann auch der Inhalt des BZ verändert werden (Sprungbefehle, Unterprogramm-Aufrufe).



Jeder Befehlszyklus besteht aus einer Anzahl von Teilschritten. Die Anzahl der Teilschritte kann für unterschiedliche Befehle verschieden sein, auch die Zeitdauer der einzelnen Befehle kann unterschiedlich sein.

→ die Befehlszyklen verschiedener Befehle können unterschiedlich lang sein. Die tatsächliche Dauer eines Befehlszyklus hängt von der Taktfrequenz der CPU ab. Der gesamte Ablauf in der CPU wird durch einen zentralen Takt gesteuert. Ein Befehlszyklus kann in eine Reihe von Maschinenzyklen zerlegt werden (z.B. Speicherzugriff in einem Maschinenzyklus). Ein Maschinenzyklus kann eine oder mehrere Taktperioden (Arbeitstakt des Prozessors) dauern.



2.5 Programm-Unterbrechungen

Häufig ist es notwendig, dem Computer zu einem beliebigen Zeitpunkt von außen ein bestimmtes Verhalten aufzuzwingen, z. B. das Lesen eines Datenwertes nach dessen Eintippen am Terminal (→ besonders wichtig in der Prozess-Steuerung). Wenn das externe Ereignis relativ selten auftritt, wäre es sehr ineffektiv, den Rechner mit einem ständigen Terminal-Abfrageprogramm zu beschäftigen (Polling).

Besser ist es, den Rechner in seiner Arbeit erst dann zu unterbrechen, wenn das externe Ereignis eintritt. Die Unterbrechung erfolgt durch ein an die CPU gesandtes Signal, die Unterbrechungsanforderung (UA, interrupt request, IRQ), das die CPU veranlasst, das gerade laufende Programm zu unterbrechen und eine Befehlsfolge auszuführen, die auf die Unterbrechung reagiert (z.B. ein Datum von der Tastatur in den Speicher bringt) → Programmunterbrechung (interrupt).

Die bei der Unterbrechung zu startende Befehlsfolge ist das Unterbrechungs-Antwortprogramm (interrupt service routine, ISR). Nach dem Abarbeiten der ISR fährt der Rechner mit der Ausführung des unterbrochenen Programms fort. Auf gleiche Weise lassen sich Reaktionen des Computers auf andere seltene oder unvorhersehbare Ereignisse (z.B. Ausfall der Stromversorgung) behandeln. Als Reaktion auf eine Unterbrechungsanforderung geschehen zwei Dinge:

- Retten des aktuellen Programmzustands: Der Maschinenstatus, d.h. der Inhalt der Register der CPU, muss festgehalten werden. Dies geschieht durch Wegspeichern der Registerinhalte des gerade unterbrochenen Programms (→ Zustandsvektor).
- Laden der Register mit dem Zustandsvektor der ISR (z.B. Startadresse der ISR): Der Zustandsvektor steht an einer festgelegten Adresse im Speicher (→ Interrupt-Vektor). Das Programm wird nun ab der Startadresse der ISR fortgesetzt.

Die Unterbrechung darf nicht mitten in einer Befehlsausführung erfolgen (undefinierter Prozessorstatus!), sondern erst nachdem der gerade laufende Befehl vollständig abgearbeitet ist, also am Ende des Befehlszyklus (siehe Struktogramm). Nach Ende der ISR wird der ursprüngliche Zustandsvektor wieder vom Stack geholt und das Programm an der Unterbrechungsstelle fortgesetzt.

Während des Rettens des Programmzustands und des anschließenden Ladens des Zustandsvektors der ISR darf keine neue Unterbrechungsanforderung auftreten (Zustandsinformation unvollständig!). Der Befehlssatz der Prozessoren enthält daher spezielle Befehle zum Sperren und Freigeben von Unterbrechungsanforderungen.



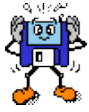
[Zum vorhergehenden Abschnitt](#)



[Zum Inhaltsverzeichnis](#)



[Zum nächsten Abschnitt](#)



Einführung Datenverarbeitungssysteme

von Prof. Jürgen Plate

3. Informationsdarstellung in Rechenanlagen

3.1 Grundlagen der Zahlensysteme

Definitionen

- **Ziffer:** Element einer Menge von Symbolen zur Darstellung von Zahlen
- **Zahl:** Anordnung von Ziffern, die in einem vereinbarten Zahlensystem einen bestimmten Wert darstellen.
- **Kennzeichen eines Zahlensystems sind:**
 - ♦ Zeichenvorrat, der die Menge der zulässigen Ziffern darstellt (z. B. Dezimalsystem: 0,1,2,3,4,5,6,7,8,9)
 - ♦ Bildungsgesetz, das die Anordnung für die Ziffern und die Ermittlung eines Zahlenwertes festlegt

Der allgemeine Umgang mit dem Dezimalsystem und dessen Bildungsgesetz lässt uns vergessen, daß verschiedenen Bildungsgesetze existieren können, z. B.:

- ♦ Ziffernwertsystem, bei denen der Zahlenwert alleine durch die Ziffer bestimmt wird (z. B. römische Zahlen)
- ♦ Stellenwertsystem, bei dem der Zahlenwert, den die Ziffer repräsentiert, durch die Position der Ziffer innerhalb der Zahl (= Stelle) bestimmt wird.

Für DV-Systeme eignet sich nur das Stellenwertsystem.

3.2 Stellenwertsysteme

Für den Wert einer Zahl $Z = a_n a_{n-1} \dots a_0 a_{-1} \dots a_{-m}$ in einem Stellenwertsystem zur Basis B gilt:

$$Z = \sum_{i=-m}^n a_i \cdot B^i$$

wobei für die Ziffern a_i gilt: $0 \leq a_i < B$

Als Basis bezeichnet man die kleinste, nicht mehr durch eine Ziffer darstellbare Zahl. Am geläufigsten ist uns das Dezimalsystem mit der Basis 10. Die Festlegung ist rein willkürlich und vermutlich auf die Zahl der Finger beider menschlicher Hände zurückzuführen. → Zahlzeichen der Azteken (Basis 20, Ziffernwertsystem), Zahlzeichen der Inkas (Basis 10, Stellenwertsystem).

Die verkürzte Schreibweise durch Aneinanderreihung von Ziffern ist eine abkürzende Schreibweise der Summenformel. Damit kann jede positive und negative reelle Zahl dargestellt werden, indem jede Stelle der Ziffernfolge mit einer Zehnerpotenz gewichtet wird.

Ganze Zahlen:

$$2018 = 2 \cdot 10^3 + 0 \cdot 10^2 + 1 \cdot 10^1 + 8 \cdot 10^0$$

allgemein:

$$a_n a_{n-1} \dots a_0 a_{-1} \dots a_{-m} = a_n \cdot 10^n + a_{n-1} \cdot 10^{n-1} + \dots + a_0 \cdot 10^0 + a_{-1} \cdot 10^{-1} + \dots + a_{-m} \cdot 10^{-m}$$

Echter Dezimalbruch:

$$0,328 = 0 + 3 \cdot 10^{-1} + 2 \cdot 10^{-2} + 8 \cdot 10^{-3}$$

allgemein:

$$0, a_{-1} \dots a_{-m} = a_{-1} \cdot 10^{-1} + \dots + a_{-m} \cdot 10^{-m}$$

Dezimalzahlen:

Radixschreibweise: $Z = a_n a_{n-1} \dots a_0 a_{-1} \dots a_{-m}$

Potenzschreibweise: $Z = a_n \cdot 10^n + a_{n-1} \cdot 10^{n-1} + \dots + a_0 \cdot 10^0 + a_{-1} \cdot 10^{-1} + \dots + a_{-m} \cdot 10^{-m}$

Potenz-Summen-Schreibweise:
$$Z = \sum_{i=-m}^n a_i \cdot 10^i$$

Prinzipiell kann jede ganze Zahl größer 1 Basis B eines Stellenwertsystems sein. In der Praxis finden aber nur wenige Werte Verwendung.

Dualzahlen

Für das Dualsystem ist die Basis $B = 2$ und a aus $\{0,1\}$, z. B. $Z = 1010,1_{\text{dual}} = 10,5_{\text{dez}}$. Dieses Zahlensystem ist speziell für die Digitaltechnik von Bedeutung, da nur zwei Zustände eine physikalischen Größe benötigt werden (Spannung, Strom, Frequenz, Magnetisierungsrichtung). Nachteil ist die unübersichtliche, monotone Ziffernfolge bei der Darstellung langer Dualzahlen. Daher werden beim Umgang mit EDV-Anlagen zwei andere Zahlensysteme verwendet:

Oktalzahlen

Zusammenfassung von 3 Dualstellen zu einer Oktalstelle (bessere Lesbarkeit, kürzer zu schreiben, leicht umzurechnen) → Oktalsystem: Basis $B = 8$ und a ist aus $\{0,1,2,3,4,5,6,7\}$, z. B. $101001_{\text{dual}} = 51_{\text{oktal}}$.

Bei der Eingabe von Oktalzahlen müssen diese - zur Unterscheidung von Dezimalzahlen - gekennzeichnet werden. Dies geschieht durch Hinzustellen der Basis, z. B.: $1257_{(8)}$. Häufig (besonders bei Programmiersprachen) geschieht dies auch durch Voranstellen von '@' oder Anfügen von 'O', 'Q', 'C' z. B.: @154 154O 154Q 154C (bei C-Compilern auch durch führende Null).

Sedezimalzahlen (= Hexadezimalzahlen)

Zusammenfassung von 4 Dualstellen ergibt eine Sedezimalstelle → Sedezimalsystem: Basis $B = 16$ und a aus $\{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F\}$, z. B. $101001_{\text{dual}} = 29_{\text{sedezimal}}$. Übliche Darstellung der eigentlich binären Information in einem Rechner (Kurzschreibweise binärer Info). Kennzeichnung bei der Programmierung durch Voranstellen von '\$', '0x' oder Anfügen von 'H' (für "hexadezimal") z. B.: \$FFC2 FFC2H 0xFFC2. Gelegentlich wird eine führende Null verwendet, wenn die Zahl mit einem Buchstaben beginnt.

Weitere, manchmal verwendete Zahlensysteme, sind das Ternärsystem ($B = 3$) und das Quinärsystem ($B = 5$). Es stellt sich nun die Frage, wie man Zahlen von einem Zahlensystem in ein anderes umrechnen kann.

3.3 Umrechnung zwischen den Stellenwertsystemen

Umwandlung Dezimalsystem → anderes Stellenwertsystem

Die Darstellung als Potenzreihe kann als Polynom zur Basis B aufgefasst werden.

$$Z = a_n \cdot B^n + a_{n-1} \cdot B^{n-1} + \dots + a_0 \cdot B^0 + a_{-1} \cdot B^{-1} + \dots + a_{-m} \cdot B^{-m}$$

Eine andere Form der Darstellung ist das Hornerschema (hier nur für den ganzzahligen Teil):

$$Z_0 = Z = ((\dots((a_n \cdot B + a_{n-1}) \cdot B + \dots a_2) \cdot B + a_1) \cdot B + a_0$$

$$Z_1 = \frac{Z}{B} = ((\dots((a_n \cdot B + a_{n-1}) \cdot B + \dots a_2) \cdot B + a_1 \quad \text{Rest } a_0$$

$$Z_2 = \frac{Z_1}{B} = ((\dots((a_n \cdot B + a_{n-1}) \cdot B + \dots a_2 \quad \text{Rest } a_1$$

usw.

$$Z_n = \frac{Z_{n-1}}{B} = a_n \quad \text{Rest } a_{n-1}$$

$$Z_{n+1} = \frac{Z_n}{B} = 0 \quad \text{Rest } a_n$$

Bei der Umwandlung ganzer Zahlen gibt es nur positive Potenzen des Basis B. Bei fortgesetzter Division durch die Basis B' des gesuchten Zahlensystems fallen die gesuchten Koeffizienten als Reste der ganzzahligen Division an. Die Division wird fortgesetzt, bis das Divisionsergebnis 0 geworden ist. Die Divisionsreste sind die Ziffern des Zielsystems in aufsteigender Reihenfolge (1. Rest = niederwertigste Ziffer, letzter Rest = höchstwertige Ziffer).

Dezimal-Dual-Wandlung ganzer Zahlen (Basis B = 2)

1. Dualzahl auf 0 setzen. Wir führen einen Wert I ein, der die gerade bearbeitete Stelle der erzeugten Dualzahl enthält. I wird auf 1 gesetzt.
2. I-te Stelle der Dualzahl := Dezimalzahl mod 2. (Der Operator "mod" bezeichnet den Rest der Division)
3. Dividiere die Dezimalzahl durch 2. Dieser Wert ergibt die neue Dezimalzahl für die Berechnung der nächsten Stelle.
4. Erhöhe den Wert von I um 1. Wenn die Dezimalzahl > 0 ist, fahre fort bei Schritt 2.

Dezimal-Dual-Wandlung echtgebrochener Zahlen (Basis B = 2)

Horner Schema für den gebrochenen Teil:

$$Z_0 = Z = ((\dots((a_{-1} \cdot B^{-1} + a_{-2}) \cdot B^{-1} + \dots a_{-m}$$

Bei der Umwandlung von echten Brüchen wird nach dem gleichen Schema verfahren, nur wird hier fortgesetzt mit der Basis des Zielsystems multipliziert. Die ganzzahligen Anteile der einzelnen Multiplikationsschritte ergeben die Koeffizienten des Zielsystems. Die Rechnung ist zuende, wenn der gebrochene Anteil des Multiplikationsergebnisses 0 wird. Die ganzzahligen Anteile der Multiplikationen werden dann in absteigender Reihenfolge aufgeschrieben (1. Anteil = höchste Stelle).

Vorsicht: Ein endlicher Bruch in einem Stellenwertsystem ist nicht immer auch ein endlicher Bruch in einem anderen.

1. Dualzahl auf 0 setzen. Wir führen einen Wert I ein, der die gerade bearbeitete Stelle der erzeugten Dualzahl enthält. I wird auf 1 gesetzt.

2. Die Dezimalzahl wird mit der Basis 2 multipliziert. Die I-te Stelle der Dualzahl ergibt sich aus dem ganzzahligen Anteil der Dezimalzahl (Vorkommastelle).
3. Man nehme den gebrochenen Anteil der Dezimalzahl (Nachkommastellen) für die weitere Berechnungen.
4. Man erhöhe den Wert von I um 1. Wenn die Dezimalzahl > 0 ist, fahre fort bei Schritt 2.

Umwandlung anderes Stellenwertsystem → Dezimalsystem

Der offensichtliche Weg ergibt sich aus der Definition einer Zahl im Stellenwertsystem:

$$Z = a_n \cdot B^n + a_{n-1} \cdot B^{n-1} + \dots + a_0 \cdot B^0 + a_{-1} \cdot B^{-1} + \dots + a_{-m} \cdot B^{-m}$$

Die Umwandlung erfolgt durch Auswertung der Summe, wobei die Koeffizienten und die Potenzen der Basis B im Dezimalsystem dargestellt werden.

Beispiel: Umwandlung aus dem Dualsystem

$$Z = 101,11_{(2)} = ??_{(10)}$$

$$Z = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2}$$

$$Z = 4 + 0 + 1 + 1/2 + 1/4 = 5,75$$

Eine andere Möglichkeit ist wieder die Anwendung des Hornerschemas. Das oben dargestellte Verfahren der Dezimal-Dual-Umwandlung wird einfach umgekehrt. Beginnend bei der höchstwertigen Ziffer wird bei ganzen Zahlen mit der Basis des Ausgangssystems (dargestellt im Zielsystem) multipliziert und zur nachfolgenden Stelle addiert. Dies wird bis zur niederwertigsten Ziffer fortgesetzt. Das letzte Ergebnis ist die Zahl im Zielsystem.

Dual-Dezimal-Umwandlung, ganzzahlig

1. Setze die Dezimalzahl auf 0. Beginne bei der höchstwertigen Stelle n der Dualzahl. Wir führen einen Wert I ein, der die gerade bearbeitete Stelle der erzeugten Dualzahl enthält. I wird auf n gesetzt.
2. Multipliziere die Dezimalzahl mit 2 und addiere die I-te Stelle der Dualzahl zur Dezimalzahl.
3. Wiederhole Schritt 2 solange, bis alle Stellen der Dualzahl verarbeitet sind.

$$\text{Beispiel: } 11011101_{(2)} = ??_{(10)}$$

	1	1	0	1	1	1	0	1			
2 *	1	+1								=	3
2 *	3		+0							=	6
2 *	6			+1						=	13
2 *	13				+1					=	27
2 *	27					+1				=	55
2 *	55						+0			=	110
2 *	110							+1		=	221

$$\text{Beispiel: } 7C2_{(16)} = ??_{(10)}$$

	7	C	2			
16 *	7		+12		=	124
16 *	124			+2	=	1986

Dual-Dezimal-Umwandlung, echter Bruch

Bei echten Brüchen wird fortgesetzt durch die Basis des Ausgangssystems (dargestellt im Zielsystem) dividiert. Die Ziffern werden von der niederwertigsten Stelle aus abgearbeitet (rechts nach links) → Abbau zum Komma hin.

1. Man setze die Dezimalzahl auf 0. Man beginne bei der niederwertigen Stelle n der Dualzahl. Wir führen einen Wert I ein, der die gerade bearbeitete Stelle der erzeugten Dualzahl enthält. I wird auf n gesetzt.
2. Man addiere die I -te Stelle der Dualzahl zur Dezimalzahl und dividiere das Ergebnis durch 2.
3. Man wiederhole Schritt 2 solange, bis alle Stellen der Dualzahl verarbeitet sind.

Beispiel: $0,1011_{(2)} = ??_{(10)}$

$$\begin{array}{rcl}
 0, & 1 & 0 & 1 & 1 \\
 & & & 1 / 2 & = & 0,5 \\
 & & (1 + 0,5) / 2 & = & 0,75 \\
 & (0 + & 0,75) / 2 & = & 0,375 \\
 & (1 + & 0,37) / 2 & = & 0,6875
 \end{array}$$

Umwandlung Dual - Oktal - Hexadezimal

Die behandelten Umwandlungsroutinen können hier genauso verwendet werden. Da die Rechenoperationen immer im Zielsystem durchgeführt werden, ist dies für uns zumindest ungewohnt. Es gibt aber ein einfacheres Verfahren, das über das Dualsystem führt. Alle Basen sind Zweierpotenzen:

- oktal → 3 Dualstellen = 1 Oktalstelle
- hexadezimal 4 Dualstellen = 1 Hexadezimalstelle

Die Umwandlung erfolgt in zwei Schritten:

1. Umwandlung in das Dualsystem. Jede Stelle wird in 3 (oktal) oder 4 (hexadezimal) Dualstellen umgewandelt.
2. 3 bzw. 4 Dualstellen werden zusammengefasst und in eine Oktal- bzw. Hexadezimalstelle gewandelt. Die Dualzahl ist gegebenenfalls vorher auf volle 3er- oder 4er-Gruppen zu ergänzen (0!). Die Umwandlung erfolgt immer vom Komma weg (ganzzahliger Anteil nach links, gebrochener Anteil nach rechts).

Beispiel: $237.54_{(8)} = ??_{(16)}$

oktal	2	3	7	.	5	4	
dual	10	11	111	.	101	100	→ 1001 1111 . 1011 0000
sedezimal							9 F . B 0000

3.4. Arithmetik im Dualsystem

Grundsätzlich wird im Dualsystem nach genau denselben Regeln gerechnet, wie im Dezimalsystem. Lediglich durch das ungewohnte Zahlensystem erscheint und die Rechner schwieriger zu sein.

Arithmetische Grundoperationen

Für die Addition zweier einstelliger Dualzahlen $S = a + b$ gilt:

a	b	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Bei der letzten Operation entsteht ein Übertrag auf die folgende Stelle (Carry). Das Ergebnis ist nicht mehr einstellig darstellbar. Die Addition mehrstelliger Dualzahlen wird wie im Dezimalsystem stellenweise durchgeführt (Übertrag berücksichtigen).

Beispiel:

23	1 0 1 1 0	3,75	0 0 1 1 , 1 1
+13	0 1 1 0 1	+5,25	0 1 0 1 , 0 1
	1 1 1 0 0 0 Carry		1 1 1 1 , 1 0 Carry
35	1 0 0 0 1 1	9,00	1 0 0 1 , 0 0

Für die Subtraktion einstelliger Dualzahlen $D = a - b$ gilt:

a	b	D	B
0	0	0	0
0	1	1	-1
1	0	1	0
1	1	0	0

Bei der Operation $0 - 1$ ist von der nächsthöheren Stelle zu "borgen" (Borrow). Die Subtraktion mehrstelliger Zahlen erfolgt wieder stellenweise.

9,25	1 0 0 1 , 0 1
-4,75	0 1 0 0 , 1 1
	1 0 0 1 0 0 Borrow
4,50	0 1 0 0 , 1 0

Darstellung negativer Zahlen

Ein sehr einfacher Ansatz wäre es, ein Bit als Vorzeichen definieren und den Betrag des Zahlenwerts in den restlichen Bits des Wortes darstellen → Betrags-Vorzeichen-Form. Bei der technischen Realisierung gibt es jedoch einige Nachteile. In der Datenverarbeitung ist das Subtraktionsverfahren mit einem "Borrow"-Signal von der nächsthöheren Stelle nicht gut durchführbar. Daher führt man die Subtraktion auf die Addition einer negativen Zahl zurück. Die Darstellung negativer Zahlen erfolgt durch die sogenannte Komplementdarstellung.

Rückführung der Subtraktion auf die Addition:

$a - b = a + \bar{b}$, wobei $\bar{b}$ das Komplement von b ist.

Es stellt sich nun sofort die Frage, wie man das Komplement einer Zahl erhält. Dazu wird die Gleichung erweitert:

$a - b = a + (K - b) - K$, wobei K die "Komplementärzahl" ist.

Beispiel:

$$\begin{array}{r} 835 \\ -267 \\ \hline 568 \end{array} \rightarrow \begin{array}{r} 835 \\ +733 \\ \hline 1568 \\ -1000 \\ \hline 568 \end{array} \quad \begin{array}{l} (K-b)=1000-267, K=1000 \\ (K \text{ subtrahieren}) \end{array}$$

Die Subtraktion von K zum Schluss kann man recht einfach durch Weglassen der vordersten Stelle beim Ergebnis realisieren. K ist im Prinzip beliebig wählbar, jedoch würde für $K = 1000$ eine Zahl in Komplementdarstellung nicht von einer natürlichen Zahl unterscheidbar. Wir treffen daher zwei Vereinbarungen:

1. Für alle Rechnungen wird eine maximale Stellenzahl n festgelegt.
2. Für K wird (im Dezimalsystem) 10_{n+1} gewählt. Dann kann man nämlich eine Vorzeichenstelle verwenden. Wir vereinbaren:

- ◆ Vorzeichenstelle = 0: positive Zahl
- ◆ Vorzeichenstelle = 9: negative Zahl

Die Vorzeichenstelle dient zwar als Indikator für das Vorzeichen, ist jedoch Bestandteil der Zahl!

Beispiele (Komplement mit VZ-Stelle):

$$\begin{array}{r} 835 \\ -267 \\ \hline 568 \end{array} \rightarrow \begin{array}{r} 0835 \\ +9733 \\ \hline 10568 \end{array}$$

|
 |————— Vorzeichenstelle=0 → Ergebnis positiv
 |
 |————— Weglassen des Überlaufs

$$\begin{array}{r} 267 \\ -835 \\ \hline -568 \end{array} \rightarrow \begin{array}{r} 0267 \\ +9165 \\ \hline 9432 \end{array}$$

|
 |————— Vorzeichenstelle=0 → Ergebnis negativ

Re-Komplement: -0568

$$\begin{array}{r} -535 \\ -267 \\ \hline -802 \end{array} \rightarrow \begin{array}{r} 9465 \\ +9733 \\ \hline 19198 \end{array}$$

|
 |————— Vorzeichenstelle=0 → Ergebnis negativ
 |
 |————— Weglassen des Überlaufs

Re-Komplement: -0802

Nun ist also auch ein negatives Ergebnis erkennbar und der Betrag (bzw. das Endergebnis) kann durch nochmaliges Komplementieren ermittelt werden. Interessant wird diese Methode jedoch erst im Dualsystem.

Komplementbildung im Dualsystem

Zunächst eine allgemeine Definition. Es sind zwei Arten von Komplementen gebräuchlich. X sei eine n -stellige positive Zahl im Zahlensystem zur Basis B , dann ist:

- B-Komplement (Zweierkomplement): $\bar{X} = B^{n+1} - X$
("echtes Komplement")
- (B-1)-Komplement (Einerkomplement): $\bar{X} = \{B^{n+1} - 1\} - X$
("unechtes Komplement", "Stellenkomplement")

Auch hier gilt: Werden alle n Stellen für den Zahlenwert benutzt, dann ist nicht unterscheidbar, ob eine pos. Zahl oder das Komplement dargestellt wird $\rightarrow$ $(n+1)$ -te Stelle hat VZ-Funktion.

Die Bildung des Einerkomplements geschieht durch Invertieren jeder einzelnen Stellen (sehr einfach). Es gibt jedoch zwei "Nullen", beispielsweise für $n = 4$: $0000 = +0$, $1111 = -0$. Vor allem bei EDV-Systemen wird deshalb das Zweierkomplement verwendet. Die Bildung kann auf zwei Wegen erfolgen:

1. Einerkomplement bilden und dann 1 addieren:

0110 1100	X	
1001 0011	1-Komplement	
+	1	1 addieren
= 1001 0100 2-Komplement		

2. Direkt: Übernahme aller Stellen von rechts bis zur ersten 1 einschließlich, alle weiteren Stellen invertieren:

0110 1100	X
1001 0100	übernehmen/komplementieren
= 1001 0100 2-Komplement	

Wertebereich bei $n+1$ Stellen (n Ziffern und VZ): -2^n bis $+2^n-1$

Beispiel: 3 Stellen und Vorzeichen

		1000	-8
0111	+7	1001	-7
0110	+6	1010	-6
0101	+5	1011	-5
0100	+4	1100	-4
0011	+3	1101	-3
0010	+2	1110	-2
0001	+1	1111	-1
0000	0		

Frage: Warum nicht -2^n bis $+2^n$? Antwort: Der Wert 2^n wäre 1000, die Darstellung passt nicht mehr in 3 Stellen mit Vorzeichen $\rightarrow$ Verwechslung mit -8!

Die Gesamtheit der so dargestellten Zahlen nennt man *konegative Zahlen* (gesprochen "konegativ", kommt von "Komplement-negativ"). Der Vorteil der Komplementdarstellung von negativen Zahlen liegt in der Ausführbarkeit der Subtraktion als Addition.

B-Komplement:

$$Y - X = Y - (X + B^n) - B^n = Y + X - B^n$$

$$B^n = 1000...000 \text{ (n Nullen)}$$

Beispiele für die Komplement-Bildung

(Basis = 2, Stellenzahl $n = 4 \rightarrow 3$ Ziffern und VZ)

Dez.	Dual	1-kompl.	2-kompl.
0	0000	1111	0000
2	0010	1101	1110
5	0101	1010	1011
7	0111	1000	1001

Bei der Subtraktion wird das Komplement addiert und (bei pos. Ergebnis) der Übertrag in die $n+1$ -te Stelle weggelassen. Bei negativem Ergebnis: Komplementdarstellung. Das Carry-Bit (Übertrag) wird gesetzt, wenn in der höchstwertigen Stelle ein Übertrag auftritt.

Wenn man bei dem oben angegebenen Zahlenbereich $5 + 7$ addiert, ergibt sich folgende Rechnung:

$$\begin{array}{r} 0101 \\ 0111 \\ \hline 1100 \end{array}$$

Das Ergebnis 1100 ist eindeutig negativ, nämlich -4. Es liegt nicht im darstellbaren Zahlenbereich! Untersucht man die verschiedenen Möglichkeiten und fasst man die Erkenntnisse zusammen, ergibt sich folgende Tabelle:

Vorzeichen Operanden	Vorzeichen Ergebnis	Bereichs-überschr.
beide positiv	positiv negativ	nein ja
positiv negativ	positiv negativ	nein nein
beide negativ	positiv negativ	ja nein

Merksatz: Wenn beide Operanden das gleiche Vorzeichen haben und das Ergebnis ein davon abweichendes Vorzeichen, dann gab es eine Bereichsüberschreitung (Overflow).

Die bisher gesammelten Erkenntnisse führen dann zur Rechenvorschrift für die Subtraktion durch Komplementaddition (B-Komplement):

1. Stellenzahl von X und Y angleichen
2. B-Komplement X_k des Subtrahenden X bilden
3. Y und X_k addieren
4. Eventuellen Übertrag streichen
5. Ist VZ-Stelle von $Y =$ VZ-Stelle von X und unterscheidet sie sich von der VZ-Stelle des Ergebnisses, dann ist ein Zahlenbereichsüberlauf aufgetreten $\rightarrow$ Fehlermeldung.
6. Ist die VZ-Stelle des Ergebnisses = 0, dann ist das Ergebnis positiv, sonst ist das Ergebnis negativ (in B-Komplement-Darstellung). Will man in diesem Fall den Betrag des Ergebnisses feststellen, muss man das Ergebnis komplementieren.

Multiplikation

Bei Multiplikation und Division werden selten konegative Zahlen verwendet. Es werden vielmehr die Beträge multipliziert und dann das Vorzeichen betrachtet. Die duale Multiplikation erfolgt prinzipiell nach den selben Regeln, wie wir sie in der Schule für das Dezimalsystem gelernt haben: Der Multiplikand wird stellenweise mit dem Multiplikator malgenommen und die so erhaltenen Teilergebnisse stellenrichtig aufaddiert. Für Dualzahlen ist die Bestimmung eines Teilergebnisses besonders einfach. Da an der Multiplikatorstelle nur die Werte 0 oder 1 stehen können, ist bei einer 0 das Teilergebnis auch 0, im Fall der 1 ist das Teilergebnis identisch mit dem Multiplikanden.

Beispiel: $23 * 12 = 276$

$$\begin{array}{r}
 10111 \quad * \quad 01100 \\
 \hline
 \begin{array}{r}
 00000 \\
 10111 \\
 00000 \\
 00000 \\
 00000
 \end{array} \\
 \hline
 100010100
 \end{array}$$

Das Ergebnis zweier n-stelliger Dualzahlen hat nahezu die doppelte Stellenzahl, nämlich $2n - 1$.

Die Multiplikation von Dualzahlen kann durch Addition und Bitverschiebung vereinfacht werden (Verschieben um eine Bitposition entspricht Multiplikation mit 2 (links) oder Division durch 2 (rechts). Alle Rechenanlagen besitzen Bit-Verschiebe-Befehle. Angefangen bei niederwertigsten Bit des Multiplikators wird fortlaufend der Multiplikand zum Ergebnisregister (das zu Beginn auf Null gesetzt wurde) addiert, wenn das entsprechende Bit des Multiplikators den Wert 1 besitzt und dann um eine Position nach links verschoben. Die stellenrichtige Addition wird also nur in Teilschritte zerlegt.

Beispiel: $9 * 13 = 117$

$$\begin{array}{r}
 10111 \quad * \quad 1101 \\
 + \quad \quad \quad 1001 \\
 \hline
 \quad \quad \quad 1001 \\
 + \quad \quad \quad 0000 \\
 \hline
 \quad \quad \quad 01001 \\
 + \quad \quad \quad 1001 \\
 \hline
 \quad \quad \quad 101101 \\
 + \quad \quad \quad 1001 \\
 \hline
 1110101
 \end{array}$$

Algorithmus für $X * Y$:

1. Setze Ergebnisspeicher auf 0
2. wenn die letzte Stelle von $Y = 1$ ist, addiere X zum Ergebnisspeicher

3. Verschiebe Y um eine Stelle nach rechts (die in Schritt 1. behandelte Stelle wird "hinausgeschoben", sie verschwindet also).
4. Verschiebe X um eine Stelle nach links (entspricht Multiplikation mit 2 $\rightarrow$ stellenrichtige Addition)
5. Wenn Y nicht 0 ist, gehe zu Schritt 2.

Die 5. Anweisung hat zur Folge, dass die Multiplikation beendet wird, sobald nichts mehr zu addieren ist. Bei einer Wortbreite von n , ist die Multiplikation auf jeden Fall nach n Schritten zuende.

Division

Auch bei der Division X/Y wird nur Addition, Subtraktion (= Komplementaddition) und das Multiplizieren/Dividieren mit 2 benötigt. Zunächst wird Y in einem Hilfsregister W gespeichert und W dann solange nach links verschoben (also verdoppelt), bis $W > X$ ist. Der Divisionsrest R erhält den Wert von X . Danach wird fortlaufend W durch 2 dividiert und der Quotient mit 2 multipliziert, bis $W = X$ ist. Wenn dabei W kleiner als der Rest wird, dann wird W vom Rest abgezogen und Q erhöht (Korrektur).

Beispiel im Dezimalsystem: $651 / 5$

$$\begin{array}{rcl}
 651 : 500 & = & 1 \\
 -500 & & \\
 \hline
 151 : 50 & = & 3 \\
 -150 & & \\
 \hline
 1 : 5 & = & 0 \\
 -0 & & \\
 \hline
 1 & \rightarrow & \text{Rest}
 \end{array}
 \quad \left. \begin{array}{l} \\ \\ \\ \end{array} \right\} \rightarrow 130, \text{ Rest } 1$$

Algorithmus für X / Y :

1. Setze Quotient auf 0. Setze Rest = X . Setze Zwischenspeicher $W = Y$.
2. Verdopple W solange, bis $W > \text{Rest}$ ("wie oft geht Y in X ")
3. Wenn $W = Y$ ist, dann ist die Division zuende.
4. Verdopple den Quotientenwert ($Q := Q * 2$). Halbiere W ($W = W / 2$).
5. Wenn $W \leq \text{Rest}$ dann subtrahiere W vom Rest und Erhöhe den Quotienten um 1.
6. Fahre fort bei Schritt 3.

Der Trick hinter diesem Algorithmus ist die Gleichung

$$X = \text{Quotient} * W + \text{Rest}$$

Die Gleichung ist nach jedem Durchlauf der Rechenschritte 3. bis 6. erfüllt.

Beispiel:

Datenverarbeitungssysteme

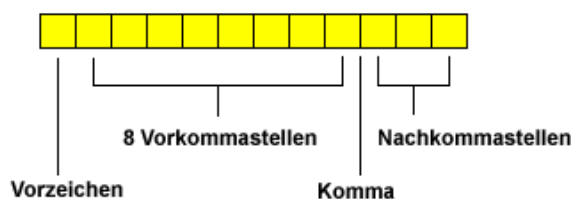
Schritt	Q	W	R	
1	0	11000	1110	Anfangswerte
	0	1100	1110	$Q \cdot 2; W \div 2$
	1	1100	10	$W \leftarrow R \rightarrow R - W,$ $W \neq Y \rightarrow \text{Schleife}$
2	1	1100	10	Ergebnis Schritt 1
	10	110	10	$Q \cdot 2; W \div 2$
	10	110	10	$W > R \rightarrow \text{nichts tun}$ $W \neq Y \rightarrow \text{Schleife}$
3	10	110	10	Ergebnis Schritt 2
	100	11	10	$Q \cdot 2; W \div 2$
	100	11	10	$W > R \rightarrow \text{nichts tun}$ $W \neq Y \rightarrow \text{Schleife}$
4	100	11	10	Ergebnis Schritt 3 $W = Y \rightarrow \text{Fertig}$

Ergebnis: $Q(\text{quotient}) = 100, R(\text{est}) = 10$

3.5 Festpunkt- und Gleitpunktdarstellung

Festpunktzahlen

Die bisher betrachtete Schreibweise gebrochener Zahlen wird Festpunktdarstellung (Festkommadarstellung) genannt. Bei einer vorgegebenen Gesamtstellenzahl steht der Radixpunkt immer an der gleichen Stelle. Wie weit die erste von Null verschiedene Ziffer vom Punkt entfernt steht, hängt von der Größe der Zahl ab, z. B. Darstellung mit Vorzeichen, 8 Stellen vor dem Komma und 3 Stellen dahinter (dezimal).



Die Zahl wird mit führenden Nullen dargestellt und gegebenenfalls gerundet. Im obigen Beispiel reicht der darstellbare Bereich von -99999999,999 bis +99999999,999. Der Nachteil liegt im begrenzten Zahlenbereich $\rightarrow$ bei sehr kleinen Zahlen gehen durch die Rundung Stellen verloren (0,0009 würde z. B. zu 0), sehr große Zahlen sind nicht mehr darstellbar.

Gleitpunktzahlen

Gebrochene Zahlen (reelle Zahlen) werden i.a. in Gleitpunktdarstellung (Gleikomma-Darst.) bearbeitet. $\rightarrow$ Gleitpunktzahlen, floating point numbers). Wegen der endlichen Stellenzahl können reelle Zahlen nur unvollkommen dargestellt werden, sie repräsentieren nur einzelne Punkte auf der reellen Zahlengerade $\rightarrow$ Rundungsfehler.

Bei der Gleitpunktdarstellung wird die Zahl so gespeichert, daß der Radixpunkt immer zur ersten von Null verschiedenen Ziffer gleitet. Dies erreicht man durch Abspalten einer

entsprechenden Potenz der Basis:

$$Z = M * B^E, n \text{ ganzzahlig}$$

$$M = 0.xxxxxxx..., 1/B \leq M < 1$$

Da die Basis bekannt ist, kann die Zahl durch die Mantisse M und den Exponenten E dargestellt werden. → halblogarithmische Darstellung → normalisierte Darstellung. Die Anpassung der GP-Zahl an die angegebene Darstellungsform wird als "Normalisieren" bezeichnet.

Beispiel:

$$Z = 42.5456 \rightarrow 0.425456 * 10^2 \rightarrow M = 425456, E = 2$$

Man bezeichnet E als **Exponent** (bzw. Charakteristik) → Größenordnung der Zahl und M als Mantisse → Angabe der gültigen Zifferndarstellungen

Die Art der Darstellung von Mantisse und Exponent ist recht unterschiedlich. Für Mantisse und Exponent wird jeweils eine feste Stellenzahl vorgegeben, in der auch das Vorzeichen von Mantisse und Exponent untergebracht werden muss. Am üblichsten ist Betrags-Vorzeichen-Darstellung der Mantisse und Exponentendarstellung mit verschobenem Wertebereich (Charakteristik) so, dass der Wert der Charakteristik immer 0 ist (so wird ein zusätzliches Bit für das Vorzeichen des Exponenten eingespart). Sie stellt also eine einfache Verschiebung des Wertebereichs dar und hat keinen Einfluss auf die Berechnung.

Beispiel:

Wertebereich Exponent:	-128 ... 127	
Charakteristik (C = E + V):	0 ... 255	(V = 128)

Umwandlung einer Dezimalzahl in eine Gleitpunkt-Dualzahl:

1. Methode:

- ◆ Getrennte Umwandlung des ganzzahligen und gebrochenen Anteils
- ◆ Verschieben des Radixpunktes bis zur ersten von Null verschiedenen Ziffer (Normalisierung).

Beispiel: (8 Stellen Mantisse, 4 Stellen Exponent, V = 8)

$$\begin{aligned} 27.75 \text{ dez.} &= 11011.11 \text{ dual} \rightarrow M = 0.1101111, E = 0101 \\ &\rightarrow M = 0.1101111, C = 1101 \end{aligned}$$

2. Methode:

- ◆ Abspalten der nächstgrößeren Zweierpotenz (d.h. dividieren durch 2, bis das Ergebnis kleiner 1 ist)
- ◆ Getrennte Umwandlung von Mantisse (echter Bruch) und Exponent (ganzzahlig).

Beispiel: (8 Stellen Mantisse, 4 Stellen Exponent, V = 8)

$$\begin{aligned} 27.75 \text{ dez.} &\rightarrow M = 0.8671875, E = 5 \\ &\rightarrow M = 0.1101111, E = 0101 \\ &\rightarrow M = 0.1101111, C = 1101 \end{aligned}$$

Eigenschaften von Gleitpunktzahlen:

- ◆ Die Gleitkommadarstellung ist immer auf eine bestimmte Zahl von Stellen genau (die Anzahl der Mantissenstellen) → bestimmte relative Genauigkeit, z.B. 32-Bit-Mantisse (einschl. VZ) → 31 Dualst. → 9 Dezimalstellen. Reelle Zahlen werden also immer auf die nächstgelegene GK-Zahl gerundet.
- ◆ Die Gleitkommadarstellung hat den Vorteil, daß man mit bestimmter Stellenzahl einen viel größeren Zahlenbereich darstellen kann, als mit Fixkommazahlen.
- ◆ Manchmal wird auch eine Darstellung verwendet, bei welcher der Radixpunkt rechts der ersten von Null verschiedenen Ziffer steht (z.B. 1.100101). Im Dualsystem ist diese Ziffer immer 1 und kann deshalb bei der Speicherung in EDV-Systemen weggelassen werden (impliziter ganzer Teil der Mantisse).

Gleitpunkt-Arithmetik

Auch hier gilt wieder, dass die Rechenverfahren verwendet werden, die wir in der Schule gelernt haben - eben nur angewendet auf Dualzahlen.

Addition und Subtraktion:

1. Angleichen der Exponenten der beiden Operanden (Verschieben der Mantisse des Operanden mit kleineren Exponenten)
2. Addition/Subtraktion der Mantissen
3. Normalisieren des Ergebnisses

Multiplikation und Division:

1. Multiplikation/ Division der Mantissen
2. Addition/ Subtraktion der Exponenten
3. Normalisieren des Ergebnisses

Beispiel für Gleitpunkt-Addition im Dualsystem:
(6 Stellen Mantisse, 4 Stellen Charakteristik)

10,11 + 111,01	VZ	Charakt.	Mantisse	
	0	1 0 0 1	1 0 1 1 0 0	X
	0	1 0 1 0	1 1 1 0 1 0	Y
1. Schritt:	0	1 0 1 0	0 1 0 1 1 0	X
(angleichen)	0	1 0 1 0	1 1 1 0 1 0	Y
2. Schritt:			1 0 1 0 0 0 0	Summe
(addieren)			Übertrag	
3. Schritt:	0	1 0 1 1	1 0 1 0 0 0	Summe
(normalisieren)				

Assoziativität und Distributivität gilt nicht mehr

Durch die begrenzte Stellenzahl entstehen u. U. Rundungsfehler beim Angleichen der Operanden (Rundungsfehler). Das Assoziativgesetz und das Distributivgesetz gelten nicht mehr.

$$a + (b + c) \neq (a + b) + c$$

$$a * (b * c) \neq (a * b) * c$$

$$(a + b) * c \neq (a * c) + (b * c)$$

Einfaches Beispiel:

10,11 + 11101	VZ	Charakt.	Mantisse
	0	0 0 0 1	1 0 1 1 0 0
	0	1 1 0 0	1 1 1 0 1 0

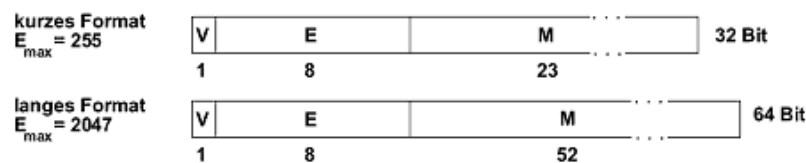
1. Schritt:	0	1 1 0 1	0 0 0 1 0 1	(1)
(angleichen)	0	1 1 0 1	1 1 1 0 1 0	

Beim Angleichen geht die letzte Stelle des ersten Summanden verloren.

Beispiel für Gleitpunkt-Formate

Bei EDV-Anlagen mit großer Maschinenwortlänge werden Mantisse und Exponent meist in einem Maschinenwort untergebracht. Die GK-Arithmetik wird meist von der Hardware unterstützt. Bei Mikrocomputern ist die GK-Arithmetik meist durch Software realisiert (Ausnahme: spezielle GK-Koprozessoren). Häufig wird neben einem Format für einfache Genauigkeit ein weiteres Format für erweiterte Genauigkeit angeboten.

Beispiel: DIN IEC 47B für Mikroprozessoren



3.6 Informationsdarstellung in Rechenanlagen

Integer-Zahlen

Integerzahlen (ganze Zahlen) werden in der Regel als konegative Dualzahlen dargestellt. Je nach Datenverarbeitungssystem (DVS) beträgt die Wortbreite 8, 16, 32, ... Bit.

Real-Zahlen

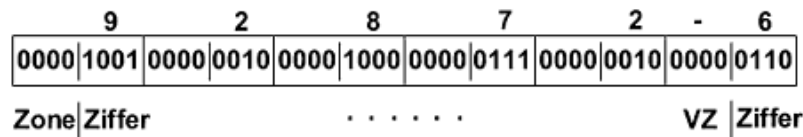
Real-Zahlen können, bedingt durch die Beschränkung auf endliche Stellenzahl, nur unvollkommen dargestellt werden → Folge sind Rundungsfehler bei der Arithmetik → numerische Mathematik. Die Darstellung erfolgt in Form von Gleitpunktzahlen (*siehe oben*) oder als BCD-Gleitpunktzahlen (*siehe unten*).

BCD - Zahlen

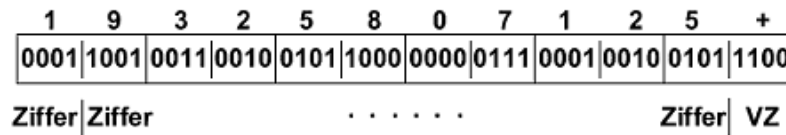
Für einige Anwendungen ist es sinnvoll und/ oder notwendig mit Dezimalzahlen zu arbeiten (z.B. kaufmännische Anwendungen, exakte Rechnung mit vorgegebener Stellenzahl). Die Dezimalzahlen werden dann als BCD-Zahlen dargestellt. Es gibt zwei Möglichkeiten der Darstellung:

- ♦ **ungepackt:** Pro Byte wird eine Dezimalziffer gespeichert (IBM 360/ 370). Das rechte Halbbyte enthält den Wert, das linke Halbbyte wird mit 0000 oder 1111 (EBCDIC) belegt ("Zone"). Vorzeichen: + = 1100, - = 1101.

Datenverarbeitungssysteme



- ♦ **gepackt:** Pro Byte zwei Dezimalziffern



Die BCD-Arithmetik ist komplizierter als rein duales Rechnen. Intern wird jedoch immer dual gerechnet. Die duale Addition zweier Tetraden (8421-BCD) erfordert immer dann eine Korrektur, wenn das Ergebnis > 9 ist. Zum Beispiel:

+5	0101		5	0101	
3	0011		+6	0110	
8	1000	richtig!	11	1011	Pseudotreadel
			+6	0110	Korrekturaddition +6!
			1	0001	Ergebnis & Übertrag

Einige Rechner kennen Befehle zur BCD-Arithmetik (automatische Korrekturaddition).

BCD-Gleitkommadarstellung:

- ♦ Mantisse (und Exponent) als BCD-Zahlen darstellen
- ♦ Normalisierung mit Zehnerpotenzen (nicht Zweierpotenzen)
- ♦ Vorzeichen entweder einzelnes Bit oder Halbbyte

Zeichen

Jedes Zeichen wird durch ein Codewort eines festgelegten Codes verschlüsselt. Die Wahl des Codes ist beliebig. Bei DVS meist ASCII (American Standard Code for Information Interchange), z. T. erweitert auf den vollen Wertebereich eines Byte (z. B. IBM-PC). Andere verwendete Zeichencodes:

- ♦ EBCDIC (Extended Binary Code for Digital InterChange)
- ♦ CDC-Display-Code (6-Bit): CDC-Cyber, keine Kleinbuchstaben
- ♦ 12-Bit Lochkarten-Code: Zur Darstellung von Informationen auf Lochkarten

Zeichenketten(Strings) werden durch aufeinanderfolgende Zeichen dargestellt. Je nach Wortlänge des DV-Systems in gepackter oder ungepackter Darstellung (z. B. 8 Zeichen in einem 64-Bit-Speicherwort). Es gibt zwei Formen der Speicherung in aufsteigender Reihenfolge im Speicher des DV-Systems:

- ♦ Reservierung des Speicherplatzes für einen String der maximal vorgegebenen Länge (= Anzahl der Codeworte). Die Anzahl der gültigen Zeichen wird im ersten Codewort gespeichert.
- ♦ Bei einem String der Länge n Reservierung von $n+1$ Codeworten. Das letzte Codewort enthält ein spezielles Abschlusszeichen (z.B. Nullwort)

ASCII Tabelle (sedezimal)

00 nul	01 soh	02 stx	03 etx	04 eot	05 enq	06 ack	07 bel
08 bs	09 ht	0a nl	0b vt	0c np	0d cr	0e so	0f si
10 dle	11 dc1	12 dc2	13 dc3	14 dc4	15 nak	16 syn	17 etb
18 can	19 em	1a sub	1b esc	1c fs	1d gs	1e rs	1f us

Datenverarbeitungssysteme

20	sp	21	!	22	"	23	#	24	\$	25	%	26	&	27	'	
28	(	29	)	2a	*	2b	+	2c	,	2d	-	2e	.	2f	/	
30	0	31	1	32	2	33	3	34	4	35	5	36	6	37	7	
38	8	39	9	3a	:	3b	;	3c	<	3d	=	3e	>	3f	?	
40	@	41	A	42	B	43	C	44	D	45	E	46	F	47	G	
48	H	49	I	4a	J	4b	K	4c	L	4d	M	4e	N	4f	O	
50	P	51	Q	52	R	53	S	54	T	55	U	56	V	57	W	
58	X	59	Y	5a	Z	5b	[	5c	\	5d	]	5e	^	5f	_	
60	`	61	a	62	b	63	c	64	d	65	e	66	f	67	g	
68	h	69	i	6a	j	6b	k	6c	l	6d	m	6e	n	6f	o	
70	p	71	q	72	r	73	s	74	t	75	u	76	v	77	w	
78	x	79	y	7a	z	7b	{	7c		7d	}	7e	~	7f	del	

Latin1-Tabelle

128	144	160	176 °	192 À	208 Đ	224 à	240 ð
129	145	161 ĩ	177 ±	193 Á	209 Ñ	225 á	241 ñ
130	146	162 ¢	178 ²	194 Â	210 Ò	226 â	242 ò
131	147	163 £	179 ³	195 Ã	211 Ó	227 ã	243 ó
132	148	164 ¤	180 ´	196 Ä	212 Ô	228 ä	244 ô
133	149	165 ¥	181 µ	197 Å	213 Õ	229 å	245 ö
134	150	166 ¦	182 ¶	198 Æ	214 Ö	230 æ	246 ö
135	151	167 §	183 ·	199 Ç	215 ×	231 ç	247 ÷
136	152	168 ¨	184 ¸	200 È	216 Ø	232 è	248 ø
137	153	169 ©	185 ¹	201 É	217 Ù	233 é	249 ù
138	154	170 º	186 º	202 Ê	218 Ú	234 ê	250 ú
139	155	171 «	187 »	203 Ë	219 Û	235 ë	251 û
140	156	172 ¬	188 ¼	204 Ì	220 Ü	236 ì	252 ü
141	157	173 –	189 ½	205 Í	221 Ý	237 í	253 ý
142	158	174 ®	190 ¾	206 Î	222 Þ	238 î	254 þ
143	159	175 ¯	191 ¿	207 Ĩ	223 ß	239 ĩ	255 ÿ

Logische Werte

Die Zuordnung zwischen internen Binärwerten kann im Prinzip beliebig vorgenommen werden (über Software). Häufig gilt: "FALSE" = "0", "TRUE" = "1".

- ♦ **ungepackte Darstellung:** Ein logischer Wert wird in einem Byte (oder auch einem Maschinenwort) gespeichert → große Platzverschwendung (hier Zuordnung oft: "TRUE" = 0, "FALSE" ungleich 0).

- ♦ **gepackte Darstellung:** Jedes einzelne Bit des Maschinenworts repräsentiert einen logischen Wert → 8 Werte/Byte → Problem des schnellen Zugriffs auf einen Wert.

Die DVS gestatten i.a. logische Operationen (AND, OR, XOR, NEG), die aber

- ♦ immer auf jedes einzelne Bit eines Bytes (Maschinenworts) wirken
- ♦ immer die obige Zuordnung log. Wert zu Binärwert voraussetzen



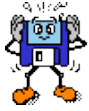
[Zum vorhergehenden Abschnitt](#)



[Zum Inhaltsverzeichnis](#)



[Zum nächsten Abschnitt](#)



Einführung Datenverarbeitungssysteme

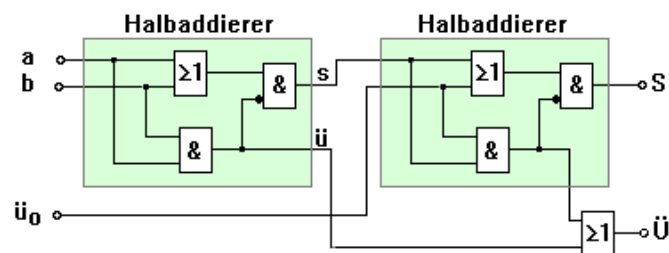
von Prof. Jürgen Plate

4. Rechenwerk

Das Rechenwerk gestattet:

- arithmetische Operationen (Addition, Subtraktion, ...)
- logische Operationen (UND, ODER, NICHT, ...)
- Verschiebe-Operationen
- u. U. Bitmanipulation
- Vergleichs- und Bit-Test-Operationen

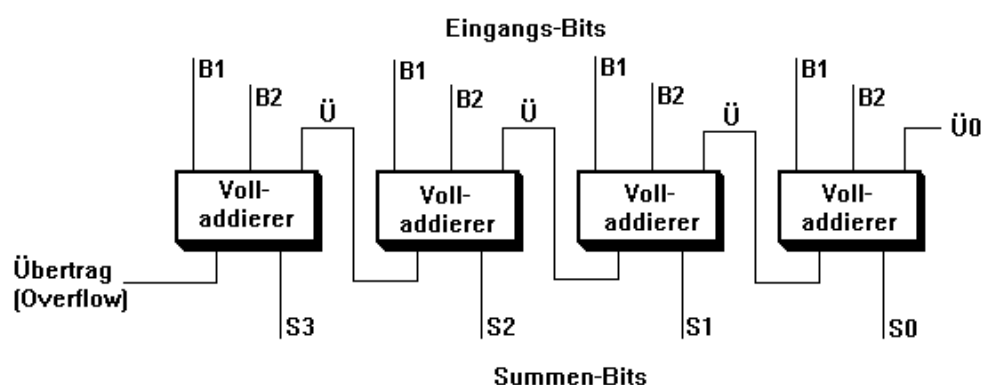
Bei einfachen CPUs besteht das Rechenwerk aus einer einzigen Einheit für die Durchführung aller Operationen - insbesondere werden alle arithmetischen Operationen auf die Addition zurückgeführt. Die Rechnung erfolgt parallel.



Binäre Addierschaltung

Größere CPUs haben ein Rechenwerk, das aus mehreren Werken (Unterwerken) für die einzelnen Operationen besteht (→ schneller, gewisse Parallelarbeit möglich). z. B. Addition, Subtraktion, Multiplikation, Division, Gleitkomma-Addition, Normalisierung, Increment, log. Verknüpfung, Verschiebung). Es gibt spezielle Prozessoren für bestimmte Operationen (z.B. Gleitkommaprozessoren), die in Zusammenarbeit mit der (Universal-) CPU deren Leistungsfähigkeit (Geschwindigkeit) steigern (Co-Prozessoren).

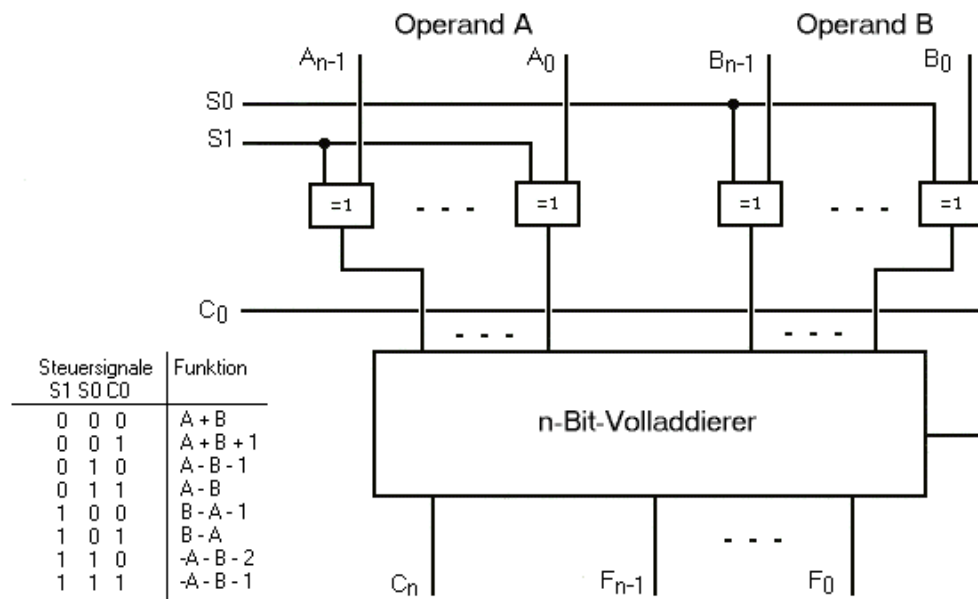
4 - Bit - Addierer



4.1 Arithmetisch-Logische Einheit (ALU)

Die ALU (Arithmetic and Logic Unit) bildet den Kern des Rechenwerks. Sie kann einfache arithmetische und logische Operationen durchführen (im wesentlichen Addition, Subtraktion, Increment, Decrement, Negieren, UND, ODER, EXOR, NEXOR (=Äquivalenz)). Heute werden i. A. parallel arbeitende ALUs verwendet → Parallelrechenwerk; früher wurden seriell arbeitende Rechenwerke gebaut (Sonderfall heute: 6804 Single-Chip). Kern der ALU ist wiederum ein Addierer.

Der in der Regel verwendete Addierer ist ein Schaltnetz zur Addition von zwei Dualzahlen, die jeweils n Bit lang sind → n -Bit-Volladdierer.

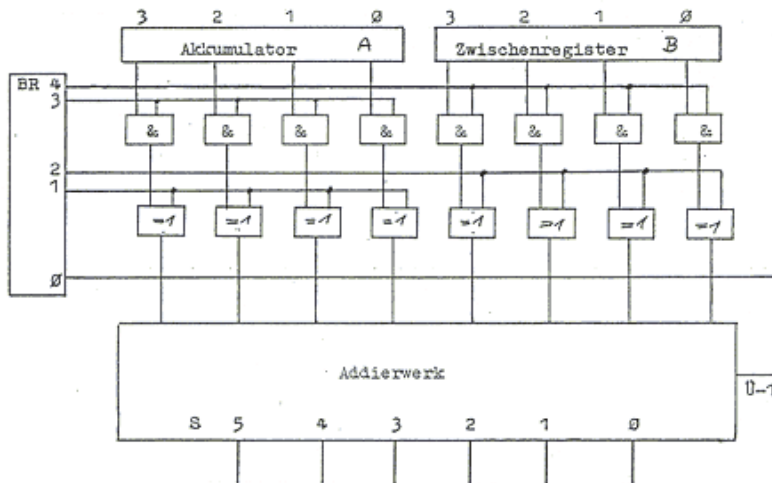


Einfache ALU

Realisierungsmöglichkeiten eines Paralleladdierers:

1. rein zweistufiges Schaltnetz
sehr schnell, sehr hoher Aufwand an Gatterfunktionen
2. iterativer Aufbau aus 1-Bit-Volladdierern
Addierer mit durchlaufendem Übertrag (Carry-Ripple-Addierer) → kleinster Aufwand, am langsamsten
3. iterativer Aufbau aus z Teiladdierern für je t Stellen ($n = z \cdot t$)
Jeder Teiladdierer ermittelt den Gesamtübertrag für t Stellen (zweistufig) und das Ergebnis wird der nächsten Teilschaltung zugeführt: Addierer mit vorab ermitteltem Übertrag (Carry-Look-Ahead-Addierer) → mittlerer Aufwand

Durch die Erweiterung des Paralleladdierers kann dann die ALU gebildet werden. Mit weiteren, teilweise in den Volladdierer integrierten Steuersignalen lassen sich weitere - auch stellenweise - logische Verknüpfungen realisieren. Zum Beispiel EXOR-Verknüpfung durch Abschalten des Übertrags von einer Stelle zur nächsten. Von der Funktion her entspricht die ALU einer Anzahl verschiedener Verknüpfungsfunktionen mit je 2 n -stelligen Eingängen A und B und einem n -stelligen Ausgang F . Dabei ist jeweils immer nur eine Funktion aktiviert. Die Auswahl der Funktion erfolgt mit einem binären Steuerwort S (Steuersignale). In der ALU können auch noch weitere Funktionen integriert werden, z.B. Increment/ Dekrement; es sind jedoch immer sehr einfache und grundlegende Funktionen. Es werden meist nicht alle der theoretisch möglichen Bitkombinationen des Steuerwortes verwendet, da nicht jede Kombination eine sinnvolle Funktion darstellt.

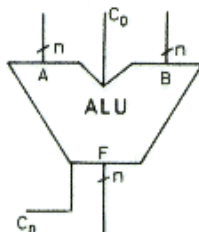


4-Bit-ALU für Addition und Subtraktion

Entstehung einiger Befehle (Steuerbits 43210):

1. Addition $S = A + B$: 11000
2. Subtraktion $S = A - B$: 11101
3. Inversion von A: 01010
4. Negation von A: 01011
5. Increment von A: 01001
6. Decrement von A: 01101

Blockschaltbild für eine ALU



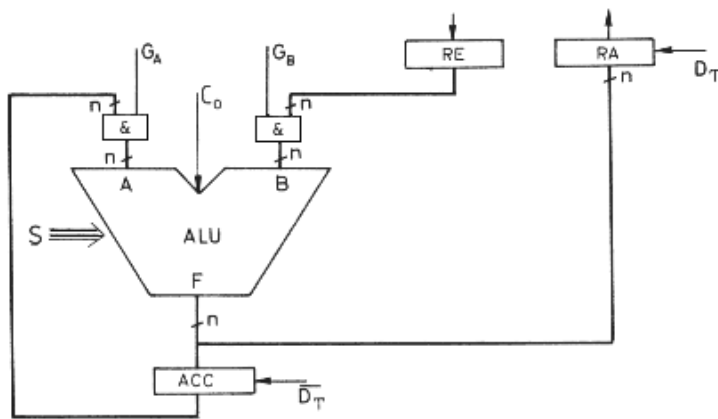
Funktionstabelle einer realisierten ALU (n=4: 74S381)

Steuercode S2 S1 S0			Funktion		Bezeichnung
			C0=0	C0=1	
0	0	0	Clear (alle 0)		
0	0	1	$B-A-1$	$B-A$	SUB (B,A)
0	1	0	$A-B-1$	$A-B$	SUB (A,B)
0	1	1	$A+B$	$A+B+1$	ADD
1	0	0	$A \oplus B$		XOR
1	0	1	$A \vee B$		OR
1	1	0	$A \wedge B$		AND
1	1	1	Preset (alle 1)		

Blockschaltung einer ALU

4.2 ALU mit Registern

Wie bei einigen Erklärungen und Abbildungen oben bereits gezeigt, benötigt man zur Zwischenspeicherung der Operanden und des Ergebnisses (und zur Verschiebung bei Multiplikation und Division) Register, die eine ALU zur Register-ALU (RALU) ergänzen.



Blockschaltung einer Register-ALU

- Puffer-Register: Eingabe-Register RE (MD-Multiplikandenreg.) und Ausgabe-Register RA
- Zwischenspeicher-Register: Akkumulator

Es ergeben sich folgende Funktionen:

- Puffer-Register: Eingaberegister dann sinnvoll, wenn - wie oft - die Operanden nacheinander in die ALU geladen werden, d.h. die ALU nur über einen Datenweg mit der Außenwelt verbunden ist (1-Adreß-Maschine).
- Zwischenspeicher-Register (Akkumulator): Speicherung des ersten Operanden und Aufnahme des Ergebnisses (RE → ALU → ACC).
- Datenwege zu den ALU-Eingängen sind über Tore (UND-Gatter) geführt, die geöffnet oder gesperrt werden können.
- Zusätzliche Steuersignale sind notwendig:
 - ◆ Ga und Gb zum Schalten der Tore am ALU-Eingang
 - ◆ Dt Richtungssteuerung des ALU-Ausgangs
 - ◆ Dt = 0: Ergebnis im Akku
 - ◆ Dt = 1: Ergebnis in RA

Mit der vorgestellten RALU lassen sich Addition, Subtraktion und bitweise Logikfunktionen programmgesteuert durchführen. Die in einem Schritt durchgeführte Operation wird vollständig durch das Steuerwort bestimmt.

Beispiel: Addition zweier Operanden: $Z = X + Y$

1. $X \rightarrow RE; 0 + \langle RE \rangle = \langle RE \rangle \rightarrow ACC$

S2	S1	S0	C0	Ga	Gb	Dt
0	0	0	0	0	1	0
2. $Y \rightarrow RE; \langle ACC \rangle + \langle RE \rangle \rightarrow RA$

S2	S1	S0	C0	Ga	Gb	Dt
0	0	1	0	1	1	1

Beispiel: Subtraktion zweier Operanden: $Z = X - Y$

1. $X \rightarrow RE; 0 + \langle RE \rangle = \langle RE \rangle \rightarrow ACC$

S2	S1	S0	C0	Ga	Gb	Dt
0	0	0	0	0	1	0
2. $Y \rightarrow RE; \langle ACC \rangle - \langle RE \rangle \rightarrow RA$

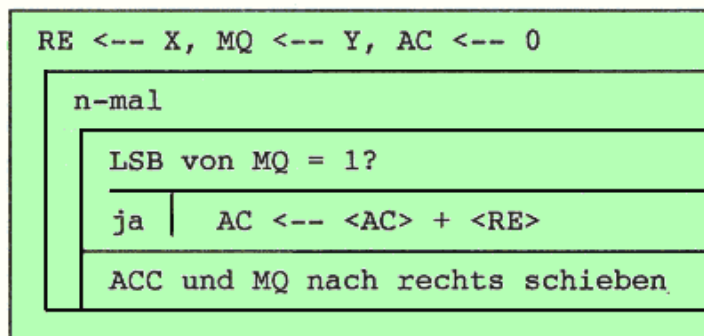
S2	S1	S0	C0	Ga	Gb	Dt
0	0	1	1	1	1	1

Die RALU ist in der bisher vorgestellten Form zur direkten Durchführung von Addition, Subtraktion und logischen Verknüpfungen geeignet. Für die Multiplikation und Division sind Erweiterungen notwendig:

- ein weiteres Register: häufig als Multiplikator-Quotienten- Register (MQ-Register) bezeichnet. In vielen CPUs wird aber auch ein zweiter Akkumulator oder ein Universalregister verwendet.
- Verschiebemöglichkeit für Registerinhalte: Ermöglichung der stellenrichtigen Addition von Teilergebnissen; häufig werden Akkumulator und MQ-Register als untereinander verbundenes Rechts-Links-Schieberegister ausgeführt.
- separate Verfügbarkeit des LSB-Stelle von MQ
- weitere Steuersignale:
 - ◆ Steuerung der Verschiebeoperation SHL (links), SHR (rechts)
 - ◆ Multiplexer-Steuersignal für ALU-Ergebnis auf Akku oder MQ
 - ◆ Ersatz Richtungssteuersignal Dt durch 2 Signale Dt0 und Dt1, da jetzt die ALU auf 3 Register geschaltet werden kann.

4.3 Realisierung von Multiplikation und Division

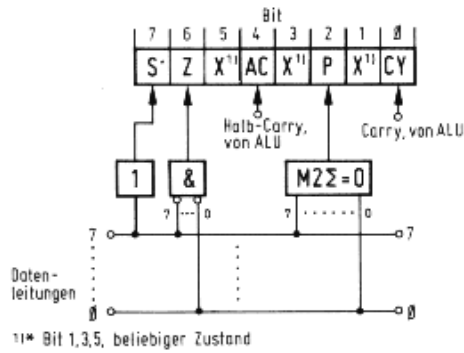
Multiplikation durch Addition und Verschiebung (Verschieben nach links = Multiplikation mit 2). Wie beim Rechnen von Hand wird der Multiplikand mit den einzelnen Stellen des Multiplikators mal genommen und die Teilergebnisse (Partialprodukte) dann stellenrichtig summiert (Verfahren *siehe Teil 1 der Vorlesung*). Bei der Division wird durch Verschieben nach rechts die Division durch 2 realisiert. Das Verfahren wird hier nicht weiter vertieft.



Schema der Multiplikation

4.4 Ergänzungen zur RALU

- Statusregister (Condition Code Register):
Dieses Register enthält einzelne Bits (Flip-Flops) zur Anzeige bestimmter Zustände (Parameter) des Rechenwerks. Die einzelnen Werte des Statusregisters können in den folgenden Programmbefehlen weiterverwendet werden (z.B. bedingte Sprünge).



Der Inhalt des Statusregisters wirkt auf das Leitwerk ein.

- ◆ C- (Carry-) Bit: zeigt den Übertrag in die (n+1)-te Stelle an (kein Kennzeichen für Bereichsüberschreitung). Wird z.B. bei der Komplementaddition gesetzt.
- ◆ V- (Overflow-) Bit: zeigt Zahlenbereichsüberschreitung an.
- ◆ Z- (Zero-) Bit: zeigt an, ob das Ergebnis der vorhergehenden Rechenoperation Null ist (wird 1, wenn AC-Inhalt = 0 ist).
- ◆ N- (Negative-) Bit: zeigt an, ob das Ergebnis der vorhergehenden Rechenoperation negativ ist (MSB des AC = 1).

Je nach Prozessortyp gibt es weitere Flags, z.B.:

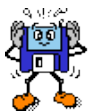
- ◆ H- (Half Carry-) Bit: zeigt Übertrag zwischen dem niederwertigen und höherwertigen Halbbyte an. Für Dezimalkorrektur bei der Umwandlung binär zu BCD.
- ◆ P- (Parity-) Bit: zeigt je nach Prozessor gerade/ungerade Parität des AC an (Anzahl der 1-Bits gerade/ungerade).
- Vielfältige Verschiebemöglichkeiten
Zum Beispiel Rotieren des AC (das an einem Ende herausgeschobene Bit wird am anderen Ende des AC wieder eingespeist - Prinzip Ringzähler). Verschieben und Rotieren unter Einbeziehung des Carry-Bits → ermöglicht Rechenoperationen über mehrere Worte hinweg.
- Vergleicherschaltungen
Vergleich von Registerinhalten - setzen von Z- und N-Bit des Statusregisters. Die Werte von Z- und N-Bit können dann von bedingten Sprungbefehlen ausgewertet werden. Die Vergleicherschaltung wird nicht nur bei expliziten Vergleichsbefehlen, sondern auch bei Transportbefehlen aktiv.

[◀ Zum vorhergehenden Abschnitt](#)

[▲ Zum Inhaltsverzeichnis](#)

[▶ Zum nächsten Abschnitt](#)

Die HTML-Fassung entstand unter Mitwirkung von Volker Arndt
Copyright © FH München, FB 04, Prof. Jürgen Plate



Einführung Datenverarbeitungssysteme

von Prof. Jürgen Plate

5. Befehle (Maschinensprache)

5.1. Befehlsaufbau und Befehlsformate

Die Befehle, die der Computer versteht und ausführen kann sind Bitfolgen → Maschinensprache, für jede CPU unterschiedlich. Der Befehl muss zwei Angaben enthalten (v. Neumann-Rechner):

- durchzuführende Operation (Was!)→ Operations-Code
- verwendeter Operand (Womit!)→ Adresse

OP-Teil	Adress-Teil ?
----------------	----------------------

Manchmal enthält der OP-Teil noch ein Indikator-Feld für spezielle Angaben (z.B. Adressierungsmode); häufig gehen diese Angaben auch direkt in den Op-Code ein. Nach der Anzahl der Operanden im Adressteil unterscheidet man:

Ein-Adress-Befehle	<table><tr><td>OP-Code</td><td>Adresse ?</td></tr></table>	OP-Code	Adresse ?
OP-Code	Adresse ?		

Zwei-Adress-Befehle	<table><tr><td>OP-Code</td><td>Adresse 1</td><td>Adresse 2</td></tr></table>	OP-Code	Adresse 1	Adresse 2
OP-Code	Adresse 1	Adresse 2		

Drei-Adress-Befehle	OP-Code	Adresse 1	Adresse 2	Adresse 3
---------------------	---------	-----------	-----------	-----------

- 8-Bit-Mikrocomputer arbeiten fast immer mit Einadress-Befehlen (1. Quelle und Zieladresse implizit gegeben, z.B. Akkumulator), 16-Bit-CPU's dagegen oft mit Zweiadress-Befehlen.
- Ein Befehl kann aus einem oder mehreren Speicherworten bestehen
- Bei Computern mit großer Wortlänge können auch mehrere Befehle in einem Speicherwort stehen
- Bei 8-Bit-Mikros besteht der OP-Code aus einem Byte, der Adress-Teil aus einem oder zwei Bytes (= eine Adressangabe)
- Es gibt auch Befehle, die keinem Adressteil benötigen, z.B. weil die Adressangabe implizit gegeben ist.

Die Einadress-CPU hat ein spezielles Register, in dem die Datenmanipulations-Operationen durchgeführt werden, den Akkumulator (Akku, Accu). Der erste Operand muss immer bereits im Akku stehen (→ "Lade"-Befehl) und nach der Operation befindet sich das Ergebnis wieder im Akku. Das Maschinenprogramm eines Rechners läuft in der Regel nicht auf einem Rechner mit anderer CPU. Ausnahme: (aufwärts-)kompatible Rechnerfamilien.

Vereinfachung der binären Darstellung

Die binäre Darstellung ist (für Menschen) unübersichtlich. Für die Darstellung der Maschinenbefehle haben sich daher Vereinfachungen entwickelt:

- dezimale Schreibweise
- symbolische Schreibweise → mnemotechnische Schreibweise → Assemblersprache

Es existiert ein Dienstprogramm, das den Quelltext in Assemblersprache in die maschineninterne Darstellung übersetzt (der "Assembler"). Eine Zeile im (Assembler-) Quellprogramm wird in einen Maschinenbefehl übersetzt.

5.2 Befehlsklassen, Befehlslisten

Der Aufbau der Befehle sowie Art und Umfang des Befehlsvorrates sind (neben technologischen Unterschieden) die wichtigsten Kennzeichen eines Prozessors. Die Befehlsliste legt die Menge der ausführbaren Operationen fest und bestimmt damit die Leistungsfähigkeit der CPU. Der Befehlsvorrat typischer Universalrechner umfasst zwischen 50 und 500 Befehlen. Hierzu kommen noch Modifikationen infolge unterschiedlicher Adressierungsarten.

Typische Befehlsklassen

- Die Aufzählung der Befehlsklassen ist nicht vollständig.
- Nicht jeder Rechner enthält Befehle aller Klassen.
- Einige Klassen sind nicht immer eindeutig gegeneinander abgrenzbar.
- Mit "m" wird eine beliebige Speicherzelle bezeichnet, mit "adr" eine beliebige Adresse.
- Die Befehle sind als illustrative Beispiele gedacht und den Assemblersprachen verschiedener Prozessoren entnommen.

Folgende Befehlsklassen sind repräsentativ f&uumL;r viele CPUs:

1. Transportbefehle

Lade Register A aus m	Load	LDA m
Speichere Register A in m	Store	STA m
Transportiere von m2 nach m1	Move	MOV m1,m2
Lade 0 in Register A	Clear	CLRA

2. Ein-/Ausgabebefehle

Lade A vom Kanal (Port) n	Input	IN A,n
Gib A aus auf Kanal (Port) n	Output	OUT A,n

3. Arithmetische Befehle

Addiere Speicherinhalt zu A	Add	ADDA m
Subtrahiere Speicherinhalt von A	Subtract	SUBA m
Multipliziere A mit m	Multiply	MULA m
Erhöhe A um 1	Increment	INCA
Vermindere A um 1	Decrement	DECA

4. Logische Befehle

A UND Speicherinhalt	And	AND A m
A ODER Speicherinhalt	Or	ORA m
Komplementierung von A	Complement	COMA

5. Verschiebe-Befehle

Rechtsschieben, arithmetisch	A. shift right	ASRA
Rechtsschieben, logisch	L. shift right	LSRA
Linksrotieren	Rotate left	ROLA

6. Bitmanipulationsbefehle

Setze Bit Nr. n	Bit set	BSET n,m
Teste Bit n auf 1	Bit test	BIT n,m

7. Vergleichsbefehle

Vergleiche A mit m	Compare	CMPA m
Teste auf "0" oder "Minus"	Test	TST m

8. Sprungbefehle

Unbedingter Sprung	Jump	JMP adr
Sprung bei Gleichheit	Jump if equal	JE adr
Sprung bei Ungleichheit	J. if not eq.	JNE adr
Sprung ins Unterprogramm	Jump to Subr.	JSR adr
Rückkehr vom Unterprogramm	Return from S.	RET
Software-Interrupt	Software int.	SWI
Rückkehr vom Interrupt	Return from I.	IRET

9. Befehle zur Beeinflussung des Systemzustandes

Interrupts verbieten	Disable Int.	CLI
Interrupts zulassen	Enable Int.	STI
Keine Operation	No Operation	NOP
Anhalten (und Warten auf Int.)	Halt	HLT

5.3 Adressierungsarten

Der Befehl muss zwei Angaben enthalten (v. Neumann-Rechner):

1. durchzuführende Operation (Was!) → Operations-Code
2. verwendeter Operand (Womit!) → Adresse

Bisher wurde davon ausgegangen, daß bei der Adressangabe eine bestimmte Speicheradresse angesprochen wird, in welcher der gewünschte Operand zu finden ist. Häufig enthält der Adressteil nur eine Teil-Information, mit deren Hilfe sich die tatsächliche, die effektive Adresse, ermitteln läßt → Adressmodifikation.

Die Adresse muss sich nicht unbedingt auf den Arbeitsspeicher beziehen, sie kann sich auch auf ein Register (Zwischenspeicher in der CPU) oder das E/A-Werk (Memory Mapped I/O) beziehen. Weiterhin ist eine verkürzte Adressangabe möglich → kürzere Befehle. Die Adresse kann während des Programmablaufs in Abhängigkeit von gewissen Bedingungen verändert werden (→ flexible und effektive Programmierung).

Vor allem bei CPUs mit größeren Wortbreiten enthält der Befehl einen eigenen Modifikatorteil (dem Adressteil zugeordnet). Bei kleineren Rechnern mit Einadreß-Befehlen (z.B. 8-Bit-Mikrocomputern) wird dagegen häufig (nicht immer) kein Modifikatorteil verwendet, sondern die Adressmodifikation implizit durch den OP-Code gegeben → OP-Code legt Operation und Adressierungsart fest.

Im Folgenden werden die wesentlichen Adressierungsarten vorgestellt. Die Bezeichnung durch die einzelnen Hersteller ist nicht einheitlich. Auch nicht jede CPU besitzt alle Adressierungsarten und häufig sind nicht alle Adressierungsarten mit jedem Befehl oder jedem Register möglich.

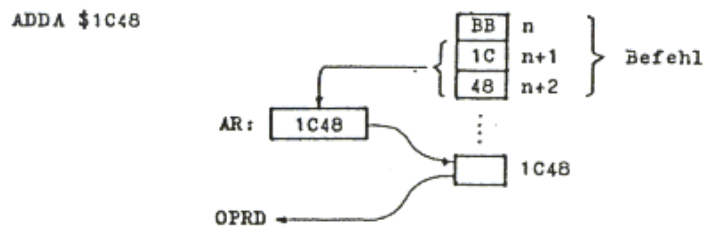
Programmiermodell

Bedingt durch die komplexe Struktur der Hardware einer CPU betrachtet man bei der Programmierung nicht mehr den realen Aufbau aus Leitwerk, Rechenwerk, E/A-Werk und Speicher. Die CPU wird nur noch als Programmiermodell betrachtet, das aus den "von außen sichtbaren" Teilen von Leit- und Rechenwerk, der Registerstruktur und der Befehlsliste besteht. Speicher und E/A-Werk werden auf die gleiche Weise betrachtet (z.B. Programmiermodell für komplexe E/A-Bausteine).

Absolute Adressierung

Der Adressteil (AD) des Befehls stellt die effektive Adresse (EA) dar. Die EA enthält den Operanden des Befehls, sie kann Speicher- oder Registeradresse sein.

- ♦ **Speicheradresse:** In manchen Rechnern existieren zwei Formen, die sich durch die Länge des Adressteils unterscheiden:
 - ◊ vollständig absolut(6009: Extended, Ext. Direct, 68000: Absolute Long): Der Adressteil kann den vollen Adressraum beschreiben (bei 8-Bit-µP 2 Byte: (OP-Code, Adr.-MSB, Adr.-LSB).
 - ◊ abgekürzt absolut(6809: Direct, 68000: Absolute Short): Der Adressteil kann nur einem eingeschränkten Adressraum beschreiben (meist bei Sprungbefehlen).



- ♦ **Registeradresse:** Da in der Regel nur wenige adressierbare Register vorhanden sind, "kurze" Angabe.

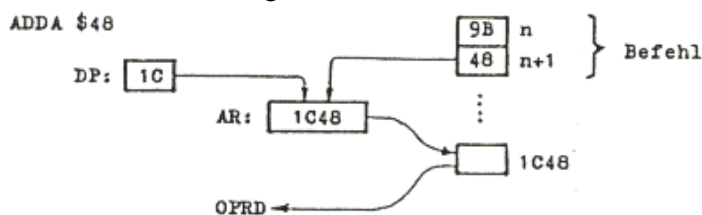
Seitenbezogene Adressierung

Der Speicher wird in einzelne Seiten (Pages) eingeteilt. Die EA setzt sich aus zwei Teilen zusammen:

- ♦ p höherwertige Stellen: Seitenadresse (SA)
- ♦ w niederwertige Stellen: Wortadresse (WA)

Durch den Adressteil des Befehls wird nur die Wortadresse festgelegt (AD kann daher kurz sein). Die Seitenadresse kann:

- ♦ in speziellen Registern stehen (Direct Page Reg. bei 6809)
- ♦ den höherwertigen Stellen des Befehlszählers entnommen werden



Vorteil:

Trotz verkürzter Adressangaben kann der gesamte Adressraum erreicht werden.

Adressbestimmung schnell (ohne Addition), solange die aktuelle Seite nicht verlassen wird.

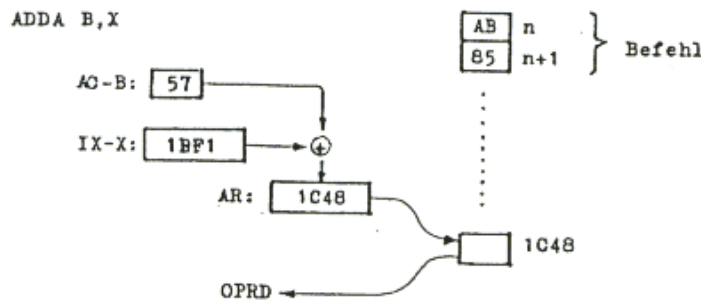
Anwendungen:

- ♦ Verkürzung der erforderlichen Adressangabe (Speicherzugriffe!)
- ♦ gemeinsame Behandlung von Speicherinhalten in Blöcken (Felder!)
- ♦ Zugriff-Schutz

Indizierte Adressierung

Die EA wird durch Addition einer *Distanz D* zum Inhalt eines *Bezugsregisters* (*Adreßregister, Indexregister, etc.*) ermittelt. Die Distanz (Offset, Displacement) kann sein:

- ◆ Adressteil des Befehls
- ◆ Der Inhalt eines durch den Adressteil spezifizierten Registers
- ◆ Die Summe eines Registers und des Befehls-Adressteils
- ◆ keine Distanz ($D=0$)
- ◆ Als Register kann auch der Befehlszähler verwendet werden
- ◆ positiv und negativ

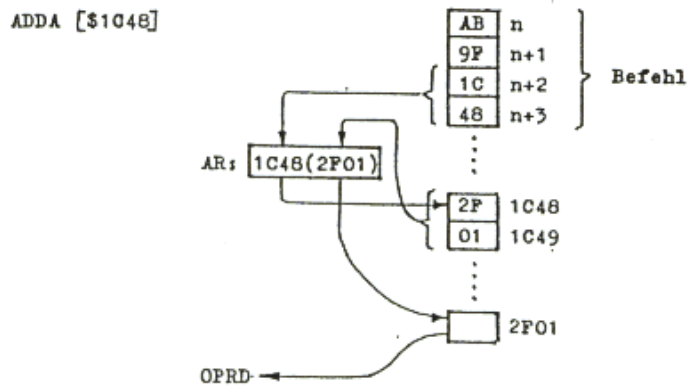


Die indizierte Adressierung wird nach verwendetem Bezugsregister und Ermittlung der Distanz unterschieden:

- ◆ Adressierung über spezielle Indexregister (Indexed Addressing)
erfolgt über ein spezielles Indexregister (6800), mehrere Indexregister (6809) oder ein/mehrere Universalregister (68000). Anwendungen:
 - ◇ Bearbeitung von Datenblöcken (Felder, Strings) durch Incrementierung/Decrementierung des Indexregisters (bei manchen Mikrocomputern automatische Veränderung des Indexregisters)
 - ◇ Implementierung von Schleifen bei der Übersetzung von höheren Programmiersprachen
- ◆ (Befehlszähler-) Relative Adressierung (PC Relative Addressing)
Der Befehlszähler ist Bezugsregister. Vorteil: Programm im Speicher frei verschieblich.
- ◆ Basis- (Distanz-) Adressierung
Ein Basisregister ist Bezugsregister. Häufig sind mehrere Basisregister vorhanden (IBM 360: 16 Basisregister); in diesem Fall muss das gewünschte BR angegeben werden. Die Programme sind so geschrieben, dass alle Adressangaben relativ zum Programmanfang (Adr. 0) erfolgen. Die aktuelle Anfangsadresse steht dann im Basisregister. Beim Laden des Progr. erhält das Basisregister einen konstanten Inhalt (Progr.-Startadresse) zugewiesen.

Indirekte Adressierung

Alle Adressierungsarten, bei denen die Adressierung über zwei oder mehr Referenzstufen erfolgt → Zum Lesen/Schreiben des tatsächlichen Operanden sind zwei oder mehr Speicherzugriffe erforderlich. Im Befehl wird nicht der Operand direkt adressiert, sondern die Speicherzelle, in der sich die Adresse des Operanden befindet. In Mikrocomputer meist nur Adressierung über zwei Referenzstufen, bei Großrechnern oft mehr als zwei Stufen.



Die indirekte Adressierung kann mit den o. g. Adressierungsarten kombiniert werden, sofern der Aufbau der CPU dies zulässt.

- ♦ Universell verwendbare Programme oder Unterprogramme → UP als Parameter. Die effektive Adresse muss bei der Programmierung nicht bekannt sein (indirekt-absolute A.), es genügt ein einziger Einsprungspunkt und eine Funktionsnummer.
- ♦ Bearbeiten von Datenblöcken mit unterschiedlicher Länge und beliebiger Lage im Speicher (indirekt indizierte A.).
- ♦ Zugriff auf Speicherplätze, die außerhalb eines eingeschränkten Speicherbereichs liegen (bei Rechnern, die nur abgekürzt-absolute Adressierung kennen, z.B. PDP-8) oder die außerhalb der aktuellen Seite liegen (seitenbezogene Adressierung).

Unmittelbare Adressierung (Immediate Addressing)

Der Adressteil stellt unmittelbar den Operanden dar (der Operand folgt unmittelbar dem OP-Code). Der Wert des Operanden wird bereits zum Zeitpunkt der Programmierung festgelegt und kann zusammen mit dem Programm gespeichert werden.

- ♦ Konstanten-Adressierung $OPRD = AD$
- ♦ unmittelbare Adressierung $OPRD = (PC+1)$



Implizite Adressierung (Inherent Addressing)

Alle Adressangaben sind implizit im OP-Code enthalten (meist Register). Zusätzliche Adressangabe nicht erforderlich, z.B. "Setze Akkumulator auf Null".

[◀ Zum vorhergehenden Abschnitt](#)

[▲ Zum Inhaltsverzeichnis](#)

[▶ Zum nächsten Abschnitt](#)

Die HTML-Fassung entstand unter Mitwirkung von Volker Arndt
Copyright © FH München, FB 04, Prof. Jürgen Plate



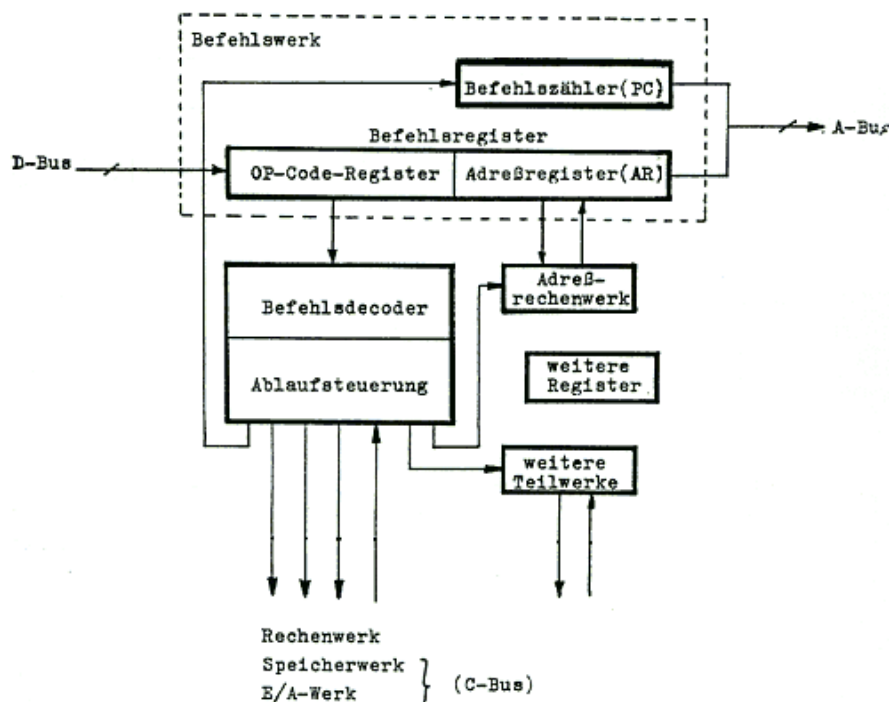
Einführung Datenverarbeitungssysteme

von Prof. Jürgen Plate

6. Leitwerk

Das Rechenwerk verarbeitet Daten(-worte) aus dem Speicher - das Leitwerk verarbeitet Befehle (Befehlswords) → Befehlsprozessor. Das Leitwerk wird auch als Steuerwerk bezeichnet. Das Leitwerk hat folgende Aufgaben:

- Steuerung des Ablaufs der Befehlszyklen in der CPU
(Steuerung des Programmablaufs)
 - ♦ Befehle holen und decodieren
 - ♦ Operanden holen/speichern
- Steuerung der Befehlsausführung
Koordination und Steuerung der einzelnen Werke der CPU



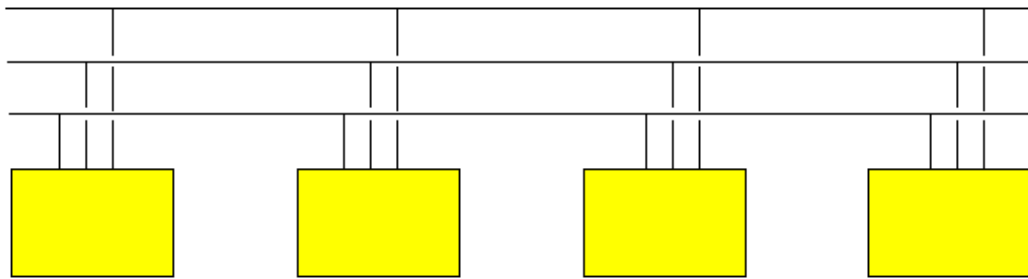
Diese Kommunikation erfordert Steuer- und Meldeleitungen:

- Direkte Verbindung von Leitwerk und Rechenwerk über eine Reihe von Steuer- und Statusleitungen.



- Verbindung von Leitwerk und Speicherwerk EA-Werk über Steuerleitungen oder Bus.

Datenverarbeitungssysteme



Von der Funktion her ist das Leitwerk in eine Reihe von Unterwerken zu gliedern:

- Befehlswerk
- Befehls-Dekoder und Ablaufsteuerung
- Adressrechenwerk
- Unterbrechungswerk
- Bus-Steuerwerk

Einige Komponenten des Leitwerks haben Sie bereits bei der Betrachtung des Befehlszyklus und der Adressierungsarten kennen gelernt:

- Befehlszähler (PC, BZ)
- Befehlsregister (IR) = Opcode-R. (OR) + Adressregister (AR)
- Befehls-Dekoder und Ablaufsteuerung
- Indexregister (Adressmodifikation)

Befehlszähler und Befehlsregister bilden das Befehlswerk, das den Ablauf des Befehlszyklus festlegt:

1. Befehlshol-Phase:
 $\langle PC \rangle = \text{Adr. des zu holenden Befehls}$
 $PC = \langle PC \rangle + 1$
2. Befehlsausführungs-Phase:
 $\langle IR \rangle = \text{auszuführende (geholter) Befehl}$
Das Instruktionsregister (IR) wird unterteilt in Operandenregister (OR) und Adressregister (AR). Das OR nimmt den Befehlscode auf (was zu tun ist) und das AR die Adresse des Operanden (womit ist es zu tun).
 $\langle OR \rangle \rightarrow \text{Befehlsdecoder} \rightarrow \text{Ablaufsteuerung}$
 $\langle AR \rangle \rightarrow \text{Adressrechnung (Ermitteln effektiver Adresse)}$

Das Befehlswerk legt fest, welcher Befehl als nächster zu holen ist und nimmt ihn zur Verarbeitung auf. Gesteuert wird das Befehlswerk und die Ausführung des Befehls durch die Ablaufsteuerung.

6.1 Ablaufsteuerung und Befehlsdecoder

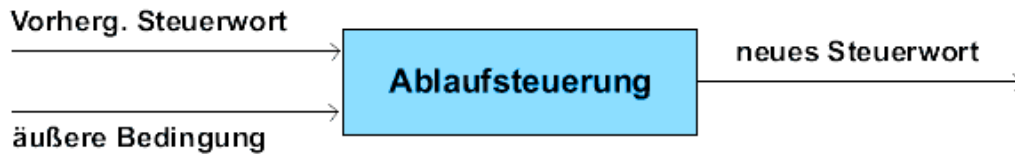
Sie ist das Herz des Leitwerks: Erzeugung aller in den übrigen Unterwerken des Leitwerks benötigten Steuersignale.

- Steuerung und Überwachung aller Operationen, die die CPU beherrscht
- Realisierung des Befehlszyklus, Steuerung der Befehlsausführung. Üblicherweise sind die einzelnen Steuersignale während eines Taktzyklusses konstant
- synchroner Ablauf

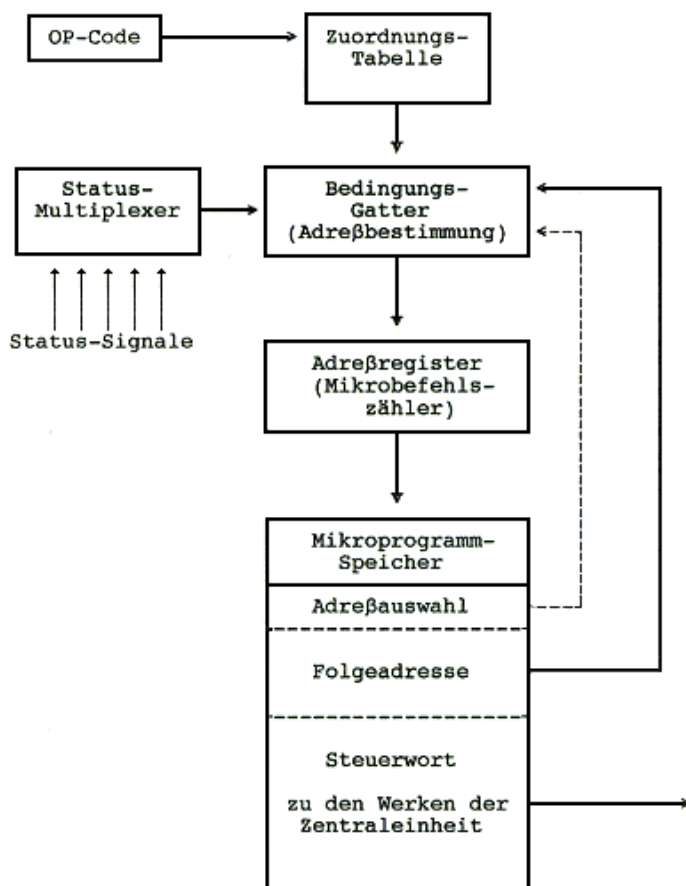
Die Gesamtheit der Steuersignale bildet ein Steuerwort, das in einzelne Felder unterschiedlicher Länge aufgeteilt wird. Jedes Feld enthält die Steuersignale für eine bestimmte Einheit der CPU (z.B. ALU). Durch das Steuerwort werden die bei jedem Taktzyklus in den einzelnen Werken/Unterwerken

auszuführenden Teiloperationen bestimmt. Zu jedem Taktzyklus muss ein neues Steuerwort erzeugt werden.

Dabei kann jedes Steuerwort vom vorhergehenden Steuerwort und von äußeren Bedingungen abhängen. Die äußeren Bedingungen können z.B. der Op-Code und/oder Modifikatorteil des auszuführenden Befehls oder Status-Signale sein → Ablaufsteuerung ist synchrones Schaltwerk.



Die Anzahl der Taktzyklen zur Ausführung eines Befehls hängt vom jeweiligen Befehl ab (ca. 1 ... 170 Taktzyklen). CPUs haben typisch 50 bis einige 100 Befehle → Ablaufsteuerung hat dann 500 bis einige 1000 verschiedene Steuerworte. Daher erfolgt heute bei CISC die Realisierung als mikroprogrammiertes Schaltwerk. Die einzelnen Steuerworte stehen dann in einem Festwertspeicher (ROM) - siehe Umdruck. Die Auswahl eines bestimmten Steuerworts erfolgt durch Anlegen seiner Adresse. Das adressierte Steuerwort wird aus dem Speicher ausgelesen und die einzelnen Steuersignale über direkte Leitungen auf die Steuereingänge der entsprechenden Werke geschaltet. Die Auswahl des nächsten Steuerwortes erfolgt dann durch Auswahl der Folgeadresse → Sequencer.



Normalfall: Sequentielle (lineare) Folge von Steuerworten im Mikroprogrammspeicher.

Die Folgeadresse kann von der linearen Folge abweichen, wenn z. B.:

- Ein neuer Maschinenbefehl begonnen wird
- Statussignale zu berücksichtigen sind (bedingte Sprünge)

- im vorhergehenden Steuerwort ein Mikroprogrammsprung ausgeführt wurde

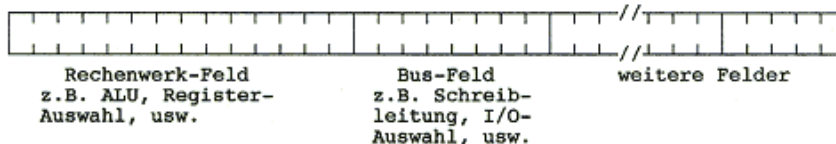
Jedes Steuerwort wird um einen Folgeadresteil (Adressauswahlfeld und Sprungadresteil) erweitert. → Ein derart ergänztes Steuerwort heißt Mikrobefehl. → Die für die Durchführung eines bestimmten CPU-Befehls erforderliche Folge von Mikrobefehlen heißt Mikroprogramm. In diesem Zusammenhang wird der CPU-Befehl manchmal auch Makrobefehl genannt. Für die Realisierung der Adressbestimmung existieren eine Reihe unterschiedlicher Konzepte. An dieser Stelle soll beispielhaft nur ein relativ einfaches Konzept vorgestellt werden. Die Startadresse jedes Mikroprogramms wird durch den Op-Code des entsprechenden CPU-Befehls festgelegt. Da die Startadressen der einzelnen Mikroprogramme weiter auseinander liegen, wird ein Zuordnungsspeicher (Mapping ROM) zwischengeschaltet.

Die jeweils nächste Adresse kann dann bestimmt werden durch:

- Incrementieren des Mikroprogrammzählers (linearer Ablauf)
- Sprungadresse(n) aus dem Sprungadresteil des aktuellen Mikrobefehls
- Zuordnungstabelle (Mapping ROM)
- Die Auswahl der Adressquelle erfolgt:
 - mit Adressauswahlfeld
 - und durch Statussignale (Bedingungsgeber)

Zum Beispiel kann das Vorliegen einer Programmunterbrechung (bzw. PU-Anforderung) durch die Abfrage eines Status-Signals überprüft werden. Am Ende der Befehls-Ausführungsphase muss ein entsprechender Mikrobefehl stehen. Der letzte Mikrobefehl jedes CPU-Befehlszyklus muss ein unbedingter Sprung in die Mikrobefehls-Sequenz der Befehls-Holphase sein.

Charakteristisch für Mikroprogrammspeicher ist die relativ geringe "Programmlänge" für die Bearbeitung eines CPU-Befehls (typisch ca. 500 bis 2000 Mikrobefehle) und die große Wortbreite (typisch 32 .. 128 Bit, bei IBM-Großrechnern mehrere 100 Bit).



Im allgemeinen ist bei heutigen Rechnern die Ablaufsteuerung für den Benutzer unzugänglich und kann auch nicht geändert werden. Es gibt jedoch Prozessoren (z.B. Bit-Slice-P. AM 29xx-Serie) und CPUs, bei denen der Benutzer die vorhandenen Mikroprogramme selbst erstellen oder ergänzen kann → speziell angepasste Maschinenbefehle → effektivere Programmierung und Einsatz des Rechners → "Wunsch-CPU". Durch Verwendung von speziellen IC-Bausteinen wäre sogar eine CPU denkbar, die während der Abarbeitung eines Programms die CPU-Befehlsliste ändern kann.

Bei RISC-Prozessoren wird zur schnelleren Ablaufsteuerung ein anderer Weg beschritten. Hier besteht die Ablaufsteuerung aus einer fest verdrahteten Logik. Zusammen mit einer passenden Architektur der übrigen Werke gelangt man so zu Prozessoren, die einen Befehl innerhalb eines Taktzyklusses ausführen können.

6.2 Unterbrechungs-Behandlung

Die Komponenten des Leitwerks, die der Unterbrechungs-Behandlung dienen, werden als Unterbrechungswerk bezeichnet (Unterbrechung = Interrupt).

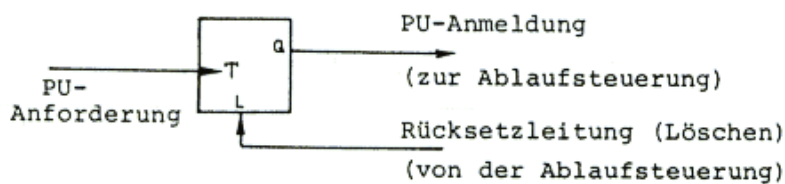
Aufgabe des Unterbrechungs-Werks:

Datenverarbeitungssysteme

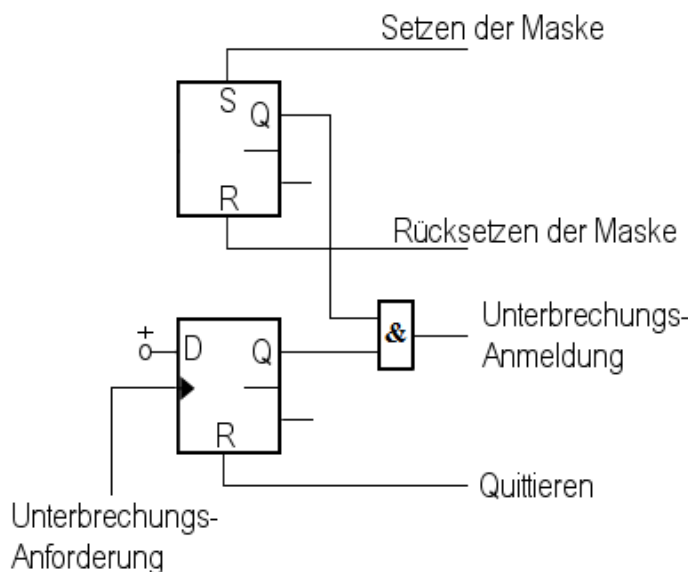
- Auswertung der Programmunterbrechungs-Anforderung, die über Signalleitungen von außen kommt.
- Anmeldung der PU bei der Ablaufsteuerung über eine Status-Signal
- Unterstützung der Ausführung der PU

Die eigentliche Durchführung der PU wird in der Regel von der Ablaufsteuerung durchgeführt. Die PU-Anforderung (interrupt request, IRQ) kommt auf einer oder mehreren Leitungen an die CPU.

Einfachster Fall ist eine einzige IRQ-Leitung. Diese führt auf ein Flipflop, das durch die IRQ-Anforderung gesetzt wird. Nach Annahme der PU durch die Ablaufsteuerung wird das FF durch die Ablaufsteuerung zurückgesetzt.



Häufig ist es sinnvoll, die sofortige Ausführung einer Unterbrechung zu verhindern (u. a. dann, wenn gerade eine Routine zur PU-Behandlung abläuft). Die PU-Anforderung wird gesperrt (interrupt disable) → maskierbarer Interrupt. Nur, wenn das Masken-FF gesetzt ist, führt eine PU-Anforderung auch zu einer PU-Anmeldung (d.h. die PU wird akzeptiert). Das Masken-FF kann per Software gesetzt oder gelöscht werden.



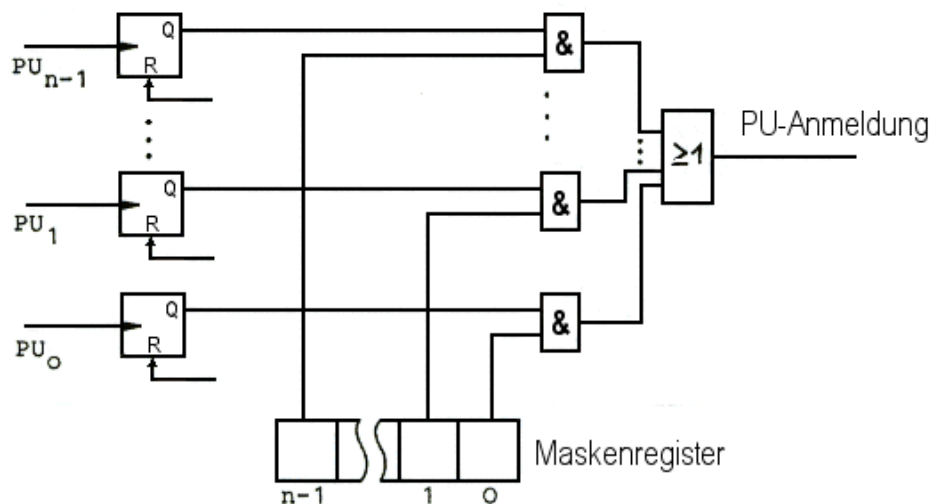
Als Reaktion auf eine PU erfolgt:

- Retten des aktuellen Programmzustands: Der Maschinenstatus, d.h. der Inhalt der Register der CPU muss festgehalten werden (→ Zustandsvektor). Speicherung meist im Kellerspeicher (Stack). Teils per HW (PC, Status), teils per SW (z.B. übrige Register).
- Laden der Register mit dem Zustandsvektor der ISR: Der Zustandsvektor steht an einer festgelegten Adresse im Speicher (→ Interrupt-Vektor). Das Programm wird nun ab der Startadresse der ISR fortgesetzt.

Nach Ende der ISR wird der ursprüngliche Zustandsvektor wieder vom Stack geholt und das Programm an der Unterbrechungsstelle fortgesetzt. Während des Rettens des Programmzustands und des anschließenden Ladens des Zustandsvektors der ISR darf keine neue Unterbrechungsanforderung

auftreten (Zustandsinformation unvollständig!).

Fast immer gibt es mehrere Quellen für PU-Anforderungen. Viele CPUs besitzen mehrere IRQ-Eingänge (aber man kann über zusätzlichen HW-Aufwand auch bei einem einzigen IRQ-Eingang mehrere PU-Quellen bedienen). Die Freigabe der einzelnen IRQ-Eingänge erfolgt über ein spezielles Register, das IRQ-Masken-Register (dem Masken-FF entsprechend). Das Masken-Register kann wieder per SW geladen werden. Neben den maskierbaren IRQ-Eingängen gibt es meist auch nicht-maskierbare IRQ-Eingänge (NMI, RESET). NMI kann z.B. verwendet werden, um Ausfall der Versorgungsspannung zu signalisieren (Nothalt und Datensicherung bei Prozesssteuerung).

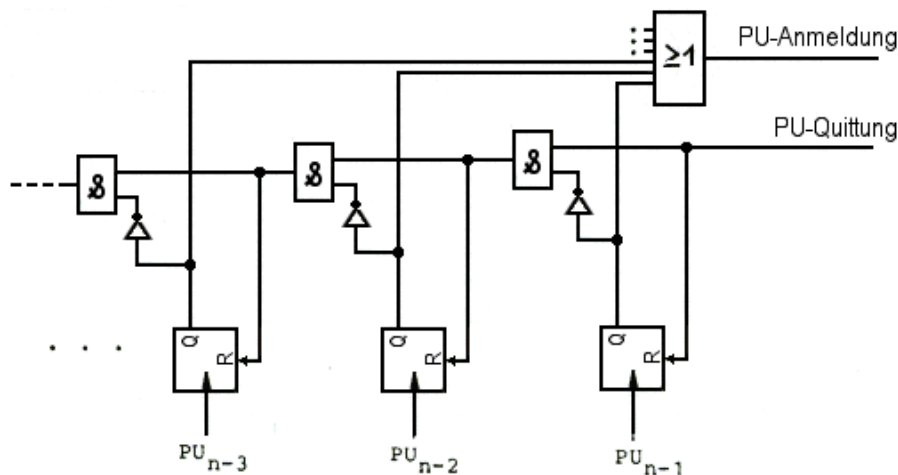


Wenn mehr PU-Quellen als IRQ-Eingänge vorhanden sind, müssen mehrere PU-Anforderungen mittels zusätzliche Hardware zusammengefasst werden, z. B. über ODER-Verknüpfung (bei 0-aktiven PU-Anforderungen über NAND). Daraus ergibt sich das Problem, dass die ISR (Interrupt Service Routine) zunächst die jeweilige Quelle identifizieren muss, da in der Regel auf unterschiedliche PU-Quellen vom Programm aus auch unterschiedlich reagiert werden muss. Die Identifizierung der Quelle ist auch notwendig, um das entsprechende IRQ-Flipflop beim Interrupt-Verursacher zurücksetzen zu können. Für die Identifizierung der PU-Quelle gibt es folgende Methoden:

- **Polling:**
Programmgesteuerte Abfrage der möglichen PU-Quellen, von welcher PU-Quelle der IRQ abgesandt worden ist. Es gibt nur eine ISR, die nach Identifizierung der PU-Quelle entsprechend verzweigt.
- **Vektorisierter Interrupt:**
Nach der Annahme der PU-Anforderung durch die Ablaufsteuerung sendet diese (über den Steuer-Bus) ein Quittungs-Signal (interrupt acknowledge). Die PU-Quelle, die den IRQ ausgelöst hat sendet daraufhin (über den Daten-Bus) einen Identifikations-Code an die CPU zurück. Der Code kann z.B. die ISR-Startadresse oder die Nummer des IRQ-Vektors (eindeutige Zuordnung zur IRQ-Startadresse) sein. Gleichzeitig wird das IRQ-FF zurückgesetzt.
- **Eindeutige Zuordnung:**
Jedem IRQ-Eingang ist eine feste ISR-Startadresse hardwaremäßig zugeordnet. Pro Eingang gibt es nur eine PU-Quelle.

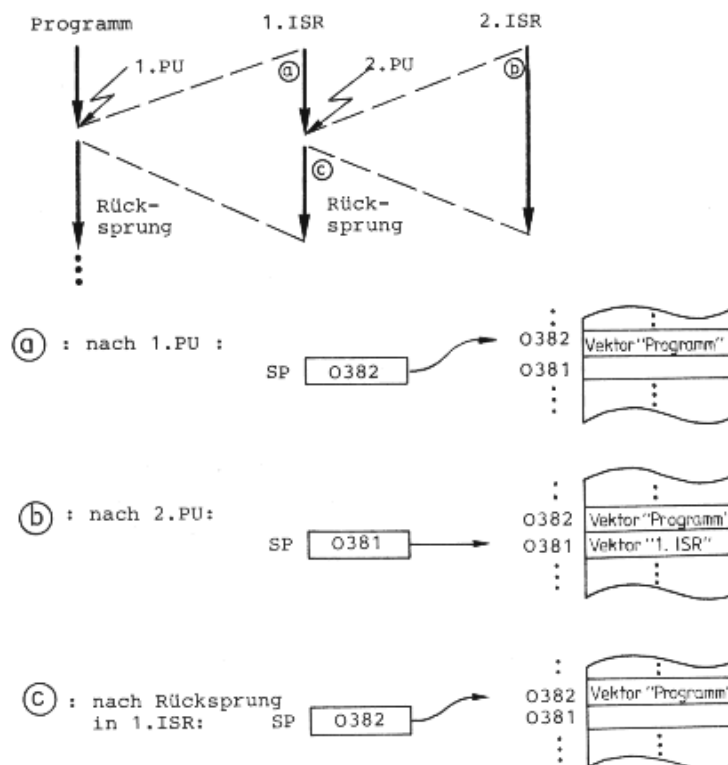
Sind mehrere PU-Quellen an einem IRQ-Eingang angeschlossen, ist es häufig zweckmäßig, zwischen wichtigen und weniger wichtigen PU-Quellen zu unterscheiden → prioritätsgesteuertes Interrupt-System. Die einfachste Form der Realisierung einer Prioritäts-Steuerung ist die Vorrang-Kette (daisy-chain = "Gänseblümchen-Kette"). Das von der Ablaufsteuerung kommende Quittungssignal gelangt über die UND-Gatter nur bis zu der PU-Quelle mit höchster Priorität. Nur von dieser wird dann der Identifikations-Code ausgesendet und das PU-Anforderungs-FF zurückgesetzt.

Die Prioritätsreihenfolge ist durch die Verschaltung der Quittungsleitungen festgelegt → "geographische Priorität".



Das Retten des aktuellen Programmzustands des unterbrochenen Programms geschieht häufig in einem Kellerspeicher (Stack), der i. a. Teil des Arbeitsspeichers ist, wobei der Top of Stack über ein spezielles Register (Stackpointer) des Leitwerks adressiert wird → beliebig geschachtelte Programmunterbrechungen sind möglich.

Während des Rettens des Programmzustands und des anschließenden Ladens des Zustandsvektors der ISR darf keine neue Unterbrechungsanforderung auftreten (Zustandsinformation unvollständig!) → Bei den meisten CPUs wird daher vom Leitwerk die PU-Anforderung gesperrt. In der ISR muss dann die Möglichkeit für PU-Anforderungen wieder per CPU-Befehl freigegeben werden.

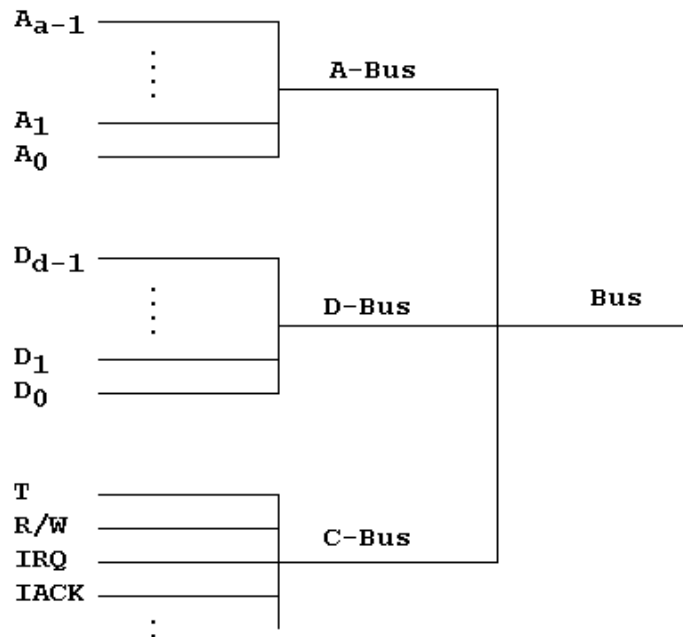


Nach Ende der ISR wird der ursprüngliche Zustandsvektor wieder vom Stack geholt und das Programm an der Unterbrechungsstelle fortgesetzt.

6.3 Bus-System und Bussteuerung

Der Bus ist die gemeinsame Verbindungsschiene (deutsche Bezeichnung für Bus: "Datensammelschiene") für die Kommunikation zwischen Prozessor, Speicherwerk und Ein-/Ausgabe-Werk. Er gliedert sich in die drei Signalgruppen "Adressen", "Daten" und "Steuersignale".

Die a-stelligen Adressen und die d-stelligen Daten werden jeweils parallel übertragen. Es gibt Mikroprozessoren, bei denen aus Gründen der Pin-Reduzierung (kleineres Gehäuse!) die Daten und Adressen im Zeitmultiplex über die gleichen Leitungen geschickt werden, dies ändert jedoch nichts an deren funktionaler Trennung!



Wesentliches Merkmal des Busprinzips ist die Möglichkeit, mit einer Signalleitung nicht nur einen Sender und einen Empfänger zu verbinden (Punkt-zu-Punkt-Verbindung), sondern mehrere Empfänger und sogar mehrere Sender anschalten zu können. Je nachdem, ob die am Bus angeschlossenen Einheiten jeweils nur als Sender bzw. nur als Empfänger wirken, oder ob einzelne oder alle Einheiten beides gleichzeitig sein können, ist der Informationsfluß auf dem Bus unidirektional oder bidirektional.

Der Adreß-Bus z.B. ist im einfachsten Fall unidirektional, wenn die Adressen nur vom Prozessor auf den Bus gegeben werden, während Speicher und E/A-Werk nur Adreßempfänger sind. Der Daten-Bus dagegen ist typisch bidirektional, denn Daten müssen von jeder Einheit zu jeder anderen transportiert werden können.

Während beim A- und D-Bus jeweils alle Leitungen parallel betrieben und deshalb auch identisch verschaltet werden, besteht der Steuerbus (Control-Bus oder C-Bus) meist aus verschiedenen voneinander unabhängigen Takt-, Status- und Steuerleitungen. Über sie wird das sogenannte Busprotokoll realisiert, d.h. die Regeln, nach denen die einzelnen Signalübertragungen abzuwickeln sind. Die Tatsache, daß an einem Bus mehrere Sender und Empfänger angeschaltet sind, bringt zwei Probleme mit sich:

- a. Es darf zu jedem Zeitpunkt nur ein Sender aktiv sein.

Würden zwei Treiberbausteine gleichzeitig auf die selbe Signalleitung senden, so würde zumindest die Information gestört (z. B. das logische ODER der beiden binären Informationen entstehen), im ungünstigsten Fall könnte es sogar zu einer Zerstörung der Treiberbausteine kommen! Bei Bussen kommen zwei Treiberschaltungs-Varianten zum Einsatz:

Open-Collector-Ausgänge oder Tri-State-Ausgänge (= Three-State).

Die "offenen" Kollektoren der Bustreiber-Transistoren bei Open-Collector-Ausgängen können nur den Low-Pegel aktiv durchschalten; sind sie gesperrt, so muß ein für alle gemeinsamer Pull-up-Widerstand (Zieh-Widerstand) für den High-Pegel sorgen. Damit ergeben sich unterschiedliche Flankenformen der Rechtecksignale auf dem Bus: schaltet ein Treiber-Transistor durch, so wird rasch (weil niederohmig) der Low-Pegel erreicht (steile High-Low-Flanke); beim Sperren des Transistors erzeugt der hochohmigere Pull-Up den High-Pegel. Die immer vorhandenen Kapazitäten der Busleitungen bilden mit dem Pull-up ein R-C-Glied mit relativ langer Zeitkonstante, die Flanke Low-High ist flacher.

Tri-State-Treiber besitzen den Vorteil, daß beide Pegel aktiv durchgeschaltet werden und damit beide Flanken steil sind. Für die Abschaltung der jeweils inaktiven Treiber sorgt ein Steuereingang, der den Ausgang hochohmig macht; so als wäre kein Treiber angeschaltet (dritter Zustand). Durch die Verschaltung geeigneter Steuersignale des C-Busses muß zu jedem Zeitpunkt dafür gesorgt werden, daß alle Busteilnehmer - mit Ausnahme des einzigen aktiven - im hochohmigen (dritten) Zustand sind. Sind auch nur kurzzeitig zwei aktiv, die unterschiedliche Pegel anlegen wollen, so fließt über die Transistoren der zwei Treiber ein sehr hoher Strom, der zu ihrer Zerstörung führen kann!

b. der Empfänger muß ausgewählt werden (wen betrifft eine gesendete Info?)

Sendet der Prozessor Daten auf den Bus, so muß vorher das Ziel (der Adressat) bestimmt sein, denn prinzipiell kann jeder Busteilnehmer "mithören" was über den Bus läuft. Die potentiellen Ziele einer Daten-Nachricht sind Speicherkarten und E/A-Karten; welche von ihnen gemeint ist (bei Speichern: welche Zelle), entscheidet die vorher übertragene Adresse. Alle Teilnehmer müssen die Adressinformation mit ihrer "eigenen" eingestellten Adresse vergleichen und nur dann ihre Empfängerbausteine für die Daten durchschalten, wenn Adressübereinstimmung besteht. Da die Bussignale nur für bestimmte (kurze) Zeit gültig sind, müssen Adress- bzw. Datenbus-Infos in den Busteilnehmern ggf. zum richtigen Zeitpunkt in Register übernommen und dort zwischengepuffert werden.

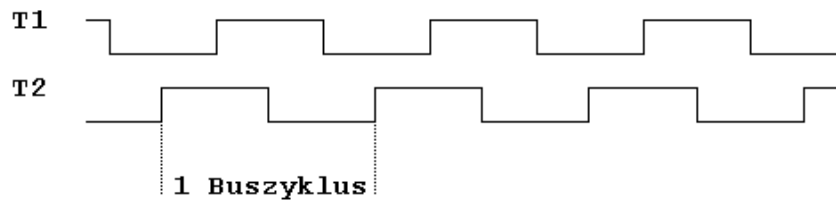
Bei einfachen Rechnersystemen steuert der Prozessor das Busprotokoll, d.h. er aktiviert die wichtigsten Steuersignale, während die übrigen Teilnehmer in der geforderten Weise mit entsprechenden Antwortsignalen reagieren müssen (Master-Slave-Prinzip). In Systemen mit mehreren Mastern (z.B. Mehrprozessorsysteme) müssen zusätzliche Steuersignale vorhanden sein, um den sogenannten Master-Transfer zu steuern, d.h. die Übergabe des Rechtes, die Steuerleitungen zu aktivieren. Mit steigenden Prozessorgeschwindigkeiten wachsen auch die Anforderungen an die Übertragungskapazität der Busse. Das führt zu steileren Flanken der Signale und damit zu höheren Frequenzanteilen (Fourier-Analyse!). Entsprechend wächst das Problem der Leitungsreflexionen an den Bus-Enden. Nur durch sorgfältigen Abschluß der Busleitungen mit dem Leitungs-Wellenwiderstand und durch Verwendung der vorgeschriebenen Treiber- und Empfängerbausteine können "saubere" Rechtecksignale mit steilen Flanken und ohne Überspringen sichergestellt werden!

Das Busprotokoll für den Datentransport kann auf zwei grundsätzlich verschiedene Arten realisiert werden:

a. Synchroner Bus:

Von einem "synchronen Bus" spricht man, wenn der Transfer von einem Taktsignal gesteuert wird, das für alle am Bus angeschlossenen Teilnehmer verfügbar ist. Häufig sendet der Prozessor sogar zwei phasenverschobene Taktsignale aus, um in einer Taktperiode insgesamt 4 Flanken zur Definition bestimmter Zeitpunkte zur Verfügung zu haben.

Datenverarbeitungssysteme

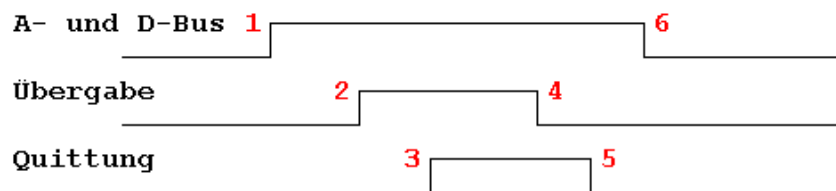


Durch die starre Zeiteinteilung kann der Ablauf eines Transfers ohne viel Aufwand gesteuert werden. Bei synchronen Bussen ist es sinnvoll, den Prozessortakt und die Zugriffszeiten der Speicher- und Peripheriebausteine aufeinander abzustimmen um Wartezyklen möglichst zu vermeiden.

b. Asynchroner Bus:

Die oben genannten Probleme sind beim asynchronen Bus unbekannt, denn hier ist kein festes Zeitraster für die Datenübertragungen auf dem Bus vorgeschrieben. Stattdessen quittiert der adressierte Empfänger die Übernahme der Daten vom Bus bzw. das Aussenden der Daten auf den Bus mit einem sogenannten Acknowledge-Signal (ACK) zum Prozessor. Erst damit ist dann der Buszyklus beendet, darf also ein neuer Buszyklus angestoßen werden.

Prinzip eines Schreibzyklus auf einem asynchronen Bus:

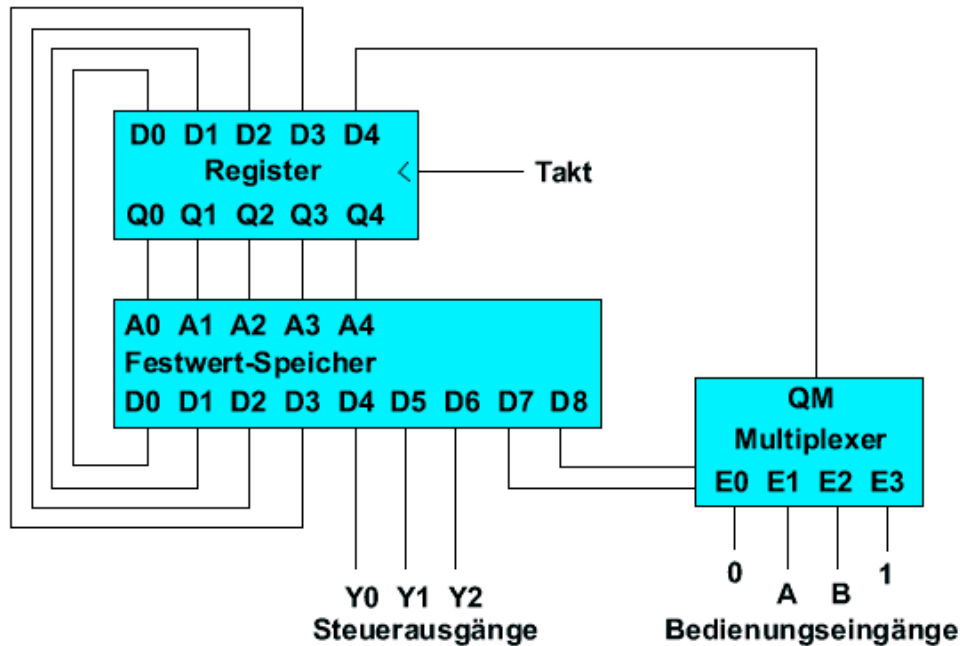


1. Adressen und Daten werden vom Prozessor auf den Bus gesendet
2. Prozessor signalisiert Gültigkeit von Adresse und Daten mit dem Übergabe-Signal
3. Speicher quittiert die Übergabe
4. Prozessor hat Quittung erhalten und deaktiviert Übergabe
5. Speicher erkennt, daß Quittung angekommen ist und deaktiviert sie
6. Prozessor erkennt an Rückflanke der Quittung das Ende des Zyklus und deaktiviert Adresse und Daten

Ein Vorteil dieses Prinzips ist die Möglichkeit, Speicherbausteine beliebiger Zugriffszeit verwenden zu können (z.B. Austausch gegen schnellere oder Verwendung verschiedener in einem System), wobei jeweils mit der optimal erreichbaren Buszykluszeit operiert wird. Notwendig ist jedoch ein gewisser Aufwand auf der Speicherkarte für das Signal-Handshaking. Auf der Prozessorseite wird außerdem eine Time-Out-Schaltung benötigt, die dann ansprechen muß, wenn innerhalb einer vorgegebenen Zeit keine Quittung empfangen wird (nicht existierende Adresse oder defekte Speicherkarte). Ein Ansprechen des Time-Out muß zu einem Interrupt des Prozessors führen ("Non existing memory trap").

Anhang: Einfache Beispiele zur Ablaufsteuerung

Es ist die folgende einfache Schaltung einer Ablaufsteuerung aus Festwertspeicher (mit 32 Worten), Register und Bedingungs Multiplexer gegeben. Der zeitliche Ablauf wird durch die Taktfrequenz festgelegt.



Die Ausgänge Q0 bis Q4 des Registers wählen die Adressen des Speichers aus und bestimmen so das Datenwort, das an den Speicherausgängen D0 bis D8 anliegt. Die Speicherausgänge D0 bis D3 bestimmen zusammen mit dem Multiplexerausgang QM, der auf den Registereingang D4 geleitet wird, den Folgebefehl. Die Speicherausgänge D7 und D8 wählen einer der Multiplexereingänge E0 bis E3 aus:

D7	D8	Folgeadresse					
0	0	D0	D1	D2	D3	0	unbedingter Sprung
1	1	D0	D1	D2	D3	1	unbedingter Sprung
0	1	D0	D1	D2	D3	A	bedingter Sprung A
1	0	D0	D1	D2	D3	B	bedingter Sprung B

Bei den Eingangskombinationen 01 oder 10 für D7,D8 hängt die Folgeadresse von Wert der Eingänge A und B ab. Als Beispiel soll Eingang A betrachtet werden:

- Ist der Wert von A=0, wird bei (D0,D1,D2,D3,0) fortgefahren.
- Ist der Wert von A=1, wird bei (D0,D1,D2,D3,1) fortgefahren.

Die Speicherausgänge D4 bis D6 können zur Steuerung von angeschlossenen Einheiten verwendet werden.

Die folgenden Beispiele sind absichtlich nicht auf die (komplexe) Ablaufsteuerung im Leitwerk einer CPU bezogen, sonder sollen durch Ihre einfache Aufgabenstellung die Arbeitsweise der einfachen Ablaufsteuerung demonstrieren.

1. Abfüllanlage

Auf einem Fließband werden Schachteln unter eine Abfüllstation transportiert. Die Schachtel wird jeweils mit 5 Teilen gefüllt und danach geschlossen. Während des Schließens muss 3 Taktperioden gewartet werden. Anschließend wird auf die nächste Schachtel gewartet. Die Ein- und Ausgänge werden folgendermaßen belegt:

Y0 0: Band stop, 1: Band läuft

1. Abfüllanlage

Datenverarbeitungssysteme

Y1 Taktimpulse für die Abfülleinrichtung
Y2 Taktimpuls für Schließmechanismus
A 1: Schachtel in Position

Der Ablauf sieht dann folgendermaßen aus:

1. Warten, bis Schachtel in Position
2. 5 Taktimpulse für die Abfülleinrichtung
3. 1 Taktimpuls für die Schließeinrichtung
4. 3 Taktperioden warten
5. Sprung nach Schritt 1

(Bei den Taktimpulsen wirkt jedes Mal die steigende Flanke)

Speicherbelegung:

A0	A1	A2	A3	A4		D0	D1	D2	D3	D4	D5	D6	D7	D8	
0	0	0	0	0		0	0	0	0	1	0	0	0	1	Schritt 1
0	0	0	0	1		0	0	1	0	0	1	0	0	0	Schritt 2 (T1)
0	0	1	0	0		0	0	1	1	0	0	0	0	0	Schritt 2 (T1)
0	0	1	1	0		0	1	0	0	0	1	0	0	0	Schritt 2 (T2)
0	1	0	0	0		0	1	0	1	0	0	0	0	0	Schritt 2 (T2)
0	1	0	1	0		0	1	1	0	0	1	0	0	0	Schritt 2 (T3)
0	1	1	0	0		0	1	1	1	0	0	0	0	0	Schritt 2 (T3)
0	1	1	1	0		1	0	0	0	0	1	0	0	0	Schritt 2 (T4)
1	0	0	0	0		1	0	0	1	0	0	0	0	0	Schritt 2 (T4)
1	0	0	1	0		1	0	1	0	0	1	0	0	0	Schritt 2 (T5)
1	0	1	0	0		1	0	1	1	0	0	0	0	0	Schritt 2 (T5)
1	0	1	1	0		1	1	0	0	0	0	1	0	0	Schritt 3
1	1	0	0	0		1	1	0	1	0	0	0	0	0	Schritt 3/4 (1)
1	1	0	1	0		1	1	1	0	0	0	0	0	0	Schritt 4 (2)
1	1	1	0	0		1	1	1	1	0	0	0	0	0	Schritt 4 (3)
1	1	1	1	0		0	0	0	0	0	0	0	0	0	Schritt 5

Schritt 1 stellt einen bedingten Sprung dar. Mit D7/D8 wird über den Multiplexer der Eingang A als Steuerung für Bit 4 des Registers ausgewählt. Solange der Eingang A Null ist, erfolgt ein Sprung nach 00000, also auf den Befehl selbst. Wechselt der Eingang auf 1, erfolgt ein Sprung nach 00001 (Schritt 2): Die beiden Sprungalternativen sind allgemein immer xxxx0 und xxxx1. Da die erste Adresse (00000) festliegt, ergibt sich die zweite zwangsläufig. Alle nun noch freien Speicherzellen werden sicherheitshalber wie Schritt 5 programmiert (Sprung nach Schritt 1).

2. Automatische Schranke

Eine im Ruhezustand geschlossene Schranke soll durch Tastendruck geöffnet werden und für die Dauer von 4 Taktimpulsen offen bleiben. Danach wird die Schranke wieder geschlossen. Gleichzeitig wird ein Lichtsignal (rot/grün) gesteuert. Der Schrankenmotor wird in den jeweiligen Endstellungen (offen/geschlossen) durch Kontakte am Schranken Antrieb stromlos geschaltet. Die Ein- und Ausgänge werden folgendermaßen belegt:

Y0 0: Schranke öffnen, 1: Schranke schließen
Y1 Lampe rot
Y2 Lampe grün
A 0: Schranke nicht offen, 1: Schranke ist offen
B Taster (1 = öffnen)

Der Ablauf sieht dann folgendermaßen aus:

1. (Ruhezustand) Schranke schließen, Lampe rot an, Lampe grün aus

2. Mit den Ausgangssignalen von Schritt 1 warten, bis Taste=1 ist
3. Schranke öffnen
4. Warten bis Schranke offen ist
5. Lampe rot aus, Lampe grün an, 4 Taktimpulse abwarten
6. Weiter bei Schritt 1

Speicherbelegung:

A0	A1	A2	A3	A4		D0	D1	D2	D3	D4	D5	D6	D7	D8	
0	0	0	0	0		0	0	0	1	1	1	0	0	0	Schritt 1
0	0	0	1	0		0	0	0	1	1	1	0	1	0	Schritt 2
0	0	0	1	1		0	0	1	0	0	1	0	0	1	Schritt 3
0	0	1	0	0		0	0	1	0	0	1	0	0	1	Schritt 4
0	0	1	0	1		0	0	1	1	0	0	1	0	0	Schritt 5 (1)
0	0	1	1	0		0	1	0	0	0	0	1	0	0	Schritt 5 (2)
0	1	0	0	0		0	1	0	1	0	0	1	0	0	Schritt 5 (3)
0	1	0	1	0		0	1	1	0	0	0	1	0	0	Schritt 5 (4)
0	1	1	0	0		0	0	0	0	0	0	1	0	0	Schritt 6

Alle nun noch freien Speicherzellen werden sicherheitshalber wie Schritt 6 programmiert (Sprung nach Schritt 1).

Erläuterungen: Die Schritte 2 und 3 enthalten bedingte Sprünge. Bei Schritt 2 wird abhängig von Eingang B (Taster) verzweigt:

B=0 → 0 0 0 1 0: Sprung auf die gleiche Stelle (das letzte Bit wird ja von Eingang B festgelegt).

B=1 → 0 0 0 1 1: Sprungziel, wenn Taste = 1. Da B = D4 das letzte Bit festlegt, ergibt sich das Sprungziel zwangsläufig.

Bei Schritt 3 wird nach dem gleichen Schema verfahren:

A=0 → 0 0 1 0 0: Sprung auf die gleiche Stelle (Letztes Bit = Eingang A)

A=1 → 0 0 1 0 1: Sprungziel für A = 1 → Schranke öffnen.

Schritt 6 ist ein unbedingte Sprung nach 0 0 0 0 0.



[Zum vorhergehenden Abschnitt](#)

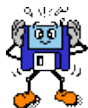


[Zum Inhaltsverzeichnis](#)



[Zum nächsten Abschnitt](#)

Die HTML-Fassung entstand unter Mitwirkung von Volker Arndt
Copyright © FH München, FB 04, Prof. Jürgen Plate



Einführung Datenverarbeitungssysteme

von Prof. Jürgen Plate

7. Speicherwerk (Arbeitsspeicher)

7.1 Speicherhierarchie

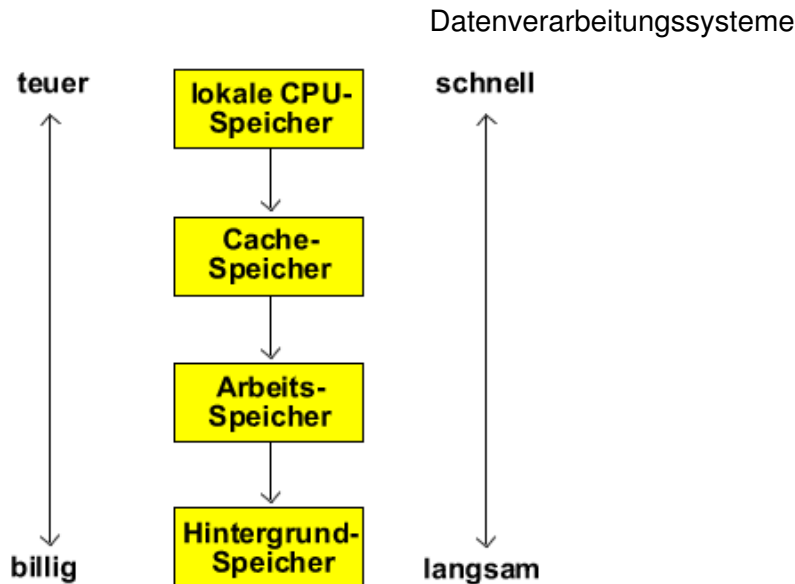
Zum Aufbewahren der Daten und Befehle dienen Speicher. Der zentrale Speicher ist das Speicherwerk (Hauptspeicher, Arbeitsspeicher). Daneben befinden sich auch in der CPU Speicher, die Sie bereits kennen gelernt haben:

- lokale Speicher: Register (teils mit speziellen Funktionen, z.B. Akkumulator, Befehlszähler, Indexregister, usw.; teils frei verwendbar, z.B. Universalregister)
- lokale Pufferspeicher für Befehle: (z.Z. noch nicht häufig) die alle an einer überlappenden Befehlsausführung beteiligten Befehle enthalten (Pipeline-Rechner) oder die eine gewisse Zahl von Befehlen enthalten (Queue, z.B. sehr schnelle Ausführung kurzer Schleifen).

Der Arbeitsspeicher soll einen möglichst schnellen Zugriff der CPU auf Daten und Befehle gestatten → relativ teuer. Er besitzt einen eingeschränkten Adressumfang (festgelegt durch die Adressbus-Breite der CPU). Da nicht alle benötigten Informationen ständig direkt für die CPU verfügbar sein müssen (und können) wird der Arbeitsspeicher durch periphere Speicher (Hintergrundspeicher, externe Speicher) ergänzt (z.B. Plattenspeicher, Magnetbandspeicher, etc.). Diese Speicher zeichnen sich aus durch:

- sehr hohe Speicherkapazität (gegenüber ASP)
- geringe Kosten pro Bit
- längere Zugriffszeit gegenüber ASP

Die **Zugriffsgeschwindigkeit** zum Arbeitsspeicher sollte möglichst der Arbeitsgeschwindigkeit der CPU angepasst sein, sonst muss die CPU beim Speicherzugriff unnötig lange warten. Sehr schnelle Speicher lassen sich nur mit sehr kleiner Kapazität realisieren und sind außerdem teuer. Andererseits wird für den ASP eine bestimmte Kapazität benötigt. Eine Abhilfe ist das Zwischenschalten eines schnellen Pufferspeichers (Cache Memory) geringer Kapazität zwischen ASP und CPU. Er enthält den "aktuellen" Teil der ASP-Informationen. Der Cache ist funktionell Teil des Arbeitsspeichers, es gibt jedoch inzwischen Mikroprozessoren, bei denen er im Chip integriert ist (z.B. Z8000, i486, 68020). Der Cache wird nicht mit dem gerade adressierten Wort des ASP, sondern mit einem ganzen Block von dieser Adresse ab geladen (typische Blockgröße: 4 - 16 Worte). Auf Grund der Lokalitätseigenschaft von Programmen (die Mehrheit der ASP-Zugriffe erfolgt in einer gewissen "Umgebung" der aktuellen Adresse) ist die Wahrscheinlichkeit, dass die (folgenden) benötigten Informationen schon im Cache stehen, sehr hoch, sodass sie entsprechend schnell in die CPU gelangen können (die "Trefferquote" ist i. a. größer als 90%). Die Gesamtheit aller Speicher bildet eine Hierarchie:



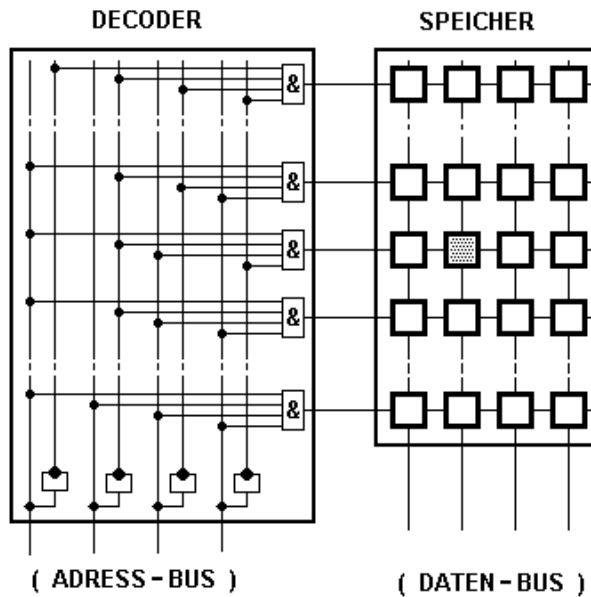
Die Speicherkapazität nimmt nach unten hin zu, Preis und Zugriffsgeschwindigkeit nehmen dagegen ab. Der Zugriff auf einen Speicher tieferer Hierarchiestufe erfolgt nur, wenn die gewünschte Information sich nicht in der höheren Stufe befindet. Der Informationstransport erfolgt blockweise. Eine weitere Verfeinerung der Hierarchiestufen ist möglich, z.B. Sekundärspeicher (Platte) und Tertiärspeicher (Magnetband).

7.2 Speicherarten

Für die verschiedenen Einsatzbereiche der Speicher werden unterschiedliche Speicherarten verwendet, die sich unterscheiden hinsichtlich:

- Speichermedium und physikalischem Arbeitsprinzip
- Organisationsform
- Zugriffsart
- Leistungsparameter
- Preis

Die Eigenschaften der in einem DVS eingesetzten Speicher bestimmen entscheidend dessen Leistungsfähigkeit. Die gespeicherte Information ist ausschließlich binär. Die Speicher bestehen aus Grundelementen, die nur zwei verschiedene diskrete Zustände annehmen können. Bei der Behandlung von Registern (aus Flipflops) haben Sie eine Art solcher Grundelemente bereits kennen gelernt. Weitere Formen werden in diesem Kapitel behandelt. Die Speicherung kann aber auch in jeweils begrenzten Bereichen kontinuierlicher Medien erfolgen (z.B. Magnetplatte, Magnetband). Mit Ausnahme der Register enthalten Speicher eine (mehr oder weniger) hohe Anzahl von Informationen → Problem der Auswahl der gewünschten Einzelinformation.



Speicherzellen sind meist als Matrix angeordnet

Begriffsdefinitionen:

- Lesen: Entnahme der Information aus dem Speicher
- Schreiben: Eingabe der Information in den Speicher
- Speicherzugriff: Lesen oder Schreiben

Die relevanten Informationseinheiten müssen nicht notwendigerweise einzelne Bits sein, sondern auch Bitfolgen (z.B. Worte oder Bytes). Den Speicherbereich für ein Bit nennt man Speicherzelle. Je nachdem, ob die Informationseinheiten, die mit einem einzigen Speicherzugriff erreicht werden können, einzelne Worte oder mehrere Worte sind, unterscheidet man:

- wortorganisierte Speicher

wahlfreier Zugriff	orts-adressiert	Schreib-/Lese-Speicher - statisch - dynamisch	RAM	Halbleiter (Magnetkern)
		Festwert-Speicher	ROM	Halbleiter
	inhalts-adressiert *	Assoziativ-Speicher	CAM	Halbleiter (Magnetkern)
quasi-wahlfreier Zugriff	zeit-adressiert *	Umlaufspeicher		Halbleiter (CCD) Magnetblasen
impliziter Zugriff	zeiger-adressiert	Kellerspeicher	LIFO	Halbleiter
		Silospeicher	FIFO	Halbleiter
direkter Zugriff	Registerspeicher			Halbleiter

- blockorganisierte Speicher

Datenverarbeitungssysteme

quasi-wahlfreier Zugriff	mechanisch bewegter Datenträger	(Magnet-Trommel)	
		Magnet-Platte	Festkopfplatte Winchesterplatte Wechselplatte Floppy-Disk
		optische Platte	CD-ROM WORM MO-Platte
	magnetisch bewegtes Datenmuster	Magnetblasen "Bubble-Memory"	Permalloy spulenlos Kreuzschraffur
sequenzieller Zugriff	mechanisch bewegter Datenträger	Magnet-Band	Magnetbandgerät Kassettengerät = Streamer
		Magnet-Karte	

Die innerhalb der Zentraleinheit eingesetzten Speicher sind in der Regel wortorganisiert, während die externen Speicher i. a. blockorganisiert sind. Ein weiteres Unterscheidungskriterium ist die Zugriffsart, d.h. Die Art der Auswahl der gewünschten Information. Bei wortorganisierten Speichern unterscheidet man:

- Speicher mit wahlfreiem Zugriff (Random Access Memory)
Auf jeden Speicherplatz kann über einen festverdrahteten Adressierungsmechanismus zugegriffen werden (Arbeitsspeicher). Die Zugriffszeit ist immer gleich lang und definiert.
- Speicher mit quasi-wahlfreiem Zugriff
Auch hier kann auf jeden Speicherplatz in jeder beliebigen Reihenfolge zugegriffen werden. Der Adressierungsmechanismus ist aber nicht oder nur teilweise festverdrahtet; die gespeicherte Information ist bezüglich der Ein-/Ausgabeortes nicht ortsfest → sie läuft um → Umlaufspeicher. Die Zeit stellt ein zusätzliches Auswahlkriterium dar. Die Zeitdauer zum Auffinden der Information ist variabel (CCD-Speicher, Magnetblasen-Speicher). Es lässt sich eine mittlere Zugriffszeit angeben.
- Speicher mit implizitem Zugriff
Es kann jeweils nur zu einem implizit vorgegebenen Speicherplatz zugegriffen werden. Nur die dort stehende Information ist zugänglich (Stack, FIFO).
- Speicher mit direktem Zugriff
sind Ein-Wort-Speicher in der CPU (Register mit fester Funktion). Die Information steht direkt zur Verfügung - eine Auswahl ist weder möglich noch nötig (z. B. Akkumulator).

Bei blockorganisierten Speichern unterscheidet man:

- Speicher mit quasi-wahlfreiem Zugriff
Auch hier kann auf jeden Speicherplatz in jeder beliebigen Reihenfolge zugegriffen werden. Der Adressierungsmechanismus ist aber nicht festverdrahtet. Im Gegensatz zum wortorganisierten Speicher sind i. a. die zum Auffinden der gewünschten Daten verwendeten Informationen (= Blockanfang) komplexer (gespeicherte Adressierungsinformation). Zugriffszeit ist variabel. (Magnetplatte, Magnettrommel).
- Speicher mit sequenziellem Zugriff
Speicherung und Auffinden der Information geschieht rein sequentiell. Der Zugriff zu beliebigen Speicherplätzen ist nicht möglich. Das Auffinden von Daten erfolgt nicht über eine Adressinformation sondern über die Reihenfolge der Anordnung der Daten. (Magnetband, Magnetbandkassette)
- Kellerspeicher (LIFO = Last In First Out, Stack, Stapelspeicher)
speichern die Daten in der Reihenfolge des Schreibens, das Auslesen erfolgt in umgekehrter Reihenfolge. Die implizit vorgegebenen Speicheradresse, zu der ein Zugriff möglich ist, ist:
 - ♦ für das Lesen: die Zelle, die das zuletzt eingelesene Wort enthält
 - ♦ für das Schreiben: die nächste freie Zelle

Realisierung z. B. mit Vorwärts-Rückwärts-Zähler als Schreibregister. Häufig erfolgt die Realisierung auch im Arbeitsspeicher: Die Adressierung erfolgt mit einem Vorwärts-/Rückwärts-Zähler, in dem immer die Adresse der zuletzt beschriebenen Speicherzelle steht (Top of Stack, TOS).

- ♦ Vor jedem Schreiben wird der Zähler um 1 erniedrigt, also auf die Adresse der nächsten freien Zelle. Schreiben: PUSH
- ♦ Nach jedem Lesen wird der Zähler um 1 erhöht. Der Stack beginnt also mit der höchsten Adresse und wächst "nach unten". Lesen: PULL, POP

Anwendungen: Bei CPU zur Speicherung der Rücksprungadresse (und Parametern) beim Unterprogramm-Aufruf, bei Unterbrechungsbehandlung Speicherung des Prozessorstatus.

- Silospeicher (Queue, FIFO = First In First Out)

speichern die Daten in der Reihenfolge des Schreibens, das Auslesen erfolgt in der gleichen Reihenfolge. Die implizit vorgegebenen Speicherzelle, zu der ein Zugriff möglich ist, ist:

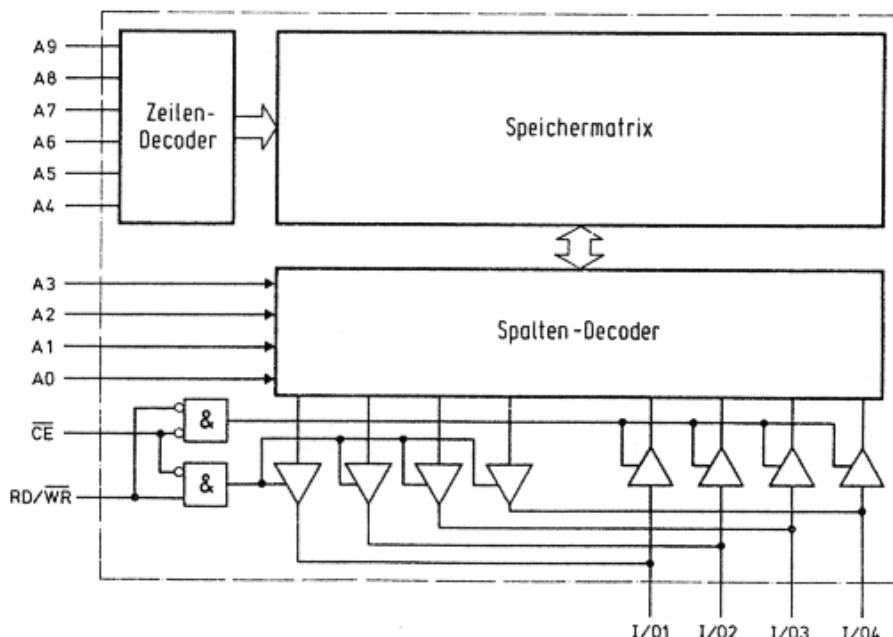
- ♦ für das Lesen: die Zelle, die das erste noch nicht ausgelesene Wort enthält
- ♦ für das Schreiben: die nächste freie Zelle nach der zuletzt beschriebenen

Realisierung z.B. durch ortsadressierten Speicher und zwei Ringzähler für Einlese- und Ausleseadresse. Je nachdem, ob geschrieben oder gelesen werden soll, wird der entsprechende Zähler (= Adresszeiger) auf den Adressdecoder geschaltet. Mit jedem Zugriff wird einer der beiden Zähler weitergeschaltet. Eine Steuerlogik verhindert, dass mehr ausgegeben wird, als eingeschrieben wurde oder dass beim Einschreiben die Speicherkapazität überschritten wird, d.h. dass die Zeiger sich gegenseitig "überholen":

7.3 Arbeitsspeichermedien

Neben der eigentlichen Speichermatrix enthält ein Speicherbaustein noch (*siehe Umdruck*):

- ♦ Zeilen-Adressdecoder (Wort-Decoder) und -Register
- ♦ Spalten-Adressdecoder und -Register
- ♦ Dateneingang und Datenausgang
- ♦ Lese-/Schreib-Steuersignale
- ♦ Chip-Select-Signal (Chip Enable)
- ♦ Leseverstärker
- ♦ Spaltenschalter



Das Adressregister dient als Zwischenspeicher für die gewünschte Adresse. Über den Adressdecoder wird eine einzelne Speicherzelle (Bit oder Speicherwort) ausgewählt. Lese- und Schreibregister bilden die Schnittstelle zur CPU oder zum Datenbus. Da die Wortbreite gegenüber der Anzahl der Speicherworte sehr klein ist, würde sich bei der vollständigen Decodierung der Adresse eine ungünstige Topologie des Speichers ergeben. Deshalb teilt man die Adressbits und ordnet die Speicherzellen (bzw. Speicherworte) in einer annähernd quadratischen Matrix an. Der Decoder wird so auch in einen Zeilen- und Spaltendecoder aufgeteilt.

Vor allem sehr hoch integrierte Halbleiterspeicher sind bitorganisiert, d. h. die Wortbreite ist 1 Bit. Es gibt aber auch Speicherbausteine, die selbst schon wortorganisiert sind (meist 4 oder 8 Bit). Heute werden im Arbeitsspeicher - bis auf wenige Ausnahmen - Halbleiterspeicher eingesetzt.

Vorteile der Halbleiter-Speicher:

- ◆ größere Speicherdichte
- ◆ Ein- und Ausgänge direkt kompatibel mit den übrigen Bauteilen eines DVS (TTL-Pegel); keine Interface-Schaltungen notwendig
- ◆ Geringer Platzbedarf, einfache Anwendbarkeit als "Bauteil"
- ◆ Kostengünstiger (Preis/Bit) als Kernspeicher
- ◆ Schneller (bestimmte Realisierungen von Halbleiterspeichern weisen um den Faktor 10 - 50 kürzere Zugriffszeiten auf)
- ◆ Geringer Energiebedarf

Nachteil der Halbleiterspeicher:

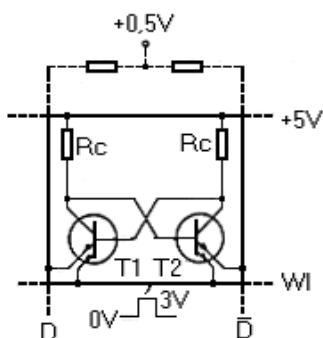
- ◆ flüchtige Speicher (volatile). Beim Abschalten der Energiezufuhr geht die Info im Speicher verloren.
- ◆ Eine Ausnahme bilden hier batteriegepufferte RAMS, PROMS oder EEPROMS (*siehe später*).

Halbleiter-Schreib-Lese-Speicher

Nach der Art des Speicherverfahrens unterscheidet man statische und dynamische RAMs (RAM = Random Access Memory):

Statische Halbleiterspeicher (SRAM)

Das Speicherelement ist ein Flip-Flop. Solange die Energieversorgung anliegt, bleibt die gespeicherte Info erhalten (→ statisch). Realisierung sowohl in bipolarer Technologie (TTL, ECL) als auch in MOS-(FET-) Technologie.



Transistor-Zelle (bipolar):

Das Speicher-FF besteht aus 2 MultiEmitter-Transistoren → schneller, höherer Leistungsbedarf.

Datenverarbeitungssysteme

Ruhezustand (Zelle nicht angewählt):

Wortleitung auf 0-Potential; beide Datenleitungen auf 1-Potential. Je nach Info ist T1 oder T2 leitend, sein Emitterstrom fließt über WL.

Lesen:

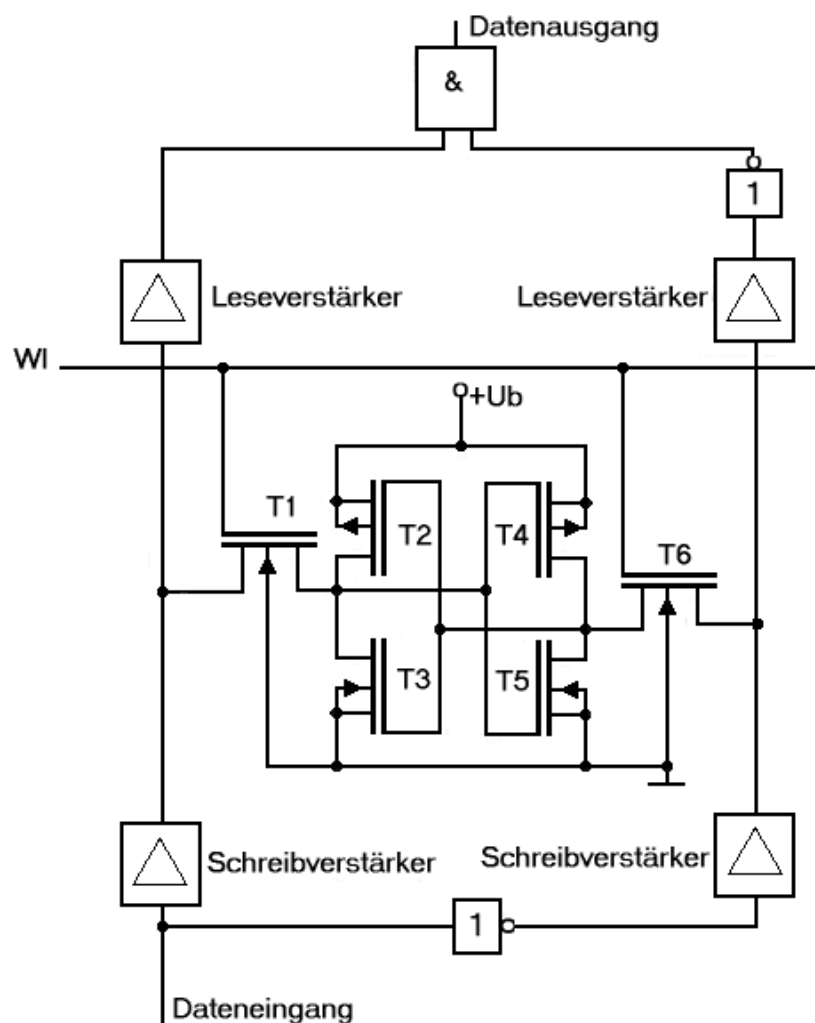
Wortleitung auf 1-Potential; beide Datenleitungen auf 0-Potential; Emitterstrom beider Transistoren fließt über zugeordnete DL. Der in beiden DL fließende Strom wird über einen Differenzverstärker ausgewertet.

Schreiben:

Wortleitung auf 1-Potential. Setzen des FF durch eine Datenleitung auf "0" und andere Datenleitung auf "1".

"1": $D = 1 \rightarrow T2$ leitend, $T1$ gesperrt

"0": $D = 0 \rightarrow T1$ leitend, $T2$ gesperrt



CMOS-2
Speiche
besteht
aus
2
CMOS-1
Extrem
geringe
Rest-Ver

Ruhezustand (Zelle nicht angewählt):

Wortleitung auf 0-Potential; T1 & T6 sperren. Datenleitungen sind abgekoppelt.

Lesen:

Wortleitung auf 1-Potential --> T1 & T6 leitend. Die beiden Datenleitungen der gew. Speicherstelle werden auf einen Differenzverstärker geschaltet. Alle anderen Datenleitungen

sind inaktiv (Spaltenwahl). Potentialdifferenz wird ausgewertet.

Schreiben:

Wortleitung auf 1-Potential. Setzen des FF durch eine Datenleitung auf "0" und andere Datenleitung auf "1" (bei CMOS abgeschaltet).

"0": $D = 0 \rightarrow T3, T4$ leitend, $T2, T5$ gesperrt

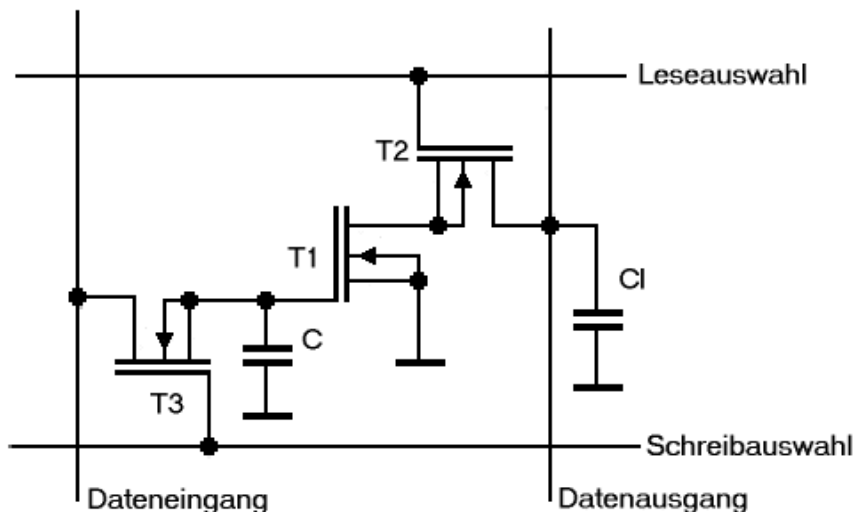
"1": $D = 1 \rightarrow T2, T5$ leitend, $T3, T4$ gesperrt

Die Verbindung der einzelnen Speicherelemente an Eingangs- und Ausgangsverstärker sowie die Zeilen- und Spaltenauswahl sind - je nach Technologie - Transistoren oder FETs. Im Beispiel oben haben alle Spalten den gleichen Leseverstärker. Es gibt auch Realisierungen mit je einem Leseverstärker pro Spalte - deren Ausgänge liegen dann am gemeinsamen "Data Sense Bus". Mit dem Spaltenauswahl-Signal wird dann auch der entsprechende Verstärker auf den Ausgangspuffer geschaltet. Über das WE-Signal wird zwischen Schreiben und Lesen umgeschaltet. Das CS (Chip Select)-Signal erlaubt die Aktivierung des Bausteins. Ein nicht aktivierter Baustein verhält sich in der Schaltung passiv $\rightarrow$ Zusammenschalten mehrerer Bausteine zur Erweiterung der Kapazität möglich.

Dynamische Halbleiterspeicher

Das Speicherelement ist hier eine Kapazität, die Information wird also als Ladung gespeichert. Wegen der unvermeidlichen Leckströme gibt es ständige Ladungsverluste, was ein periodisches Auffrischen der Info erforderlich macht ("refresh", typische Periode 2ms) $\rightarrow$ dynamische Speicher). Eigenschaften:

- ◆ hohe Integrationsdichte (einfacherer Aufbau der Speicherzelle)
- ◆ billiger als statisches RAM gleicher Kapazität
- ◆ geringerer Leistungsbedarf
- ◆ komplizierter in der Anwendung (wegen Refresh)



Realisierung nur in MOS-Technologie. Wegen des geringen Platzbedarfs verwenden Speicher höherer Kapazität (ab ca. 16 KBit) ausschließlich die 1-Transistor-Zelle, die hier genauer betrachtet werden soll. Die reale

Zelle wird mit einem Kondensator und einem Transistor aufgebaut. Im Ruhezustand liegt die Wortleitung auf 0-Pegel, T1 und T3 sperren und C ist von der Datenleitung abgekoppelt.

Schreiben:

Die Schreibauswahlleitung liegt auf 1-Potential, T3 leitet und C lädt sich auf das Potential der Datenleitung (0 oder 1) auf.

Lesen:

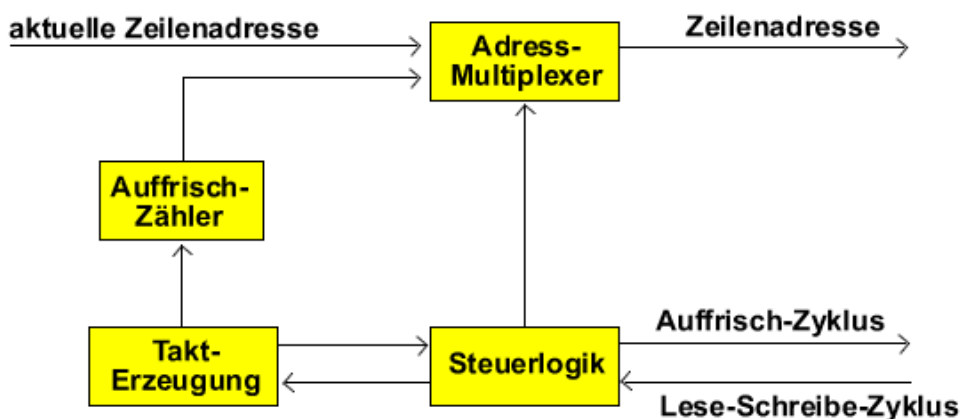
Die Datenleitung liegt auf 1-Potential, wodurch die Leitungskapazität C1 auf 1-Potential

geladen wird ("Precharge"). Die Leseauswahlleitung liegt ebenfalls auf 1-Potential, T2 leitet und es erfolgt ein Ladungsaustausch zwischen C und Cl. War C auf 0-Potential, wird Cl teilweise entladen und es erfolgt eine Potentialänderung auf der Datenleitung. War C auf 1-Potential, wird Cl nicht entladen und es gibt keine Potentialänderung auf der Datenleitung.

Die Ladung in C wird durch das Lesen zerstört. Daher ist nach jedem Lesezugriff ein erneutes Einschreiben der Info notwendig.

Besonderheiten der DRAM-Bausteine:

Im allgemeinen ein Leseverstärker pro Spalte. Es wird immer eine ganze Zeile gelesen (auch beim Schreibzugriff) und in einem Register zwischengespeichert. Hier erfolgt dann die Bit-Auswahl gemäß der Spaltenadresse. Beim Schreiben wird das entsprechende Bit geändert, beim Lesen ausgegeben. Anschließend wird der Registerinhalt in die Zeile zurückgespeichert. Um Bauteileanschlüsse zu sparen, wird ab 16 KBit-Baustein die Adresse im Multiplex zugeführt → gleiche Anschlüsse für Spalten- und Zeilenadresse. Es wird zuerst die Zeilenadresse und danach die Spaltenadresse zugeführt. Statt eines CS- (Chip Select) Anschlusses gibt es nun RAS (Row Address Strobe) und CAS (Column Address Strobe). Zusätzlich ist eine Auffrisch-Logik erforderlich, da der Speicherkondensator durch Leckströme an Ladung verliert. Das Auffrischen kann so organisiert werden, dass es den normalen Betrieb nicht behindert. Teilweise ist eine externe Auffrischlogik notwendig, bei manchen Speichern ist sie bereits in den Baustein integriert → quasistatische RAMs. In der Regel ist für jede Spalte ein Leseverstärker vorhanden und es kann gleichzeitig eine ganze Zeile zwischengespeichert werden. Damit ist zeilenweises Auffrischen möglich (nur RAS-Signal).



Die Steuerlogik bewirkt, dass ein Auffrischzyklus nicht während eines normalen Speicherzugriffs anläuft und der Speicherbaustein während eines Auffrischzyklus für Schreiben und Lesen gesperrt ist.

7.3.3 Halbleiter-Festwertspeicher

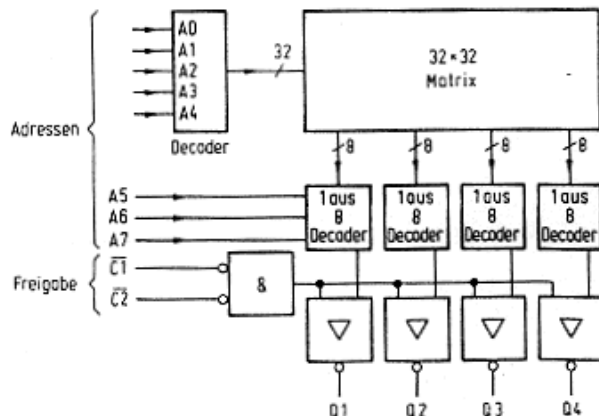
Festwertspeicher (ROM, Read Only Memory, Nur-Lese-Speicher) sind ebenfalls Speicher mit wahlfreiem Zugriff. Im normalen Betrieb können sie nur gelesen werden, die gespeicherte Info ist nicht flüchtig - geht also beim Abschalten der Versorgungsspannung nicht verloren. Sie besitzen einen einfacheren Aufbau als Schreib-Lese-Speicher und haben daher eine hohe Integrationsdichte. Die Bausteine sind i.a. wortorganisiert (meist 8-Bit-Wort = Byte). Man unterscheidet (grob) in:

- ◆ **ROM:** Informationen werden beim Herstellungsprozess eingegeben → maskenprogrammiertes ROM (ROM = Read Only Memory)
- ◆ **PROM:** Einmal programmierbares ROM. Die Info kann vom Anwender mittels eines

Programmiergeräts eingeschrieben werden. (PROM = Programmable ROM)

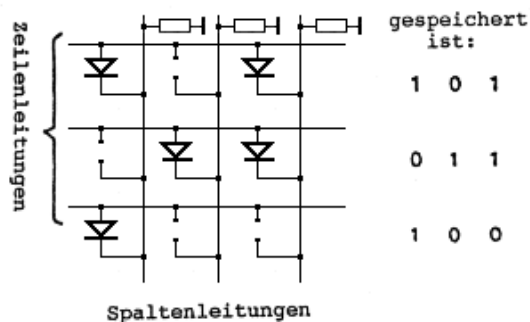
- ◆ **EPROM:** Vom Anwender programmierbares PROM, dessen Inhalt auch wieder gelöscht werden kann (mit Ultraviolett-Licht außerhalb des DVS). (EPROM = Erasable PROM)
- ◆ **EEPROM:** Vom Anwender programmierbares PROM, dessen Inhalt elektrisch wieder gelöscht werden kann (innerhalb des DVS). (EEPROM = Electrically Erasable PROM)

Speicherelement ist jeder Kreuzungspunkt zwischen Zeilen- und Spaltenleitung der Speichermatrix. Der jeweils gespeicherte Binärwert wird durch das Vorhandensein oder Fehlen einer leitenden Verbindung zwischen Zeilen- und Spaltenleitung bestimmt. Damit die einzelnen Zeilenleitungen entkoppelt sind, muss die Verbindung unidirektional sein.



Maskenprogrammierbares ROM

Integrierte Brücken bei Diodenmatrix; Spaltenleitungen über Widerstand auf Masse.

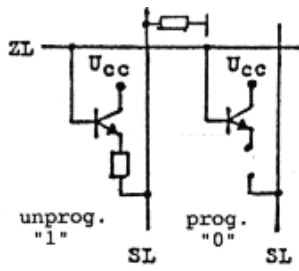


MOS:

Dicke der Gate-Oxidschicht; bei normaler Dicke kann der FET leiten ("0"), andernfalls reicht das 1-Potential auf der WL nicht aus, den FET durchzuschalten → "1".

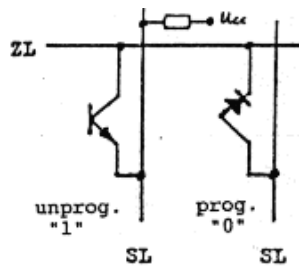
bipolares PROM (anwenderprogrammierbar)

Im unprogrammierten Zustand sind alle Kreuzungspunkte entweder leitend oder unterbrochen (Abhängig vom Herstellungsverfahren). Durch die Programmierung (= Einschreiben der Daten) wird die Verbindung (je nach Herstellungsverfahren) unterbrochen oder hergestellt → der binäre Wert, der dem durch die Herstellung vorhandenen nicht entspricht, wird programmiert.



Programmierung mit Ausbrennwiderständen

Der Ausbrennwiderstand ("fusible link") liegt an der Substrat-Oberfläche; er hat die Funktion einer Schmelzsicherung (z. B. Titan-Wolfram, Chrom-Nickel, Polysilizium). Das Durchbrennen geschieht mittels eines hohen Programmierstroms, z. B. durch Adressieren der gewünschten Zelle und Anlegen einer Spannung an die Ausgänge des Bausteins. Programmierzeit ca. 1 ms/Bit.



Programmierung durch kurzgeschlossene Sperrschicht

Das Koppellement ist ein npn-Transistor mit nicht angeschlossener Basis ("junction fuse"). Durch Anlegen einer hohen Spannung erfolgt ein Durchbruch der BE-Diode - es bleibt eine Diodenstrecke übrig (Basis an Spaltenleitung). Die Adressierung erfolgt im Baustein durch 0-Potential an den Zeilenleitungen.

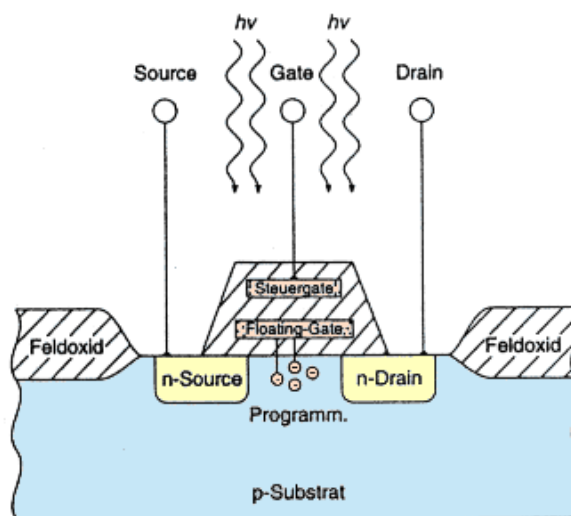
Programmierzyklus (Beispiel): Chip Select auf 1-Pegel, dann 20 V an Datenausgänge anlegen und Chip Select auf 15 V (5 ms). Programmierzeit: ca. 0,2 ms Bit

NMOS-PROM

EPROM ohne Löschmöglichkeit → EPROM, das nur ein einziges Mal programmiert werden kann → production EPROM, one time programmable EPROM.

EPROM

Das Koppelungselement ist eine Ladung, die auf einem isolierten Gate eines FET gespeichert wird ("floating gate"). Das Speicherelement besteht i.a. aus einem n-Kanal-MOSFET mit einem 2-Lagen-Silizium-Gate (zwischen Kanal und eigentlichem Gate befindet sich noch ein isolierter Gate (FAMOS - FET: floating gate avalanche injection FET). Source und Substrat auf Masse.



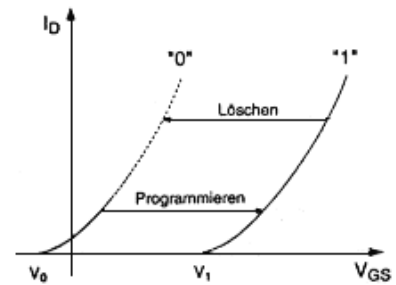
unprogrammiert:

keine Ladung auf isoliertem Gate, FET leitet, 0-Potential auf SL wird durch Ausgangsverstärker invertiert ausgegeben.

programmiert:

negative Ladung auf isoliertem Gate, durch Ladungsverschiebung leitet FET nicht, 1-Potential auf SL, 0-Potential am Ausgang.

Bei der Programmierung wird über die Zeilen- und Spaltenleitung eine hohe Spannung an Gate und Drain angelegt (Auswahl der Zelle). Source und Substrat liegen auf Massepegel. Infolge des starken elektrischen Feldes erfolgt eine Injektion von Elektronen auf das isolierte Gate (Lawinendurchbruch, avalanche effect). Die Ladung bleibt auch nach dem Abschalten der Versorgungsspannung erhalten. Der Ladungsverlust beträgt schätzungsweise 30% in 10 Jahren. Die negative Ladung auf dem isolierten Gate verschiebt die Schwellenspannung des FET und verhindert ein Durchschalten.

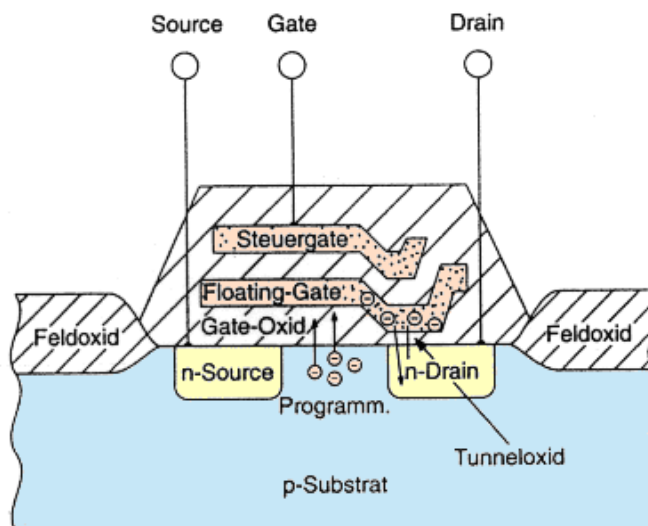


Zur Programmierung wird eine höhere Spannung (12 - 25 V) an den Baustein (V_{pp}) angelegt. Die Auswahl des zu programmierenden Wortes erfolgt genauso, wie beim Lesezugriff. Die zu programmierenden Daten werden an die Ausgangspins angelegt (TTL-Pegel). Mit einem 0-Impuls von 50 ms Dauer am Programmierpin wird der Wert programmiert. Moderne EPROMS verwenden einen "intelligenten" Algorithmus: es werden solange Impulse von 1 ms Dauer gegeben, bis die Daten beim Prüfllesen stimmen; danach erfolgen noch 5 - 10 weitere "Sicherheits-Impulse".

Das Löschen der Gate-Ladung erfolgt durch Bestrahlung des Chips mit UV-Licht (15 - 20 min) durch ein Fenster aus Quarzglas, wodurch die Isolierschicht ionisiert wird und so die "gespeicherten" Elektronen entweichen.

EEPROM

Das Speicherprinzip ist ähnlich wie beim EPROM (auch "floating gate"), jedoch das Programmierverfahren ist unterschiedlich. Statt des Lawinen-Durchbruchs wird der Tunneleffekt verwendet. "Poly-Poly-Silizium" - Die Elektronen "tunneln" durch eine sehr dünne Isolationsschicht aus Siliziumoxid (Fowler-Nordheim-Tunnelung). Es ist eine sehr hohe Feldstärke erforderlich, daher muss die Schicht sehr dünn sein (ca. 150 Å). Durch eine Modifizierung (gewellte Oberfläche) kann auch mit dickeren Schichten gearbeitet werden (ca. 800 Å).



unprogrammiert:

Der FET leitet (bei Auswahl über ZL); 1-Potential auf der Spaltenleitung.

programmiert:

Negative Ladung auf isoliertem Gate, durch Ladungsverschiebung leitet FET nicht, 0-Potential auf SL.

Zur Programmierung Gate (ZL) und Source (SL) auf 1-Potential legen. Drain liegt auf 0-Potential. Die Elektronen "tunneln" auf das Floating Gate. Programmierzeit: 5 - 10 ms/Byte. Zum Löschen Gate (ZL) auf 1-Potential und Source (SL) auf 0-Potential legen. Drain liegt dann auf 0-Potential. Die Elektronen "tunneln" vom Floating Gate nach Drain und der FET leitet wieder.

EEPROMs können in eingebautem Zustand (in der Schaltung) programmiert werden, da keine zusätzliche Programmierspannung erforderlich ist. Die benötigte hohe Programmierspannung wird während des Programmierens auf dem Chip selbst erzeugt. Auch Adress- und Datenregister sind auf dem Chip integriert. Nach dem Auslösen des Programmiervorgangs (=Schreibzyklus) kann die CPU sich anderen Aufgaben widmen (manche EEPROMs haben einen speziellen READY/BUSY-Pin zur Anzeige des internen Programmiervorgangs; andere liefern beim Lesen invertierte Daten solange die Programmierung läuft).

EEPROMs verhalten sich somit wie RAMs mit einem sehr langen Schreibzyklus (5 - 10 ms), die aber die gespeicherte Info nach dem Abschalten der Versorgung behalten (Lesezyklus 150 - 450 ns). Die Zahl der Schreibzyklen ist begrenzt (10.000 - 100.000 mal). Manche EEPROMs bieten die Möglichkeit, den gesamten Chip auf einmal zu löschen ("chip erase") → Flash-Speicher (Intel).

7.4 Organisation des Speicherwerks

Das Speicherwerk besteht aus:

- ♦ den eigentlichen Speichereinheiten
- ♦ Ankoppelschaltung an CPU und E/A-Werk

Im allgemeinen enthält das Speicherwerk mehrere (u. U. unterschiedliche) Speicherbausteine. Die Ankoppelschaltung (meist als Ankoppelung an den Systembus) besteht aus:

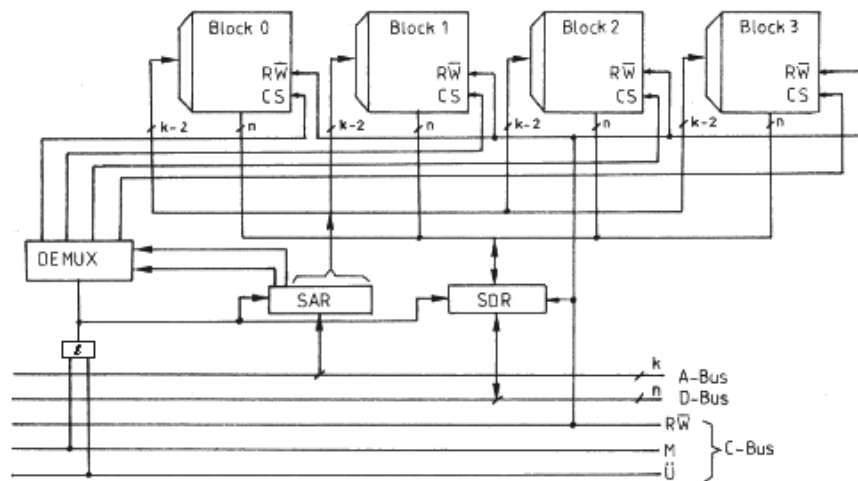
- ♦ Speicherdatenregister (SDR) - Zusammenfassung von Lese- und Schreibregister (nicht unbedingt notwendig)
- ♦ Speicheradressregister (SAR) (nicht unbedingt notwendig)
- ♦ Adressdecoder - Ableitung der Ansteuersignale für die einzelnen Bausteine (Chip-Select) aus Teilen der Adresse (Anzahl der Adressleitungen der Bausteine ist in der Regel kleiner als die Adressbreite der CPU)
- ♦ Anschluss von Steuerleitungen, z.B. Festlegung Lesen/Schreiben, Taktleitungen (Zeitpunkt des Zugriffs), Speicheraktivierung.

7.4.1 Blockweise Organisation des Arbeitsspeichers

Der Gesamtspeicher zerfällt in der Regel in mehrere Speichereinheiten (Blöcke, Module).

Eine blockweise Organisation des Arbeitsspeichers liegt vor, wenn die einzelnen Zellen eines Speicherblocks durch aufeinanderfolgende Adressen angesprochen werden → physikalischer Speicherblock = logischer Adressenblock.

Datenverarbeitungssysteme

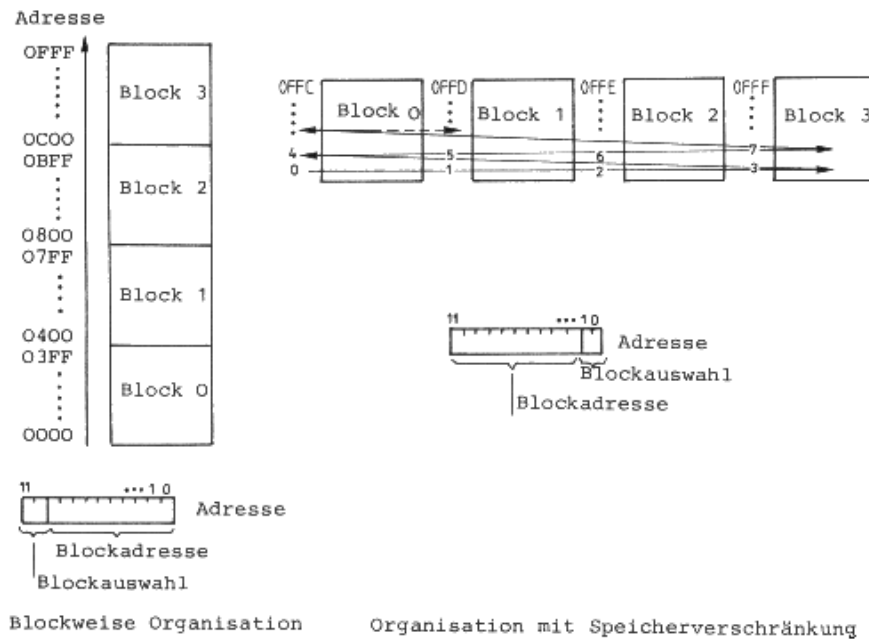


Die höherwertigen Bits der Adresse bestimmen die Blocknummer, der niederwertige Adressteil bestimmt die Adresse innerhalb eines Blocks → aufeinanderfolgende Befehle/Daten befinden sich i.a. im gleichen Block → Einfachste und bei kleineren Systemen fast ausschließlich verwendete Organisation. Ein Speicherblock kann ohne weiteres aus mehreren Bausteinen bestehen, wenn z. B. die Wortbreite eines Bausteins kleiner als die Datenwortbreite ist. Ein Speicherwort ist dann in mehreren parallelen Bausteinen gespeichert, z.B. 16K x 8 Bit - Block = 8 Bausteine 16 K x 1.

Speicheradressregister und Speicherdatenregister können bei dieser Organisationsform auch fehlen (und fehlen auch bei Mikrocomputer-Systemen). Es muss nur sichergestellt sein, dass Adressen und Daten auf dem Bus genügend lange bereitstehen. Zum Teil sind entsprechende Register in den Speicherbausteinen selbst enthalten. Die Datenausgänge der Speicherbausteine sind i.a. als Tri-State-Ausgänge gestaltet und können direkt auf den Bus geschaltet werden (TriState = Möglichkeit, den Ausgang hochohmig zu schalten, so dass er den Datenbus nicht beeinflusst).

7.4.2 Speicherverschränkung = Adressverschränkung = "memory interleave"

Hier liegen aufeinanderfolgende Adressen in aufeinanderfolgenden Speicherblöcken (zyklisch!) → physikalischer Speicherblock ungleich logischer Speicherblock. Die niederwertigen Bits der Adresse bestimmen die Blocknummer, der höherwertige Adressteil bestimmt die Adresse innerhalb eines Blocks (also genau umgekehrt, wie vorher). Bild: Umdruck. Üblich ist eine 2-fache bis 8-fache Verschränkung.



Diese Organisationsform eignet sich zur Beschleunigung des Speicherzugriffs - insbesondere bei blockweiser Datenübertragung - da hier die "effektive" Zugriffsgeschwindigkeit erhöht werden kann. Da aufeinanderfolgende Adressen in verschiedenen Blöcken liegen, kann ein neuer Speicherzyklus bereits angestoßen werden, während der vorhergehende Speicherzyklus noch andauert → Speicher mit überlappendem Zyklus. Die tatsächlich erreichte Geschwindigkeit hängt von der Adress-Reihenfolge beim Zugriff ab. Die Beschleunigung wirkt nur dann optimal, wenn immer zu aufeinanderfolgende Adressen zugegriffen wird.

Da die für einen Speicherzugriff benötigte Information (Adresse, Datum) nach wie vor während der gesamten realen Zugriffszeit benötigt wird, müssen für jeden Block Speicheradressregister, Speicherdatenregister und Register für Steuersignale vorhanden sein. Diese sind heute in der Regel im Speicherbaustein integriert oder werden von einem speziellen Baustein zur Speicheransteuerung (Memory Management Unit) bereitgestellt.

7.5 Pufferspeicher (Cache)

Speicher mit sehr kurzer Zugriffszeit und relativ kleiner Kapazität der zwischen Arbeitsspeicher und CPU geschaltet ist. Er enthält den jeweils "aktuellen" Teil des Arbeitsspeicher- Inhalts. Im Idealfall befindet sich der Inhalt der jeweils durch die CPU adressierten Arbeitsspeicher-Zelle im Cache → bei jedem Speicherzugriff ist nur die Zykluszeit des Cache und nicht jene des Arbeitsspeichers abzuwarten (Geschwindigkeitszuwachs um den Faktor 4 ... 5). Für den Programmierer und die CPU ist der Cache "transparent", d. h. es spielt keine Rolle, ob das referierte Wort im Arbeitsspeicher oder im Cache steht → die DVS hat scheinbar einen Arbeitsspeicher mit der Kapazität des realen Arbeitsspeichers und der Zykluszeit des Cache.

Im realen Betrieb ist die Wahrscheinlichkeit, daß ein referiertes Arbeitsspeicher-Wort im Cache steht (=Trefferate H), immer kleiner 1 ("page fault"). Typische Werte für H liegen im Bereich 50 bis 98%. Die Trefferrate wird umso größer sein, je besser es gelingt das "working set" des aktuellen Programms im Cache zu halten. Auf Grund der Lokalitätseigenschaft von Programmen (die überwiegende Zahl von Speicherzugriffen erfolgt "in der Nähe" der zuletzt referierten Adresse), ist die Wahrscheinlichkeit, dass die benötigte Info bereits im Cache steht, relativ groß.

Der Cache wird immer dann mit neuen Werten geladen, wenn ein Speicherzugriff zu einem nicht in ihm enthaltenen Wert erfolgt. In diesem Fall wird immer ein ganzer Datenblock ab der gerade referierten Adresse in den Cache geladen (typisch 4 - 16 Worte). Ursprünglich wurden Pufferspeicher nur bei Großrechnern eingesetzt, heute gibt es Prozessoren mit Cache auf dem Chip (z. B. 68020), wobei der Cache aber funktionell zum Speicherwerk gehört. Für andere Prozessoren (z.B. 80386) gibt es spezielle Cache-Controller (z.B. 82385), die zwischen Prozessor und Speicherwerk geschaltet werden und den Speicher unabhängig vom Prozessor ansteuern.

7.6 Neue Speichermedien

Memory Stick



Seit Ende 1998 ist der von Sony entwickelte Memory Stick erhältlich. Er ist 21 x 50 x 2,8 mm groß und vier Gramm leicht. Dieser Speicher wird bislang fast ausschließlich in Sony-Produkten eingesetzt, beispielsweise im Sony-Walkman, der Musikdateien im ATRAC3-Format abspielt oder in den Geräten der Vaio-Serie. Der Memory Stick basiert auf Flash-Technik. Derzeit gibt es den "normalen" Memory Stick und den Memory Stick mit Kopierschutz "Magic Gate". Die "normalen" sind blau und mit Kapazitäten bis 128 MB erhältlich. Die weißen Sticks hingegen sind für die Aufnahme und Wiedergabe von Musik konzipiert und mit einem Kopierschutz ausgestattet. Die Musik wird verschlüsselt gespeichert. Erkennbar ist dieser Memory Stick auch an dem Aufdruck MG "Magic Gate". Der Anschluss an ein Notebook funktioniert über einen PC-Card-Adapter. Mit Adaptern für Diskettenlaufwerk, USB-Schnittstelle und Parallel-Port lassen sich die Daten auf den Computer übertragen. Auf den MB-Preis umgerechnet sind die Memory Sticks teuer.

Smart Media Card

Die nur knapp 0,8 mm hohen Smart-Media-Karten sind etwa halb so groß wie eine Scheckkarte. Durch die geringe Bauhöhe passt die Elektronik zur Steuerung (Controller) nicht mehr mit auf die Karte. Damit ist die SmartMedia von den Geräten abhängig, in denen sie eingesetzt wird. Kann ein Gerät nur 64 MB verwalten, nützt eine 128 MB-Karte nichts.

SmartMedia-Karten werden sowohl in MP3-Playern, als auch in Digitalkameras eingesetzt. Allerdings können Musik- und Bilddaten nicht gleichzeitig darauf gespeichert werden, denn die beiden Datentypen benötigen eine unterschiedliche Formatierung der Karte. Die alten 2- und 4-MB-Karten arbeiten mit verschiedenen Spannungen: 5,5 Volt oder 3 Volt. Damit keine Verwechslungen stattfinden, haben die 5,5-Volt-Karten eine abgeschrägte Ecke. So können Karten, die eine 3-Volt-Versorgung benötigen, nicht aus Versehen ins falsche Gerät eingesetzt werden.



Compact Flash Card



Die CompactFlash Cards wurden bereits 1994 von SanDisk eingeführt. Seit 1995 kümmert sich die CompactFlash Association um die Standardisierung der CompactFlash Cards. Die Speicherkarten sind mit ihren Abmessungen von 43 x 36 x 3,3 mm etwa halb so groß wie eine PC-Card. Sie beherbergen einen Flash-Speicherbaustein und einen Controller, der sie bis auf den Stecker zu einer AT-Bus-Festplatte kompatibel macht. Deshalb gibt es bei Geräten mit CF-Steckplatz keine Probleme, wenn neue Medien auf den Markt kommen. Die Daten gelangen dann am einfachsten über einen externen Card-Reader in den PC. Sie sind robuster als ihre SmartMedia-Kollegen und haben eine höhere Zukunftssicherheit sowie Kapazität. Die momentane Obergrenze beim Speicher liegt bei 512 Megabyte.

Multimedia Card

Mit einer Größe von 24 x 32 x 1,4 mm sind sie die kleinsten Medien unter den mobilen Speichern und nur halb so groß wie Compact Flash Karten. Die nur 1,5 Gramm leichten Karten haben eine maximale Kapazität von 128 MB. Aufgrund ihrer Größe eignen sie sich besonders gut für den Einsatz in Mobiltelefonen. Nokias Communicator zum Beispiel oder das Siemens SL 45 sind mit Multimedia Cards erweiterbar. Mit einer Card-Station lassen sich Daten über die Parallel- und USB-Schnittstelle auf den PC übertragen. Die Steuerelektronik für den Speicher (Controller) ist auf der Karte eingebaut.



[Zum vorhergehenden
Abschnitt](#)

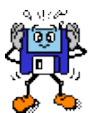


[Zum Inhaltsverzeichnis](#)



[Zum nächsten
Abschnitt](#)

*Die HTML-Fassung entstand unter Mitwirkung von Volker Arndt
Copyright © FH München, FB 04, Prof. Jürgen Plate*



Einführung Datenverarbeitungssysteme

von Prof. Jürgen Plate

8. Rechnerperipherie

Die Gesamtheit der Einrichtungen, über welche die CPU mit den peripheren Geräten kommuniziert, nennt man E/A-Werk. I.a. findet ein (bidirektionaler) Datentransport zwischen Speicher und Peripherie statt. Neben dem reinen Datentransport ist eine Anpassung der Datendarstellung zwischen CPU und Peripherie erforderlich = Umsetzung der Protokolle Bus - Peripherieschnittstelle: Zeitliche Anpassung (CPU und Peripherie arbeiten i.a. nicht synchron, d.h. sie sind zeitlich zu entkoppeln).

- Signalumsetzung (z.B. Pegelanpassung)
- Datenformatanpassung (z.B. seriell/parallel-Wandlung)
- Codewandlung
- Fehlererkennung und -korrektur Das E/A-Werk kann von unterschiedlicher Komplexität sein:
 - Im einfachsten Fall besteht es aus einer oder mehreren einfachen nicht-intelligenten E/A-Schnittstellen zur rein hardwaremäßigen Anpassung und Umsetzung. Die gesamte Steuerung des Datentransfers erfolgt durch die CPU (Leitwerk) → programmierter E/A-Transfer.
 - Komplexere Schnittstellen können den Datentransfer autonom durchführen. Sie werden von der CPU nur angestoßen und wickeln den Datentransfer ohne weitere Hilfe der CPU ab → Direkt- Speicher-Zugriff (DMA = Direct Memory Access) ohne Ablauf eigener Programme.
 - Flexiblere und leistungsfähigere Schnittstellen besitzen einen speziellen E/A-Prozessor, der nach einem Anstoß durch die CPU unabhängig vom (Haupt-)Prozessor mit eigenem Programm arbeitet und getrennten Zugriff zum Speicher besitzt → Kanäle (Großrechner) oder I/O-Prozessoren (Micro-Computer.). Kanäle gestatten eine eingeschränkte Fehlererkennung.
 - Eine noch größere Entlastung der zentralen CPU bieten eigenständige E/A-Rechner. Sie können auch Codewandlung und eine allgemeine Fehlererkennung und -korrektur durchführen.

8.1 Methoden des Datentransfers

8.1.1 Programmierter E/A-Transfer

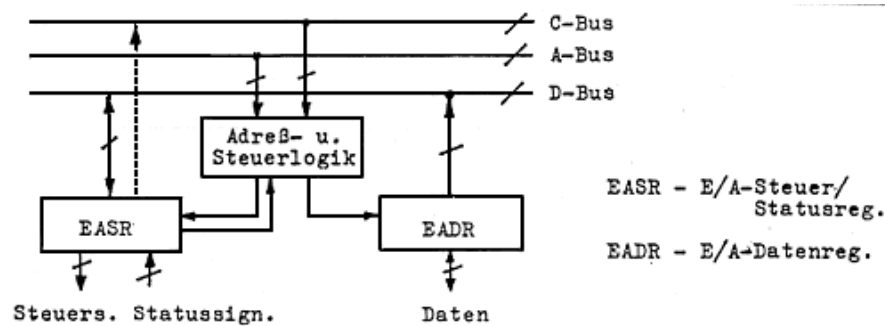
Einfache E/A-Schnittstellen für programmierten E/A-Transfer bestehen wesentlich aus (siehe Umdruck):

- E/A-Datenregister (EADR)
- E/A-Steuer- und Statusregister (EASR)
- Adreß- und Steuerlogik

Sie ist normalerweise an den Systembus der CPU angeschlossen. Das EADR dient zur Zwischenspeicherung des zu übertragenden Datenwerts (zeitliche Anpassung). Neben den zu übertragenden Daten braucht die Schnittstelle Signale zur richtigen Abwicklung des Datentransfers:

- Statussignale von der Peripherie
z. B. Anzeige, dass das Peripheriegerät einen neuen Wert ins EADR geliefert hat oder bereit ist, den nächsten Wert zu übernehmen.
- Steuersignale zur Peripherie
z. B. Anzeige, dass der vorhergehende Datenwert gelesen wurde und die Peripherie einen neuen Wert ins EADR einschreiben kann.
- Steuersignale zur Schnittstelle selbst
z. B. Signale zum Einstellen bestimmter Betriebsmodi bei programmierbaren Schnittstellen.

Datenverarbeitungssysteme



Diese Signale werden - soweit sie für längere Zeit zur Verfügung stehen müssen - im EASR gespeichert. Das EASR kann auch aus separaten Registern für Status- und Steuersignale bestehen. Normalerweise verfügt ein Computer über mehr als eine E/A-Schnittstellen → Adressierung erforderlich. Für die Adressierung der E/A-Schnittstellen gibt es zwei Möglichkeiten:

- **Getrennte Adressierung** (Isolated I/O, I/O-Mapped I/O)

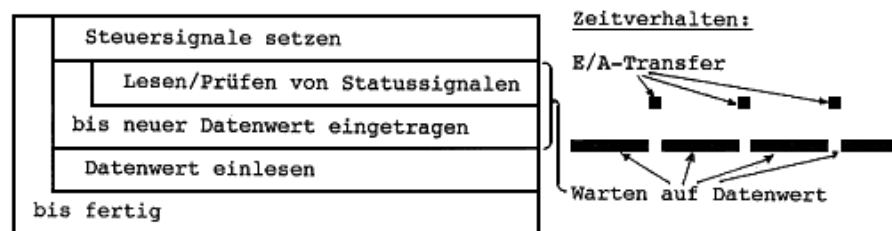
Der Rechner verfügt über spezielle E/A-Befehle und einen eigenen E/A-Adressraum. Die E/A-Adresse wird oft auch Port genannt. Sie wird entweder auf separaten Adressleitungen zu den Schnittstellen geführt oder über den normalen Adressbus geleitet. In diesem Fall gibt es (eine) zusätzliche Steuerleitung(en) zur Unterscheidung zwischen E/A- und Speicheradressen. Da i.a. der E/A-Adressraum kleiner als der Speicheradressraum ist, wird nur ein Teil der A-Bus-Leitungen verwendet (z.B. 8080/8085: 8 Bit, 80386: 10 Bit).

- **Speicheradressierung** (Memory Mapped I/O)

Der E/A-Adressraum wird in den Speicher-Adressraum abgebildet, d.h. bestimmte Adressen stellen keine Speicheradressen, sondern E/A-Adressen dar → geringfügige Einschränkung des Speicheradressraums. Die E/A-Adressen werden von der CPU genauso behandelt und angesprochen, wie Speicheradressen. E/A-Operationen werden mit den normalen Transportbefehlen durchgeführt (z. B. Load, Store) → Einsparung zusätzlichen Aufwandes für Adress-/Steuerleitungen und für spezielle E/A-Befehle.

Im allgemeinen erhalten EADR und EASR eine eigene, separate Adresse. Beide sind am Datenbus angeschlossen. Steuersignale werden wie Datenwerte ins EASR geschrieben, Statussignale werden wie Datenwerte aus dem EASR gelesen - die Verarbeitung muss in der CPU durch geeignete Befehle erfolgen (z.B. log. Verknüpfung).

Programmgesteuerter programmierter Transfer

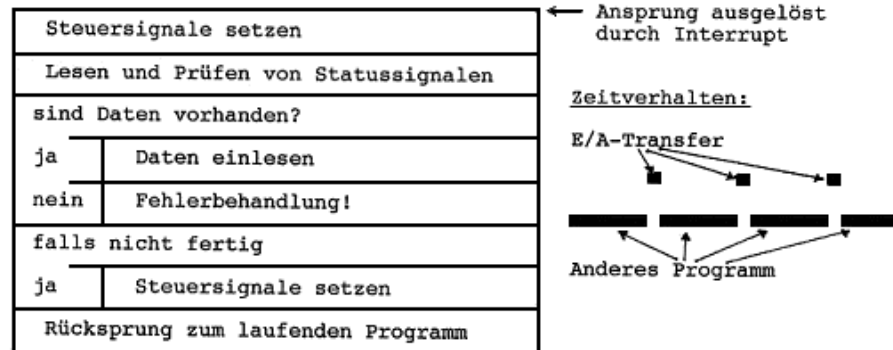


Die E/A-Befehlsfolge steht innerhalb eines Programms und wird im Rahmen der normalen Programmausführung abgearbeitet. Die gesamte Initiative zur E/A geht vom Programm aus. Zeigt der Inhalt des EASR an, dass keine neuen Daten im EADR stehen, muss das Programm erneut das EASR abfragen → Warteschleife, "Polling". Dabei können zwei "Randbereiche" Probleme verursachen:

- Die Zeit, die das Programm braucht, um einen Eingabewert zu bearbeiten ist manchmal (oder sogar immer) länger als die Zeitspanne innerhalb der sich das Eingangssignal ändern kann → es werden Ereignisse "übersehen".

- Bei "langsamen" Peripheriegeräten ist der Computer während der Übertragung eines Datenblocks die meiste Zeit mit Abfragen beschäftigt, die in den meisten Fällen negativ ausfallen.

Unterbrechungsgesteuerter programmierter Transfer



Die E/A-Befehlsfolge wird durch einen von der E/A-Schnittstelle ausgelösten Interrupt gestartet. Der Interrupt wird von der Schnittstelle abgesetzt, wenn ein Datenwort empfangen wurde und im EADR steht (bzw. wenn ein Datenwort aus dem EADR abgeholt wurde). Die CPU braucht nicht in einer Programmschleife zu warten, sondern kann in der Zwischenzeit ein anderes Programm bearbeiten, das durch den Interrupt kurzzeitig unterbrochen wird. Der interruptgesteuerte Transfer gestattet zudem die quasi-simultane Bedienung mehrerer langsamer Schnittstellen. Zur Realisierung muß eine Interrupt-Leitung von der E/A-Schnittstelle (EASR) zum Unterbrechungswerk der CPU vorhanden sein.

Vorteil des programmierten Transfers: Sehr einfache Schnittstellen, die übersichtlich und leicht zu verwenden sind.

Nachteil des programmierten Transfers: Für den Transfer jedes einzelnen Datenworts wird die CPU benötigt. Diese muss dabei relativ viel Verwaltungsaufwand leisten (Laden/lesen EASR, Interrupt-Behandlung) → die maximale Datenrate ist sehr begrenzt.

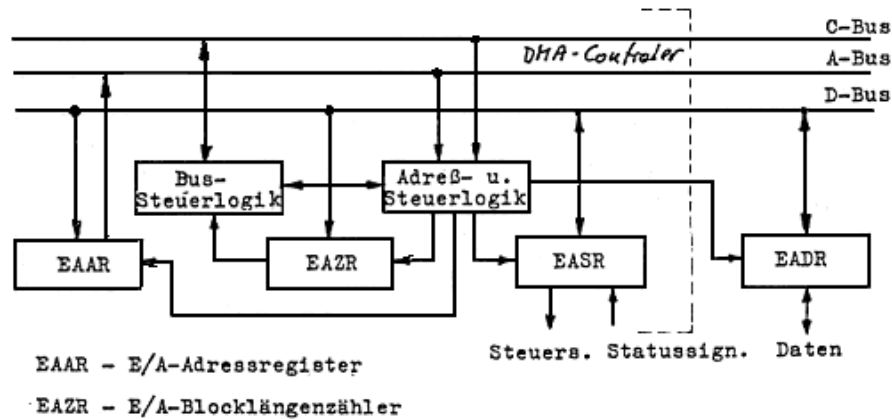
8.1.2 Direktspeicherzugriff (DMA = Direct Memory Access)

Beim DMA steuert die E/A-Schnittstelle den Transfer nach Anstoß durch die CPU selbstständig und ohne Zuhilfenahme der CPU → Datentransport zwischen Speicher und Peripherie unter Umgehung der CPU. Selbst, wenn die CPU in dieser Zeit nichts anderes tut, ist ein wesentlich schnellerer Datentransport möglich (weitgehender Wegfall der Verwaltungsarbeit). Daher ist DMA für den schnellen Transfer großer Datenmengen besonders geeignet. Zum Anstoßen des DMA-Transfers übermittelt die CPU der DMA-Schnittstelle:

- die Anfangsadresse (im ASp) des zu übertragenden Datenblocks
- die Länge des zu übertragenden Datenblocks
- Gegebenenfalls muss noch - wie beim programmierten Transfer - das EASR entsprechend geladen werden → Initialisierung der Schnittstelle.

Die DMA-Schnittstelle benötigt zwei weitere Register:

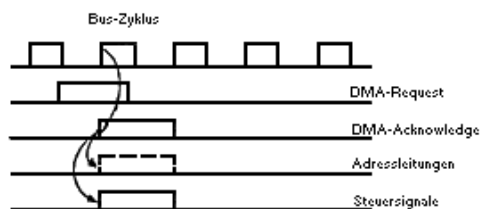
- E/A-Adreßregister (EAAR)
- E/A-Blocklängenzähler (EAZR)



Da während des DMA-Transfers anstelle der CPU die DMA-Schnittstelle die Kontrolle über den Systembus besitzt, muss sie zusätzlich über eine Bus-Steuerlogik verfügen (Konkurrenz zwischen DMA Schnittstelle und CPU um den Bus).

Ablauf eines DMA-Transfers:

- Initialisieren der DMA-Schnittstelle
- Bus-Freigabe-Anforderung an die CPU (DMA-Request)
- CPU bestätigt Bus-Freigabe (DMA-Acknowledge) (Die Bus-Freigabe der CPU geschieht durch Umschalten der Bustreiber-Ausgänge in den hochohmigen Zustand)
- Nun bedient die DMA-Schnittstelle den Adreß-, Daten- und Steuerbus auf die gleiche Art und Weise, wie die CPU. Der Inhalt des EAAR wird an den A-Bus gelegt und die für die Speichersteuerung notwendigen Signale auf den Steuerbus. Das EADR wird mit dem Datenbus verbunden.



Man unterscheidet verschiedene DMA-Modi:

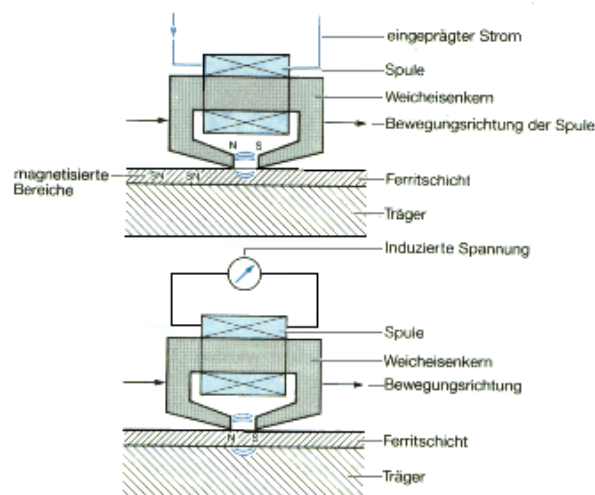
- Byteweise Übertragung (Byte Mode)
Die CPU gibt auf DMA-Request hin den Bus nach Beendigung des gerade laufenden Maschinenzyklus frei. Nach dem Transfer eines Datenwortes erhält die CPU die Kontrolle über den Bus zurück und sie setzt den laufenden Befehl weiter fort (bis zum nächsten DMA-Request) → Cycle Stealing.
- Burst-Modus
Auch hier wird die CPU mitten in der Bearbeitung eines Befehls kurz angehalten, jedoch für mehrere Zyklen → Transfer eines ganzen Datenblocks.
- Halt-Modus
Die CPU wird nach Beendigung der gerade laufenden Instruktion angehalten - solange, bis der DMA-Transfer eines ganzen Blocks abgewickelt ist.
- Transparent-Modus
DMA-Schnittstelle und CPU arbeiten zeitmultiplex während jedes Maschinenzyklus jeweils in einer halben Periode. Dazu ist ein sehr schneller Speicher erforderlich → keine Beeinträchtigung der CPU.

8.2. Periphere Speicher (Massenspeicher)

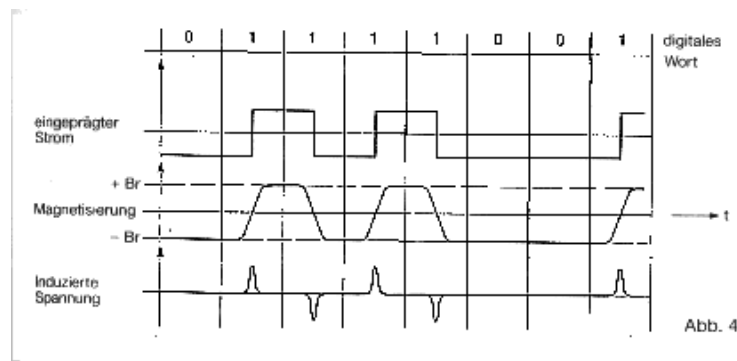
Periphere Speicher sind Speicher, die sich außerhalb der Zentraleinheit befinden (Sekundärspeicher). Sie dienen der Speicherung von Informationen, die nicht immer aktuell in Arbeitsspeicher benötigt werden. Sie besitzen wesentliche höhere Speicherkapazität als der ASp, haben aber auch eine größere Zugriffszeit. Der Datentransport zwischen peripheren Speicher und ASp findet blockweise statt (blockorganisierte Speicher). Funktionell können sie als Erweiterung des ASp aufgefasst werden (siehe virtuelle Speicher), strukturell stellen sie E/A-Geräte dar. Heute werden als periphere Speicher vornehmlich magnetomotorische Speicher eingesetzt: Magnetplatte, Magnetband und magnetooptische Speicher.

8.2.1 Speicherprinzip magnetomotorischer Speicher

Auf dem Magnetband bzw. einer als Träger dienenden runden Scheibe befindet sich eine Schicht aus hartmagnetischem Material (Eisenoxid mit Zusätzen anderer Ferrite) - wie z.B. auch bei einem Tonband zur Musikaufzeichnung. Die nadelförmigen Ferrite mit etwa 1μ Länge und $0,1\mu$ Dicke sind in Bewegungsrichtung ausgerichtet. Wie beim Magnetkern wird die Materialeigenschaft ausgenutzt, unter dem Einfluss eines äußeren Magnetfeldes eine nach Betrag und Richtung bestimmte remanente Magnetisierung anzunehmen. Das erregende Feld wird mittels eines Magnetkopfes erzeugt, an dem sich die Platte mit konstanter Umdrehungsgeschwindigkeit vorbei bewegt → Relativbewegung zwischen Speichermedium und Erregerfeld → Ausrichtung der magnetischen Elementardipole in der Bewegungsrichtung.



Die Magnetisierung erfolgt bis zur Sättigung. Den einzelnen Bits sind nicht diskrete Elemente zugeordnet, sondern kleine Bereiche eines kontinuierlichen Mediums. Speicherelement ist somit der einem Bit zugeordnete Bereich. Beim Lesen erzeugt die örtliche Änderung der magnetischen Kraftflussdichte auf dem bewegten Speichermedium aufgrund der Relativbewegung Kopf-Platte eine zeitliche Änderung der die Kopfspule durchströmenden Flussdichte → Induzieren einer Spannung.



Normalerweise wird für das Schreiben und Lesen derselbe Magnetkopf verwendet. Um eine hohe Aufzeichnungsdichte zu erreichen, muss der Kopfspalt möglichst klein sein. Bei Ferritköpfen erreicht man eine Spaltbreite von 1μ , bei in Dünnschicht-Technik hergestellten Köpfen sogar $0,1\mu$.

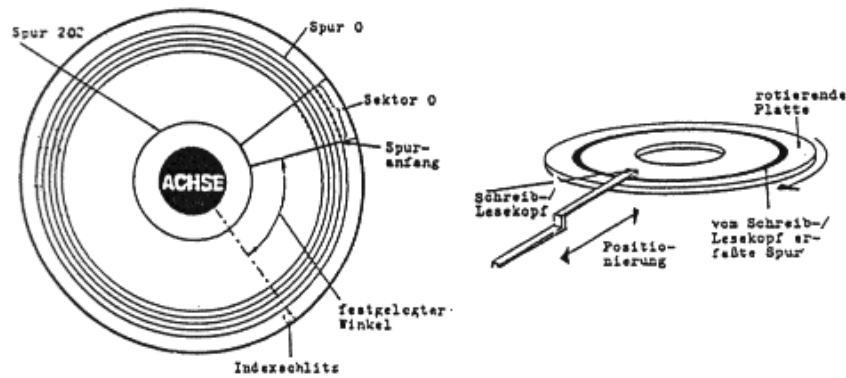
8.2.2 Magnetplattenspeicher

Der Magnetplattenspeicher ist ein blockorganisierter Speicher mit quasi-wahlfreiem Zugriff. Es gibt ihn in verschiedenen Arten und Ausführungen. Die Speicherung erfolgt in konzentrischen Spuren. Zum Aufsuchen einer bestimmten Spur lässt sich der Magnetkopf in radialer Richtung verschieben.

Bei Systemen mit größerem Abstand der Spuren voneinander (z. B. Floppy-Disk, bis zu 135 Spuren/Zoll) kann die Positionierung des Kopfes mittels eines Schrittmotors geschehen. Bei Systemen mit kleinem Spurbabstand (z.B. Festplatte, bis zu 1000 Spuren/Zoll) verwendet man ein Tauchspulensystem mit Lageregelung (closed loop), das eine sehr präzise Steuerung des Kopfes erlaubt. Jede Spur ist durch eine Nummer gekennzeichnet, unter der sie adressiert werden kann → Spuradresse (äußerste Spur: 0). Zur feineren Adressierbarkeit der gespeicherten Information ist jede Spur in einzelne Abschnitte (Sektoren) unterteilt → Sektoradresse. Der Spuranfang wird oft durch einen Schlitz oder ein Loch an bzw. auf der Platte (Indexschlitz, Indexloch) festgelegt. Er markiert den Beginn des Sektors 0.

Die Festlegung der Sektoren erfolgt durch spezielle, zusätzlich zu den Daten auf die Platte geschriebene, Informationen. Das Aufschreiben der o.g. Verwaltungsinformationen muss vor Verwendung der Platte erfolgen → Formatierung. Ein oder mehrere Sektoren bilden einen Block (Cluster), der mit einem Zugriff zwischen Platte und CPU transportiert werden kann. Neben den eigentlichen Daten enthält jede Spur Adress- und Formatierungsinformationen:

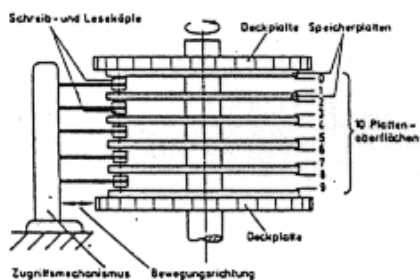
- Spuradresse
- Sektoradressen
- weitere Markierungs- und Kennzeichnungsinfo
- Zwischenräume (Gaps)
- Prüfinfo zur Fehlererkennung (CRC-Prüfsumme) (für Adressinfo-, Formatierungsinfo und eigentliche Daten)



Die Speicherkapazität für Nutzdaten ist nun zwar geringer geworden, die Formatierung erlaubt aber auch Datenträger zu lesen, die auf einem Gerät mit (innerhalb einer Toleranzgrenze) abweichenden mechanischen Parameter beschrieben wurden. Die Aufzeichnung der Daten erfolgt bytewise bitseriell (MSB zuerst). Ein zusätzliches Prüfbit wird nicht aufgezeichnet, sondern eine Polynom-Prüfsumme für jeden Block.

Arten von Magnetplattenspeichern

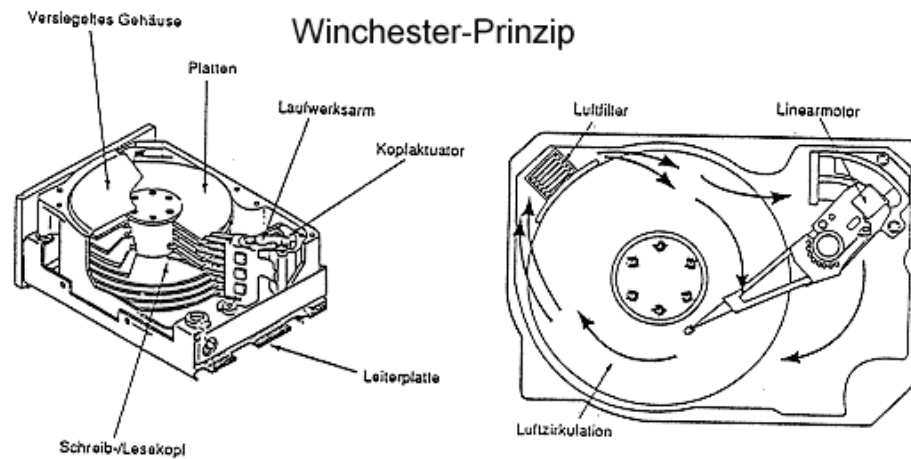
- Magnetplatte (Festplatte, Harddisk)
 - ◆ Einzelplatte
 - ◆ Einzelplattenkassette
 - ◆ Festkopffplatte
 - ◆ Festplattenstapel
 - ◆ Wechsellplattenstapel
- Magnetfolienspeicher (Floppy Disk)
 - ◆ Einzeldiskette
 - ◆ ZIP-Platte
 - ◆ Floptical



Beim Plattenstapel wird oft die oberste und/oder unterste Plattenfläche oft nicht für Info-Speicherung verwendet (Schutz, Sektorplatte, Servoinfo für Steuerung). Pro Fläche ein eigener Kopf; alle Köpfe fest gekoppelt (Kamm).

Geschichtliches:

- Erste Festplatte 1956 bei IBM vorgestellt. 24" Plattendurchmesser, 5 MByte Kapazität.
- 1973 entwickelt IBM unter dem Namen "Winchester" ein Plattenlaufwerk mit 14" Durchmesser. Hier befindet sich die Platte in einem "versiegelten" Raum, der über Mikrofilter mit Luft versorgt wird. Die Köpfe wiegen nur etwa 10g und "fliegen" in extrem geringer Höhe (0.5μ) über die Platte. Durch spezielle aerodynamische Bauform des Kopfes wird dieser Effekt durch die Rotation der Platte (mitgerissene Luft) erreicht → Kopf gleitet auf Luftpolster (Bernoulli-Effekt).



- 1977 brachte Shugart das erste preiswerte Laufwerk auf den Markt (14", 30 MByte). Die weitere Entwicklung führte zu kleineren Platten (8", 5,25", 3½"), z.B. ST 506: 5,25", 6,4 MByte.
- 1983 Winchester-Wechselplatte
- 1988 Plattenlaufwerke höherer Kapazität, z.B. NEC D 5662: 5,25", 319 MByte (1224 Cylinder, 15 Köpfe).
- 1992 1,8"-Platte mit 60 MByte, 5,25"-Platte mit 1830 MByte
- 2000 3,5"-Platte mit 10 GByte für unter 200 DM

Die Platte rotiert mit konstanter Geschwindigkeit (z. B. die ersten Winchester-Platten: 3600 U/min). Der Zugriff setzt sich aus zwei Phasen zusammen:

- Auswahl der Spur: Bewegung des Kopfes in radialer Richtung (Positionieren)
- Auswahl des Sektors
- Lesen der Sektoradresse auf der sich unter dem feststehenden Kopf vorbeibewegenden Plattenspur, bis der gewünschte Sektor gefunden ist.

→ mittlere Zugriffszeit = Positionierzeit + 1/2 Umdrehungszeit

Als "Zylinder" bezeichnet man die Gesamtheit aller senkrecht übereinanderliegender Spuren (bei Einzelplatten die beiden Seiten, bei Plattenstapeln alle aktive Flächen) = alle Spuren auf allen Flächen mit gleicher Nummer = alle Spuren, die mit einer Kopfpositionierung angesprochen werden. Die Spuradresse wird zusammengesetzt aus:

- Zylinderadresse (Spurnummer)
- Kopfadresse (Nummer der Speicherfläche; beginnend bei 0 von oben nach unten gezählt)

Die sogenannte Aufzeichnungsdichte wird in BPI (Bits per Inch = Bits pro Zoll) angegeben. Diese erreicht Werte von 40000 BPI und mehr. Ein ebenso gebräuchliches Maß für die Aufzeichnungsdichte ist "Flux Changes per Inch" (FCI). Übersetzt bedeutet das soviel wie "Flußwechsel pro Zoll" und gibt an, wie oft die Ausrichtung der Magnetpartikel pro Zoll geändert werden kann, denn der Abstand zwischen zwei Flußwechseln kann eine bestimmte Grenze aus physikalischen Gründen nicht unterschreiten. Je höher jedoch die Werte von BPI beziehungsweise FCI sind, desto mehr Daten lassen sich auf der Festplatte unterbringen.

Schon bald entwickelte man ein Verfahren, um den Platz auf den äußeren, längeren Spuren besser zu nutzen: "Zone-Bit-Recording" (ZBR). Die Platte wird hierzu in mehrere Spurguppen eingeteilt. Dabei wird für jede Gruppe die maximale Anzahl von Sektoren pro Spur bestimmt. Je kleiner diese Gruppen sind, desto besser ist die Ausnutzung der Platte. Im Idealfall würde für jede Spur der optimale Wert errechnet. Der Rechneraufwand für den Controller würde in diesem idealen Fall stark ansteigen, da er bei jedem Zugriff erst berechnen müßte, wie viele Sektoren auf der zu

lesenden Spur untergebracht sind. Um den Rechenaufwand gering zu halten, faßt man mehrere Spuren zu einer "Zone" zusammen, in der die Sektorenanzahl der Spuren gleich ist. Das Verfahren ist also ein Kompromiß aus Geschwindigkeit und Platzgewinn.

Ansteuerung der Festplatte

Es gibt zwei Arten von Interfaces zur Festplattenansteuerung, die in der Computerwelt sehr verbreitet sind. Einmal das kostengünstige, aber ziemlich unflexible IDE-Protokoll (*Integrated Drive Electronics*, auch als ATA, AT Attachment, bekannt), auf der anderen Seite gibt es das teurere und vielseitig verwendbare SCSI-Protokoll (*Small Computer Systems Interface*). Der für den Normalanwender offensichtlichste Unterschied zwischen beiden Techniken ist der Preis. SCSI-Festplatten sind bei gleicher Größe und Geschwindigkeit ungefähr doppelt so teuer wie IDE-Platten. Dies hat u. a. seine Gründe im Aufbau der Platten.

An ein normales E-IDE-System kann man normalerweise bis zu vier Gräte anschließen, dabei werden je zwei Geräte an einen IDE-Port angeschlossen. Die beiden Ports bezeichnet man als primären und sekundären Anschluss. Die beiden Geräte an jedem Port werden in Master und Slave aufgeteilt. Ein E-IDE-System bootet (normalerweise) von der Master-Platte am primären Port. Der IDE-Bus war ursprünglich nur zum Anschluss von Festplatten gedacht, mittlerweile kann man aber auch CD-ROM-Laufwerke und Brenner, Bandlaufwerke und große Diskettenlaufwerke anschließen.

Im Gegensatz dazu kann man an einem SCSI-Controller bis zu sieben Gräte betreiben, bei Wide-SCSI sogar bis zu 15 Geräte, die jeweils über eine eindeutige ID-Nummer angesteuert werden. SCSI war schon von Anfang an dafür ausgelegt, Geräte aller Art ansteuern zu können, so ist es nicht verwunderlich, dass man an den SCSI-Bus außer den Geräten, die man bei IDE findet, auch noch Dinge wie Scanner anschließen kann.

Da an den IDE-Bus nur je zwei Geräte angeschlossen werden können, sind keine besonderen Maßnahmen zur Abschirmung getroffen worden (das hat sich aber bei Ultra-ATA2 geändert, hier wird ein 80-poliges Kabel verwendet, wobei die 40 zusätzlichen Adern nur der Abschirmung dienen). Im Gegensatz dazu sind SCSI-Kabel robuster, was elektrische Störstrahlung angeht, des weiteren sorgen Terminatoren (Abschlusswiderstände) für Sicherheit. An jedem Ende des Busses muss ein Terminator befestigt werden, der eventuelle Signalreflexionen an den Kabelenden verhindert. (Beispiel: Nur interne Geräte --> Terminator am Hostadapter (meist automatisch) und am hintersten Gerät am Kabel; oder: interne und externe Geräte > Terminator am äußersten externen und äußersten internen Gerät, keine Terminierung am Hostadapter.)

Während bei SCSI-Platten die Ansteuerungselektronik zu großen Teilen auf einem (teilweise recht teuren) Host-Adapter untergebracht ist, befindet sich diese bei IDE-Platten im Festplattengehäuse.

Geschwindigkeit

Lange Zeit galt, dass SCSI-Platten besonders schnell sind, dies ist aber schon seit einiger Zeit nicht mehr so, denn auch die IDE-Front hat sich rasant entwickelt, so dass sich beide Systeme in punkto realer Plattengeschwindigkeit nichts mehr nehmen. Die Übertragungsgeschwindigkeit bei Festplatten hat sich in den letzten Jahren rasant gesteigert. Das führte aber teilweise zu Problemen, denn die Geschwindigkeitssteigerungen wurden durch eine Erhöhung der Taktfrequenz auf dem IDE bzw. SCSI-Bus erreicht. Dadurch wurde die Gefahr durch elektrische Störungen größer. Deshalb ist die maximale Kabellänge immer kleiner geworden, z. B. waren IDE-Kabel vor 5 Jahren noch fast einen Meter lang, heute soll man moderne Festplatten nur an Kabel anschließen, die maximal 45 cm lang sind. Bei SCSI wurde die Geschwindigkeit zwar auch durch Takterhöhungen realisiert, allerdings kam dazu eine Verdopplung der Busbreite von 8 auf 16 Bit (Wide-SCSI). Im gleichen Zuge wurde dabei auch die Abschirmung der Kabel verbessert. Mit der neuesten Technik

(Ultra2-Wide) wurde die Signalqualität nochmals verbessert, so dass trotz erneuter Taktverdopplung auch die maximal zulässige Kabellänge vergrößert werden konnte. Die eben erwähnten Übertragungsprotokolle haben bei genauerer Betrachtung eigentlich recht wenig mit der eigentlichen Übertragungsrate einer Festplatte zu tun, sie zeigen nur, wie viele Daten *theoretisch* über die Schnittstelle transportiert werden könnten. Die reale Datenübertragungsrate hängt viel mehr davon ab, wie schnell die Scheiben mit den Daten rotieren und wie dicht die Daten auf ihnen gepackt sind. Je dichter die Daten gepackt sind und je höher die Umdrehungszahl ist, desto schneller können die Daten übertragen werden. Bei Festplatten für durchschnittliche Rechner sind 5400 U/min üblich, bei stärker belasteten Computern werden aber auch Festplatten mit 7200 oder sogar 10 000 U/min eingesetzt. Man muss dabei aber beachten, dass hohe Umdrehungszahlen auch zu einer starken Lärmbelastung führen. Neben der reinen Datenübertragungsrate ist die durchschnittliche Zugriffszeit ein weiteres Kriterium für die Geschwindigkeit einer Festplatte. Die Zugriffszeit ist die Zeit, die benötigt wird, um die angeforderten Daten zu lesen. Sie setzt sich zusammen aus der Zeit, die die Platte braucht, bis die richtige Stelle beim Lesekopf angekommen ist (also im Durchschnitt eine halbe Plattenumdrehung), was wiederum von der Umdrehungsgeschwindigkeit abhängt, und der Zeit, die der Lesekopf benötigt, um zur richtigen Spur auf der Platte zu gelangen.

Disketten

Seit der Einführung 1970 hat sich die Diskette (Floppy Disk) als schneller und preiswerter Massenspeicher durchgesetzt. Das Arbeitsprinzip ist dasselbe, wie bei der Festplatte, jedoch wird hier eine Kunststoffolie verwendet, die mit einer nichtorientierten Magnetschicht versehen ist. Die Datenaufzeichnung erfolgt entweder einseitig (SS) oder doppelseitig (DS). Zum Schutz und zur besseren Handhabung befindet sich die Scheibe in einer rechteckigen Kunststoffhülle, die mit einem Gleit- und Reinigungsvlies ausgekleidet ist. Die Hülle besitzt Öffnungen für den Arbeitskonus (über den die Scheibe angetrieben wird), das Indexloch und den Schreib/Lesekopf. Zusätzlich besitzt die Hülle noch eine Aussparung für das Setzen eines Schreibschutzes. Je nach System wird der Schreibschutz durch Abdecken oder Freilassen dieser Aussparung gesetzt (üblich: 5,25" abgedeckt = Schreibschutz, 3½" offen = Schreibschutz). Der Schreib/Lese-Kopf berührt beim Schreiben und Lesen die Diskettenoberfläche - er wird nur in den Pausen abgehoben. Die Lebensdauer liegt bei optimalen Bedingungen bei 1 .. 10 Mio. Abfragen/Spur → "Spanabhebende Datenverarbeitung". Disketten werden/wurden nach Durchmesser unterschieden:

- 8 Zoll (Standard-Diskette, veraltet)
- 5,25 Zoll (Mini-Diskette, veraltet)
- 3,5 Zoll (Mikro-Diskette - heute Standard!)
- 3 Zoll (konnte sich nicht durchsetzen)
- 2 Zoll (konnte sich nicht durchsetzen)

Gegenüber den Festplatten haben Disketten eine weitaus geringere Spurdichte (geringere Kapazität, Laufwerk mechanisch weniger präzise), eine geringere Datenrate und eine größere Zugriffszeit. Als Aufzeichnungsverfahren werden FM (SD = single density) und MFM (DD = double density) verwendet (siehe Festplatten). Die Umdrehungsgeschwindigkeit liegt zwischen 300 und 360 U/min, die Positionierzeit bei 3...10 ms und die Aufsetzzeit für den Kopf beträgt ca. 20...30 ms. Die mittlere Zugriffszeit liegt zwischen 100 und 300 ms. Die ersten 8"-Disketten hatten eine Speicherkapazität von 256 KByte, heute gibt es 3,5"-Disketten mit 1,4 MByte. Versuche einer Markteinführung von einer 2,8-MByte-Diskette sind fehlgeschlagen. ZIP-Disketten, bei denen beim Lesen und Schreiben das Bernoulli-Prinzip verwendet wird, haben Kapazitäten von 100 bzw. 250 MByte.

8.2.3 Magnetbandspeicher

Der Magnetbandspeicher ist ein blockorganisierter Speicher mit rein sequentiellm Zugriff. Auf einem flexiblen Kunststoffband ist eine ferromagnetische Schicht aufgebracht (wie bei Platte). Das Band ist auf eine Spule aufgewickelt. Wie schon bei der Platte dient auch hier die in zwei Richtungen auftretende Remanenz zur Speicherung. Durch die Bewegung des Bandes (konst. Geschwindigkeit) am Magnetkopf entlang wird auch hier eine Spannung induziert (wie beim Tonband). Zu den gespeicherten Informationen kann - im Gegensatz zur Platte - nur in der Reihenfolge der Aufzeichnung zugegriffen werden → sequentieller Zugriff.

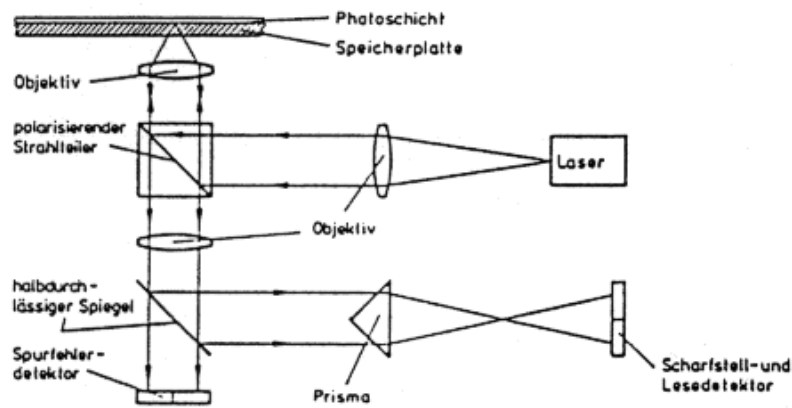
Der Magnetkopf hat Kontakt mit dem Speichermedium. Um die Abnutzung von Kopf und Band zu reduzieren, wird es nur im Bedarfsfall (beim Zugriff) bewegt → sehr schnelles Anfahren und Abbremsen des Bandes notwendig. Dazu weist die Bandführung Pufferschleifen auf, die in Unterdruckschächten geführt werden. In einfachen Geräten mit geringer Geschwindigkeit werden auch häufig Pufferarme eingesetzt. Durch den Start-Stop-Betrieb entstehen nicht beschriebene Blockzwischenräume. Der Magnetkopf weist getrennte Spaltenzonen für Lesen und Schreiben auf. Zuerst läuft das Band an der Schreibspalte vorbei, danach an der Lesespalte → sofortiges Lesen zur Schreibkontrolle. Bandanfang und -ende werden durch Reflektormarken auf der Bandrückseite gekennzeichnet und durch Photozellen detektiert.

Heute sind Magnetbänder im herkömmlichen Stil (Bandbreite 1/2", 7/8") Spuren, Bitdichte 800/1600/6250 bpi, Bandlänge 730 m) kaum noch in Verwendung. Jedoch werden Magnetbandkassetten (auch DAT-Bänder) als Backup-Medium eingesetzt. Sie sind einfacher, billiger und robuster als große Magnetbandstationen. Es gibt verschiedene, herstellerabhängige Ausführungsformen. Einige Formate sind genormt. Die Aufzeichnung erfolgt byteweise bitseriell in einer Spur (MSB zuerst - wie bei Platte) in Datenblöcken, die aus Präambel, Datenteil und Postambel bestehen (Präambel/Postambel jeweils 01010101). Der Datenteil kann zwischen 32 und 2064 Bit lang sein (einschl. 16 CRC-Bits = 2 CRC-Bytes). Als Aufzeichnungsverfahren wird i.a. Richtungstaktschrift (phase encoding) verwendet. Andere Bauformen, z.B. 14"-Kassetten werden vor allem zum Datenaustausch und zur Datensicherung bei Festplattensystemen (Winchester!) eingesetzt → Streamer.

8.2.4 Optische Speicherplatten

Bis vor wenigen Jahren waren magnetomotorische Speicher "die" Massenspeicher der DVS. Inzwischen etablieren sich optische Speicher als Massenspeicher mit sehr hoher Speicherkapazität. Das Besondere an den neuen Speicherverfahren ist der Einsatz von Lasern, die ein kohärentes Licht erzeugen (nur eine Frequenz, gleiche Phasenlage). Sie lassen eine, gegenüber Magnetplatten, wesentlich höhere Schreibdichte zu. Eine optische Platte mit 5,25 Zoll Durchmesser kann bis zu 600 MByte speichern. Auch die Sicherheit der optischen Platte ist höher, als bei Magnetplatten. Die zur Speicherung verwendete Plattenoberfläche ist durch eine transparente Kunststoffschicht geschützt und die Abtastung erfolgt berührungslos. Die Steuerung des Lese- und Schreibkopfes erfolgt prinzipiell wie bei der Magnetplatte. Auch hier ist der Kopf auf einen beweglichen Arm montiert, der radial verschiebbar ist.

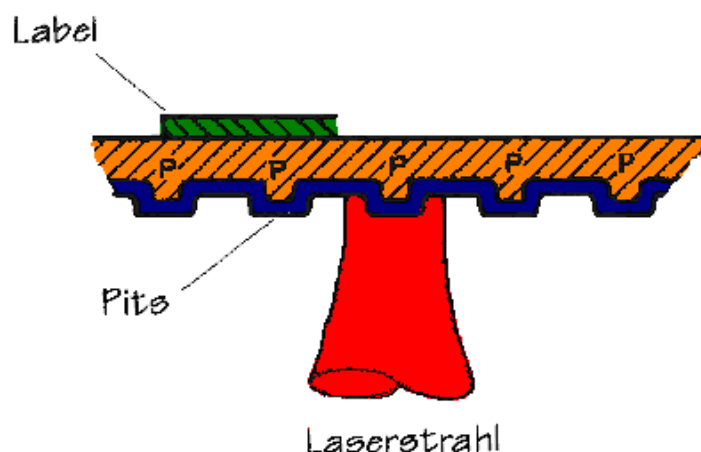
Arbeitsprinzip:



Die optische Platte dreht sich mit einer konstanten Umdrehungsgeschwindigkeit (z.B. 150 U/min). Mit dem Laserstrahl wird über ein Fokussierungs- und Spurnachführungssystem die ca. 300 Angström dicke Speicherfläche der Platte abgetastet. Der reflektierte Strahl wird in einem halbdurchlässigen Spiegel geteilt. Er trifft einmal auf eine geteilte Photodiode, den Spurfehlerdetektor, der die Nachführung des Strahls auf der Spur regelt. Das durch den Spiegel abgelenkte Strahlenbündel gelangt auf den kombinierten Fokus- und Lesedetektor. Auch dieser Detektor besteht aus einer geteilten Photodiode. Die Stromdifferenz steuert die Fokussierungseinrichtung und die Summe der Ströme bildet die Dateninformation. Der Lesekopf wird zunächst sehr schnell auf etwa 10 Spuren genau an das Ziel herangeführt. Danach erfolgt das Anfahren der richtigen Spur mittels des optischen Regelsystems auf 0.1μ genau. Die Positionierzeit beträgt etwa 100 ms. Je nach Typ der Platte unterscheiden sich die einzelnen Systeme ein wenig. Es gibt derzeit folgende Typen von optischen Speicherplatten:

CD-ROM (Compact Disk Read Only Memory)

Diese Platte ist ähnlich aufgebaut, wie die Musik-CD. Die Datenspeicherung erfolgt während der Herstellung der Platte und die Daten können nur gelesen werden (Analogie: ROM). Im Gegensatz zu Magnetplatten erfolgt die Aufzeichnung - wie bei einer Schallplatte - in einer einzigen, spiralförmigen Spur. In diese vorgeprägte, reflektierende Schicht werden bei der Herstellung der Masterplatte mit einem Laser Löcher (pits) eingebrannt. Von der Masterplatte lassen sich dann beliebig viele Kopien herstellen.



Die Kopie wird vom Laserstrahl abgetastet, der durch die unterschiedliche Struktur der Speicherfläche mit einer digitalen Information moduliert wird. Die Spurdichte beträgt bis zu 16'000 Spuren/Zoll. Als Aufzeichnungsstandard hat sich das Format ISO 9660 durchgesetzt (Transferrate: 1,2 MBit/s, Kapazität: ca. 600 MByte). Die CD-ROM dient hauptsächlich der Verbreitung größerer Datenmengen und jünger als Photo-CD.

WORM (Write Once Read Many) als Vorgänger der CD-R

WORM-Platten lassen sich vom Anwender beschreiben, jedoch nur einmal (Analogie: PROM). Bei 5+-Zoll-Platten sind Speicherkapazitäten bis 1 GigaByte (Transferrate 1,5 MByte/s) möglich. Die WORM kann zur Archivierung von Daten aller Art verwendet werden (Backup-Medium). Die Platte arbeitet wie ein Magnetplattenlaufwerk und kann genauso angesprochen werden, die Treibersoftware sorgt dafür, daß bei mehrfacher Speicherung einer Datei immer die jüngste Version angesprochen wird (ältere Versionen lassen sich über spezielle Programme lesen) --> Speicherung einer Dateichronologie. Beim Schreiben wird durch hohe Laserenergie die Plattenstruktur dauerhaft verändert. Beim Lesen wird diese Veränderung mit niedriger Laserenergie abgetastet und detektiert. Man unterscheidet zwei Speichertechniken:

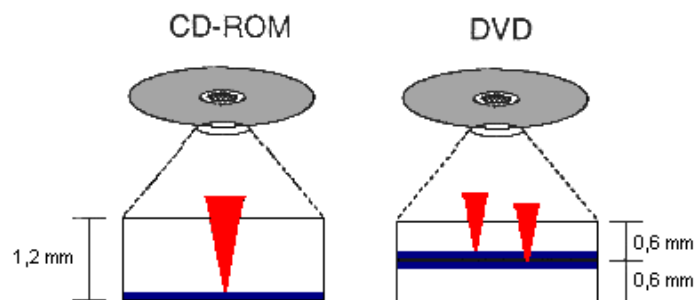
- Bei der Blasenherzeugung wird durch den Laserstrahl eine Polymerschicht erhitzt, die unter einem dünnen Metallfilm liegt. Es kommt zur Bildung einer Blase, die den Metallfilm dauerhaft verformt. Bei der Abtastung mit geringer Laserenergie kann die geänderte Streuung ausgewertet werden.
- Bei der Pit-Erzeugung durchbrennt der Laserstrahl eine lichtundurchlässige Schicht, die über einer Reflexionsschicht liegt (Pit entsteht). Beim Lesen werden die so entstandenen Hell-Dunkel-Zonen ausgewertet.

Die beschreibbare CD - CD-R

Bei der CD-R ist der Aufbau komplexer als bei der CD-ROM. Unten liegt die Trägerschicht aus Polycarbonat, darauf folgt eine lichtempfindliche organische Substanz, die durchscheinend ist. Dann kommt eine reflektierende Goldschicht und schließlich eine Lack-Schutzschicht. Mit erhöhter Laserenergie kann das organische Material verfärbt bzw. verschmolzen werden und es erhält so eine andere Reflexionseigenschaft. Die Platte kann danach wie eine CD-ROM gelesen werden.

DVD - Digital versatile Disc

DVD steht für 'Digital Versatile Disk' (ehemals 'Digital Video Disk'). Das Medium ist so groß wie eine normale CD-ROM, jedoch wird mit einer wesentlich höheren Speicherdichte gearbeitet. Dabei unterscheidet man vier verschiedene Medien. Die einfache einseitige DVD kann 4,7 GB auf einer Schicht speichern. Es gibt aber auch zweischichtige DVDs. Dabei wird die Information auf zwei übereinanderliegenden Schichten gespeichert, eine davon ist durchsichtig.



Durch unterschiedliche Fokussierung des Lasers wird die richtige Schicht angesteuert. Damit sind 8,5 GB möglich. Und dann gibt es das ganze noch zweiseitig. Damit sind 17 GB Daten auf einer einzigen DVD möglich. Die ersten Laufwerke kommen jetzt gerade auf den Markt und können einschichtige, einseitige DVDs lesen. Leider gibt es im Moment noch wenig DVD-Titel mit Videos. Die Videos werden in MPEG-2 kodiert, was eine sehr gute Qualität bei der Wiedergabe ergibt. Auch die ersten Brenngeräte für einseitige, einschichtige DVDs sind schon vorgestellt worden, der Brenner von Pioneer war im Herbst 97 für etwas über 10000 Mark auf den Markt gekommen.

Aufgenommen wird mit ca. 1 - 2 MB/s, und speichern kann er maximal 3,9 GB. Inzwischen sind die Preise auf dem Niveau von CD-ROM-Laufwerken.

Die Lesegeräte können auch normale CDs lesen, jedoch meist keine CD-Rs, also die beschreibbaren CDs. Dies kommt daher, daß ein Laser mit einer kürzeren Wellenlänge verwendet wird, der die selbstgebrannten CDs nicht mehr richtig lesen kann.

Die Zeit der DVD als "Nur-Lese-Medium" währte nur recht kurz. Seit der Jahrtausendwende gibt es auch DVD-Brenner. Viele Konsumenten werden vom Kauf eines DVD-Brenners allein deshalb abgehalten, weil sie nicht wissen, welcher der von den Firmengruppen propagierte Standard sich langfristig durchsetzen wird. Hier ein kleiner Überblick. DVD+R, DVD+RW, DVD-R, DVD-RW und DVD-RAM und sind allesamt Standards für widerbeschreibbare DVDs.

Die **DVD-R** ist ein einmal brennbarer Rohling für DVD, die DVD-RW ist ein wiederbeschreibbares DVD-Medium (ähnlich verwendbar wie die CD-RW). Diese Aufzeichnungsformate werden u. a. von Panasonic und Pioneer propagiert. DVD-R Medien können von vielen, aber nicht allen DVD-Laufwerken gelesen werden. Bei neuen DVD-Laufwerken ist die Wahrscheinlichkeit sehr hoch, nach Marktumfragen bei etwa 95%. Anfängliche Kompatibilitätsprobleme konnten inzwischen mit neuer Brenn-Software behoben werden. Im Zweifelsfall hilft nur probieren. DVD-R-Medien sind in *DVD-R for Authoring*: DVD-R(A) und *DVD-R for General*: DVD-R(G) aufgeteilt. DVD-R(A)-Medien werden bei der Produktion von DVD-Inhalten eingesetzt und dienen im Presswerk als Vervielfältigungsvorlage. DVD-R(G)-Medien sind für den privaten Gebrauch gedacht. Im Gegensatz zu DVD-R(A)-Medien lassen sich mit DVD-R(G)-Rohlingen keine 1:1-Kopien kopiergeschützter DVDs anfertigen. DVD-R(A) und DVD-R(G) sind beide kompatibel mit DVD-ROM Laufwerken und DVD-Playern. Aufgrund unterschiedlicher Wellenlängen des Schreiblasers (635 nm für DVD-R(A) und 650 nm für DVD-R(G)) können die Rohlinge *nur in den entsprechenden Recordern* beschrieben werden.

Sowohl im Aufbau der Medien als auch in der Struktur, wie die Daten auf die DVD gebrannt werden, können auch **DVD+R/DVD+RW**-Medien von nahezu allen Geräten (ebenfalls ca. 95%) verarbeitet werden. Das bedeutet, dass eine im Computer hergestellte DVD+RW auf nahezu jedem DVD-Player oder auch auf nahezu jedem Computer-DVD-ROM-Laufwerk abgespielt werden kann, nur nicht auf einem DVD-R- bzw. DVD-RW-Laufwerk (und umgekehrt). DVD+R (DVD+Recordable) ist als Bestandteil des DVD+RW-Standards definiert. Die DVD+R Medien sind wesentlich günstiger als DVD+RW Medien; der Anwender hat aber die gleichen Möglichkeiten, die er mit einer DVD+RW hat, abgesehen davon, dass sich das Medium nicht löschen lässt. DVD+R Medien können in DVD+RW-Geräten beschrieben werden. Die DVD+R hat die gleichen Charakteristika wie eine DVD+RW (abgesehen von der Möglichkeit des Löschsens und Neubeschreibens) wie auch den gleichen Grad an Kompatibilität in Bezug auf DVD-Player und DVD-ROM-Laufwerken, die gleiche Kapazität (4,7 GB) und wird über die gleichen Programme beschrieben.

Das Phase-Change-Aufzeichnungsverfahren findet bei DVD-RW wie auch DVD+RW Anwendung. Bei den Phase-Change-Medien besteht die Aufzeichnungsschicht aus vier Lagen: einer unteren dielektrischen Schicht, der Aufzeichnungsschicht (*recording layer/alloy*), der oberen dielektrischen Schicht und aus der Reflexionsschicht, welche meistens aus einer Aluminium- oder Messing-Legierung oder Gold besteht. Die Aufzeichnungsschicht wird mit einem Laserstrahl auf ungefähr 200 Grad Celsius erhitzt. Dadurch ordnen sich die Atome innerhalb der Metalllegierung kristallin an. Dieser Zustand wird durch langsames Abkühlen beibehalten, wodurch dieser Bereich einen hohen Reflexionsgrad besitzt. Er wird als "Land" oder logische 0 interpretiert. Durch Erhöhen der Leistungsstufe des Lasers erhitzt sich das Material auf 500 bis 700 Grad Celsius und die Atome geraten in einen amorphen, das heißt nichtkristallinen Zustand. Ein starker Wärmeentzug durch die beiden dielektrischen Schichten kühlt die Metallschicht schnell ab. Die Atome erstarren in ihrem ungeordneten Zustand und reflektieren nun weniger Licht.

DVD-RAM verwendet eine vollständig andere Speichertechnologie, die mit dem DVD-Standard in keiner Weise vereinbar ist. Die einzige Ähnlichkeit, die das Medium eventuell mit einer DVD besitzen könnte, ist seine ähnliche Kapazität - mehr nicht. Kein auf dem Markt befindlicher DVD-Player kann diese Medien lesen. DVD-RAM-Medien sind in einem Caddy untergebracht - und auch wenn man Medien ohne Caddy herstellen wird, so wird dies nicht von Erfolg gekrönt sein. Die Laufwerke und Medien sind also nur zur Datenspeicherung bzw. zum Datenaustausch von PC zu PC geeignet.

Aktuell werden CD-R und CD-RW von vielen Usern als preiswertes Medium für das Speichern von Daten und Audio verwendet. Daran wird sich auch die nächsten Jahre nicht viel ändern. DVD+RW-Geräte werden von Personen verwendet, die große Datenmengen sichern (PC-Anwender) oder aber auch für Videoanwendungen (Heimelektronik/PC). Mittel- bis langfristig werden wohl die DVD+RW-Geräte die CD-Brenner im Heim-PC ablösen, da die DVD+RW-Geräte auch in der Lage sind, CD-R- und auch CD-RW-Medien zu verarbeiten (nur schneller und zuverlässiger). Für ganz vorsichtige Anwender gibt es inzwischen Geräte, die alle erwähnten Formate (außer DVD-RAM) verarbeiten können (z. B. von SONY), man muss aber mit einem Mehrpreis von etwa 50% rechnen.

Magnetooptische Platte

Die magnetooptische Platten erlaubt - wie eine Magnetplatte - Lesen und Schreiben. Die Speicherkapazität reicht bis 1 Giga- Byte (Transferrate 1,5 MByte, mittl. Zugriffszeit ca. 100 ms). Im Gegensatz zu CD-ROM und WORM wird hier die Information nicht optisch, sondern magnetisch gespeichert. Das Lesen/Schreiben der Information erfolgt jedoch durch den Laser. Auf der Platte wird eine Terbium-Eisen-Kobalt-Legierung so aufgebracht, daß die Vorzugsachse der Magnetisierung senkrecht zur Plattenoberfläche steht.

Das magnetooptische Material hat bei Zimmertemperatur eine hohe Koerzitivität, bei hohen Temperaturen jedoch eine niedrige Koerzitivität. Deshalb kann ein kleiner Bereich der Platte, wenn er durch den scharf fokussierten Laserstrahl erhitzt wird, durch ein angelegtes Magnetfeld magnetisiert werden.

Zum Lesen wird der Kerr-Effekt genutzt. Danach dreht linear polarisiertes Licht bei der Reflexion an einen magnetischen Medium seine Polaritätsebene. Je nach Polung des Magnetfeldes im oder gegen den Uhrzeigersinn. Beim Lesen arbeitet der Laser mit verminderter Leistung, sodaß die Magnetisierung erhalten bleibt.

Die Magnetschicht besteht aus einer Kombination von Terbium (seltene Erden) und Eisen-Kobalt (Übergangsmetall). Die magnetischen Momente der Terbium-Atome sind entgegengesetzt zu denen der Übergangsmetall-Atome.

- Bei tiefen Temperaturen überwiegt die Magnetisierung der Terbium-Atome.
- Bei einer bestimmten mittleren Temperatur (Kompensations- Temperatur) beträgt die resultierende Magnetisierung null.
- Bei hohen Temperaturen überwiegt die Magnetisierung der Übergangsmetalle.
- Bei noch höheren Temperaturen wird durch die thermische Bewegung die Orientierung der magnetischen Momente ganz aufgehoben (Neel-Temperatur).

Beim Schreiben wird durch den Laserstrahl eine Stelle der Platte bis knapp unter die Neel-Temperatur aufgeheizt (geringe Koerzitivität). Es reicht nun ein relativ schwaches Magnetfeld aus, um die Magnetisierungsrichtung der Übergangsmetalle umzu- kehren. Nahe der Kompensationstemperatur (Zimmertemperatur) ist die Koerzitivität hoch und die Information bleibt erhalten. Der Bereich des Magnetfeldes kann also wesentlich größer sein als die durch den Laser erhitzte Stelle. Es ergibt sich so eine höhere Speicherdichte als bei der Magnetplatte.

Da es Schwierigkeiten macht, die Magnetisierung mit der für die hohen Datenraten erforderlichen Geschwindigkeit umzukehren, wird eine neu zu beschreibende Spur zunächst gelöscht (gleiche Ausrichtung aller Bereiche) und bei der folgenden Umdrehung neu beschrieben.

Inzwischen konnte gezeigt werden, daß auch in einem magnetooptischen Material direktes Überschreiben möglich ist. Der Trick besteht darin, das entmagnetisierende Feld (Terbium) der Beschichtung groß genug zu machen (Koerzitivität bei hoher Temperatur hinreichend niedrig). Die Magnetisierung eines Bereichs kehrt sich immer dann um, wenn der Bereich aufgeheizt wird. Ein äußeres Magnetfeld ist dann nicht mehr notwendig.

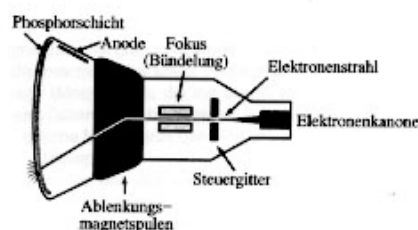
8.3 Periphere Geräte

8.3.1 Datensichtgerät und Tastatur

- Datensichtgerät (Terminal, DSG) im engeren Sinn: Bildschirmgerät zur Darstellung der vom Rechner ausgegebenen Informationen in alphanumerischer oder grafischer Form.
- DSG im weiteren Sinn: Wie oben einschließlich Tastatur zur Eingabe alphanumerischer Informationen → heute das verbreitetste Dialoggerät → Standard-Kommunikationsgerät bei DVS. Heutige Terminals sind i. a. mikroprozessorgesteuert; u.U. sogar durch mehrere Mikroprozessoren. Oft werden auch PCs als "Terminalersatz" verwendet. Verbindung zur CPU meist über serielle Schnittstelle
- Bei Personal Computern ist das DSG in die Zentraleinheit integriert. Im einfachsten Fall übernimmt die Haupt-CPU auch die Steuerung der Terminalfunktionen. Die DSG-Baugruppen sind direkt an den Systembus angeschlossen.

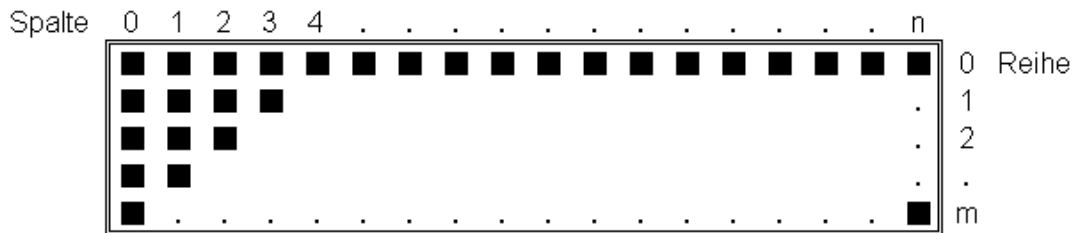
Die Darstellung der Information erfolgt auf einer Kathodenstrahlröhre nach dem gleichen Prinzip, wie bei einem Fernseh-Monitor. Einfache Systeme liefern sogar ein geeignetes Signal zur Ansteuerung von handelsüblichen Fernsehgeräten. Ein Elektronenstrahl wird periodisch über ein festgelegtes Zeilenraster geführt. Die darzustellenden Zeichen werden durch Modulation der Strahlintensität abgebildet → Raster-Abtast-Verfahren, Raster-Scan-Verfahren. Während des schnellen Strahlrücklaufs wird der Strahl dunkelgetastet. In Fernsehgeräten wird das Zeilensprungverfahren (interlace) angewendet. Die Darstellung erfolgt in zwei zeilenversetzten Halbbildern, d.h. es werden zunächst die Zeilen 1, 3, 5, 7, ... und danach die Zeilen 2, 4, 6, 8, ... auf den Bildschirm geschrieben. Die Bildwechselfrequenz beträgt dabei 50 Hz. Das Gesamtbild (625 Zeilen) wird dann mit 25 Hz wiederholt → flimmerfreie Darstellung bei niedriger Bildwechselfrequenz.

Prinzip der Kathodenstrahlröhre



DSG arbeiten in der Regel ohne Zeilensprung (non-interlaced), aber mit einer Bildwiederholrate von 50 .. 100 Hz und wesentlich höherer Rasterzeilenzahl (bis 4096 Zeilen). Um die Anforderungen an den Monitor bei sehr hohen Auflösungen niedriger zu halten, wird in manchen Fällen auch hier mit dem Zeilensprungverfahren gearbeitet. Jede Zeile wird in Bildpunkte zerlegt. Zur Darstellung von Grafik kann bei geeigneten DSG jeder Bildpunkt angesteuert werden. Zur alphanumerischen Darstellung wird jedes Zeichen als Punktraster in einer rechteckigen Matrix beschrieben und dargestellt. Es gibt unterschiedliche Matrixgrößen, z.B. 5 x 8, 8 x 8, 7 x 9, ... Die eigentliche

Darstellungsmatrix wird um eine Zeile/Spalte zur Trennung der einzelnen Zeichen ergänzt → Zeichenfeld ist größer, z.B. bei 5 x 8: 7 x 10. Alle nebeneinander dargestellten Zeichen bilden eine Reihe. Der Bildschirm wird in Zeichenfelder unterteilt:



Am häufigsten sind Geräte mit Textdarstellung von 24/25 Reihen zu 80 Spalten. Die Abbildung der Zeichen erfolgt (Raster-)zeilenweise für alle Zeichenfelder der einzelnen Reihen von oben nach unten.

Bei der Darstellung von Schwarzweiß-Grafik (Ansprechen jedes Bildpunkts möglich) muss jedem Bildpunkt ein Bit des Speichers zugeordnet werden (benötigte Speicherkapazität hoch). Bei der alphanumerischen Darstellung wird eine effektivere Form der Speicherung verwendet: Die einzelnen Zeichen werden codiert (z.B. in ASCII) im Bildwiederholpeicher gehalten. Die benötigte Speicherkapazität ist viel geringer (z.B. bei 25 x 80 Byte: 2 KByte). Die Umsetzung der Zeichencodes in die Matrixdarstellung erfolgt mittels eines Zeichengenerators (character generator). Dieser ist heute i. a. als Festwertspeicher (ROM, PROM, ...) realisiert.

Bei der farbigen Darstellung enthält die Bildröhre drei Elektronenstrahl-Kanonen, die eine, punktwise in den drei Grundfarben rot, grün und blau eingefärbte, Leuchtschicht des Bildschirms anregen. Zur "sauberen" Darstellung der Farben befindet sich innerhalb der Bildröhre ein Schlitz- oder Lochmaske. Bei der digitalen Ansteuerung ist die Zahl der darstellbaren Farben begrenzt (8 - 16 Farben). Bei der analogen Ansteuerung können praktisch beliebig viele Farben erzeugt werden. Je nach Auflösung ist die maximale Zahl der darstellbaren Farben begrenzt (m aus n Farben bei Grafik). Manche DSG bieten die Möglichkeit aus den zur Verfügung stehenden Farben eine Auswahl zu treffen. Dazu wird zwischen Bildwiederholpeicher und Analogausgang eine Tabelle für die Farbzuordnung geschaltet (Color Look Up Table, CLOUT).

Flache Bildschirme:

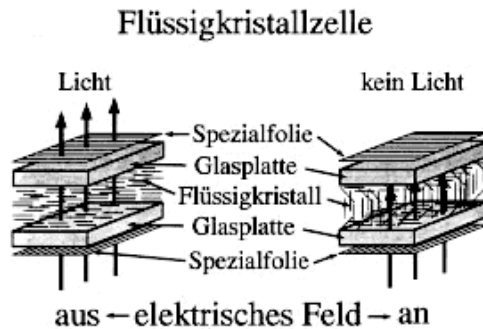
Von der Bauart her benötigen Kathodenstrahl-Monitore recht viel Platz. Für die Entwicklung transportabler Computer sind flache Bildschirme entwickelt worden:

- **Plasma-Anzeigen (Plasma-Display)**
Plasmaanzeigen bestehen im wesentlichen aus zwei ebenen, parallelen Glasplatten, deren Zwischenraum mit einem Edelgas (z.B. Neon) gefüllt ist. Auf die Glasplatten sind senkrecht zueinander linienförmige (durchsichtige) Elektroden aufgebracht, z.B. auf der Vorderseite lotrecht und auf der Rückseite waagrecht. An die Elektroden wird eine Spannung angelegt, die eine Glimmentladung am Kreuzungspunkt hervorruft. Da man durch die Elektroden hindurchschaut und wegen der Glimmentladung wirken Plasmaanzeigen etwas unscharf.
- **Flüssigkristall-Anzeigen (Liquid Crystal Display, LCD)**
Im Aufbau ähneln sie den Plasma-Anzeigen, jedoch befindet sich hier eine spezielle Flüssigkeit (nematische Phasen) zwischen den Glasplatten. Das angelegte elektrische Feld ändert die optische Eigenschaft der Moleküle (von reflektieren auf durchsichtig oder umgekehrt). Damit sich die Flüssigkeit nicht zersetzt, erfolgt eine Ansteuerung mit Wechselspannung. Durch eine Reflektorschicht oder Beleuchtung von der Rückseite des

LCD kann der Kontrast erhöht werden. Der Betrachtungswinkel ist bei LCDs relativ gering. Die Änderung der Anzeige ist - bedingt durch die chemischen Prozesse - langsam. <

- **Aktiv-Matrixanzeigen (TFT-LCD)**

LCD-Anzeigen sind relativ träge (ca. 14 ms Ansteuerperiode). Bei Aktiv-Matrix-LCDs erfolgt die Ansteuerung mit einem Feldeffekt-Transistor je Bildpunkt. Die FETs werden in Dünnschichttechnik (TFT = Thin Film Technologie) auf das Glassubstrat aufgebracht. Bei einem Farbdisplay bedeutet dies 3 FETs pro Bildpunkt → bei einer Auflösung von 640 x 480 Punkten: 921600 FETs. Durch TFT sind Kontrastwerte bis zu 100:1 und 2 ms Ansteuerperiode möglich.



Tastatur

Sie ist der Eingabeteil des DSG. Die Tastenanordnung ist ähnlich, wie bei der Schreibmaschine. Das Betätigen einer Taste erzeugt ein Codewort (im 1 aus m-Code) → Codierung des mit der Taste eingegebenen Zeichens erforderlich (meist ASCII, aber auch andere Codes (z.B. IBM-PC)). Die Tasten sind mehrfach belegt. Es gibt Tasten zur Umschaltung der Bedeutung, die gleichzeitig mit einer anderen Taste betätigt werden müssen:

- SHIFT-Taste (Kleinbuchstabe/Ziffern → Großbuchstabe/Sonderzeichen)
- CONTROL-Taste (Steuerzeichen)
- ALTERNATE-Taste (weitere Zeichenebene)

Für einige Steuerzeichen sind i.a. eigene Tasten vorhanden. Die meisten Tastaturen werden um Funktionstasten ergänzt, die z.T. spezielle Codeworte erzeugen. Manchmal wird die Tastatur um einen Ziffernblock ergänzt, mit dem eine schnelle Eingabe von Ziffern und Rechenzeichen ("+", "-", "*", "/") möglich ist --> Parallelschaltung mit den entsprechenden Tasten der Haupttastatur. Es gibt verschiedene Ausführungsformen von Tasten:

- mechanische Taste:
Gegen eine Federkraft wird ein elektrischer Kontakt betätigt
- Membran-Taste:
Elektrisch leitende, elastische Matte wird gegen den Kontakt gedrückt. Tastatur ist hermetisch abdichtbar gegen Umwelteinflüsse.
- Kapazitive Taste:
Ein Metallblatt an der Taste ändert bei Betätigung der Taste eine Kapazität.
- Hall-Effekt-Taste:
Ein an der Taste befestigter Permanentmagnet ändert den Widerstand einer Hall-Feldplatte → Erzeugen einer Spannung (Hall-Spannung)
- Piezo-Taste:
Eine Piezokeramik gibt bei Tastendruck eine Spannung ab.

Die Tasten sind üblicherweise in einer Matrix angeordnet, bei der durch das Drücken einer Taste eine Spalte mit einer Zeile verbunden wird. Aus der Zeilen- und Spaltennummer kann die Taste eindeutig erkannt werden und der zugeordnete Zeichencode ausgegeben werden. Es werden Zeile

und Spalte der gedruckten Taste ermittelt und daraus dann das zugehörige Codewort abgeleitet. Der Zeichencode ist oft in einer Tabelle (häufig ROM) abgelegt über das die Zeilen-/Spaltenzuordnung (Adressleitungen) das codierte Zeichen abgeleitet wird. Die Codierung kann erfolgen:

- per Hardware: Keyboard-Encoder-Baustein
- per Software: eigener Mikrocomputer für die Tastatur

8.3.2 Drucker

Mechanische Drucker:

Erzeugung eines Zeichens durch Anschlag eines Tinten- oder Carbonbandes auf das Papier; starke Geräuscentwicklung. Ausnahme: Tintenstrahldrucker – "Spritzen" von Tinte auf das Papier

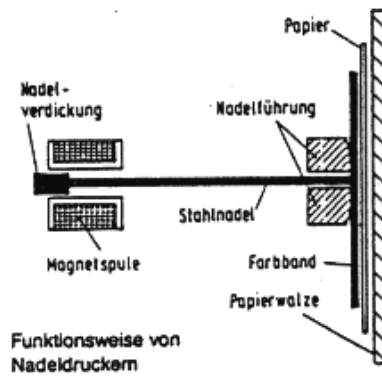
- Vollzeichendrucker: Anschlag eines Zeichens als Ganzes
- Matrixdrucker: Zusammensetzen eines Zeichens durch Einzelpunkte
- Serieller Drucker: Die Zeichen werden nacheinander gedruckt
- Zeilendrucker: Eine ganze Zeile wird auf einmal gedruckt

Nichtmechanische Drucker:

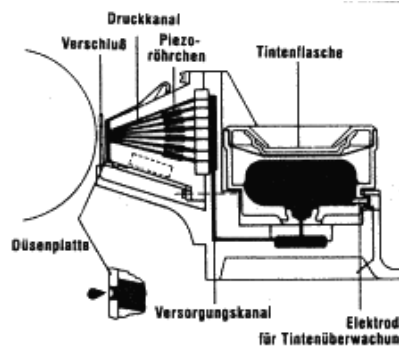
- Thermodrucker: Spezialpapier, das auf Hitze reagiert
- Laserdrucker: Übertragung durch Elektrostatisches Feld
- Elektrochemische Drucker

Druckerarten:

- Typenhebelldrucker: Wie bei der Schreibmaschine schlägt ein Typenhebel gegen Farbband und Papier (Fernschreiber). Max. 10-15 Zeichen/s.
- Kugelkopfdruker, Typenzylinderdrucker Die Drucktypen sind auf dem Kugelkopf (Zylinder) angeordnet, der durch Drehbewegung in Position gebracht wird. Ein Hammer schlägt den Kopf/Zylinder auf Farbband und Papier. Max. 20 Zeichen/s.
- Typenkorbdrucker: Die Drucktypen sind kreisförmig in Korbform angeordnet, der durch Drehbewegung in Position gebracht wird. Ein Hammer schlägt den Kopf/Zylinder auf Farbband und Papier. Max. 40-60 Zeichen/s.
- Typenraddruker (Daisywheel-Printer): Die Drucktypen sitzen strahlenförmig an einem drehbaren Typenrad, Ein Hammer schlägt die Type gegen Farbband und Papier. Max. 60 Zeichen/s.
- Nadeldruker (Matrixdrucker): Der Druckkopf besteht aus mehreren senkrecht übereinander angeordneten Nadeln (8, 9, 24), die einzeln durch Elektromagnete gegen Farbband und Papier geschlagen werden. Die Zeichen werden aus mehreren Spalten zusammengesetzt (wie beim DSG). Max. 600 Zeichen/s - laut. Zur Verbesserung der Druckqualität setzt man Köpfe ein, bei denen zwei Nadelreihen gegeneinander versetzt sind. Ihre Druckbilder liegen dann übereinander (Punkte überlappen).



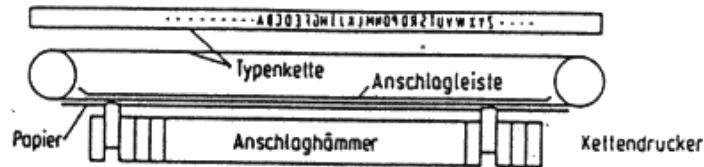
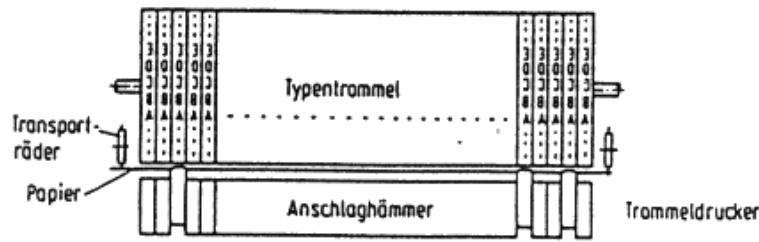
- **Thermodrucker (Matrixdrucker):** Ein wärmeempfindliches Papier wird punktwise von einem Druckkopf erhitzt und so geschwärzt. Es gibt Druckköpfe für Rasterpalten (wie Nadeldrucker), ganze Zeichenmatrix und ganze Punktzeilen (Thermokamm). Die selektiv erhitzbaren Punkte werden durch Widerstände gebildet. Max. 30-50 Zeichen/S - leise.
- **Tintenstrahldrucker:** Es werden gezielt Tröpfchen einer schnell trocknenden Spezialtinte aufs Papier gespritzt. Es gibt verschiedene Verfahren:
 - ♦ **Hoch- und Niederdruckverfahren:** Der Tintenstrahl aus der Düse wird durch ein elektrisches Feld in zwei Dimensionen abgelenkt. Erzeugen des Drucks z.B. durch Erhitzen. Bei einfacheren Matrix-Typen wird die Tinte durch Erhitzen eines integrierten Widerstandes im Tintenkanal herausgedrückt (Dampfblase erzeugt einen definierten Tropfen).
 - ♦ **Unterdruckverfahren:** Die Düsen stehen senkrecht übereinander; der Düsenkanal wird von einem piezoelektrischen Röhrchen umschlossen. Durch Unterdruck in den Düsen wird die Tinte in den Öffnungen etwas nach hinten gesaugt (Selbstreinigung). Zum Drucken wird kurzzeitig ein elektrisches Feld an das Piezo-Röhrchen gelegt, das sich zusammenzieht (Radialschwinger) → Druckerhöhung → Tintentröpfchen wird ausgestoßen. Max. 300 Zeichen/s - leise, nahezu verschleißfrei.



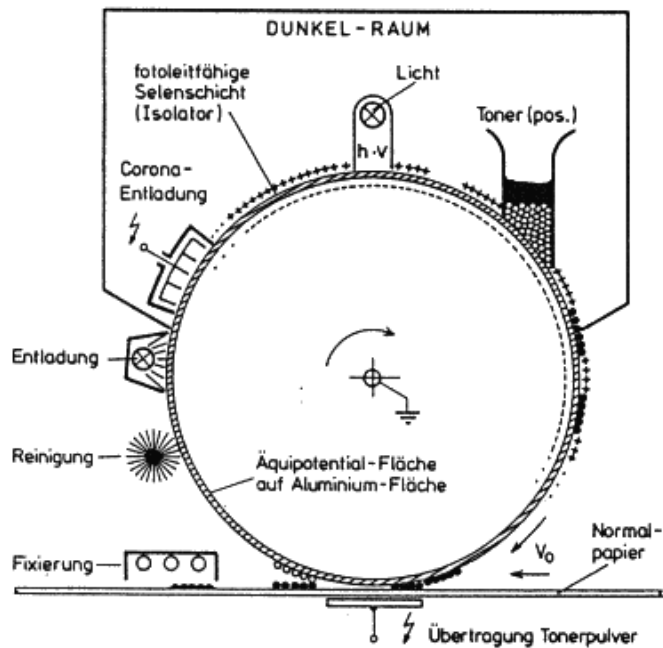
Aufbau des Tintendruckkopfes

- **Kammdrucker (Schnelldrucker):** Ein Druckkamm mit z.B. 132 federnden Kammsegmenten, von denen jedes eine Stahlkugel (für einen Rasterpunkt) trägt. Die Segmente werden von Elektromagneten in gespannter Lage gehalten und zum Drucken freigegeben. Der Kamm kann waagrecht um eine Zeichenbreite bewegt werden, sodass jedes Segment ein vollständiges Rasterzeichen druckt. Ca. 80 Zeilen/s.
- **Trommeldrucker (Schnelldrucker):** Auf der Oberfläche einer Typentrommel befinden sich an jeder Schreibstelle sämtliche druckbare Zeichen, also eine Zeile mit lauter "A", dann eine Zeile mit "B" usw. Die Trommel dreht sich mit konstanter Geschwindigkeit. Die Hämmer schlagen Farbband und Papier zum richtigen Zeitpunkt gegen die Trommel. Je Trommelumdrehung kann eine vollständige Zeile gedruckt werden. Ca. 30-40 Zeilen/s.
- **Kettendrucker (Schnelldrucker):** Eine Kette mit den Drucktypen läuft mit konstanter Geschwindigkeit vor dem Papier vorbei. Wenn das zu druckende Zeichen an der Schreibposition vorbeikommt, schlägt der entsprechende Hammer Papier und Farbband an. Meist ist der Zeichensatz mehrfach auf der Kette vorhanden.. Ca. 30-40 Zeilen/s.

Datenverarbeitungssysteme



- **Elektrostatische Drucker:** Es wird ein Spezialpapier verwendet, das mit einem dünnen Dielektrikum (einige μ) beschichtet ist $\rightarrow$ unempfindlich gegen elektrische Felder. Der Schreibkopf besteht aus übereinander angeordneten Schreibspitzen, die mit Hilfe einer Gegenelektrode ein elektrisches Feld erzeugen. Auf dem Papier entsteht ein latentes Ladungsbild. Das Papier durchläuft anschließend ein Tonersystem. Hier wird ein sehr feines Tonerpulver durch die Ladungen auf dem Papier angezogen. In der nachfolgenden Fixierstation wird das Tonerpulver durch Erwärmen fest mit dem Papier verbunden. Ca. 300 Zeilen/s.
- **Elektrochemische Drucker:** Das Papier ist mit einer Chemikalie getränkt. Der Schreibkopf besteht aus übereinander angeordneten Schreibspitzen, die eine lokale Änderung des pH-Wertes hervorruft, wodurch sich das Spezialpapier verfärbt. Ca. 100 Zeilen/s.
- **Laser-Drucker (Elektrofotografische/Xerographische Drucker):** Eine rotierende Trommel trägt eine foto-leitfähige Schicht (früher Selen, jetzt organische Materialien. Neueste Entwicklung: polykristallines Silizium mit einer Standzeit von ca. 300'000 Seiten). Mittels einer Ladecorona wird die Trommeloberfläche positiv geladen. Die Schicht verhält sich im Dunkeln wie ein Isolator, bei Beleuchtung wie ein Halbleiter. Durch Belichtung mit einem Laser-Ablenksystem wird ein latentes Bild punktwise auf der Trommeloberfläche erzeugt. Das Papier durchläuft anschließend ein Tonersystem. Hier wird ein sehr feines Tonerpulver durch die Ladungen auf die Trommel gebracht und anschließend von dort auf das Papier übertragen. In der nachfolgenden Fixierstation wird das Tonerpulver durch Erwärmen fest mit dem Papier verbunden. Ca. 400 Zeilen/s. In der Reinigungsstation wird der restliche Toner entfernt und die Trommel vollständig entladen.
 - ♦ **Laserdrucker sind Seitendrucker:** Die Druckinformation für eine Druckseite wird zunächst im Speicher des Druckers gepuffert und dann die Seite komplett erzeugt und ausgegeben. Laserdrucker sind voll grafikfähig. Inzwischen sind Laserdrucker mit einer Auflösung von 300 Dots per Inch und einer Druckleistung von 5 - 10 Seiten/min für knapp 3000 Mark erhältlich.

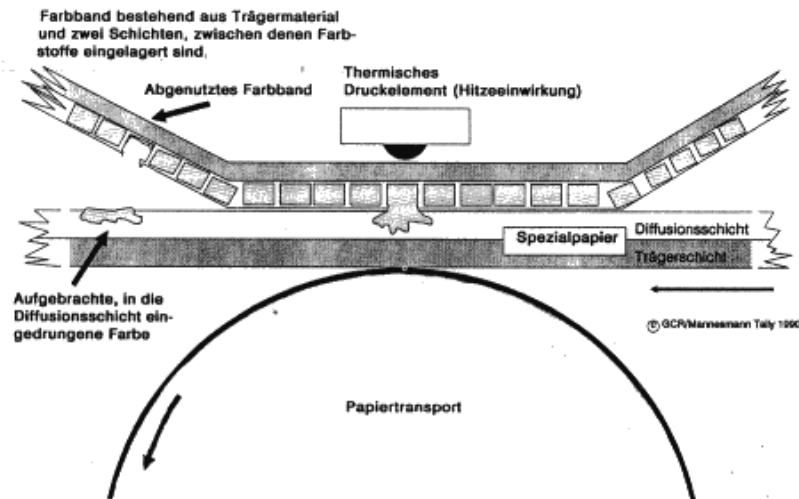


- ♦ LED- und LCS-Drucker: Bei diesen Druckern wird das gleiche Arbeitsprinzip, wie bei Laserdruckern verwendet. Die Belichtung erfolgt jedoch über LED-Zeilen (2432 / 3456 Leuchtdioden) oder einen Liquid Crystal Shutter, der das Licht einer starken Leuchtstoffröhre punktwise freigibt. Preis unter 2000 Mark.
- Ionendrucker: Sie erreichen eine Druckleistung von bis zu 80 Seiten/min. Als Belichtungseinheit für die Trommel (Prinzip, wie Laserdrucker) dient eine zeilenförmige Ionenkanone, die durch ein Lochraster die Punktladungen erzeugt.

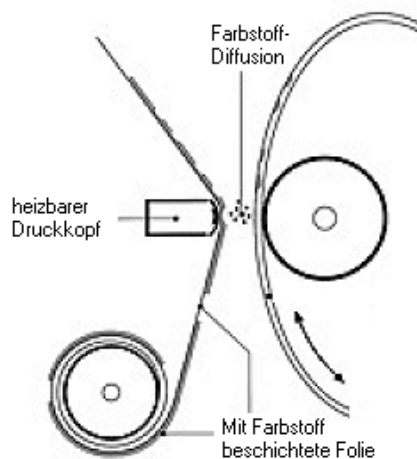
Farbdrucker

Um farbig zu drucken wird additive Farbmischung verwendet, d. h. die zu druckende Farbe wird aus den Grundfarben blau, rot, gelb zusammengesetzt. Da bei drei Farben schwarz gedruckte Partien eher bräunlich wirken, wird oft als vierte Farbe schwarz hinzugenommen. Es werden i. a. Matrixverfahren mit unterschiedlichen Technologien verwendet:

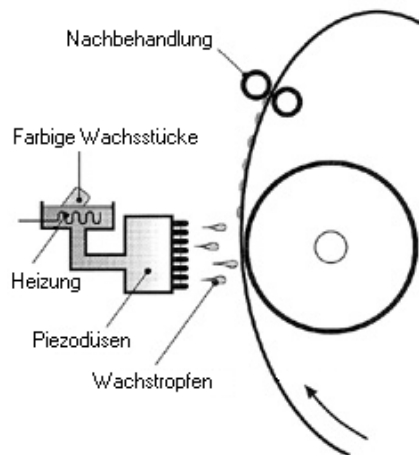
- Nadeldrucker mit breiterem Farbband, das 3 oder 4 Farbfelder enthält. Durch die Positionierung des Farbbandes wird die jeweilige Farbe gedruckt. Mischfarben entstehen durch mehrfaches bedrucken einer Zeile.
- Thermotransferdrucker: Hier liegt eine hitzeempfindliche, wachsartige Farbschicht auf einer Trägerfolie zwischen Druckkopf und Papier. Durch die Wärmeeinwirkung wird die Farbe auf das Papier übertragen. Häufig wird ein Thermokamm verwendet, der eine komplette Rasterzeile druckt. Farbmischung durch mehrfaches überdrucken.



- Thermosublimationsdrucker arbeiten ähnlich wie Thermotransferdrucker, es wird jedoch die Druckfarbe durch die Wärme des Druckkopfes verdampft und sie schlägt sich dann auf dem Papier nieder. Durch die Heizdauer kann die Farbsättigung gesteuert werden → Photoqualität mit nahezu beliebig vielen Zwischentönen. Dieser Vorgang muss nacheinander für vier Farben wiederholt werden. Die Verbrauchsmaterialien sind recht teuer (Zweiblattprozess, Farbband und Empfangsschicht). Die Bilder sind mechanisch sehr robust, da die Farbstoffe in die Empfangsschicht eingedrungen sind. Die Feuchte- und Hitzebeständigkeit ist aber auch bei den besten Vertretern noch geringer als die der Farbfotografie.

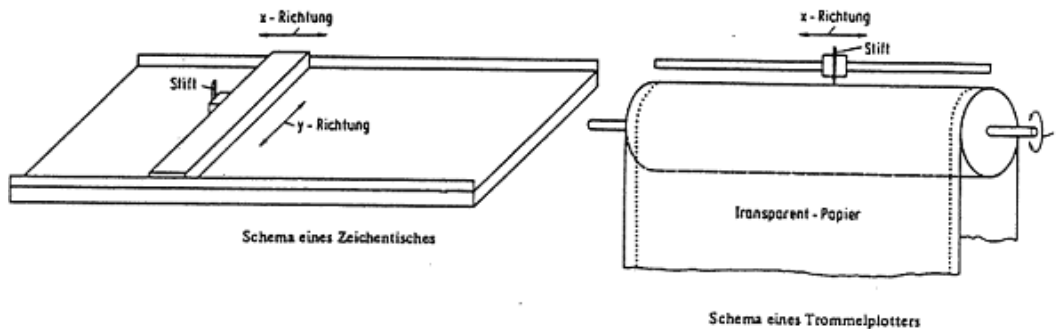


- Tintenstrahlprinter mit Wachstinten: Diese Technik wird hier separat behandelt, da sie einige wesentliche Eigenschaften hat, die sie von Flüssigtinten-Tintenstrahl unterscheidet. Die Druckköpfe sind ähnlich aufgebaut, aber die Tinten sind bei Raumtemperatur fest, d.h. Wachse. Sie werden zuerst geschmolzen. Das heiße Tröpfchen wird meist durch Verformung der Druckerdüse (Piezomechanismus) ausgestossen. Beim Auftreffen auf den Träger wird es schlagartig fest. Da kein Wasser auf dem Träger trocknen muss, kann schneller als mit Flüssigtinten gedruckt werden. Wachstinten sind vom Trägermaterial unabhängiger als Flüssigtinten. Sie dringen kaum in das Medium ein. Ähnlich wie beim Thermotransferdruck liegen die Farbpunkte daher auf der Oberfläche, sie sind jedoch häufig viel kleiner und liefern eine bessere Bildqualität, die im besten Fall an die der Flüssigtintendrucker heranreicht. Mit einer Drucknachbehandlung werden die Wachspunkte zusätzlich fixiert.



8.3.3 Plotter

Plotter sind elektromechanische Zeichengeräte, bei denen eine Zeichnung mit einem Zeichenstift auf Papier erstellt wird. Die Auflösung erreicht hier Schritte von 0.1 mm bis 0.01 mm. Trommelplotter, Flachbettplotter und Schrägbettplotter sind die wohl bekanntesten Bauarten.



Der Trommelplotter kommt in seinem Aufbau dem Drucker am nächsten. Das Zeichenpapier wird hier ebenfalls über eine Trommel oder Walze gelegt, wobei durch Drehen der Walze - vorwärts oder rückwärts - eine der beiden Bewegungsrichtungen dargestellt wird. Die andere Bewegungsrichtung - also nach rechts oder links - wird mit einem Schlitten ausgeführt, der von einem Schrittmotor bewegt wird. Die Halterung, die den Zeichenstift aufnimmt, befindet sich am Schlitten. Der Trommelplotter beansprucht - im Vergleich zu Flachbett- oder Schrägbettplottern - eine geringere Stellfläche. Der Vorteil des Trommelplotters besteht darin, dass nicht nach jeder Zeichnung das Papier neu eingelegt werden muss, da sich das Plotterpapier von einer großen Vorratsrolle zu einer Aufspulvorrichtung bewegt. Eine Abart stellen Reibungsplotter dar, bei denen das Papier durch eine mit Quarzsand beschichtete Rolle unter dem Stift hin- und herbewegt wird.

Beim Flachbettplotter sind auf einer Grundplatte oben rechts zwei Schrittmotoren angeordnet. Der X-Schrittmotor bewegt über ein Zugseil den X-Schlitten nach rechts oder nach links. Der Schlitten sitzt auf einer stationären, festen Schlittenführung. Der Y-Schrittmotor dreht eine meist mehrkantige Antriebswelle, auf der verschieblich ein Rad angebracht ist. An diesem Rad ist das Zugseil für den Y-Schlitten befestigt, so dass der Y-Schlitten durch Drehen der Antriebswelle hin- und her bewegt werden kann. Da der Y-Schlitten auf einer Schlittenführung läuft, die am X-Schlitten befestigt ist, kann der daran befestigte Stift in X- und Y-Richtung bewegt werden.

Unter dem Begriff "Zeichenstift" sind die unterschiedlichsten Zeichenwerkzeuge zu verstehen: Kugelschreiber und Tintenkugelschreiber, Gasdruckkugelschreiber, Faserschreiber, Keramik- und Faserstifte, Tuschestifte und Tuschespitzen, Bleistifte, Schneidwerkzeuge für Folien, ... Die meisten Plotter besitzen eine Aufnahmevorrichtung für mehrere Stifte und erlauben daher mehrfarbige

Zeichnungen. Ein Plotter kann von sich aus nur in 8 Richtungen zeichnen → Elementarschritte. Das Charakteristikum des Schrittmotorantriebs liegt im quantenweisen Betrieb des Bewegungssystems. Jede zu zeichnende Kurve muss durch diese acht möglichen Grundschriffe approximiert werden. Um eine hinreichende Genauigkeit zu erreichen, muss ein Elementarschritt möglichst klein sein (typisch 0.1 ... 0.025 mm). Eine Verbesserung würde sich durch die Verwendung von Servomotoren ergeben, die jede beliebige Bewegungsrichtung zulassen. Hier ist jedoch ein Meßsystem zur Positionsbestimmung nötig.

Neben der Auflösung ist die Wiederholgenauigkeit ein wichtiges Leistungskriterium bei Plottern. Sie liefert eine Aussage darüber, wie genau ein bestimmter Punkt beim mehrfachen Anfahren derselben Koordinaten getroffen wird (wichtig beim Ansetzen nach einem Stiftwechsel). Ein weiteres Kriterium stellt die Maximalgröße der Zeichenfläche (bei Flachbettplottern) bzw. die maximale Rollenbreite (bei Trommelplothern) dar. Da das Zeichnen relativ langsam erfolgt (relativ große bewegte Massen) besitzen neuere Plottersysteme (Graphtec) ein eingebautes 3½-Zoll-Diskettenlaufwerk zur Pufferung der vom Rechner eingehenden Daten. Die Zeichnung wird dann quasi offline erstellt. Zur Wiederholung derselben Zeichnung ist keine Neuberechnung am Computer mehr nötig.

8.3.4 Lochkarten- und Lochstreifengeräte

Lochkarten und Lochstreifen waren früher die gebräuchlichsten Datenträger bei DVS. Sie wurden zunehmend von anderen Datenträgern verdrängt. Die Lochkarte war einer der ersten Datenträger zur Speicherung von Daten und Programmen auf DVS. In den Anfängen der Datenverarbeitung gab es neben Lochkartenstanzern und Lochkartenlesern auch Geräte zum Mischen und Sortieren von Karten. Die Lochkarte ist in 80 Spalten zu 12 Zeilen eingeteilt. Die Karten, die gelesen oder gestanzt werden sollen befinden sich in einem Magazin. Die unterste Karte wird durch eine Friktionswalze oder einen Schieber mit Nase der Lesestation zugeführt.

Die Lochstreifengeräte in der Datentechnik haben sich aus der Fernschreibtechnik entwickelt. Dort werden 5-Kanal-Lochstreifen mit einer zusätzlichen Transportlochung verwendet. Aus der Transportlochung kann auch ein Synchronisationssignal für die Abtastung gewonnen werden. Bei Geräten der Datenverarbeitung verwendet man 6-, 7- oder 8-Kanal-Lochstreifen. Der Transport des Lochstreifens erfolgt bei langsamen Geräten (bis 50 Zeichen/s) durch ein Zahnrad, das in die Transportlochung greift. Bei schnellen Geräten (bis 1000 Zeichen/s) wird der Streifen durch Friktionwalzen transportiert. Abtastverfahren siehe Umdruck.

8.3.5 Joysticks (grafische Eingabe)

Ein Joystick besteht aus einem kleinen Kästchen, aus dem ein Betätigungshebel hervorschaut, der sich nach vier Richtungen bewegen lässt. Man kann durch Bewegen des Steuerknüppels z.B. ein Fadenkreuz auf dem Bildschirm bewegen, um Koordinaten für die Zeichnung festzulegen. Es gibt zwei technisch unterschiedliche Arten von Joysticks. Bei der einen Art wird der Knüppel von einem Kugelgelenk gehalten und schließt einen von vier Schaltern, je nach dem, ob man ihn nach oben, unten, rechts oder links bewegt. Bei diagonalem Druck auf den Knüppel werden dann die zwei entsprechenden Schalter betätigt. Mit solchen Joysticks lässt sich nur eine Richtungsinformation geben. Soll das Fadenkreuz durch die Auslenkung des Knüppels absolut positioniert werden, braucht man einen Joystick, bei dem Potentiometer eine der Knüppelauslenkung proportionale Spannung abgreift (Proportionalsteuerung). Die Spannungsmessung erfordert einen gewissen Hardware- und Softwareaufwand. Für die Joystick-Abfrage bieten sich zwei Prinzipien an:

- das Messen der Impulsdauer eines Monoflops (Potentiometer steuert Impulsdauer)
- Messen der abgegriffenen Spannung mittels eines Analog-Digital-Wandlers.

8.3.6 Trackball und Maus

Ein Trackball (Rollkugel), ist ein Gehäuse, aus dem oben das Viertel einer Kugel von 5 cm bis 10 cm Durchmesser herausragt. Im Gehäuse sitzen zwei Abnehmer-Räder im Winkel von 90 Grad zueinander, die beide die Kugel berühren. Jede Bewegung der Kugel wird so in X- und Y-Impulse umgesetzt. Durch Drehen der Kugel mit der Handfläche oder den Fingerspitzen kann ein Fadenkreuz auf dem Bildschirm positioniert werden. Es wird auch hier dem Computer nicht nur eine Richtungsinformation sondern auch eine Bewegungsgeschwindigkeit mitgeteilt. Man kann das gewünschte Ziel mit hoher Geschwindigkeit anfahren und dann vorsichtig fein positionieren.

Die Maus wurde schon 1961 erfunden. Sie ist nichts anderes als ein umgedrehter Trackball. Um die Maus verwenden zu können, braucht man ein Stück freie Schreibfläche, auf der die Maus "herumgefahren" wird. Es gibt auch Mäuse ohne Mechanik, die per Fotozelle eine spezielle Unterlage abtasten ("optische" Maus).

Trackball und Maus sind Eingabegeräte für relative Positionierung des Cursors oder Fadenkreuzes. Meist sind noch ein oder mehrere Tasten auf dem Gehäuse, mit denen, abhängig von der Software, gewisse Funktionen ausgelöst werden.

8.3.7 Digitalisierer

Digitalisierer (Digitalisiertablets, Grafiktablets) erlauben die direkte Eingabe einer zweidimensionalen Positionsinformation in den Computer. Auf einer vorgegeben Fläche wird ein "Stift" auf eine bestimmte Stelle gesetzt, worauf der Computer die Koordinaten des Punktes direkt übernimmt. Zur Bestimmung der Koordinaten werden verschiedene Methoden verwendet. Eine recht preiswerte Sorte Digitalisierer (zum Beispiel Koala-Pad), verwendet eine homogene Widerstandsschicht. Hier erfolgt die Messung in zwei Schritten:

1. Zuerst wird die Widerstandsschicht in horizontaler Richtung beschaltet und die X-Position des Stiftes gemessen
2. Danach die Y-Position durch Beschaltung in vertikaler Richtung

Beim kapazitiven Digitalisierer befindet sich im Tablett ein feines Netz von vertikalen und horizontalen Leitern, die phasenverschoben angesteuert werden. Der aufgesetzte Stift detektiert das Feld über die sich ergebenden Koppelkapazitäten zwischen Leitern und Stift. Durch die Auswertung benachbarter Leitungen kann der Abstand der Leiter wesentlich größer, als die Auflösung sein.

Elektromagnetische Digitalisierer verwenden ebenfalls ein feines Netz von vertikalen und horizontalen Drähten. Eine Interface-Schaltung stellt die Positionen eines Stiftes mit einer Aufnehmerspule fest. Das Arbeitsprinzip ist das gleiche, wie beim kapazitiven Digitalisierer.

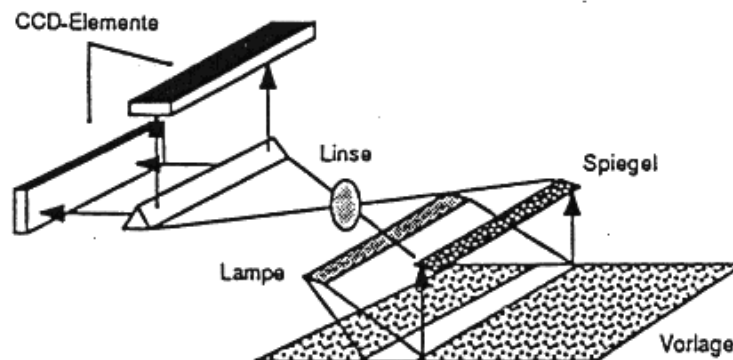
Nach einem ähnlichen Verfahren arbeiten die magnetorestriktiven Digitalisierer. Hier besteht das Netz aus ferromagnetischen Stahldrähten. Solche Stoffe besitzen die Eigenschaft, ihre Form unter magnetischen Einfluss geringfügig zu verändern. Eine Sendespule erzeugt im Draht eine magnetorestriktive Welle. Diese Längenänderung pflanzt sich als mechanische Spannungswelle (Geschwindigkeit ca. 5000 m/s) längs des Drahtes fort. Diese Welle induziert in der Aufnehmerspule eine Spannung. Aus der Laufzeit kann die Position des Stiftes ermittelt werden. Die mechanischen Anforderungen sind gering: die Drähte müssen nicht exakt parallel laufen und können einen Abstand von 2-3mm besitzen.

8.3.8 Scanner (Abtaster)

Scanner tasten eine beliebige Vorlage punktweise ab. Die Auflösung ist i. a. einstellbar. Es gibt verschiedene Abtastverfahren:

- feststehende Abtastvorrichtung, die Vorlage wird in Abtastrichtung bewegt
- feststehende Vorlage, der Abtastpunkt bewegt sich über die Vorlage (flying spot)
- Abtastung mit bewegter Blende

Bei den ersten Abtastern wurde die Vorlage auf eine Trommel aufgespannt, die mit konstanter Geschwindigkeit rotiert. Eine Fotozelle wird in Richtung der Rotationsachse entlang der Vorlage bewegt. Bei jeder Umdrehung wird eine Zeile der Vorlage abgetastet.



Bei modernen Scannern wird die Vorlage über ein optisches System auf eine Photodiodezeile oder einen CCD-Sensor (CCD = Charge Coupled Device) abgebildet, der das eingeleseene Punktraster-Bild an das DVS weiterleitet. Anstelle eines zweidimensionalen CCD-Sensors wird aus Kostengründen meist ein Liniensensor verwendet und die Vorlage zeilenweise abgetastet. Bei manchen Modellen ist der Sensor über der Vorlage angeordnet.

Die Detailauflösung eines Scanner wird angegeben in dpi (dots per inch, Bildpunkte pro Zoll)
Typischer Wert: 300 dpi 1 Zoll = 2,54 cm

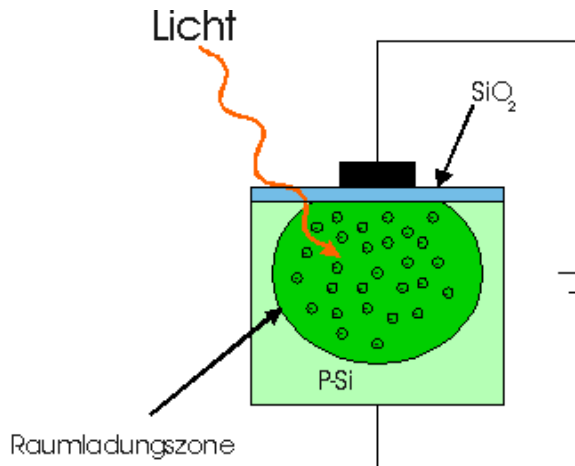
Eine weitere Möglichkeit der Abtastung ist die Digitalisierung eines Videobildes mit einem sogenannten 'Frame-Grabber'. Das Fernsehbild kann über eine relativ einfache Schaltung in einzelne Bildpunkte zerlegt werden:

- Zeilen- und Bildsynchronsignale stehen bereits zur Verfügung; die einzelnen Zeilen eines Bildes lassen sich einfach detektieren.
- Eine Bildzeile wird in kontinuierlichen Zeitabständen abgetastet und der entsprechende Grauwert über einen Analog-Digital-Wandler in ein Datenwort umgewandelt.

Sensortechnik

Der Sensor ist das Herzstück eines jeden Scanners und einer jeden Digitalkamera. Er ist in erster Linie für die Qualität der aufgenommenen Bilddaten verantwortlich. Die Bauart des Sensors und seine Eigenschaften entscheiden auch hauptsächlich über die Eignung für eine bestimmte Anwendung. Im folgenden soll einzig der Aufbau und die Funktionsweise von CCD-Sensoren (Charge Coupled Devices = "Ladungssammelnde Bauteile") beschrieben werden. Diese Sensorbauart hat sich in den Videokameras für industrielle Bildverarbeitung durchgesetzt. CMOS-Sensoren und Sensoren für Wärmebildkameras werden hier nicht betrachtet. Diese unterscheiden sich ohnehin in erster Linie nur durch die Methode, mit denen sie aus Licht oder

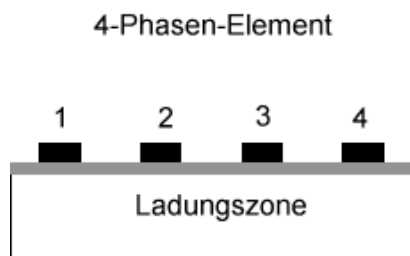
Wärmestrahlung Ladungen erzeugen. Der Ladungstransport ist ähnlich oder gleich wie bei CCDs.



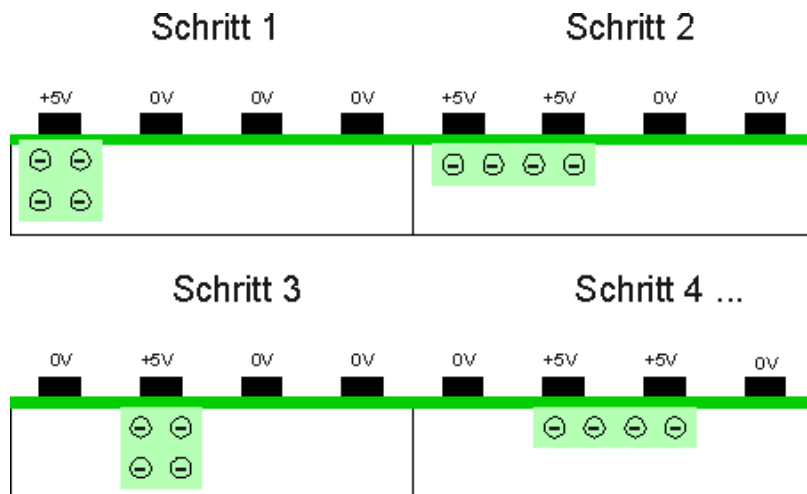
Ein CCD-Sensor besteht aus einem geometrisch sehr exakten Raster von lichtempfindlichen Zellen (Pixeln). Bei Lichteinfall (Photonen) auf einer dieser Zellen, wird eine Ladung (Elektronen) in dieser Zelle aufgebaut. Je mehr Licht oder je länger Licht auf die Zelle fällt, desto größer wird die Ladung (Zahl der Elektronen), die sich in der Zelle sammelt. Hier wird auch deutlich, daß die weitverbreitete Meinung falsch ist, die Bildinformation läge auf dem Sensor bereits in digitaler Form vor. Die Information wird durch die Anzahl der Elektronen in den Zellen repräsentiert und ist daher analog.

Die einzelnen Pixel berühren sich nicht direkt, sondern sind je nach Sensortyp voneinander durch Stege oder Potentialwälle getrennt. So wird einerseits verhindert, daß die Ladungsträger (Elektronen) von der einen Zelle in die andere überlaufen, zum andern sind diese Stege auch für das Auslesen des Zelleninhalts von Bedeutung. Folglich füllen je nach Sensortyp die lichtempfindlichen Zellen nicht den ganzen Sensor, sondern ein Teil der Sensorfläche wird als Transport- bzw. Sperrflächen genutzt. Die von lichtempfindlichen Elementen bedeckte Fläche bezeichnet man auch als *Fillfaktor*. Ein Fillfaktor von 100% würde bedeuten, daß die ganze Fläche lichtempfindlich wäre. In der Realität ist der Fillfaktor immer kleiner als 100%, da die lichtunempfindlichen Sperr- und Transportmechanismen Platz beanspruchen. Nach der Belichtung der einzelnen Zellen werden die Ladungen ausgelesen. Dieser Vorgang erfolgt bei allen Sensortypen nach dem sogenannten Eimerkettenprinzip.

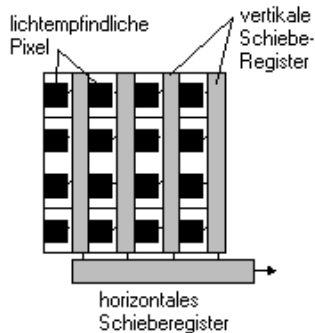
Die Ladungen aus den Pixelelementen werden über Schieberegister ausgelesen. Im Schieberegister wird der Inhalt einer Zelle einer benachbarten Zelle übergeben. Dies kann durch verschiedene Schaltungen erreicht werden, welche die Barrieren zwischen zwei Zellen (Eimern) auf- oder abbauen. Um die Ladungen aus den Pixeln gerichtet zu transportieren, wird im allgemeinen ein Verfahren verwendet, das auf einem 2-, 3- oder 4-Phasen-Schieberegister aufbaut. Am Beispiel eines 4-Phasen-Schieberegisters soll hier das Verfahren beschrieben werden. Ein Schieberegister ist aus einzelnen 4-Phasen-Elementen aufgebaut.



Jede der vier Zellen eines solchen Elements kann einzeln angesteuert werden. Durch eine geeignete Ansteuerung (4-Phasen-Clocking) kann erreicht werden, daß die Ladungen in dem Schieberegister gerichtet bewegt werden. Der Transport wird durch Auf- und Abbau von Ladungsschwellen und -Senken bewirkt.

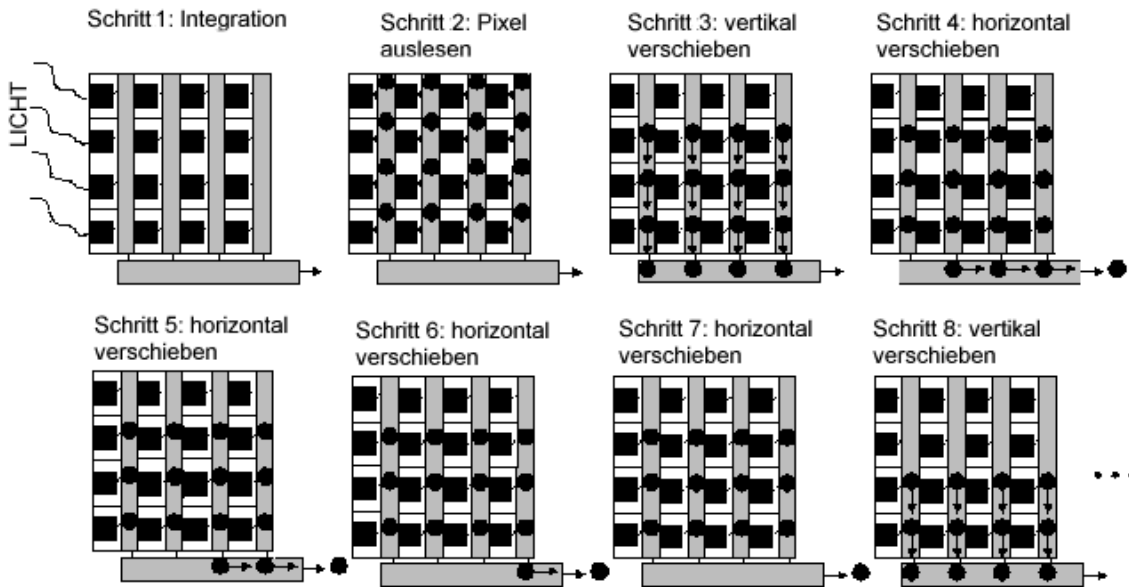


Mit Hilfe dieses Schieberegisters kann der CCD-Sensor ausgelesen werden. Alle gängigen Flächensensoren arbeiten nach dem gleichen Prinzip. Die Methode, mit der die Daten aus dem Sensor geschoben werden ist einfach und soll an einem vereinfachten Sensor gezeigt werden.



1. Belichten des Sensors über einen definierten Zeitraum (Integration).
2. Verschieben der gesammelten Ladung aller Pixelelemente in die benachbarten vertikalen Ausleseregister.
3. Die Ladungen anschließend zeilenweise in das horizontale Schieberegister bringen.
4. Das horizontale Schieberegister entleeren.
5. Nun die nächste Zeile in das horizontale Schieberegister auslesen.

Je nach Sensorgröße müssen diese Schritte solange wiederholt werden, bis der Sensor komplett ausgelesen ist.



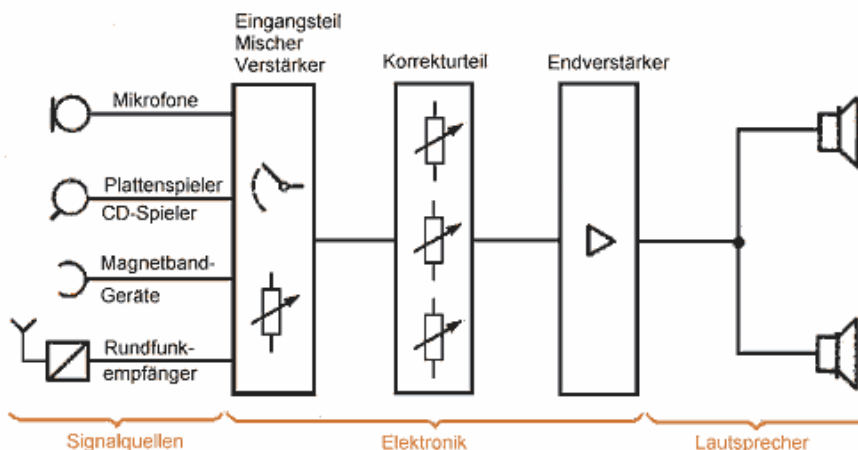
8.4 Audiotechnik im Computer und MIDI

8.4.1 Elektroakustik

Zu den klassischen elektronischen Einrichtungen, mit denen Informationen übermittelt werden, gehören elektroakustische Anlagen. Als Übermittler dient dabei primär der Schall.

Komponenten im Überblick

Grundsätzlich bestehen elektroakustische Anlagen aus verschiedenen Signalquellen, aus elektronischen Einrichtungen (Verstärker, Misch- und Verteileinrichtungen) sowie den Schallsendern, also den Lautsprechern.



Als Signalquellen können Schallempfänger (Mikrofone) oder Speichereinrichtungen, wie Magnetbandgeräte, Schallplatten- und CD-Abspielgeräte, dienen. Rundfunkempfangseinrichtungen (Tuner) oder die Audioteile von Videorecordern sind ebenfalls häufig Signalquellen für solche Anlagen.

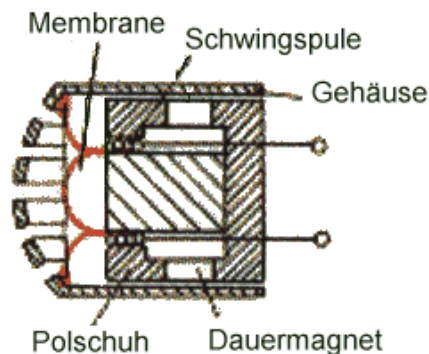
Mikrofone geben eine Spannung von ungefähr 2 mV/Pa ab. Da die menschliche Stimme einen Schalldruck von ca. 0,5 Pa erzeugt, liefern Mikrofone eine Spannung von ca. 1 mV. In der gleichen Größenordnung liegen die Spannungswerte des Tonabnehmers am Schallplattenspieler. Wesentlich

höhere Spannungswerte erhält man aus CD-Spielern, Magnetbandgeräten und dem Rundfunkempfangstuner, nämlich 100 ... 200 mV.

Das Eingangsteil einer elektroakustischen Anlage hat die wichtige Aufgabe, die verschiedenen Pegel der Signalquellen aneinander anzupassen. Hier sind spezielle Eingänge für ein oder mehrere Mikrofone vorhanden, die mit zusätzlichen Vorverstärkern ausgestattet sind. Einen Vorverstärker hat auch der Eingang für den Plattenspieler mit elektromagnetischem Abtastsystem. Allerdings ist dieser gleichzeitig mit Korrekturgliedern zum Anpassen an die Schneidkennlinie der Schallplatten ausgestattet. Bei diesen werden nämlich, um einen geringen Rillenabstand und damit lange Spielzeiten zu erzielen, die tiefen Töne beim Aufnehmen mit konstanter Auslenkung und nicht mit ihrem tatsächlichen Stichelausschlag in die Platte geschnitten. Das hat zur Folge, dass die tiefen Frequenzen bei der Wiedergabe mit geringerer Spannung als die mittleren und hohen Frequenzen abgegeben werden. Sie müssen folglich mit Korrekturgliedern gegenüber diesen angehoben werden. Auf Verstärker und Korrekturglieder kann bei piezoelektrischen Kristalltonabnehmern verzichtet werden, weil diese ohnehin eine höhere Spannung als die mit elektromagnetischem Abtastsystem abgeben und die tiefen Frequenzen von Natur aus anheben. Wegen ihrer mäßigen Qualität (Verzerrungen, Frequenzgang) verwendet man sie in Beschallungsanlagen nur in Notfällen.

Mikrofone

Heute verwendet man als Schallempfänger meistens dynamische Mikrofone. Kristall-, Kohlegrieß- und magnetische Mikrofone beeinflussen die Verständlichkeit sehr ungünstig, weshalb man sie möglichst meidet. Kondensatormikrofone bieten zwar sehr hohe Aufnahmequalität, sind aber sehr teuer und damit nur im Studiobetrieb zu finden. Elektrodynamische Mikrofone, auch Tauchspulmikrofone, nutzen das elektrodynamische Prinzip aus, das heißt, eine Spule wird durch den auf die Membran einwirkenden Schall im Luftspalt eines Topfmagneten bewegt. Dadurch wird in der Spule eine Spannung induziert.



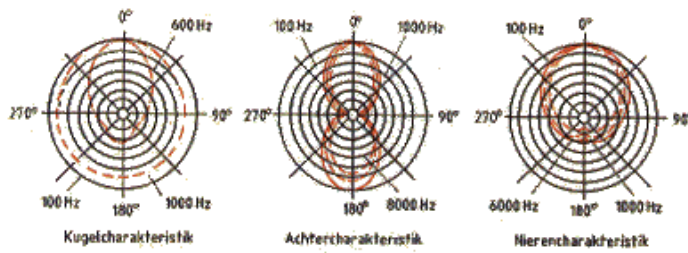
Wie man im Schnitt des Tauchspulmikrofons erkennt, ist die Schwingspule mit einer Membran verbunden, die den Schalldruck direkt in Spulenbewegungen umsetzt.

Eine besondere Form des elektrodynamischen Mikrofons ist das Bändchenmikrofon. Bei ihm schwingt zwischen den Polen eines Magneten ein zwei Mikrometer dickes und vier Millimeter breites Metallbändchen, das gleichzeitig als Membran dient. Durch seine geringe Masse erhält man geringe Verzerrungen und ebenso hochwertige Aufnahmen wie mit

Kondensatormikrofonen. Leider liefern sie nur geringe Ausgangsspannungen, sind recht groß und schwer und

dazu noch relativ teuer. Deshalb werden sie kaum noch verwendet.

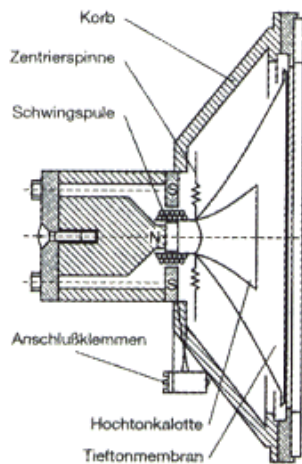
Zu den Mikrofonkenngrößen, die nicht vom Wandlungsprinzip abhängen, zählt die Richtcharakteristik. Die Richtcharakteristik gibt die Empfindlichkeit in Abhängigkeit vom Schalleinfallswinkel an. Durch konstruktive Maßnahmen lassen sich Kugel-, Achter- und Nierencharakteristik erzielen.



Verständlichkeit

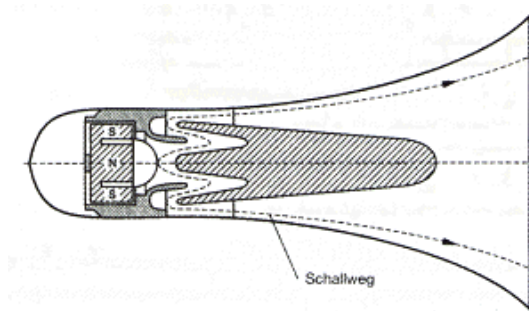
Für die Verständlichkeit sind mehrere Faktoren maßgebend:

- Qualität des Mikrofons,
- Leistung und Qualität der Endverstärker,
- Leistung und Qualität der Lautsprecher und deren Anordnung und
- die akustischen Eigenschaften des Raumes.



Bei den Lautsprechern hat sich auch das dynamische Wandlerprinzip durchgesetzt, bei dem sich eine vom Signalstrom durchflossene Spule im Luftspalt eines Permanentmagneten bewegt und eine Membran antreibt.

Bei der Lautsprecheranordnung müssen die akustischen Eigenschaften des Raumes unbedingt berücksichtigt werden. Unterlässt man das, so erhält man Zustände wie auf vielen Bahnhöfen. Für eine verständliche Sprachdurchsage haben sich Hornlautsprecher bewährt. Das sind Kalottenlautsprecher mit vorgesetztem Exponentialtrichter. Mit ihnen lassen sich hohe und mittlere Frequenzen bei gutem Wirkungsgrad abstrahlen. Sie werden auch Druckkammerlautsprecher genannt, weil die Kalotte die Luft in eine abgeschlossene Kammer geringeren Querschnitts hineindrückt. Dabei findet eine sogenannte Geschwindigkeitstransformation statt, das heißt, die Geschwindigkeit der Luftteilchen erhöht sich im Verhältnis zwischen Kalottenquerschnitt und Druckkammerquerschnitt. Mit den schnellen Luftteilchen wird ein vor die Druckkammer gesetzter Exponentialtrichter, ähnlich wie ihn viele Blasinstrumente haben, angeregt. Für tiefe Frequenzen müsste der Trichter zu große Abmessungen haben. Deshalb wird er meistens gefaltet.

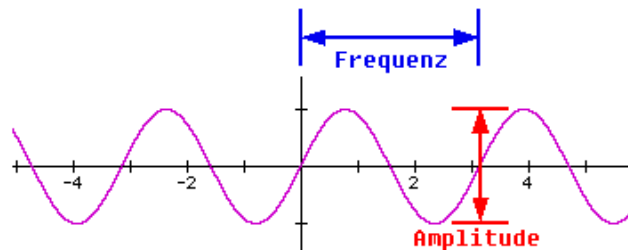


8.4.2 Klang- und Tonerzeugung

Zunächst wollen wir uns mit der Erzeugung von Klängen beschäftigen. Künstliche Töne oder Klänge werden ja dadurch erzeugt, daß die Membran eines Lautsprechers durch elektrische Signale in Schwingung versetzt wird. Je nachdem, wie oft in der Sekunde die Polarität des Stroms durch die Magnetspule des Lautsprechers wechselt, hören wir einen tiefen (wenige Wechsel) oder einen hohen Ton (viele Wechsel). Die Höhe eines Tons (= Frequenz) wird in Hertz (= Schwingungen pro Sekunde) angegeben. Ausgegangen wird dabei zuerst immer von Sinusschwingungen. Man kann eine Sinusschwingung durch folgende Gleichung beschreiben:

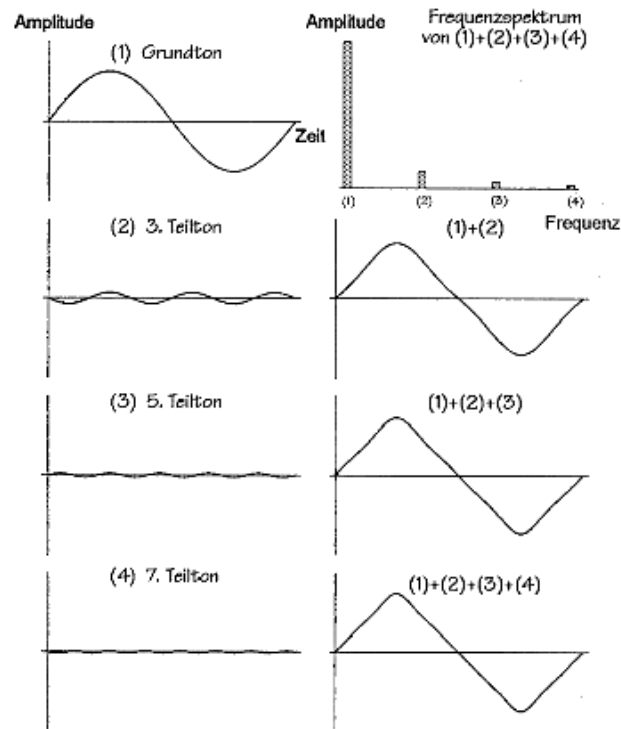
$$f(t) = a \cdot \sin(2 \cdot \pi \cdot f \cdot t)$$

Wie man sieht, hat die Funktion nur zwei Parameter, die sich variieren lassen: Die Amplitude a (= Lautstärke, gemessen in dB) und die Frequenz f (= 1/Wellenlänge = Tonhöhe). Eine einfache Sinusschwingung gibt somit einen recht langweiligen Ton.

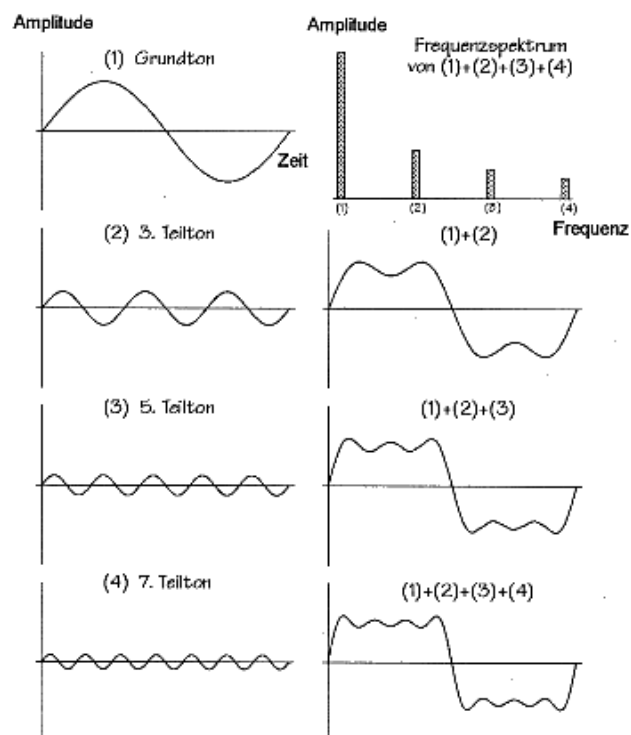


Um unterschiedliche Klangfarben zu erzeugen, muß man verschiedene Wellenformen wählen und diese Wellenformen miteinander mischen. Bezeichnet wird die Wellenform nach Ihrem Aussehen auf dem Bildschirm eines Oszilloskops. Die "reinste" Form ist die oben erwähnte Sinuslinie. Eine Dreieckschwingung klingt schon härter. Sie wird folgendermaßen aus Sinusschwingungen zusammengesetzt:

Datenverarbeitungssysteme

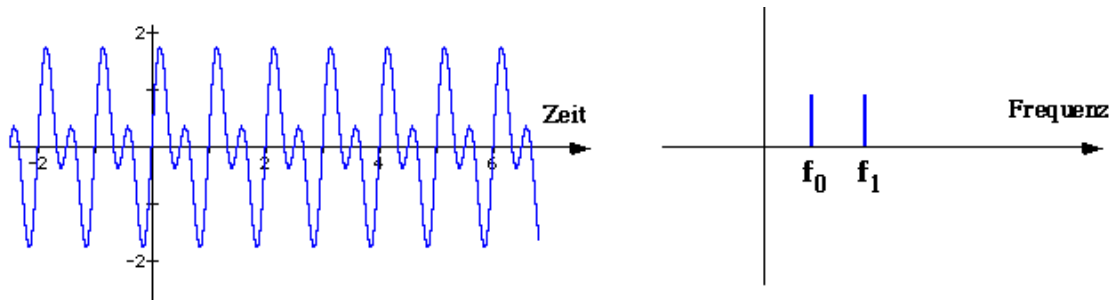


Die Rechteckschwingung (praktisch das Ein- und Ausschalten des Stroms) hat den schärfsten Klang. Sie entsteht ebenfalls aus einem Gemisch von Sinusschwingungen:



Durch Überlagern und Mischen von unterschiedlichen Schwingungsformen von verschiedener Frequenz entstehen dann ganz charakteristische Klänge. Die Erzeugung dieser Töne faßt man unter dem Begriff 'Klangsynthese' zusammen.

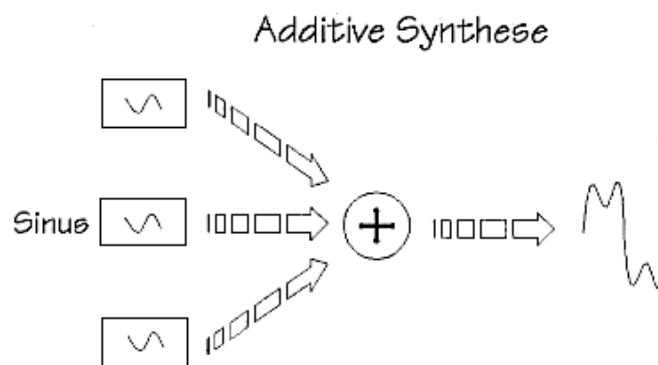
Fourier: Jede Schwingung kann als Summe von Sinusschwingungen dargestellt werden:



Lineare Synthese

In der Elektronik gibt es zwei grundsätzliche Methoden, um die Erzeugung von Klängen aus Teiltongemischen technisch zu verwirklichen. Bei der additiven Klangerzeugung werden Sinustöne gemischt, bei der subtraktiven Sinustöne aus einem Klanggemisch ausgefiltert. Beides sind Verfahren der linearen Synthese, in der ein geradliniger Zusammenhang zwischen Eingabe und Ausgabe besteht. _brigens werden auch Farben mit additiven und subtraktiven Verfahren erzeugt, denn von der mathematischen Theorie her besteht in dieser Hinsicht kein wesentlicher Unterschied zwischen Licht- und Tonschwingungen. Etwas von dieser Gemeinsamkeit kommt darin zum Ausdruck, daß Musiker ebenso gern von 'Klangfarben' wie Maler von 'Farbtönen' sprechen. Es kommt bei einem System, das mit der additiven Synthese arbeitet, nur das heraus, das man vorher hineingetan hat. Heißt z. B. eine Eingabe a , bei der das Ergebnis a' lautet, und eine andere b mit dem Ergebnis b' , dann erhält man bei Eingabe von $a+b$ das Ergebnis $a' + b'$. Man spricht von einem *linearen System*, und auch die subtraktive Synthese verhält sich in dieser Weise linear. Die *nichtlinearen Systeme* der Klangsynthese zeichnen sich hingegen dadurch aus, daß sie Frequenzen erzeugen, deren Komponenten man vorher nicht 'direkt' eingegeben hatte.

Der additiven Synthese liegt die Idee zugrunde, komplexe Schwingungen durch Addition von einfachen zu erhalten, wie es die Fourier-Methode ermöglicht. Sie heißt daher auch Fourier-Synthese. Als Ausgangspunkt dienen Sinusschwingungen, die im richtigen Mischungsverhältnis, d.h. mit passender Tonhöhe und Lautstärke, gleichzeitig erklingen. Dadurch entstehen die oben beschriebenen Wellenformen (Dreieck-, Sägezahn- oder Rechtecksschwingungen).

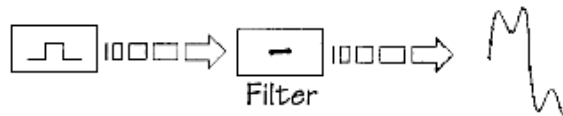


Die additive Synthese spielte besonders in frühen elektronischen Instrumenten wie der Hammond-Orgel eine wichtige Rolle. Bei den Konstrukteuren digitaler Computer ist die additive Synthese wegen ihres hohen Rechenaufwandes wenig beliebt. Eine weitere sehr interessante Anwendung ist das Erzeugen akustischer Illusionen. Man kann Töne synthetisieren, die die Illusion erzeugen, ständig höher zu werden, ohne jemals den Hörbereich zu verlassen (Shepard-Effekt).

Führt bei der additiven Analyse der Syntheseweg von einfachen Klanggemischen zu Komplizierten, so ist es bei der subtraktiven Synthese gerade umgekehrt. Aus einem komplexen Signal, das möglichst viele Teiltöne enthält (z. B. ein Rechtecksignal), werden die unerwünschten

Komponenten ausgefiltert.

Subtraktive Synthese

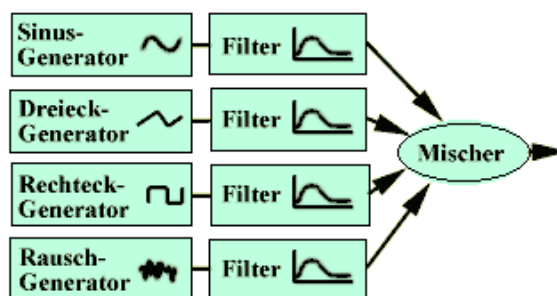


Viele der ersten Synthesizer benutzen diese Anordnung, z. B. die Moog-Synthesizer. Auch bei Elektroorgeln sind Sägezahnwellen, die (fast) alle Teiltöne enthalten, oder Rechteckswellen beliebt. Die subtraktive Synthese ist auch das natürliche Prinzip, nach dem traditionelle Instrumente wie Flöte, Geige, Klavier usw. funktionieren, denn bei ihnen wirken Stoff und Form der Resonanzmaterialien wie Holz, Metall, Darm usw. als natürliche Filter. Resonanz bedeutet, daß auf Grund der Eigenschaften eines Materials bestimmte Frequenzbereiche besonders hervorgehoben werden.

Ausgeprägte Resonanzen über feste Frequenzbereiche heißen Formantbereiche. Für den Vokal 'u' liegt ein solcher Bereich beispielsweise zwischen 200 und 400 Hz, beim 'o' von 400 bis 600 Hz und beim 'a' zwischen 800 und 1200 Hz. 'e' und 'i' haben sogar jeweils zwei Formanten, das 'e' bei 400 bis 600 Hz bzw. 2200 bis 2600 Hz, das 'i' bei 200 bis 400 Hz bzw. 3000 bis 3500 Hz. Mund- und Zungenstellung wirken dabei wie Filter, die bestimmte Frequenzen durchlassen oder abschwächen. In der Elektronik kann sich Resonanz so stark aufschaukeln, daß selbständige Töne entstehen (Selbsterregung). Für solche Effekte wurden besonders die Filter der Moog-Synthesizer berühmt. Man baut also mit elektronischen Mitteln natürliche Modelle nach, und das Basismodell, auf dem die subtraktive Synthese beruht, besteht darin, daß die Klänge einer irgendwie angeregten Klangquelle einem Resonanzsystem zugeführt werden.

Kombiniert man additive und subtraktive Synthese, erhält man weitere Variationsmöglichkeiten.

Sound Synthese



Nichtlineare Synthese

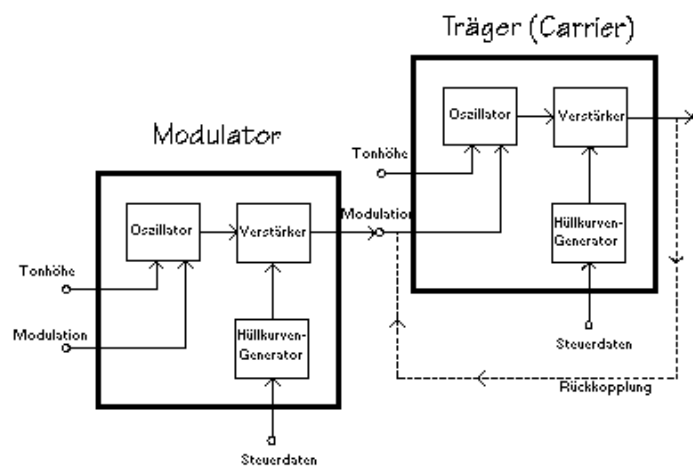
Durch die digitalen Synthesizer wurden ältere, seit dem vorigen Jahrhundert entwickelte technische Verfahren zur Klangsynthese zwar nicht überflüssig. Im Gegenteil: auch der digitale Synthesizer machte reichlich von ihnen Gebrauch. Seine Stärke ist jedoch das Rechnen die neuentwickelten nicht-linearen Verfahren zur Klangsynthese sind daher mathematische Rechenverfahren, die erst durch die Digitaltechnik technisch zu realisieren waren.

Zur Erzeugung von Schwingungsformen braucht der digitale Synthesizer Zahlenwerte. Diese erzeugt er entweder selbst durch Rechenmechanismen, indem er z. B. die Werte einer Dreiecksschwingung ausrechnet, oder aber er entnimmt sie einer vorher im Speicher abgelegten Wellentabelle (Wavetable). Diese kann als Ausgangsmaterial für neue Töne digitalisierte Klänge

traditioneller Instrumente enthalten. Bei der Klangsynthese werden die Wavetables vielfach verändert und einer nichtlinearen Wellenformung (waveshaping) unterworfen. Nichtlineare Klangformungsmethoden sind sehr wirksam, weil sie mit relativ wenig Aufwand sehr komplexe Klänge erzeugen können. Daraus ergeben sich aber auch Probleme in der Beherrschbarkeit solcher Verfahren. Man will ja nicht irgendwelche Klänge, sondern musikalisch brauchbare, die sich leicht steuern lassen, und das ist bei der nichtlinearen Synthese nicht mehr selbstverständlich. Ihre größten Erfolge haben nichtlineare Methoden bei komplizierten Glocken und Perkussionsklängen, bei denen auch starke Geräuschanteile benötigt werden.

FM-Synthese

Um 1973 entwickelte John Chowning, Professor an der amerikanischen Stanford-Universität, die FM-Synthese (von Frequency Modulation = Frequenzmodulation). Sie beruht auf Methoden der höheren Mathematik, die nicht nur Summen und Integrale (wie die Fourier-Analyse) benötigen, sondern auch bestimmte Formen von Differentialgleichungen (Besselfunktionen). Chownings Verfahren wurde in unterschiedlicher Form in zahlreichen Synthesizern der 80er und 90er Jahre eingesetzt.



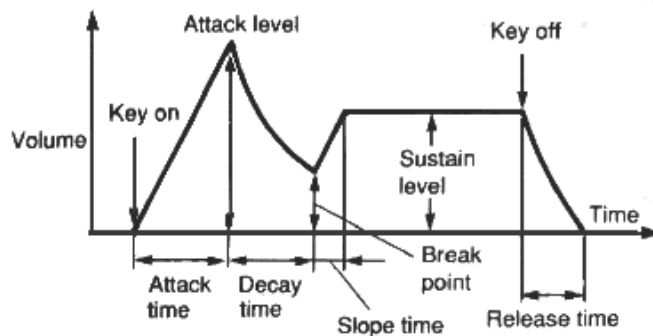
Man verwendet für FM-Synthese im einfachsten Fall zwei Frequenzgeneratoren, wobei der eine Generator die 'Trägerschwingung', also den eigentlichen Ton, erzeugt. Der zweite Generator erzeugt die 'Modulatorschwingung', welche die Trägerschwingung beeinflusst. Das Signal besteht also aus der Addition beider Schwingungen und folgt dann der (nichtlinearen) Gleichung:

$$f(t) = a_1 \cdot \sin(2 \cdot \pi \cdot f_1 \cdot t + a_2 \cdot \sin(2 \cdot \pi \cdot f_2 \cdot t))$$

Während der Träger direkt mit dem Ausgang verbunden ist, hört man den Modulator nur in seiner Auswirkung auf den Träger. Die Veränderungen des Trägers sind durch die Ausgangslautstärke des Modulators einstellbar. Die Veränderung dieses Ausgangspegels kann durch eine Hüllkurve erfolgen, wodurch der Ton lebendiger und dynamischer wird. In der Praxis wird die FM-Synthese noch mit diversen Rückkopplungsmöglichkeiten verknüpft. Dazu schaltet man den Ausgang des Trägers zurück auf seinen Modulationseingang, wie dies im Bild oben durch die gestrichelte Linie angedeutet ist.

Bei einem Klang, zum Beispiel der Ton eines Klaviers, sind aber noch weitere Parameter zu berücksichtigen. Der wichtigste Parameter ist hier wohl die Änderung der Lautstärke über die Zeit. Wenn Sie eine Taste anschlagen, hört nach dem Loslassen der Taste der Ton ja nicht sofort auf, sondern er fällt langsam ab. Man bezeichnet den Verlauf der Lautstärke als Hüllkurve. Die einzelnen Teile der Hüllkurve haben alle eine bestimmte Bedeutung. Beim Anschlagen der Taste (Anblasen einer Flöte, Streichen mit dem Bogen) "Attack" steigt die Lautstärke von Null auf ihr Maximum, um dann sofort wieder abzufallen "Decay". Lässt sich der Ton halten ("Sustain") sinkt

die Lautstärke nur auf den Haltewert, um dann nach den Ausschalten auf Null abzusinken ("Release").

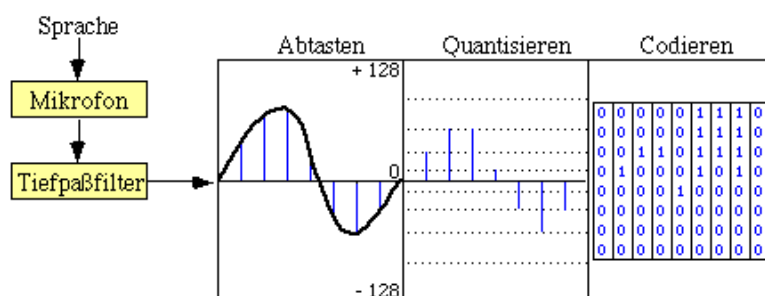


Man kann den Ton nun weiter manipulieren. Durch einen weiteren Generator mit sehr niedriger Frequenz (LFO) kann man ein Vibrato erzeugen, indem man die Frequenz des Tons mit dieser Niedrigst-Frequenz (einige Hertz) moduliert. Hinter die Frequenzerzeugung lassen sich Filter schalten, die den Klang weiter verändern. Das läßt sich zum Teil sogar ohne jegliche Hardware rein rechnerisch machen; viele Computer verwenden jedoch bestimmte integrierte Bausteine für diesen Zweck, sogenannte Soundgeneratoren. Die von diesen Bausteinen erzeugten Töne klingen je nach dargestelltem Instrument stärker oder weniger stark "künstlich".

Um die Klänge realer Instrumente naturgetreu wiederzugeben, läßt sich aber auch eine andere Methode verwenden. Man schließt an den Computer einen Analog-Digital-Wandler an, der die komplette Wellenform von Klängen aufzeichnet. Um hier gute Ergebnisse zu erzielen, braucht man nicht nur Bausteine, die sehr schnell arbeiten, sondern der Computer benötigt auch einen großen Speicher zur originalgetreuen Aufzeichnung der Klänge. So aufgezeichnete Instrumente klingen dafür auch sehr realistisch. Früher hat man solche Klänge nur extern gespeichert (z. B. auf der Festplatte) und per Abspielprogramm wiedergegeben. Heute bieten auch die Soundkarten bereits gespeicherte Klänge ('Wavetable-Modul').

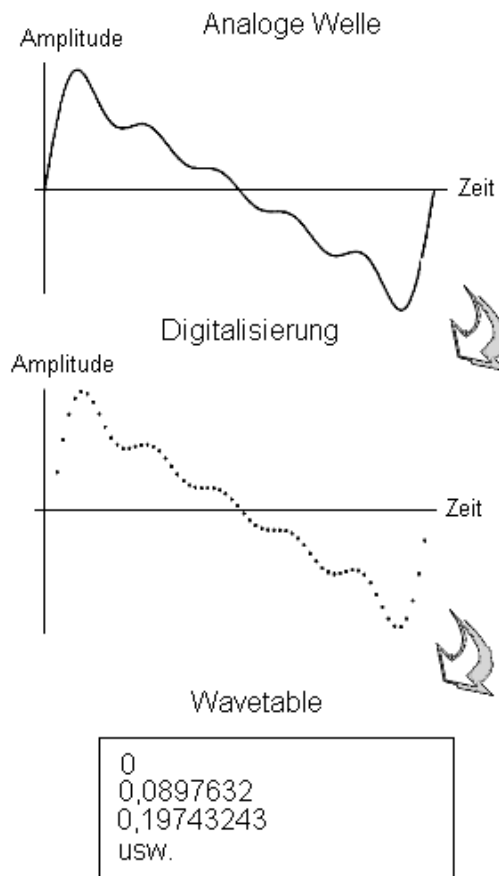
8.4.3 Digitalisierung von Tönen

Zur Aufnahme wird ein Analog-Digital-Umsetzer verwendet, der die vom Mikrofon oder der Audio-Quelle stammenden analogen Signale in digitale Werte umwandelt.



Digitalisierung ist ein Prozeßblock, der aus zwei Teilen besteht. Der erste hat die Aufgabe, das aufgezeichnete Material in eine computergerechte Form zu bringen, es wird analog/digital gewandelt (A/D-Wandlung). Das folgende Bild zeigt, wie aus einer analogen Sinuskurve eine Wavetable, d. h. eine Zahlenmenge, wird. Der zweite Teil muß die Ergebnisse der Arbeit des Computers wieder den menschlichen Sinnen zugänglich machen, und zwar durch den umgekehrten Vorgang, die Digital-Analog-Wandlung (D/A-Wandlung).

Datenverarbeitungssysteme

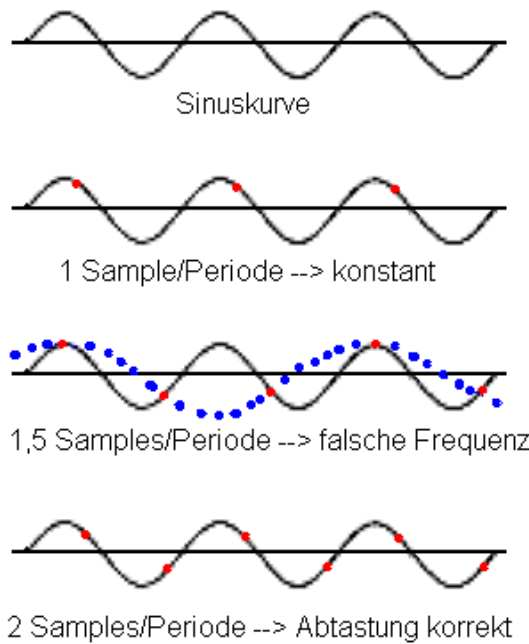


Das entscheidende Charakteristikum der Digitalisierung besteht darin, daß aus einem kontinuierlichen Kurvenzug mit unendlich vielen Zwischenwerten eine kleine Menge diskreter Punkte wird. Der analoge Ton kann rein theoretisch in unendlich viele Werte zerlegt werden, er ist ein kontinuierliches Signal, das durch den Digitalisierungsvorgang in eine endliche Anzahl von Einzelpunkten aufgelöst wird. Auch wenn eine analoge Schwingung nur die endliche Zeitdauer von einer Sekunde hat, setzt sich diese doch aus unendlich vielen Zeitpunkten zusammen. Die Digitalisierung ändert daher eine wichtige Eigenschaft des analogen Signals, indem sie es aus einem kontinuierlichen und unendlichen in ein diskretes und endliches verwandelt.

Wie läuft so etwas technisch ab? Der Schlüsselbegriff der modernen Digitaltechnik heißt Sampling (sample = Beispiel, Muster). Ein digitales Signal entsteht dadurch, daß von einem analogen Signal Samples genommen werden. Diese Abtastung geschieht in regelmäßigen Abständen mit hoher Geschwindigkeit, typisch sind 40 bis 50 kHz. Die 'Löcher' der digitalen Welle im Bild oben werden dadurch gefüllt, es entsteht eine treppenförmige Kurve. Der Abstand jedes Punktes von der Null-Linie ist der Digitalwert, der als reiner Zahlenwert gespeichert wird. Mit einer für Soundadapter üblichen Sampling-Rate von 48 KHz sind das 48000 Werte pro Sekunde, die einen entsprechenden Speicherplatzbedarf haben. Ein zweites Kriterium ist die Auflösung eines Samples, d. h. in wieviele Stufen der Spannungsbereich zwischen Null und der maximalen Amplitude aufgeteilt wird.

Die Digitalisierung führt zu einer Reihe von störenden Nebenwirkungen, die grundsätzlicher Natur sind und durch besondere Maßnahmen beseitigt werden müssen. Sie treten bei der Wiedergabe, d. h. der D/A-Wandlung, auf. Angenommen man hätte pro Welle einer Schwingung nur ein einziges Sample genommen. Dann gibt es bei der D/A-Wandlung mehrere (!) Möglichkeiten, aus diesem eine analoge Schwingung zu rekonstruieren. Da man dann nicht weiß, wo die negative Halbwelle liegt, paßt z. B. auch eine Welle mit halber Frequenz zu diesem Wert. Wir brauchen also mindestens zwei Werte, um eine Schwingung aus Wellenberg und Wellental richtig vermessen zu können. Die folgende Grafik illustriert diese Problematik:

Datenverarbeitungssysteme



Daraus erklärt sich auch z. B. die bei CD-Spielern verbreitete Sampling-Rate von 44,1 KHz. Da der Mensch Töne bis etwa 20 kHz hören kann, müssen diese mit der doppelten Frequenz abgetastet werden, damit sie eindeutig analysierbar sind. Man kann beweisen, daß bei Erfüllung der beiden folgenden Forderungen keine Informationsverluste entstehen, wenn das digitale Signal wieder in ein analoges zurückverwandelt wird:

1. Die Sampling-Rate muß mindestens doppelt so groß sein wie die höchste zu verarbeitende Frequenz.
2. Das Signal darf keine Frequenzen enthalten, die über der halben Sampling-Rate liegen.

Beide Forderungen folgen aus zwei wichtigen Forschungsergebnissen der Nachrichtentechnik und Informationstheorie. Harry Nyquist formulierte 1928 das Nyquist-Kriterium, nach der man eine Sampling-Frequenz finden kann, mit der es möglich ist, ein digitalisiertes Signal verlustfrei wieder zurückzuverwandeln. Der amerikanische Mathematiker Claude E. Shannon stellte 1949 das Abtasttheorem auf. Es gibt an, wie viele Werte man für eine verlustfreie Rekonstruktion des Ausgangssignals benötigt. Aus technischer Sicht folgt aus allem, daß man unbedingt sicher gehen muß, daß keine höheren, möglicherweise außerhalb des Hörbereichs liegende Frequenzen in dem zu digitalisierenden Signal enthalten sind. Andernfalls treten hörbare Störungen auf, die man als *Aliasing* (von lat. alias = anders) bezeichnet. Daher wird am Eingang eines A/D-Wandlers ein Tiefpaßfilter eingesetzt, das alle unerwünschten Frequenzen unterdrückt. Die moderne CD-Technik erleichtert dies noch durch *Oversampling*. Dabei wird das Eingangssignal mit einer sehr hohen Frequenz abgetastet (bis zu 6 MHz), die die Filterung verbessert, wonach dann die erhaltenen Werte auf die eigentliche Sampling-Frequenz (z. B. 48 kHz) heruntergerechnet werden.

Zur Wiedergabe wird umgekehrt ein Digital-Analog-Umsetzer verwendet, der die digitalisierten Wellenformen wieder in analoge Spannungsschwankungen umsetzt. Danach kann über einen Verstärker und Lautsprecher die Umsetzung in akustische Schwingungen erfolgen. Die 'Lücken' bzw. 'Treppen' eines digitalen Signals müssen wieder durch ein Filter beseitigt werden. Auch hier hilft Oversampling bei der Unterdrückung von Störungen, u. a. bei der Reduzierung von Quantisierungsrauschen, das während des Digitalisierungsprozesses entsteht. Die Manipulation der Abtastrate ermöglicht aber auch typische Sampler-Effekte. Wird etwa das Sample bei der Wiedergabe mit einer anderen Abtastrate als bei der Aufnahme abgespielt, ändert sich die Tonhöhe.

Hardware zur Aufnahme und Wiedergabe von Tönen

In den Soundinterfaces und -karten werden Synthesizer eingesetzt, die den Ton aus Komponenten zusammensetzen:

- Sinus-, Dreieck- und Rechteckwellen unterschiedlicher Frequenz,
- Rauschen,
- und gefilterte Komponenten aus den beiden o. a. Komponenten.

Dabei überlagern sich die einzelnen Wellen zu einer Gesamtwellenform, die den Klang ergibt. Dieser Klang ist jedoch immer etwas künstlich. Um originalgetreuere Klänge zu erzeugen müssen andere Wege beschritten werden.

Mit der Wavetable-Synthese wird eine Kompromißlösung beschrieben. Hier werden die Klangformen digital abgespeichert und nur die Tonhöhe und Dauer programmiert. Damit lassen sich natürlich klingende Instrumente erzeugen ohne zuviel Speicher zu belegen.

Will man noch mehr Flexibilität, so kann die gesamte Wellenform digitalisiert werden. Damit lassen sich dann auch Sprache und Geräusche auf dem Rechner wiedergeben. Zur Aufnahme wird ein Analog-Digital-Umsetzer verwendet, der die vom Mikrofon, oder der Audio-Quelle stammenden analogen Signale in digitale Werte umwandelt.

Soundkarten erlauben meist unterschiedliche Werte für die Abtastrate: 11 kHz, 22 kHz, 44 kHz und 48 kHz oder stufenlose Einstellung. Um den Alias-Effekt bei niederen Abtastraten zu vermeiden, ist meist ein Filter vorhanden, der höhere Frequenzanteile unterdrückt, also werden beispielsweise bei einer Abtastrate von 11 kHz Frequenzen die höher als 5.5 kHz sind unterdrückt.

Wenn man Stereo-Sound ausgeben will, so benötigt man für jeden Kanal eine Wave-Datei. Die Wave-Dateien benötigen der Abtastrate entsprechend viel Speicherplatz. Bei 22 kHz sind das entsprechend 22000 kByte pro Sekunde Audio. Die Werte liegen bei den Soundkarten im übrigen nicht genau bei 11 kHz, 22 kHz oder 44 kHz, sondern es sind 11025 Hz, 22050 Hz und 44100 Hz.

Diese Werte kommen nicht von ungefähr, sondern sind typische Werte aus dem täglichen Leben. Dazu ein paar Beispiele:

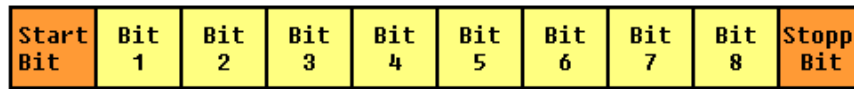
Qualität	Abtastrate [kHz]	Auflösung [bit]	Mono/Stereo	Datenrate [KByte/s]
Telefon	8	8	Mono	8
AM Radio	11	8	Mono	11
FM Radio	22	16	Stereo	88
CD	44	16	Stereo	176
DAT	48	16	Stereo	192

8.4.4 MIDI

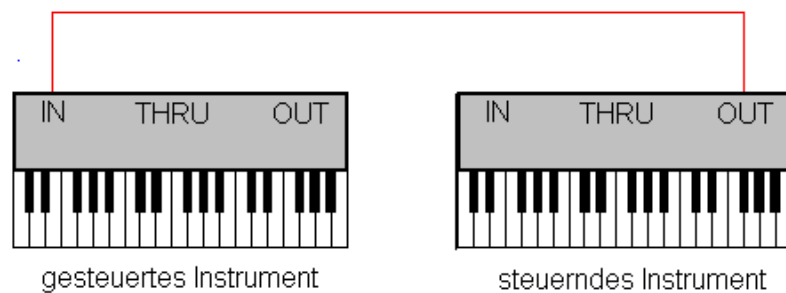
MIDI ist die Abkürzung für "Musical Instruments Digital Interface". Es handelt sich um eine genormte Schnittstelle für elektronische Musikinstrumente, die Anfang der achtziger Jahre entwickelt wurde.

In dieser MIDI-Norm wurde festgelegt, wie ein Steuercomputer mit den Musik-Geräten wie Synthesizer, Keyboard oder Drum-Box zu kommunizieren haben. Die Signale werden seriell, also

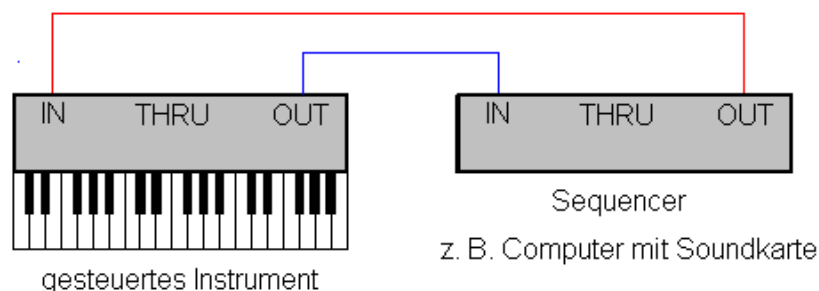
Bit für Bit gesendet und empfangen und zwar mit einer Geschwindigkeit von 31250 Bits/Sekunde. Die Übertragung erfolgt asynchron, das heißt, daß den Daten ein Startbit vorangestellt wird und ein Stopbit folgt:



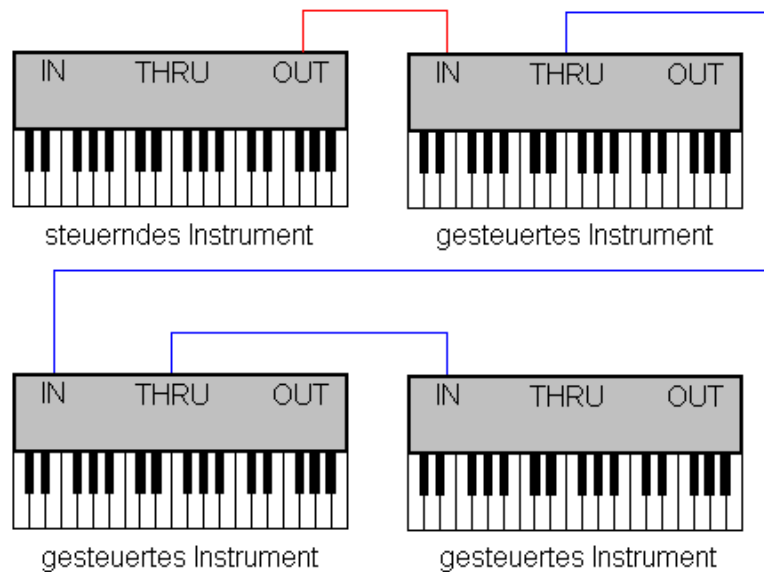
Das Datenformat erinnert stark an die Übertragung mit der seriellen V.24-Schnittstelle, lediglich die elektrischen Eigenschaften sind anders. Man kann aber die gleichen Schnittstellen-Bausteine verwenden, die man auch für die serielle Computerschnittstelle verwendet. Die Datenbits, die zum Beispiel vom Keyboard an die anderen Geräte gesendet werden, haben unterschiedliche Funktionen. Zuerst wird ein Statusbyte gesendet, das durch ein Einsbit an der ersten Stelle erkennbar ist. Dieses Byte beschreibt den Inhalt und die Anzahl der nachfolgenden Datenbits und bestimmte logische Einheiten des angesprochenen Geräts selektiert. Mit den Datenbytes, deren erste Stelle immer Null ist, lassen sich dann Informationen über die gerade gedrückte Taste, die Stärke des Anschlags, Änderungen in Klangfarbe, Tonhöhe oder Hüllkurve übermitteln. Steuert man zum Beispiel mit einem Keyboard weitere Keyboards, dann verhalten sich alle angeschlossenen Klaviaturen so, als würde bei ihnen die gleiche Taste mit der gleichen Anschlagstärke gedrückt.



Ist nun ein Computer als 'MIDI-Sequencer' ans Keyboard angeschlossen, kann er die Daten aufnehmen und speichern. Durch späteres 'Abspielen' der MIDI-Daten lassen sich die elektronischen Musikinstrumente wieder ansteuern. Man kann die Daten aber auch auf einem Synthesizer (extern oder auf der Soundkarte ausgeben und so die Instrumente 'simulieren'.



Die MIDI-Schnittstelle kann insgesamt 16 verschiedene, voneinander unabhängige Übertragungswege (Kanäle) auf einer Leitung bedienen. Die gesteuerten Instrumente werden dabei jeweils mit ihrem MIDI-IN-Eingang an die MIDI-THRU-Buchse des Vorgängers angeschlossen. Das erste gesteuerte Gerät wird an den MIDI-OUT-Ausgang des steuernden Instruments angeschlossen.

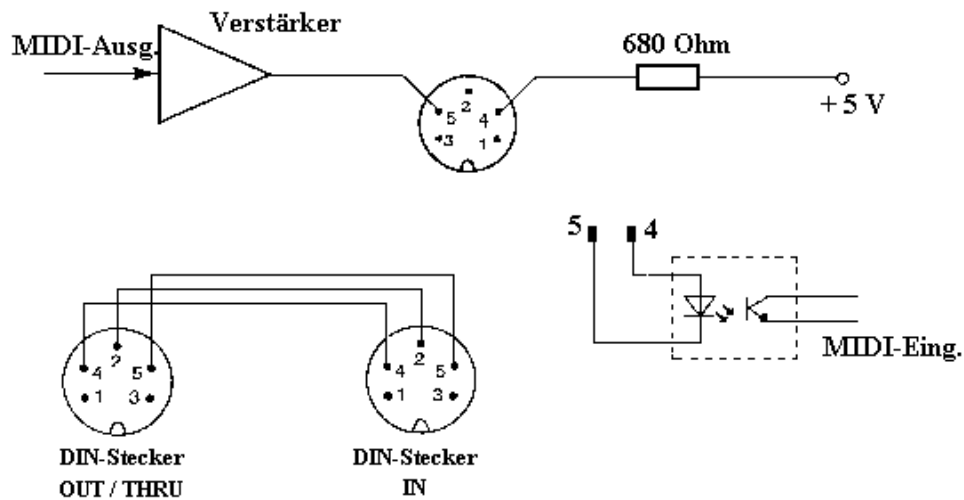


Das MIDI-Betriebssystem bietet drei Betriebsarten (Modi) um die Informationsübertragung auch sinnvoll zu nutzen: Omni-, Poly- und Mono-Mode. Diese Betriebsarten werten die Informationen, die von der Schnittstelle kommen, unterschiedlich aus:

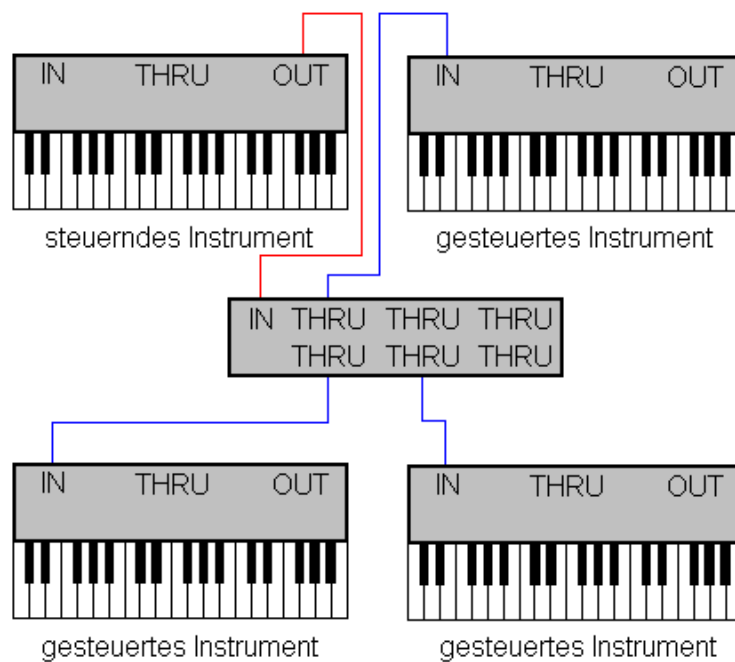
- Im *Omni-Mode* ignorieren die Geräte die Kanal-Umschaltbefehle, es spielen also alle Geräte dauernd mit. Sinnvoll wäre in diesem Mode eine Kombination aus zwei Keyboards und einem Rhythmusgerät. Auf einem Keyboard wird gespielt, das zweite wertet die Tastaturbefehle aus und die Drum-Box beachtet die Programmwechsel und Synchronisations-Information.
- Im *Poly-Mode* reagieren die angeschlossenen Geräte nur auf Befehle mit der Kanalnummer, auf die sie vorher eingestellt wurden. In diesem Modus könnte ein Computer als Sequencer arbeiten und mit verschiedenen Kanalnummern unterschiedliche Geräte ansteuern. Keyboard 1 empfängt so auf Kanal 1 andere Daten als Keyboard 2 auf Kanal 2 und kann so eine andere Tonfolge mit anderer Klangfarbe spielen (z. B. die zweite Stimme).
- Der *Mono-Mode* ist die komplexeste Stufe der MIDI-Möglichkeiten. In diesem Modus, der noch bei sehr wenigen Geräten zu finden ist, kann jede Stimme, die in einem Keyboard vorhanden ist, einzeln angesteuert werden. Auf einem achtstimmigen Synthesizer können also acht verschiedenen Klänge bzw. Instrumente gleichzeitig gespielt werden.

Auf den derzeit existierenden Geräten existieren jedoch noch viele Einschränkungen der mannigfachen Möglichkeiten, die von MIDI geboten werden.

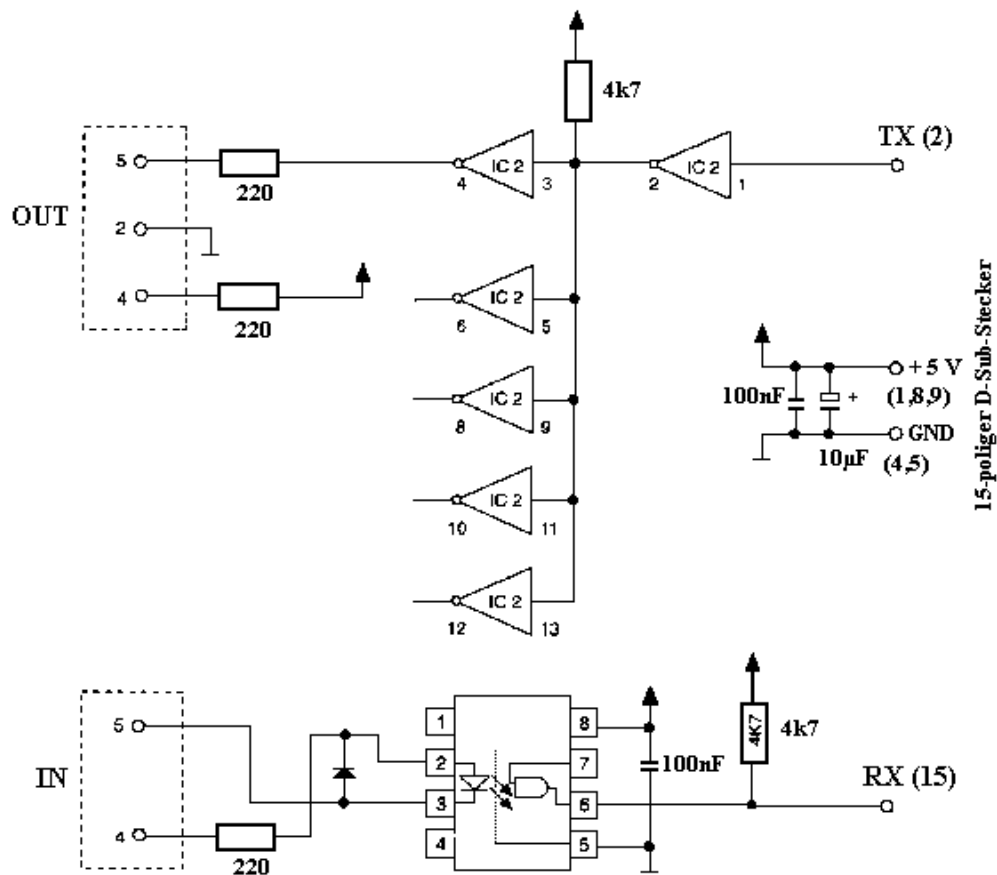
Auch die Schaltung der MIDI-Schnittstelle unterscheidet sich von der seriellen Computerschnittstelle. Damit sich die verschiedenen Musik-Maschinen der unterschiedlichsten Hersteller auf jeden Fall gut vertragen, ist der MIDI-Eingang durch einen Optokoppler elektrisch vollkommen von der Schaltung des jeweiligen Geräts getrennt (so kann auch durch einen Kurzschluß nicht die gesamte Anlage 'in die Luft gehen'). Vom Optokoppler geht es dann zum Schnittstellenbaustein (UART), der aus den seriellen Bits wieder ein Byte zusammensetzt. Gleichzeitig läuft das Signal zu einer Buchse (MIDI-THRU), die das Signal an nachfolgende Geräte weitergibt. Der Ausgang des Geräts (also vom UART) geht direkt an die Ausgangsbuchse. Zur Verbindung der Geräte werden die im Audio-Bereich üblichen 5poligen DIN-Buchsen verwendet und man kann beliebige Verbindungskabel mit 5poligen DIN-Steckern zum Koppeln der einzelnen Geräte verwenden.



Aber auch mit solchen Sicherheits-Schaltungen gibt es Probleme, wenn die Kette aus zuvielen Geräten besteht. Es entstehen geringe Laufzeiten und Signalverzerrungen. Man greift dann zu einen MIDI-Verteiler und bedient die angeschlossenen Geräte sternförmig vom Sequencer aus.



Bei den meisten Soundkarten, z. B. Soundblaster, ist dieser Teil der MIDI-Schnittstelle nicht auf der Karte enthalten, er befindet sich in der externen MIDI-Box. Die folgende Schaltung zeigt, wie so eine MIDI-Box aussehen könnte. Die Pinnummern auf der rechten Bildseite bezeichnen die Pins der 15poligen SUB-D-Buchse auf der Soundkarte, die Pinnummern auf der linken Seite die Anschlüsse der MIDI-DIN-Buchse.



Es gibt noch zahlreiche weitere E/A-Geräte, die hier nicht besprochen wurden, z. B. Lesegeräte für Strichcodes, berührungsempfindliche Bildschirme, "Datenhandschuhe" mit Lageerkennung (X-, Y-, Z-Koordinate im Raum) und Dehnmessstreifen in den Fingern, Datenhelme mit Lageerkennung und integriertem Stereo-Monitor sowie die gesamte Prozessperipherie (Robotersteuerung, Mustererkennung, etc.).



[Zum vorhergehenden Abschnitt](#)



[Zum Inhaltsverzeichnis](#)



[Zum nächsten Abschnitt](#)

Die HTML-Fassung entstand unter Mitwirkung von Volker Arndt
Copyright © FH München, FB 04, Prof. Jürgen Plate