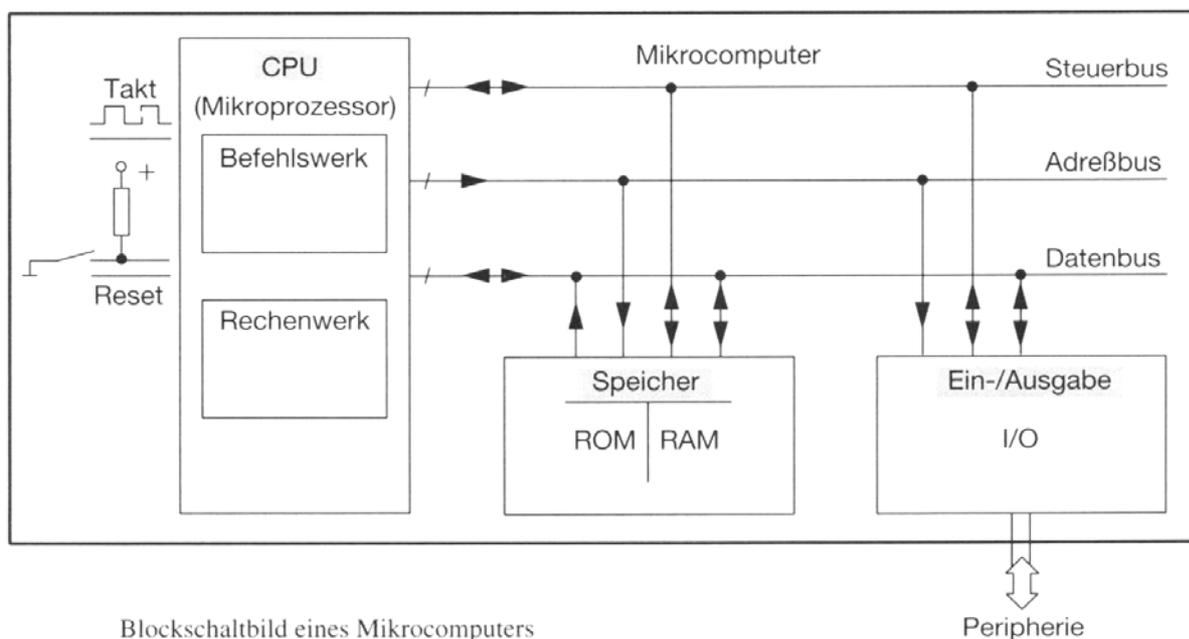
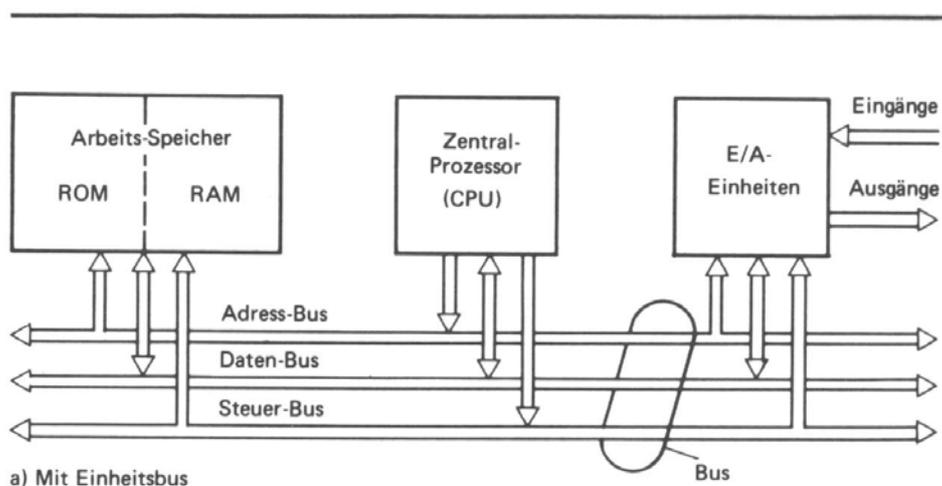


Übungen zur Vorlesung: Betriebssysteme und Rechnernetze für Ingenieure

1) Rechnerarchitektur

Blockschaltbild eines Mikrocomputers:

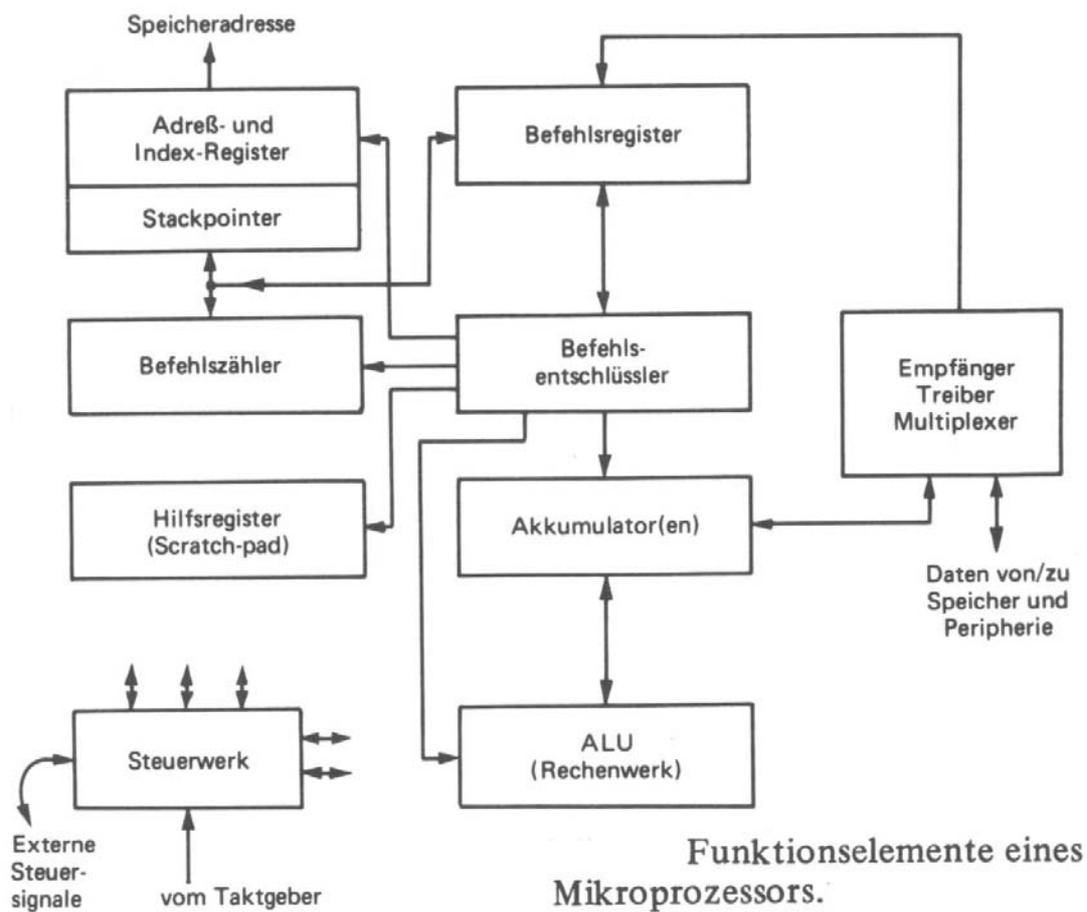
Zentralprozessor (central processing unit CPU)



Blockschaltbild eines Mikrocomputers

Peripherie

Aufbau und Funktion des Prozessors



Begriffe:

Zentraleinheit	CPU	Central-Processor-Unit
Steuerwerk	CU	Control-Unit
Rechenwerk	ALU	Arithmetic-Logic-Unit
Arbeitsspeicher	ROM	Read Only Memory
	RAM	Random Access Memory
Bus	Sammelleitung zum Austausch elektrischer Signale	

Teile des Rechenwerkes

Arbeitsregister: schnelle lokale Speicher zur Aufnahme der Operanden
speziell: Akkumulator – Arbeitsregister

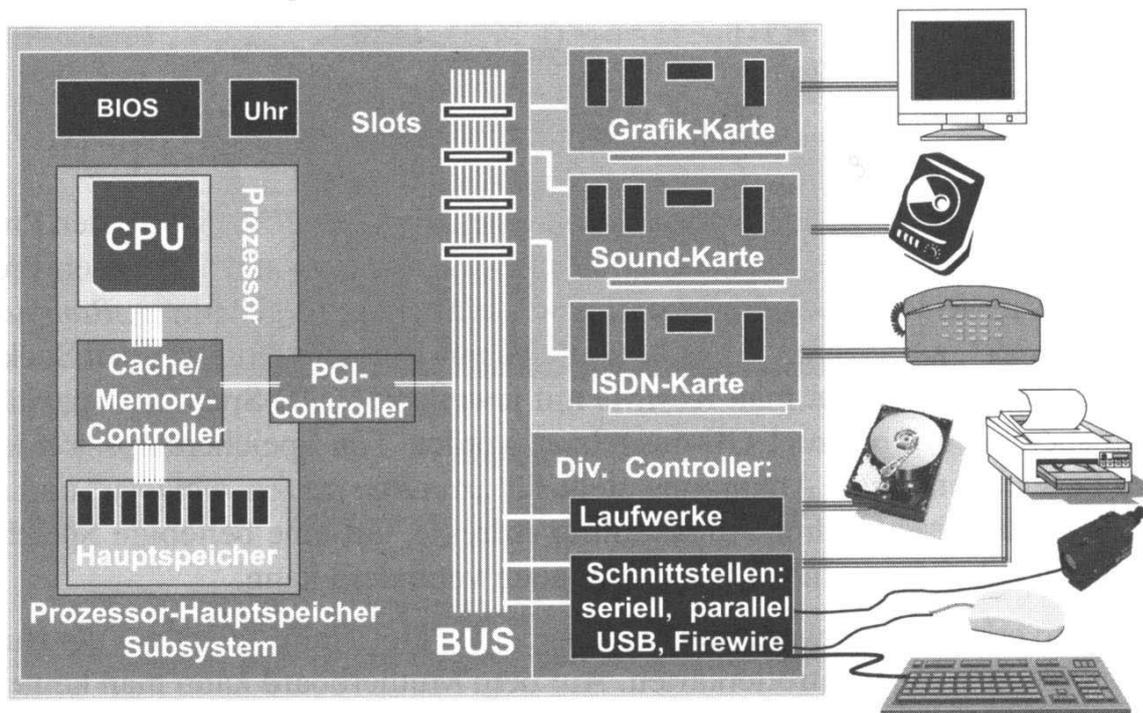
ALU: Ausführen der Befehle

Verknüpfung der Operanden: Arithmetische Verknüpfungen der Operanden
Schiebeoperationen (links, rechts)
logische Operationen („UND“, „ODER“)
Registermanipulationen

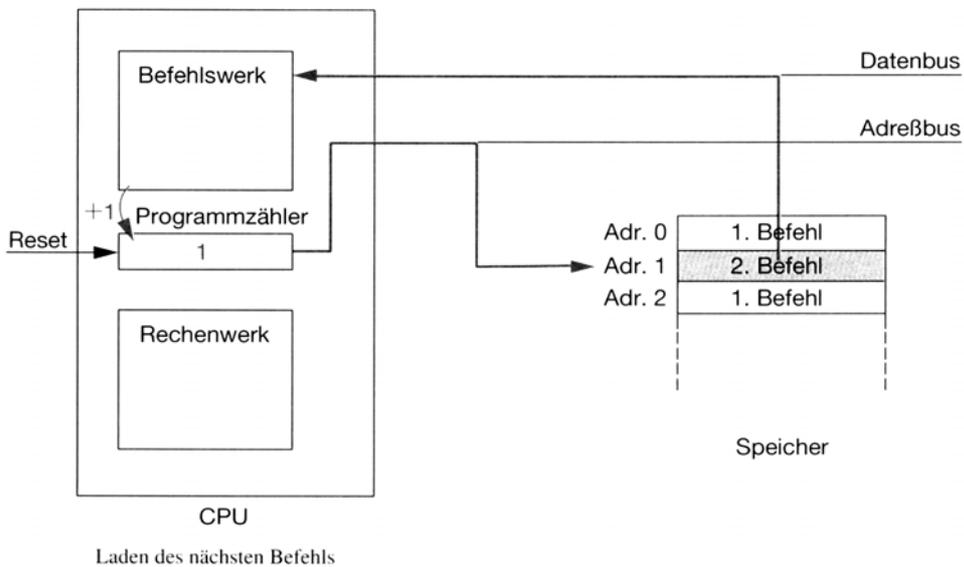
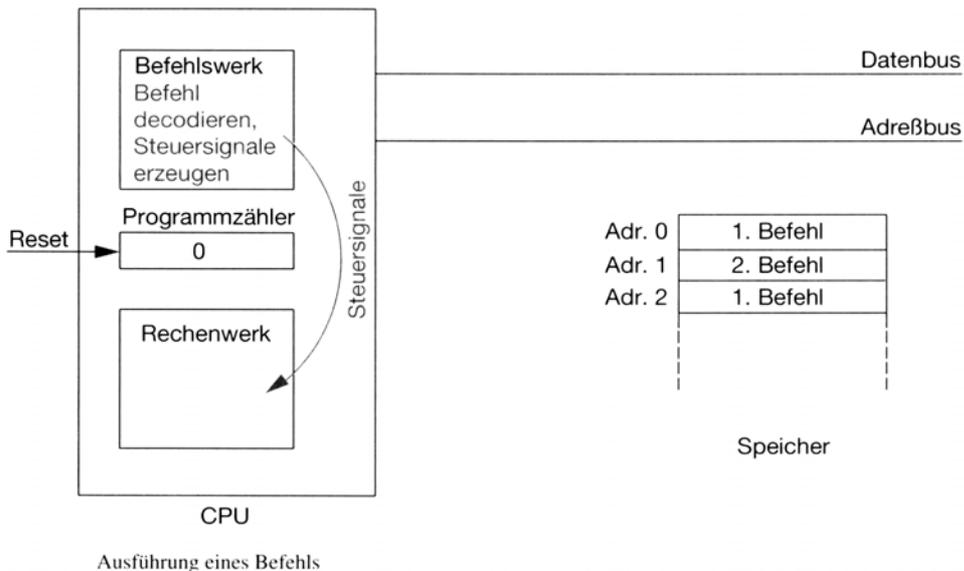
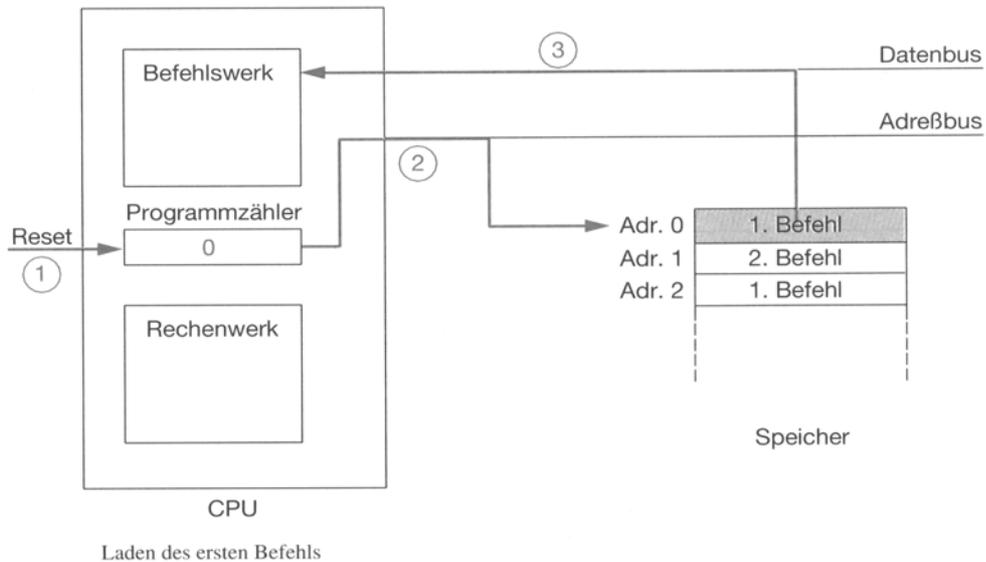
Statusregister: enthält die „FLAGS“ Zustände nach Operationen
Zero Ergebnis der Verknüpfung war 0
Sign -,- war negativ
Carry Übertrag ist aufgetreten

Teile des Steuerwerkes:

Befehlszähler:	Enthält die Adresse des nächsten auszuführenden Befehls
Befehlsregister:	enthält den auszuführenden Befehl
Befehlsdecoder:	Entschlüsselung des Befehles (Mikrobefehle, Mikroprogramm)
Stackpointer:	Kellerspeicher (LIFO last in first out) Dient häufig zur Abspeicherung von Rückkehradressen bei Unterprogrammaufrufen

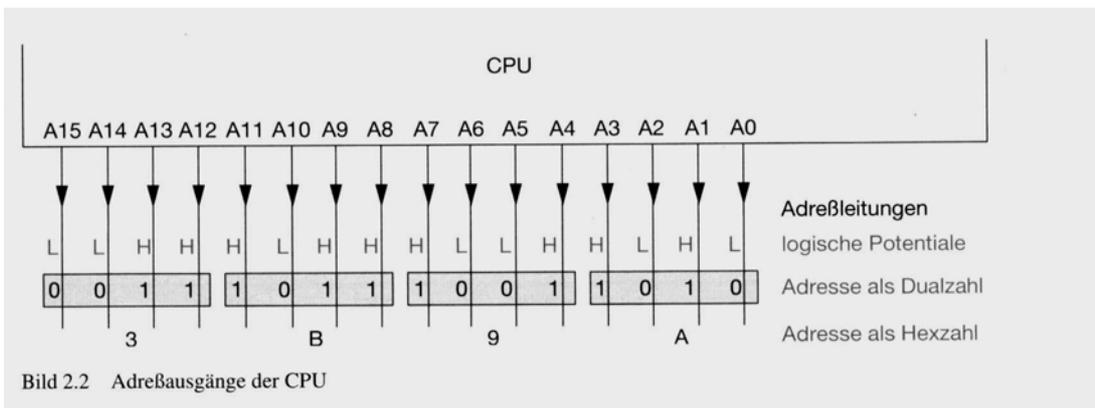


Überblick über die Komponenten eines Rechners



3. Der Speicher

Funktionseinheit eines Digitalrechners zur Aufbewahrung von Daten. Speicher lassen sich grob danach unterteilen, ob sie als interner Speicher Teil der Zentraleinheit eines Rechners sind (Arbeitsspeicher), oder als externer Speicher außerhalb der CPU und ihrer Umgebung liegen und als Massenspeicher dienen. Speicher können entweder als adressierbare Schreib- und Lesespeicher (RAM) bzw. Nur-Lese-Speicher (ROM) ausgeführt sein oder als Langzeitspeicher; wie z. B. Magnetbandkassette, Diskette (Floppydisk) oder Festplatte, bei denen die Daten auf den Träger geschrieben und bei Bedarf der Reihe nach wieder herunter gelesen werden können. Solche Langzeitspeicher sind stets externe Speicher. Dabei beträgt die Speicherkapazität eines Halbleiterspeichers typisch einige Mio. Byte, die von Festplatten bis zu einigen Mrd. Byte. Optische und magnetooptische Speicher, die mit Laser abgetastet (oder auch beschrieben) werden, können ebenfalls mehrere Mrd. Byte speichern. Neben der Speicherkapazität ist die Zugriffszeit ein wichtiges Speichermerkmal. Die Zugriffszeit ist die Zeit, die benötigt wird, um eine bestimmte Speicherzelle zu beschreiben oder zu lesen. Sie reicht vom Nanosekundenbereich bei Halbleiterspeichern bis in den Minutenbereich bei Magnetbändern.



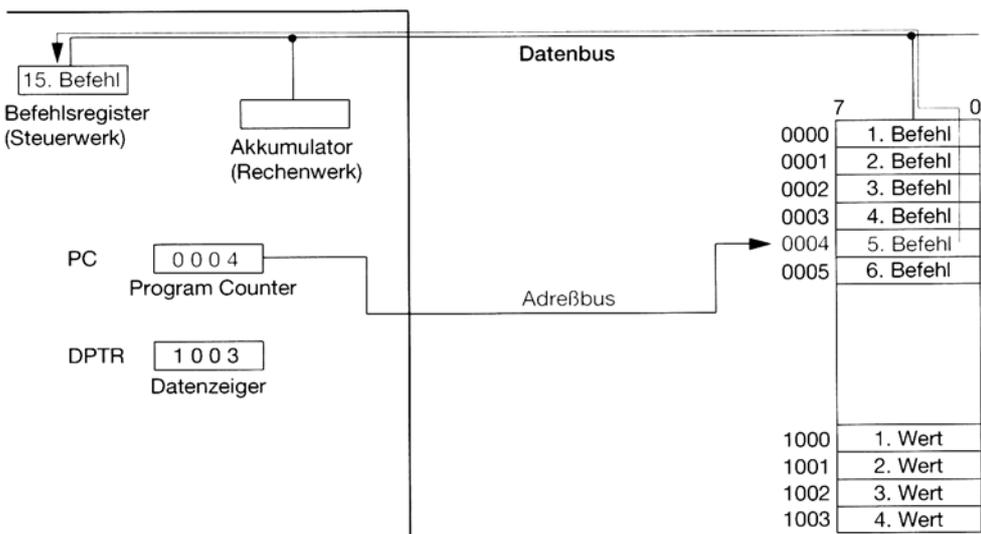
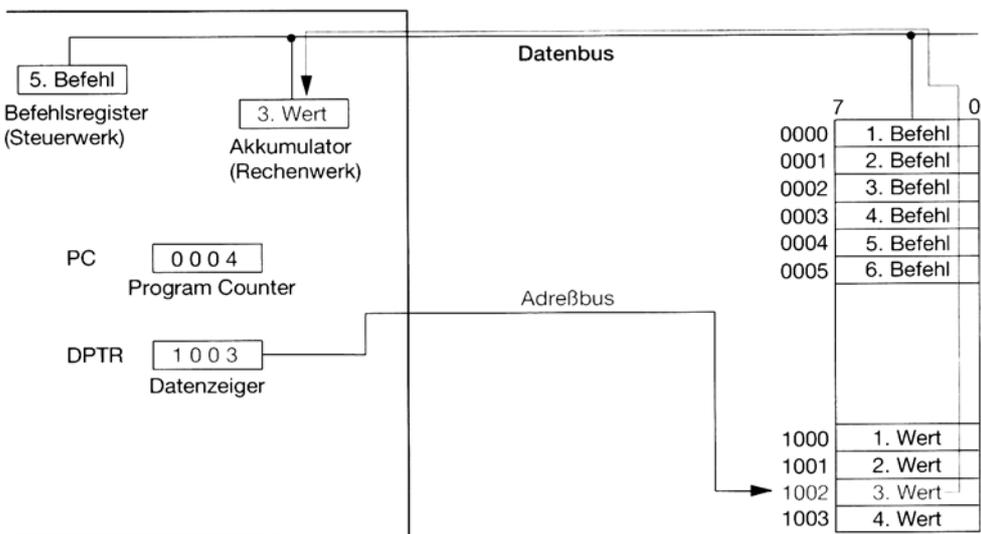


Bild 2.11 Program Counter



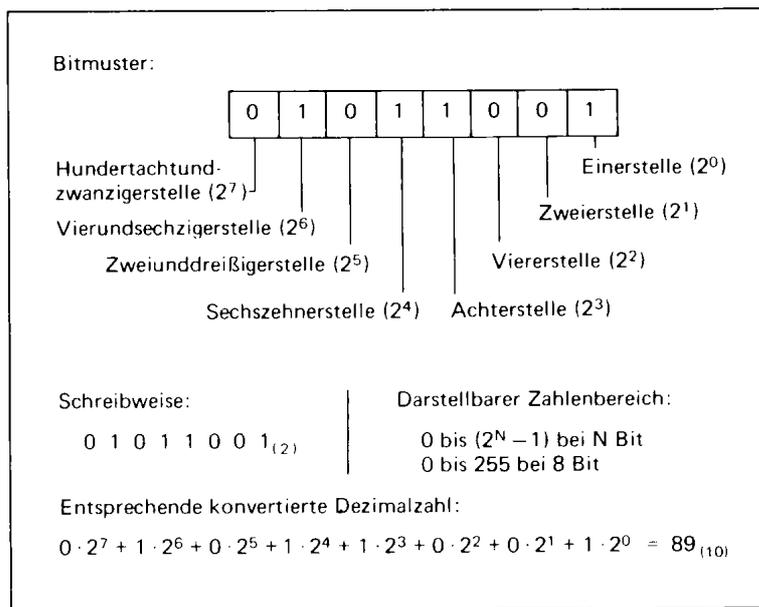
Datenzeiger

4. Interne Darstellung von Information

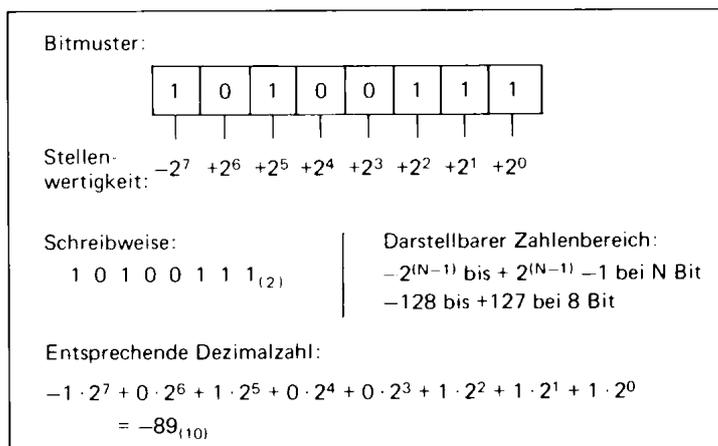
Bit: 0 oder 1

1 Byte = 8Bit $0_7 0_6 0_5 0_4 0_3 1_2 0_1 1_0$

Dualzahlen, Oktalzahlen, Sedizimal-(Hexadezimal-)Zahlen



Beispiel: Positive ganze Dualzahl mit 8 Bit Länge.

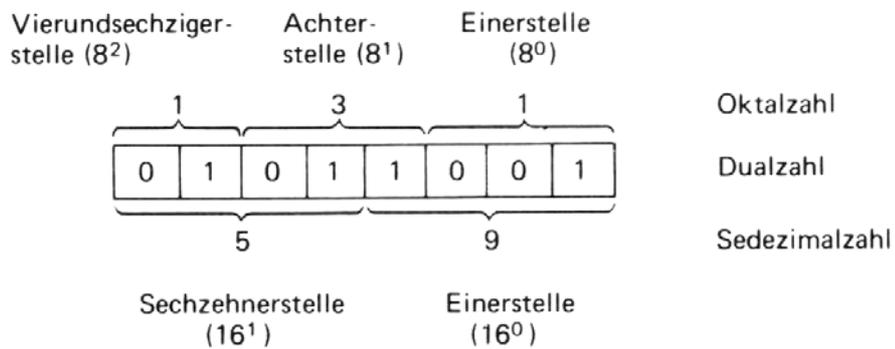


Beispiel: Negative ganze Dualzahl (8 Bit) im Zweierkomplement.

Zifferndarstellung der Dezimalzahlen von 0 bis 16

Dezimalzahl	Oktalzahl	Dualzahl	Sedezimalzahl
0	0	0	0
1	1	1	1
2	2	10	2
3	3	11	3
4	4	100	4
5	5	101	5
6	6	110	6
7	7	111	7
8	10	1000	8
9	11	1001	9
10	12	1010	A
11	13	1011	B
12	14	1100	C
13	15	1101	D
14	16	1110	E
15	17	1111	F
16	20	10000	10

Stellenwertigkeiten:



Schreibweise:

$131_{(8)}$

$59_{(16)}$

Wert:

$= 1 \cdot 8^2 + 3 \cdot 8^1 + 1 \cdot 8^0 = 89_{(10)}$

$= 5 \cdot 16^1 + 9 \cdot 16^0 = 89_{(10)}$

Zahlendarstellung im Oktal- und Sedezimalsystem.

Umwandlung in das Binär-, Oktal- oder Hexadezimalsystem

<u>z</u>	<u>z div 2</u>	<u>z mod 2</u>
2001	1000	1
1000	500	0
500	250	0
250	125	0
125	62	1
62	31	0
31	15	1
15	7	1
7	3	1
3	1	1
1	0	1

Abb. 1-2: Berechnung der Binärdarstellung einer natürlichen Zahl

Alles was über die Binärzahlen gesagt wurde, gilt sinngemäß auch in anderen Zahlensystemen. Verwandeln wir z.B. die dezimale Zahl 2001 in das Oktalsystem, so liefert fortgesetztes Teilen durch die Basiszahl 8 die Reste 1, 2, 7, 3. Genauer gilt:

$$\begin{aligned} 2001 &= 250 \times 8 + 1 \\ 250 &= 31 \times 8 + 2 \\ 31 &= 3 \times 8 + 7 \\ 3 &= 0 \times 8 + 3 \end{aligned}$$

Wir lesen an den Resten die Oktaldarstellung 3721 ab. Ganz entsprechend erhalten wir die Hexadezimaldarstellung durch fortgesetztes Teilen durch 16. Die möglichen Reste 0 bis 15 stellen wir jetzt durch die Hex-Ziffern 0 ... 9, A ... F dar und erhalten

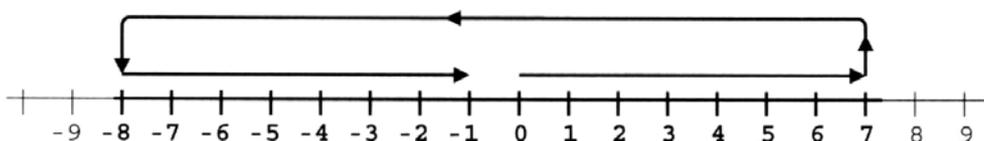
$$\begin{aligned} 2001 &= 125 \times 16 + 1 \\ 125 &= 7 \times 16 + D \\ 7 &= 0 \times 16 + 7, \end{aligned}$$

also die Hex-Darstellung 7D1. Zur Sicherheit überprüfen wir, ob beidesmal tatsächlich der gleiche Zahlenwert erhalten wurde, indem wir vom Oktalsystem in das Binär- und von dort ins Hexadezimalsystem umrechnen:

$$(3721)_8 = (011\ 111\ 010\ 001)_2 = (0111\ 1101\ 0001)_2 = (7D1)_{16}.$$

Das Zweierkomplement

Mit 4 Bits kann man einen Bereich von $2^4 = 16$ ganzen Zahlen abdecken. Den Bereich kann man frei wählen, also z.B. die 16 Zahlen von -8 bis $+7$. Man zählt nun von 0 beginnend aufwärts, bis man die obere Grenze $+7$ erreicht, anschließend fährt man an der unteren Grenze -8 fort und zählt aufwärts, bis man die Zahl -1 erreicht hat.



Auf diese Weise erhält man folgende Zuordnung von Bitfolgen zu ganzen Zahlen:

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

Beispiel: Zweierkomplement von -6

- | | |
|---------------------------|---------|
| 1. Binärdarstellung von 6 | 0 1 1 0 |
| 2. bitweises Komplement: | 1 0 0 1 |
| 3. Addition von 1 | 1 0 1 0 |

ganze Zahlen:

Bereich	Format	Delphi	Java
$-128..127$	8 Bit	<i>Shortint</i>	<i>byte</i>
$-32768..32767$	16 Bit	<i>Integer</i>	<i>short</i>
$-2^{31}..2^{31}-1$	32 Bit	<i>Longint</i>	<i>int</i>
$-2^{63}..2^{63}-1$	64 Bit		<i>long</i>
0..255	8 Bit	<i>Byte</i>	
0..65535	16 Bit	<i>Word</i>	

Gleitkommazahlen:

Bit	Vor-zeichen	Exponent	Mantisse	gültige Dezimalstellen	Im Bereich	
					von	bis
32	1 Bit	8 Bit	23 Bit	~ 7	$\pm 1 \times 10^{-38}$	$\pm 3 \cdot 10^{38}$
64	1 Bit	11 Bit	52 Bit	~ 15	$\pm 1 \times 10^{-308}$	$\pm 1 \cdot 10^{308}$
80	1 Bit	15 Bit	64 Bit	~ 19	$\pm 1 \times 10^{-4932}$	$\pm 1 \cdot 10^{4932}$

5. Befehle, Daten, Kommandos

- Befehle

Befehl besteht aus Befehlscode (op-code) und aus Operanden

Operand kann sein: Adresse oder Datenwort (Zahl)

Maschinencode	Mnemotechnischer (symbolischer) Code	Erklärung
<p>Einzelwort-Befehl</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;">1 0 0 1 0 0 1 1</div> <p style="margin-left: 20px;">Befehlscode Adresse</p>	ADD E	Addiere den Inhalt des Registers E zum Inhalt des Akkumulators
<p>Zweiwort-Befehl</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;">1 1 0 0 0 1 1 0</div> <p style="margin-left: 20px;">Befehls- code</p> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-left: 20px;">0 0 0 0 0 1 0 0</div> <p style="margin-left: 20px;">Daten</p>	ADI 4	Addiere zum Inhalt des Akkumulators die Zahl 4
<p>Dreiwort-Befehl</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;">1 1 0 0 0 0 1 1</div> <p style="margin-left: 20px;">Befehls- code</p> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-left: 20px;">1 0 1 0 1 0 0 0</div> <p style="margin-left: 20px;">} Adresse</p> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-left: 20px;">0 0 0 0 0 1 0 1</div>	JMP 05A8 ₁₆	Springe auf die Adresse 05A8 ₁₆

Befehlsformat am Beispiel eines 8-Bit-Prozessors.

- Daten

Daten werden nicht als Befehle interpretiert

ASCII-Tabelle: (ursprünglich nur die ersten 7 Bit, (8.Bit-Kontrollbit))

000	NUL	033	!	066	B	099	c	132	ä	165	Ñ	198	ä	231	þ
001	Start Of Header	034	"	067	C	100	d	133	å	166	ª	199	Å	232	ƒ
002	Start Of Text	035	#	068	D	101	e	134	ä	167	º	200	Ľ	233	ú
003	End Of Text	036	\$	069	E	102	f	135	ç	168	¿	201	ƒ	234	Û
004	End Of Transmission	037	%	070	F	103	g	136	ê	169	®	202	⌚	235	Ü
005	Enquiry	038	&	071	G	104	h	137	ë	170	¬	203	ƒ	236	ý
006	Acknowledge	039		072	H	105	i	138	è	171	½	204	ƒ	237	Ÿ
007	Bell	040	(073	I	106	j	139	ï	172	¼	205	=	238	–
008	Backspace	041)	074	J	107	k	140	î	173	ı	206	ƒ	239	˘
009	Horizontal Tab	042	*	075	K	108	l	141	ì	174	«	207	×	240	-
010	Line Feed	043	+	076	L	109	m	142	Ä	175	»	208	ð	241	±
011	Vertical Tab	044	,	077	M	110	n	143	Å	176	∴	209	Ð	242	_
012	Form Feed	045	-	078	N	111	o	144	É	177	⋮	210	É	243	¼
013	Carriage Return	046	.	079	O	112	p	145	Ê	178	⋮	211	Ê	244	⌈
014	Shift Out	047	/	080	P	113	q	146	Ë	179		212	Ë	245	§
015	Shift In	048	0	081	Q	114	r	147	Ë	180		213	Ë	246	÷
016	Delete	049	1	082	R	115	s	148	ö	181	Á	214	Í	247	,
017	-- frei --	050	2	083	S	116	t	149	ò	182	À	215	Î	248	ª
018	-- frei --	051	3	084	T	117	u	150	ù	183	À	216	Ï	249	˘
019	-- frei --	052	4	085	U	118	v	151	ù	184	©	217	ƒ	250	.
020	-- frei --	053	5	086	V	119	w	152	ÿ	185	ƒ	218	ƒ	251	'
021	Negative Acknowledge	054	6	087	W	120	x	153	Ö	186		219	■	252	˘
022	Synchronous Idle	055	7	088	X	121	y	154	Û	187	ƒ	220	■	253	˘
023	End Of Transmission Block	056	8	089	Y	122	z	155	ø	188	ƒ	221	ı	254	■
024	Cancel	057	9	090	Z	123	{	156	£	189	ƒ	222	ı	255	
025	End Of Medium	058	:	091	[124		157	ø	190	¥	223	■		
026	Substitute	059	;	092	\	125	}	158	×	191	ƒ	224	Ó		
027	Escape	060	<	093]	126	~	159	f	192	ƒ	225	ß		
028	File Separator	061	=	094	^	127	o	160	á	193	⌚	226	Ö		
029	Group Separator	062	>	095	_	128	Ç	161	í	194	ƒ	227	Ö		
030	Record Separator	063	?	096	`	129	ü	162	ó	195	ƒ	228	ð		
031	Unit Separator	064	@	097	a	130	é	163	ú	196	–	229	Ö		
032		065	A	098	b	131	â	164	ñ	197	+	230	µ		

Die einfachen Datentypen in Java

Gruppen	Variablentyp	Speicherplatz	Wertebereich
Integrale	int	4 Byte	+/- 2 Milliarden
	short	2 Byte	+/- 32.767
	long	8 Byte	+/- 9,2 * 10 ¹⁸
Fließkomma	float	4 Byte	7 bis 8 Ziffern
	double	8 Byte	16 bis 17 Ziffern
Zeichen	char	2 Byte	65.536
Boolesche Werte	boolean	---	true und false

- Unicode

16 Bit Codierung (65536 Zeichen)

landesspezifische Zeichen (ä, ö, ü, kyrillische, arabische, japanische, tibetische,...)

ersten 128 – identisch mit ASCII-Code

- Kommandos

Betriebssystem muss geladen werden und meldet sich durch die Eingabeaufforderung (MSDOS/ UNIX/ LINUX)

C: \> rechnername/nutzer>

Windows meldet sich mit graph. Oberfläche

Danach sind Kommandos des Betriebssystems eingebbar.

Kommandos sind im allgemeinen Zeichenketten, die durch den Kommandointerpreter des BS ausgeführt werden.

Sie bestehen aus Kommandonamen und eventuellen Parametern (Optionen)