



DATEN CODIEREN THEORIE

Der erste Teil in M114 befasst sich mit der Codierung von Daten in der Form von:

- vorzeichenlosen Ganzzahlen (Unsigned Integer)
- vorzeichenbehafteten Ganzzahlen (Signed Integer)
- Fließkommazahlen (Float)
- alphanumerischen Codes (ASCII, UTF8/16)

Und wird optional ergänzt durch:

- IT-Grundlagen (Bit/Byte, Massvorsätze, logische Operatoren)
- Datenübertragung Parallel/Seriell, Taktsignal
- Speichergrössenberechnung
- Zusammengesetzte Codierung, Barcodes

Ausserdem lernen wir als wichtiges Werkzeug den Notepad++ und den HEX-Editor kennen.

Zuerst führt kein Weg an den wichtigen **Zahlensystemen** wie Binär (BIN), Dezimal (DEZ) und Hexadezimal (HEX) vorbei:

- BIN: **Binärsystem**, Zweiersystem, Dualsystem
Basis: 2
Zeichenvorrat: 0, 1
- DEZ: **Dezimalsystem**, Zehnersystem
Basis: 10
Zeichenvorrat: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- HEX: Hexadezimalsystem, Sechzehnersystem
Basis: 16
Zeichenvorrat: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A (10), B (11), C (12), D (13), E (14), F (15)
(Eine Hex-Ziffer entspricht einer vierstelligen Dualzahl oder 4 Bit)

1. Fragestellung: 16 Bit ergeben wie viele Kombinationen?
So berechnen: $\text{BitkombinationenAnzahl} = 2^{\text{BitstellenAnzahl}}$
Resultat: $2^{16} = 65'536$

2. Fragestellung: Bei einer Distanzmessung sind 1000 unterscheidbare Kombinationen von z.B. 0mm bis 999mm verlangt.
So berechnen: $\text{BitAnzahl} = \text{LOG}(\text{BitkombinationenAnzahl}) / \text{LOG} 2$
Das Ergebnis ist auf die nächsthöhere Ganzzahl aufzurunden.
Mit LOG meint man den Zehnerlogarithmus.

Resultat: $\text{BitAnzahl} = \text{LOG}(1000 / \text{LOG} 2)$
 $\text{BitAnzahl} = 3 / 0.301 = 9.966$
 $\text{BitAnzahl} = 10$
Kontrolle: $2^{10} = 1024$
(24 Kombinationen ergeben Redundanz. Ist aber unvermeidbar, weil 9 Bit nur 512 Kombinationen ergäben.)



Vorzeichenbehaftete Dezimalzahlen in Binärschreibweise:

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

Zweierkomplement bilden:

Variante 1

Negative Zahl erhält man:

- Betrag der negativen Zahl
- Betrag bitweise invertieren
- Resultat um 1 addieren

Variante 2

Negative Zahl erhält man

durch Wertigkeit -8 / 4 / 2 / 1

Beispiele: (-5) 1011
(-2) -1110
(-3) 1101
(-5) 1011
(+7) +0111
(+2) 0010
(+3) 0011
(+4) +0100
(+7) 0111
(-3) 1101
(+4) -0100
(-7) 1001

Binäres Rechnen und Datenüberlauf:

0 1 0 0 1 1 1 0 + 78

1 0 1 0 0 1 1 1 = 167

1 1 1 1 0 1 0 1 245

1 1 0 0 1 0 0 0 + 200

1 0 1 1 0 1 0 0 = 180

0 1 1 1 1 1 0 0 124 -> DATA-OVERFLOW!

0 + 0 = 0

0 + 1 = 1

1 + 1 = 10 (Übertrag von 1 auf die nächsthöhere Stelle)

1 + 1 + 1 = 11 (Übertrag von 1 auf die nächsthöhere Stelle)

Der Wertebereich vom Datentyp INTEGER:

Der Integer (int) ist aktuell eine 32 Bit-Ganzzahl. (Früher 16 Bit)

232 ergibt 4'294'967'296 Kombinationen.

- Vorzeichenlos/unsigned: 0 bis 4'294'967'295
- Vorzeichenbehaftet/signed: -2'147'483'648 bis +2'147'483'647

Gleitkommazahlen:

Die Norm IEEE 754 definiert Standarddarstellungen für binäre Gleitkommazahlen in Computern in unter anderem den beiden Grunddatenformate 32 Bit → Single Precision und 64 Bit → Double Precision. Eine Gleitkommazahl wird wie folgt dargestellt:

$x = v * m * b e$

v: Vorzeichen 1 Bit

m: Mantisse bei Single 23Bit, bei Double 52Bit

b: Basis bei normalisierten Gleitkommazahlen 2

e: Exponent bei Single 8Bit, bei Double 11Bit



Die Byte-Reihenfolge Big/Little-Endian:

Die Byte-Reihenfolge bezeichnet die Speicherorganisation für einfache Zahlenwerte (z.B. Integer) im Arbeitsspeicher. Wir begegnen diesem Thema z.B. bei UTF16.

- **Big-endian-Format** (Grossendig): Das höchstwertige Byte wird zuerst gespeichert, (an der kleinsten Speicheradresse). Die höchstwertige Komponente wird zuerst genannt. Bsp. Uhrzeit → Stunde:Minute:Sekunde.
Mikroprozessor: Das Motorola-Format steht für Big-Endian
Serielle Übertragung: Big-Endian-Byte-Reihenfolge → Das höchstwertige Bit eines Bytes wird zuerst übertragen. Bsp.: I²C
- **Little-endian-Format** (Kleinandig): Das kleinstwertige Byte wird an der Anfangsadresse gespeichert. Die kleinstwertige Komponente wird zuerst genannt. Bsp. Datum → Tag.Monat.Jahr.
Mikroprozessor: Das Intel-Format steht für Little-Endian.
Serielle Übertragung: Das niederwertigste Bit eines Bytes wird zuerst übertragen. Bsp.: RS-232

HEX-Editor und Notepad++

Machen sie sich mit diesen beiden Editoren vertraut:

Notepad++

Notepad++ ist ein freier Texteditor für Windows und kompatible Betriebssysteme und dem Standard-Texteditor von Windows eindeutig überlegen. Als Zeichensätze werden ASCII und verschiedene Unicode-Kodierungen unterstützt. Notepad++ findet man unter dem folgenden Link: <https://notepad-plus-plus.org/>

HEX-Editor HxD

Unter einem HEX-Editor versteht man ein Computerprogramm, mit dem sich die Bytes beliebiger Dateien als Folge von Hexadezimalzahlen darstellen und bearbeiten lassen. Der Hex-Editor stellt eine ausgewählte Datei so dar:

Adress- kolonne	Dateiinhalte (Eine Zeile entspricht 16 Byte)	ASCII- Darstellung
HelloWorld.txt x		
00000000	48 65 6C 6F 20 57 6F 72 6C 64 21 0D 0A 44 69	Hello World!..Di
00000010	65 73 20 69 73 74 20 65 69 6E 65 20 54 65 78 74	es ist eine Text
00000020	70 72 6F 62 65 20 66 C3 BC 72 20 65 69 6E 65 6E	probe für einen
00000030	20 48 65 78 2D 45 64 69 74 6F 72 2E 0D 0A 45 73	Hex-Editor...Es
00000040	20 68 61 6E 64 65 6C 74 20 73 69 63 68 20 68 69	handelt sich hi
00000050	65 72 20 75 6D 20 65 69 6E 20 41 53 43 49 49 2D	er um ein ASCII-
00000060	46 69 6C 65 2E 20 28 3D 54 65 78 74 64 61 74 65	File. (=Textdate
00000070	69 29 +	i)

6C entspricht hier dem Buchstaben l

Adresskolonne:	Adresse des ersten Bytes der entsprechenden Zeile in hexadezimaler Darstellung.
Dateiinhalte:	16 Daten-Bytes. Pro Byte zwei Hex-Ziffern.
ASCII:	Den Versuch die 16 Bytes als ASCII-Character darzustellen.

Im Internet findet man einige Online-HEX-Editoren wie z.B. diesen: <https://hexed.it/>
Wer es gerne lokal als Applikation mag, findet z.B. die HEX-Editor-App HxD unter dem folgenden Link: <https://mh-nexus.de/de/hxd/>



Alphanumerische Codes:



Der ASCII-Code (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange) ist in seiner Ursprungsversion eine **7-Bit-Zeichenkodierung** und wurde bereits damals eingesetzt, wo noch Textnachrichten per Fernschreiber (Telex) übermittelt wurden. Die druckbaren Zeichen umfassen das lateinische Alphabet in Gross- und Kleinschreibung, die zehn arabischen Ziffern sowie einige Interpunktionszeichen (Satzzeichen, Wortzeichen) und andere Sonderzeichen. Der Zeichenvorrat entspricht weitgehend dem einer Tastatur oder Schreibmaschine für die englische Sprache. Die nicht druckbaren Steuerzeichen enthalten Ausgabezeichen wie Zeilenvorschub oder Tabulator, Protokollzeichen wie Übertragungs-ende oder Bestätigung und Trennzeichen wie Datensatztrennzeichen.

Der 7Bit-ASCII-Code:

DEC	HEX	BIN	CHAR	BEZEICHNUNG	DEC	HEX	BIN	CHAR	BEZEICHNUNG
000	00	00000000	NUL	Null Character	032	20	00100000	SP	Space
001	01	00000001	SOH	Start of Header	033	21	00100001	!	Exclamation mark
002	02	00000010	STX	Start of Text	034	22	00100010	"	Double quote
003	03	00000011	ETX	End of Text	035	23	00100011	#	Number sign
004	04	00000100	EOT	End of Transmission	036	24	00100100	\$	Dollar sign
005	05	00000101	ENQ	Enquiry	037	25	00100101	%	Percent
006	06	00000110	ACK	Acknowledgment	038	26	00100110	&	Amperсанд
007	07	00000111	BEL	Bell	039	27	00100111	'	Single quote
008	08	00001000	BS	Backspace	040	28	00101000	(Left opening parenthesis
009	09	00001001	HT	Horizontal Tab	041	29	00101001)	Right closing parenthesis
010	0A	00001010	LF	Line Feed	042	2A	00101010	*	Asterisk
011	0B	00001011	VT	Vertical Tab	043	2B	00101011	+	Plus
012	0C	00001100	FF	Form Feed	044	2C	00101100	,	Comma
013	0D	00001101	CR	Carriage Return	045	2D	00101101	-	Minus or dash
014	0E	00001110	SO	Shift Out	046	2E	00101110	.	Dot
015	0F	00001111	SI	Shift In	047	2F	00101111	/	Forward slash
016	10	00010000	DLE	Data Link Escape	048	30	00110000	0	
017	11	00010001	DC1	XON Device Control 1	049	31	00110001	1	
018	12	00010010	DC2	Device Control 2	050	32	00110010	2	
019	13	00010011	DC3	XOFF Device Control 3	051	33	00110011	3	
020	14	00010100	DC4	Device Control 4	052	34	00110100	4	
021	15	00010101	NAK	Negative Acknowledgement	053	35	00110101	5	
022	16	00010110	SYN	Synchronous Idle	054	36	00110110	6	
023	17	00010111	ETB	End of Transmission Bloc	055	37	00110111	7	
024	18	00011000	CAN	Cancel	056	38	00111000	8	
025	19	00011001	EM	End of Medium	057	39	00111001	9	
026	1A	00011010	SUB	Substitute	058	3A	00111010	:	Colon
027	1B	00011011	ESC	Escape	059	3B	00111011	;	Semi-colon
028	1C	00011100	FS	File Separator	060	3C	00111100	<	Less than sign
029	1D	00011101	GS	Group Separator	061	3D	00111101	=	Equal sign
030	1E	00011110	RS	Request to Send Record Separator	062	3E	00111110	>	Greater than sign
031	1F	00011111	US	Unit Separator	063	3F	00111111	?	Question mark

DEC	HEX	BIN	CHAR	BEZEICHNUNG	DEC	HEX	BIN	CHAR	BEZEICHNUNG
064	40	01000000	@	AT symbol	096	60	01100000	`	
065	41	01000001	A		097	61	01100001	a	
066	42	01000010	B		098	62	01100010	b	
067	43	01000011	C		099	63	01100011	c	
068	44	01000100	D		100	64	01100100	d	
069	45	01000101	E		101	65	01100101	e	
070	46	01000110	F		102	66	01100110	f	
071	47	01000111	G		103	67	01100111	g	
072	48	01001000	H		104	68	01101000	h	
073	49	01001001	I		105	69	01101001	i	
074	4A	01001010	J		106	6A	01101010	j	
075	4B	01001011	K		107	6B	01101011	k	
076	4C	01001100	L		108	6C	01101100	l	
077	4D	01001101	M		109	6D	01101101	m	
078	4E	01001110	N		110	6E	01101110	n	
079	4F	01001111	O		111	6F	01101111	o	
080	50	01010000	P		112	70	01110000	p	
081	51	01010001	Q		113	71	01110001	q	
082	52	01010010	R		114	72	01110010	r	
083	53	01010011	S		115	73	01110011	s	
084	54	01010100	T		116	74	01110100	t	
085	55	01010101	U		117	75	01110101	u	
086	56	01010110	V		118	76	01110110	v	
087	57	01010111	W		119	77	01110111	w	
088	58	01011000	X		120	78	01111000	x	
089	59	01011001	Y		121	79	01111001	y	
090	5A	01011010	Z		122	7A	01111010	z	
091	5B	01011011	[Left opening bracket	123	7B	01111011	{	Left opening brace
092	5C	01011100	\	Back slash	124	7C	01111100		Vertical bar
093	5D	01011101]	Right closing bracket	125	7D	01111101	}	Right closing brace
094	5E	01011110	^	Caret circumflex	126	7E	01111110	~	Tilde
095	5F	01011111	_	Underscore	127	7F	01111111	DEL	Delete



ASCII-Erweiterung gemäss ISO 8859 auf 8Bit-Code:

ISO: International Organization for Standardization / Internationale Organisation für Normung

ASCII belegte ursprünglich 7 Bit pro Character (0...127) und wurde später um ein Bit auf 8 Bit (0...255) erweitert. Um den verschiedenen Sprachen gerecht zu werden, wurde der ISO-Standard 8859 definiert, der nun im zweiten Teil des ASCII-Zeichensatz (128...255) deren 16 Sprachzusätze unterscheidet.

Bsp.: ISO-Standard 8859-1 = Latin-1, Westeuropäisch oder ANSI-ASCII.

Der Unicode:

Der ASCII-Code mit seinen 256 Zeichen genügt den heutigen Anforderungen nicht mehr. Es fehlen z.B. die chinesischen Schriftzeichen oder wie wär's mit einem Violinechlüssel? Es muss ein umfangreicherer Zeichencode her, nämlich der Unicode.

- Ein Unicode kann max. 8 Byte lang sein (64 Bit): U+XXXX'XXXX, wobei Unicode V2.0 bisher erst 1'114'112 verschiedene Zeichen U+0000'0000 bis U+0010'FFFF nutzt.
- UTF ist die verbreitetste Unicode-Kodierungsform (UTF=Universal-Coded-Character-Set Transformation Format)
- UTF-8: Je nach Zeichen beträgt die Codelänge von UTF-8 ein bis vier Bytes. UTF-8 ist in den ersten 128 Zeichen (Indizes 0–127) deckungsgleich mit ASCII.
- UTF-16: Je nach Zeichen beträgt die Codelänge von UTF-16 zwei oder vier Bytes. Zusätzlich muss mit der Byte-Order-Mark BOM (=Bytereihenfolge) angegeben werden, ob nach BigEndian BE oder LittleEndian LE verfahren wird. (Bsp. Datum : BE wäre yyyy.mm.dd, LE wäre dd.mm.yyyy)
- UTF-32: Ein einzelnes Zeichen belegt immer exakt 32 Bit.
- Notation bei HTML: &#x<unicode>; (hexadezimale Notation des Unicodes)
- Eingabe bei Windows-Word: U+<unicode> gefolgt von der Tastenkombination Alt+C.
- Wenn das Unicode-Zeichen im gewählten Font-Satz (Arial, CourierNew etc.) nicht implementiert ist, wird auf dem Bildschirm oder am Drucker ein Stellvertreter-Zeichen oder ein Leerzeichen dargestellt.

Beispiele:

Zeichen	Unicode	Unicode (Binär)	UTF-8 (Binär)	UTF-8 (Hexadez.)	
Buchstabe y	U+0079	00000000 01111001	01111001	0x79	In diesem Bereich (128 Zeichen) entspricht UTF-8 genau dem ASCII-Code. Das höchste Bit ist 0, die restliche 7-Bit-Kombination ist das ASCII-Zeichen
Buchstabe ä	U+00E4	00000000 1110 0100	11000011 1010 0100	0xC3 0xA4	
Zeichen für eingetragene Marke ®	U+00AE	0000 0000 1010 1110	11000010 1010 1110	0xC2 0xAE	Das erste Byte enthält binär 11xxxxxx, die folgenden Bytes 10xxxxxx; die x stehen für die fortlaufende Bitkombination des Unicode-Zeichens. Die Anzahl der Einsen vor der höchsten 0 im ersten Byte ist die Anzahl der Bytes für das Zeichen.
Eurozeichen €	U+20AC	0010 0000 1010 1100	11100010 10000010 10101100	0xE2 0x82 0xAC	

ARJ

Hinweis: Notepad++ versteht nebst ANSI-ASCII auch UTF-8 und UTF-16 mit LE-BOM und BE-BOM.



Codevergleiche für das Wort «EURO»

Der ANSI-ASCII-Code für das €-Zeichen lautet **0x80** oder **1000'0000**

Der Unicode für das €-Zeichen lautet **U+20AC** oder **0010'0000 1010'1100**

EURO in ANSI-ASCII

€		U		R		O	
8	0	5	5	5	2	4	F
1000	0000	0101	0101	0101	0010	0100	1111

EURO in UTF-8

€						U		R		O	
E	2	8	2	A	C	5	5	5	2	4	F
1110	0010	1000	0010	1010	1100	0101	0101	0101	0010	0100	1111

EURO in UTF-16-BE

BE-16 €						U				R				O	
FE	FF	2	0	A	C	0	0	5	5	0	0	5	2	0	0
		0010	0000	1010	1100	0000	0000	0101	0101	0000	0000	0101	0010	0000	0000

EURO in UTF-16-LE

LE-16 €						U				R				O	
FF	FE	A	C	2	0	5	5	0	0	5	2	0	0	4	F
		1010	1100	0010	0000	0101	0101	0000	0000	0101	0010	0000	0000	0100	1111

Bemerkungen:

Bei UTF16 ist **FE-FF** die **Byte-Order-Mark**. Man beachte, dass bei UTF8 und UTF16 der Unicode für das €-Zeichen immer gleich lautet. Bei UTF8 kann der Code 1,2,3 oder 4 Byte lang werden. Die ersten drei Bits 111 beim UTF-8 Code weisen darauf hin, dass der Character durch 3 Bytes repräsentiert wird. Bei UTF16 sind es jeweils nur 2 oder 4 Byte, wobei heute fast nur 2-Bytes Code üblich sind, ausser man verwendet z.B. einen **Violinschlüssel**, der mit **U+1D11E** vier Bytes beanspruchen würde, wenn er vom System denn auch noch richtig dargestellt wird, was nicht immer der Fall ist. Ein weiterer Stolperstein ist die Byteorder. Da in unserem Fall U, R und O ASCII-kompatibel sind, also 7 bzw. 8Bit verwenden, bleiben die weiteren 8 Bit jeweils 0.

Tipp: Probieren sie es mit Notepad++ und www.hexed.it gleich selber aus!



Optionale Themen

Grundlagen, die sie kennen sollten:

Die kleinste Einheit in der Datenverarbeitung ist das Bit (**B**inary **d**igit), wobei gilt:

8 Bit = 1 Byte

16 Bit = 1 Word

Das kleine b steht als Abkürzung für Bit

Das grosse B steht als Abkürzung für Byte

LSB = "Least Significant Bit" oder das kleinstwertigste Bit

MSB = "Most Significant Bit" oder das höchstwertigste Bit

(Die Beschriftung der LSB- bzw. MSB-Leitung ist z.B. bei Parallelverbindungen wichtig, damit ein Stecker nicht falsch herum angeschlossen wird)

In der Technik, also auch der Informatik, werden Zahlen mit sogenannten Massvorsätzen versehen. Das Internationale Einheitensystem oder SI «**S**ystème international d'unités» ist das am weitesten verbreitete Einheitensystem für physikalische Grössen. Die durch das SI definierten Maßeinheiten nennt man SI-Einheiten, wobei gilt:

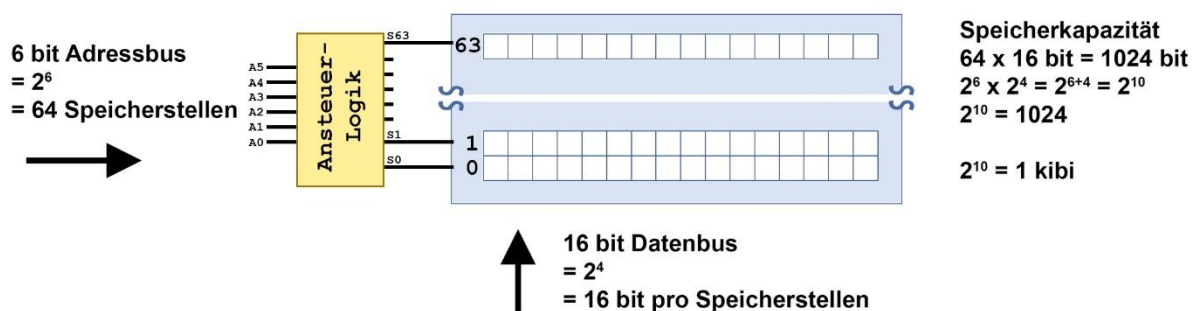
[T] → **Tera** → 10^{12} → 1'000'000'000'000 → **Billion**

[G] → **Giga** → 10^9 → 1'000'000'000 → **Milliarde**

[M] → **Mega** → 10^6 → 1'000'000 → **Million**

[k] → **kilo** → 10^3 → 1'000 → **Tausend**

Eine Ausnahme bilden die Kapazitätsangaben bei **Speichermedien**. Dort gelten die sogenannten IEC-Präfixe «International Electrotechnical Commission». Der Grund dafür liegt in der Besonderheit von Datenspeicher die binär adressiert werden und sich Speicherkapazitäten von 2^n Byte, d. h. Zweierpotenzen ergeben.



Es gilt:

[Ti] → **Tebi** → 2^{40} → 1'099'511'627'776

[Gi] → **Gibi** → 2^{30} → 1'073'741'824

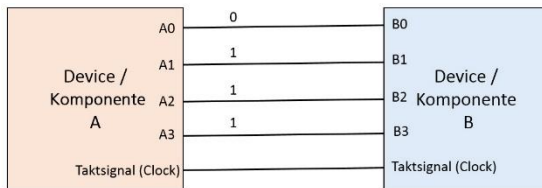
[Mi] → **Mebi** → 2^{20} → 1'048'576

[Ki] → **Kibi** → 2^{10} → 1'024

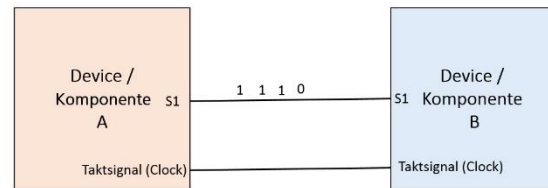


Datenübertragung

Eine Datenübertragung kann parallel oder seriell erfolgen:



Parallele Verbindung



Serielle Verbindung

- Eine **parallele Verbindung** zwischen zwei oder mehreren Komponenten nennt man Datenbus. Ein Codewort wird auf parallelen Leitungen auf einen Schlag bzw. Takt übertragen. Ist das Codewort z.B. 4 Bit breit, benötigt man 4 Leitungen.
Datenbus auf dem Mainboard: Verbindet CPU, RAM und I/O.
Adressbus auf dem Mainboard: Verbindet CPU, RAM und I/O.
SCSI (Small Computer System Interface): Verbindung und Datenübertragung zwischen Peripheriegeräten und Computern.
P-ATA (Parallel Advanced Technology Attachment): Paralleler Datentransfer zwischen Speichermedien bzw. Laufwerken und der entsprechenden Schnittstelle eines Computers.
- **Serielle Verbindung:** Um Leitungen einzusparen, kann ein Codewort auch seriell übertragen werden. Dann werden die Bit's nacheinander "auf den Weg geschickt". Der Takt ist jeweils der Startschuss für das "Loslaufen" des folgenden Bits. Um die selbe Performance wie bei der parallelen Datenübertragung zu erreichen, muss die Elektronik entsprechend schneller sein. Um zum Beispiel die gleiche Datenmenge einer 4-Bit-Parallelverbindung zu erreichen, muss die serielle Verbindung 4x schneller liefern.
S-ATA (Serial Advanced Technology Attachment): Serieller Datentransfer zwischen Speichermedien bzw. Laufwerken und der entsprechenden Schnittstelle eines Computers.
USB (Universal Serial Bus) für Drucker, Speicher-Sticks etc.
SAS (Serial Attached Small Computer System Interface): Die serielle Variante von SCSI.
- **Taktsignal:** Die Benutzung eines Taktsignals (Clock) ist ein Verfahren, den richtigen zeitlichen Ablauf beim Betrieb einer elektronischen Schaltung sicherzustellen und zwar sowohl bei parallelen als auch seriellen Verbindungen. Insbesondere benötigen viele digitale Schaltungen ein entsprechendes Signal zur zeitlichen Koordination bzw. Synchronisation. Meistens ist es ein periodisches Signal, das durch seine Frequenz Einheit Hertz, Hz) bzw. deren Kehrwert (Periodendauer, Einheit Sekunde) charakterisiert ist. Es wechselt dabei zwischen den zwei Logikpegeln High und Low. Hinweis: Es gibt auch Codierungen, sogenannte Leitungscodes wie z.B. der Manchestercode, wo das Taktsignal quasi in den Code «eingebaut» ist und somit keine separate Taktleitung mehr nötig ist. Solche Codeierungen werden z.B. bei Ethernet verwendet.



Datenspeicherung

Es wird zwischen nichtflüchtigem (permanentem) und flüchtigem Speicher unterschieden:

- **Nichtflüchtiger** oder permanenter **Speicher**: Dieser Speicher verliert seine Daten im stromlosen Zustand nicht.
Typische Vertreter: Magnet-Harddisk, SSD, USB-Speicherstick.
Man nennt solchen Speicher auch Sekundärspeicher.
- **Flüchtiger Speicher**: Dieser Speicher verliert seinen Inhalt, wenn er stromlos wird.
Die Technologie solcher Speicher lässt wesentlich höhere Datenraten zu, als bei nichtflüchtigem Speicher.
Typische Vertreter: Cache-Speicher in der CPU, RAM,
Man nennt den RAM-Speicher auch Primärspeicher.
Diese Speicher zeichnen sich darin aus, dass sie elektrisch bzw. verbindungstechnisch immer sehr nahe an der CPU liegen und von der CPU oft benötigte Daten sehr schnell liefern bzw. zwischenspeichern können. (Effizienz, Performance)

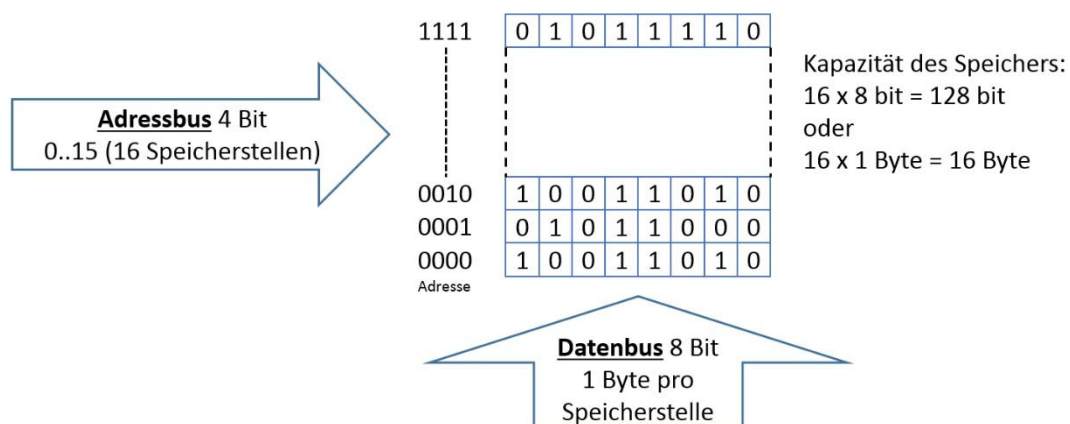
Und so wird auf flüchtigen Speicher, wie es der RAM-Speicher im PC darstellt zugegriffen:

Dazu vorerst eine Analogie aus der Bücherwelt mit einem Bücherarchiv:

Möchte man gerne seine archivierten Bücher wieder finden, muss man sich bei deren Ablage merken, wo man sie hinlegt. So ist es zum Beispiel sicher keine schlechte Idee, sich Regal- und Tablarnummer zu merken. Vielleicht sind die Bücher dann ja auch noch durchnummeriert. Gemeint ist selbstverständlich nicht die 12 bändige Micky-Maus-Best-Of-Sammlung sondern eine fast unüberschaubare Büchersammlung wie sie z.B. eine Universität besitzt.

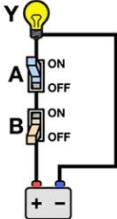
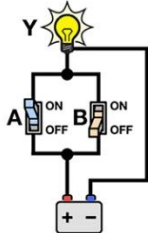
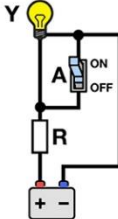
Wir unterscheiden also Ware (Daten) und Ablageort (Adresse).

Beim Computer ist die Problemstellung dieselbe: Die erzeugten und gespeicherten Daten wollen wieder gefunden werden. Dafür verwendet man einen Speicher-Chip, mit vielen "Speichernischen". Jede "Speichernische" wird über eine eindeutige Adresse erreicht:





Als Informatiker und Programmierer bzw. Freund/in der **Kombinatorik** sind ihnen sicher die **logischen Operatoren AND, OR und NOT** bekannt. Hier eine kurze Repetition:

Bezeichnung	UND/AND &&	ODER/OR	NICHT/NOT/INVERTER !																																				
Schaltschema																																							
Wahrheits-tabelle	<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Y	0	0	0	0	1	0	1	0	0	1	1	1	<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	1	<table><tr><th>A</th><th>Y</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	Y	0	1	1	0
A	B	Y																																					
0	0	0																																					
0	1	0																																					
1	0	0																																					
1	1	1																																					
A	B	Y																																					
0	0	0																																					
0	1	1																																					
1	0	1																																					
1	1	1																																					
A	Y																																						
0	1																																						
1	0																																						
0=false 1=true																																							

Zusammengesetzte Codierung:

Mit zusammengesetzter Codierung ist ein Datensatz (Record, Tupel etc.) gemeint. Ein Datensatz ist eine Gruppe von inhaltlich zusammenhängenden und zu einem Objekt gehörenden Datenfeldern. (Z.B. Artikelnummer, Artikelname, Farbe, Länge, Breite etc.) Datensätze entsprechen einer logischen Struktur, die bei der Softwareentwicklung (Datenmodellierung) festgelegt wurde. In der Datenverarbeitung werden zu Datensätzen zusammengefasste Daten in Datenbanken oder in Dateien gespeichert. Sie sind Gegenstand der Verarbeitung von Computerprogrammen und werden von diesen erzeugt, gelesen, verändert und gelöscht.

Barcodes:

- **EAN-13:** Das Bildchen mit den verschiedenbreiten schwarzen und weissen Balken, wie man es heutzutage auf allen Food- und Non-Food-Artikeln antrifft, repräsentiert eine 13-stellige Zahl. Diese 13 Zahlen sind vom Produkthersteller oder einer Organisation weiter aufgeschlüsselt wie z.B. in Ländercode, Produktcode, Lotnummer, Datum, Prüfziffer etc.
- **QR-Code:** Der QR-Code ist im Gegensatz zum EAN-13-Code ein zweidimensionaler Code. Dank einer "eingebauten" Fehlerkorrektur können bis zu 30% der QR-Grafik beschädigt oder verschmutzt sein, ohne die Lesbarkeit zu beeinträchtigen (gilt für Fehlerkorrektur-Level H). Diese Eigenschaft wird oft zur Platzierung von Werbebildchen und Logos missbraucht. Der maximale Informationsgehalt (177x177 Elemente, max. Verlust von 7% der Daten bzw. Fehlerkorrektur-Level L) beträgt knapp 3kB. Das würde dann ca. 7000 Dezimalziffern oder 4300 alphanumerische Zeichen ergeben. Vorsicht: Weil der Inhalt eines QR-Codes nicht auf den ersten Blick ersichtlich ist, kann man einem Fake-Link auf schädliche Webseiten aufsitzen oder das Smartphone führt Malware aus.

