

Wir haben in dieser Stunde begonnen, uns mit dem realen Speichern von Daten zu befassen. Außerdem haben wir uns überlegt, wie ein Rechner Daten verarbeiten kann. Also wie werden Bilder addiert (geht ja in Programmen), wie werden Texte kopiert oder Zahlen addiert? John von Neumann war der Wegbereiter der heutigen Rechner. Er hat sich dazu viel bei der schon viele Jahrzehnte vorher entwickelten Booleschen Algebra anschauen können. Diese mathematische Teildisziplin beschäftigt sich mit Aussagen mit den Wahrheitswerten w und f (wahr und falsch). Ihr könnt euch schon denken, dass mit  $w=1$  und  $f=0$  ein Rechner ziemlich schnell durch so eine Logik durchsteigen kann und das tut er auch.

### Nachtrag zur letzten Stunde

bmp ist das unkomprimierte Format und jpg das komprimierte. Sorry! jpg versucht, größere, aber gleiche Strukturen zu komprimieren. Stell dir ein komplett blaues Bild vor. Da muss man in der Anleitung nicht für jede Pixel blau setzen... Schlauer wäre es, anzugeben, dass die Farbe Blau für alle Pixel gilt. Das macht jpg so gut es geht und zerlegt dafür ein Bild in Teilbilder. Gibt es aber keine großen Flächen gleicher Art, so scheitert jpg meist kläglich und das komprimierte Bild ist ziemlich verzerrt und klobig. Insbesondere bei Portraits ist das der Fall.

### Speichern von Daten

Physikalisch Daten zu speichern ist nicht so schwer. Man kann beispielsweise ein Magnetband haben, auf dem die Magnetisierung entweder von oben nach unten oder von unten nach oben geht. Fährt ein Testmagnet (Lesekopf) darüber, spürt er, welcher Zustand gerade eingenommen wird. Das ist das Grundprinzip der meisten Festplatten. Wir werden an dieser Stelle nicht auf weitere technische Details eingehen. Bei Bedarf schaut in das entsprechende Uniskript, das auf eurer Seite verlinkt ist.

Die Kapazität von Festplatten hat unglaublich zugenommen. Für einen Eindruck siehe die unten stehende Tabelle!

#### Überblick über die Speicherkapazitäten der verschiedenen Baugrößen [\[Bearbeiten\]](#)

Jahr	5,25"	3,5"	2,5"	1,8"	1,0"	0,85"	andere Größe	typ. Modell(e) mit hoher Kapazität	Quelle	Drehzahl
1956	5 MB	–	–	–	–	–	61 cm Durchmesser, 1 Tonne Gewicht	<a href="#">IBM Ramac 305</a>	–	3.600 min <sup>-1</sup> ; 8,8 KB/s
1962	ca. 25/28 MB	–	–	–	–	–	?	<a href="#">IBM Ramac 1301</a>	<a href="#">[6]</a>	1.800 min <sup>-1</sup> ; ca. 88 KB/s
1981	10 MB	–	–	–	–	–	–	<a href="#">Seagate ST-412</a> (Aus dem IBM PC XT)	–	
1988	360 MB	20 MB	–	–	–	–	–	<a href="#">Maxtor XT-4380E</a> (5,25") bzw. <a href="#">Fuji FK309-26</a>	–	
1990	676 MB	106 MB	–	–	–	–	–	<a href="#">Maxtor XT-8760E</a> (5,25") bzw. <a href="#">Conner CP3104</a>	–	
1992	2 GB	426 MB	120 MB	–	–	–	–	<a href="#">Digital (DEC) DSP-5200S</a> ("RZ73", 5,25"), <a href="#">Seagate ST1480A</a> (3,5") bzw. <a href="#">Conner CP2124</a> (2,5")	–	
1993		1,06 GB	–	–	–	–	–	<a href="#">Digital RZ26</a> (3,5")	–	
1994		2,1 GB	–	–	–	–	–	<a href="#">Digital RZ28</a> (3,5")	–	
1995	9,1 GB	1,6 GB	422 MB	–	–	–	–	<a href="#">Seagate ST410800N</a> (5,25" FH), <a href="#">Conner CFS1621A</a> (3,5") bzw. <a href="#">Conner CFL420A</a> (2,5")	–	
1997	12 GB	16,8 GB	4,8 GB	–	–	–	–	<a href="#">Quantum Bigfoot</a> (12 GB, 5,25"), Nov. 1997, <a href="#">IBM Deskstar 16GP</a> (3,5") bzw. <a href="#">Fujitsu MHH2048AT</a> (2,5")	<a href="#">[7]</a>	–

2005 #	500 GB	120 GB	60 GB	8 GB	6 GB	–	<a href="#">Hitachi Deskstar 7K500 (500 GB, 3,5")</a> , Juli 2005	[10]	–
2006 #	750 GB *	200 GB	80 GB	8 GB	#	–	<a href="#">Western Digital WD7500KS</a> , <a href="#">Seagate Barracuda 7200.10 750 GB</a> , u. a.	[11]	–
2007 #	1 TB *	320 GB *	160 GB	8 GB	#	–	<a href="#">Hitachi Deskstar 7K1000 (1000 GB, 3,5")</a> , Januar 2007	[12]	–
2008 #	1,5 TB *	500 GB *	250 GB	40 GB *	#	–	<a href="#">Seagate ST31500341AS (1500 GB, 3,5")</a> , Juli 2008 <a href="#">Samsung Spinpoint M6 HM500LI (500 GB, 2,5")</a> , Juni 2008 <a href="#">Toshiba MK2529GSG (250 GB, 1,8")</a> , September 2008 <a href="#">LaCie LF (40 GB, 1,0")</a> , Dezember 2008	[13] [14] [15] [16]	–
2009 #	2 TB *	1 TB *	250 GB	40 GB *	#	–	<a href="#">Western Digital Caviar Green WD20EADS (2000 GB, 3,5")</a> , Januar 2009, <a href="#">Seagate Barracuda LP ST32000542AS (2 TB, 3,5", 5.900 min<sup>-1</sup>)</a> <a href="#">Western Digital Scorpio Blue WD10TEVT (1000 GB, 2,5", Bauhöhe 12,5 mm)</a> , Juli 2009 sowie <a href="#">WD Caviar Black WD2001FASS</a> und <a href="#">RE4</a> (beide 2 TB, September 2009) <a href="#">Hitachi Deskstar 7K2000 (2000 GB, 3,5")</a> , August 2009	[17] [18] [19]	–

Die Tabellen sind aus Wikipedia.

## Verarbeiten von Daten

Wir haben versucht, zwei binäre Zahlen zu addieren. Das klappt auch ganz gut, allerdings muss man mit den Übertragungen ständig aufpassen. Der Rechner geht so vor. Er macht aus 0+0 wieder 0. Aus 0+1 macht er 1 und aus 1+1 macht er wieder 0, speichert aber in einer Übertragszahl eine 1 ab. 10001010 plus 00001001 so wird zu:

$$\begin{array}{r}
 10001010 \\
 + 00001001 \\
 \hline
 10000011
 \end{array}$$

mit dem Übertrag 00010000. Denn an der vierten Stelle gab es den 1+1-Fall und dann wird ein Übertrag an 5. Stelle notiert. Nun addiert die Maschine das Ergebnis 10000011 mit dem Übertrag 00010000 und erhält so:

$$\begin{array}{r}
 10000011 \\
 + 00010000 \\
 \hline
 10010011
 \end{array}$$

Das wars. Sieht ganz schlaun aus und ist es auch. Wären hier neue Überträge zu notieren, würde dieser Schritt wiederholt bis der Übertrag 00000000 ist. Das Ergebnis ist übrigens richtig, denn 10001010 ist 138 und 00001001 ist die Zahl 9. Die Summe wäre also 147 und das ist gerade 10010011.

## John von Neumann

John von Neumann hat sich zu der obigen Rechnung überlegt, dass das ziemlich genau der Booleschen Algebra entspricht. Dort werden Aussagen verknüpft. Eine falsche Aussage mit einer falschen Aussage verknüpft bleibt falsch. 0+0=0! Beispiel: „Ich bin 100 Jahre alt“ ist falsch und „Heidelberg ist keine Unistadt“ ist auch falsch. Die Aussage „Ich bin 100 und HD ist keine Unistadt“ ist dann erst recht falsch...

## **Boolesche Algebra**

Die Boolesche Algebra hat als Verknüpfung von zwei Aussagen A und B den das UND, das ODER und für eine einzelne Aussage das NICHT zur Verfügung. Damit lassen sich ziemlich viele Aussagen bilden und es gibt Regeln, wie man dann die Wahrheitswerte einer zusammengesetzten Aussage aus den Wahrheitswerten der Einzelaussagen gewinnen kann.