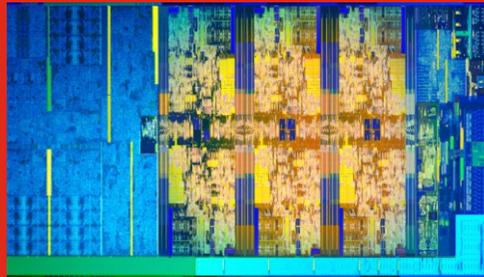


Wolf-Dieter Rase

# Vom Röhren-Flipflop zum 10-nm-Chip



Eine persönliche Geschichte der  
Digitaltechnik und Computergraphik

Wolf-Dieter Rase

# Vom Röhren-Flipflop zum 10-nm-Chip

Eine persönliche Geschichte der  
Digitaltechnik und Computergraphik

Bibliografische Information der Deutschen Nationalbibliothek: Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über [dnb.dnb.de](http://dnb.dnb.de) abrufbar.

© 2018

Herstellung und Verlag: BoD – Books on Demand, Norderstedt.

ISBN: 9783746065199

Satz und Umschlaggestaltung: Wolf-Dieter Rase

E-Mail: [wolf.rase@t-online.de](mailto:wolf.rase@t-online.de)

Die vorliegende Publikation ist urheberrechtlich geschützt, alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, in digitalen Formaten und elektronischen Systemen, bedarf der schriftlichen Zustimmung des Autors.

Es wird darauf hingewiesen, dass einige der im Buch verwendeten Bezeichnungen für Hardware und Software, Markennamen und Produktbezeichnungen dem warenzeichen-, marken- und patentrechtlichem Schutz unterliegen, auch wenn das im Einzelfall nicht besonders vermerkt ist.

Alle Informationen, Programme und Hyperlinks wurden nach bestem Wissen überprüft. Der Autor kann nicht für Schäden haftbar gemacht werden, die in Zusammenhang mit diesem Text und den Abbildungen stehen.

Die Titelseite zeigt ein Logikmodul mit einer Vakuumröhre aus der ersten Generation der Rechner-technik (Zuse Z22). Daneben ist eine Platine mit diskreten Transistoren und Dioden der zweiten Generation (Electrologica X1) abgebildet. Die dritte Generation ist repräsentiert durch das Layout des Mikroprozessors Intel Core i7-8700K mit sechs Kernen.

## **Verweise auf Literatur und Hyperlinks**

Die Verweise in den Kapiteln wurden in zwei verschiedenen Listen gesammelt. Publikationen finden sich im Verzeichnis mit der Literatur. Sie sind meistens daran zu erkennen, dass die Namen der Autoren mit KAPITÄLCHEN geschrieben sind, gefolgt vom Jahr der Veröffentlichung. Verweise auf Hyperlinks sind im Text mit dem Zeichen @ versehen. Im Hyperlink-Verzeichnis sind die Verweise mit den WWW-Adressen alphabetisch aufgelistet.

## **Abbildungen**

Die überwiegende Zahl der Abbildungen sind eigene Fotografien oder Graphiken. Für die anderen Abbildungen wurden nach Möglichkeit gemeinfreie Bilder verwendet. Bei älteren Geräten und/oder Firmen, die heute nicht mehr existieren, war es manchmal nicht möglich, den Urheber oder Rechteinhaber ausfindig zu machen. Ich bitte um Entschuldigung, wenn mir in dieser Hinsicht Fehler unterlaufen sein sollten.

## **Dank**

Ich danke unserer Tochter Susanne Wanasek für die Durchsicht des Textes auf Fehler in Rechtschreibung und Zeichensetzung

# Inhalt

<b>1</b>	<b>Ein Ferienjob bei IBM</b>	<b>1</b>
	Lochkartentechnik und Programmieren	1
	Unterschiedliche Länge der Innovationszyklen	2
	Der zwanghafte Programmierer	2
	Meine Geschichte der Digitalelektronik und Computergraphik	3
	Kartographische Visualisierung	5
	Weitere Quellen zur Geschichte der Rechnertechnik	5
<b>2</b>	<b>Entwicklung der Rechnertechnik</b>	<b>7</b>
	Binärzahlen und Boolesche Algebra	7
	1. Generation: Vakuumröhren	8
	2. Generation: Transistoren	9
	3. Generation: Integrierte Schaltungen	10
	4. Generation: Quantencomputer?	11
<b>3</b>	<b>Bergmannsakten im Alten Mann</b>	<b>13</b>
	Viel Papier und Kohlenstaub	13
	IBM 1401	14
	Hardware	14
	Magnetkern-Speicher	15
	Rechner-Architektur	16
	Peripheriegeräte	17
	Software-Ausstattung	17
	Programmierung und Programmtest	18
	Erweiterungen der 1401-Familie	19
	IBM 1440 mit Magnetplatten	19
	Olympischer Rechenbetrieb	20
	Datenfernverarbeitung	20
	IBM 1410, 1460	21
	Der wissenschaftliche Rechner IBM 1620	22
	IBM 1130	22
	Betriebsnahe Software	23
	Der erste Arbeitsplatz-Rechner von IBM?	24
<b>4</b>	<b>Datenverarbeitung an der Universität</b>	<b>25</b>
	Programmieren in Algol 60	25
	Rechner Zuse Z22	26
	Lehrveranstaltungen für Informationstechnik	27
	Computer Electrologica X1	29
	Hardware	29
	Programmierung	31
	Rechenbetrieb im Uni-Rechenzentrum	31
	2D-Graphik	32
	Datenverarbeitung für die Kristallographie	32
	Das Einkristall-Diffraktometer	33

	Kristallstruktur-Analyse mit Neutronenstrahlen	33
<b>5</b>	<b>Das Deutsche Rechenzentrum in Darmstadt</b>	<b>35</b>
	Neue Anwendungsgebiete erfordern mehr Rechenleistung	35
	Numerische Röntgen-Strukturanalyse	36
	Die Aufgaben des Deutschen Rechenzentrums	36
	IBM 7090/7094	38
	Software	39
	Rechenbetrieb im Deutschen Rechenzentrum	39
	Erhöhte Nachfrage nach Rechenleistung	40
	Das Ende einer Erfolgsstory	41
	Neue Aufgaben für das Deutsche Rechenzentrum?	42
<b>6</b>	<b>Studienabschluss</b>	<b>43</b>
	Diplomarbeit	43
	Der menschliche Interrupt	43
	Abschluss der Diplomarbeit am Deutschen Rechenzentrum	44
	Telefunken TR440	45
	Betriebssystem und Programmiersprachen	45
	Vermarktung	46
	Technische Zuverlässigkeit	47
<b>7</b>	<b>Aufbaustudium in Kanada</b>	<b>49</b>
	Simon Fraser University	49
	Die Rechnerfamilie IBM System/360	50
	Hauptmerkmale der /360-Architektur	51
	Unterschiede in den Instruktionssätzen	53
	Betriebssysteme	54
	Programmiersprachen	55
	Datenverarbeitung an der Simon Fraser University	56
	Das akademische Terminal	57
	Karten aus dem Zeilendrucker	58
	Computergesteuertes Zeichengerät	59
	Oberflächen-Darstellung	60
	Methode der schrägen Schnittflächen	60
	Simulierte Beleuchtung	61
	Isolinien mit Schatten	62
	Computergraphik an der University of British Columbia	63
<b>8</b>	<b>Seekarten aus dem Computer</b>	<b>65</b>
	Raymond Boyle und seine Arbeitsgruppe	65
	Der Rechner DEC PDP-8	66
	Auswirkungen auf meine berufliche Zukunft	68
<b>9</b>	<b>Elektronische Datenverarbeitung in der BfLR</b>	<b>69</b>
	Das Institut für Landeskunde	69
	Rechner Zuse Z25	70
	Das EDV-Konzept für die BfLR	71
	Der Rechner DEC PDP-11	71

Adressierungsmodi	72
Universalregister (R0 bis R5)	72
Stapelzeiger	73
Programmzähler (R7 bzw. PC)	73
Maschinenkode ist wartungsfeindlich	74
Die Modelle der PDP-11-Familie	75
Weitere Bus-Systeme	76
Betriebssysteme und Programmiersprachen	76
Laufende Raumbbeobachtung	76
Minicomputer DEC PDP-11/20	77
Farbige Karten aus dem Zeilendrucker	78
Zeichentische Calcomp 738 und 748	78
Sichtgerät Tektronix 4006	79
Digitalisiergerät	80
Probleme mit dem CAMAC-Anschluss	81
Digitalisierer ohne Servomechanik	81
Nutzung des akustischen Kanals	82
Vorlesen von Text	82
Sprach-Synthesizer MicroVox	83
Erkennung von Sprache	84
Vektorisierung macht Digitalisiergeräte überflüssig	84
Rechner PDP-11 Modell 45	85
Hardcopy-Geräte für 2D-Graphik	86
Unterstützung des Maschinendrucks	86
Ausgabegeräte für thematische Karten	87
Meine letzte Begegnung mit einer PDP-11	88
<b>10 Das Geo-Informationssystem der BfLR</b>	<b>89</b>
Geo-Informationssystem = Datenbasis + Software	89
Bestandteile eines Geo-Informationssystems	89
Numerische Datenbasis	90
Computerunterstützte Kartenzeichnung	91
<b>11 Die VAX-Familie</b>	<b>93</b>
PDP-11 mit 32-Bit-Adressierung?	93
Engpass PDP-11/45	93
DEC VAX-11/780	94
Hardware des Modells VAX-11/780	96
PDP-11 als Wartungsrechner	97
To confuse the Russians	97
Betriebssystem VAX/VMS	98
Programmiersprachen und Standard-Pakete	99
Weiterentwicklung der VAX/VMS-Familie	99
Nachfolger der VAX11/780 in der BfLR	100
Erhöhung der Rechenleistung	100
Mehrprozessor-Rechner	100
VAXcluster	101
VAX-Workstations	101

Von 32 Bit zu 64 Bit, von CISC zu RISC	102
DEC Alpha	103
VAX/VMS wird OpenVMS	104
DEC User Society	104
Zwanzig Jahre VAX/VMS	105
<b>12 Mikrochips und Mikrocomputer</b>	<b>107</b>
Computer-Chips	107
Intel 8008	107
Intel 8080	107
Zilog Z80	108
National Semiconductor SC/MP	109
Home-Computer MITS Altair 8800	110
Prozessor-Chip MOS Technology 6502	111
Ohio Scientific Challenger	111
PDA - Personal Digital Assistant	113
Arduino	114
Hardware	114
Schaltungs- und Platinen-Entwurf mit Fritzing	116
Programmierung	116
Raspberry Pi	116
Software-Unterstützung für Raspberry und Verwandte	118
<b>13 Arbeitsplatzrechner</b>	<b>119</b>
Der „Personal Computer“ von IBM	119
Prozessor-Familie x86 als Zentraleinheit des PC	120
Intel 8086	120
Intel 80386	121
Multiprozessor-Chips	121
Multithreading	122
Multilevel-Cache	123
Erweiterung der Adressierung auf 64 Bit	123
Typische Konfiguration der ersten PCs	123
Betriebssysteme für Arbeitsplatz-Computer	124
PC-DOS, MS-DOS	124
Graphische Benutzer-Oberfläche für MS-DOS	124
Vernetzung der Arbeitsplatzrechner	125
Vernachlässigte Kosten	126
Windows NT	127
Anwendungs-Software	128
Programmentwicklung	128
Visual Studio und Compiler für Fortran und C/C++	129
Graphik-Bibliothek	130
Meine privaten Arbeitsplatz-Rechner	130
Notebook-Rechner	131
PC-Nachfolger zuhause	132

<b>14 Hoch- und Höchstleistungsrechnen</b>	<b>133</b>
Großrechenanlagen	133
Nutzung freier Kapazitäten auf Arbeitsplatz-Rechnern	134
Höchstleistungs-Computer	135
Programmierung und Leistungsmessung	135
TOP500-Liste	136
<b>15 Software für GIS und Visualisierung im BBR</b>	<b>137</b>
Fusion zum Bundesamt für Bauwesen und Raumordnung	137
Neuorganisation der IT-Infrastruktur	137
Laufende Raumb Beobachtung und kartographische Visualisierung	138
Das Software-Paket Arc/INFO	139
Interpolation von kontinuierlichen Oberflächen	140
Interpolation und Visualisierung mit Surfer	141
Das Programm Konkar	141
Interpolation aus Punkten	142
Pyknophylaktische Interpolation	143
Portierung von Konkar auf Windows	145
Unregelmäßige Dreiecksnetze (TIN)	145
Qualitätsnetze mit dem Programm Triangle	146
Visualisierung mit Ray-Tracing-Verfahren (POV-Ray)	147
Brücken zwischen den Software-Paketen	148
<b>16 Virtuelle und reale 3D-Modelle</b>	<b>149</b>
Perspektivische Darstellungen	149
Stereogramme	149
Bildtrennung mit elektronisch gesteuerten Vorsatzbrillen	150
Stereogramme ohne elektronische Zusatzgeräte	150
Stereo-Betrachter	151
Verfahren mit Folienbrillen	151
Lentikularbilder	152
Autostereogramme	154
Techniken für die Betrachtung von Autostereogrammen	154
Stereogramm-Software	156
Reale 3D-Modelle	157
3D-Drucker für den Hausgebrauch	157
3D-Farbdrucker für Geo-Modelle	159
Kontur-Schichten aus Papier	159
Schichten aus Pulver mit Farbauftrag	159
Kartographische 3D-Modelle für die Regionalforschung	160
Erreichbarkeits-Oberfläche in 3D	160
Trend-Oberfläche mit Residuen	162
3D-Modelle für Outdoor-Enthusiasten	163
Neue Entwicklungen bei 3D-Farbdruckern	165
Nicht nur gucken, auch anfassen	165
Repräsentative Funktion und Konversationsobjekt	166

<b>17</b>	<b>Rechenhilfen, aber keine Computer</b>	<b>167</b>
	Logarithmentafel	167
	Rechenschieber	167
	„Griffeladdierer“ für Hexadezimal-Zahlen	169
	Der chinesische Abakus	169
	Japanischer Soroban	170
	Die Rechenmaschine Curta	171
	Produktion in Liechtenstein	171
	Die Curta als Hilfe beim Programmieren	172
	Taschenrechner mit trigonometrischen Funktionen	173
	Jubiläumsrechner HP-35s	174
	Eingabe von Formeln mit Klammern	174
<b>18</b>	<b>Literaturverzeichnis</b>	<b>175</b>
<b>19</b>	<b>Hyperlinks</b>	<b>181</b>

# Ein Ferienjob bei IBM

# 1

## Lochkartentechnik und Programmieren

Eigentlich wollte ich Lehrer für Sport, Biologie und Erdkunde werden, aber eine Menge alter Akten und der Rechner IBM 1401 haben es verhindert. Ende Juli 1964 fuhr ich zum Campus meiner Universität etwas außerhalb von Saarbrücken. Ich hoffte, auf dem Anschlagbrett des Studentenwerks ein Angebot für einen Job in der vorlesungsfreien Zeit zu finden. Meine Eltern erwarteten von mir, dass ich diese Zeit für eine bezahlte Tätigkeit nutzte, um ein wenig zum Familieneinkommen beizutragen. Vielleicht konnte ich auch durch den Ferienjob etwas Geld für mein Hobby, die Musik, abzweigen, für eine neues Instrument oder einen Verstärker zum Beispiel.

Am Mitteilungsbrett des Studentenwerks hing ein Zettel mit dem Briefkopf der Firma Internationale Büro-Maschinen, kurz IBM. Die Niederlassung in Saarbrücken suchte Studenten für eine zeitlich beschränkte Beschäftigung. Was das genau war, war nicht ausführlich beschrieben. Als zusätzlicher Anreiz für die Arbeitskräfte wurde eine Einführung in die Lochkartentechnik und ein Programmierkurs für den Rechner IBM 1401 angeboten. Als ehemaliger Schüler eines naturwissenschaftlichen Gymnasiums mit Interesse an der Elektronik hatte ich schon von Digitalrechnern gelesen, auch für welche Zwecke sie eingesetzt wurden. Zusammen mit dem Programmierkurs war das Angebot von IBM deshalb wesentlich attraktiver als die schwere körperliche Arbeit, die mich als Handlanger auf einer Baustelle erwartet hätte wie einige Jahre zuvor. Ich fuhr umgehend zur IBM-Niederlassung am Neumarkt in Saarbrücken. Ich wurde sofort eingestellt und fing zwei Tage später an, bei IBM zu arbeiten.

Ich konnte zu diesem Zeitpunkt nicht ahnen, wie dieser Ferienjob, an den ich rein zufällig geraten war, meinem Lebensweg eine neue Biegung verpasst hatte. Die Arbeit mit Computern und das Programmieren haben mich nicht wieder losgelassen. Ich war dann etwa drei Jahre für IBM als studentischer Programmierer tätig. Gegen Ende dieser Zeit gab mir mein Manager einen guten Rat, für den ich ihm heute noch dankbar bin: „Schließen Sie auf jeden Fall ihr Studium ab. Dann kommen Sie wieder zu mir, und wir reden über Ihren weiteren beruflichen Werdegang, vielleicht bei IBM“. Dazu ist es dann nicht gekommen, weil sich andere Wege für mich öffneten.

Bis heute faszinieren mich Computer und ihre Programmierung. Als die Computer kleiner, schneller und billiger wurden, war es möglich, einen persönlichen Arbeitsplatzrechner für zuhause zu erwerben. Ich betreibe heute die Erstellung von Computerprogrammen als Hobby. Die Programme sollen vorwiegend Probleme lösen, die mit Geo-Informationssystemen und rechnerunterstützter Kartographie

zusammenhängen. Ein Motiv ist der Ehrgeiz des Ingenieurs, noch bessere Lösungen für ein Problem zu finden. Eine weiterer Grund ist die wissenschaftliche Neugier für neue Methoden zur Visualisierung von räumlichen Zusammenhängen.

Wie für jede andere Liebhaberei fallen Kosten an, die man als Pensionär nicht einmal steuermindernd geltend machen kann. Wenn ich aber sehe, wie teuer zum Beispiel Lokomotiven und Steuerungselektronik für moderne Modelleisenbahnen sind, bin ich sicher, dass meine Lieblingsbeschäftigung im Vergleich zu anderen Hobbys nicht übermäßig kostspielig ist.

## Unterschiedliche Länge der Innovationszyklen

Bei der Diskussion von historischen Abfolgen in der Informationstechnik muss man immer im Gedächtnis behalten, dass die Innovationszyklen unterschiedlich lang sind und sich überlappen können.

- **Hardware:** Es dauert ein bis zwei Jahre, bis eine neue Prozessor-Generation mit höherer Geschwindigkeit und geringerem Stromverbrauch auf den Markt kommt. Die Einführung einer grundlegend neuen Rechnerarchitektur dauert länger, ganz abgesehen von der Einführung einer revolutionären Technik, vergleichbar mit dem Übergang von Vakuumröhren über diskrete Transistoren zu integrierten Schaltkreisen.
- **Software:** Fünf bis zehn Jahre vergehen, bis die Fortschritte, die durch die verbesserte Hardware möglich wurden, in die Betriebssoftware, optimierte Algorithmen und Anwendungen umgesetzt sind.
- **Personal:** Man schätzt, dass die Fortbildung und Erneuerung des Personals als Voraussetzung für die fachkundige Nutzung neuer Hardware und Software zehn bis zwanzig Jahre dauern kann.

Aufgrund der unterschiedlichen zeitlichen Dauer der Zyklen lassen sich die Verzögerungen in der Realisierung von Innovationen erklären. Bis integrierte Lösungen in den Standard-Paketen zur Verfügung stehen, muss man sich mit Brücken für den Übergang zwischen spezialisierten Programmen zufrieden geben, etwa für die Konvertierung von Dateiformaten und Steuerungsanweisungen.

## Der zwanghafte Programmierer

Das Programmieren von Computern hat ohne Zweifel Suchtpotential. Im englischen Sprachraum wird dieses Phänomen durch den Begriff „compulsory programmer“ ausgedrückt, annähernd übersetzt mit „zwanghafter Programmierer“. Die Geisteshaltung gleicht Viren, die man nach der ersten Infektion sein Leben lang mit sich herumträgt. Dieser Zustand ist aber nicht mit dem Verlangen eines Süchtigen zu vergleichen, der Gier nach dem angenehmen Gefühl, das eine Zigarette, Alkohol, Drogen oder

Glücksspiel mit sich bringen. Es ist eher eine Art sportliche Herausforderung. Für ein Problem wird die optimale Lösung gesucht, etwa durch Auswahl, Anpassung oder Neuentwicklung eines geeigneten Algorithmus, die Auswahl der passenden Datenstruktur und die Minimierung der Zugriffe auf externe Speichergeräte.

Das Optimum liegt meist im guten Zusammenwirken aller Faktoren. Wie so oft im Leben kann man das beste Ergebnis nur durch Ausprobieren finden. Das kostet Zeit und erhöht damit die Kosten für die Applikation, die sich nicht immer amortisieren lassen. Wenn diese Einschränkung nicht besteht, hat jeder gute Programmierer den Ehrgeiz, dem Optimum so nahe wie möglich zu kommen, unter Berücksichtigung der nicht veränderbaren Restriktionen. Wenn das Problem gelöst ist, hat man ein gutes Gefühl oder spürt sogar eine Art Euphorie. Dieses Verlangen nach Perfektion kann zur Sucht werden, die aber in der Regel durch die Kosten eingehegt wird.

Ein anderer Aspekt unterscheidet den zwanghaften Programmierer von einem Süchtigen in der üblichen Bedeutung: Mit der Programmierung von Computern kann man Geld verdienen. Auch sehr viel Geld, wenn man es richtig anpackt und Glück hat. Bill Gates wird in diesem Zusammenhang oft genannt, der es vom Studenten ohne Abschluss zum Multimilliardär gebracht hat. Auch die Entwickler von Facebook, Google und WhatsApp gehören zu diesem Kreis, weiterhin die Personen, deren Namen eng mit dem Aufstieg der Firmen Amazon, Ebay oder Paypal verbunden sind.

Es wird häufig vergessen, dass der geschäftliche Erfolg oft weniger in der Qualität der technischen Lösung an sich liegt, sondern im Erkennen der Relevanz und Zukunftsfähigkeit des Produkts oder der Marke, wie im Fall von Apple. Die Kompetenz der Gründer und Firmeninhaber für das Management und die Motivation der Mitarbeiter spielt sicher eine sehr wichtige Rolle, um die ursprüngliche Idee zum wirtschaftlichen Erfolg zu führen.

## Meine Geschichte der Digitalelektronik und Computergraphik

Seit dem Beginn meiner Programmierfähigkeit sind über fünfzig Jahre vergangen. In dieser Zeit habe ich mit vielen verschiedenen Computern gearbeitet oder bin mit ihnen in Berührung gekommen. Die Reihe reicht von Röhrenrechnern bis zu Computern mit höchstintegrierten Multiprozessor-Chips. Wie die Hardware hat sich die unterstützende Software weiterentwickelt, zum Beispiel die Betriebssysteme, Programmiersprachen und Anwendungsprogramme. Für Betrieb und Nutzung der Computer haben sich im Laufe der Zeit durch die Fortschritte in der Hardware und Software viele Änderungen ergeben.

Meine eigene Erfahrung mit Computern und elektronischer Datenverarbeitung ist ein Spiegel der allgemeinen Entwicklung in der Informationstechnik. Das betrifft die Schaltungstechnik, die Rechnerarchitekturen, Betriebssysteme, Compiler und die

wechselnden Paradigmen in der Softwaretechnik. Ich hielt es für eine gute Idee, einmal meine ganz persönliche Geschichte der Computertechnik aufzuschreiben. Das war auch ein nicht ganz zufälliges Hineinwachsen in die Anwendung von Informationstechnik in den raumbezogenen Wissenschaften. Dazu gehörte auch die Programmierung von Software für die frühen Vorläufer von Geo-Informationssystemen und die rechnergestützte Herstellung von thematischen Karten.

Die Beschreibung der manchmal trockenen technischen Einzelheiten wird durch Geschichten aufgelockert, die einen direkten Bezug zu meinen persönlichen Erfahrungen mit Computern und Programmierung haben. Deshalb wurde in vielen Fällen die Ich-Form benutzt, die in wissenschaftlichen Texten so weit wie möglich vermieden wird. Ich hoffe, dass diese Abweichung von der Konvention mehr Authentizität vermittelt. Bei dem einen oder anderen Leser, der schon längere Zeit mit Computern arbeitet, werden vielleicht Erinnerungen beim Lesen des Textes wach, gute wie vielleicht auch weniger gute. Die Angaben zur Anzahl der produzierten Rechner sind ein Hinweis auf die Entwicklung der Informationstechnik generell und die Verbreitung der beschriebenen Rechner und Rechnerfamilien in ihrer Zeit.

Mit den im folgenden Text erwähnten Computern und Geräten habe ich meistens selbst gearbeitet oder bin mit ihnen wenigstens näher in Berührung gekommen. Sie sind nur ein Ausschnitt aus dem umfangreichen Spektrum der digitalen Rechentechnik. Deshalb fehlen auch die Namen von Personen und Firmen, die eine wichtige Rolle in der Computertechnik seit den sechziger Jahren und früher gespielt haben. Das waren zum Beispiel Gene Amdahl, Seymour Cray oder Heinz Nixdorf mit ihren Firmen gleichen Namens. Die Hersteller Control Data Corporation (CDC), Scientific Data Systems (SDS) oder Sperry Rand (Univac) produzierten Computer, die in Anwendungsgebieten mit hohem Bedarf an Rechenleistung eingesetzt wurden. Weitere Firmen, zum Beispiel Bull, ICL, Honeywell, Fujitsu oder Hitachi waren nicht unbedeutende Hersteller von Rechenanlagen für die betriebliche Datenverarbeitung und wissenschaftliches Rechnen.

Die Hersteller von Peripheriegeräten waren wesentlich zahlreicher als die hier erwähnten Firmen. Manchmal haben die größeren Computerfirmen Peripheriegeräte vom Original-Hersteller bezogen, wenn der zu erwartende Absatz sehr gering war und sich eine eigene Fertigung nicht lohnte. Den Mehrpreis für das veränderte Firmenschild konnte man sich als Kunde sparen, wenn man das baugleiche Gerät direkt vom Hersteller kaufte. Neben der Verantwortung für den ordnungsgemäßen Anschluss bestand aber das Risiko, im Fall einer Störung oder eines Ausfalls keine Reparatur oder keinen Ersatz in angemessener Zeit zu erhalten.

Manchmal wurde auch überteuertes Zubehör verkauft. Ich erinnere mich an ein sogenanntes „Null-Modem“, das zwei Rechner über eine serielle Leitung verbinden sollte. Aus Neugier über die darin enthaltene Elektronik schraubte ich das Gehäuse

auf. Zu meiner Überraschung waren lediglich vier Steckkontakte über Kreuz mit Drähten verbunden, keine Spur von Elektronik. Der Materialwert betrug mit Sicherheit weniger als 50 DM, weit unter dem Kaufpreis von circa 700 DM.

In diesem Text werden nur Peripheriegeräte genannt, mit denen ich selbst gearbeitet habe. Einige Geräte gehörten zur Standard-Ausstattung der Computersysteme, sie müssen nicht besonders beschrieben werden. Andere Geräte waren entweder unbedeutend oder wurden nur für Experimente ohne weitere Auswirkungen auf die Anwendung eingesetzt.

## Kartographische Visualisierung

Eng verbunden mit der Nutzung von Computern in den Regionalwissenschaften und Geo-Informationssystemen ist die rechnergestützte Visualisierung der räumlichen Variablen und Strukturen. Für die Umsetzung der analogen geometrischen Grundlagen in digitale Form, den interaktiven Entwurf von Karten und die Präsentation auf Papier und anderen Medien sind graphische Geräte notwendig. Seit einigen Jahren werden die traditionellen Karten, in der Regel zweidimensionale Graphiken auf Papier und anderen Medien in Aufsichtsprjektion, durch echte 3D-Darstellungen ergänzt, zum Beispiel Stereogramme oder reale 3D-Modelle. Für deren Realisierung sind neue elektronische Geräte entwickelt worden.

Bei der Computergraphik habe ich mich auf die Gebiete beschränkt, die in meinem Studium, für den Erwerb der akademischen Qualifikationen und die Graphik-Anwendungen in der beruflichen Tätigkeit eine Rolle gespielt haben. Das Geo-Informationssystem der BfLR und des Nachfolgers BBR und die dafür verwendeten Software-Werkzeuge werden in der angemessenen Kürze beschrieben.

Die Phasen meiner persönlichen Geschichte des Computers, so auch die Kapitel in diesem Text, sind nicht immer streng chronologisch geordnet und zeitlich voneinander abgegrenzt. Das gilt sowohl für meine akademische Ausbildung und der professionellen Tätigkeit als auch für die Entwicklung der Digitaltechnik und Mikroelektronik allgemein. Die Phasen überlappten sich hin und wieder und verliefen zum Teil parallel.

## Weitere Quellen zur Geschichte der Rechnertechnik

Die frühen Entwicklungen bis in die fünfziger Jahre des letzten Jahrhunderts beschreibt GOLDSTINE (1972). In seiner Autobiographie behandelt KONRAD ZUSE (2010) schwerpunktmäßig die von ihm und seinen Mitarbeitern entwickelten Rechner und die Entstehungsgeschichte seiner Firma. Die historische Übersicht von O'REAGAN (2008) zur Technik der Rechenmaschinen reicht von den Babyloniern bis ins 21. Jahrhundert. Ähnlich umfassend ist das Buch von MATIS (2002). Die Beschreibung

der Rechenmaschinen reicht von den ersten Rechenbrettern bis zum World Wide Web, mit abschließenden Gedanken über die gesellschaftlichen Auswirkungen der Informationstechnik.

Das Buch von CERUZZI (2016) ist eine kurze Geschichte des Computers seit den Anfängen in den dreißiger Jahren bis zu den neuesten Entwicklungen in Hardware, Software und der weltweiten Kommunikation. Der Text ist gut lesbar, geht aber wegen der Kürze nicht allzu sehr in die Tiefe. Der Titel des Buchs von LEITENBERGER (2014) ist etwas irreführend. Der Begriff PC wird sehr weit gefasst und beschränkt sich keineswegs auf den *Personal Computer* von IBM aus dem Jahr 1981 und der Zeit danach.

# Entwicklung der Rechnertechnik

# 2

## Binärzahlen und Boolesche Algebra

Der Titel des Buches bezieht sich auf die grundlegenden Schaltelemente, die im Laufe der Jahre zur Realisierung von digitalen Schaltungen für Computer eingesetzt wurden. Die folgenden technischen Erklärungen sollen die Entwicklung der Schaltungstechnik von digitalen Rechenanlagen im Zusammenhang erläutern. Bei der Beschreibung der realen Computersysteme in den folgenden Kapiteln geht vielleicht der Zusammenhang verloren.

Grundsätzlich basieren digitale Schaltungen auf dem binären Zahlensystem. Eine binäre Zahl kann nur zwei Zustände einnehmen: 0 oder 1, falsch oder wahr, *false* oder *true*. Die binären Zahlen werden mit der *Booleschen Algebra* verknüpft (@ Boolesche Algebra). Der Name geht auf George Boole und sein Logikkalkül von 1847 zurück. Mehrere Mathematiker entwickelten die Idee weiter zu den Regeln, die heute für die Verbindung von binären Zahlen mit den logischen Operatoren *und*, *oder* und *nicht* angewendet werden. Dazu kommen die mengentheoretischen Verknüpfungen *Durchschnitt*, *Vereinigung* und *Komplement*.

In einer elektronischen Schaltung werden die binären Zahlenwerte durch Spannungs-Niveaus repräsentiert. Dem Zahlenwert 0 entspricht meistens eine Spannung von 0 Volt (keine Spannung). Der Wert 1 ist zum Beispiel durch eine Spannung von 5 Volt repräsentiert. Die tatsächlich verwendete Spannungshöhe ist eine Sache der Vereinbarung. In hochintegrierten Mikroprozessoren werden geringere Spannungen als 5 Volt verwendet, die nach Bedarf auch dynamisch nach unten und oben verändert werden können. Damit sollen Überhitzungsprobleme vermieden werden, insbesondere bei wechselnder Last.

Die gesamte Logik eines Computer setzt sich aus wenigen grundsätzlichen Schaltungen zusammen. Die Grundschialtung kombiniert eingehende binäre Signale zu einem neuen Ausgangssignal. Die Boolesche Algebra ist die Grundlage für die Verknüpfung der binären Signalzustände. Das neue Signal wird in der nachfolgenden Schaltungsstufe weiterverarbeitet, zusammen mit anderen Signalen. Hauptelement jeder Grundschialtung ist ein elektronischer Schalter, der die Veränderung der Spannung von 0 nach 1 und umgekehrt vornimmt.

Der Übergang vom Niveau für 0 nach dem Niveau für 1 und umgekehrt muss schnell erfolgen. Das ist im streng genommen nicht realisierbar, weil An- und Abstieg auch bei hoher Schaltgeschwindigkeit eine Mindestzeit erfordern. Aufgrund der parasitären Kapazitäten und Induktivitäten in den Übertragungsleitungen weist das

Rechtecksignal häufig ein Unter- und Überschwingen auf. Die Abweichung von der idealen Rechteckform muss durch zusätzliche Bauelemente so gut wie möglich kompensiert werden, etwa durch Dioden, Widerstände oder Kondensatoren. Ein Schalter für die Änderung des Spannungsniveaus lässt mit unterschiedlichen Bauteilen realisieren.

## 1. Generation: Vakuumröhren

Ein digitaler Schalter kann rein mechanisch funktionieren, wie im ersten Rechner von Konrad Zuse, der Z1. In der Telefontechnik verwendete man elektromagnetische Relais, die auch in den frühen Computern eingesetzt wurden. Der Durchbruch in der Schaltgeschwindigkeit kam mit der Verwendung von Vakuumröhren als Schalter. Vakuumröhren werden seit dem frühen 20. Jahrhundert zur Verstärkung von hoch- und niederfrequenten Signalen eingesetzt, etwa in Rundfunk- und Fernsehempfängern und -sendern.

In einem röhrenförmigen Glasgehäuse, das nahezu luftleer ist, setzt ein glühender Heizfaden, die Kathode, Elektronen frei. Im Vakuum fließt ein Strom von Elektronen, verursacht durch den Spannungsunterschied zwischen Kathode und Anode. Bringt man zwischen Kathode und Anode weitere Elektroden (Gitter) an, lässt sich durch die Gitterspannung die Intensität des Elektronenstroms und damit die Verstärkung regeln. Die Heiz- und Steuerspannungen werden über Kontaktstifte zugeführt, die in einem Sockel mit Federklemmen gesteckt sind. Vakuumröhren arbeiten als elektronische Schalter, wenn durch eine entsprechende Spannung am Gitter der Elektronenstrom vollständig unterbunden oder durchgelassen wird.

Der größte Nachteil von Logikschaltungen mit Röhren ist die Hitzeentwicklung durch die hohe Temperatur der beheizten Kathode. Eine Vakuumröhre hat eine relativ kurze Lebensdauer aufgrund des Metallverlusts am Heizfaden. Der große Raumbedarf der Module ist bedingt durch die Hitzeentwicklung, damit die Luft zum Kühlen die Röhren gut erreichen kann. Das linke Bild auf der Titelseite zeigt ein Logik-Modul mit einer Röhre und Zusatz-Bauteilen für den Rechner Zuse Z22.

Vakuumröhren werden heute noch in hochpreisigen HiFi-Anlagen und Musik-Verstärkern verwendet, weil sie besser klingen sollen als Transistorverstärker. Auch Hochfrequenz-Sender mit sehr hoher Leistung sind mit entsprechend groß dimensionierten Röhren ausgestattet, die nur mit Kühlaggregaten und ausreichender Wärmeabfuhr betrieben werden können.

Rechenanlagen mit Vakuumröhren bezeichnet man als Computer der ersten Generation. Diese Rechner waren im Vergleich zu ihren Vorgängern, die noch mit elektromechanischen Schaltern (zum Beispiel Telefonrelais) arbeiteten, sehr schnell. Das Programm, der Arbeitsplan für die Rechenvorgänge, war im Hauptspeicher ab-

gelegt, zusammen mit den häufig benutzten Variablen. Zu dieser Generation der Computertechnik gehörten unter anderen die Rechner Zuse Z22, IBM 650 und IBM 701 bis 709.

## 2. Generation: Transistoren

Ein Transistor ist ein elektronisches Halbleiter-Bauelement zum Verstärken niedriger elektrischer Spannungen und Ströme. Transistoren sind aus Halbleitern wie Germanium oder Silizium aufgebaut, denen durch Dotierung mit Fremdatomen bestimmte elektrische Eigenschaften verliehen wurden. Der Begriff *Transistor* ist eine Kurzform von *transfer resistor*, also einem durch eine elektrische Spannung oder einen elektrischen Strom steuerbaren elektrischen Widerstand. Eine Reihe von Transistortypen für verschiedene Anwendungen sind in Gebrauch, etwa Bipolar-Transistor, Sperrschicht-Feldeffekttransistor, Metall-Oxid-Feldeffekttransistor und weiteren Spezialformen, zum Beispiel der lichtempfindliche Fototransistor (@Transistor). In Schaltdiagrammen sind die Typen durch unterschiedliche Symbole repräsentiert. Der geringe Raum- und Energiebedarf und die zunehmenden Transitfrequenzen führten dazu, dass die Vakuumröhren allmählich durch Transistoren ersetzt wurden.

Wie die Vakuumröhren können Transistoren nicht nur für das Verstärken analoger Signale und die Regelung von Spannungen und Strömen, sondern auch als digitale Schalter verwendet werden. Die ersten Digital-Schaltkreise enthielten diskrete Transistoren, die zusammen mit den notwendigen Bauteilen zur Signalformung – Dioden, Widerstände und Kondensatoren – auf einer Platine aufgelötet waren. Die Platinen bestanden aus isolierendem Material mit Leiterbahnen aus dünnem Kupfer, mit Lochösen zum Einstecken und Festlöten der Bauteile. Bei zweiseitigen Platinen kamen Durchkontaktierungen zum Verbinden von Ober- und Unterseite hinzu. Die Platinen konnten in Serie gefertigt werden, aber die Montage der Bauteile war meistens noch Handarbeit. Da die Transistoren weniger Hitze abgaben, war der Aufwand für die Klimatisierung eines Transistorrechners niedriger als für einen Röhrenrechner. Die zweite Generation der Digitaltechnik war entstanden.

Die Designer der Logikschaltungen für die Computer versuchten die Anzahl der Grundsaltungen auf wenige Typen zu reduzieren. Ein häufig genutztes Logikelement ist das NAND-Gatter, das mit anderen NAND-Gattern und weiteren Logikschaltungen kombiniert werden kann. NAND-Gatter lassen sich mit unterschiedlichen elektronischen Bauteilen realisieren, zum Beispiel als DTL (Dioden-Transistor-Logik), TTL (Transistor-Transistor-Logik), NMOS (negative metal oxide semiconductor) oder PMOS (positive metal oxide semiconductor).

Das Bild rechts oben auf der Titelseite zeigt eine Logik-Platine mit zwei NAND-Gattern, so jedenfalls meine Erinnerung. Die Transistoren im schwarzen Gehäuse sind Standard-Bauteile, wie sie auch von Elektronikbastlern für NF-Verstärker verwendet

wurden, so wie die Dioden im gläsernen Gehäuse. Das Modul war ein Bestandteil des Rechners *Electrologica X1*.

### 3. Generation: Integrierte Schaltungen

Man kann immer wieder lesen, dass der Sputnik-Schock, als die Sowjetunion 1957 den ersten künstlichen Satelliten in den Orbit brachte, der Auslöser für die verstärkten Anstrengungen der USA in der Digitalelektronik gewesen sei. Das ist nur zum Teil richtig. Damals war die Raumfahrt ein Teil der militärischen Entwicklungen im kalten Krieg. Für die Interkontinental-Raketen wurden schon früher sehr kleine und zuverlässige Computer benötigt. Insbesondere musste die Anzahl der Lötstellen und Kabelverbindungen radikal verringert werden, um mögliche Störungen unter widrigen Bedingungen zu minimieren, etwa die hohe Beschleunigung, die Temperaturdifferenzen und die Strahlenbelastung im Weltraum.

Die gerade entwickelten integrierten Schaltkreise boten dazu die Möglichkeit, den Raumbedarf der Platinen mit Einzeltransistoren erheblich zu verringern. Ein integrierter Schaltkreis (*integrated circuit*, IC) ist eine auf einem dünnen, einige Millimeter großen Plättchen aus Halbleiter-Material aufgebrachte elektronische Schaltung. Sie wird manchmal auch als Festkörperschaltkreis oder monolithischer Schaltkreis (*solid-state circuit*, *monolithic integrated circuit*) bezeichnet. Das Halbleiter-Plättchen ist meist zum Schutz und zur sicheren Montage in einem größeren Gehäuse mit einem standardisierten Format eingegossen. Ein IC enthält typischerweise eine Kombination von zahlreichen miteinander elektrisch verbundenen elektronischen Halbleiter-Bauelementen wie Transistoren, Dioden und anderen aktiven und passiven Komponenten. Die genormten Abmessungen der ICs ermöglichten die maschinelle Bestückung der Platinen, ein wichtiger Faktor für weitere Kostensenkungen in der Fertigung.

Die Fertigung der integrierten Schaltungen läuft fast vollkommen automatisiert ab, mit nur minimaler menschlicher Intervention. Die Kostensenkung für die Produktion der ICs machte die daraus aufgebauten Computer zunehmend preiswerter. Mit den Fortschritten in der Produktionstechnik wurden die Schaltelemente immer kleiner und die Schaltkreise dichter gepackt, bis komplette Prozessoren oder ganze Computer auf einen Chip passten. Die Verbindungsleitungen auf den Chips sind heute nur noch 14 Nanometer breit, mit fallender Tendenz in der Verdichtung für die weiter folgenden Stufen. Ein Nanometer (nm) ist der milliardste Teil eines Meters.

Die Strukturbreite ist aber nicht der alleinige Anhaltspunkt für die Packungsdichte und die Anzahl der Transistoren auf einem Chip. Von den bekannten Herstellern werden Dichten von 51 bis 100 Millionen Transistoren/mm<sup>2</sup> angegeben. Der letztere Wert entspricht ungefähr 30 Milliarden Transistoren auf der Fläche eines Fingernagels (BANCHY & LEBSACK 2017). Mikroprozessor-Chips mit Leitungsbreiten von 10 nm sind

für 2018 angekündigt. Für ICs mit regelmäßigen Strukturen, etwa Speicherbausteine, werden bald Breiten von 5 nm realisiert sein. Für die Verkleinerung der Schaltelemente auf der Chipfläche gibt es physikalische Grenzen, die aber erst in einigen Jahren erreicht sein werden. Der nächste Schritt zur Verkleinerung von Transistoren ist wahrscheinlich der geführte selbstorganisierende Aufbau von Schaltkreisen (*directed self-assembly*, DSA). Es wird noch einige Zeit dauern, bis das Konzept in die Praxis verwendbar ist (SAVAGE 2017).

Die fortschreitende Verkleinerung der Bauelemente hat zu einer Erhöhung der Rechengeschwindigkeit und der Anzahl der Prozessoren auf einem Chip geführt, verbunden mit der Verminderung des Stromverbrauchs und der Hitzeentwicklung. Die Geschwindigkeit des internen Arbeitsspeichers erhöhte sich mit der Verkleinerung der Bauelemente, ebenso die Geschwindigkeit und Kapazität der externen Langzeit-Speicher auf Halbleiter-Basis. Das Verhältnis von Rechenleistung zu Kosten wird sich auch in Zukunft weiter verbessern. Die wirtschaftlichen Vorteile sind die Voraussetzung für die Nutzung der Informationstechnik in neuen rechenintensiven Anwendungsfeldern.

Das Bild rechts unten auf der Titelseite zeigt das Layout des Prozessorchips Intel Core I7-8700, der 2017 in den Handel kam. Dieser Chip enthält sechs Prozessoren mit Hyperthreading, so dass insgesamt zwölf Befehlsstränge fast simultan abgearbeitet werden können.

## 4. Generation: Quantencomputer?

Etwa ab 1965 erreichte die dritte Generation der Computertechnik mit integrierten Schaltkreisen den Markt. Die Schaltkreise und Rechner wurden zwar immer schneller und kleiner, arbeiteten aber mit den gleichen technischen Konzepten der Halbleiter-Technik. Anstatt einer plötzlichen Umwälzung durch neue Schaltelemente wie bei den vorherigen Generationen hat sich die Technik der 3. Generation evolutionär weiterentwickelt. Einige Neuerungen in der Halbleiterphysik, die anfangs vielversprechend erschienen, haben sich als nicht realisierbar, unpraktisch oder unwirtschaftlich erwiesen. Sie haben die Hürde zur praktischen Anwendung nicht (oder noch nicht?) überwunden. Dazu gehören etwa der Josphson-Kontakt, der Memristor oder die Kohlenstoff-Nanoröhre.

Der *Quantencomputer* wird regelmäßig als der nächste revolutionäre Schritt in der Rechnertechnik angekündigt, also die Generation nach den integrierten Schaltkreisen (GU. 2017). Der Quantencomputer ist gegenwärtig ein überwiegend theoretisches Konzept. Im Unterschied zum herkömmlichen Digitalrechner arbeitet er nicht auf der Basis der klassischen Physik, sondern nutzt quantenmechanische Eigenschaften. Die Zahlen 0 und 1 werden durch unterschiedliche Zustände von Quanten-Bits (*Qubits*) kodiert.

Für Nicht-Physiker, die mit der Quantenmechanik nicht vertraut sind, ist die Wissenschaft um die Qubits ziemlich rätselhaft. Die Verarbeitung der Qubits erfolgt nach quantenmechanischen Prinzipien. Begriffe wie Zweizustands-Quantensystem, Superposition, Quantenverschränkung, Quantenpunkte und noch einige mehr sind sehr weit von unserer schulischen Ausbildung und allgemeinen Erfahrungswelt entfernt. Eine kurze Einführung zu Qubits und Quantencomputern findet man bei RAUNER (2018).

Es wird weiter an praktikablen Lösungen für Quantenchips geforscht (CHOI 2017). Der Aufwand für das Festhalten, Auslesen und die Verknüpfung von Qubits ist ziemlich hoch. Die Stichworte in diesem Zusammenhang sind Laserkühlung, Ionenfallen und Supraleitung mit sehr tiefen Temperaturen bis hinunter zu 20 MilliKelvin (STILLER 2017). Es gibt mehrere Vorschläge, wie ein Quantencomputer realisiert werden könnte. In kleinem Maßstab wurden einige dieser Konzepte im Labor erprobt und Quanten-Chips mit wenigen Qubits gebaut (MANDAU 2018). Sie sind für Experimente geeignet, aber noch nicht für den Bau eines ausreichend leistungsfähigen Computers für die routinemäßige Anwendung.

Der praktische Nutzen und die Wirtschaftlichkeit des Quanten-Konzepts sind gegenwärtig noch nicht abzusehen. Auf jeden Fall wird es noch einige Jahre dauern, bis Quanten-Computer das experimentelle Stadium verlassen haben und vielleicht mit den Hochleistungsrechnern in Halbleiter-Technik konkurrieren werden (@ Quantencomputer). Die Firma D-Wave liefert seit 2015 einen Quanten-Computer mit 2.000 QuBits. Die Experten sind sich nicht einig, ob dieser Rechner wirklich ein Quanten-Computer im Sinne der Definition ist oder eher ein konzeptionelles Modell zur Verdeutlichung des Prinzips.

Theoretische Studien lassen vermuten, dass mit Quanten-Computern bestimmte Probleme der Informatik um Größenordnungen schneller gelöst werden können als mit Halbleiter-basierten Rechnern. Das sind zum Beispiel der Zugriff auf sehr große Datenbanken und die Produktzerlegung großer Zahlen. Dies würde das mathematische Problem leichter lösbar machen, das die Grundlage für die Sicherheit von bestimmten kryptographischen Verfahren ist. Diese speziellen Fälle machen aber nur einen verschwindend geringen Anteil der IT-Anwendungen aus, die heute genutzt werden. Es scheint im Augenblick unwahrscheinlich, dass Quantencomputer die Rechner der 3. Generation in naher Zukunft ersetzen werden.

## Bergmannsakten im Alten Mann

# 3

### Viel Papier und Kohlenstaub

Anfang der sechziger Jahre wurden in einem stillgelegten Stollen der Saarbergwerke – einem „Alten Mann“ in der Sprache der Bergleute – Personalakten aus der Vorkriegszeit gefunden, die man im Krieg dort eingelagert und dann vergessen hatte. Die wieder aufgetauchten Akten müssen auf möglicherweise unbekannte Informationen überprüft werden. Ein Bergmann könnte aufgrund der dort vermerkten Lohnzahlungen und Tätigkeitsbezeichnungen Anspruch auf eine höhere Rente haben.

Die Saarknappschaft, der Versicherungsträger für die Bergleute, nahm den Fund zum Anlass, die gesamte Rentenberechnung und -verwaltung auf Elektronische Datenverarbeitung umzustellen. Die Fundstücke aus dem Stollen, etwa 120 000 Aktenordner, mussten ausgepackt und sortiert werden. Den Auftrag für die Aufbereitung der Akten, die Übernahme der relevanten Informationen auf Datenträger, die Prüfung der Ansprüche und die Berechnung der Renten erhielt die Niederlassung Saarbrücken der Internationalen Büromaschinen-Gesellschaft (IBM).

Die IBM-Niederlassung suchte dringend Arbeitskräfte für die vorbereitenden Arbeiten für die Übertragung der rentenwirksamen Inhalte auf Datenträger. Anfang August 1964 fand ich den Anschlag am Schwarzen Brett des Studentenwerks mit dem Angebot, in diesem Projekt mitzuarbeiten. Der Stundensatz war höher als auf der Baustelle, dem traditionellen Arbeitsplatz für Studenten in den Semesterferien. Bald war ich damit beschäftigt, die Informationen aus den Altakten probeweise in Strichcode-Formulare umzusetzen, die dann von einem der ersten elektronischen Belegleser auf Magnetband übertragen werden sollten. Wie sich herausstellte, war die Fehlerrate noch zu hoch. Es blieb bei der bewährten Lochkarte als Erfassungsmedium für die Knappschafts-Daten.

Als zusätzlichen Anreiz für die studentischen Hilfskräfte bot IBM eine Einführung in die Lochkartentechnik und einen Programmierkurs für den Rechner IBM 1401 an. Mitarbeiter von IBM präsentierten eine Stunde täglich die wesentlichen Fakten zu Lochkarten und die dafür verwendeten Geräte, etwa Locher, Sortier- und Tabelliermaschinen. Im Programmierkurs lernten wir, Arbeitsabläufe für den Rechner IBM 1401 zu programmieren. Die meisten der Hilfskräfte interessierten sich kaum für Lochkartentechnik und Programmieren, sie wollten lediglich Geld verdienen. Nur wenige nahmen am Programmierkurs bis zum Schluss teil. Einige der Finalisten wurden eingeladen, in ein richtiges Programmierprojekt zu wechseln. Das war attraktiver als in Akten mit Kohlenstaub zu wühlen, und ich nahm das Angebot an.

Wir Studenten wurden übrigens angehalten, die Abkürzung IBM nicht „ei-bi-em“ auszusprechen, sondern mit einem „i“ am Anfang. Man wollte damit betonen, dass IBM ein deutsches Unternehmen sei, mit Niederlassungen und Produktionsstätten in Deutschland und entsprechend vielen Arbeitnehmern. Ich habe erst einige Jahre später den Grund verstanden: Die deutsche Wissenschaftsbürokratie vergab Mittel für Rechnerbeschaffungen und Projektförderung vorzugsweise an deutsche Hersteller, obwohl deren Rechner in Technik und Leistung erheblich der amerikanischen Konkurrenz hinterherhinkten. Die Leidtragenden dieser Politik waren zahlreiche Wissenschaftler an Universitäten und Forschungseinrichtungen, die mit weniger leistungsfähigen und zuverlässigen Rechenanlagen, Betriebssystemen und Programmiersprachen vorlieb nehmen mussten.

## IBM 1401

Die Firma IBM hatte die Computerentwicklungen fast verschlafen. Das Geschäft mit der Lochkartentechnik, das die Firma monopolistisch beherrschte, warf so gute Gewinne ab, dass die Firmenleitung keinen Anlass sah, viel Geld in die Entwicklung von kleineren Computern zu investieren. Aber eines Tages waren die klassischen Tabelliermaschinen mit ihrem hohen Anteil an mechanischen Bauteilen zu teuer in der Herstellung geworden, wie auch die mechanischen Sortierer oder Lese-Stanzer für Lochkarten. So beschloss die Firma IBM Ende der fünfziger Jahre die Entwicklung eines kleinen Rechners mit der Bezeichnung IBM 1401, der die Lochkartenmaschinen ersetzen sollte. Der Computer war ein System der zweiten Generation, also aus Schaltungen mit Einzeltransistoren aufgebaut.

### Hardware

Die Basis-Konfiguration der IBM 1401 bestand aus einem oder zwei Rechnerschränken. Darin waren die Elektronikmodule und die Netzteile herausklappbar montiert, um die Fehlersuche und Wartung zu erleichtern (Abb. 3-1). Die primären Peripheriegeräte waren eine kombinierte Lese- und Stanzeinheit für Lochkarten und ein Zeilendrucker mit der damals beeindruckenden Leistung von 600 Zeilen/Minute mit 132 Zeichen pro Zeile. Die Lese-Station besaß fünf Ablagefächer, die vom Programm angesteuert werden konnten, um Lochkarten je nach ihrer Kennung, wenn gewünscht, getrennt abzulegen. Die Stanz-Station konnte sowohl Lochkarten lesen als auch stanzen, wenn auch mit geringerer Geschwindigkeit als die Lese-Station. Der Arbeitsspeicher hatte in der Grundausstattung eine Kapazität von 4 000 Zeichen. Mit einer Erweiterungseinheit, etwas größer als eine Waschmaschine, wurde die maximale Speicherkapazität von 16 000 Zeichen erreicht.

Nicht alle Instruktionen der IBM 1401 waren in der Grundkonfiguration serienmäßig vorhanden. Funktionen, die heute selbstverständlich sind, mussten zusätzlich ge-

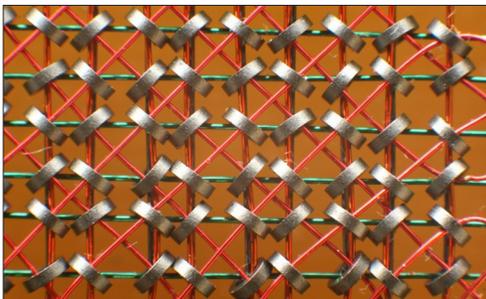


Abbildung 3-1: Die Zentraleinheit des Rechners IBM 1401 kombiniertem Lochkarten-Leser/Stanzer und Zeilendrucker (im Anschnitt links), Magnetband-Einheiten (rechts) und mit Speicherweiterung (hinten links)

11,6 Mikrosekunden. Für die Division von zwei Integer-Dezimalzahlen waren je nach Stellenanzahl bis zu 120 Taktschritte erforderlich, was ungefähr 1,4 Millisekunden entspricht.

### Magnetkern-Speicher

Der Arbeitsspeicher der IBM 1401 war ein Magnetkern-Speicher oder kurz Kernspeicher. An den Kreuzungspunkten eines rechteckigen Gitters aus lackisoliertem Kupferdraht waren kleine Ringkerne aus Ferrit aufgefädelt, deshalb der Name *Kernspeicher* für diese Technik (@Kernspeicher). Der Zustand der Magnetisierung legte fest, ob der Magnetkern eine binäre Null oder Eins repräsentierte. Sehr vereinfacht beschrieben, wurde der Magnetkern durch eine Spannung auf dem jeweiligen Zeilen- und Spaltendraht ausgewählt und magnetisiert. Der binäre Wert wurde mit dem



Lesedraht (hellrot) ausgelesen, der diagonal zum Gitter durch die Magnetringe geführt war (Abb. 3-2).

Das Auffädeln der Kerne auf die Drähte der Matrix und das Durchfädeln des Lesedrahts konnten nur teilweise von einer Maschine durchgeführt werden. Die

Abbildung 3-2: Magnetkern-Speichermatrix (Ausschnitt)

kaufte werden, etwa Befehle für die Multiplikation und Division oder der Unterprogramm-Sprung mit automatischer Rückkehr-Adresse. Der Programmierer musste also genau wissen, welche Befehle im Zielrechner vorhanden waren. Fehlte ein im Programm benutzter Befehl, wurde das Programm abgebrochen und eine Warnlampe leuchtete auf.

Der Rechner hatte keine Befehle für das Rechnen mit Gleitkomma-Zahlen. Dieser Datentyp war für die typischen Anwendungen auf der IBM 1401 nicht notwendig. Ein Taktschritt der zentralen Rechenheit dauerte

Montage erforderte zusätzliche manuelle Arbeitsschritte und viele Kontrollen. Kernspeicher war dadurch ziemlich kostspielig, ein Grund, warum viele Computer weniger Arbeitsspeicher enthielten als mögliche Maximum. Bei den ersten Magnetkern-Speichern hatten die Kerne noch einen Durchmesser von ca. 5 mm. Im Laufe der Zeit verringerten sich die Abmessungen bis auf 3 mm. Mit der Entwicklung von Halbleiter-Speichern, die weit preiswerter zu fertigen waren und weit geringeren Raum- und Energiebedarf hatten, ging die Zeit des Kernspeichers zu Ende.

### Rechner-Architektur

Die kleinste adressierbare Speichereinheit der IBM 1401 war ein Zeichen mit 7 Bit Länge. Eine Dezimalzahl wird mit vier Bit kodiert (BCD, *binary coded decimal*). Zwei weitere Bits wurden für die Kodierung von Alpha- und Sonderzeichen benötigt, allerdings nur Großbuchstaben wie bei einer Lochkarte, insgesamt 64 Zeichen. Das siebte Bit diente als *Wortmarke*, die die Einerstelle eines Feldes anzeigte. Mit der Wortmarke war es möglich, beliebig lange Dezimalzahlen aus aufeinander folgenden Dezimalziffern zu definieren. Ein Feld konnte auch ein Programmbefehl sein, denn die Befehle der IBM 1401 hatten unterschiedliche Längen.

Bei der Planung des Programmablaufs musste man darauf achten, dass das Ergebnisfeld ausreichend lang war. Bei einer Addition sollte das Ergebnisfeld mindestens ein Stelle größer sein als das größere der beiden Herkunftsfelder, um einen eventuellen Übertrag in der höchsten Stelle zu speichern. Bei einer Multiplikation muss das Ergebnisfeld so lang sein, dass das Produkt aus den Zahlen der beiden Herkunftsfelder hineinpasst. Für die typischen Anwendungen in Buchhaltung und Rechnungswesen genügte die rein dezimale Arithmetik. Bei der Umwandlung in binäre Zahlendarstellung für Rechenvorgänge und zurück zur Dezimaldarstellung können Rundungsfehler auftreten, die nicht immer der gewohnten kaufmännischen Rechenweise entsprechen. Gleitkomma-Operationen bei kleineren wissenschaftlichen Anwendungen wurden mit Unterprogrammen emuliert, was die Ausführung solcher Programme sehr langsam machte.

Mit sechs Bits war die Adressierung der Speicherstellen bis zum Maximum von 16 000 Zeichen möglich. Aufgrund der relativ geringen Speicherkapazität des Rechners versuchte man als Programmierer, die Anzahl der Stellen für eine Dezimalzahl so klein wie möglich zu halten. Diese Notwendigkeit zur sparsamen Verwendung von Speicherplatz hatte noch Auswirkungen bis zum Jahrtausendwechsel. Für die Speicherung eines Datums wurden nur die letzten beiden Stellen der Jahreszahl genutzt. Beim Übergang ins 21. Jahrhundert führte das zu Konflikten, weil jetzt vier Stellen für die Jahreszahl notwendig waren. Niemand hatte es für möglich gehalten, dass Programme – vor allem in der Programmiersprache Cobol – und gespeicherte Informationen vierzig Jahre überdauern könnten.

Die IBM 1401 war eine sogenannte Drei-Operanden-Maschine. Bei einer Addition wurde die Summe von zwei Feldern in ein drittes Feld geschrieben. Der Inhalt der Herkunftsfelder blieb erhalten. Die IBM 1401 hatte kein spezielles Rechenregister oder einen Akkumulator für die Ausführung des Rechengangs wie andere Rechner-Architekturen, die mit Zahlen konstanter Bitzahl (Worten) arbeiten. Für die Programmierung von Schleifen konnten Indexregister genutzt werden. Mit einem Befehl, der zusätzlich gekauft werden musste, war die Speicherung der Rückkehradresse vor einem Sprung in ein Unterprogramm möglich. Beim Verlassen des Unterprogramms wurde das rufende Programm an der nächsten Adresse nach dem Sprung fortgesetzt, ohne dass weiterer Programmieraufwand erforderlich war.

### Peripheriegeräte

An den Rechner konnten Magnetband-Laufwerke angeschlossen werden. Mit vier Laufwerken ließen sich größere Datenmengen verhältnismäßig schnell durch fortschreitendes Mischen in eine Sortierfolge bringen. Die mechanischen Sortiergeräte für Lochkarten, langsam und personalaufwendig, wurden damit obsolet. Mit der Zeit wurden weitere Zusatzgeräte für die IBM 1401 realisiert, etwa Geräte für strichkodierte Formulare und das Lesen von Dokumenten mit OCR-Schrift. Auch Zeichengeräte konnten über spezielle Elektronik an die IBM 1401 angeschlossen werden, etwa der Trommelplotter der Firma Calcomp.

### Software-Ausstattung

Die Rechanlage wurde wie vorher die Lochkarten-Geräte meistens an die Kunden vermietet, in selteneren Fällen auch verkauft. Die Anwendungsprogramme entstanden in der Regel beim Kunden nach dessen individuellen Bedürfnissen. Standard-Software für typische Geschäftsvorgänge wie Bestellungen, Auslieferung, Buchhaltung, Lohn- und Gehaltsabrechnung, Produktionsunterstützung, Statistik und andere Anwendungen in Handel und Fertigung gab es noch nicht. Die Kunden konnten entweder die Dienstleistungen von IBM für die Programmierung der Anwendungen in Anspruch zu nehmen oder die Programmierung durch eigene Mitarbeiter durchführen. Bei vielen Firmen war es üblich, sich die wichtigsten Programme von IBM bereitstellen zu lassen und nach und nach mit Eigenprogrammierung die Grundausrüstung der Geschäftssoftware zu erweitern.

IBM schickte auch Studenten wie mich zu ihren Kunden, um Programmieraufgaben zu erledigen. Mit solchen Ferienjobs habe ich einige Jahre gutes Geld bei IBM verdient, zuerst in der IBM-Niederlassung in Saarbrücken, im Jahr darauf in der Niederlassung Wiesbaden, bei der Firma Erdal in Mainz, und ein Jahr später beim Nähmaschinen-Hersteller Pfaff in Kaiserslautern, der heute nicht mehr existiert. Die Firmen nutzten alle das System IBM 1401 für die Datenverarbeitung in Verwaltung und Produktion.

## Programmierung und Programmtest

Die Programme für die IBM 1401 wurden in einem symbolischen Kode aufgeschrieben und von einem speziellen Programm, dem Übersetzer, in die internen Maschinenbefehle und Adressen umgewandelt. Der Begriff *Assembler* war noch nicht erfunden, der kam erst mit dem System IBM /360. Die mnemonische Kodierung der Maschinenbefehle war eine große Erleichterung für die Programmierer, denn die internen Befehle des Rechners waren ziemlich kryptisch. Die tatsächlichen Adressen der Speicherstellen konnte man sich schlecht merken oder man machte Fehler bei ihrer direkten Adressierung als Zahlen.

Zuerst wurde für die Kodierung ein festes Format mit Spalteneinteilung benutzt, das *Symbolische Programmiersystem*, kurz SPS. Auf SPS folgte die Kodiersprache *Autocoder*, für die kein festes Format vorgeschrieben war. Autocoder enthielt einige zusätzliche Funktionen, wie die Möglichkeit zum Einfügen von Makros für größere Programme mit sich wiederholenden Befehlsfolgen. Die symbolischen Befehle und Adressen wurde in den Maschinenkode übersetzt und das Objekt-Programm auf Lochkarten ausgestanzt.

Später gab es einen Compiler für eine der ersten Versionen von Fortran (*Formula Translator*). Für einfache Berechnungen war Fortran eine Erleichterung, da die Rechenvorgänge nahe an der mathematischen Formelsprache definiert werden konnten. Für größere Rechenarbeiten setzte die Architektur und Geschwindigkeit des Rechners IBM 1401 aber technische Grenzen, die auch mit Tricks nicht überwunden werden konnten.

Das „Betriebssystem“ der IBM 1401 bestand lediglich aus der Taste *Laden* am Lochkartenleser. Wenn man sie betätigte, wurden die ersten beiden Lochkarten eingelesen, die das Speichern des Anwendungsprogramms auf den folgenden Lochkarten in den Arbeitsspeicher veranlassten. Wenn dieser Vorgang beendet war, begann der Ablauf des Programms. Für Rechner mit Magnetband-Einheiten gab es die Programmbibliothek namens IOCS (*input/output control system*), mit der die Kennsätze der Magnetbänder in einem standardisierten Format geschrieben und gelesen werden konnten.

Eine erste Hilfe für die Fehlersuche in den Programmen war ein Schalter, mit dem man den Prozessor auf Einzelschritt stellen konnte. Der Binärkode des Speicherplatzes wurde dann mit Lämpchen angezeigt und konnte durch Eingabe über Kippschalter verändert werden. Ein Tastschalter diente dazu, den Arbeitsspeicher schrittweise in Zeilen von je hundert Zeichen auf dem Zeilendrucker auszudrucken. Die Fehlersuche konnte dann am Schreibtisch stattfinden, denn Maschinenzeit war teuer. In Anlehnung an die bekannte Samstagabend-Sendung „Jede Sekunde einen Schilling“ des österreichischen Fernsehens wurden die Kosten nach dem Merksatz

”Jede Sekunde einen Hering“ überschlagen. Nach den damals geltenden Preisen für Meeresfische kam man annähernd auf die Kosten, die externen Nutzern der IBM 1401 in Rechnung gestellt wurden.

Ein großer Fortschritt für den Programmierer war der *Debugger*, unter dessen Kontrolle das zu testende Programm ablief. Der Debugger druckte neben den ausgeführten Befehlen und Adressen die aktuellen Inhalte der Operanden aus. Das ergab zwar manchmal längere Listen, aber man kam damit Programmfehlern schneller auf die Spur als mit Einzelschritt-Schaltung oder dem einfachen Ausdruck der Speicherstellen. Der Debugger war ein Geheimtipp unter 1401-Programmierern. Wer einmal erfahren hatte, wie schnell man damit Programmfehlern auf die Spur kam, hat sich schnell eine Kopie angefertigt. Ein komfortabler Debugger ist heute Bestandteil fast aller Entwicklungsumgebungen, aus gutem Grund.

## Erweiterungen der 1401-Familie

Die unerwartet hohen Verkaufszahlen der ersten Maschinen und der damit erzielte Gewinn bewogen die Firma IBM, weitere Computer mit der annähernd gleichen Technik und Architektur auf den Markt zu bringen.

### IBM 1440 mit Magnetplatten

Mein Gruppenleiter während des Ferienjobs bei IBM in der Niederlassung Wiesbaden war ein EDV-Experte, der im Jahr vorher bei den Olympischen Winterspielen in Innsbruck zum Team von IBM gehört hatte. In seinem Zweitberuf war er geprüfter Berg- und Hochtourenführer und deshalb zusammen mit seinen Computerkenntnissen bestens für diese Aufgabe qualifiziert. Bei den Spielen von Innsbruck wurden zum ersten Mal Computer im größeren Maßstab für die schnelle Übermittlung von Informationen, für die Versorgung der Medien mit Ergebnislisten und für die Steuerung der Anzeigetafeln eingesetzt. Für die Realisierung dieser Aufgaben waren unter anderem Geräte für die schnelle Speicherung und den wahlfreien Zugriff auf die Daten notwendig. Magnetbänder waren für diese Anforderungen zu langsam. Deshalb wurde auf die kurz vorher zur Anwendungsreife gebrachten Magnetplatten mit beweglichen Schreib- und Leseköpfen zurückgegriffen.

Das grundsätzliche Arbeitsprinzip der Magnetplatten (Abkürzung HDD für *hard disk drive*) hat sich bis heute nicht verändert. Ein Stapel von Einzelplatten rotiert mit hoher Geschwindigkeit in einem Gehäuse. Auf jeder Seite einer Platte ist eine permanent magnetisierbare Schicht aufgebracht. Ein beweglicher Kamm mit Schreib- und Leseköpfen fährt in den Raum zwischen den einzelnen Platten des Stapels ein. Entlang radialer Spuren bringen die Schreibköpfe ein Muster von Magnetpunkten an. Mit den Leseköpfen werden diese Muster wieder ausgelesen und in binären Code zurückverwandelt. Der Direktzugriff auf die gespeicherten Daten ist wahlfrei, der

wichtigste Vorteil gegenüber einem Magnetband. Die Zeitverzögerung auf Grund der Rotation des Stapels und der Positionierung der Köpfe auf die gewünschte Spur ist vorhanden, liegt aber immer noch im akzeptablen Bereich.

Durch eine Reihe von technischen Verbesserungen – zum Beispiel Erhöhung der Umdrehungsgeschwindigkeit des Plattenstapels, schnellere Bewegung des Schreib/Lese-Kamms durch Gewichtsverringering, höhere Aufzeichnungsdichte, Reduzierung der Spurbreite, neue magnetische Materialien und Aufzeichnungstechniken, Füllung des Gehäuses mit Helium anstatt Luft – ist im Laufe der Jahre die Kapazität der Magnetplatten um Größenordnungen gestiegen. Heute sind 3½-Zoll-Festplatten mit 8 TeraByte Kapazität erhältlich, vielleicht bald HDDs mit 20 TeraByte (LABS 2018). In der Entwicklung befinden sich Magnetplatten, die mit Hilfe von Mikrowellen die Schreibdichte in der Magnetschicht noch weiter erhöhen. Für das Jahr 2019 werden Laufwerke mit 40 TeraByte erwartet (DI MARCOBERARDINO 2018).

### Olympischer Rechenbetrieb

Die Magnetplatten waren an den Rechner IBM 1440 angeschlossen, eine für den Betrieb von Magnetplatten optimierte Version der 1401. Um die Ausfallsicherheit während der Winterspiele zu gewährleisten, standen in der zweckentfremdeten Sporthalle der Universität Innsbruck zwei Rechner IBM 1440 nebeneinander, mit gemeinsamen Anschlüssen an redundante Magnetplatten. Im Fall einer Störung konnte der Betrieb manuell mit einer sehr kurzen Verzögerung vom einem auf den anderen Rechner umgeschaltet werden. Der Parallelbetrieb von zwei oder mehr eng gekoppelten Rechnern mit automatischer Ausfallsicherung (*failover*) waren noch nicht möglich.

Die Rechner IBM 1440 mit Magnetplatten wurden generell bei Anwendungen eingesetzt, für die sehr kurze Speicher- und Antwortzeiten notwendig waren, etwa für Auskunftssysteme mit kurzen Antwortzeiten oder Buchungssysteme in verhältnismäßig hochpreisigen Anwendungsbereichen, etwa bei Luftfahrt-Unternehmen. Was heute für den normalen Computernutzer selbstverständlich ist – zum Beispiel Suchmaschinen, online-Bestellungen, E-Mail, World Wide Web – wurde erst möglich durch die Verbesserung der wahlfreien Speicherung und des Zugriffs auf die Datenbasis, wie sie mit der IBM 1440 realisiert worden war.

### Datenfernverarbeitung

Eine weitere Neuerung, die bei den Olympischen Spielen von Innsbruck 1964 zum ersten Mal im größeren Maßstab zur Anwendung kam, war die Datenfernverarbeitung, *teleprocessing* im Marketing-Jargon von IBM. Die Peripheriegeräte in der Sporthalle waren wie üblich direkt an die beiden Rechner angeschlossen. An allen Wettkampfstätten kommunizierten Schreibmaschinen, Drucker und die Zeitnahme-Geräte über Telefonleitungen mit dem Rechenzentrum. Einige Stellen im Arbeitsspeicher jedes Rechners konnten direkt auf den Anzeigetafeln im Stadion angezeigt werden. Wenn

das gerade laufende Programm eine Information in diese Speicherstellen schrieb, wurde der Inhalt sofort sichtbar, ohne dass die Zeichen nochmals per Hand eingegeben werden mussten.

Die Reporter und Journalisten erhielten meistens innerhalb einer halben Stunde die Ergebnislisten der Wettkämpfe. Früher hatte man für deren Druck zwei Tage benötigt. Auch vom Verlauf eines Wettbewerbs fern vom Stadion, etwa Alpin-Ski, Langlauf oder Biathlon, gelangten noch während des Rennens Informationen zum Zwischenstand an die Zuschauer im Stadion und an die Medien.

IBM stellte dem Veranstalter die gesamte EDV-Infrastruktur – Hardware, Software und Personal – kostenlos zur Verfügung. Es war zu erwarten, dass die auf jedem Fernsehgerät gut sichtbare Werbung mit den drei Großbuchstaben die nicht unbedeutenden Kosten wieder hereinbringen würde. Die Bedeutung für die zukünftige Informationstechnik und die Nutzung war noch nicht abzusehen. Der mit *teleprocessing* erbrachte „proof of concept“ bei den Spielen von Innsbruck hat mit dazu beigetragen, dass die EDV-Industrie die Technik und Verfahren nach und nach verbesserte, bis zu dem Stand, den wir heute gewohnt sind.

### IBM 1410, 1460

Der maximale Arbeitsspeicher von 16 000 Zeichen war eine ernste Limitierung der IBM 1401. Beim Modell IBM 1410 wurde deshalb der Adressraum auf 80 000 Speicherstellen vergrößert. Dazu war es notwendig, die bisher drei Stellen für die Adressen der Speicherstellen auf fünf Stellen zu erweitern. Damit wurden größere Programme mit mehr Speicherplatz möglich. Aufgrund des erweiterten Adressraums waren jetzt Algorithmen mit größerem Speicherbedarf einsetzbar, die indirekt den Durchsatz beschleunigten.

Wegen der Erweiterung der Adressfelder war es nicht möglich, Programme, die für die IBM 1401 übersetzt worden waren, direkt auszuführen. Die Quellenprogramme mussten neu in den Maschinenkode übersetzt werden. Durch Umlegen eines Schalters an der Bedienungskonsole wurde der 1401-Mode aktiviert, mit dem die alten Programme für die IBM 1401 unverändert ausgeführt wurden. Natürlich waren die Beschränkungen des kleineren Computers weiter in Kraft. Dieser Trick für den leichteren Übergang auf das nächste Modell oder sogar die nächste Generation wurde später auch bei anderen Computerfamilien übernommen, etwa mit dem 1401-Mode beim Rechner IBM /360-30 oder dem PDP-11-Mode beim Computer DEC VAX11/780.

In der Endphase der Lebenszeit der 1400-Familie kam noch der Rechner IBM 1460 auf den Markt, der etwa doppelt so schnell war wie das ursprüngliche Modell der IBM 1401. IBM hatte anfangs einen weltweiten Bedarf von ca. 3 000 Rechnern der Familie 1401 geschätzt. Am Ende der Produktionszeit waren etwa 15 000 Stück ausgeliefert worden.

Damals war IBM der größte Computerhersteller der Welt, mit einem Umsatz größer als der Umsatz der anderen Hersteller zusammen, was das Computergeschäft anbetraf. Die Erkennungsfarbe von IBM war Blau, deshalb sprach man von „Big Blue and the seven dwarfs“. Einige der Zwerge von damals – Burroughs, Control Data, Digital Equipment, General Electric, Honeywell, RCA, Univac – existieren heute nicht mehr oder haben sich aus dem Computergeschäft zurückgezogen. Neue Firmen sind dazugekommen und einige davon in Umsatz und Gewinn an IBM vorbeigezogen, etwa Microsoft.

Vor einigen Jahren habe ich den Rechner IBM 1401 wiedergesehen, in der Computerabteilung des Deutschen Museums in München. Er war nicht mehr vollständig und nicht mehr lauffähig. Ich hätte gern noch einmal gehört, wie der Zeilendrucker den Marine-Marsch *Anchors away* von Philip Sousa ratterte oder die Recheneinheit Weihnachtslieder dudelte. Geschickte Programmierer hatten herausgefunden, wie man die Störfrequenzen des Prozessors im Mittelwellen-Bereich in Töne umsetzen konnte. Durch ein Programm mit der richtigen Reihenfolge der Operationen und den korrekten Zeitintervallen konnte man die programmierten Störungen als Weihnachtslieder mit einem Mittelwellen-Empfänger hörbar machen.

## Der wissenschaftliche Rechner IBM 1620

In dieser Zeit wurde noch zwischen Rechnern für betriebliche und wissenschaftliche Anwendungen unterschieden. Das Gegenstück zur IBM 1401 war der Computer IBM 1620. Der Rechner war in der gleichen Schaltungstechnik aufgebaut, hatte aber einen anderen Befehlssatz mit einem maximal möglichen Arbeitsspeicher von 100 000 Stellen. Zahlenfelder bestanden aus aufeinander folgenden Stellen, die wie bei der 1401 durch eine Wortmarke in der Einerstelle begrenzt wurden. Befehle und Register für Gleitkomma-Rechnungen konnten als Option dazugekauft werden. Ein Computer IBM 1620 wurde bei der ersten Hochrechnung der Wahlergebnisse der Bundestagswahl 1965 für das Deutsche Fernsehen eingesetzt. Insgesamt wurden von der IBM 1620 ca. 2 000 Systeme gebaut (@IBM 1620).

## IBM 1130

Insgesamt arbeitete ich dreimal in den Sommerferien als studentischer Programmierer bei IBM, zuletzt 1966. Aufgrund meiner guten Erfahrungen schlug ich meiner Freundin vor, sich für das Studentenprogramm von IBM zu bewerben. Sie wurde angenommen und erhielt eine dreiwöchige Einführung in die Programmierung und Bedienung des wissenschaftlichen Rechners IBM 1130. Anschließend hat sie die Kenntnisse in einem Projekt bei einem Kunden von IBM angewendet.

Meine Hoffnung, dass meine Freundin – wir sind jetzt seit fast fünfzig Jahren verheiratet – durch ihre Tätigkeit bei IBM mehr Verständnis für meine Computer-



Abbildung 3-3: Steuerpult des Rechners IBM 1130 mit Tastatur, Schreibwerk und Anzeigeeinheit

Der Computer IBM 1130, ausgeliefert seit 1965, war zu dieser Zeit der preiswerteste Rechner von IBM. Er war als Nachfolger des Computers IBM 1620 geplant. Wie beim wissenschaftlichen Rechner IBM 1620 waren die Haupt-Zielgruppen Bildungseinrichtungen und Ingenieurbüros, die sich eine größere Rechenanlage nicht leisten konnten. Die IBM 1130 war aus integrierten Schaltungen aufgebaut, was in einem kleineren Raumbedarf und einem geringeren Preis als beim Computer IBM 1620 resultierte. Das System sollte auch als intelligente Steuerelektronik für ein Graphik-Sichtgerät und als Eingabestation für eine räumlich entfernte Großrechenanlage dienen (Abb. 3-3).

Für die Adressierung wurden 15 Bits genutzt, was eine maximale Speicherkapazität von 32 768 Wörter mit 16 Bits oder 65 536 Bytes ergab. Der Arbeitsspeicher war ein Magnetkern-Speicher. Die Befehle hatten drei Adressmodi, direkt, indirekt und indiziert. An den Rechner konnten Wechselplatten-Speicher, Magnetbandgeräte, Lochkartenleser, Zeilendrucker und ein Zeichengerät (Trommelplotter von Calcomp) angeschlossen werden.

Es wurden mehrere Modelle der IBM 1130 mit unterschiedlichen Taktfrequenzen und Arbeitsspeicher-Kapazitäten angeboten. Die Grundausstattung kostete 32 000 bis 41 000 US\$, mit einer Wechselplatte. Von dem Rechner wurden bis 1972 ungefähr 10 000 Stück produziert (@IBM 1130).

### Betriebsnahe Software

Das Betriebssystem des Computers IBM 1130 war das *Disk Monitor System* (DM2). Dafür war ein Rechner mit mindestens 4 KByte Arbeitsspeicher und einem Festplattenspeicher notwendig. Während der Ausführung eines Programms verblieb nur ein sehr kleiner Supervisor mit ungefähr einem Kilobyte im Arbeitsspeicher. War

Affinität aufbringen würde, hat sich im Großen und Ganzen erfüllt. Sie war aber gegen das Programmervirus immun und hat seit ihrer Tätigkeit bei IBM nie wieder ein Computer-Programm geschrieben. Wie viele Millionen anderer IT-Anwender benutzt sie heute ganz selbstverständlich ihren Arbeitsplatz-Rechner für E-Mails, Recherchen und Bestellungen im Internet, auch die Speicherung und Verwaltung ihres Fotoarchivs und das Erstellen von Fotobüchern.

Der Computer IBM 1130, ausgeliefert seit 1965, war zu dieser Zeit der preiswerteste Rechner von IBM. Er war als Nachfolger des Computers IBM 1620 geplant. Wie beim wissenschaftlichen Rechner IBM 1620 waren die Haupt-Zielgruppen Bildungseinrichtungen und Ingenieurbüros, die sich eine größere Rechenanlage nicht leisten konnten. Die IBM 1130 war aus integrierten Schaltungen aufgebaut, was in einem kleineren Raumbedarf und einem geringeren Preis als beim Computer IBM 1620 resultierte. Das System sollte auch als intelligente Steuerelektronik für ein Graphik-Sichtgerät und als Eingabestation für eine räumlich entfernte Großrechenanlage dienen (Abb. 3-3).

der Rechenlauf beendet, wurde die anderen Teile des Betriebssystems wieder in den Arbeitsspeicher geladen und der nächste Lauf gestartet. Der weit überwiegende Anteil der Anwendungen wurde in der Sprache Fortran programmiert. Der Fortran-Compiler der IBM 1130 konnte auf einem Rechner mit nur 4 096 Worten Arbeitsspeicher ausgeführt werden. Das übersetzte Programm durfte größer sein, um auf Modellen der IBM 1130 mit mehr Arbeitsspeicher genutzt zu werden. Der Compiler war in der Regel auf der Festplatte gespeichert, konnte aber auch von Lochstreifen oder Lochkarten eingelesen werden.

Andere Programmiersprachen, die mit der IBM 1130 genutzt werden konnten, waren APL, BASIC, Cobol, FORTH und PL/I (Interpreter). Die Eastern Michigan University entwickelte einen Compiler für Fortran IV als Alternative zum Compiler von IBM, der nur eine Untermenge des Sprachumfangs von Fortran IV verarbeiten konnte. Die Oklahoma State University stellte einen Compiler für Algol 68 bereit, der in Fortran programmiert war. Eher eine Kuriosität war ein Algol-Compiler aus französischer Quelle, mit Programmbefehlen in Französisch, mit „Debut...Fin;“ anstatt „Begin...End;“.

Die Unterprogramm-Bibliotheken für das System IBM 1130 enthielten vorwiegend Funktionen für technisch-wissenschaftliche Anwendungen. Für kleinere Betriebe gab es eine Bibliothek für kommerzielle Anwendungen und die Planung von Projekten. Für die Verwendung als Eingabestation für einen entfernten Großrechner war ein Programm verfügbar, ebenfalls ein System für den computerunterstützten Schriftsatz.

### **Der erste Arbeitsplatz-Rechner von IBM?**

Wenn man so will, war der Rechner 1130 der erste Arbeitsplatz-Computer von IBM. Anstatt des „closed shop“-Betriebs, also die Operateure an der Großrechenanlage hinter Glasscheiben und die Anwender davor, arbeiteten die Nutzer der IBM 1130 im „open shop“-Betrieb. Sie waren für die Programmierung, den Ablauf und die Organisation des Rechenbetriebs selbst verantwortlich. Sie mussten nicht mit anderen Anwendern um die Ressourcen konkurrieren, wenigstens solange sie am Rechner arbeiteten. Heute ist das für die Nutzer von Arbeitsplatz-Rechnern die übliche Arbeitsweise.

Die IBM 1130 war noch ein gutes Stück größer als heutige Arbeitsplatz-Rechner, aber immerhin kleiner als die großen Schränke des Systems IBM /360 im Rechenzentrum. Die IBM 1130 konnte arbeitsplatznah in einem Büro stehen. Wenn nicht zu viele Peripheriegeräte angeschlossen waren, brauchte man für den Betrieb auch keine zusätzliche Klimatisierung. Der wesentliche Fortschritt der heute benutzten Arbeitsplatz-Computer im Vergleich zur IBM 1130 ist, neben der Verringerung der Geräte-Dimensionen, die Vernetzung der Rechner innerhalb des Betriebes und nach außen ins World Wide Web, mit allen Vor- und Nachteilen.

# Datenverarbeitung an der Universität

# 4

## Programmieren in Algol 60

Nach der Infizierung mit dem Programmier-Virus wollte ich unbedingt mehr über Theorie und Praxis der Elektronischen Datenverarbeitung erfahren. Die Suche im Vorlesungsverzeichnis der Universität war in dieser Hinsicht nicht sehr ergiebig. Das Fach Informatik gab es 1964 noch nicht, zumindest nicht an meiner Universität. Durch Zufall hörte ich von Mitstudenten, die Mathematik und Sport fürs Lehramt studierten, dass in der Einführungsübung für das Gebiet Analysis auch Programmieren in der Sprache Algol behandelt wurde. Nach dem Abitur und der Wahl der Fächerkombination Sport, Biologie und Geographie war ich sicher, mit dem Fach Mathematik bis an mein Lebensende nichts mehr zu tun zu haben. Die Abneigung hatte unterschiedliche Gründe, darunter auch meine Faulheit. Ich überwand in diesem Fall meine Vorbehalte und ging jede zweite Woche zum Programmierkurs. Damals konnte man ohne Probleme an Lehrveranstaltungen in Fächern teilnehmen, für die man nicht eingeschrieben war.

Der Unterricht im Programmieren lief in einer Form ab, die man sich heute nicht mehr vorstellen kann. Ein Assistent der Abteilung Angewandte Mathematik behandelte in jeder zweiten Woche die wesentlichen Elemente der Sprache Algol 60, etwa die Umsetzung von mathematischen Formeln, Datenstrukturen, Schleifen, bedingte Verzweigungen, Unterprogramm-Sprünge und die sehr rudimentären Möglichkeiten für die Ein- und Ausgabe. Die Studenten bekamen eine kleine Programmieraufgabe für die nächste Programmierstunde, die sie mit bisher behandelten Sprachelementen lösen konnten. Ob die Mathematikstudenten ihre Lösung abgeben mussten und der Dozent sie auch durchsah, weiß ich nicht mehr. Das Programmieren war eine reine Trockenübung ohne Programmtest an einem realen Computer.

Erst in der letzten Woche des Semesters stellte das Rechenzentrum eine Stunde Rechenzeit insgesamt für alle Teilnehmer der Übung zur Verfügung. Der Computer, an dem die Tests stattfanden, war ein betagter Rechner Zuse Z22, ein Computer der ersten Generation mit Vakuumröhren. Übersetzen und Ausführen des Programms an dem echten Computer war keine Pflicht, deshalb kamen nur die wahren Programmier-Verrückten. Wir wollten herausfinden, wie man ein Programm übersetzt und testet, um später vielleicht den Computer selbst ohne Hilfe benutzen zu können. Wenn bei dieser Gelegenheit die Übersetzung des Programms aufgrund eines Synthaxfehlers abgebrochen wurde oder das Programm in einem Fehler endete, hatte man keine Chance, den Versuch zu wiederholen. Die Anzahl der Teilnehmer war zwar nicht sehr hoch, die Bedienung des Rechners dauerte aber so lange, dass die Stunde Rechenzeit

nur für einen Versuch pro Teilnehmer reichte. Von einem Test des Programms konnte also keine Rede sein, zumindest nach heutigen Maßstäben.

## Rechner Zuse Z22

Unter der Ägide von Konrad Zuse wurde in der gleichnamigen Firma der Rechner Z22 entwickelt. Die interne Logik war aus etwa 600 Modulen mit Vakuumröhren aufgebaut. Die beiden Einsteckteile in Abbildung 4-1 sind wahrscheinlich ein Flipflop und ein NAND-Gatter. Programm und Daten wurden auf einem Magnettrommel-Speicher mit einer Kapazität von 8 192 Worten mit je 38 bit gespeichert. Der Rechner hatte 14 Register zu je 38 bit mit „schnellem“ Kernspeicher als Speichermedium. Die Taktfrequenz der Z22 betrug 3 kHz. Das entsprach der Geschwindigkeit des Trommelspeichers für Schreib- und Lesefunktionen.



Abbildung 4-1: Zwei Basis-Schaltelemente der Z22 mit Vakuumröhren.

Als Geräte für 5-Kanal-Lochstreifen konnten ein Fernschreiber mit Leser und Stanzer, ein schneller elektro-optischer Leser und ein Schnell-Stanzer angeschlossen werden (Abb. 4-2). Am Bedienpult waren Glimmlampen für die Anzeige und Kippschalter für die Eingabe der Bits eines Speicherworts vorhanden. Der Rechner wurde mit 380-V-Drehstrom versorgt, der aber sehr stabil und störungsfrei sein musste. Deshalb gehörte zu jedem Z22-Rechner ein eigener Generator, um den Computer galvanisch vom Netz zu trennen und dadurch Überspannungen und Störspitzen fernzuhalten. Wenn der Generator überbrückt und der Rechner direkt an das Netz angeschlossen wurde, konnte schon das Betätigen eines Lichtschalters im Gebäude den Rechner aus dem Takt bringen. Damit war der aktuelle Rechenlauf beendet.

Da die Vakuumröhren ziemlich viel Hitze entwickelten, war in jedem Rechner-schrank eine Kühlvorrichtung eingebaut, die an eine Wasserzuleitung und einen Abfluss angeschlossen sein musste. Aufgrund der hohen Temperatur der Kathode und der hohen Anodenspannung waren die Vakuumröhren relativ kurzlebig. Der Rechner Z22 in der Universität wurde jeden Tag etwa drei Stunden darauf überprüft, welche Röhren kurz vor dem Ausfall waren. Bei einem begründeten Verdacht wurde das Modul oder die Röhre vorsorglich ausgetauscht, damit der anschließende Rechenbetrieb einigermaßen störungsfrei ablaufen konnte.



Abbildung 4-2: Rechner Zuse Z22. Im zweiten Schrank von rechts ist der Magnettrommel-Speicher mit Ansteuerungs-Elektronik zu erkennen. Neben und auf dem Bedienpult im Vordergrund sind mehrere Peripheriegeräte aufgestellt (Technik-Museum Berlin).

Der Hersteller Zuse lieferte so gut wie keine Software für den Betrieb des Rechners Z22, das größte Manko im Vergleich mit Röhrenrechnern amerikanischer Provenienz aus dieser Zeit. Die Firma verließ sich weitgehend auf die Software-Kompetenz der Universitäten, bei denen die ersten Computer des Typs Z22 aufgestellt wurden. Ein Übersetzer für das Programmieren im Maschinenkode kam von der Universität Freiburg, nach der Herkunft Freiburger Code genannt. Die Universität Mainz steuerte einen Compiler für eine Untermenge von Algol 60 bei, nach den Algol-Spezifikationen der ALCOR-Gruppe. Der Compiler musste sehr klein sein, um in den Trommelspeicher der Z22 zu passen. Deshalb konnte nur ein Teil des Sprachumfangs des Standards Algol 60 implementiert werden.

## Lehrveranstaltungen für Informationstechnik

Um den dringenden Bedarf für die Vermittlung von Computerwissen abzudecken, wurden vom Mathematischen Institut Lehraufträge an Experten von außerhalb der

Universität vergeben. Im Semester nach dem Algol-Kurs hielt ein Professor der benachbarten Fachhochschule eine Lehrveranstaltung zur Elektronischen Datenverarbeitung. Damit sollten interessierten Studenten die Anfangsgründe der Computertechnik und des Programmierens vermittelt werden. Ich nahm an der Vorlesung teil, denn das Programmiervirus hatte mich nicht losgelassen. Inhalt der Lehrveranstaltung waren die Grundprinzipien der Technik und des Aufbaus von Computern, die Funktion der Komponenten, die verfügbaren Peripheriegeräte und ihre Vor- und Nachteile, auch ein wenig Programmieren in der Sprache Algol 60. Die Vorlesung wurde ab und zu ins Uni-Rechenzentrum verlegt, um die theoretischen Ausführungen an der dort vorhandenen Rechenanlage EL X1 zu demonstrieren. Dadurch sollten die Teilnehmer Gelegenheit bekommen, eigene Algol-Programme zu übersetzen und auszutesten, um den Betrieb und die Schwierigkeiten bei der Handhabung der Rechenanlage kennenzulernen.

Im Fach Betriebswirtschaft wurde eine Übung zur Anwendung der Lochkartentechnik für betriebliche Anwendungen angeboten. Die technischen Grundlagen kannte ich aus meinem Ferienjob bei IBM, wahrscheinlich besser als die meisten anderen Studenten im Kurs. Die Klausur am Ende des Semesters war deshalb ziemlich einfach. Nur bei einer Frage musste ich passen, nach der Anwendung der Lochkartentechnik für die Grenznutzen-Berechnung. BWL war nicht mein Fach, deshalb wusste ich nicht, was ein Grenznutzen ist und erst recht nicht, wie er berechnet wird.

Um dem wachsenden Bedarf zu genügen, berief die wirtschaftswissenschaftliche Fakultät 1966 einen Professor, der sich schwerpunktmäßig um Elektronische Datenverarbeitung, Computertechnik und die Anwendung der EDV in der betrieblichen Praxis kümmern sollte. Professor Poensgen hatte einige Jahre an der Harvard University in Boston gearbeitet und gelehrt. In seinen Vorlesungen wurden die aktuellen technischen Entwicklungen der Digital- und Softwaretechnik präsentiert. Fallstudien der Harvard Business School dienten unter anderem als Material in Übungen und Seminaren. Ich nahm an den meisten dieser Lehrveranstaltungen teil, obwohl meine Kenntnisse in der Betriebswirtschaft eher marginal bis nicht existent waren. Dafür kannte ich mich in der Computertechnik aus und konnte programmieren, was ich den anderen Studenten voraus hatte. Als es auf das Ende des Studiums zuing, fragte ich Professor Poensgen, ob er mich im Nebenfach für das Geographie-Diplom prüfen könnte. Er stimmte zu und nahm die mündliche Prüfung im Fachgebiet Computer und EDV ab.

Ungefähr ein Jahr zuvor war ich Mitglied in der *Association for Computing Machinery* geworden. Die ACM ist der weltweit wichtigste Verband, der sich vorwiegend mit den wissenschaftlichen Grundlagen der Informationstechnik befasst. Die mir fachlich näher stehenden ACM-Publikationen sind mir bis heute eine wichtige Informationsquelle geblieben, ebenso wie die online-Bibliothek der ACM.

## Computer Electrologica X1

Die Universität hatte einige Jahre vorher ein Computersystem der niederländischen Firma Electrologica mit dem Modellnamen X1 erworben. Das Design der EL X1 basierte auf der Dissertation des Computer-Pioniers Edsger W. Dijkstra (DIJKSTRA 1959). Dijkstra hat später viele wegweisende Aufsätze zur Entwicklung der höheren Programmiersprachen, zu Algorithmen und der Informatik allgemein geschrieben. Insbesondere für seine Beiträge zu den Programmiersprachen wurde er 1972 von der Association for Computing Machinery mit dem *A. M. Turing Award* geehrt. Diese Auszeichnung gilt als der Nobelpreis für die Computerwissenschaften.

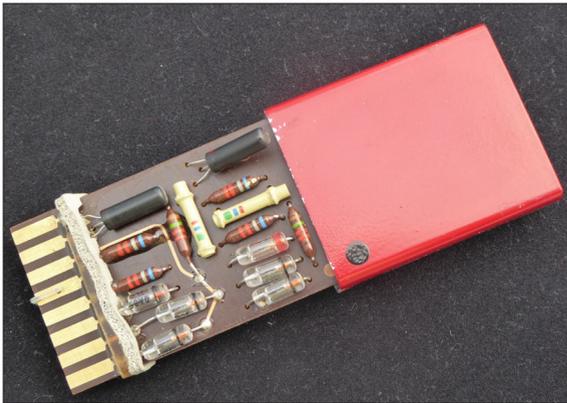


Abbildung 4-3: Logik-Modul der EL X1, wahrscheinlich zwei NAND-Gatter. Die diskreten Transistoren (schwarz) und die Germaniumdioden (Glas) sind gut zu erkennen. Der rote Metallbecher war die Hülle der Platine.

in den technischen Unterlagen für die Rückwand-Platinen vermerkt, um Fehlersuche und Wartung zu erleichtern.

Jedes Speicherelement (Wort) war 27 bit lang. Der Grund für diese ungewöhnliche Wortlänge geht aus den verfügbaren Unterlagen nicht hervor. Maximal konnten mit den 15 Adressbits 32 768 Wörter adressiert werden. Der Arbeitsspeicher war aus Kernspeicher-Modulen aufgebaut.

Der Rechner X1 enthielt eine technische Neuerung, den *Interrupt*, also Unterbrechung. Während einer Ein- oder Ausgabe, die relativ langsam ablief, konnte das Programm weiter fortfahren. War die Ein- oder Ausgabe abgeschlossen, schickte das Gerät ein Signal an die Zentraleinheit. Durch die parallele Ausführung wurden die Rechengeschwindigkeit und der Durchsatz erhöht.

### Hardware

Die Basis-Baueinheiten der EL X1 waren doppelseitige Platinen in Metallbechern, etwa halb so groß wie eine Zigarettenschachtel (Abb. 4-3). Auf der Platine waren digitale Schaltkreise aus diskreten Transistoren, Germaniumdioden und passiven Bauteilen aufgebaut. Die Module wurden in Rückwand-Platinen (*backplanes*) eingesteckt, deren Kontakte auf der Rückseite der Backplanes durch viele Drähte in Wickeltechnik verbunden waren. Die Farbe der Metallbecher repräsentierte den Typ des Schaltelements. Farbe und Position der Module waren

## Literaturverzeichnis

- ALBINUS HJ (1996) Das Hardwaremuseum (XII): AEG-Telefunken TR440. Ein deutscher Großrechner. DIE ZEIT vom 8. 11. 1996  
<http://www.online.uni-marburg.de/hrz/chronik/quellen/zeit-erinnert-an-tr440-961108.pdf>
- Anleitung für die chinesische Rechenmaschine Abacus. 9. verbesserte und überarbeitete Auflage 1972, Robert Oscar Meier & Co., Bremen
- Arduino-Variante (2017) c't magazin für computertechnik, 18/2017, 65–69
- BAGER J (2017) Assistent allgegenwärtig. Digitale Assistenten: vom Spielzeug für Nerds zur Bedienoberfläche für alles. c't magazin für computertechnik, 22/2017, 23
- BANCHY J, LEBSACK R (2017) Open collaboration and IBM Power systems. International Business Machines Corporation, January 2017
- BILL R (2016) Grundlagen der Geo-Informationssysteme. 6., völlig neu bearb. u. erw. Aufl., Wichmann
- BOND E, AUSLANDER M, GRISOFF S, KENNEY R, MYSEWSKI M, SAMMET J, TOBEY R, ZILLES S (1964) FORMAC, an experimental formula manipulation compiler. Proceedings of the 1964 19th ACM national conference
- BRUDERER H (2015) Meilensteine der Rechentechnik: Zur Geschichte der Mathematik und der Informatik. De Gruyter Oldenbourg
- BRUDERER H (2017) Computing history beyond the U.K. and U.S.: selected landmarks from continental Europe. Communications of the ACM, Vol. 60/2, 76–84
- BUCHROITHNER MF, BOULOS MNK, ROBINSON LR (2012) Stereoscopic 3-D solutions for online maps and virtual globes. In: BUCHROITHNER M, True3D in Cartography. Autostereoscopic and solid visualisation of geodata. Springer, Heidelberg, 391–412
- BUZIEK G, DRANSCH D, RASE WD (Hrsg.) (2000) Dynamische Visualisierung. Grundlagen und Anwendungsbeispiele für kartographische Animationen. Springer-Verlag
- CERUZZI PE (2016) Computer: Eine kurze Geschichte. Berlin University Press
- CHOI CQ (2017) Quantum leaps. Advances in „qbit“ design could lead to more powerful computers. Scientific American, December 2017, 14
- CHRISMAN N (2006) Charting the unknown. How computer mapping at Harvard became GIS. ESRI Press, Redlands, CA
- CIARCIA S (1982) Build the Microvox text-to-speech synthesizer, Part 1: Hardware. Part 2: Software. Ciarcia's Circuit Cellar IV, 96–118  
<https://books.google.de/books?id=zQWNinpbFxoC&printsec=frontcover&hl=de#v=onepage&qt&f=false>
- DIJKSTRA EW (1959) Communication with an automatic computer. Dissertation. University of Amsterdam
- DÖLLE M (2017) Das ätzt richtig! Mit Fritzing und dem Bauteile-Editor zur eigenen Platine. c't 14/2017, 142–145

- DUTTON G (1979) American Graph Fleeting. United States population growth 1790–1970. A computer-holographic map animation. Graphic summary, overview, and technical synopsis. Laboratory for Computer Graphics and Spatial Analysis, Harvard University
- EIKENBERG R (2017) Kopflos glücklich. Raspberry Zero per USB-Kabel einrichten. c't magazin für computertechnik 20/2017, 146–147
- EYNARD JD, JENNY B (2016) Illuminated and shadowed contour lines: improving algorithms and evaluating effectiveness. International Journal of Geographical Information Science, Volume 30, Issue 10, 1–21
- GIL D (2017) Quantum computing. New algorithms and techniques open the door to innovative applications. Scientific American, December 2017, 31
- GO J, HART AJ (2017) Fast desktop-scale extrusion additive manufacturing. Additive Manufacturing, Vol.18, December 2017, 276–284  
<http://www.sciencedirect.com/science/article/pii/S2214860416303220>
- GOLDSTINE HH (1972) The computer from Pascal to von Neumann. Princeton University Press
- IBM System/360 principles of operation. IBM Press  
[http://dl.acm.org/ft\\_gateway.cfm?id=1102026&ftid=411285&ftdn=1&CFID=945896632&CF\\_TOKEN=97494457](http://dl.acm.org/ft_gateway.cfm?id=1102026&ftid=411285&ftdn=1&CFID=945896632&CF_TOKEN=97494457)
- JAENICKE H (2017) Wer der Herde folgt, sieht nur Ärsche: Warum wir dringend Helden brauchen. 2. Aufl., Gütersloher Verlagshaus
- JANSSEN JK (2017) Plug-and-Play-VR. Windows Mixed Reality im Test. c't magazin für computertechnik 24/2017, 99–103
- JENNY B (2001) Computergestützte Schattierung. Institut für Kartographie, ETH Zürich  
[www.cartography.oregonstate.edu/pdf/2001\\_Jenny\\_Relief.pdf](http://www.cartography.oregonstate.edu/pdf/2001_Jenny_Relief.pdf)
- KENNELLY P, KIMERLING AJ (2001) Modifications of Tanaka's Illuminated Contour Method. Cartography and Geographic Information Science, Vol. 28, No. 2, 111–123  
[http://www.mbm.gatech.edu/pdf/gis\\_illum.pdf](http://www.mbm.gatech.edu/pdf/gis_illum.pdf)
- KERTSCHER K (2002) Eine Ära und Legende – Die Curta-Rechenmaschine. Zeitschrift für Vermessungstechnik 1/2002, 1–3
- KIRSCHENBAUER S (2004) Empirisch-kartographische Analyse einer echt-dreidimensionalen Darstellung am Beispiel einer topographischen Hochgebirgskarte. Mensch & Buch Verlag, Berlin
- LABS L (2017) Grundlagen zu RAID. Datensicherheit durch Redundanz. c't magazin für computertechnik 17/2017, 176–177
- LABS L (2018) Auf dem Weg zur 100-TByte-Festplatte. Neue Techniken für mehr Kapazität. c't magazin für computertechnik 1/2018, 164–171
- LAMA T (2004) 3D-Welten: Professionelle Animationen und fotorealistische Grafiken mit Raytracing. Hanser-Verlag
- LEITENBERGER B (2014) Computergeschichte(n). Die ersten Jahre des PC, Edition Computer, 2. Auflage. Books on Demand, Norderstedt
- MANDAU M (2018) Unmögliches möglich machen. Im nächsten Jahrzehnt steht uns buchstäblich ein Quantensprung bevor. CHIP 2/2018, 44–47
- DI MARCOBERARDINO R (2018) Mehr Kapazität per Mikrowelle. CHIP 2/2018, 104–105

- MATIS H (2002) Die Wundermaschine. Die unendliche Geschichte der Datenverarbeitung: Von der Rechenuhr zum Internet. Wirtschaftsverlag Carl Ueberreuter, Frankfurt/Wien
- MUELLER S, BAUDISCH P (2017) Personal fabrication. *Communications of the ACM*, 10/2017, 48–49
- O'REAGAN G (2008) A brief history of computing. Springer Verlag
- PEUCKER TK, RASE WD (1970) Computer cartography at Simon Fraser. *Simon Fraser University Computing Centre, Newsletter Vol. 1, No. 4, May 1970*
- PEUCKER TK, TICHENOR M, RASE WD (1972) Automatisierung der Methode der schrägen Schnittflächen. *Kartographische Nachrichten, Heft 4, 1972, 145–148*
- PEUCKER TK, TICHENOR M, RASE WD (1975) The computer version of three relief representations. DAVIS CD, McCULLAGH MJ (ed.), *Display and Analysis of Spatial Data, 187–197*
- PEUCKER TK, FOWLER RJ, LITTLE JJ, MARK D (1978) Digital representation of three-dimensional surfaces by triangulated irregular networks (TIN). *Proceedings Digital Terrain Modeling Symposium, May 1978, ASP, 516–540*
- RASE WD, PEUCKER TK (1971) Erfahrungen mit einem Computerprogramm zur Herstellung thematischer Karten. *Kartographische Nachrichten, Heft 2, 1971, 50–57*
- RASE WD, WUNDERLING P (1974) „Dynamische Karten“ durch computergenerierte Filme. *Der GMD-Spiegel, Informationen aus der wissenschaftlichen Arbeit der Gesellschaft für Mathematik und Datenverarbeitung, Nr. 4, 1974*
- RASE WD (1979) Preiswerte Sprachein/ausgabe: Möglichkeiten, Alternativen. 2. Symposium DECUS München an der Universität Bonn, 1979
- RASE WD (1980) A family of subroutines for plotting graduated symbol maps. *Geoprocessing, Heft 3, 1980*  
<http://www.wdrase.de/ProportionalSymbols-Geoprocessing11980.pdf>
- RASE WD (1984) Computer-controlled input and output of speech: utilizing the acoustic channel for geometric data acquisition. *Proceedings of the International Symposium on Spatial Data Handling, Zürich 1984*
- RASE WD (1987) The evolution of a graduated symbol software package in a changing graphics environment. *International Journal of Geographical Information Systems, Vol. 1, No. 1, 1987*  
<http://www.wdrase.de/ProportionalSymbols-IntJGeoIS11987.pdf>
- RASE WD (2007) Volumenerhaltende Interpolation aus polygonbezogenen Daten in einem unregelmäßigen Dreiecksnetz (TIN). In: STROBL, BLASCHKE, GRIESEBNER (Hrsg.) *Angewandte Geoinformatik 2007. Beiträge zum 19. AGIT-Symposium Salzburg*. Wichmann, 595–604  
<http://www.wdrase.de/VolumenerhaltendeInterpolationAGIT2007.pdf>
- RASE WD (2016) *Kartographische Oberflächen: Interpolation, Analyse, Visualisierung*. Books on Demand, Norderstedt
- RAUNER M (2018) Wie funktioniert ein Quantencomputer? *ZEIT Wissen, 2018/1, 70–74*
- SAVAGE N (2017) A block on the old chip. Block copolymers may help transistors shrink to tinier dimensions. *Communications of the ACM, Vol. 60/11, 12–14*
- SCHMIDT J BH (2012) *Möglichkeiten der Lentikulartechnik als themakartographisches Visualisierungsverfahren*. Dissertation, Fakultät für Geowissenschaften, Ruhr-Universität Bochum

- [www-brs.ub.ruhr-unibochum.de/metahtml/HSS/Diss/SchmidtJanBjoernHolger/diss.pdf](http://www-brs.ub.ruhr-unibochum.de/metahtml/HSS/Diss/SchmidtJanBjoernHolger/diss.pdf)
- SCHNEIDER M (2010) Entwicklung und Realisierung eines Sensornetzwerkes für das Living Place Hamburg. Bachelorarbeit, Studiengang Technische Informatik der Fakultät Technik und Informatik der Hochschule für Angewandte Wissenschaften Hamburg
- SCHULTZ R (2014) Selbstorganisierendes drahtloses Sensor- und Aktornetzwerk. Bachelorarbeit, Studiengang Technische Informatik der Fakultät Technik und Informatik der Hochschule für Angewandte Wissenschaften Hamburg
- SCHUMACHER M (2017) Raspi kann alles. Neue Projekte für den Raspberry Pi. c't magazin für Computertechnik 17/2017, 64–65
- SCHULZ K (2017) Topografische Karten der Mars-Region MC-11-E - erstellt auf Basis eines HRSC-Orthofotomosaiks der Mission Mars Express. AGIT Journal für Angewandte Geoinformatik 3/2017. Wichmann-Verlag, 166–177  
[https://gispoint.de/fileadmin/user\\_upload/paper\\_gis\\_open/AGIT\\_2017/537633018.pdf](https://gispoint.de/fileadmin/user_upload/paper_gis_open/AGIT_2017/537633018.pdf)
- SCHWARTZKOPFF P, BARTL K, BARTL R, ERNSTBERGER A (1994) PEP ART. 3-D-Bilder der neuen Art. Südwest Verlag, München
- STILLER A (2017) Prozessorgeflüster. Von Übergängen und Überraschungen. c't magazin für Computertechnik, 23/2017, 40–41
- STILLER A (2017) Letztes Prozessorgeflüster. Von Ruhe und Bewegung. c't magazin für Computertechnik, 25/2017, 16–18
- STILLER A, WINDECK C (2017) Asien führt. In der 50. Supercomputer-Bestenliste TOP500 liegt China vorne. c't magazin für Computertechnik, 25/2017, 20–21
- SZONDY D (2016) Curta calculator: The mechanical marvel born in a Nazi death camp  
<http://www.newatlas.com/curta-death-camp-calculator/45506/>
- TANAKA K (1950) The relief contour method of representing topography on maps. Geographical Review, Vol. 40 (3), 444–456.
- TOBLER WR (1979) Smooth pycnophylactic interpolation for geographical regions. Journal of the American Statistical Association, Vol. 74, No. 357, 519–535  
<http://geog.ucsb.edu/~kclarke/G232/Pycno.pdf>
- TOBLER W (2002), persönliche Mitteilung
- WATKINS CD, MALLETTE V (1996) Stereogram programming techniques. Charles River Media, Rockland, MA, USA
- WINDECK C (2017) Prozessorgeflüster. Von Kooperationen, Preisen und Verzögerungen. c't magazin für Computertechnik, 24/2017, 18–19
- XENAKIS J (1971) The PL/I-FORMAC interpreter. SYMSAC 71, Proceedings of the second ACM symposium on Symbolic and algebraic manipulation, 105–114  
[http://dl.acm.org/ft\\_gateway.cfm?id=806275&ftid=54275&dwn=1&CFID=946699605&CFTOKEN=41966945](http://dl.acm.org/ft_gateway.cfm?id=806275&ftid=54275&dwn=1&CFID=946699605&CFTOKEN=41966945)
- YOÉLI P (1966) Die Mechanisierung der Analytischen Schattierung (Facettenmethode). Kartographische Nachrichten, Jahrg. 16, Heft 3, 103–107
- YOÉLI P (1967) Die Richtung des Lichts bei analytischer Schattierung. Kartographische Nachrichten, Jahrg. 17, Heft 2, 37–44

- 
- YOÉLI P (1983) Shadowed contours with computer and plotter. *The American Cartographer* 1983, 10 (2), 101–110
- YOSHINO Y (1963) *The Japanese abacus explained*. Dover Publications, New York
- ZUSE K (2010) *Der Computer - mein Lebenswerk*, 5. Auflage. Springer-Verlag

## Hyperlinks

3D-Briefmarken

<http://philaquelymoi.blogspot.de>

3D-Easy

<http://www.3d-easy.de>

3D-PluraView

[www.schneider-digital.com](http://www.schneider-digital.com)

4D Concepts

<https://www.4dconcepts.de/3d-druck/?L=0>

ASTER

<https://asterweb.jpl.nasa.gov/gdem.asp>

Autostereogramm-Briefmarken

<http://philaquelymoi.blogspot.de/search/label/3D%20-%20Stereoscopic%20Stamps>

BBSR-Erreichbarkeitsmodell

[http://www.bbsr.bund.de/nn\\_1061104/BBSR/DE/Raumbeobachtung/](http://www.bbsr.bund.de/nn_1061104/BBSR/DE/Raumbeobachtung/)

[UeberRaumbeobachtung/Komponenten/Erreichbarkeitsmodell/erreichbarkeitsmodell.html](http://www.bbsr.bund.de/BBSR/DE/FP/MORO/Studien/2017/erreichbarkeitsmodell/01-start.html)

<http://www.bbsr.bund.de/BBSR/DE/FP/MORO/Studien/2017/erreichbarkeitsmodell/01-start.html>

Boolesche Algebra

[https://de.wikipedia.org/wiki/Boolesche\\_Algebra](https://de.wikipedia.org/wiki/Boolesche_Algebra)

Curta

<https://de.wikipedia.org/wiki/Curta>

da Vinci Color

[http://eu.xyzprinting.com/eu\\_de/Product/da-Vinci-Color](http://eu.xyzprinting.com/eu_de/Product/da-Vinci-Color)

DEC Alpha

<https://de.wikipedia.org/wiki/Alpha-Prozessor>

DEC PDP-8

<https://en.wikipedia.org/wiki/PDP-8>

DEC PDP-11

<https://en.wikipedia.org/wiki/PDP-11>

DEC VAX

[https://de.wikipedia.org/wiki/Virtual\\_Address\\_eXtension](https://de.wikipedia.org/wiki/Virtual_Address_eXtension)

Digi-art

<http://www.digi-art.de>

Flick

<https://www.pi-supply.com/brand/pi-supply/flick/>

Fly-eye

<http://www.finlens.com/finlens-flyeye/>

Fritzing

<http://www.fritzing.org>

Google AIY Voice Kit

<https://shop.pimoroni.de/products/google-aiy-voice-kit>

Green500

<https://www.top500.org/green500/lists/2017/11/>

HP-35

<https://de.wikipedia.org/wiki/HP-35>

IBM 1130

[https://en.wikipedia.org/wiki/IBM\\_1130](https://en.wikipedia.org/wiki/IBM_1130)

IBM 1620

[https://en.wikipedia.org/wiki/IBM\\_1620](https://en.wikipedia.org/wiki/IBM_1620)

IBM System /360

<https://de.wikipedia.org/wiki/System/360>

INKAR

<http://www.inkar.de>

Kernspeicher

<https://de.wikipedia.org/wiki/Kernspeicher>

Leia

<https://www.leiainc.com/lit-by-leia/>

Linsenrasterbild

<https://de.wikipedia.org/wiki/Linsenrasterbild>

LOM

[https://en.wikipedia.org/wiki/Laminated\\_object\\_manufacturing](https://en.wikipedia.org/wiki/Laminated_object_manufacturing)

Magics

<http://www.materialise.com/de/software/magics>

Mathematica

<https://de.wikipedia.org/wiki/Mathematica>

Mooresches Gesetz

[https://de.wikipedia.org/wiki/Mooresches\\_Gesetz](https://de.wikipedia.org/wiki/Mooresches_Gesetz)

Nano Pi

<http://www.friendlyarm.com/index.php?route=product/category&path=69>

Ogon3D

<http://www.ogon3d.com>

OpenMP

<https://de.wikipedia.org/wiki/OpenMP>

POV-Ray

<http://www.povray.org>

Quantencomputer

<https://de.wikipedia.org/wiki/Quantencomputer>

RasPiGNSS „Aldebaran“

<http://www.drfasching.com/products/gnss/raspignss>

Rechenschieber

<https://de.wikipedia.org/wiki/Rechenschieber>

RISC-Computer

[https://de.wikipedia.org/wiki/Reduced\\_Instruction\\_Set\\_Computer](https://de.wikipedia.org/wiki/Reduced_Instruction_Set_Computer)

Shapeways

<https://www.shapeways.com>

Shapewerk

<http://www.shapewerk.com>

Stratasys

<http://www.stratasys.com/3d-printers>

Tanaka Surfer

<http://www.goldensoftware.com/blog/variations-in-hillshading-creating-tanaka-style-illuminated-contour-maps>

Transistor

<https://de.wikipedia.org/wiki/Transistor>

top500

<https://www.top500.org/list/2017/11/>

Triangle

<http://www.cs.cmu.edu/~quake/triangle.html>

Winteracter

<http://www.winteracter.com>

Venus Willendorf

[https://onlineshop.post.at/onlineshop/stamps---philately/stamps/block-issues/the-willendorf-venus-3d\\_3029](https://onlineshop.post.at/onlineshop/stamps---philately/stamps/block-issues/the-willendorf-venus-3d_3029)

x86-Prozessor

<https://de.wikipedia.org/wiki/X86-Prozessor>

Zahlenschieber

<https://de.wikipedia.org/wiki/Zahlenschieber>

Zuse Z25

<https://de.wikipedia.org/wiki/Z25>