



Title: Die Entwicklung der programmgesteuerten Rechenanlagen
Author(s): Konrad Zuse
Date: ?
Published by: Konrad Zuse Internet Archive
Source: Document - ZIA ID: 0662

The Konrad Zuse Internet Archive preserves and offers free access to the digitized original documents of Konrad Zuse's private papers and to other related sources.

The Konrad Zuse Internet Archive is a nonprofit service that helps scholars, researchers, students and other interested parties discover, use and build upon a wide range of content in a digital archive. For more information about the Konrad Zuse Internet Archive, please contact zusearchive@zib.de.

Your use of the Konrad Zuse Internet Archive indicates your acceptance of the Terms & Conditions of Use (<http://zuse.zib.de/tou>) including the following license agreement. If you do not accept the Terms & Conditions of Use you are not permitted to use the material.

This work by Konrad Zuse Internet Archive is licensed under a
Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License
(<http://creativecommons.org/licenses/by-nc-sa/3.0/>).

Based on a work at <http://zuse.zib.de>



Attribution (BY) - You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work). Attribute with "Konrad Zuse Internet Archive (<http://zuse.zib.de>)".

Noncommercial (NC) - You may not use this work for commercial purposes.

Share Alike (SA) - If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

The usage of this document requires the consideration of possible third party copyrights, and might necessitate obtaining the consent of the copyright holder. The Konrad Zuse Internet Archive assumes no liability with respect to the rights of third parties. The Konrad Zuse Internet Archive is not responsible for the claims of any third party resulting from any infringement of copyright laws.

045/020

0662

I. Die Entwicklung
der programmgesteuerten
Rechenanlagen

Professor Dr. Konrad Zuse, Bad Hersfeld

(

(

rechts und zurück zwischen den Kettfäden hindurchgeschossen. Dadurch, daß die Kettfäden nach einem Programm in bestimmter Folge angehoben und abgesenkt werden, entsteht ein Muster, das bei Verwendung mehrerer Farben natürlich auch mehrfarbig sein kann. Diese verschiedene Stellung der Kettfäden wurde bei den traditionellen Webstühlen von Hand eingestellt. Jacquard steuerte nun die Lage der Kettfäden über Zugfäden von einer Steuereinrichtung her, die im wesentlichen aus einer Abtastung von Lochkarten bestand (Fig. 2).

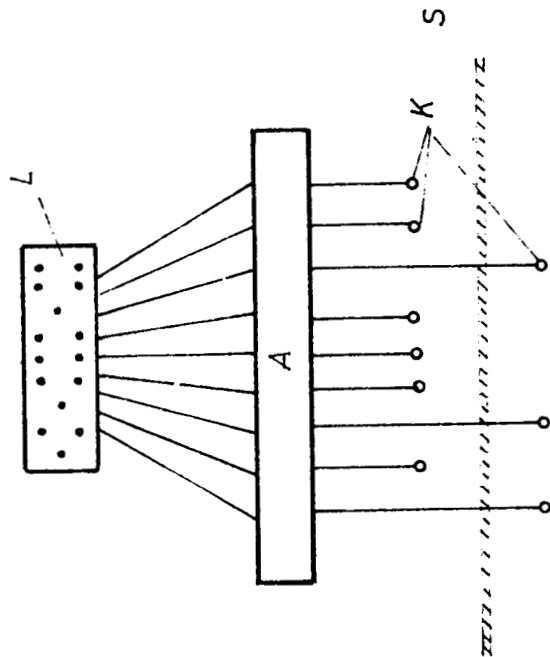


Fig. 2

Die programmgesteuerten Arbeitsmaschinen sind also keineswegs eine Erfindung unserer Zeit, sondern gingen der programmgesteuerten Rechenmaschine voraus. Ebenso ist die Lochkarte keine Erfindung der Rechenmaschinenfachleute, sondern wurde von den Webstühlen übernommen.

Konstruktiv wollte Babbage sein Problem nun so lösen, daß er als Kernstück der Anlage ein vollautomatisches Rechenwerk für die vier arithmetischen Grundoperationen $+$, $-$, \times , $:$ baute, welches mit einem Speicherwerk verbunden war. Das Rechenwerk und die einzelnen Speicherzellen sollten für je 50 Dezimalstellen gebaut sein, das Speicherwerk sollte 1000 solcher 50-stelligen Dezimalstellen aufnehmen können.

Eine Programmsteuerung wollte er konstruktiv so vorsehen, daß er getrennte Operationskarten und Variablenkarten einführt. Die Operationskarten sollten vier Lochfelder aufweisen, wobei beispielsweise bei Ausführung der Multiplikation in das dieser Operation zugehörige Lochfeld ein Loch eingestanzt war (Fig. 3).



Fig. 3

Der Vater der programmgesteuerten Rechenmaschine ist zweifelsohne Charles Babbage, ein englischer Mathematiker, der bereits in der Mitte des vorigen Jahrhunderts die wesentlichen Grundgedanken für Rechner entwickelte und auch ein Modell baute. Dieses Modell ist allerdings trotz erheblicher Summen (100 000 engl. Pfund) und dreißigjähriger Arbeitszeit nicht fertig geworden. Babbage war mit seinen Gedanken eben den technischen Gegebenheiten seiner Zeit zu weit voraus. Die Möglichkeiten der Elektrotechnik standen ihm noch nicht zur Verfügung. So war er auf die Mechanik angewiesen, und er versuchte, in der traditionellen Rechenmaschinentechnik mit Ziffernrädchen, Zahnstangen und dergleichen der komplizierten Aufgabe Herr zu werden.

Angeregt wurde Babbage durch die damals gerade erfundenen Jacquard-Webstühle. Der Grundgedanke ihrer Konstruktion ist verhältnismäßig einfach. Ein Gewebe besteht bekanntlich aus Kettfäden (K) und Schußfäden (S) (Fig. 1); das

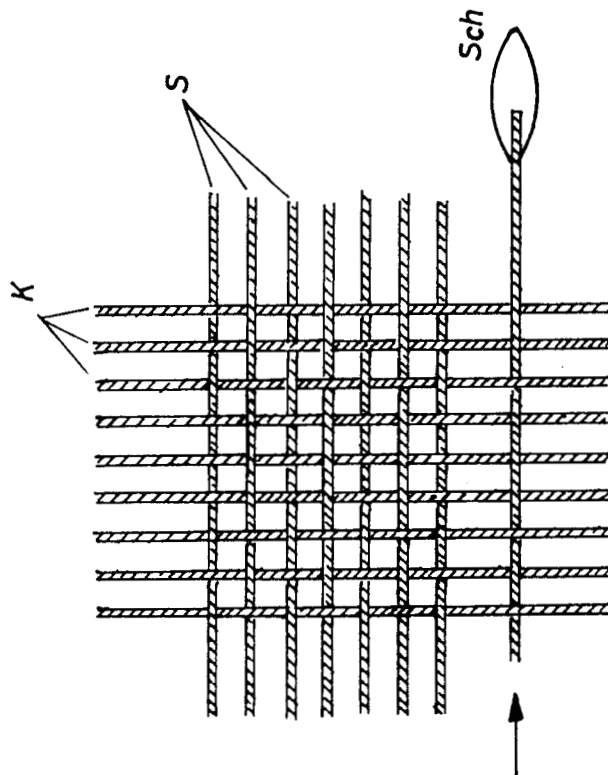


Fig. 1

Fig. 1

verbindende Zelle ein Loch vorgesehen war. Auf diese Weise sollte das Rechenprogramm durch Aneinanderreihen von aus Vorratsstapeln zu entnehmenden Karten zusammengestellt werden, ein Verfahren, welches zweifelsohne einige Vorteile hat, da komplizierte Tastaturen und Locheinrichtungen nicht erforderlich sind und Änderungen leicht ermöglicht werden.

Babbage hat nicht nur die Grundkonzeption für eine programmgesteuerte Rechenmaschine erstmalig konzipiert, sondern deutet auch schon die Möglichkeit des bedingten Befehls an, wenn er auch diese konstruktiv nicht ausgebaut hat. Ferner zwang ihn die hohe Stellenzahl (50 Dezimalstellen), in seiner Rechenmaschine ein besonderes Stellenübertragungsprinzip einzuführen, welches es erlaubte, nach Vorbereitung einer Kette von Übertragungsgliedern diese Stellenübertragung über sämtliche 50 Stellen gleichzeitig durchzuschalten. Auch diese Idee ist von den Rechenmaschinenfachleuten nicht aufgegriffen worden, sondern erst später wieder im Rahmen der Entwicklung der heutigen programmgesteuerten Rechenmaschine unabhängig an mehreren Stellen „erfunden“ worden.

Die Entwicklung von Babbage geniet fast völlig in Vergessenheit. Den Fachleuten der Rechenmaschinentechnik waren seine Gedanken damals schon nicht geläufig und sind auch in der Folgezeit nicht von ihnen aufgegriffen worden. Deshalb wurde auch die ein Jahrhundert später einsetzende Entwicklung der heutigen programmgesteuerten Rechenmaschine von Außenseitern getragen. Jedoch mag es interessant sein, vor der Betrachtung dieser Entwicklung einige andere Vorläufer zu erwähnen.

Die eigentliche Rechenmaschinentechnik bewegte sich im wesentlichen um die mehr oder weniger automatischen Mehrspeziesrechenmaschinen, die auf die Ausführung arithmetischer Operationen beschränkt waren. Einige Geräte wiesen nach heutigen Begriffen verhältnismäßig primitive Speichermöglichkeiten auf. Am weitesten war wohl die Lochkartenmaschine ins Neuland vorgedrungen, deren Entwicklung ziemlich unabhängig von der übrigen Rechenmaschinentechnik verlief. Bei den Lochkartengeräten konnten z.B. über eine Schaltbox verschiedene Teile eines Gerätes, im allgemeinen Zähler, Rechenwerke und Lochkarten-Ein- und Ausgabegeräte, miteinander verbunden werden.

Diese Schaltbox konnte der Aufgabe entsprechend durch Steckverbindung beliebig geschaltet werden. Auch eine gewisse – wenn auch primitive – Abfolge von Operationen konnte damit gesteuert werden. Es liegt also eine Art Programmsteuerung vor. Von einer Programmsprache (oder Symbolik) konnte allerdings noch kaum die Rede sein, es sei denn, daß man eine hintereinandergeschriebene Aufzählung der zu verbindenden Pole der Schaltbox als Programm wertete.

Einen verhältnismäßig hohen Stand erreichten auch die Analoggeräte. Diese arbeiten bekanntlich nicht ziffernmäßig, sondern stellen Zahlenwerte durch physikalische Größen dar. Auch hier waren die ersten Geräte mechanisch, und die Zahlenwerte wurden durch die Position von Gliedern bzw. Drehwinkel und Winkelgeschwindigkeiten von Achsen repräsentiert.

Wesentlich ist dabei, daß die einzelnen Aggregate einer mathematischen Formel entsprechend zusammengeschaltet werden können (Fig. 4).

Diese Schaltungen erfolgen nach einer Art „Blockdiagramm“ und stellen eine besondere Art von „Programmierung“ dar. Man kann diese Geräte jedoch kaum als „Vorläufer“ der programmgesteuerten digitalen Rechenmaschine bezeichnen,

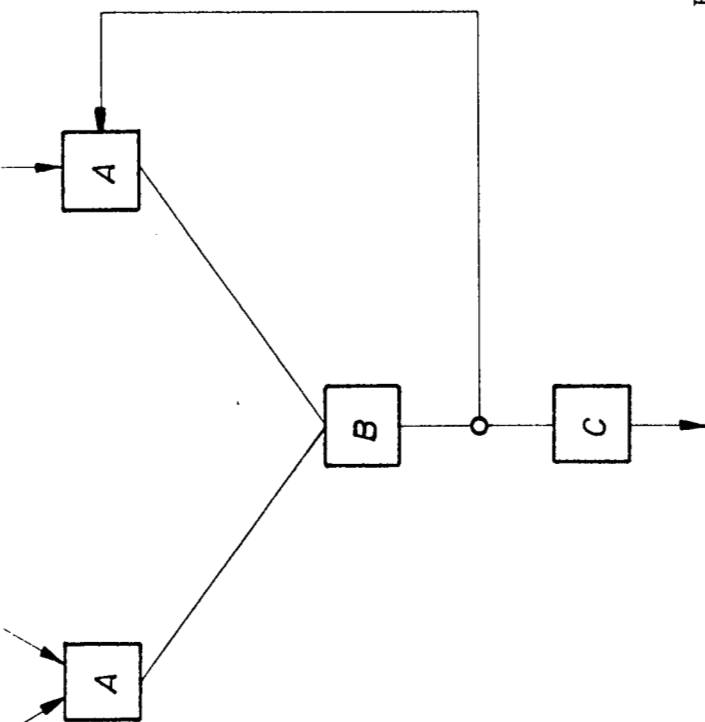


Fig. 4

da diese, im Gegensatz zum Aufbau des Programmes als Schaltung, das Programm als Folge von zeitlich hintereinander ablaufenden einzelnen Operationen formulieren. Interessant ist in diesem Zusammenhang, daß die Analoggeräte trotz aller Fortschritte der digitalen Rechenautomaten heute immer noch eine große Bedeutung haben, wobei sich inzwischen auch bei diesen Geräten die elektronische Technik durchgesetzt hat.

Eine gewisse Vorarbeit für die modernen Rechenmaschinen war auch die Idee der Verwendung des binären Zahlensystems. Dieses mathematisch von Leibniz bereits um 1700 konzipierte Zahlensystem mit den Ziffern 0 und 1 (bei Leibniz unter dem Namen „Dyadik“) hat verschiedene Erfinder angeregt.

In den dreißiger Jahren meldete der Franzose Valtat ein Patent an, welches eine mechanisch im Binärsystem arbeitende Rechenmaschine zum Gegenstand hat, die aus drei Teilen besteht: einem Übersetzer vom Dezimal- ins Binärsystem, einem Rechenwerk im Binärsystem und einem Übersetzer vom Binär- ins Dezimalsystem. An eine Programmsteuerung war dabei nicht gedacht. Durch den dreiteiligen Aufbau wurde der Vorteil des einfachen Rechenwerkes infolge des Aufwandes wieder kompensiert, so daß sich diese Idee für reine Tischrechenmaschinen nicht durchsetzen konnte.

Etwas weiter ging ein anderer Franzose, Couffignal, der in seiner Doktorarbeit „Sur l'Analyse Mécanique. – Application aux Machines à Calculer et aux Calculs

geschlossenen universellen Lösung führten.

Auch der Amerikaner *Stibitz* baute bereits vor dem zweiten Weltkrieg ein arithmetisches Rechenwerk im Binärsystem, welches darüber hinaus in der Lage war, arithmetische Operationen mit komplexen Zahlen automatisch durchzuführen.

Als einen Vorbereiter der Idee der programmgesteuerten Rechenmaschine kann man auch den englischen Mathematiker *Turing* bezeichnen. Er ging allerdings von einer völlig anderen Problemstellung aus. Der Bau praktisch brauchbarer Rechengereäte interessierte ihn zunächst überhaupt nicht. Es ging ihm viel mehr darum, das Modell einer Rechenmaschine zu besitzen, um mathematische Sätze im Sinne der mathematischen Logik zu beweisen. Er stellte dem Begriff der „Beweisbarkeit“ den Begriff der „Berechenbarkeit“ von Funktionen gegenüber und entwarf ein verhältnismäßig einfaches Modell einer universellen Rechenmaschine. Dieses Modell ist praktisch zwar nicht brauchbar, spielt aber doch eine sehr wesentliche Rolle in der Theorie der Berechenbarkeit und auch der Programmsteuerung.

Eine ausgesprochen vorbereitende Arbeit wurde indirekt von denjenigen geleistet, welche elegante numerische Berechnungsverfahren für praktisch zu lösende Aufgaben aus Wissenschaft und Technik ausarbeiteten. Besonders für die Baustatik waren hervorragend ausgearbeitete Rechenverfahren entwickelt worden, welche es erlaubten, längere Rechengänge mit verhältnismäßig wenig geschulten Mitarbeitern durchzuführen. Gerade diese vorbereitenden Arbeiten regten dann ja auch zum Bau der programmgesteuerten Rechenmaschine an.

Die eigentliche Entwicklung der programmgesteuerten Rechenmaschine im heutigen Sinne setzte etwa in den Jahren 1936 – 38 ein, und zwar unabhängig voneinander in Deutschland und in den USA. Die verschiedenen Entwicklungen liefen zeitlich etwa parallel, wobei die deutsche Entwicklung einen kleinen zeitlichen Vorsprung hatte. Folgendes Schema gibt einen Überblick über Namen und Geräte der Beteiligten:

U S A		DEUTSCHLAND	
Aiken Mark I	Stibitz	ENIAC Eckert, Mauchly	Z 1 – Z 4 1941, 1945 (Berlin)
1944	1945	1945	Dirks

Betrachten wir zunächst die Entwicklungen in den USA.

Die Entwicklung von Aiken an der Harvard University bei Boston war wohl zunächst die erfolgreichste. Das Gerät war elektromechanisch aufgebaut; man nannte es „automatic sequence controlled calculator“ (also: „befehlsfolgesteuert“). Es arbeitete im Dezimalsystem mit festem Komma und war angeblich bereits im letzten Kriegsjahre im Einsatz. Die Aiken'sche Entwicklung wurde nach dem Kriege durch den Bau weiterer Geräte (bis Mark IV) weiter vorangetrieben. Von ihr aus gingen wesentliche Anregungen in alle Welt und sie hat wohl

wicklungen bekräftigt.

Der Name *Stibitz* wurde schon im Zusammenhang mit dem Binärsystem erwähnt. *Stibitz* baute bei der Firma Bell während des Krieges auch ein lochstreifen-gesteuertes Gerät in Relais-technik. Es arbeitete im Binärsystem.

Große Berühmtheit hat das Gerät ENIAC erlangt, welches von Eckert und Mauchly in der Moore School, University of Pennsylvania gebaut wurde. Es war 1945 im wesentlichen fertiggestellt, aber erst 1947 wirklich arbeitsfähig. Es war das erste rein elektronische Rechengereät der Welt. Zu bewundern ist die amerikanische Größzügigkeit; denn es enthielt 18 000 Röhren. Allerdings war es in der Programmgebung noch verhältnismäßig primitiv. Auch die Lösung der arithmetischen Operationen mit Hilfe von Röhren war nach heutigen Begriffen verhältnismäßig umständlich. Die Speicherkapazität war auf 20 Worte beschränkt. Als Modell diente das mechanische dezimale Addierwerk mit Ziffernrädchen. Die zehn Stellungen eines Ziffernrädchens wurden mit 10 Flip-Flops dargestellt, welche in einer Ringschaltung so geschaltet waren, daß ein Impuls von einem auf das nächste übertragen werden konnte. Das ergab pro Dezimale bereits 20 Röhren, bei 20 Dezimalstellen also bereits 400 Röhren, wozu noch eine ganze Reihe weiterer Röhren für Übertragungen, Steuerungen usw. kam. Dementsprechend war der Aufwand unverhältnismäßig hoch im Vergleich zum erzielten Erfolg. Nichtsdestoweniger stellt die ENIAC eine durchgreifende Pioniertat dar und hat wesentliche Impulse für die daraufhin schlagartig einsetzende Entwicklung der elektronischen Rechenmaschinen gegeben.

Die Entwicklung der Geräte Z 1 – Z 4 wurde von mir etwa in den Jahren 1936 – 1945 in Berlin durchgeführt. Die Z 1 war rein mechanisch, Z 2, Z 3, Z 4 waren in Relais-Technik mit einigen Schrittschaltern zu Steuerungszwecken gebaut, wobei die Z 4 einen mechanischen Speicher hatte. Sämtliche Geräte hatten Lochstreifensteuerung, arbeiteten konsequent im Binärsystem und – außer der Z 2 – mit gleitendem Komma. Das Gerät Z 3, welches 1941 fertiggestellt war, wurde für eine Reihe von Versuchsrechnungen eingesetzt. Noch während des Krieges wurde der Bau eines verbesserten Modells Z 4 durchgeführt, welches 1945 einfache Rechnungen ausführen konnte. Sämtliche Modelle, außer dem Gerät Z 4, wurden im Kriege durch Bombeneinwirkung zerstört. Das historische Modell Z 3 wurde später nachgebaut.

In enger Zusammenarbeit mit dieser Entwicklung liefen die Arbeiten meines Studienfreundes *Dr. Schreyer*. Er setzte sich zum Ziel, die gleichen Schaltungen, die in den Geräten Z 1 – Z 4 mit elektromechanischen Relais aufgebaut waren, mit elektronischen Bausteinen, vornehmlich Glühlampen und Röhren, aufzubauen. Uns schwebte dabei eine Art eindeutige Abbildung von einer Technik in die andere vor. Die theoretischen Untersuchungen haben inzwischen ergeben, daß es genügt, für die aussagenlogischen Grundoperationen konstruktive Lösungen zu finden, außer einigen Speicher- und Übertragungselementen. Der Grundgedanke war nun der, ein Gerät in „abstrakter Schaltungstechnik“ zu entwerfen und dann in eine beliebige Relais-technik zu übertragen. Dieser Gedanke ist zweifelsohne zunächst verlockend und auch durchführbar; jedoch führt er zu recht aufwendigen Lösungen bei der systematischen Übertragung von elektromechanischen Relaisschaltungen auf Röhrenschaltungen. Herr *Dr. Schreyer* begann 1937 mit der Entwicklung elementarer Schaltungen für die Grundoperationen und baute während des Krieges einen Elektronenrechner für 10 Binärstellen.

Schon während der Konstruktion dieser Geräte lief ebenfalls unabhängig voneinander auf beiden Seiten des Atlantik die Entwicklung von Programmierungssystemen.

In den USA waren es *Goldstine* und *von Neumann*, welche als erste eine mathematische Formulierung der Programmierungstechnik schufen und veröffentlichten. In dieser Arbeit wurde bereits das Arbeiten mit bedingten Befehlen behandelt und die verschiedenen Möglichkeiten der Adressenänderungen besprochen.

Ich selbst wurde durch die theoretischen Untersuchungen, die mit der Konstruktion der Geräte einhergingen, bald zu der Idee des universellen Rechnens hingeführt, die über das Zahlenrechnen weit hinausgeht. Hierauf basierend entwickelte ich in den Jahren 1938 – 45 eine universelle Programmsprache, den „Plankalkül“, welche grundsätzlich alle rechnerischen Probleme zu formulieren gestattet. Diese Arbeit blieb unveröffentlicht. Hinweise befinden sich lediglich in den späteren Veröffentlichungen „Archiv der Mathematik“ Band 1, 1948/49, Heft 6, „Elektronische Rechenanlagen“ 1. Jahrgang, 1959, Heft 2.

Zu erwähnen sind auch noch die Arbeiten von *Norbert Wiener*, der den inzwischen sattsam bekanntgewordenen Begriff „Kybernetik“ schuf und eine Reihe wichtiger theoretischer und praktischer Versuche durchführte, die wesentliche Anregungen für spätere Entwicklungen gegeben haben. Es ist jedoch falsch, ihn – wie es manchmal geschieht – als Vater der programmgesteuerten Rechenmaschine oder des „Elektronengehirns“ zu bezeichnen; denn die eigentliche Entwicklung dieser Geräte verlief, zumindest in der Pionierzeit, unabhängig von ihm. Heute hat der Begriff Kybernetik eine etwas umstrittene Bedeutung. Einige nennen zu einer sehr umfassenden Auslegung dieses Begriffes. Es würde dann aber eine Reihe von Disziplinen wie wesentliche Teile der Biologie, Psychologie, Fernmeldetechnik, Regelungstechnik, Datenverarbeitung, Informationstheorie darunter fallen. Zweifelsohne liegt es aber nicht im Sinne der Forscher und Theoretiker auf diesen verschiedensten Gebieten, plötzlich als „Kybernetiker“ bezeichnet zu werden. In den USA, wo dieses Schlagwort entstanden ist, wird in der Welt der Datenverarbeitung und der elektronischen Rechenmaschinen dieses Wort kaum erwähnt. In Rußland hingegen hat es in den letzten Jahren eine weite Verbreitung gefunden, und man zählt wohl z.B. die gesamte Prozeßsteuerung dazu. Am besten ist es wohl, die Kybernetik als „Brücke zwischen den Wissenschaften“ zu bezeichnen (H. Frank 1961 – 62, Umschauverlag; Zemanek, MTW-Mitteilungen, 4. Jahrgang 1957, Heft 5, Seite 281 – 189).

Wesentliche Vorarbeiten für die theoretische Durchleuchtung der Gesetze der Datenverarbeitung und indirekt auch für die Entwicklung von Programmsprachen

theory of relay and switching circuits". Eine weitere Veröffentlichung gab Hansi Piesch 1939 „Begriff der allgemeinen Schaltungstechnik“ und „Über die Vereinfachung von allgemeinen Schaltungen“.

Im wesentlichen laufen die Arbeiten auf die Anwendung der Boole'schen Algebra (oder auch des Aussagenkalküls) auf Relaischaltungen hinaus, wenn auch z.B. bei H. Piesch diese Gegenüberstellung noch nicht klar ausgesprochen ist. Heute ist die Schaltungsmathematik weit ausgebaut und bildet einen Teil der Automatentheorie.

Eine ebenfalls verhältnismäßig frühe Arbeit stellt die Informationstheorie dar, welche im wesentlichen von Neumann und Shannon entwickelt wurde. Jedoch erfolgte dies zu einer Zeit, als die datenverarbeitenden Maschinen im heutigen Sinne noch keine Bedeutung hatten. Ziel der Informationstheorie ist es in erster Linie, ein einwandfreies Maß für die Leistungsfähigkeit von Informationsübertragungskanälen zu erhalten. Der Begriff „Informationsgehalt“ ist dabei streng definiert. Jedoch deckt sich diese Definition nicht mit der dem üblichen Sprachgebrauch entsprechenden Auffassung. Schon bei Wiener finden sich widersprechende Erklärungen. Aus diesem Grund ist es auch gefährlich, Begriffe der Informationstheorie wie Informationsgehalt, Redundanz und Entropie leichtfertig auf andere Gebiete zu übertragen.

Bei oberflächlicher Betrachtung könnte man vielleicht meinen, daß die Informationstheorie die zur Informationsverarbeitung gehörige Theorie sei. Das ist jedoch nicht so. Leider werden in der Informationsverarbeitungstheorie oft Begriffe der eigentlichen Informationstheorie falsch angewandt, und zwar in einem Sinne und mit Zielen, die den Schöpfern der eigentlichen Informationstheorie fernlagen.

Heute haben wir neben der praktischen Entwicklung der programmgesteuerten Rechenmaschine ein Gebäude von verschiedenen Theorien, die damit in mehr oder weniger engem Zusammenhang stehen und teils auch ineinander übergehen. So sind die Automatentheorie und die Theorie der Formelsprachen eng miteinander verwandt, ja können aufeinander abgebildet werden. Die Grundgedanken der Automatentheorie sind an sich jedem Ingenieur angeboren, nur formuliert er sie nicht explizit. Automaten lassen sich auf verschiedene Weise in ihren Gesetzen erfassen und beschreiben. Damit ist die Brücke zu den Formelsprachen geschlagen. Erwähnt wurde schon, daß die Schaltungsmathematik als Zweig der Automatentheorie betrachtet werden kann, wobei die eindeutige Beziehung zum Formalismus der Boole'schen Algebra, also einer Formelsprache, besonders einfach zu verstehen ist. Aufgrund dieses Zusammenhangs ist aber ein Theoretiker für Formelsprachen ebenso berechtigt zu behaupten, daß die Automatentheorie nur ein Teil dieser Theorie ist, da jeder Automat durch entsprechende Formelsprachen beschrieben werden kann und seine Gesetze formal abgeleitet werden können. In diesem Zusammenhang sei noch die Theorie der selbstorganisierenden Systeme erwähnt, die zunehmende Bedeutung erlangt und bereits das Thema eigener Tagungen darstellt.

Auch die lernenden Systeme gehören hierher und können mit den selbstorganisierenden Systemen in Beziehung gesetzt werden.

Die verschiedenen genannten Theorien finden ihre Anwendung nicht nur in der Technik, sondern auch bei der Untersuchung biologischer Systeme. Vergleichende

maschine wurden schon während der Pionierzeit der programmgesteuerten Rechenmaschine entwickelt. Es entstand eine Reihe von Modellvorstellungen. Die mehr ingenieurmäßig denkenden Theoretiker der programmgesteuerten Rechenmaschine entwickelten Modelle, die im wesentlichen in Vorstellungen der nach den Gesetzen der Boole'schen Algebra arbeitenden Relaischaltungen aufgebaut waren. Die Biologen und Neurologen zeigten dann allerdings, daß diese Modelle nicht ohne weiteres auf die Natur übertragen werden können und entwickelten ihre eigenen Modellvorstellungen, welche mehr mit Schwellwerten und vielfachen Verflechtungen arbeiten. Die Brücke zwischen diesen verschiedenen Schulen bilden Kybernetik, Automatentheorie, Theorie der selbstorganisierenden Systeme usw., wobei die Theorie der Formelsprachen selbstverständlich auch eine wesentliche Rolle spielt.

Programmwerk und am Speicherwerk die mit dem Rechenwerk zu verbindenden Speicherzellen eingestellt.

Versetzen wir uns zurück in die Zeit vor rund 20 Jahren, so stellte diese Einrichtung gegenüber den damals bestehenden Rechenvorrichtungen bereits einen erheblichen Fortschritt dar. Erstmals war es möglich, jede beliebige, aus den arithmetischen Grundoperationen aufgebaute, explizit gegebene mathematische Formel nach Eingabe der Zahlenwerte für die Variablen auszurechnen. Es konnte z.B. in der Baustatik ein komplettes Stabwerk, für dessen Struktur einmal ein Programm angefertigt war, mit beliebigen Maßen und Kräfteverteilungen vollautomatisch durchgerechnet werden.

Ein anderes Beispiel sei aus der Optik gewählt. Bei der Konstruktion eines neuen Objektes muß eine Unzahl von Variablen von Linsen verschiedener Krümmungsradien, Lage bzw. Abstände und Glasarten in unterschiedlicher Zusammenstellung durchgerechnet werden. Jede Durchrechnung zerfällt wiederum in eine große Zahl von einzelnen Strahldurchrechnungen, deren Verlauf, den Gesetzen der geometrischen Optik entsprechend, an jeder brechenden Fläche neu errechnet werden muß. So besteht die Berechnung des Objektivs im wesentlichen aus einer großen Zahl von Strahldurchrechnungen durch ein gegebenes optisches System (meist auch für Strahlen verschiedener Farben), so daß sich der Schwierigkeitsgrad im wesentlichen aus der Zahl der brechenden Flächen ergibt.

Vor Erscheinen der programmgesteuerten Rechenmaschine waren große Rechenbüros nur mit dieser Aufgabe befaßt. Die Durchrechnung eines einzelnen Strahles konnte bereits Stunden in Anspruch nehmen, die Durchrechnung eines größeren Objektives in verschiedenen Variationen Jahre. Mit dem in Fig. 5 gezeigten Maschinentyp war es nun möglich, die Berechnung für einen Strahldurchgang durch ein beliebig kompliziertes optisches System vollautomatisch durchzuführen. Dies löste in der optischen Industrie bereits eine kleine Revolution aus. Man hielt damit die Aufgabe für gelöst und hielt alle Vorschläge, noch mehr automatisch zu berechnen, zunächst für Spielerei.

Nach diesem kurzen Überblick über die heutigen theoretischen Gebäude wollen wir zu den programmgesteuerten Geräten zurückkehren. Die geschichtliche Entwicklung nach 1945 setzte in einer solchen Breite ein, daß es schwer ist, hier die wichtigsten Marksteine herauszustellen. Es seien daher nur einige Entwicklungstendenzen aufgeführt. Diese lassen sich nach verschiedenen Gesichtspunkten aufzeigen. Ich möchte hier zunächst einmal die wesentlichen Tendenzen in der Entwicklung des Programmablaufes darlegen.

Wir haben bereits das Grundprinzip der Maschine von Babbage kennengelernt. Bei dem Gerät werden die zu errechnenden Formeln in eine Reihe von Einzelbefehlen aufgelöst, welche nacheinander ablaufen. Nach ähnlichen Prinzipien sind auch andere Geräte aus der Pionierzeit aufgebaut (Mark I, Z 1 - Z 4). Dieses Grundprinzip läßt sich in dem Blockbild von Fig. 5 darstellen. Das Programmwerk P tastet einen Lochstreifen oder eine Lochkartenfolge L ab, auf dem die Befehlsfolge aufgezeichnet ist. Im Rechenwerk R werden die Grundoperationen der Arithmetik nach Einstellung der Operanden selbsttätig durch-

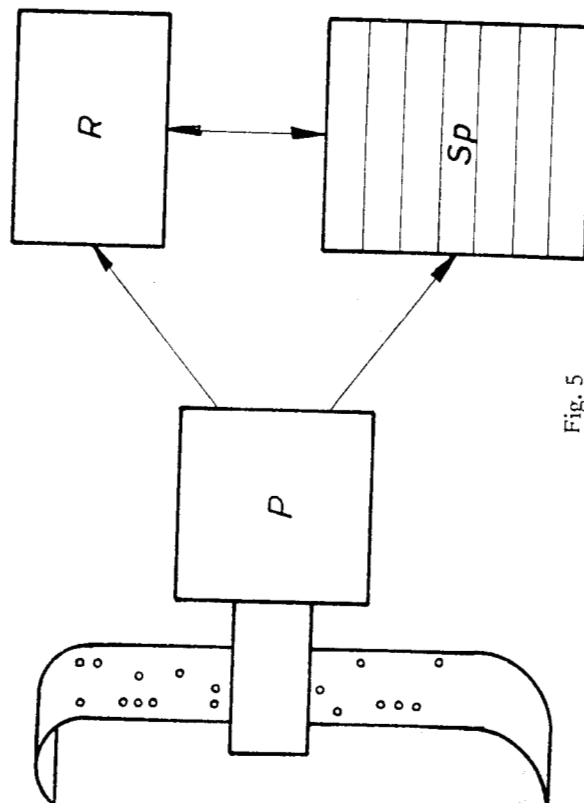


Fig. 5

Heute nach 20 Jahren Entwicklung haben wir eine grundsätzlich andere Situation. Von einer leistungsfähigen Rechenmaschine und einem guten Programm erwartet man heute eine hohe Beweglichkeit und Anpassungsfähigkeit an die verschiedenen Situationen. Am Beispiel des Stabwerks bedeutet dies folgendes. Das Programm soll grundsätzlich so aufgebaut sein, daß alle Systeme einer bestimmten Klasse, beispielsweise ebene Stabwerke beliebiger Struktur, durchgerechnet werden können. Eingegeben werden dabei nicht nur Maße und Zahlenwerte, sondern in erster Linie zunächst die Struktur des Systems, die in der Aufzählung beispielsweise der einzelnen Stäbe und Knotenpunkte und ihrer Gestaltung und Beziehung untereinander besteht. Erst dadurch erlangen die programmgesteuerten Rechenmaschinen in der Praxis ihre volle Bedeutung. Denn solange nach dem alten Prinzip für jede gesonderte Struktur eines Stabwerkes ein gesondertes Programm „von Hand“ gefertigt werden mußte, war der Programmierungsaufwand im allgemeinen zu hoch, um den Einsatz der Maschine lohnend zu machen.

Ähnlich liegt der Fall beim anderen genannten Beispiel aus der Optik. Daß eine elektronische Rechenmaschine den Strahlengang durch ein gegebenes System durchrechnen kann, ist heute vollkommen selbstverständlich; davon spricht man überhaupt nicht mehr. Verlangt wird heute eine weitgehend selbsttätige Variation der Daten des Objekts, wobei aufgrund der errechneten Fehler nach bestimmten Formeln Verbesserungen für das System errechnet werden und dieses verbesserte System dann wieder durchgerechnet wird, bis zur immer stärkeren Annäherung an ein System, das der gestellten Aufgabe genügt (beispielsweise eines Weitwinkelobjektivs).

Ähnliche Beispiele ließen sich auf hundert anderen Gebieten finden. Woran liegt es nun, daß unsere heutigen Maschinen diesen Aufgaben gewachsen sind? Um dies zu erkennen, verfolgen wir am besten die geschichtliche Entwicklung der Blockschemen solcher Geräte.

1. DER BEDINGTE BEFEHL

Den wesentlichen Schritt in dieser Richtung stellt die Einführung der bedingten Befehle dar. Konstruktiv läßt sich dies durch einen Draht B darstellen, der vom Rechenwerk auf das Programm einwirkt und den Ablauf des Programms von den Ergebnissen der Rechnung abhängig macht. (Fig. 6)

Angewandt auf ein Gerät vom Typ nach Fig. 5 sind die Möglichkeiten einer solchen Rückwirkung allerdings noch recht begrenzt; jedoch löst der Grunde-

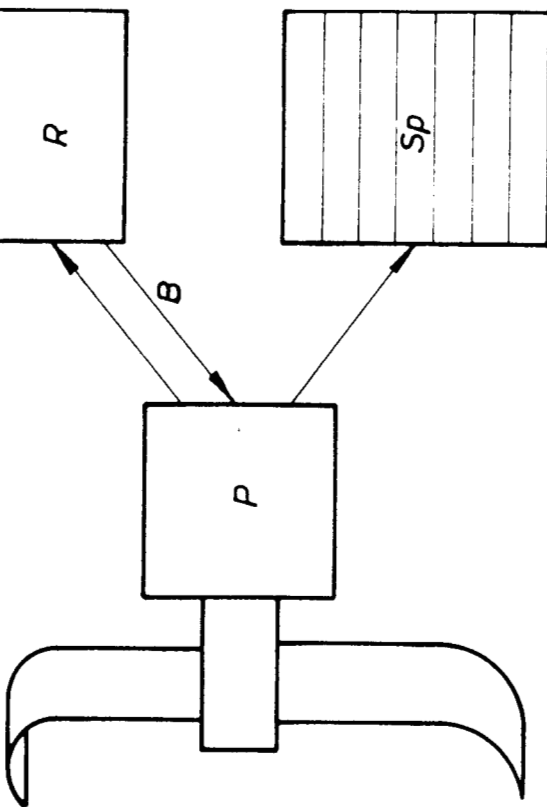


Fig. 6

danke selbst bei konsequenter Anwendung eine Reihe von weiteren Entwicklungen aus, die die Möglichkeiten der programmgesteuerten Rechenmaschine enorm steigern.

In der Ausführung nach Fig. 6 ist es z.B. möglich, abhängig vom Ergebnis einen bedingten Schlußbefehl in das Programm einzubauen.

2. EINFÜHRUNG VON PROGRAMM- UND ZAHLENSPEICHER

Interessanter ist jedoch die Beeinflussung des Programmablaufes selbst, in dem abhängig von den Resultaten des Rechenwerkes verschiedene Programmverschiebungen und Unterprogramme aufgerufen werden. Konstruktiv kann dies auf verschiedene Weise erfolgen. Die ersten Geräte waren ja im wesentlichen lochstreifengesteuert. Bei Verwendung nur eines Abtasters entsprechend Fig. 5 und 6 mußten verschiedene Programmteile auf dem Lochstreifen hintereinander untergebracht werden und an ihrem Beginn mit „Rufzeichen“ versehen werden. Damit war es im Prinzip möglich, durch blindes Überlaufen von Programmfolgen variable Abläufe von Programmen durchzuführen. Allerdings zeigte sich bald der Nachteil, daß das Aufsuchen eines bestimmten Unterprogramms erhebliche Zeit für das Durchlaufen der Streifen erforderte. So lag der Gedanke nahe, mehrere Abtaster zu verwenden und auf diesen verschiedene Teile des Programms unterzubringen. (Fig. 7, A 1 – A 3) Dabei konnten die einzelnen Programme selbst zyklisch ausgebildet sein. (Schleifen)

Damit war eine weitere Stufe der Beweglichkeit erreicht. Aber dies löste den nächsten Gedanken aus: der mechanische Aufwand des Abtastens der Lochstreifen mit einer Reihe von Abtastern ist unangenehm. Die decodierten Programme

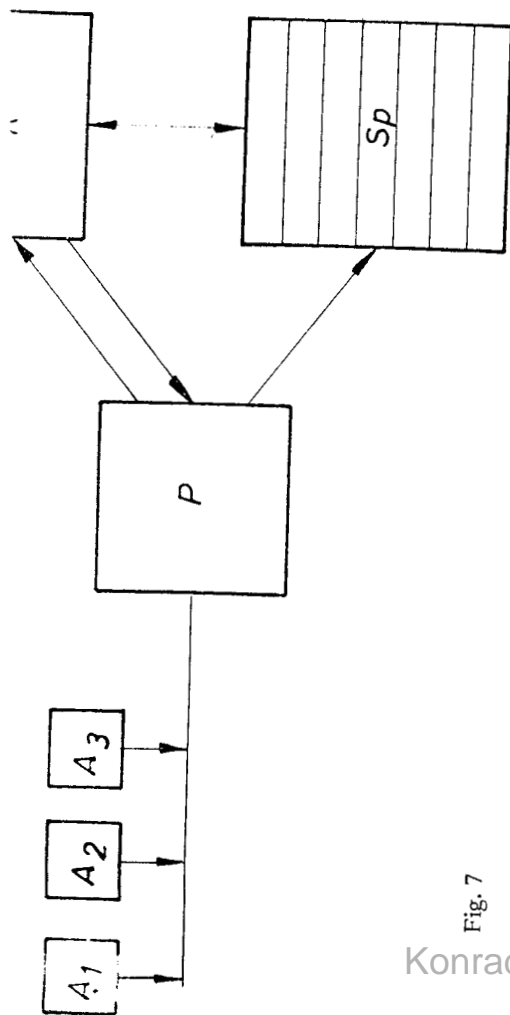


Fig. 7

sind aber nichts weiter als eine Anordnung von Folgen von Ja-Nein-Werten; ebenso wie man inzwischen gelernt hatte, die Zahlen, gleich ob Dezimal- oder Binärzahlen, in Folgen von Ja-Nein-Werten aufzulösen. Die Speicher brauchen dann auch nur Folgen von Ja-Nein-Werten aufzunehmen, also mußten solche Speicher auch Programme speichern können. Diese werden nur zu Beginn von einem Lochstreifen her in das Speicherwerk eingelesen, nachher entnimmt das Gerät die Befehle diesem Speicher. Fig. 8 zeigt eine solche Anordnung mit zusätzlichem Programm-Speicher (Sp P). Dadurch war wiederum eine höhere Stufe der Beweglichkeit der Programme erreicht, denn innerhalb des Speichers ist das Aufspalten in einzelne Programmteile und ihr Abrufen vielgestaltiger und einfacher möglich und das Wechseln der Programme sowie das Auswechseln von

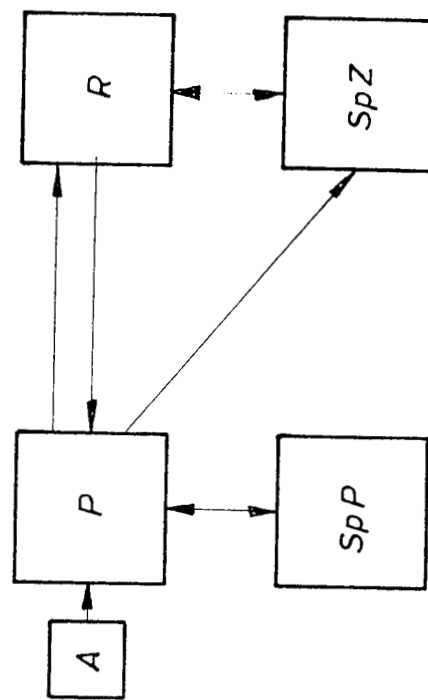


Fig. 8

nannte Speicher, der der Speicherung von Zahlen dient mit Sp Z bezeichnet. Konstruktiv sind die beiden Speicher Sp P und Sp Z gleich gebaut; daraus ergibt sich die Frage, ob es nötig ist, beide Speicher zu trennen bzw. ob es sinnvoll ist, nur einen Speicher zu verwenden, der Zahlen und Befehle aufnimmt. Tatsächlich wird dadurch eine weitere Stufe der Beweglichkeit erreicht, denn nunmehr können in dem Gerät „Informationen“ allgemein behandelt werden, und es ist ein weiterer Schritt zum allgemeinen Rechnen getan. Ein solches Gerät zeigt Fig. 9.

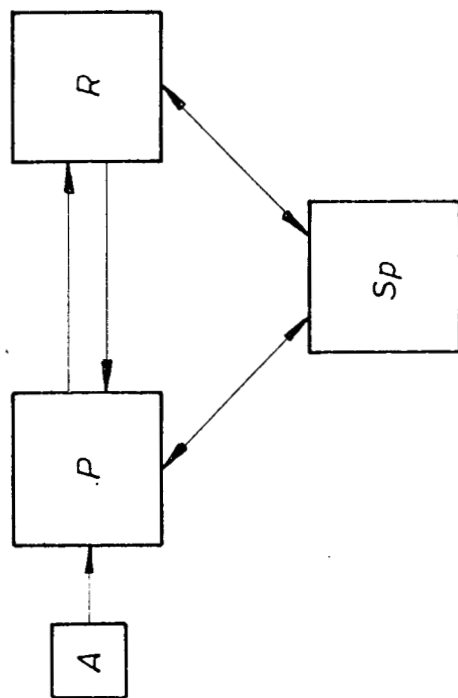


Fig. 9

Wir haben jetzt wieder nur drei Teile wie in Fig. 5; jedoch durch die gemischte Speicherung von Programmen und Zahlen eine höhere Entwicklungsstufe erreicht.

3. ADRESSENRECHNUNG

In den Beispielen von Fig. 6 – 8 haben wir Maßnahmen besprochen, die es gestatten, den Ablauf von Programmen zu variieren. Damit sind aber die Möglichkeiten, Programme beweglicher zu gestalten, noch nicht erschöpft. Ein einfaches Beispiel möge dies veranschaulichen: Betrachten wir das Programm für die Berechnung einer Determinante. Ist der Grad n der Determinante gegeben, so kann das Programm durch eine Maschine vom Babbage-Typ leicht aufgestellt werden. Wechseln nur die Zahlenwerte, so ist eine beliebig häufige Verwendung des gleichen Programms möglich. Wechselt jedoch der Grad der Determinante, so muß für eine Maschine dieses Typs für jedes n ein neues Programm aufgestellt werden. Auch die in Fig. 6 – 9 besprochenen Verbesserungen helfen hier nur wenig weiter.

Tatsächlich ist aber das Aufstellen eines Programms für eine Determinante n -ter Ordnung eine rein schematische Arbeit, die nach strengen Gesetzen, nach einem strengen „Algorithmus“, vor sich geht. Die verschiedenen Programme un-

sch, die Berechnung dieser Adressen kann aber wiederum zum Gegenstand eines Programms gemacht werden, da die auszuführenden Operationen im wesentlichen aus einzelnen Abzähloperationen bestehen, wobei bestimmte, sich aus dem Grad n ergebende Grenzen einzuhalten sind. Diese Adressenänderungen wurden in ihren verschiedenen Möglichkeiten durch Goldstine und Neumann untersucht. Auf die von ihnen formulierten Unterschiede zwischen Adressenänderungen erster, zweiter und dritter Art brauchen wir hier im einzelnen nicht einzugehen. Konstruktiv besteht die Aufgabe darin, abhängig vom Programm selbst und von den Ergebnissen des eigentlichen Rechnens, Umrechnungen mit den Adressen innerhalb der Programme durchzuführen.

Die ersten Geräte dieser Art ergänzten hierzu das Programmwerk durch ein gesondertes Rechenwerk PR (Fig. 10).

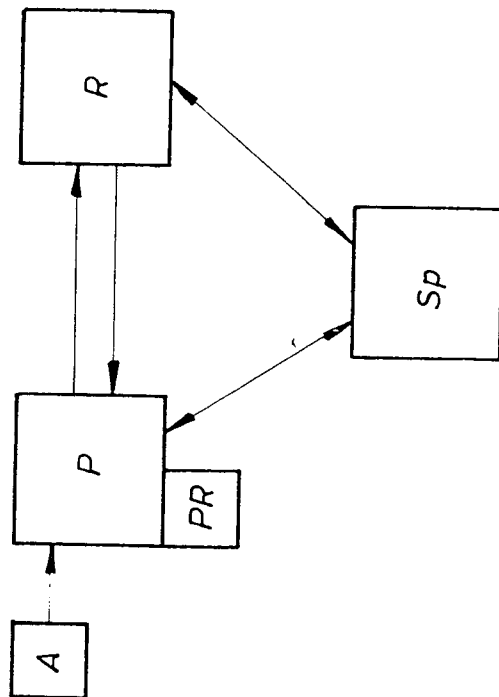


Fig. 10

Aber auch hier entdeckte man bald, daß kein logischer Grund vorlag, für diese Adressenumrechnungen ein gesondertes Rechenwerk einzubauen. Diese wenigen, einfachen Rechenoperationen konnten leicht vom Hauptrechenwerk mit erledigt werden, zumal ja Befehle und Zahlen sowieso bereits in einem gemeinsamen Speicher aufgenommen werden und somit auch die Befehle leicht in das Rechenwerk überführt werden können.

Hiermit kommen wir nun wieder zu einem verhältnismäßig einfachen Rechenmaschinentyp, der auch hier im wesentlichen aus drei Teilen besteht: Programmwerk, Rechenwerk, Speicherwerk (äußerlich entsprechend Fig. 9), wobei der Grad der Beweglichkeit wiederum erheblich erhöht ist.

Alle heutigen auf dem Markt befindlichen Geräte sind im wesentlichen von diesem Typ, wobei allerdings die Art der gegenseitigen Verflechtung und insbesondere die Organisation der bedingten Befehle eine Fülle von Variationen zuläßt, so daß eine große Zahl unterschiedlicher Systeme vorhanden ist, die sich in Bezug auf logische Beweglichkeit, Möglichkeiten der Adressenumrechnung, Organisation, arithmetische Operationen usw. unterscheiden.

Rückblickend können wir feststellen, daß wir bei der Entwicklung von einem zunächst verhältnismäßig einfachen Typ (Fig. 5) über komplizierte Zwischenstufen wieder zu einem einfachen Typ gekommen sind, wobei wir uns allerdings auf einer höheren Ebene des Zusammenspiels der einzelnen Teile befinden. Es liegt nun der Gedanke nahe, diese Entwicklung zu extrapolieren und die Frage zu stellen, ob weitere Zusammenfassungen von heute noch aufgabenmäßig und konstruktiv unterschiedlichen Geräteteilen zu noch höheren Stufen der Beweglichkeit führen können.

Bei genauer Untersuchung heutiger Geräte können wir bereits feststellen, daß die Teile Programmwerk und Rechenwerk nicht mehr streng getrennt sind. Die Möglichkeit der Befehlsumrechnungen bringt bereits eine weitgehende Verschmelzung mit sich. Wir haben dann einen Gerätetyp entsprechend Fig. 11.



Fig. 11

Bei fast allen heutigen Gerätetypen haben wir aber noch konstruktiv eine strenge Unterscheidung zwischen den Teilen, in denen Informationen verarbeitet werden (Steuern (P), Rechnen (R)) und den Teilen, in denen die Information nur gespeichert wird (Sp). Dies ist in erster Linie konstruktiv und ökonomisch bedingt, da informationsverarbeitende Bauelemente das Mehrfache von solchen kosten, welche nur speichern.

Angenommen, diese Schranken ließen sich beseitigen, würde eine Maschine, bei der eine weitgehende Verschmelzung dieser Teile eintritt, neue logische Möglichkeiten erschließen?

zwischen verarbeitenden und speichernden Elementen. Zwar haben wir z.B. im menschlichen Gehirn eine Aufteilung in verschiedene Gebiete, die verschiedene Funktionen haben, z.B. Geruchsgehirn, akustisches Gehirn usw., jedoch scheinen sämtliche beteiligten Zellen sowohl zu verarbeiten als auch zu speichern. Jedfalls ist die Forschung noch nicht so weit, hier einen strengen Unterschied zu erkennen. Zumindest ist die Verflechtung dieser Aufgaben äußerst intensiv. Für diese Verhältnisse hat die moderne Datenverarbeitung auch schon ein Schlagwort bereit: das assoziative Gedächtnis bzw. die assoziative Speicherung.

Die normalen Speicherwerke unserer heutigen Rechenmaschinen arbeiten nach dem Prinzip der Adressierung und des Aufrufs der Zellennummer. Der assoziative Speicher arbeitet dagegen nach dem Prinzip des Aufrufs nach dem gespeicherten Inhalt. Ebenso scheint das menschliche Gehirn organisiert zu sein. Alle Teile sind speichernd, rechnend und organisierend zugleich. Dadurch ergibt sich tatsächlich ein hohes Niveau der Beweglichkeit der Informationsverarbeitung, welches über das Niveau unserer heutigen Rechenmaschinen erheblich hinausgeht.

Man ist sich heute bereits darüber im klaren, daß die Übertragung dieses Prinzips auf Rechengereäte erhebliche Vorteile bringen könnte. Allerdings steht dem noch eine harte Wirklichkeit entgegen. Es ist bis heute nicht möglich, assoziative Speicher großer Kapazität zu erträglichen Preisen zu bauen. Aber dieses Problem ist in vollem Fluß. Es wird intensiv daran gearbeitet (Vgl. die Arbeit von Dr. Jessen, „Assoziative Speicherung“).

Um allerdings die Leistungsfähigkeit des menschlichen Gehirns in bezug auf seine assoziativen Kombinationsmöglichkeiten auch nur annähernd zu erreichen, ist darüber hinaus noch eine Erschließung der Quantität erforderlich. Geistige Schlüsse, wie sie im menschlichen Gehirn vor sich gehen, erfordern eine unvorstellbar große Zahl gespeicherter und geordneter Information, welche laufend und gleichzeitig zur Bearbeitung und zum freien Aufruf zur Verfügung steht. Bis jetzt war die Aussicht, dieses Ziel zu erreichen, noch gering. Einen gewissen Lichtblick bildet die Technik der integrierten Schaltungen. Jedoch müßten sie selbst noch aus dem Pionierstadium herauskommen und wirklich gestatten, große Mengen von Schaltelementen zu Preisen zu produzieren, die um Zehnerpotenzen niedriger liegen, als die heutigen Bauelemente.

Nach diesem Überblick, der Programmabläufe betrifft und uns von Geräten des Typs Babbage zu Geräten großer assoziativer Speicher führte, sei noch ein Überblick über die Möglichkeiten und Tendenzen der Informationsdarstellung selbst und den Aufbau der Rechenwerke gegeben.

Die ersten Geräte (z.B. Babbage, Z 3 und andere) haben Rechenwerke mit voll ausgebauter Arithmetik. Die im Speicher aufzunehmende Informationseinheit stellt eine Zahl dar, die durch eine Folge von Ja-Nein-Werten (heute Bits genannt) dargestellt wird. Die Einführung des „gleitenden Kommas“ und die Fortsetzung nach automatischer Übersetzung solcher Zahlen, beispielsweise vom dezimalen ins binäre System und umgekehrt, erforderte recht komplizierte Rechenwerke, wobei diese Rechenwerke auch logistische Entscheidungselemente und Steuerelemente enthielten. Ihr Aufbau in elektromechanischer Relais-technik ließ sich aber mit Hilfe der gleichzeitig entwickelten Gesetze der Schaltungsmathematik noch gut beherrschen. Die Übertragung derartiger Rechenwerke mit voll ausgebauter Arithmetik im gleitenden Komma auf die elektronische Technik führte allerdings zu sehr umfangreichen komplizierten Konstruktionen.

Nun hatte man ja schon vorher erkannt, daß grundsätzlich jede Information in Ja-Nein-Werte aufgelöst werden kann und entsprechend auch sämtliche Rechenoperationen in Elementaroperationen zwischen Ja – Nein zerlegt werden können. Dies gilt selbstverständlich auch für die Zahlenrechnung. Bereits Turing hatte dies erkannt und arbeitete mit der Zerlegung von „berechenbaren“ Funktionen in einfache Grundelemente. Allerdings ist die Idee der Turing-Maschine für den Ingenieur ohne praktischen Wert, da die Programmierung außerordentlich kompliziert ist und bereits bei der Darstellung der Addition oder Multiplikation zu außerordentlich komplizierten Programmierungen führt, die kaum überblickt werden können.

Etwas mehr dem Denken des Ingenieurs angepaßt ist die Übertragung des Prinzips der Babbage-Maschine auf das reine Ja-Nein-Wert-Prinzip. Der grundsätzliche Aufbau bleibt: Programmwerk, Rechenwerk, Speicherwerk. Jedoch ist das Rechenwerk lediglich in der Lage, die Grundoperationen des Aussagenkalküls auszuführen, während das Speicherwerk je Zelle nur einen Ja-Nein-Wert aufnimmt. Ein derartiges Gerät sei als Babbage-Boole-Maschine bezeichnet, da es nach dem Prinzip von Babbage mit Boole'schen Variablen arbeitet.

Ein derartiges Gerät ist, da es keine bedingten Befehle und keine Adressenumkehrungen kennt, wie wir bereits gesehen haben, nur in der Lage, „geradeaus“ zu rechnen, d.h. es können nur starre Programmabläufe damit durchgeführt werden. Dies ist ein eindeutiger Nachteil gegenüber der Turing-Maschine, denn diese kennt

tem Kontakt mit der Außenwelt, beispielsweise im industriellen Verfahrens- oder Fertigungsprozeß. Meßgeräte, sogenannte Geber, Stückzähler usw. geben ihre Daten direkt in vercodeter Form an den Rechner, der diese richtig auswertet und Steuerungs- bzw. Regelungsdaten an den Prozeß zurückgibt, wobei bei einem geschlossenen Kreislauf die entsprechenden Einstellungen ebenfalls durch direkten Anschluß des Rechners erfolgen.

Ebenso sind heute ganze Systeme von Anlagen aktuell. Diese beginnen mit der einfachen Fernübertragung von einer Außenstelle zu einem Rechner, über das Zusammenspiel mehrerer eventuell räumlich getrennter Rechner zum ausgetragenen „time sharing“, wobei ein leistungsfähiger zentraler Rechner mit einer Reihe von Außenstellen so in Verbindung steht, daß die einzelnen Außenstellen unabhängig voneinander in direktem laufendem Kontakt mit dem Rechner stehen, ohne daß die Programmabläufe sich gegenseitig stören. Auch hier liegen nicht nur technische Probleme vor, sondern in erster Linie Probleme der Programmierung. Rein organisatorische, übergeordnete Programme müssen das flüssige Zusammenspiel der verschiedenen Teile einer Anlage regeln. Meist muß auch noch verhindert werden, daß unbefugte Zugriffe zu gesperrten Speicherbezirken erfolgen können. Auch das Problem der Sicherheit ist nicht nur ein technisches Problem, sondern auch ein Problem der Programmierung. Der Gedanke, daß ein Ersatzrechner die Aufgabe eines ausgefallenen Rechners im Rahmen einer Prozeßsteuerung übernehmen soll, erfordert eine sorgfältig vorbereitete Organisation mit Vorrangprogrammen, Fehlerbegrenzungs-Programmen, Umstellungs-Programmen usw.

Alle diese Verflechtungen haben auch dazu geführt, daß man nach einem neuen Verhältnis des Menschen zur Maschine sucht. Die verwirrende Vielfalt der Organisation und Programmierung übersteigt oft die Leistungsfähigkeit der mit der Maschine zusammenarbeitenden Menschen. Die ersten programmgesteuerten Maschinen ließen sich noch leicht in ihrer Wirkung überblicken. Die Erlernung eines besonderen Befehlscode war z.B. bei den Geräten Z 3, Z 4 kaum erforderlich. Mit der Steigerung der Leistungsfähigkeit der Anlagen wurde diese Aufgabe jedoch immer schwieriger. Der „Befehlscode“ beherrscht und beherrscht noch die Welt der Computer. Der Mensch paßt sich den Eigenarten der Maschine an, denn die vercodete Sprache ist die Sprache, welche die Maschine am besten versteht. Heute kommt man dahin, zu verlangen, daß die Maschine sich dem Menschen anpaßt. Das Schlagwort von der „man machine interaction“ eröffnet neue Perspektiven. Der Mensch soll Mensch bleiben und die Maschine sich in der Kommunikation dem Menschen anpassen. Dies erfordert Ein- und Ausgabegeräte, die mehr der Arbeitsweise des Menschen angepaßt sind, wie Geräte zur grafischen Darstellung usw. Auch die verwendeten Formelsprachen sollen mehr der Arbeitsweise des Menschen entsprechen. Hier ist ein umfangreiches Neuland, und nach diesem Überblick über die Entwicklung der programmgesteuerten Rechenmaschine haben wir wohl ein ungefähres Gefühl dafür, welche Fülle von Problemen hier noch im Fluß ist und welche Arbeit für die Zukunft noch zu leisten ist.

nach diesem Prinzip gebaut. Der Vorteil der einfachen Rechenwerke wird allerdings praktisch weitgehend durch die Nachteile aufgewogen. Innerhalb der zugehörigen Speicherwerke ist ja die Zugriffsmöglichkeit zu jedem einzelnen Bit erforderlich, wodurch insbesondere die Auswahlrichtungen keineswegs einfacher werden. Die Länge der Programme ist sehr groß, was ebenfalls ein Nachteil ist, wenn etwa umfangreiche Rechnungen durchgeführt werden sollen.

Gibt es nun zwischen diesen beiden Extremen: Rechenwerk mit ausgebauter Arithmetik und Rechenwerk nur mit Ja-Nein-Werten einen eleganten Mittelweg? Die Antwort lautet: Ja. In fast allen modernen Rechenanlagen wählt man einen solchen Kompromiß. Als Informationseinheit dient, wie bei der Babbage-Maschine, ein „Wort“, also im allgemeinen eine arithmetische Zahl oder ein Befehl. Als Grundrechenoperationen können z.B. gewählt werden: Addition, Subtraktion, Linksverschiebung, Rechtsverschiebung und Intersektion. Letztere besteht in paarweiser Konjunktion zwischen den zugeordneten Ziffern zweier solcher als Binärzahlen aufgefaßter Wörter. Sie kann gut zum „Herausschneiden“ von Informationsteilen aus dem Wort verwendet werden. Nach diesem Prinzip sind z.B. Rechengерäte wie Z 22, Z 23, Z 25 und viele andere gebaut. Dabei werden die arithmetischen Operationen aus den genannten Grundoperationen aufgebaut. Hierzu ist ein sogenanntes „Grundprogramm“ erforderlich, welches in gewissem Maße zur Grundausrüstung der Maschine gehört, wenn es auch nur den Inhalt von Speicherzellen darstellt und im Prinzip von Maschine zu Maschine frei gewählt werden kann.

Bei all den bisherigen Überlegungen haben wir nur den Kern der eigentlichen programmgesteuerten Rechenmaschine betrachtet. Zur Gesamtanlage gehört aber wesentlich mehr. Zunächst braucht man Mittel, um Daten in die Maschine einzugeben und wieder herauszuholen, sogenannte Ein- und Ausgabegeräte. Bei den ersten Geräten der Pionierzeit wurde dieses Problem vernachlässigt, da man zunächst überhaupt froh war, daß Programme automatisch abließen. Bald stellte sich jedoch heraus, daß die hohe Leistungsfähigkeit, insbesondere auch der elektronischen Rechengерäte, auch entsprechend schnelle Ein- und Ausgabegeräte erfordert. Es setzte ein Wettlauf zwischen Elektronik und Mechanik ein. Die gute alte Mechanik ist keineswegs durch die Elektronik abgelöst worden. Diese hat sie vielmehr zu immer größeren Leistungen angespornt. Einst waren Lochstreifenleser im Fernschreibwesen mit 8 Zeichen/Sek. üblich, heute benötigen wir solche mit 1000 Zeichen/Sek., und auch das ist noch zu langsam. Schwieriger war es, schnellere Drucker zu bauen, die die von der Rechenmaschine herausgeworfenen Zahlenkolonnen mit genügender Geschwindigkeit herausgeben konnten.

Zu diesen eigentlichen Ein- und Ausgabegeräten kommen andere Zusatzgeräte wie Magnetbandgeräte, Plattenspeicher usw. Alle diese Geräte bezeichnet man als Peripheriegeräte. Heute verursachen diese Peripheriegeräte oft das Mehrfache an Aufwand in bezug auf die Entwicklung, Preis und Wartung, als der eigentliche Rechner.

All das hat natürlich auch seine Rückwirkungen auf die Programmierung und die Formelsprachen. Man erkennt heute, daß mit einer guten Programmsprache nicht nur der eigentliche Rechenablauf, sondern auch gerade dieses Zusammenspiel des Rechenwerks mit den verschiedenen Peripheriegeräten exakt formulierbar sein muß.