

INTELLIGENCE REPORT

INTELLIGENCE REPORT

50X1-HUM

NOTE: Reproduction of this document in whole or in part is prohibited, if secret or top secret, except with permission of the issuing office. All requests for authority to reproduce will be directed to the Assistant Chief of Staff or Intelligence, Department of the Army.

DEPARTMENT OF THE ARMY

NOTE: This document contains information affecting the national defense of the United States within the meaning of the espionage laws, Title 18, U.S.C., Secs. 793 and 794, the transmission or revelation of which in any manner to an unauthorized person is prohibited by law.

50X1-HUM

(Classification)
C O N F I D E N T I A L

COUNTRY **East Germany**

REPORT

SUBJECT **The Zeiss Digital Computer "ZRA-1" (U)**

DATE OF REPORT

NO. OF PAGES **1**

50X1-HUM

DATE OF INFO
PLACE & DATE ACQ.
EVALUATION SOURCE
SUMMARY

(UNCLASSIFIED)

This report pertains to the Zeiss Digital Computer "ZRA-1".

50X1-HUM

U N C L A S S I F I E D

2 Inclosures:

- 1. Photo of Brochure (U) (To DA and as indicated above) UNCLASSIFIED
- 2. Negatives of Incl 1 (U)(To DA only) UNCLASSIFIED

REPORT

This report forwards a photostatic copy of an eight-page brochure in German on subject computer developed by WEB Carl Zeiss JENA. The brochure gives a detailed technical description of the computer.

50X1-HUM

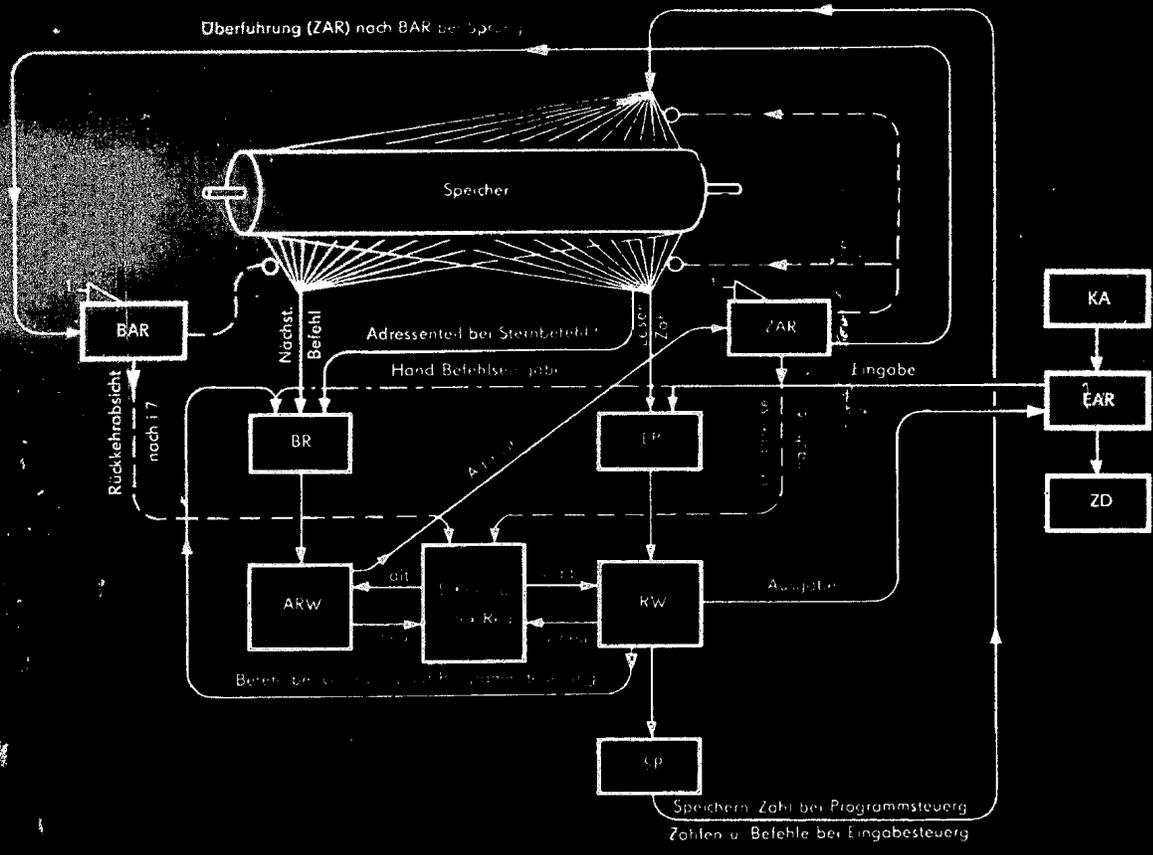
(Classification)
C O N F I D E N T I A L

50X1-HUM

INTELLIGENCE REPORT

INTELLIGENCE REPORT





2

Einleitung

Ein Rechenautomat erfordert für die Bewältigung eines Rechenproblems die Aufstellung eines Rechenprogramms, bei dem die einzelnen durchzuführenden arithmetischen oder organisatorischen Operationen mit Adressen zu versehen sind, unter denen die benötigten Operanden der Maschine zugänglich werden. Die Aufstellung eines solchen Rechenprogramms erfordert einen erheblichen Aufwand an Zeit und Konzentration, so daß ein moderner Automat so eingerichtet sein muß, daß diese Programmierungsarbeit so rationell wie möglich durchgeführt werden kann. Hierzu gehört vor allem, daß ein Rechenprogramm möglichst weitgehend aus bereits ausgearbeiteten und allgemein verwendbaren Teilprogrammen – sogenannten Bibliotheksprogrammen – zusammengesetzt werden kann, um die für die spezielle Aufgabe zu leistende Programmierungsarbeit auf ein Mindestmaß zu beschränken.

Im VEB Carl Zeiss ist ein neuer Rechenautomat mittlerer Geschwindigkeit entwickelt worden, der durch eine Reihe neuartiger Einrichtungen besonders für die Bearbeitung wissenschaftlich-technischer Aufgaben geeignet ist. Bei der Entwicklung wurde das Hauptaugenmerk darauf gerichtet, eine Struktur zu finden, die eine bequeme Programmierungsarbeit ermöglicht, da wir in dieser den wirklichen Engpaß für den rationalen Einsatz elektronischer Rechenautomaten erblicken. Dabei wurden außer theoretischen Untersuchungen die Erfahrungen verwertet, die bei der Erledigung von wissenschaftlich-technischen Aufgaben aller Art mit dem seit über 3½ Jahren im Dreischichtenbetrieb laufenden Rechenautomat „Oprema“ gewonnen wurden. Die weitere Zielsetzung, ein Höchstmaß an Betriebssicherheit zu erreichen, führte uns dazu, die gesamte logische Struktur ausschließlich auf der Ferritkernbasis aufzubauen. Von diesem mit „ZRA 1“ bezeichneten Rechenautomaten be-

findet sich ein vereinfachtes Funktionsmuster seit Monaten in Dauererprobung. Die erste vollständige Anlage, die zur Verstärkung des Rechenzentrums im VEB Carl Zeiss JENA und zur Ausbildung der für die weiteren gefertigten Anlagen einzusetzenden Kader dienen soll, wird in Kürze fertiggestellt sein. Die Fertigung weiterer Anlagen sowie die Vorbereitungen für eine größere Serie sind bereits angefallen.

Von den erwähnten Zielsetzungen ausgehend, soll in der vorliegenden Veröffentlichung versucht werden, dem am Einsatz elektronischer Rechenautomaten für vorwiegend wissenschaftlich-technische Probleme interessierten Leser ein solches Bild über die strukturellen Grundlagen dieser Neuentwicklung zu vermitteln, daß er die sich dem Benutzer darbietenden Vorteile richtig einzuschätzen vermag. Weitere Veröffentlichungen sind geplant, in denen die mathematischen Grundlagen, Befehlsstruktur und Programmierungsregeln sowie technische und technologische Einzelheiten dieses Rechenautomaten eingehender behandelt werden sollen.

Aufbau

„ZRA 1“ ist ein digitaler, im Dualsystem arbeitender programmgesteuerter Rechenautomat. Die Eingabe der zu verarbeitenden Zahlen und die Ausgabe der Resultate erfolgt dezimal. Er besteht aus folgenden Baueinheiten: Bedienungspult mit Ein- und Ausgabeaggregat, Magnettrommelspeicher, Leitwerk mit Adressenrechenwerk und Indexregistern, Rechenwerk.

Die Eingabe der Rechenprogramme und Zahlenwerte erfolgt mittels Lochkarten üblichen Formats, auf denen Befehlszeilen und Zahlen in gemischter Folge zeilenweise gelocht sind. Eine Lochkarte kann bis zu 12 Zeilen aufnehmen. Jede Dezimalziffer wird als Tetrade in direkter

2

Dualverschlüsselung dargestellt. Die Abtastgeschwindigkeit beträgt 80 Karten pro Minute, entsprechend 960 Programmzeilen pro Minute.

Zur Speicherung der eingegebenen Programme dient eine Magnettrommel mit einer Kapazität von 4096 Worten. Die Drehzahl der Trommel liegt bei 12000 Umdrehungen pro Minute. Der gesamte Speicher kann in wenig mehr als 4 Minuten durch die Eingabe gefüllt werden. Die Ausgabe der Rechenergebnisse erfolgt mittels Zeilendrucker mit maximal 2,5 Zeilen pro Sekunde. Alle Register des Rechen- und Leitwerkes, alle Steuerketten und logischen Verknüpfungen sind aus Schaltkreisen mit Ferritkernen von nahezu rechteckiger Hystereseschleife aufgebaut.

Den zwei Magnetisierungszuständen der Ferritkerne sind die Dualen „0“ bzw. „1“ zugeordnet. Die Kreise werden mit einer Frequenz von 200 kHz in Serie durchlaufen, wobei die Treibimpulse durch Elektronenröhren erzeugt werden, deren Steuerung aus Synchronisierungsspuren des Trommelspeichers erfolgt. Alle Schaltkreise enthalten Germaniumdioden, die das Übergreifen der Energie eines Kreises auf Nachbarkreise verhindern.

Elektronenröhren finden im „ZRA 1“ grundsätzlich nur Verwendung zur Erzeugung der Magnetisierungsimpulse sowie zur Verstärkung der in den „Leseköpfen“ des Speichers induzierten elektrischen Leistungen, bevor diese Magnetkerne erregen, die zur Überleitung der Befehle in das Leitwerk und der Zahlen in das Rechenwerk dienen. Die Funktion der zur Impulserzeugung dienenden Elektronenröhren wird ständig durch Kontrolllampen überwacht, während zur Kontrolle der Verstärkeröhren ein Prüfzeichen dient, welches vor der Einspeicherung in den Trommelspeicher vom Automaten selbsttätig in jedes Befehls- und Zahlenwort eingesetzt und beim Auslaufen aus dem Speicher durch Quersummenbildung wieder geprüft wird. Auf diese Weise wird auch ein kurzzeitiges Versagen einer Elektronenröhre sofort erkannt. Durch diese Maßnahme wird eine hohe Betriebszuverlässigkeit des Automaten erreicht.

Struktur

Die **Abbildung 2** läßt die wichtigsten strukturellen Merkmale des Rechenautomaten erkennen. Das **Hauptrechenwerk** (RW) arbeitet unabhängig vom Trommelspeicher und bezieht die benötigten Operanden aus **Schnellregistern**. Diese nehmen eine volle Wortlänge von 48 Bits auf.

Ihr Inhalt kann sowohl vom Hauptrechenwerk als auch vom **Adressenrechenwerk** (ARW) verändert werden, da die Schnellregister zugleich auch als **Indexregister** dienen. Die Verbindung des Rechenwerks mit dem Trommelspeicher erfolgt über zwei Pufferregister, den **Eingangspuffer** (EP), der die „gelesenen“ Zahlen aufnimmt, und den **Speicherpuffer** (SP), der die abzuspeichernden Zahlen aufbewahrt. Eingabe und Ausgabe geschehen über das **Ein- und Ausgaberegister** (EAR), mit dem der **Kartenabtaster** (KA) und der **Zeilendrucker** (ZD) verbunden sind.

Zum Einholen der Befehle aus dem Speicher in das **Befehlsregister** (BR) dient das **Befehlsaufrufregister** (BAR), welches einen **Koinzidenzvergleich** enthält, der den Auslauf aus dem Speicher freigibt, sobald Koinzidenz

zwischen der anstehenden Rufadresse und der augenblicklicheintreffenden Zellennummer eintritt. Das **Zahlenuufrufregister** (ZAR) steuert Ein- und Abspeichern der Zahlen. Dem ZAR ist ebenfalls ein Koinzidenzvergleich zugeordnet. Weitere Einzelheiten über die Struktur gehen aus der Beschreibung der Operationen und Arbeitsweise der Maschine hervor.

Befehle und Operationen

Der „ZRA 1“ besitzt zwei Steuerungsarten, von denen die eine unter der Regie des im Trommelspeicher niedergelegten Programmes geschieht, die andere unter der Regie des Eingabeaggregates. Bei der maschinellen Verarbeitung der Programme kann ein automatisches Wechselspiel zwischen beiden Regiearten eintreten. In Analogie zu den beiden Steuerungsarten soll deshalb zwischen **Programmbefehlen** und **Programmoperationen** einerseits und **Eingabebefehlen** und **Eingabeoperationen** andererseits unterschieden werden. Die den automatischen Wechsel bewirkenden **Übergangsbefehle** und **Übergangsoperationen** sollen dabei der Steuerungsart zugewiesen werden, von der sie ausgehen. Alle durch Befehle ausgelösten Programm- und Eingabeoperationen sind im Automaten fest verdrahtet. Durch sie werden Zahlen mit festem und beweglichem Komma direkt verarbeitet. Zahlen und Befehlswoorte umfassen je 12 Tetraden (48 Bits).

Bei Zahlen mit festem Komma werden 11 Tetraden durch die Dezimalziffern besetzt, während die 12. Tetrade für Vorzeichen und 2 Markierungszeichen (Q_1, Q_2) vorgesehen ist. Eine Stelle bleibt für die Aufnahme des erwähnten Prüfzeichens frei. Zahlen mit festem Komma müssen im Bereich $D \leq |Z| < 1$ gehalten werden. Wird dieser Bereich durch die Rechnung überschritten, so stoppt der Automat. Für Zahlen mit beweglichem Komma stehen für die Mantisse (m) 9 Tetraden (entsprechend 9 Dezimalziffern) und für den Exponenten $1\frac{1}{2}$ Tetraden zur Verfügung.

Die Mantisse ist im Bereich $0,1 \leq m < 1$ zu halten. Auch im Verlauf der Rechnung werden die Mantissen vom Automaten ständig in diesen Bereich gebracht (normiert) unter Angleichung der Exponenten. Der Exponent (e) ist um 20 erhöht einzugeben, so daß er durch diese Transformation für den zulässigen Bereich $10^{-19} \leq e \leq 10^{+10}$ in den immer positiven Bereich $1 \leq e \leq 39$ zu liegen kommt. Die 12. Tetrade entspricht derjenigen der Zahlen mit festem Komma.

Im Befehlswort bleibt die vorderste Stelle für das Prüfzeichen frei, die nächste Stelle ist für eventuelle Markierung des Befehls vorgesehen. Die restlichen $11\frac{1}{2}$ Tetraden sind in 4 Bereiche unterteilt, die als **Rechenteil**, **Testteil**, **Transportteil** und **Adressenteil** bezeichnet werden und deren Bedeutung aus der nachstehenden Beschreibung der Programmoperationen zu ersehen ist.

3

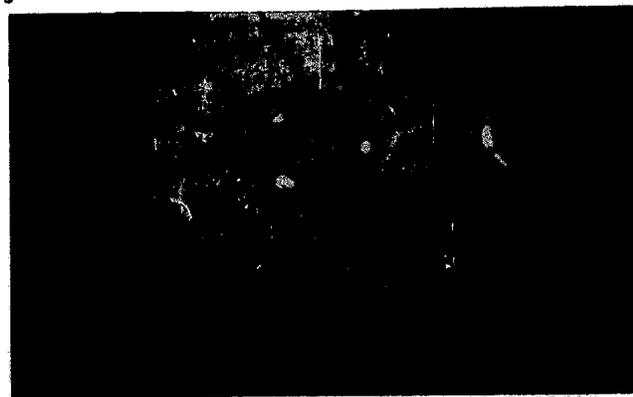


Bild 1: Prüfen von Kernbausteinen (Titelbild)

Bild 2: Strukturbild ZRA 1. Die Abkürzungen bedeuten: RW = Hauptrechenwerk, ARW = Adressenrechenwerk, BR = Befehlsregister, BAR = Befehlsaufrufregister, ZAR = Zahlenuufrufregister, EP = Eingangspuffer, SP = Speicherpuffer, EAR = Ein- und Ausgaberegister, KA = Lochkartenabtaster, ZD = Zeilendrucker

Bild 3: Ferritkern bewickelt und unbewickelt

Bild 4: Serienfertigung der Kernbausteine (S. 6)

Programmoperationen

Im Befehlswort sind den Befehlen Adressen zugeordnet, wodurch die Operanden bestimmt werden, mit denen die ausgelösten Operationen durchzuführen sind.

Der **Rechenteil** enthält die Befehle für alle Operationen, die im Bereich „Rechenwerk – Schnellspeicher“ durchgeführt werden. Sie mögen als **Rechenoperationen** bezeichnet werden. Für alle Operationen mit zwei Operanden dient der Inhalt des zum Rechenwerk gehörenden **Resultatregisters (RK)** als erster Operand, während der zweite Operand sich auf den Inhalt des Schnellregisters „s“ bezieht, dessen Adresse „s“ ebenfalls im Rechenteil angegeben ist. Ist $s = 0$, so dient als zweiter Operand der Inhalt des Eingangspuffers (EP), der in diesem Zusammenhang (aber nicht allgemein) als Schnellregister Nummer 0 aufzufassen ist. Zur Unterscheidung, ob die Rechenoperationen auf Zahlen mit festen, oder beweglichem Komma anzuwenden sind, dient ein ebenfalls im Rechenteil enthaltenes Kennzeichen. Zu den Rechenoperationen gehören **arithmetische** und **logische Operationen, Verschiebungen, innerer Transport** sowie obendrein die **Ausgabeoperationen**, da die Zahlen vor dem Ausdrucken im Rechenwerk automatisch rückkonvertiert werden.

Die **arithmetischen Operationen** sind: **Algebraische Summe, Multiplikation** und **Division**. Dabei ist in üblicher Weise der Quotient in der Form $a : b$ vorgesehen, wenn a den ersten und b den zweiten Operanden bezeichnet. Da es für das Programmieren oft bequem ist, den reziproken Wert $b : a$ zur Verfügung zu haben, wurde auch diese Operation direkt in die Befehlsliste übernommen. Diesen arithmetischen Operationen ist eine ebenfalls im Rechenteil enthaltene **Rechenzeichen-Tetrade** zugeordnet, die jedem Operanden ein Rechenzeichen (+ oder —) zuweist und außerdem festlegt, ob der Operand als **relative** Zahl aufzufassen oder sein **absoluter** Betrag zu verwenden ist. Um aus Zahlen oder Befehlen gewisse Wortteile herausziehen oder in diese einfügen zu können, sind **logische Operationen** und **Verschiebungen** vorgesehen. Zur Überführung eines Operanden aus einem Schnellregister (einschließlich EP) in das Rechenwerk dient die Operation **Übertragen**, während das Abspeichern eines Resultates aus dem Rechenwerk in ein Schnellregister (EP anschließend) die Operation **inneres Speichern** bewirkt. Dieser Befehl **inneres Speichern** kann auch als Zusatzbefehl zu jeder anderen Rechenoperation gegeben werden, wobei das Resultat in dasjenige Schnellregister zurückgespeichert wird, aus dem der zweite Operand entnommen wurde. Der Speicherzusatzbefehl ermöglicht vielfach eine wesentliche Verkürzung der Programme. Durch den **Druckbefehl** wird der Inhalt des RR in Dezimalform übersetzt und ausgedruckt.

Der **Testteil** enthält die Testoperationen, die gewisse Merkmale an Operanden feststellen, Größenvergleich zwischen zwei Worten durchführen oder den Zustand gewisser Testschalter ermitteln und das Ergebnis der Testung in einer zentralen Testzelle abspeichern. Der Zustand dieser Testzelle bestimmt dann bei **bedingten** Operationen, ob diese ausgeführt werden oder unterbleiben und greift auf diese Weise in den weiteren Verlauf der Rechnung ein. Für die Programmierung ist zu beachten, daß ein Testergebnis frühestens von der nächsten Befehlszeile verwertet werden kann. Es wird damit eine grundsätzliche Trennung der bedingten Operationen von den gleichzeitig durchzuführenden anderen Operationen erreicht und eine sofortige Entscheidung erzielt. Welche Vorteile diese Organisation bietet, ist aus dem Abschnitt Arbeitsweise zu ersehen.

Da der Inhalt der Testzelle bis zu einer neuen Testung erhalten bleibt, kann sie auch durch mehrere Testbefehle hintereinander abgefragt werden. Ein **Rechentest** fragt das Vorzeichen des Resultates nach Durchführung einer in der gleichen Programmzeile enthaltenen Rechenoperation ab. Ein **erster Indextest** prüft, ob der Inhalt desjenigen Indexregisters gleich Null ist, dessen Adresse im Adressenteil angegeben ist. Ein **zweiter Indextest** stellt fest, ob der Inhalt des befragten Indexregisters kleiner ist als das im Adressenteil enthaltene **Datum**. Die vor-

stehenden Indexteste dienen hauptsächlich zur Überwachung der Durchläufe in Induktionsschleifen. Ein **dritter Indextest** vergleicht den Inhalt des durch die Indexregisteradresse bestimmten Indexregisters mit demjenigen eines Schnellregisters, das durch die Schnellregisteradresse gekennzeichnet ist. Dieser Test kann auch als Rechentest zur Kontrolle von Iterationsschleifen dienen. Dabei muß das Indexregister die vorgegebene Genauigkeit enthalten, während im Schnellregister die erreichte Genauigkeit dagegengehalten wird. Drei **Q-Zeichenteste** überwachen das Auftreten der entsprechenden Markierungen und schließlich beziehen sich zwei **Schalterteste** auf die Stellung zweier Schalter A und B des Bedienungspultes, eine Maßnahme zur Erleichterung des Prüfens der Programme, auf die später noch eingegangen wird.

Der **Transportteil** enthält die Befehle für **Transport-, Sprung- und Stop-Operationen**. Die eigentlichen **Transportoperationen** „Lesen“ und „Speichern“ vermitteln den Verkehr zwischen dem Trommelspeicher und den mit dem Rechenwerk in Verbindung stehenden Pufferregistern EP und SP. Zusatzbefehle können das **Setzen von Q-Zeichen** an Zahlen auslösen, bevor diese in den Speicher gelangen. Als **Sprungoperationen** sind vorhanden: **unbedingter Sprung, bedingter Sprung bei erfülltem Test** und **bedingter Sprung bei nicht erfülltem Test**. Werden die bedingten Sprünge nicht ausgeführt, so wird die Rechnung mit der nächsten Programmzeile fortgesetzt. Die Sprungoperationen können mit einem Zusatzbefehl, dem sogenannten **R-Befehl**, versehen werden, der aber auch als selbständiger Befehl gegeben werden kann. Der **R-Befehl (Rückkehrabsicht)** bewirkt, daß der Inhalt des BAR nach dem Indexregister Nummer 7 überführt wird. In Verbindung mit einem Sprungbefehl hält der Zusatzbefehl „R“ die Absprungstelle fest und vermittelt so die Rückkehr in den Programmteil, von dem der Absprung erfolgt ist. Die **Stop-Operationen** unterbrechen den Programmablauf, falls sie ausgeführt werden. Die Programmzeile, die den Stop-Befehl enthält, wird noch abgearbeitet und die nächste Programmzeile in das BR geholt. Vorgesehen sind: **unbedingter Stop, bedingter Stop bei erfülltem Test** und **bedingter Stop bei nicht erfülltem Test**. Schließlich ist im Transportteil noch der Befehl **Übergang auf Eingabesteuerung (UESt)** vorgesehen. Bei Vorliegen dieses Befehls wird die den Übergangsbefehl enthaltene Programmzeile noch abgearbeitet, sodann die Programmsteuerung unterbrochen und die Eingabesteuerung in Tätigkeit gesetzt.

Transport- und Sprungoperationen entnehmen ihre Adresse aus dem **Adressenteil**. Dieser enthält die **Adressenoperationen**, die **Indexregister-Adresse** und das **Datum**. Grundsätzlich wird die Adresse gebildet als Summe aus dem Datum und dem Inhalt des angegebenen Indexregisters. Bei fehlender Indexangabe reduziert sich die Adresse auf das Datum. Adressenoperationen sind: **Indexoperation** und **Sternoperation**.¹⁾ Sie können einzeln oder kombiniert auftreten und werden gewöhnlich als Zusatzoperationen zu Transport- und Sprungoperationen gegeben.

Indexoperationen und Adressenbildung werden im Adressenrechenwerk ausgeführt; dabei ist zu beachten, daß die Adresse für den Zahlenaufufr noch mit dem alten Inhalt des Indexregisters gebildet wird, d. h. bevor dieser durch eine evtl. verlangte Indexoperation verändert wird. Es gilt die Regel:

$$\text{Adresse} = [i]_{\text{alt}} + D \Rightarrow [\text{ZAR}] \quad (D = \text{Datum})^2)$$

Indexoperationen sind:

zyklisch um 1, wodurch $[i]_{\text{alt}} + 1 \Rightarrow [i]_{\text{neu}}$ gebildet wird.

¹⁾ Diese Bezeichnung für „Adresse von Adresse“ schließt sich den Veröffentlichungen über die Münchener Rechenautomatenentwicklung „PERM“ an.

²⁾ [i] ist zu deuten: Inhalt von i; das Symbol \Rightarrow ist zu lesen „ergibt“.

Dies ist die am häufigsten verwendete Indexoperation vorwiegend für Adressänderungen 3. Art, wenn bei jedem Schleifendurchlauf des Programmes die folgende den in einem Block gespeicherten Zahlen aufgerufen werden soll.

Zyklisch um D bewirkt $[i]_{alt} \rightarrow D + [i]_{neu}$

Anfangswert Datum bewirkt $D \rightarrow [i]_{neu}$

(In diesem Falle gilt abweichend Adresse = Datum).

Die **Sternoperation (Adresse von Adresse)** bewirkt:

Der Adressenteil der den Sternbefehl enthaltenden Befehlszeile wird ersetzt durch den Adressenteil des Inhaltes derjenigen Speicherzelle, deren Adresse gleich „ $[i]_{alt} + D$ “ ist; dabei entstammen i und D dem Adressenteil mit dem Sternbefehl. Enthält dieser Adressenteil eine Indexoperation, so wird diese durchgeführt, ehe der neue Adressenteil in das BR gelangt, womit also nunmehr das Indexregister einen neuen Inhalt erhalten hat. Die Sternoperation ist ein wesentliches Faktum für die automatische Adressensubstitution.

Eingabeoperationen

Die Befehle für die Eingabesteuerung sind auf der Lochkarte vor dem eigentlichen Befehlswort in der gleichen Zeile angeordnet. Sie werden von dem Eingabeaggregat abgetastet, gelangen aber nicht in das Rechenwerk und den Speicher, sondern lösen unmittelbar die Eingabeoperation aus. Dabei treten **Regiebefehl**, **Adressenbefehl** und **Konvertierungsbefehl** kombiniert auf. Der **Regiebefehl** bestimmt, wie die abgetastete Zeile zu verarbeiten ist: bei **Zeile überlesen** wird die Zeile unverarbeitet übersprungen. **Wort eingeben** bewirkt Einspeichern in die durch den Inhalt von ZAR festgelegte Speicherzelle unter Erledigung der Adressen- und Konvertierungsbefehle. **Übergang auf Programm-Steuerung (ÜPSt)** überführt die Zeile – die in diesem Falle ein Befehl sein muß – nach BR, nachdem Adressen- und Konvertierungsbefehle durchgeführt sind, und setzt die Programmsteuerung in Gang.

Der **Adressenbefehl** dient zur Umwandlung der programmierten relativen Adresse in absolute Adresse: **Speichern Leitadresse** hält den Inhalt von ZAR in einem bestimmten Indexregister (Nr. 6) fest. **Adressenänderung 1. Art** addiert den Inhalt dieses Indexregisters zum Datum des einzugebenden Befehlswortes, bevor dieses gespeichert wird. Außerdem ist die Verbindung dieser beiden Instruktionen möglich.

Der **Konvertierungsbefehl** löst die Operationen zur Übersetzung der eingegebenen Dezimalzahlen in Dualzahlen aus: **Konvertieren Datum** (für Befehlswoorte), **Konvertieren Zahl mit festem Komma** und **Konvertieren Zahl mit beweglichem Komma**. Auch die Konvertierungsoperationen sind einschließlich der dabei benötigten Konstanten fest verdrahtet.

Arbeitsweise

Im „ZRA 1“ sind drei Betriebszustände am Bedienungspult durch Schalter einstellbar. Der eingestellte Zustand wird durch farbige Leuchtfelder angezeigt. Die Schalter greifen dabei in gewisse Steuerketten ein, wodurch die entsprechende Arbeitsweise erreicht wird. Bei **Normalbetrieb** wird sukzessiv Befehl auf Befehl abgearbeitet, wobei der Ablauf beliebig oft zwischen Programmsteuerung und Eingabesteuerung wechseln kann. Bei **Prüfbetrieb** stoppt der Automat nach Verarbeiten jeder Befehlszeile. Bei **Handbetrieb** kann der Rechenautomat ohne Verwendung des Trommelspeichers mit oder ohne Lochkarte gesteuert werden. Während jedes Betriebszustandes kann der Automat sich in verschiedenen Stadien befinden, die als **Bereitschaft**, **Programmsteuerung** und **Eingabe- bzw. Kartensteuerung** bezeichnet werden sollen.

Ist der Automat auf den Zustand **Normalbetrieb** eingestellt worden, so befindet er sich zunächst im Stadium der **Bereitschaft**. Durch Drücken

der entsprechenden Starttaste wird er in das Stadium **Eingabesteuerung** oder **Programmsteuerung** gebracht. Im Stadium **Eingabesteuerung** wird die Einordnung der Programmzeilen in den Trommelspeicher sukzessiv durch den Inhalt des ZAR geregelt, der nach jeder erfolgten Eingabe um 1 erhöht wird. Bevor mit der Eingabe begonnen werden kann, muß der Inhalt von ZAR gleich der Adresse derjenigen Speicherzelle gesetzt werden, die die erste Programmzeile aufnehmen soll. Hierzu ist folgende Lochkartenzeile geeignet:

(1)

ÜPSt	ÜEst	a
------	------	---

Hierin ist das Datum a die Adresse der gewünschten Speicherzelle. Diese Lochkartenzeile ist dann als erste abzutasten, also vor das einzuspeichernde Programm zu bringen. Das Stadium Eingabesteuerung findet sein Ende, wenn der Befehl **ÜPSt** vorliegt oder die letzte im Kartenabtaster befindliche Karte verarbeitet worden ist. Bei Eintreten des 2. Falles stoppt der Automat und geht damit wieder in das Stadium Bereitschaft über. Soll nun mit der Durchführung des Rechenprogrammes begonnen werden, so muß vorher der Inhalt des BAR gleich der Adresse a des Speicherplatzes derjenigen Programmzeile gemacht werden, mit der die Rechnung zu beginnen ist. Dies erreicht man durch folgende Lochkartenzeile:

(2)

ÜPSt	Sprung	a
------	--------	---

Diese Zeile kann in einer Lochkarte enthalten sein, die in den Kartenabtaster eingelegt wird, oder sie kann auch in das EAR eingetastet werden. In beiden Fällen wird nach Betätigung der Starttaste **Start Eingabesteuerung** diese Zeile verarbeitet und dadurch auf Programmsteuerung übergewechselt. Wenn die Programmzeile (2) als letzte Zeile im eingegebenen Programm enthalten war, erfolgt der Übergang automatisch ohne Stop. Soll jedoch in diesem Falle die Maschine ebenfalls anhalten, so wird von der ersten Zeile des Rechenprogrammes ein Stopbefehl vorgesehen. In diesem Falle geht der Automat nach erfolgtem Übergang auf Programmsteuerung in das Stadium Bereitschaft über.

Die Verarbeitung einer im Befehlsregister stehenden Befehlszeile erfolgt in zwei Phasen: in der **1. Phase** erfolgt die Bildung der Adresse für den Zahlaufruf im ARW und ihre Überführung nach ZAR unter gleichzeitiger Durchführung der Indexoperationen. Liegt ein Sternbefehl vor, so wird der bisher im BR enthaltene Adressenteil durch einen neuen, vom Inhalt ZAR bestimmten Adressenteil ersetzt, mit dem wiederum (unter Durchführung evtl. Adressenoperationen) eine neue Adressenbildung erfolgt. Dieser Zyklus wiederholt sich gegebenenfalls solange, bis ein sternfreier Befehl vorliegt, der die 1. Phase beendet. Die **2. Phase** beginnt mit dem Entschlüsseln aller nicht im Adressenteil enthaltenen Befehle unter Abfrage der zentralen Testzelle, wonach Steuerbits parallel in die Steuerketten der angesprochenen Operationsketten einlaufen. Die gewählte Struktur läßt nämlich die parallele Verarbeitung der Operationen uneingeschränkt zu, wenn die Programme unter Beachtung der Programmierungsregeln aus sinnvollen Befehlskombinationen aufgebaut sind. Alle Rechenbefehle werden nur mit Operanden durchgeführt, die in den Schnellregistern einschließlich EP enthalten sind. Aus EP – er ist Nahtstelle zweier Operationsbereiche – wird ein zu entnehmender Operand sofort bei Beginn der Rechenoperation abgeholt. Damit ist EP frei, ehe bei einem gleichzeitig anstehenden Lesebefehl eine Zahl aus dem Speicher einlaufen kann. Ebenso wird bei einem „äußeren“ Speicherbefehl der zu speichernde Inhalt des Resultatregisters im Rechenwerk bei Beginn der Operation in den Speicherpuffer geholt, ehe dieser durch die im Rechenwerk durchzuführenden Operationen verändert wird. Da die Indexoperationen bereits in der ersten Phase erledigt wurden, können Indexteste ebenso wie Q- und Schalterteste unabhängig von

anderen Operationen erledigt werden. Eine Ausnahme bildet lediglich der Rechentest, bei dem das Ende einer gleichzeitig durchzuführenden Rechenoperation abgewartet werden muß, ehe das Vorzeichen des Resultates abgefragt werden kann. Da hinsichtlich der bedingten Operationen bereits bei Entschlüsselung der Befehle durch Abfrage der Testzeile entschieden wird, ob sie ausgeführt werden oder nicht, und da das BR nunmehr frei geworden ist, so ist es grundsätzlich möglich, sofort danach die nächste Befehlszeile aufzurufen. Damit aber das Hereinholen der neuen Befehlszeile nicht mit einem evtl. gleichzeitig durchzuführenden Transportbefehl (Lesen oder Speichern) kollidiert, sind zwei Aufrufregister (BAR und ZAR) vorgesehen. BAR erhöht sofort nach Freiwerden von BR seine Inhalt um „1“ und holt den nächsten Befehl ein, während gleichzeitig ZAR die Koinzidenz mit der aufgerufenen Speicherzelle – zwecks Lesen oder Speichern einer Zahl – überwacht. Nur bei Vorliegen eines auszuführenden Sprunges, der aber nie mit einem äußeren Transportbefehl zusammenfallen kann, wird von ZAR der Inhalt nach BAR abgegeben, bevor die neue Befehlszeile eingeholt wird. Durch diese Struktur entsteht zunächst der Vorteil, daß die sonst bei Maschinen mit Trommelspeicher für die Verarbeitung einer Programmzeile erforderlichen zwei „Suchzeiten“ vermieden werden, die die praktisch erzielbare Rechengeschwindigkeit bekanntlich stark beeinträchtigen. Die gewählte Struktur bewirkt aber auch, daß bereits die Adresse für den Zahlenaufruf gebildet und auch die automatische Adressensubstitution durchgeführt, also die Phase 1 für die neue Befehlszeile durchlaufen werden kann, ohne das Ende einer aus der Verarbeitung der vorhergehenden Befehlszeile eingeleiteten Rechenoperation abwarten zu müssen, so daß sich oftmals Phase 1 und Phase 2 (allerdings verschiedener Befehlszeilen) ineinanderschieben. Das Stadium Programmsteuerung wird beendet, wenn ein Stopbefehl oder ein Übergangsbefehl auf Eingabesteuerung auftritt.

Beim Zustand **Prüfetrieb** stoppt der Automat nach Verarbeiten jeder Befehlszeile, und zwar in dem Augenblick, zu dem die nächste Befehlszeile in das Befehlsregister eingelaufen ist. Er bietet die Möglichkeit, das Programm Zeile für Zeile durchzuprüfen. Am Leuchtfeld „Stopursachen“ wird dabei angezeigt, welche Starttaste man als nächste zu drücken hat, um die Verarbeitung des Programms fortzusetzen. Im eingetretenen Zustand Bereitschaft kann durch Auswahlschalter der Inhalt jedes Registers des Rechen- und Leitwerkes in das auf dem Bedienungspult befindliche, an EAR angeschlossene Lampenregister übertragen und angesehen werden. Die Prüfung eines Programmes läßt sich rationell durchführen, wenn bedingte Stopbefehle mit vorangehenden Schaltertesten an geeigneten Programmstellen eingebaut werden. Dann können bereits geprüfte Programmteile im Normalbetrieb schnell durch-

fahren werden. Die für Prüfzwecke eingebauten Stopbefehle werden bei der Verarbeitung des Programms unwirksam gemacht, wenn die Testschalter vor Beginn der Rechnung auf „aus“ gestellt werden.

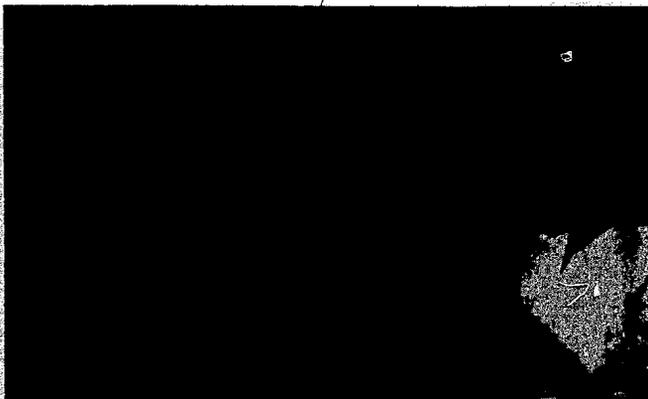
Der Zustand **Handbetrieb** ermöglicht die Durchführung von Geradeausprogrammen ohne Inanspruchnahme der Speichertrommel, also ohne Transportoperationen. In diesem Betriebszustand kann die Steuerung entweder von der Karte aus geschehen oder dadurch, daß die Programmzeilen einschließlich der Kartenbefehle in das EAR eingetastet werden. Eingegebene Zahlen bleiben in diesem Betriebszustand nach der Konvertierung im Resultatregister des Rechenwerkes. Eingegebene Befehlszeilen umgehen das Rechenwerk, laufen direkt in das Befehlsregister ein und werden sofort verarbeitet. Nach Abarbeitung jeder Befehlszeile stoppt der Automat, wonach gegebenenfalls die Registerinhalte besehen werden können. Der Handbetrieb wird für Zwischenrechnungen und Fehlersuche angewendet.

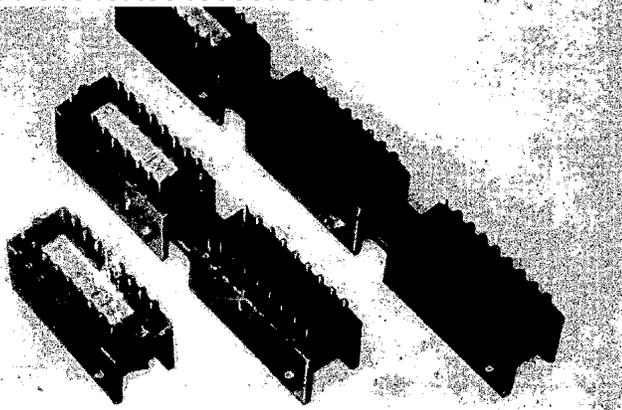
Ist der Automat in Stop gelaufen – sei es, daß ein Stopbefehl vorlag oder eine Bereichsüberschreitung oder ein Fehler eingetreten war – so wird die Stopursache durch ein Leuchtfeld am Bedienungspult angezeigt. Das gilt für alle Betriebszustände.

Zur Programmierung

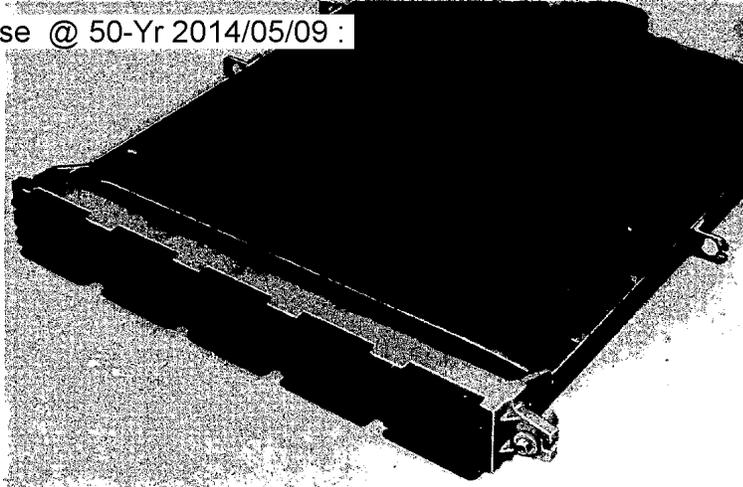
Die Aufgliederung der Befehlszeile in Befehlsgruppen, denen Zusatzbefehle zugeordnet sind, ergibt sinnvolle Kombinationen von etwa 10⁴ Befehlsanordnungen, die dem Automaten eine große Flexibilität in der Programmgestaltung verleihen und eine fast unerschöpfliche Quelle darstellen für Rationalisierung der Programmierungsarbeit. Die Einrichtungen zur automatischen Adressensubstitution, wozu die Indexregister, der R-Befehl und die Verarbeitung der Adressenbefehle in einem speziellen Adressen-Rechenwerk gehören, machen besondere organisatorische Befehlszeilen für den Anschluß eines Unterprogrammes an das Hauptprogramm überflüssig; der R-Befehl hat die Absprungstelle im Indexregister 7 festgehalten. Werden im Hauptprogramm die zur Durchführung eines Unterprogramms erforderlichen Parameter in den dem Absprung folgenden Programmzeilen bereitgestellt, so können diese vom Unterprogramm unmittelbar „gelesen“ werden, wenn kombiniert mit dem Aufruf des Indexregisters Nr. 7 als Datum n gesetzt wird und sich der zu lesende Parameter in der n-ten Zeile hinter der Absprungstelle befindet. Auf die gleiche Weise geschieht die Rückkehr in die n-te Zeile nach der Absprungstelle. Werden im Hauptprogramm nicht die Parameter selbst, sondern deren Adressen bereitgestellt, so sind die Parameter dem Unterprogramm durch Hinzufügen eines Sternbefehls ebenfalls unmittelbar zugänglich. Enthalten die Adressen für die Parameter noch die Angabe einer Indexadresse sowie etwaiger Indexbefehle, so können vom Unterprogramm bei zyklischen Problemen ganze Zahlenblocks verarbeitet werden, ohne daß besondere Befehlszeilen organisatorischer Natur vorgesehen werden müssen. Damit gestatten diese Einrichtungen eine erhebliche Verkürzung der Programme. Die vorgesehenen Tests, insbesondere die Q-Tests, können fast immer anderweitigen Befehlszeilen zugefügt werden. Eine weitere Erleichterung der Programmierungsarbeit bietet die Alternative, die bedingten Operationen auf erfüllten oder nicht erfüllten Test beziehen zu können.

Der Mehraufwand, den die Anordnung zweier Aufrufregister mit je einem Koinzidenzvergleich sowie der beiden Pufferregister erfordert, wird nicht nur durch die Erhöhung der Rechengeschwindigkeit infolge Fortfall einer Suchzeit aufgehoben; er gewährt noch weitere Vorteile. Die automatische Adressensubstitution kann großzügig zur Adressenänderung 2. und 3. Art angewendet werden, ohne wesentliche Erhöhung der Rechenzeit nach sich zu ziehen. Außer dem schon erwähnten Umstand, daß die Substitutionen vielfach gleichzeitig mit noch laufenden Rechenoperationen durchgeführt werden, kommt die damit erreichbare Verkürzung der Programme praktisch einer Steigerung der Rechen-





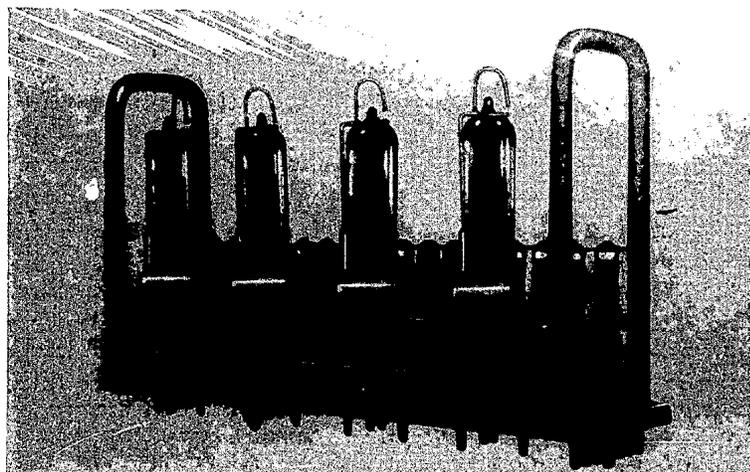
5



6



7



8

geschwindigkeit gleich. Die Anordnung von zwei Aufrufregistern, denen verschiedene Indexregister (Nr. 6 und Nr. 7) zugeordnet sind, um nötigenfalls ihren Inhalt aufzunehmen, gewährt mit Hilfe der beiden Übergangsbefehle – in Verbindung mit der Zuordnungsmöglichkeit anderer Befehle in der gleichen Programmzeile – einen beliebigen Wechsel zwischen Programm- und Eingabesteuerung. Dabei ist in jedem Falle der Anschluß an die andere Steuerungsart sichergestellt, da die erforderlichen Daten aufbewahrt sind. Auf diese Weise können an allen Programmstellen, bei denen das Rechenwerk frei von Operanden ist, neue Zahlen in den Speicher eingeholt werden, um mit diesen die Rechnung fortzusetzen. Damit stellt der Lochkartenabtaster einen äußeren Speicher großer Kapazität dar.

Die gewählte Struktur des Automaten gestattet uneingeschränkt die Verwendung von Bibliotheksprogrammen mit freien und relativen Adressen, so daß darin keine Bestandteile enthalten sind, die ihre universelle Verwendbarkeit einschränken. Auch die Hauptprogramme können mit relativen und Pseudoadressen aufgestellt werden, ohne Rücksicht auf die Speicherplätze der zugehörigen Unterprogramme und Zahlenblocks. Diese Möglichkeiten erleichtern die Programmierungsarbeit erheblich. Mit Hilfe einfacher vorher eingespeicherter Programme werden bei der Eingabe automatisch die Speicherplätze der Anfangszeilen von Teilprogrammen und Zahlenblocks in einem sogenannten „Adressenbuch“ festgehalten, welches vom Automaten während des Eingabeprozesses angelegt wird. Bei der Durchführung des Rechenprogramms kann dann dieses Adreßbuch interpretierend verwendet werden, um mittels der Pseudoadressen die absoluten Adressen zu gewinnen. Es ist aber auch möglich, vor Beginn der Rechnung mit Hilfe eines kompilierenden Programms die absoluten Adressen aus dem

Adreßbuch zu entnehmen und anstelle der Pseudoadressen in das Rechenprogramm einzusetzen.

Zusammenstellung der wichtigsten technischen Daten und strukturellen Besonderheiten vom „ZRA 1“

Ferritkernkreise ermöglichen im Baukastenprinzip die Realisierung der gesamten logischen Struktur. Die ständige automatische Kontrolle der Röhrentreiber auch während des Rechenprozesses bietet hohe Betriebssicherheit der Anlage. Der Einsatz von Regiebefehlen auf der Lochkarte steuert den Eingabeprozess direkt, so daß kein besonderes Eingabeprogramm nötig ist. Der Forderung nach einer Bibliotheksprogrammtechnik wird die mit der Eingabe automatisch verknüpfte Adressenänderung erster Art gerecht. Das automatische Festhalten der Absprungsstelle mittels R-Zusatzbefehl dient bei Adressenänderungen zweiter Art dem Anschluß des Unterprogramms an das Hauptprogramm hinsichtlich Parametern und Rückkehr. Die Kombination „Adresse von Adresse“ mit Indexregistertechnik und zusätzlichen Indexoperationen in einem gesonderten Adressenrechenwerk erledigen Adressenänderungen dritter Art, ohne daß diese in besonderen organisatorischen Befehlszeilen programmiert werden müssen, und verkürzt damit die Programme. Das Prinzip der Steuerketten in Verbindung mit dem gemischten Befehlssystem und

- Bild 5:** Kernbausteine mit Prüfkennzeichen versehen
- Bild 6:** Einschubrahmen mit Kernbausteinen bestückt
- Bild 7:** Serienfertigung der Impulsendstufen
- Bild 8:** Impulsendstufe

getrennten Transportkreisen und Koinzidenzvergleichern für Befehle und für Zahlen gibt parallelen Ablauf von Operationen aus verschiedenen Befehlsgruppen und damit Steigerung der Rechengeschwindigkeit und hohen Befehlsreichtum. Mittels dieser Kettentechnik werden neben arithmetischen Operationen mit festem Komma auch solche mit gleitendem Komma sowie Konvertierung und Rückkonvertierung bei Ein- und Ausgabe durch Kennzeichnung des Befehls ausgelöst ohne äußere detaillierte Programmierung. Das automatische Wechselspiel zwischen Eingabe- und Programmregie ermöglicht unter anderem die Verwendung freier und symbolischer Adressen und bietet das Eingabeaggregat obendrein als Großspeicher an.

Kenndaten für „ZRA 1“

Technik: Ferritkernkreise mit kontrollierten Röhrentreibern, Magnet-trommelspeicher.

Maße: 3 Gestelle, je 2,20 m hoch, 1,20 m tief, 1,30 m breit; Bedienungspult (Schreibtischgröße)

Bauelemente: etwa 770 Treib- und Verstärkerröhren, 12000 Germaniumdioden, 8500 Ferritkerne.

Netz: 220 Volt, etwa 12 KW; Heizspannung zur Schonung der Röhren stabilisiert.

Eingabe: Lochkartenabtaster (Bürsten) 80 Karten zu je 12 Zeilen/min \approx 1000 Programmzeilen/min einschließlich Konvertierung aus Dezimalsystem (festes oder gleitendes Komma) in Dualsystem. Programmierter Stop nach jeder Zeile möglich.

Ausgabe: Blockdrucker, 150 Zahlen/min, einschließlich Rückkonvertierung. Individueller Druckbefehl. Austausch gegen Schnelldrucker später vorgesehen.

Trommelspeicher: 12000 Umdr./min, 4096 Speicherplätze, 3 Bits/mm, 200 kHz, Kopf-Schichtabstand 30μ . 3 Synchronisierungs- und 10 Adressenspuren.

Zahlensystem: Dezimal 11 Stellen, Vorzeichen und 2 Markierungsstellen bei festem Komma; 9 Stellen für Mantisse, Vorzeichen und 2 Markierungsstellen, Exponent 10^{-19} bis 10^{+19} bei gleitendem Komma. In der Maschine 48 Dualstellen.

Befehlssystem: Vier Befehlsgruppen, die fast beliebig kombinierbar sind, 1. Rechenteil mit Schnellregister, 2. Testteil, 3. Transport-Sprung-Stopbefehlsteil, 4. Adressenteil mit Datum, Indexadresse, Indexoperationen. Zusatzbefehle für Rückspeicherung, Rückkehrabsicht bei Sprung, Scheinadresse (automat. Adressensubstitution). Befehlsreichtum vorsichtig geschätzt 10^6 .

Rechengeschwindigkeit: 150 bis 170 Befehlszeilen/sek. Theoretisch bis zu viermal, praktisch doppelt soviel Operationen pro sek durch das gemischte Befehls- und Adressensystem und den Parallelablauf der Operationen: bei festem Komma; Addition 3,8 ms, Multiplikation 7 ms, Division 14 ms. Bei gleitendem Komma; Addition 7 ms, Multiplikation 8 ms, Division 14 ms. Organisatorische Operationen 0,5 bis 2,5 ms. Mittlere Suchzeit 2,5 ms, die sich nur bei Operationsdauer unter 1 ms auswirken kann.

Programmierungsbeispiel: Multiplikation zweier Matrizen 20. Ordnung (Zahlen mit gleitendem Komma) Programm: 17 Befehlszeilen

Eingabe der 80 Eingangszahlen, einschl. Konvertierung \sim 1 Min.
Rechnen und Ausdrucken der 800 Komponenten der Produktmatrix \sim 4 „
Gesamtzeit \sim 5 Min.