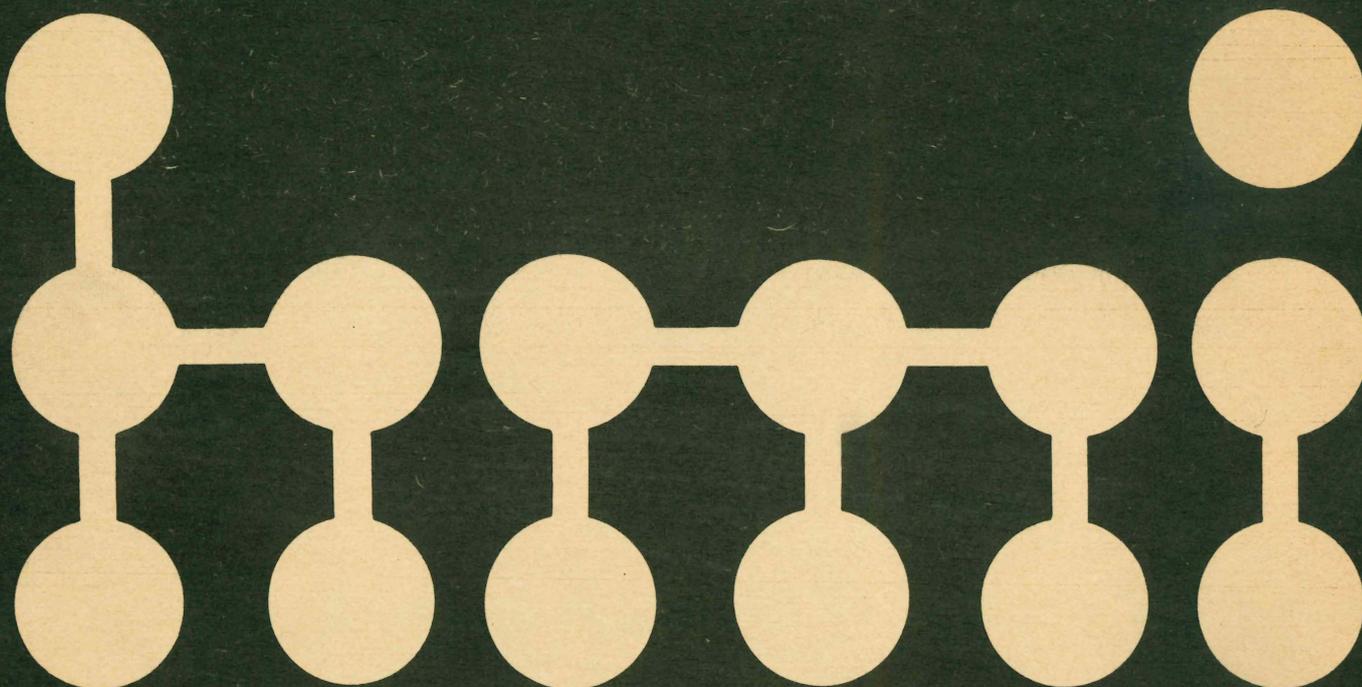


**HMI - B 120**  
März 1972

**Hahn-Meitner-Institut  
für Kernforschung Berlin**



**TAGUNGSBERICHT**

**Anwenderkreis I Siemens-Prozeßrechner  
vom 24. bis 27. Mai 1971 in Berlin**

**Veranstalter: HMI-Berlin, Sektor Elektronik  
Herausgeber: Peter Abend, Renate Bublitz**

**HMI - B 120**  
März 1972

Als Manuskript gedruckt

## BERICHTE DES HAHN-MEITNER-INSTITUTS

Das HAHN-MEITNER-INSTITUT FÜR KERNFORSCHUNG BERLIN GMBH gibt eine Serie von Berichten heraus, in der Forschungsergebnisse mitgeteilt werden und die Jahresberichte des Instituts erscheinen.

Die Berichte werden in der Referate-Zeitschrift „Informationen zur Kernforschung und Kerntechnik“ der Zentralstelle für Atomkernenergie-Dokumentation (ZAED) und in den „Nuclear Science Abstracts“ der USAEC angezeigt und referiert. Sie können von der Institutsbibliothek angefordert werden.

Die Berichte werden in der Reihe HMI-B 1, B 2 usw. fortlaufend nummeriert. (Zusätzlich nummerieren die einzelnen Sektoren und Abteilungen des Instituts ihre Berichte; die Angabe dieses Zusatzes ist bei Bestellungen nicht erforderlich.)

## REPORTS OF THE HAHN-MEITNER-INSTITUTE

The HAHN-MEITNER INSTITUTE FOR NUCLEAR RESEARCH BERLIN GMBH publishes a series of reports, in which research results are reported and annual reports appear.

These reports are listed and abstracted in the Abstract Journal „Informationen zur Kernforschung und Kerntechnik“ of the Zentralstelle für Atomkernenergie-Dokumentation (ZAED) and in the „Nuclear Science Abstracts“ of the USAEC. The reports may be requested from the institute's library.

The reports are numbered continuously in series HMI-B 1, B 2, etc. (In addition, the various sections and divisions of the institute number their reports. Reference to the addition, though, is not essential for ordering.)

**B - NDV 12**

**HMI - B 120**  
**März 1972**

## **TAGUNGSBERICHT**

**Anwenderkreis I Siemens-Prozeßrechner**  
**vom 24. bis 27. Mai 1971 in Berlin**

**Veranstalter: HMI-Berlin, Sektor Elektronik**  
**Herausgeber: Peter Abend, Renate Bublitz**

**Berlin-Wannsee**

## Vorwort

Nachdem sich die Anwender der Siemens-Prozeßrechner des Systems 300 aus dem technisch-wissenschaftlichen Bereich zum ersten Male im Mai 1970 in der Kernforschungsanlage Jülich zum Erfahrungs- und Gedankenaustausch zusammengefunden und beschlossen hatten, einen Anwender-Kreis zu gründen, fand die zweite Tagung im Hahn-Meitner-Institut für Kernforschung Berlin GmbH mit großzügiger Unterstützung des Hauses Siemens statt.

Auch bei diesem Zusammentreffen, an dem etwa 100 Anwender und Mitarbeiter des Hauses Siemens teilnahmen, wurden Probleme und Lösungen bei der Benutzung der Siemens-Prozeßrechner sowohl in Hardware- als auch Software-Gebiet vorgetragen und diskutiert.

Auf der Tagung wurde ein Geschäftsführender Ausschuß gewählt, der u. a. die Aufgaben übernahm, die nächste Sitzung an der Fachhochschule Südost-Westfalen in Paderborn vorzubereiten, einen Programm-Austausch in die Wege zu leiten und Informationen über die verschiedenen Anlagen und Aufgaben der Anwender zu sammeln und zu verbreiten. Der Geschäftsführende Ausschuß wird durch ein Sekretariat bei Siemens in Karlsruhe unterstützt.

Der vorliegende Bericht, der ebenfalls mit Unterstützung des Hauses Siemens herausgegeben wird, enthält die in Berlin gehaltenen Vorträge, soweit sie von den Referenten zur Verfügung gestellt wurden. Allen Autoren soll hier nochmals ein herzlicher Dank ausgesprochen werden.

Wir hoffen, daß Ihnen dieser Bericht bei Ihren zukünftigen Aufgaben dienlich sein wird und Sie veranlaßt, weiterhin im Anwenderkreis aktiv mitzuarbeiten.

*Die Herausgeber*

## **Tagungsprogramm**

SIEMENS-Prozeßrechner-Anwenderkreis I  
Institute und Ausbildungsstätten

im Hahn-Meitner-Institut für Kernforschung Berlin GmbH  
Hörsaal im Hauptgebäude  
24.–27. Mai 1971

## **Montag, 24. Mai 1971**

Der Vormittag steht zur Anreise der Teilnehmer zur Verfügung.

- 14.00**
1. Begrüßung und Tagungsinformation  
K. Zander, HMI Berlin
  2. Tendenzen in der modernen Prozeßrechner-Technik  
H. Offer, Siemens, Karlsruhe
  3. K 404 (Z 43) als Prozeßrechner  
R. Hirzel, Siemens, Karlsruhe
  4. Die Prozeßperipherie für das System K 404  
R. Hirzel, Siemens, Karlsruhe
  5. Das Projekt Bosch Schwieberdingen (Stand, Erfahrungen, Ausbau, Zielsetzung)  
D. Stams, Bosch, Schwieberdingen

Pause etwa 15.30 Uhr, Ende etwa 17.30 Uhr.

Die Vortragsdauer beträgt im allgemeinen etwa 20 Minuten, mit anschließender Diskussion von etwa 10 Minuten.

## **Dienstag, 25. Mai 1971**

- 9.00**
1. Ein graphisches Sichtgerät mit Speicherbildschirm am System 300  
J. Zahn, HMI Berlin
  2. Ein modulares Programmsystem für graphische Darstellungen  
P. Abend, HMI Berlin
  3. Das Display-Terminal TEKTRONIX 4002 am Rechner S 305  
W. Tenten, KFA Jülich
  4. Ein erweitertes Prozeßelement P 3  
G. Straßer, Siemens, Karlsruhe
  5. Eine Magnetbandeinheit für das System 300  
H. Printz, EDS-Systemtechnik, Aachen

**12.15** *Mittagessen*

**14.00** 6. Die Kopplung von Rechnern des Systems 300

L. Gries, Siemens, Karlsruhe

W. Woletz, HMI Berlin

7. Betriebserfahrung und Ausbaupläne mit einer Rechnerkopplung  
S 305 und 4004/55

G. Reimann, Technische Universität München

8. Erfahrungen mit der Rechnerkopplung S 305/PDP-8

W. Tenten, KFA Jülich

9. Eine Datenaustauschsteuerung zwischen S 305 und IBM 1130

H. P. Englmeier, Max-Planck-Institut, München

*Siemens-Prozeßrechner-Anwenderkreis I*

10. Aufgaben und Ziele

K. D. Müller, KFA Jülich

11. Kurzbericht über den Programmaustausch

P. Abend, HMI Berlin

12. Vorschläge des Hauses Siemens

H. Klingler, Siemens, Karlsruhe

13. Diskussion und Beschlußfassung

**18.30** *Einladung* der Firma Siemens AG zum gemeinsamen Abendessen

**Mittwoch, 26. Mai 1971**

**9.00** 1. Fortran IV H-Level für das System 300

H. Mittendorf, Siemens, Karlsruhe

2. PEARL – Eine Prozeßsprache

P. Elzer, Universität Erlangen

H. Mittendorf, Siemens, Karlsruhe

3. Standard-Ein/Ausgabe-Aufrufe für das ORG 300, speziell AEP-Nahtstelle

G. Schlimbach, Siemens, Karlsruhe

4. Ein Magnetband-/Platten-Editor für das System 300

G. Mrozynski, TU Berlin (früher HMI Berlin)

5. Ein modulares Programmsystem zur Vielkanalanalyse mit dem System 300

K.-H. Degenhardt, HMI Berlin

**12.15** *Mittagessen*

**14.00** 6. Ein Tischrechenprogramm für das System 300

K. Weise, PTB Braunschweig

7. Das CAMAC-Instrumentierungssystem

D. Kollbach, HMI Berlin

8. Ein CAMAC-Branch Driver Controller für das System 300  
H. Pangritz, HMI Berlin

9. CAMAC-Instrumentierung von Siemens  
W. Decker, Siemens, Karlsruhe

10. Das Digizet-i-Bausteinsystem  
G. Mett, Siemens, Karlsruhe

**Ende etwa 17.30**

**Donnerstag, 27. Mai 1971**

**9.00** 1. Datenerfassung und Datenauswertung mit Prozeßrechnern bei  
Gaschromatographen und Massenspektrometer  
W. Gold, Siemens, Karlsruhe

2. Das Hybridsystem Siemens 306/Dornier 720  
M. Mall, Dornier, Friedrichshafen

3. Alarmbearbeitungen von Meldungen über Rechner-Anschlußtafeln  
in Laboratorien  
U. Kiefer, Siemens, Karlsruhe

4. Extern gesteuerte Datenversorgung von Prüfautomaten in der  
ESK-Nebenstellentechnik mit dem Rechner S 305  
W. Lorenz, Siemens, Berlin

5. Das integrierte Prüfsystem 70 (IP 70)  
(Systemphilosophie, Prüfautomaten, Anwendung)  
G. Mett, Siemens, Karlsruhe  
W. Kammerer, Siemens, München

**12.15** *Mittagessen*

**14.15** *Ende der Tagung*

Abfahrt des Sonderbusses vom Restaurant zum Stadtzentrum.

## Inhaltsverzeichnis

<i>H. OFFER</i> , Siemens Karlsruhe Vergleichskriterien für Prozeßrechnerzentraleinheiten .....	11
<i>D. STAMS</i> , Robert Bosch GmbH Schwieberdingen Prozeßrechnereinsatz in Laboratorien und Prüffeldern der Kraftfahrzeugausrüstungs- industrie .....	25
<i>J. ZAHN</i> , Hahn-Meitner-Institut Berlin Ein graphisches Sichtgerät für nukleare Experimente mit On-line-Rechner .....	39
<i>P. ABEND</i> , Hahn-Meitner-Institut Berlin Ein modulares Programmsystem für graphische Darstellungen .....	51
<i>W. TENTEN</i> , ZEL/NE der KFA Jülich Das Display-Terminal Tektronik 4002 am Rechner Siemens 305 .....	57
<i>H. PRINZ</i> , EDS-Systemtechnik Aachen Magnetbandgeräteanschluß .....	63
<i>L. GRIES</i> , <i>W. WOLETZ</i> , Siemens AG Karlsruhe, HMI Berlin Die Kopplung von Rechnern des Systems 300 .....	71
<i>HP. ENGLMEIER</i> , Max-Planck-Institut München Eine Datenaustauschsteuerung zwischen S 305 und IBM 1130 .....	77
<i>H. MITTENDORF</i> , Siemens AG Karlsruhe FORTRAN IV H-Level für das Siemens-System 300 .....	87
<i>G. MROZYNSKI</i> , TU Berlin (früher Hahn-Meitner-Institut Berlin) Ein Magnetband-Platten-Editor für das System 300 .....	97
<i>K. H. DEGENHARDT</i> , <i>W. WOLETZ</i> , Hahn-Meitner-Institut Berlin Ein modulares Programmsystem zur Vielkanalanalyse mit dem System 300 .....	101
<i>K. WEISE</i> , Physikalisch-Technische Bundesanstalt Braunschweig Ein Tischrechnerprogramm für Siemens-Prozeßrechner des Systems 300 .....	107
<i>D. KOLLBACH</i> , Hahn-Meitner-Institut Berlin Das CAMAC-Instrumentierungssystem .....	115
<i>W. GOLD</i> , Siemens AG Karlsruhe Datenerfassung und Datenauswertung mit Prozeßrechnern bei Gaschromatographen und Massenspektrometern .....	121
<i>W. LORENZ</i> , Siemens AG Berlin Extern gesteuerte Datenversorgung von Prüfautomaten mit Siemens-Prozeßrechner 305 im Fg-Prüffeld Berlin .....	128
<i>W. KAMMERER</i> , Siemens AG Karlsruhe Das integrierte Prüfsystem 70 .....	137

## **H. Offer**

Siemens AG Karlsruhe

Vergleichskriterien für Prozeßrechner-Zentraleinheiten

# Vergleichskriterien für Prozeßrechner-Zentraleinheiten

## 1. Einleitung

In diesem Vortrag werden Vergleichskriterien für Prozeßrechner-Zentraleinheiten erläutert. Dazu müssen wichtige technische Merkmale dieser Zentraleinheiten betrachtet werden, und zwar die Technologie des Arbeitsspeichers,  
strukturelle Merkmale einer Zentraleinheit,  
die Wortlänge und  
die Befehlsliste,

um dann einen Prozeßrechner Mix-Wert ableiten zu können, der eine Aussage über die Zentralprozessorbelastung liefert.

Im Anschluß daran werden unterschiedliche Unterbrechungsstrukturen vorgetragen sowie die wichtigsten Formen der Abwicklung des E/A-Verkehrs.

Diese Ausführungen sollen durch einige Hinweise zur Aufbautechnik abgerundet werden.

Abschließend sind weitere Vergleichskriterien zu nennen, die für die Beurteilung eines vollständigen Prozeßrechner-Systems hinzugezogen werden müssen.

## 2. Betrachtung der wichtigsten technischen Merkmale von Prozeßrechner-Zentraleinheiten

### 2.1 Technologie des Arbeitsspeichers

Als Arbeitsspeicher werden z. Z. Magnetkernspeicher, Magnetdrahtspeicher, Festwertspeicher und in zunehmendem Maße Halbleiterspeicher eingesetzt (Abb. 1).

Der *Magnetkernspeicher* ist am weitesten verbreitet und arbeitet nach dem sicherlich bekannten Koinzidenzprinzip. Magnetkernspeicher zeichnen sich durch immer kürzer werdende Zykluszeiten aus. Ein mittelschneller Kernspeicher hat bereits eine Zykluszeit von 650 ns bei einem Listenpreis von DM 0,14–0,17/Bit.

Der *Magnetdrahtspeicher* ist entfernt verwandt mit dem Magnetkernspeicher. Es handelt sich hier um einen Speicher mit kurzer Zugriffszeit. Das Grundelement ist hier eine dünne zylindrische Schicht mit bestimmten magnetischen Eigenschaften, die um einen Draht herum angebracht ist (Abb. 2). Dieser Draht dient gleichzeitig als sogenannte Bitleitung, d. h. als Schreib- und Leseleitung. Eine binäre Null oder Eins wird in der zylindrischen, magnetischen Schicht dadurch gespeichert, daß diese in einer ihrer beiden sogenannten weichen Richtungen magnetisiert wird. Soll die Information gelesen werden, so wird durch eine um den Magnetdraht herumgeführte Leitung (Wortleitung) ein Strom geschickt, der die Magnetisierung im Magnetdraht in die sogenannte harte Richtung dreht. Dabei wird in der Bitleitung ein Impuls induziert, an dessen Polarität man erkennen kann, welche der beiden Binärziffern im Magnetdraht gespeichert war.

Magnetdrahtspeicher haben bei einer Lesezeit von 160 ns einen Listenpreis von ca. DM 0,22/Bit.

	L-Preis/Bit	L-Preis/4K	L-Preis/16K	L-Preis/32K
<i>Magnetkernspeicher</i> Zykluszeit 650 ns Zugriffszeit 275 ns Wortlänge 18 Bit	15,5 Pfg.	TDM 16,1	TDM 46,0	TDM 92,0
<i>Magnetdrahtspeicher</i> Lesezykluszeit 160 ns Schreibzykluszeit 200 ns Wortlänge 18 Bit	22,0 Pfg.	—	—	TDM 130,0
<i>Festwertspeicher</i> (Fädelspeicher) Lesezykluszeit 500 ns Wortlänge 16 Bit	16,8 Pfg.	TDM 14,3	TDM 43,8	—
<i>Halbleiterspeicher</i> (dyn. MOS-Speicher) Zykluszeit 580 ns Wortlänge 18 Bit	17,0 Pfg.	TDM 12,5	TDM 50,0	TDM 100,0

Übersicht über L-Preise von Arbeitsspeichern

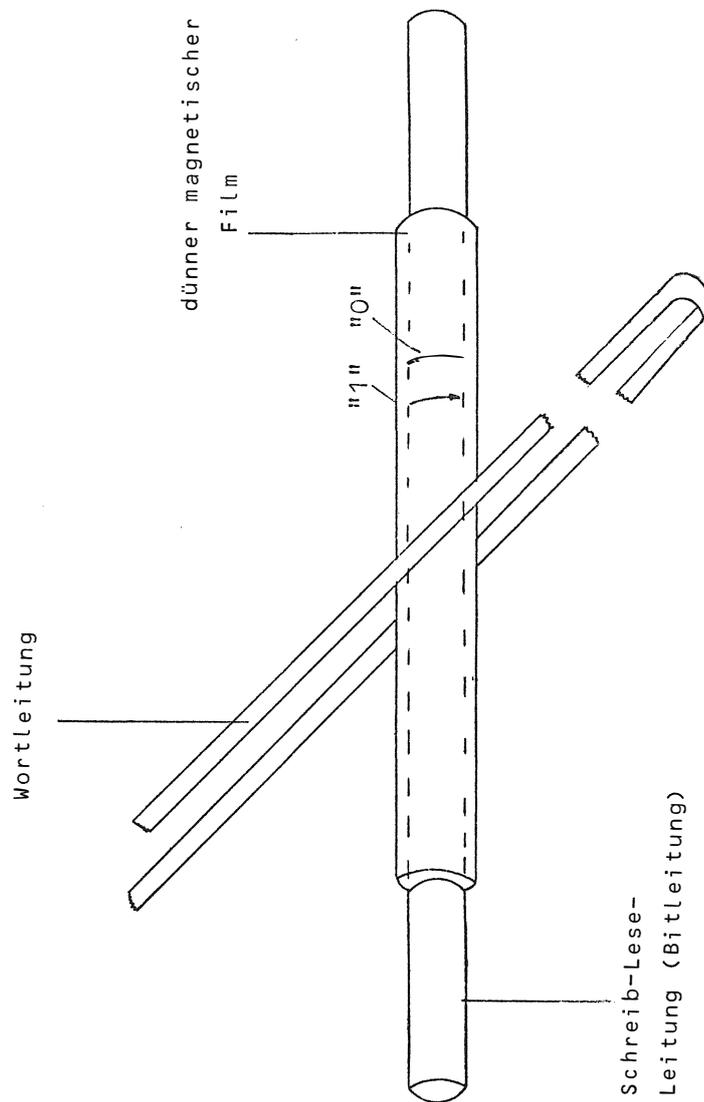
Abb. 1

Generell kann man bei der Verwendung von Magnetkernspeichern und Magnetdrahtspeichern sagen, daß der Bitpreis bei größer werdenden Speichermodulen günstiger wird. Beide Speichertypen haben einen bestimmten Grundaufwand für die Steuerung, der besonders aufwendig für einen Magnetdrahtspeicher ist. Die Verwendung von großen Speichermodulen hat allerdings den Nachteil, daß Arbeitsspeichererweiterungen relativ teuer werden.

Der *Festwertspeicher* dient der Aufnahme solcher Informationen, die vom Programm her nicht verändert werden. Es handelt sich hierbei also nur noch um einen Lesespeicher und nicht wie bei den oben genannten Speichern um Lese/Schreibspeicher. Festwertspeicher können z.B. mit Ferritübertragerkernen ausgeführt werden. Hierbei wird die Wortleitung je nach zu speichernder Binärziffer durch das Fenster des Kerns geführt oder außen herum. Die Abfrage erfolgt dann in der Weise, daß ein Impuls auf die Wortleitung geschickt wird, der seinerseits einen Impuls in der Leseleitung verursacht. Der Vorzug des Festwertspeichers ist generell in der Unzerstörbarkeit der gefädelten Programme zu sehen, eine Eigenschaft, die für zahlreiche Prozeßanwendungen von Bedeutung ist.

Der Listenpreis eines gefädelten Festwertspeichers mit einer Lesezeit von 500 ns liegt bei ca. DM 0,15–0,18/Bit.

Als *Halbleiterspeicher* wird sich der MOS-Speicher immer stärker durchsetzen. Als Beispiel sei hier das 1024-bit-MOS-RAM 1103 der Firma Intel Corp. angeführt. 1024 Bit können in einem Chip untergebracht werden! Bereits im Juni 1971 wurden 23 000 dieser Chips pro Monat verkauft, Intel rechnet mit einer Steigerung auf 60 000 Chips pro Monat. Der Vorzug



Prinzipskizze eines Magnetdrahtspeichers

Abb. 2

des MOS-Speichers ist u.a. seine geringe Leistungsaufnahme, so daß man doppelt so viele MOS-Kreise gegenüber Schaltkreisen aus bipolaren Transistoren in TTL-Technik oder ECL-Technik pro Flächeneinheit unterbringen kann. (Richtwerte: 4 K Byte / D-FBG, 1 Watt/K Byte).

Die Preise für Halbleiterspeicher ändern sich von Monat zu Monat. Stürmisch ist auch die technologische Entwicklung. Der kürzlich von Intel herausgebrachte statische MOS-Speicher 1101 ist zwischenzeitlich von dem dynamischen MOS-Speicher 1103 übertroffen worden (1101 → 256 Bit/Chip, 1103 → 1024 Bit/Chip). Die Steuerung eines dynamischen MOS-Speichers ist aufwendiger, da ein zusätzlicher Refresh-Controller notwendig ist, andererseits wird hier eine höhere Bitdichte und schnellere Zykluszeit erreicht. Man prognostiziert z. Z. den Listenpreis eines dynamischen MOS-Speichers auf DM 0,15–0,18/Bit.

Generell bleibt jedoch bei Verwendung eines Halbleiterspeichers festzustellen, daß ein Informationserhalt bei Netzausfall durch Verwendung von Pufferbatterien sichergestellt werden muß.

Die Ausnutzung der Zykluszeiten der hier angegebenen Speicher hängt entscheidend von der Verarbeitungsgeschwindigkeit des Zentralprozessors ab. Die Grenze der internen Verarbeitungsgeschwindigkeit der Zentralprozessoren, die in einfacher TTL-Technik aufgebaut werden, dürfte bei ca. 500 ns liegen. Schnellere Verarbeitungszeiten können nur durch eine aufwendigere High Speed TTL-Technik oder ECL-Technik erreicht werden.

Damit ist der Einsatzbereich der oben angeführten Speicher umrissen. Ein Magnetdrahtspeicher wird mit einem entsprechend schnellen Zentralprozessor zusammenarbeiten müssen. Zentraleinheiten mit diesen Verarbeitungsgeschwindigkeiten gehören zur Klasse der großen und komfortablen Prozeßrechner. (Registermaschine: ADD im RA-Format ca. 800 ns). Satellitenrechner, Geräterechner und kleinere Prozeßrechner verwenden vorwiegend Speicher mit Zykluszeiten zwischen 500 ns und 1  $\mu$  s. Hier ist der Einsatz eines Magnetkernspeichers oder die Kombination eines Magnetkernspeichers mit einem Festwertspeicher sinnvoll.

Aufgrund der z.Z. so stürmischen technologischen Entwicklung der Halbleiterspeicher ist es noch nicht zu einem großen Durchbruch dieser Technik gekommen. Der Halbleiterspeicher dürfte jedoch sicherlich in 2–3 Jahren den Magnetkernspeicher weitgehend am Markt verdrängen.

Die Zykluszeit eines Arbeitsspeichers ist sicherlich nicht allein ein Maß zur Beurteilung einer Zentraleinheit. Die Zykluszeit sollte nur indirekt als Maß für die Geschwindigkeit von bestimmten Zentralprozessor-Operationen herangezogen werden, da hierbei entscheidend die Struktur und Leistungsfähigkeit der in der Maschine implementierten Befehle berücksichtigt werden muß.

## *2.2 Strukturelle Merkmale von Prozeßrechner-Zentraleinheiten*

### *2.2.1 Anzahl der programmierbaren Register*

Die Anzahl der programmierbaren Register hat einen wesentlichen Einfluß auf die Struktur einer Prozeßrechner-Zentraleinheit.

Die Zentraleinheiten 301, 305 und 306 des Systems 300 sind Akkumaschinen, d. h. der Akku ist das zentrale Register des Rechenwerkes der Datenverarbeitungsanlage. In ihm werden Zwischenergebnisse addiert. Vor der Ausführung eines Befehls steht im Akku normalerweise einer der Operanden. Nach der Ausführung einer arithmetischen Operation erhält er das Resultat. Der Aufbau eines Befehlswortes einer Akkumaschine ist durch Abb. 3 erläutert und durch das System 300 hinlänglich bekannt.

Entwicklungstendenzen weisen auf Mehr-Akku-Maschinen hin, die dem Programmierer nicht nur ein oder zwei Akkus zur Verfügung stellen, sondern nunmehr vier, acht oder sechzehn universell verwendbare Register für die Informationsverknüpfung bieten. Der typische Aufbau eines Befehlswortes einer Registermaschine wird in Abb. 3 gezeigt.

Das FO Feld enthält die Codierung des Befehls bzw. der Befehlsmodifikation. Mit den Feldern F1 und F2 werden z. B. Registernummern angegeben, deren Inhalte miteinander zu verknüpfen sind. Registermaschinen zeichnen sich durch typische Befehlsformate aus, und zwar RR-, RA- und RAX-Format.

– Befehle im RR-Format bieten die Möglichkeit von direkten Register-Registerverknüpfungen. Hierbei stehen in den Feldern F1 und F2 die Nummern der Register, deren Inhalte zu verknüpfen sind.

– Befehle im RA-Format ermöglichen die Verknüpfung eines Registerinhaltes (F1-Feld) mit dem Inhalt einer Arbeitsspeicherzelle, deren Adresse im zweiten Register (F2-Feld) steht.

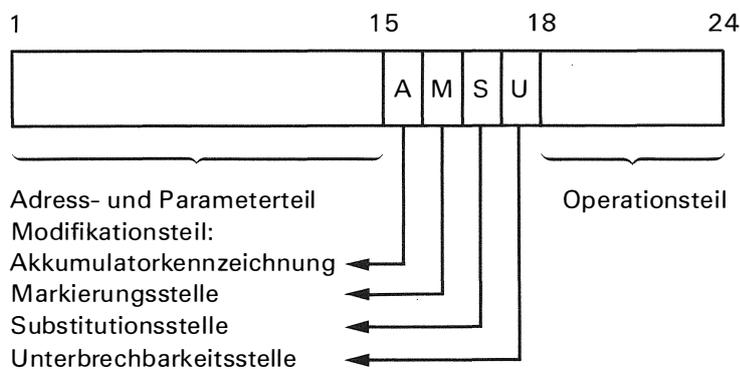
– Das RAX-Format ist außerordentlich leistungsfähig, hier kann zusätzlich zum RA-Format eine Distanzadresse (F3-Feld) hinzuaddiert werden.

Typisch für Registermaschinen ist ferner die Möglichkeit des Inkrementierens oder Dekrementierens, so daß man z. B. bei Befehlen im RAI-Format sehr leistungsfähig Adreßrechnungen ausführen kann.

Abb. 3

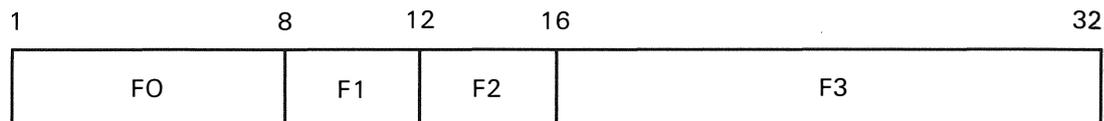
*Aufbau eines 24 Bit-Befehlswortes einer Akkumaschine*

Bitstelle:



*Aufbau eines 32 Bit-Befehlswortes einer Registermaschine*

Bitstelle:



- FO: Operationsteil (Befehl + Modifikation)
- F1: Registernummer oder Konstante
- F2: Registernummer
- F3: 16 Bit-Distanzadresse

Hinsichtlich der Programmlaufzeiten ist der Vorzug der Registermaschine gegenüber einer Akkummaschine vor allem darin zu sehen, daß eine Reihe von Rechenergebnissen nicht in den Arbeitsspeicher zurücktransferiert zu werden brauchen, sofern man sie bei weiteren Rechenoperationen wieder verwenden kann, um auf diese Weise die Programmlaufzeiten durch reine Register-Registerverknüpfungen zu verkürzen.

Bei einem Programmwechsel von einer Ebene in die andere Ebene sind Akkummaschinen zweifellos schneller, denn bei Registermaschinen muß grundsätzlich der Inhalt des gesamten Registersatzes abgespeichert werden. Das wäre nur dadurch zu vermeiden, daß man durch zusätzlichen Hardware-Aufwand pro Ebene einen Hardware-Registersatz vorsieht. Bei einem Programmwechsel innerhalb einer Ebene müßten allerdings wiederum sämtliche Registerinhalte gerettet werden.

#### 2.2.2 Anzahl der unabhängigen Prozessoren

Die Anzahl der unabhängigen Prozessoren ist weiterhin ein sehr wichtiger Faktor zur Beurteilung der Struktur einer Zentraleinheit. Als Beispiel sei hier auf unabhängige Zentral- und E/A-Prozessoren verwiesen. Dieser Punkt wird später in einem Kapitel „Abwicklung des E/A-Verkehrs“ erläutert. (Abb. 4)

### 2.3 Wortlänge

Die Wortlänge ist im allgemeinen das Vielfache einer Byte-Länge. Eine einheitliche Wortlänge bei verschiedenen Zentraleinheiten gibt es nicht. Bei kleineren Prozeßrechnern zeichnet sich recht deutlich in zunehmendem Maße die Tendenz zu 16 Bit-Maschinen ab.

### 2.4 Befehlsliste

Durch die Preisreduzierung der Halbleiter-Bauelemente wird die Anzahl der in Hardware implementierten Befehle immer größer, so daß damit dem Programmierer ein komfortablerer Befehlsvorrat zur Verfügung gestellt werden kann.

Beim Vergleich unterschiedlicher Zentraleinheiten sollte man voraussetzen, daß neben dem üblichen Befehlsvorrat mindestens Festpunkt-MLT/DIV und möglichst auch Gleitpunktarithmetik im Befehlsvorrat implementiert sein müssen.

Bei Prozeßrechnern sollten die für diese Anwendungen notwendigen Bithandling-Befehle nicht fehlen. Während bei manchen amerikanischen Firmen unter diesem Begriff immer nur das Setzen, Löschen oder Testen des ersten oder letzten Bits eines Wortes verstanden wird, sollte man bei Prozeßanwendungen bedenken, daß diese Funktionen mit jedem Bit eines Wortes ausgeführt werden müssen. Wünschenswert wären darüberhinaus Möglichkeiten zur logischen Verknüpfung von Bits.

Gleiche Forderungen ergeben sich auch für die Behandlung von Bytes. Man benötigt Befehle, die es gestatten, Bytes zu laden, zu speichern oder zu vergleichen. Darüberhinaus wären logische Verknüpfungen von Bytes wiederum erstrebenswert.

Bei der Programmierung von Datenfeldern wären als Feldbefehle beispielsweise Move-Befehle recht geschickt. In diesem Fall wäre es möglich, ein angegebenes Sendefeld in ein entsprechendes Empfangsfeld Byte für Byte zu übertragen, wobei auch eine Überlappung der beiden Felder möglich sein könnte.

Bei Feldbefehlen variiert die Operandenlänge von Befehl zu Befehl, so daß Feldbefehle grundsätzlich keine festen Wortlängen besitzen. Generell ist zu sagen, daß Maschinen, die schnelle Operationszeiten aufweisen sollen, im allgemeinen mit festen Wortlängen arbeiten. Aus diesem Grunde bleibt zu prüfen, ob die hier genannten Befehle für den jeweiligen Anwendungsfall erforderlich sind.

Interessant sind ferner sogenannte mikroprogrammierte Befehle, die u. a. auch bei einigen kleineren Prozeßrechnern zu finden sind. Darunter ist zu verstehen, daß bei Registerbefehlen mehrere Verknüpfungen in einem Befehl ausgeführt werden, z. B. AND, SHIFT, SKIP.

Die Überschaubarkeit der Befehlsliste ist für den Anwender weiterhin von Bedeutung. Sie sollte matrixförmig aufgebaut sein, eine bestimmte Anzahl von Grundbefehlen aufweisen, die möglichst alle die gleichen Befehlsmodifikationen haben.

Aus den oben genannten Ausführungen ist nun sicherlich zu verstehen, daß der Vergleich der Ausführungszeit bestimmter Operationen nicht allein als Maß für die Bewertung der Leistungsfähigkeit einer Zentraleinheit herangezogen werden kann. Um hier eine Aussage zu bekommen, ist es notwendig, den gleichen Anwendungsfall mit dem Befehlsvorrat unterschiedlicher Maschinen zu programmieren. Ein Vergleich der Programmlaufzeiten würde dann

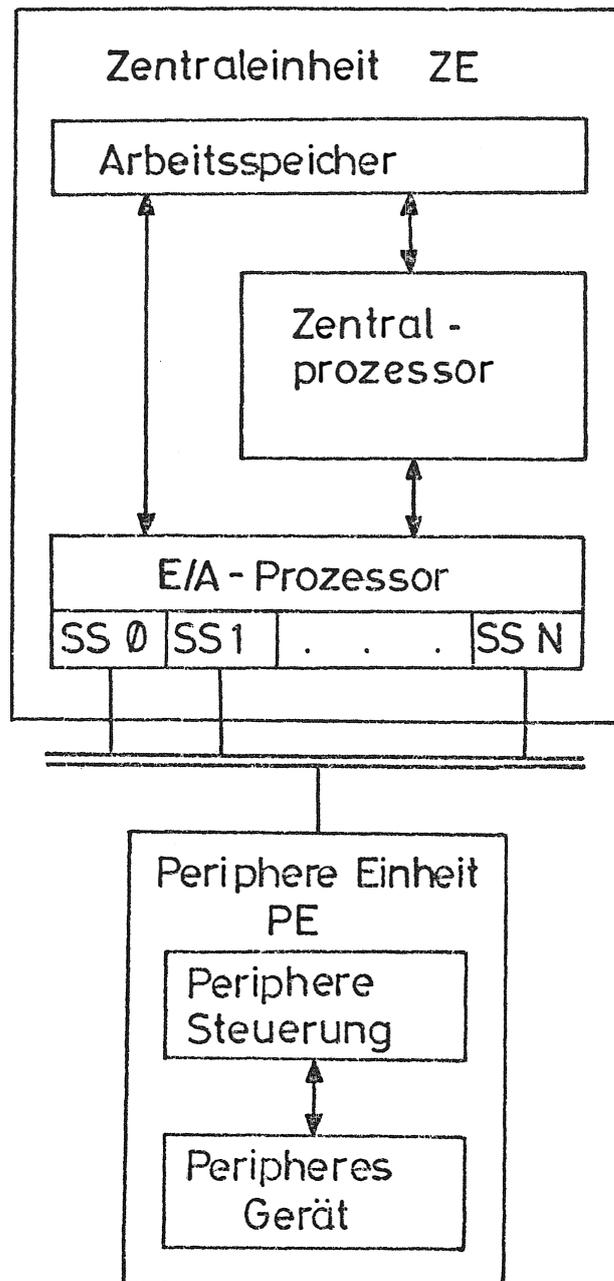


Abb. 4

eine Aussage über die Zentralprozessorbelastung und damit über die Leistungsfähigkeit einer Zentraleinheit liefern.

Diese Methode scheitert vielfach daran, daß die verschiedenen Zentraleinheiten nicht zur Verfügung stehen.

Eine andere Möglichkeit bietet die Verwendung eines Mix-Wertes. Die folgenden Ausführungen stützen sich auf eine Veröffentlichung im Data Processing Magazine 12/70. Mix-Werte werden auf statistischer Basis entwickelt. Man ermittelt für bestimmte Anwendungen, z. B. Prozeßanwendungen den prozentualen Anteil einzelner Befehle innerhalb eines Programms und unterstellt damit, daß diese prozentuale Verteilung typisch für den jeweiligen Anwendungsfall ist. Diese prozentualen Anteile sind Abb. 5 zu entnehmen. Man multipliziert dann die tatsächlichen Operationszeiten der einzelnen zu vergleichenden Befehlsvorräte unterschiedlicher ZE mit dem prozentualen Gewicht, das man aufgrund der Statistik diesem Befehl zugeordnet hat. Somit erhält man eine Aussage über die mittlere Operationszeit bei Verwendung dieser Zentraleinheit für eine bestimmte Anwendung.

Es sind weitere Korrekturfaktoren notwendig, z. B. um dem unterschiedlichen Umfang der Befehlslisten Rechnung zu tragen. In Abb. 6 wird die Anzahl der Befehle einschließlich der sinnvollen Modifikationen einem Wertungsfaktor zugeordnet. Nach den Formeln in Abb. 6 ist dann die gewertete Operationszeit und die Zentralprozessorbelastung zu ermitteln.

Mit diesem Verfahren kann man aussagekräftige Vergleiche zwischen Rechnern durchführen, wobei die Zentralprozessorbelastung in Abhängigkeit des Preises in einem Koordinatensystem aufgetragen wird.

Abb. 5

*Ermittlung der mittleren Operationszeit mit einem Prozeßrechner Mix-Wert*

Arithmetische Operationen: (einfache Genauigkeit)	Addieren, Subtrahieren	7%
	Multiplizieren	1%
	Dividieren	0,1%
Logische Operation, Verschiebeoperationen:	Vergleichen	15%
	Um 6 Bit schieben	5%
	Und/Oder	4%
Transfer und Sprungoperationen:	Laden/Speicher	33%
	Bedingter Sprung	15%
	Inkrementieren und Index speichern	7%
	Register umladen	5%
Ein/Ausgabeoperationen:	Programmierter E/A-Transfer	6%
	Blockverkehr einleiten	1%
	Unterbrechungsbearbei- tung, Retten von 4 Re- gistern	0,9%

Abb. 6

*Anzahl der Befehle einschließlich Modifikationen in Abhängigkeit von einem Wertungsfaktor*

Befehle einschließlich sinnvoller Modifikationen	30	40	50	60	70	80	90	100	125	150	250
Wertungsfaktor	0,4	0,45	0,5	0,55	0,60	0,63	0,67	0,72	0,80	0,86	1

Errechnung der Zentralprozessorbelastung in % (eine gewertete Operationszeit von 10  $\mu$ s wird gleich 100% gesetzt):

$$\text{gewertete Operationszeit} = \frac{\text{mittlere Operationszeit}}{\text{Wertungsfaktor}}$$

$$\text{Zentralprozessorbelastung} = \frac{\text{gewertete Operationszeit} \cdot 100}{10 \mu\text{s}}$$

## 2.5 Unterbrechungsstruktur

Für eine leistungsfähige Unterbrechungsstruktur sollte es generell möglich sein, eine Unterbrechung nach der Ausführung jedes Befehls zu erreichen!

Bei Kleinrechnern wird die Unterbrechungsstruktur der Zentraleinheit vielfach durch folgende Aussage gekennzeichnet: Vorrangunterbrechung mit Hardware-Priorität! Darunter ist zu verstehen, daß der Zentralprozessor auf z. B. 64 Unterbrechungsebenen ausgebaut werden kann, die hardwaremäßig gegeneinander prioritiert sind. Bei jedem Interrupt wird die Startadresse des zu bearbeitenden Programms mitgeliefert. Diese Unterbrechungsstruktur ist dann effektiv, wenn die den Unterbrechungsebenen zugeordneten Reaktionsprogramme kurz sind, keine weiteren Koordinierungsmaßnahmen notwendig werden, so daß ein Interrupt durch einen entsprechenden Ausgabeaufruf, z. B. an ein Prozelelement, sofort beantwortet werden kann. Sofern aufgrund mehrerer Interrupts Programmkoordinierungen notwendig werden, ist die oben geschilderte Unterbrechungsstruktur nicht mehr effektiv!

Hier stellt sich nämlich die Frage, welche Unterbrechungsstruktur für den Prozeßrechner-einsatz notwendig und sinnvoll ist. Allgemein gelten folgende Forderungen:

- Für die rasche Reaktion auf Alarme ist es notwendig, daß bestimmte Alarme oder Alarmgruppen ohne Mitwirken des Betriebssystems per Hardware zum Start eines spezifischen Reaktionsprogramms führen.

- Außerdem müßten diese Programme per Hardware vor Unterbrechungen durch niederpriorie Alarme geschützt werden.

- Ferner müssen Alarmeingänge maskiert werden können (z. B. abhängig von verschiedenen Betriebszuständen des Prozesses), so daß sie zu keinem unerwünschten Interrupt führen.

Die Konsequenz dieser Forderungen sähe für die Unterbrechungsstruktur der Zentraleinheit zunächst so aus, daß eine Interruptebene in mehrere Unterebenen aufgespalten werden sollte und jeder Unterebene ein Alarm oder eine Alarmgruppe fest zugeordnet wird. Dazu wäre eine entsprechende Anzahl von Eingängen notwendig. Damit nun eine echte Verbesserung der Reaktionszeit erzielt wird, müßte jeder Lauf über das Betriebssystem vermieden werden. Die häufigste Reaktion auf Alarme wird aber letztlich eine Ausgabeoperation sein,

die die Alarmbearbeitung abschließt. Werden E/A-Operationen ohne Wissen und Beteiligung des Betriebssystems durchgeführt, so kann eine störungsfreie Zusammenarbeit vieler Programme mit E/A-Verkehr nur unter großem Hardware-Aufwand realisiert werden. Zieht man hier die Mitarbeit des Betriebssystems heran, geht ein wesentlicher Zeitgewinn wieder verloren. Dann steht aber der noch verbleibende Zeitgewinn einer Hardware-Programmzuordnung in keinem Verhältnis zu seinem Aufwand.

Man darf wohl sagen, daß dieser Aufwand nur gerechtfertigt ist, wenn es der jeweilige Anwendungsfall erlaubt, am Betriebssystem und damit der Programmkoordination sparen zu können. Ist das nicht der Fall, so sollte einer Aussage über die Reaktionszeit einer Zentraleinheit die entsprechende Bedeutung beigemessen werden.

Unter diesen Gesichtspunkten darf man wohl zusammenfassend sagen, daß es bei einem Prozeßrechner durchaus ausreichen würde, wenn er z. B. lediglich vier Unterbrechungsebenen aufweisen würde. Sämtliche E/A-Operationen sollten über das Betriebssystem abgewickelt werden. Lediglich bei der Bearbeitung von sehr schnellen externen Ereignissen wäre eine Schnellreaktionsebene, die über der Betriebssystemebene liegt, wichtig. Über diese Schnellreaktionsebene können schnelle Alarme abgewickelt werden, wobei die notwendigen Ausgaberoutinen ohne Mitwirken des Betriebssystems erfolgen können.

Die vorher genannte Forderung, bestimmte Alarme befehlsgesteuert zu maskieren, erfüllt am einfachsten das Prozeßelement. Gleichmaßen sollte man über das Prozeßelement das Aussperren nieder-priorer Alarme während der Behandlung eines bestimmten Alarms abwickeln.

## *2.6 Abwicklung des E/A-Verkehrs*

Die Abwicklung des E/A-Verkehrs kann im einfachsten Fall programmiert erfolgen.

Beim Vergleich unterschiedlicher E/A-Strukturen sollte untersucht werden, inwieweit für die Abwicklung des E/A Verkehrs ein eigener E/A-Prozessor verfügbar ist (Abb. 4). In diesem Fall wird der E/A-Prozessor vom Zentralprozessor parametrisiert, um anschließend simultan zum Zentralprozessor entsprechende Operationen abzuwickeln. Der Zugriff zum Arbeitsspeicher erfolgt im einfachsten Fall in der Weise, daß sich Zentralprozessor und E/A-Prozessor den Verkehr teilen.

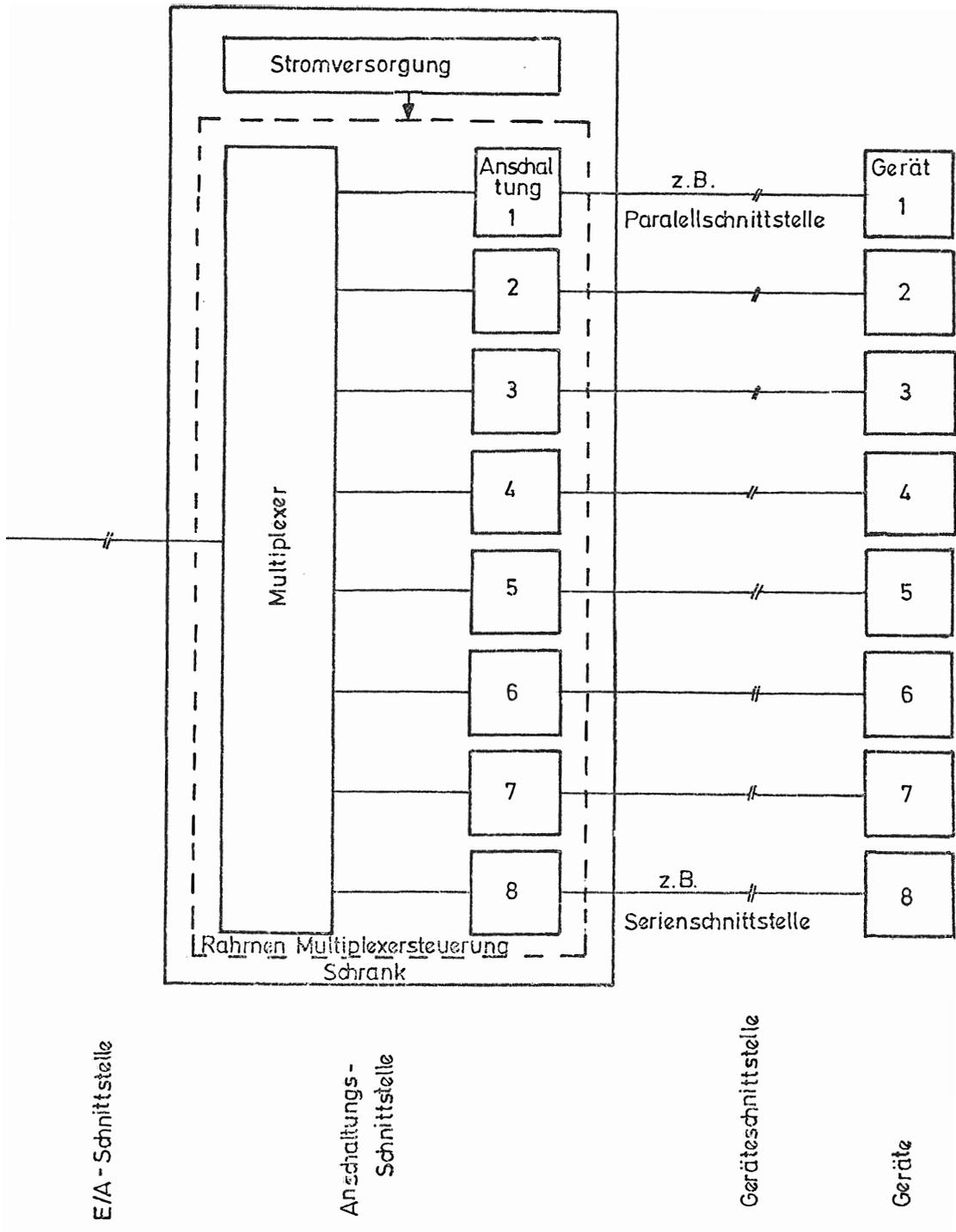
Generell ist hier der Zugriff zum Arbeitsspeicher für die unterschiedlichen Prozessoren ein Engpaß. Bei komfortableren Zentraleinheiten wäre es dann möglich, daß ein echter Zeitmultiplexbetrieb dadurch ermöglicht wird, daß Zentralprozessor und E/A-Prozessor auf unterschiedliche ASP-Module zugreifen. In diesem Fall werden bereits Datenraten bis zu 2 MHz bei einem 16 Bit breiten Datenweg erreicht.

Die Abwicklung des E/A-Verkehrs über einen eigenen E/A-Prozessor, der in die Zentraleinheit integriert wird, konnte erst durch technische und preisliche Fortschritte auf dem Gebiet der elektronischen Bauelemente erzielt werden. Während man in bisherigen Systemen Funktionen eines E/A-Prozessors in die entsprechenden Gerätesteuern implementieren mußte, kann nun die Gerätesteuerung wesentlich aufwandsärmer konzipiert werden. Als Beispiel wird auf Abb. 7 verwiesen. Der Aufwand für die Anschaltung von langsamen peripheren Geräten an eine Zentraleinheit beträgt hier nur noch 2 Doppelflachbaugruppen, die wahlweise entweder direkt im Rahmen der Zentraleinheit oder bei mehreren Geräten in einen Schnittstellen-Multiplexerrahmen gesteckt wird.

## *2.7 Aufbautechnik*

Bei den meisten Herstellern ist der Trend zu einem kompakten Auf- und Einbau erkennbar. Vielfach werden funktionsfähige Zentraleinheiten angeboten, die nicht größer als eine Schreibmaschine sind.

Rechner von großen europäischen Herstellern zeichnen sich durch hohe Rahmenstabilität, gute Führung der Flachbaugruppen im Rahmen und indirekte Steckung der Baugruppen aus (z. B. SIVAREP B)



Bausteinprinzip der Multiplexersteuerung

Abb. 7

Die meisten amerikanischen Kleinrechner sind aus wenigen aber großen Flachbaugruppen aufgebaut. Sie werden vielfach direkt gesteckt, sind schlecht geführt und häufig nicht verriegelt. Die Flachbaugruppen liegen dabei waagrecht und reichen fast über die gesamte Breite und Tiefe. Beim Anschluß von Kabeln ist häufig festzustellen, daß die Karten wegen ihrer schlechten Führung an diesen Stellen deutlich durchgebogen werden. (Gefahr von Haarrissen!)

Es sollte daher stets geprüft werden, ob die jeweilige Aufbautechnik für den Prozeßrechner-einsatz geeignet ist. Dazu kommt noch ein weiterer Gesichtspunkt. Nicht jeder Hersteller von Zentraleinheiten liefert auch ein Prozeßelement. Sofern dieses im Lieferspektrum einer Rechnerfirma ist, läßt sich fast immer feststellen, daß die Baugruppen senkrecht im Rahmen stehen. Bei gleichzeitigem Einbau von Prozeßelement und Zentraleinheit in einen Schrank geht bei unterschiedlicher Lage der Flachbaugruppen eine einheitliche Luftführung verloren. Bei Geräten für Prozeßanwendungen sollte also sehr auf eine einheitliche Aufbautechnik geachtet werden, die den Forderungen der Industrieelektronik entspricht.

### *3. Schlußbetrachtung*

Hier wurden eine Reihe von technischen Daten der Zentraleinheit untersucht, um Vergleichskriterien zur Beurteilung unterschiedlicher Zentraleinheiten zu liefern.

Es gibt sicherlich noch weitere Komponenten, die in einigen Anwendungen von besonderer Bedeutung wären, wie z. B.:

- Programmschutz
- Schutz des Zentralprozessors und des E/A-Prozessors
- Kopplungsmöglichkeiten

Wie eingangs schon gesagt, ist für den Vergleich von Zentraleinheiten weiterhin unbedingt notwendig, daß auch die zur Verfügung stehende Peripherie und die Standard-Software beurteilt werden müßten, denn auch diese Faktoren gehen in den Gesamtpreis einer Anlage entscheidend mit ein.

Beispielsweise wäre hier auf folgende Punkte zu verweisen:

- Typenvielfalt und Preis von Prozeßsignalformern
- Leistungsfähigkeit der PSF-Steuerung
- Preis und Leistung des peripheren Gerätespektrums
- Verfügbare Programmbibliothek (Assemblersprache, Betriebssystem, Assembler, Compiler-Sprachen, Compiler, Test- und Bedienprogramme, Wartungsprogramme)

**Dr. D. Stams**

Robert Bosch GmbH Schwieberdingen

Prozeßrechnereinsatz in Laboratorien und  
Prüffeldern der Kraftfahrzeugausrüstungsindustrie

# Prozeßrechnereinsatz in Laboratorien und Prüffeldern der Kraftfahrzeugausrüstungsindustrie

## *Gliederung:*

1. Einführung
2. Wirtschaftlicher Versuchsbetrieb, Beispiel Einspritz-Steuergeräte-Prüfung
3. Schnelle Versuchsabwicklung, Beispiel California-Test
4. Umfassende Informationsverarbeitung, Beispiel Dauerlauf
5. Optimale Versuchsdurchführung, Beispiel Motorenversuch
6. Rechenanlagen
7. Zusammenfassung

## 1. Einführung

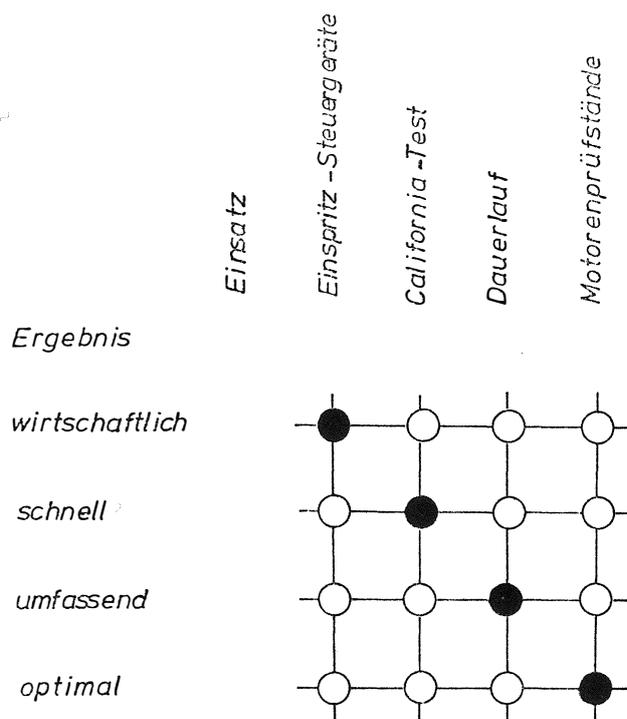
Die Entwicklung der Kraftfahrzeugausrüstungen ist in den letzten Jahren erheblich in Bewegung gekommen; einerseits durch die Forderungen, die Umweltverschmutzung zu verringern und die Sicherheit zu erhöhen, andererseits durch die neuen technologischen Möglichkeiten der Elektronik. Die Steigerung der Materialausnutzung und Rationalisierung der Großserienfertigung zur Kostensenkung ist eine weitere, durch den Konkurrenzdruck erzwungene Forderung an die Entwicklung.

Diese Weiterentwicklung der Aufgaben und Möglichkeiten führte bei Bosch zum Aufbau der Technischen Zentren Forschung, Autoelektrik und Hydraulik.

Zahlreiche Prüffelder wurden modernisiert, andere neu geschaffen. Zur *wirtschaftlichen* Gestaltung des Versuchsbetriebes, zur *schnellen* Abwicklung von Versuchen und Versuchsreihen, zur *umfassenden* Verarbeitung aller anfallenden Versuchswerte und zur *optimalen* Gestaltung der Versuchsabläufe wurden technische Rechenzentren hinzugefügt.

Alle diese Fortschritte werden bei allen Anwendungen erreicht, die folgende Diskussion wird gemäß Abb. 1 beschränkt auf:

Einspritz-Steuergeräte-Prüfung	als Beispiel für „wirtschaftlich“,
California-Test	als Beispiel für „schnell“,
Dauerlauf	als Beispiel für „umfassend“ und
Motorenversuch	als Beispiel für „optimal“.



● Diese Kreuzungspunkte der Matrix „Einsatz/Ergebnis“ werden beschrieben.

Abb. 1 Rechananlage im Technischen Zentrum Autoelektrik

## 2. Wirtschaftlicher Versuchsbetrieb, Beispiel Einspritz-Steuergeräte-Prüfung

Das Einspritz-Steuergerät ist das zentrale Aggregat der elektronisch gesteuerten Benzineinspritzung.

Der Betriebszustand des Motors wird erfaßt durch Fühler für Temperaturen, Druck, Drehung der Kurbelwelle, Spannung, Start-, Leerlauf-, Vollast- und Beschleunigungszustand. Das Steuergerät gibt elektrische Impulse verschiedener Längen ab und dosiert über die Öffnungszeit der angeschlossenen Magnetventile die Kraftstoffmenge, die dem Motor zugeführt wird.

Der Prüfstand ist so aufgebaut, daß diese beschriebenen Funktionen des Steuergerätes überprüft werden können. Über einen Blattschreiber, der neben dem Prüfstand steht, werden im Frage-Antwortspiel Identifikation des Versuchs, Typ des Prüflings, Art der Prüfung – Gesamtprüfung oder Einzelschritt mit Nr. des Prüfschritts dem Rechner mitgeteilt, siehe Abb. 2.

Wenn der Prüfstand betriebsklar ist, drückt man den Startschalter, daraufhin schließt der Rechner den Freigabeschalter und stellt über eine Anpaßsteuerung die Fühler bzw. deren Simulatoren auf die programmierten Betriebszustände ein. Die Impulse des Steuergerätes für die Magnetventile und den Pumpenschalter werden über Zähler und Schalter gemessen und vom Rechner erfaßt.

Die bei den eingestellten Betriebszuständen gemessenen Werte werden ausgewertet, mit Sollwerten verglichen und sofort direkt beim Prüfstand tabellarisch und grafisch über Blattschreiber und x-y-Schreiber ausgegeben.

Bis jetzt wurden über 20 verschiedene Prüfprogramme entwickelt; entsprechend den Anforderungen wird das Programmsystem darüber hinaus immer weiter ausgebaut. Aus diesem Grund können im Entwicklungsbereich fest verdrahtete Prüfautomaten nicht verwendet werden.

Andererseits sind diese programmierten Prüfungen im Entwicklungsbereich wichtige Vorstufen für frei- oder festprogrammierte Automaten in der Fertigung und Endkontrolle.

Anschließend soll der wirtschaftliche Erfolg des Prozeßrechnereinsatzes an diesem Beispiel erläutert werden.

Durch die Automatisierung sanken die Kosten  
 von 100% pro Steuergeräte-Prüfung manuell  
 um 38% pro Steuergeräte-Prüfung  
 auf 62% pro Steuergeräte-Prüfung mit Rechner.

Diese Einsparung wirkt sich auf Tausende von Prüfungen im Jahr aus.

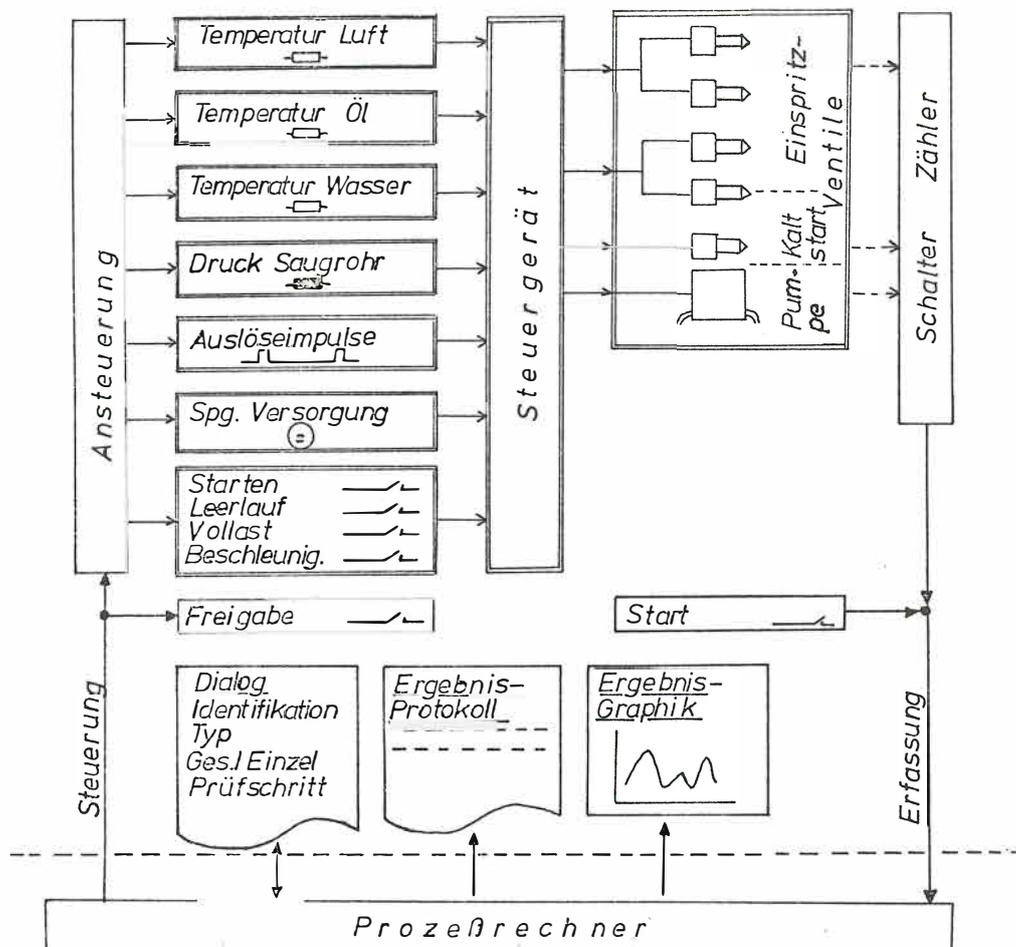


Abb. 2 Prüfanordnung für Einspritz-Steuergeräte

Der einmalige Aufwand der Programmierung und des Prüfstandsbaus und -anschlusses ist nach ca. 1 Jahr amortisiert.

Zusätzlich zu der Einsparung konnte die Leistung an diesem Prüfplatz in folgenden Punkten qualitativ verbessert werden:

- geringere Fehlerquote,
- zusätzliche Toleranzprüfungen möglich,
- zusätzliche Statistiken über mehrere Prüfungen möglich,
- zuverlässige, übersichtliche Dokumentation der Prüfergebnisse.

### 3. Schnelle Versuchsabwicklung, Beispiel Californiastest

Für die Messung und Beurteilung der Umweltverschmutzung gibt es heute eine Reihe verschiedenartiger Abgastests. Der erste dieser Tests wurde im US-Bundesstaat California entwickelt und wird daher als California-Test bezeichnet. Alle Abgastests orientieren sich am Verhalten des Fahrzeugs im Straßenverkehr; deshalb werden die Tests nach bestimmten Fahrvorschriften an kompletten Fahrzeugen auf Rollprüfständen durchgeführt. Der Test beginnt mit kaltem Motor. Beim California-Test werden bestimmte Fahrzyklen mehrmals

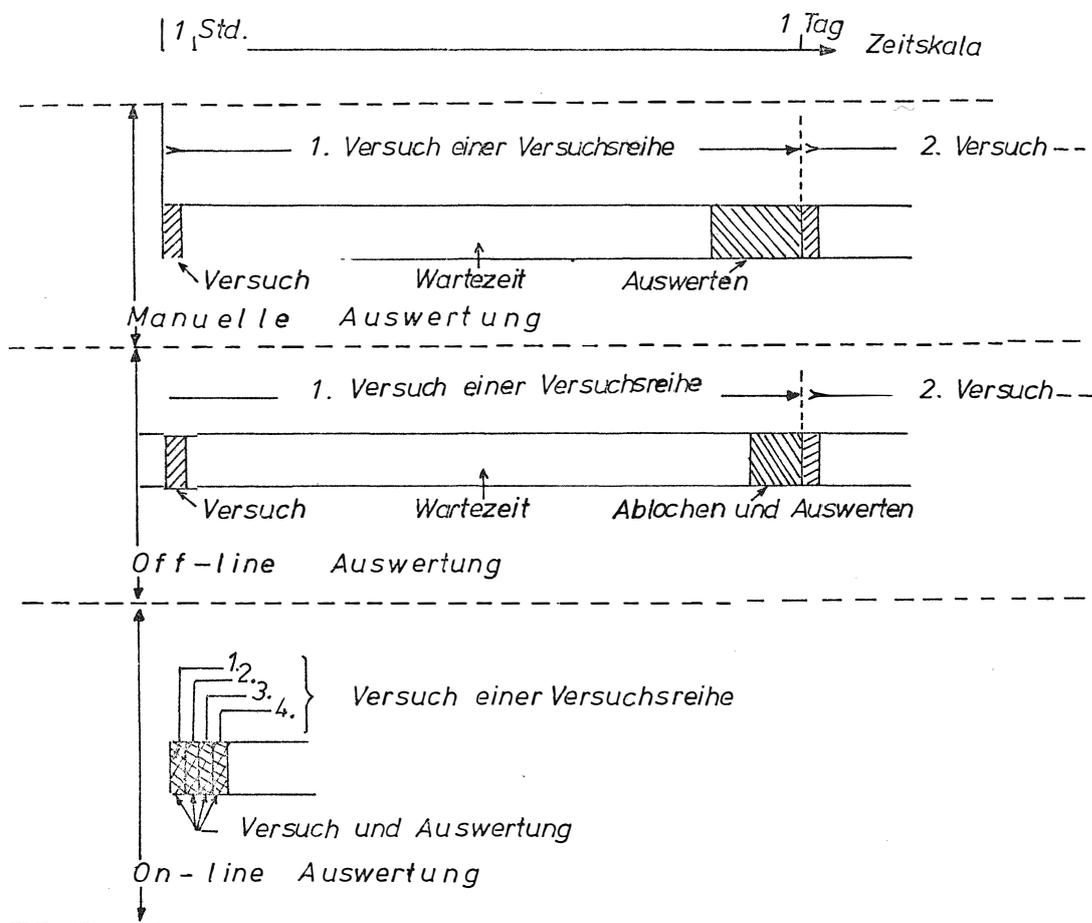


Abb. 3 Schnellere Aufeinanderfolge von California-Tests einer Versuchsreihe

hintereinander durchfahren. Die Abgasemission wird gemessen und je nach Fahrstufe und -zyklus bewertet. Die Anzahl der zu erfassenden Meßwerte ist relativ groß, die Auswertvorschriften sind kompliziert und umfassen viele Rechengänge.

Bei der manuellen Auswertung werden die Meßwerte für  $\text{CO}$ ,  $\text{C}_m\text{H}_n$ ,  $\text{NO}_x$ ,  $\text{CO}_2$  und Geschwindigkeit  $v$  von Koordinatenschreiber-Protokollen abgegriffen; nach den Auswertvorschriften werden die Werte für die mittleren Gasemissionen errechnet. Diese Arbeit dauert ca. 3 Stunden. Wenn man die aus den Protokollen abgegriffenen Werte ablocht und anschließend die Lochstreifen über Rechner auswertet, benötigt man immerhin noch ca. 2 Stunden zum Ablesen und Lochen und ca. 10 Minuten zum Auswerten und Ausdrucken.

Die Auswert- und Ablocharbeiten erfordern ein einigermaßen gutes, sorgfältiges und zuverlässiges Personal. Andererseits sind diese Arbeiten stumpfsinnig; das führt zu permanenter Personalknappheit für diese Tätigkeiten und damit zu Einschränkungen bei den Auswertungen und erfahrungsgemäß zu Wartezeiten von ein oder mehreren Tagen.

Der Prozeßrechner löst hier alle Probleme mit einem Schlag; er erfaßt, wertet aus und protokolliert.

Versuch und Auswertung laufen reibungslos ab und dauern einschl. der 16 Minuten Fahrzeit nur noch ca.  $\frac{1}{2}$  Stunde.

Abb. 3 veranschaulicht, um wieviel schneller der Versuchsbetrieb durch Einsatz des Prozeßrechners geworden ist.

Das Prozeßsystem California-Test wird durch Abb. 4 angedeutet. Die Geschwindigkeits-Sollkurve ist auf einer mit konstanter Winkelgeschwindigkeit rotierenden Fahrtrommel aufgezeichnet; die Istwert-Anzeige liegt darüber.

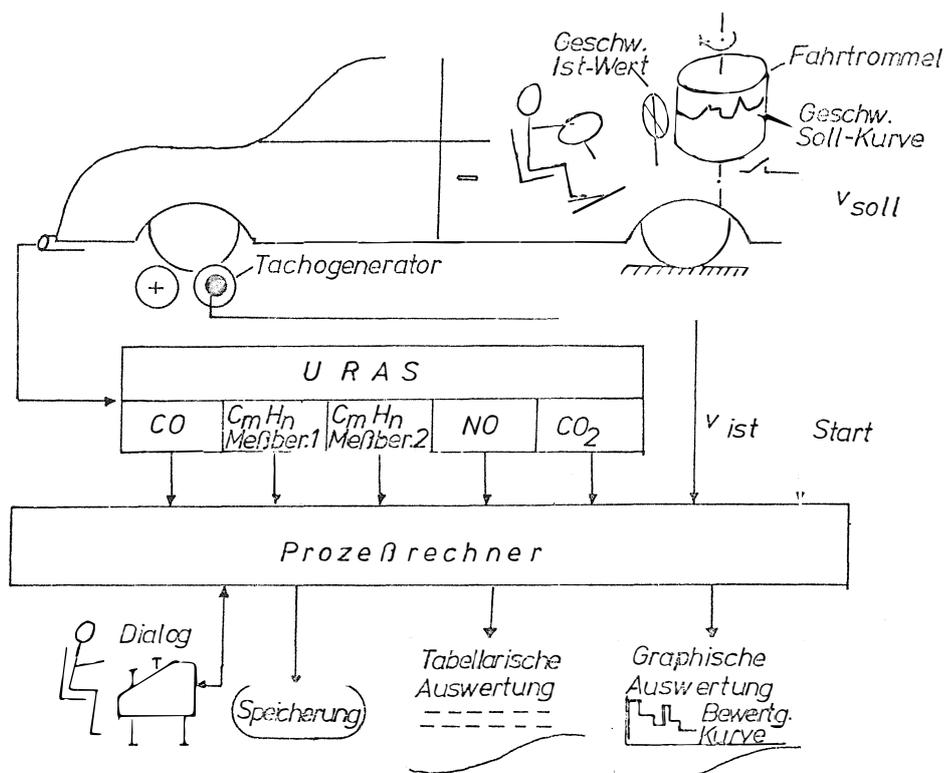


Abb. 4 Prozeßsystem California-Test

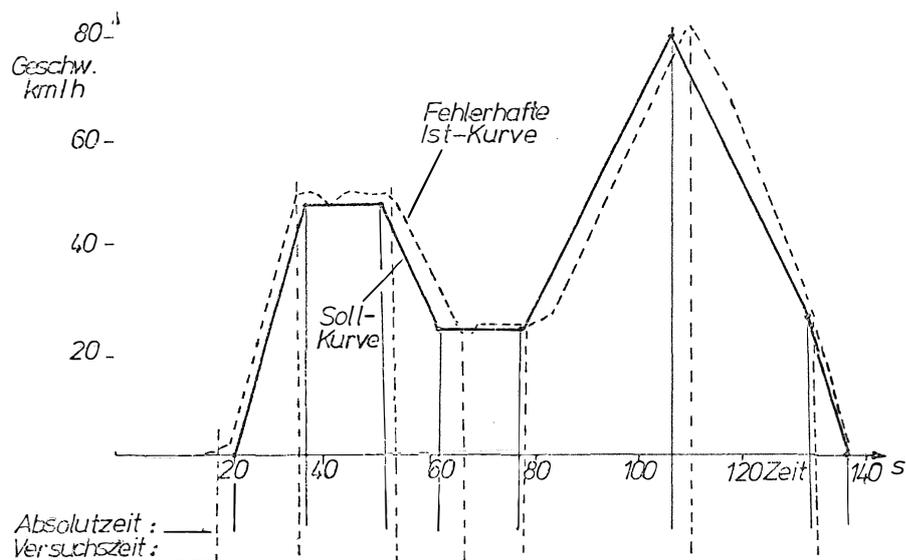


Abb. 5 California-Test, Ermittlung der „Versuchszeit“

Der Fahrer muß den Wagen so beschleunigen bzw. verzögern, daß Soll- und Istwert übereinstimmen. Die Vorschrift erlaubt eine Abweichung der gefahrenen Geschwindigkeit von der Sollgeschwindigkeit nach oben hin; dabei werden aber die Testergebnisse verschlechtert und die Erfüllung des Tests in Frage gestellt. Die Abgase werden durch ein Ultrarotabsorptionsspektrometer (URAS) geleitet, das eingepreßte Ströme proportional den Momentanwerten der Konzentrationen der verschiedenen Komponenten an den Rechner gibt.

Von der Rolle wird über einen Tachogenerator die Geschwindigkeit als Spannung an den Rechner gegeben.

Streng genommen müßte der Rechner entsprechend der Drehung der Fahrtrommel nach „Absolutzeit“ die Auswertung der Emission vornehmen. Dann wären aber fehlerhaft gefahrene Versuche unter Umständen unbrauchbar.

Um sie zumindest für die laufende Entwicklungsarbeit zu „retten“, analysiert der Rechner die Istkurve. Er ermittelt die „Versuchszeit“, also die Zeitpunkte, in denen sich die Beschleunigung ändert; daraus ergeben sich die Knickpunkte der Fahrkurve und daraus die Auswertebereiche mit den unterschiedlichen Gewichtungsfaktoren für die Auswertung, siehe Abb. 5.

Durch die dynamische Ermittlung der Versuchszeit werden die Versuchsausfälle reduziert und die Versuchsreihen noch schneller durchfahren.

#### 4. Umfassende Informationsverarbeitung, Beispiel Dauerlauf

Die Erprobung der Erzeugnisse auf ihre Dauerbelastbarkeit ist ein wichtiger Bestandteil der Entwicklung. Im Jahr werden im Technischen Zentrum Autoelektrik etwa 3000 Dauerlaufuntersuchungen mit im Durchschnitt je 3 Prüflingen, also insgesamt ca. 10 000 Dauerlaufversuche durchgeführt.

Hierfür wurde ein großes Labor mit 50 Räumen und ca. 50 Prüfständen aufgebaut, an denen 250 Prüflinge gleichzeitig untersucht werden können. An jedem Prüfling werden zwischen 5 und 40, im Mittel 10 Meßwerte abgegriffen – das sind 2500 Meßstellen im gesamten Dauerlauf. Diese Meßwerte sollen laufend überwacht werden. Sobald ein Meßwert aus

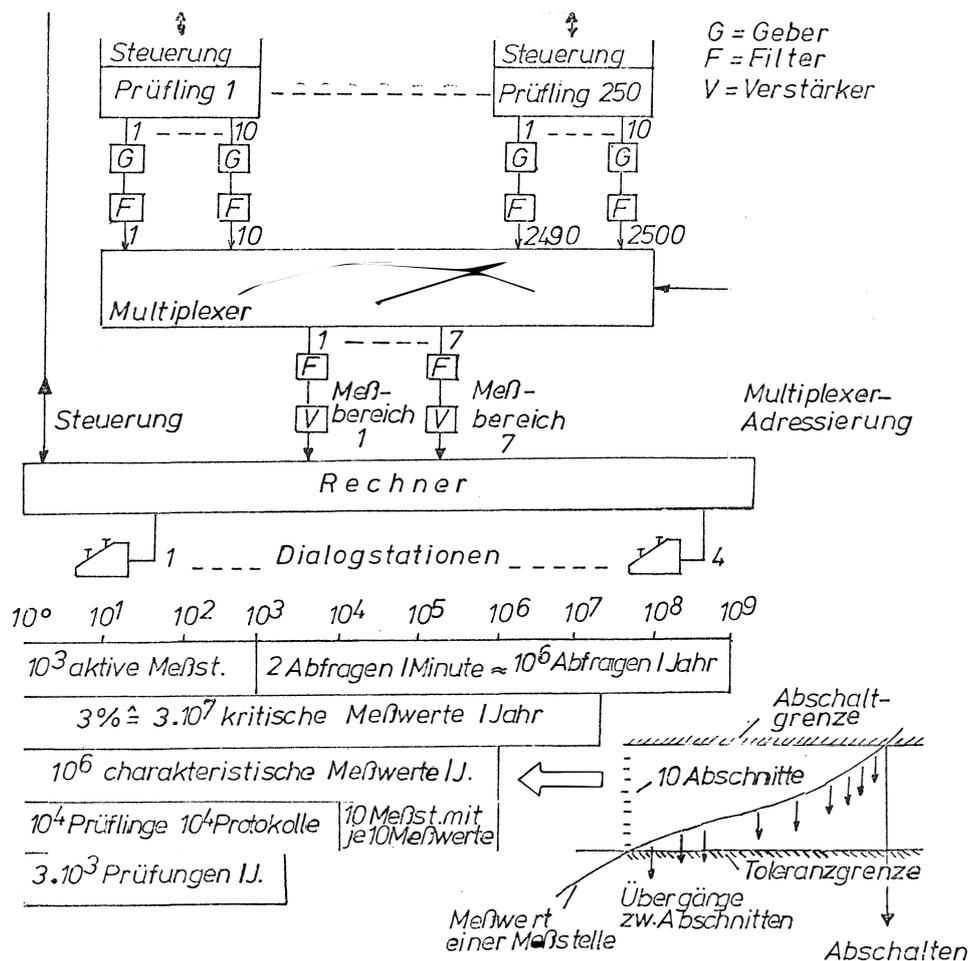


Abb. 6 Dauerlaufsystem

einem für ihn vorgegebenen Toleranzbereich hinausläuft, sollen laufend die zu diesem Prüfling gehörigen Meßwerte als Meßwertsätze registriert werden. Sobald ein Meßwert die Abschaltgrenze erreicht, soll der Prüfling abgeschaltet werden.

Diese Aufgabe wird durch ein umfangreiches System aus Gebern, Filtern, Umschaltern, Verstärkern, Prozelementen, Prozeßrechner, Dialogstationen und Programmen gelöst, siehe Abb. 6 oben.

Das Wichtige ist, daß alle Informationen verarbeitet und soweit verdichtet werden, daß jeder Prüfling drei übersichtliche Protokolle als Endresultat mitbekommt: Ein Betriebsprotokoll mit allen besonderen Vorkommnissen, ein Lebenslaufprotokoll mit einigen charakteristischen Meßwertsätzen und gegebenenfalls ein Grenzüberschreitungsprotokoll mit Meßwertsätzen für die kritischen Phasen des Versuchs, von der ersten Toleranzüberschreitung bis zur Abschaltung.

In Abb. 6 unten wird die Datenreduktion schematisch dargestellt. 1000 aktive Meßstellen werden 2mal pro Minute, also  $10^6$  mal pro Jahr abgefragt, das sind  $10^9$  Abfragen pro Jahr.

Etwa 3%, also  $3 \times 10^7$  Meßwerte liegen zwischen den Toleranz- und Abschaltgrenzen.

Jeder dieser Bereiche ist in 10 Abschnitte unterteilt. Beim Übergang eines Meßwertes von einem zum anderen Abschnitt wird ein Meßwertsatz gespeichert. Falls ein Meßwert in 150

Minuten von der Toleranz- zur Abschaltgrenze läuft, findet eine Datenreduktion von 300 aufgenommenen Meßwertsätzen auf 10 gespeicherte charakteristische Meßwertsätze statt; man gewinnt dann im Mittel aus  $3 \times 10^7$  kritischen  $10^6$  charakteristische Meßwerte. In den  $10^4$  Protokollen sind dann 10 Meßstellen mit je 10 Meßwerten enthalten.

Diese umfassende Informationsverarbeitung ermöglicht zuverlässige Aussagen über die Dauerbelastbarkeit. Dadurch wird insbesondere vermieden, daß unklare Versuche mit großem finanziellem und zeitlichem Aufwand wiederholt werden müssen.

### 5. Optimale Gestaltung der Versuchsabläufe, Beispiel Motorenversuch

Einspritz- und Zündanlage haben wesentlichen Einfluß auf die Arbeitsweise des Otto-Motors, also auf seinen elastischen klopfreien Lauf, auf einen geringen Kraftstoffverbrauch und geringe Emission schädlicher Abgase.

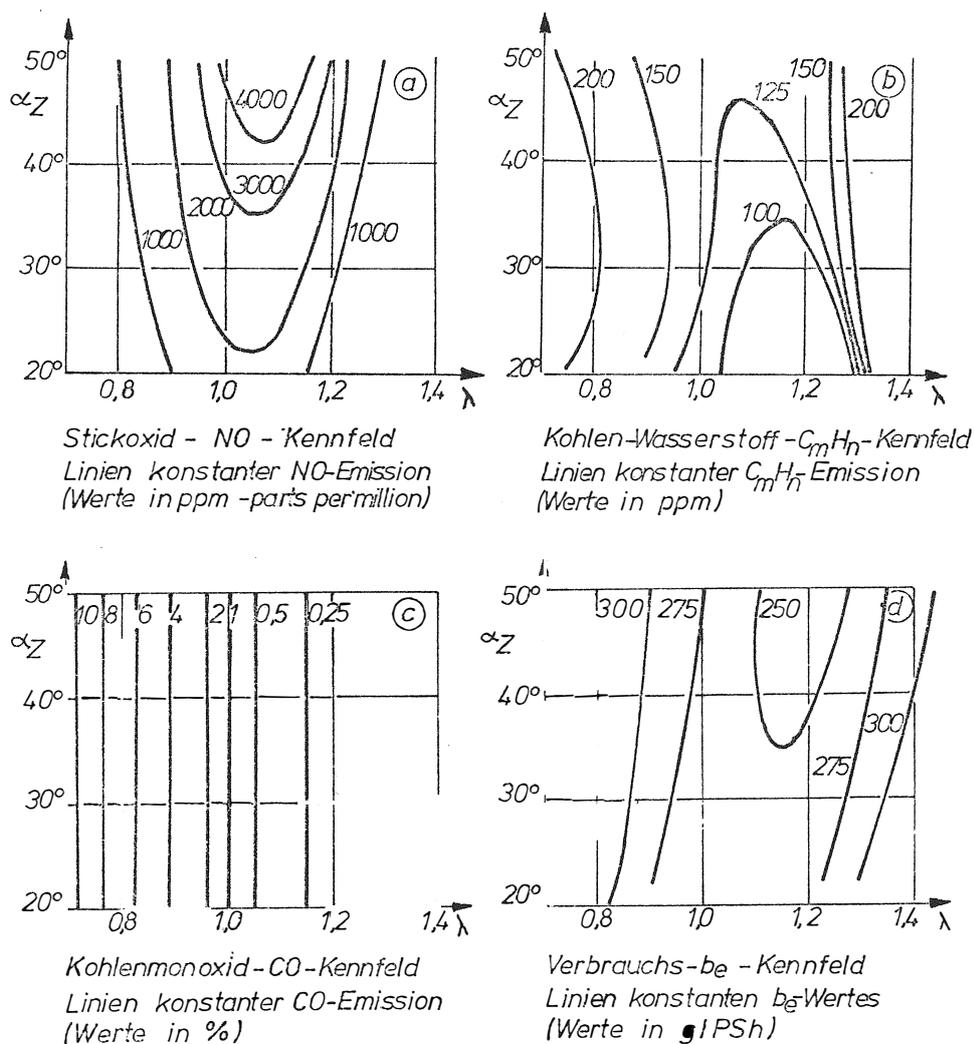


Abb. 7 Motorenkennfelder über Luftverhältnis  $\lambda$  und Zündwinkel  $\alpha_z$  bei festen Werten für Drehzahl  $n$  und Leistung  $N$

Diese Eigenschaften müssen in weiten Temperatur-, Drehzahl- und Drehmomentbereichen und daneben auch beim Starten, im Leerlauf, im Schiebepetrieb und bei raschen Änderungen der Betriebszustände erfüllt werden.

Ein Teil der Untersuchungen konzentriert sich auf die Bestimmung der Abgaskonzentrationen und des spezifischen Kraftstoffverbrauches abhängig von 4 Parametern: Luftverhältnis  $\lambda$  = (angesaugte Luftmenge / zur Verbrennung des eingespritzten Kraftstoffs erforderliche Luftmenge), Zündwinkel  $\alpha_z$ , Drehzahl  $n$  und Leistung  $N$ .

Wenn jeder der genannten 4 Parameter unabhängig vom anderen nur auf 10 verschiedene Werte eingestellt wird, ergeben sich  $10^4$  Messungen mit mindestens je 10 Meßwerten, also insgesamt  $10^5$  Meßwerte.

Die übergeordnete Aufgabe besteht z. B. darin, zu vorgegebenen Werten für Drehzahl  $n$  und Leistung  $N$  diejenigen Werte  $\lambda$  und  $\alpha_z$  anzugeben, bei denen der Motor gut läuft und die einzelnen Abgaskomponenten hinreichend schwach und der Verbrauch möglichst günstig sind.

Daraus ergeben sich Anforderungen an eine optimale Versuchsdurchführung:

1. Die zahlreichen Meßwerte müssen schnell und zuverlässig erfaßt werden.
2. Die Meßwertsätze müssen übersichtlich organisiert werden – sie müssen zur Verarbeitung leicht zu finden sein.
3. Abhängigkeiten einzelner Werte von irgendeinem beliebigen der 4 Parameter oder der anderen Werte müssen grafisch dargestellt werden können.
4. Die Messungen müssen zu übersichtlichen Kennfeld-Darstellungen verdichtet werden können, z. B. zu Linien konstanter Emission oder konstanten Verbrauchs über Luftverhältnis  $\lambda$  und dem Zündwinkel  $\alpha_z$ , Abb. 7 a–d.

Es ist unter Umständen zweckmäßig, die Versuche so zu fahren, daß ein Kennfeld unmittelbar gemessen wird, z. B. eine Versuchsfahrt längs den Linien  $NO = \text{const}$ , Abb. 7a.

5. Die Optima müssen leicht gefunden werden können. Abb. 8e zeigt den Optimierungsbereich und 8f den kürzesten Weg zum Optimum.

Die Aufgaben 1 und 2 sind gelöst, die Aufgabe 3 wird zur Zeit bearbeitet.

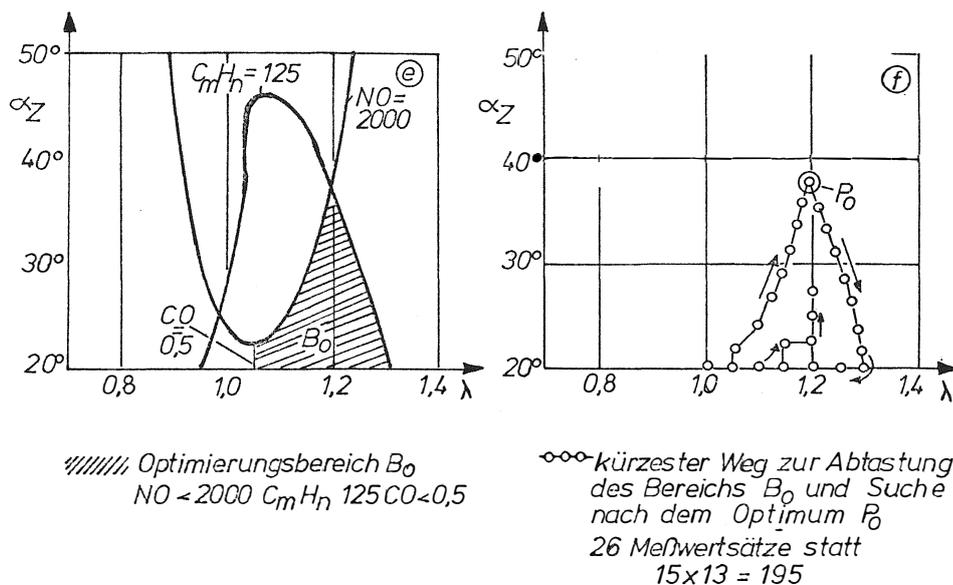


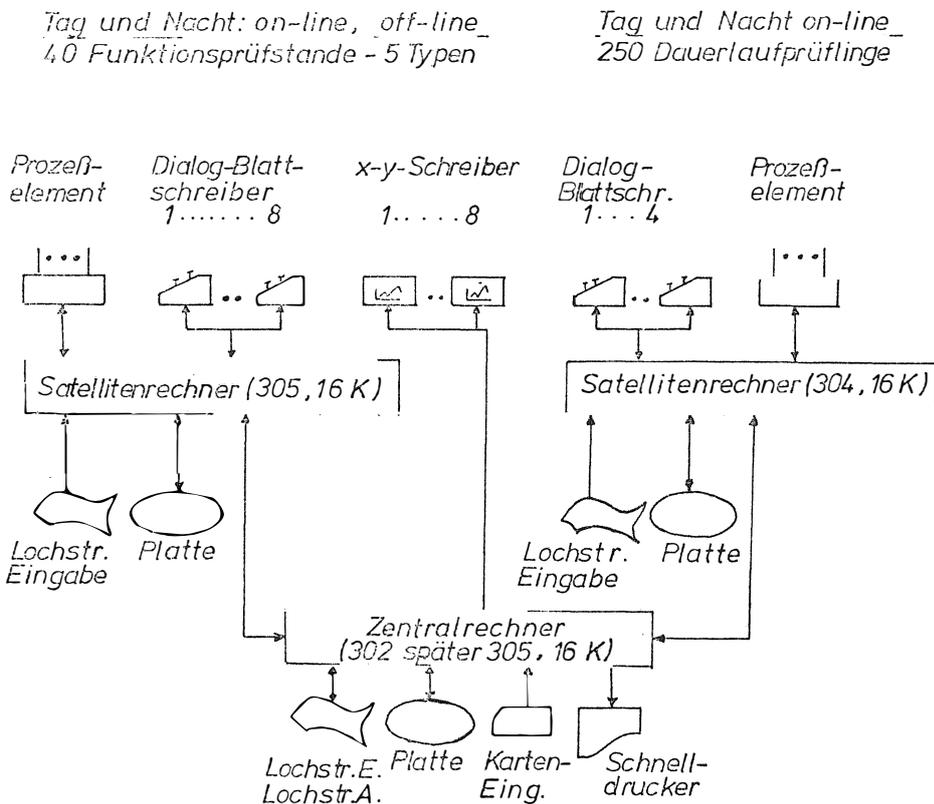
Abb. 8 Optimierung von Luftverhältnis  $\lambda$  und Zündwinkel  $\alpha_z$  bei festen Werten für Drehzahl  $n$  und Leistung  $N$

## 6. Rechenanlagen

Das Rechnersystem — siehe Abb. 9 — ist aus 3 gekoppelten Rechnern des Systems Siemens 300 so aufgebaut, daß der unmittelbare Prüfstandsanschluß mit Dialog, Meßwerterfassung, Prüfstandssteuerung sowie Speicherung, einfache Verarbeitung und Protokollierung von Daten von 2 Satellitenrechnern übernommen wird. Die umfangreicheren Auswertungen, die Protokollausgaben über Schnelldrucker, die Ausgabe von Zeichnungen, das Zwischenspeichern von Daten sowie Programmentwicklung und technischen Berechnungen werden auf dem Zentralrechner durchgeführt.

Nachts verändert sich die Wirkungsweise; der Satelliten-Rechner für den Dauerlauf bleibt on-line am Versuch, der Satelliten-Rechner für die Funktionsprüfstände führt unter der Regie des Zentralrechners off-line Auswertungen und technische Berechnungen durch.

Abb. 9 Rechnersystem, Aufbau und Funktion



*Tag und Nacht: on-line, off-line, Stapelverarbeitung*  
*Funktionsprüfstände*  
*Dauerlauf*  
*Progr. Entwicklung*  
*Technische Berechnung*

## *7. Zusammenfassung*

Nach dem in Abb. 1 gegebenen Programm wurde an Beispielen gezeigt, daß der Einsatz des Prozeßrechners im Labor- und Prüffeld zu wesentlichen Fortschritten führt, gekennzeichnet durch die Stichworte

wirtschaftlich  
schnell  
umfassend und  
optimal.

Die gegebene Darstellung sollte aber nicht darüber hinwegtäuschen, daß der Aufbau eines solchen Systems aufwendig, mühsam und risikoreich ist.

Insgesamt haben wir nach 3jähriger Arbeit die Erfahrung gemacht, daß es sich lohnt.

## **J. Zahn**

Hahn-Meitner-Institut für Kernforschung Berlin GmbH

Ein graphisches Sichtgerät für nukleare Experimente mit On-Line-Rechner

## Ein graphisches Sichtgerät für nukleare Experimente mit On-Line-Rechner

In zunehmendem Maße werden heute bei nuklearen Experimenten Rechner im On-Line-Betrieb eingesetzt. Dabei sind die typischen Aufgaben eines Rechners in einem Experimentablauf folgende:

- die Sammlung und Speicherung der Daten
- die Datenreduzierung und -analyse in Realzeit
- die selbsttätige Überwachung des Experimentes und der Instrumentierung durch eingeschobene Testphasen
- die Steuerung einzelner Experimentparameter oder des Experimentablaufes auf Grund vorgegebener Kriterien und sofortiger Meßwertverarbeitung.

Dabei spielen Datensichtgeräte ein wichtiges Bindeglied zwischen Mensch und Maschine, d. h. zwischen Experimentator und Experiment. Der Experimentator erhält über das Sichtgerät datenreduzierte Meßergebnisse sofort und anschaulich dargestellt, vorzugsweise in graphischer Form.

Im Folgenden sollen die Anforderungen an Datensichtgeräte für den Einsatz bei rechnergesteuerten nuklearen Experimenten umrissen werden.

Diese Anforderungen sind charakteristisch für den Einsatz eines Sichtgerätes im Dialogverkehr und bei der Verarbeitung und Ausgabe großer Datenmengen im On-Line-Betrieb.

Das Sichtgerät ermöglicht dem Experimentator

- die Überwachung der Datensammlung
- die Erkennung des augenblicklichen Status des Experimentes. Er kann somit sofort beurteilen, ob das Experiment noch sinnvoll und zufriedenstellend abläuft.
- die häufige Kontrolle der Meßeinrichtungen
- den sinnvollen und gezielten Eingriff in das Experiment durch Änderung von Parametern oder Änderung des Programmablaufes.

Das Sichtgerät dient also sowohl als schnelles alphanumerisches und graphisches Ausgabegerät, wie auch als Dateneingabegerät. Dazu dienen Rollkugel, Lichtgriffel, alphanumerische und Funktionstastatur, wie auch graphische Eingabegeräte.

Unter Funktionstastatur ist eine Tastatur zu verstehen, bei der die den einzelnen Tasten zugeordneten Funktionen nicht fest verdrahtet, sondern *frei programmierbar* sind. D. h. alle Funktionen, auch die Sichtgerätefunktionen werden vom Rechner gesteuert. Dies gestattet eine flexible Anpassung an die jeweiligen Erfordernisse und eine relativ einfache Hardware.

Darüberhinaus soll die Rechanlage bei optimalem Informationsfluß zum Sichtgerät und damit zum Experimentator durch die Ausgaberroutine möglichst wenig belastet werden, was besonders für Kleinrechner von Bedeutung ist. Dies führte zum Einsatz eines Sichtgerätes mit *Speicherröhre*.

Die Vorteile liegen klar auf der Hand:

- Die Darstellung erfolgt völlig flackerfrei.
- Es ist kein Bildwiederholungsspeicher erforderlich. Damit entfällt auch die Bereitstellung eines relativ großen Bereiches des Arbeitsspeichers für die Bildwiederholung.
- Bei Sichtgeräten mit Bildwiederholung ist die darzustellende Datenmenge auf ca. 4000 Punkte beschränkt, um ein noch flackerfreies Bild zu erhalten. Bei einem Speichersichtgerät ist die darzustellende Informationsmenge praktisch nur durch die Darstellungskapazität des Bildschirms begrenzt. Es können bis zu 40 000 Punkte/sec geschrieben werden.
- Die Information bleibt gespeichert, sie kann jederzeit durch weitere Informationen ergänzt werden. Die Datenverarbeitungsanlage wird dabei nur während der Informationsausgabe belastet.
- Die Übertragungsrate ist beliebig, auch langsame Übertragungsstrecken können dazwischengeschaltet werden. Der eigentliche Informationsblock steht dabei in Background-Speichern; im Arbeitsspeicher wird nur ein kleiner Pufferbereich von z. B. 64 Worten zur Verfügung gestellt.
- Im Prinzip können alle Zeichen und graphischen Darstellungen aus Punkten aufgebaut werden. Bei Bildwiederholung wäre dies schon wegen der langen Bildaufbauzeiten nicht möglich. Dies bedeutet aber eine relativ einfache Hardware.

Der Aufbau des Sichtgerätes ist folgender:

Das Sichtgerät besteht aus der Speichersichteinheit Tektronix 611, Mod. 162C, der Datenausgabesteuerung und der Dateneingabesteuerung. Dazu gehören ferner ein Datenverteiler für den parallelen Betrieb von zwei zusätzlichen Monitoren. Die Monitore können dabei bis zu 30 m vom Hauptsichtgerät entfernt sein.

Ausgabe- und Eingabesteuerung sind streng modular aufgebaut. Der modulare Aufbau gestattet unterschiedliche Ausbaustufen entsprechend dem geforderten Einsatz.

Die Ausgabe-Steuerung ist streng geteilt in ein Interface und das eigentliche Ausgabe-Steuerwerk. Die Kopplung beider erfolgt über eine standardisierte Nahtstelle mit einem 12-Bit-Datenbus und einer Reihe von Steuerleitungen, wie Auswahlleitungen für die Darstellungsarten und Submodes, sowie die Datenübernahme vom Interface zum Steuerwerk.

Das Interface wird rechner- oder übertragungsstreckenspezifisch ausgeführt. Damit ist eine Anpassung des Sichtgerätes an jede beliebige Rechnernahtstelle leicht möglich. Das z. Z. vorhandene Interface ist für 24 Bit Parallelübertragung ausgelegt. Es kann so direkt an den Nahtstellen P3AS bzw. P3KS der Siemens 301 oder 305 betrieben werden. Auch die Zwischenschaltung der Übertragungseinrichtung P3UK ist möglich.

Für die AKZ-Nahtstelle ist ein Ausgabe- und Eingaberegister entwickelt worden. Über dieses Register kann das Sichtgerät direkt mit dem Akku verkehren. Dieses Register ist aber nicht Bestandteil des Interfaces, es besitzt nur die gleichen Schnittstellenbedingungen wie die P3-Nahtstelle.

Da das Speichersichtgerät eine beliebig langsame Datenausgabe erlaubt, kann an Stelle eines parallelen Interfaces auch ein solches für serielle Datenübertragung eingesetzt werden. Ein solches Interface ist z. Z. in Entwicklung. Es enthält eine nach den CCITT-Empfehlungen V.24 standardisierte Nahtstelle.

Damit wird der Betrieb des Sichtgerätes über Fernschreib- oder Fernsprechleitungen auch unter Zwischenschaltung von Modems möglich.

Dieses serielle Interface ist auch für den Anschluß an das System 300 gedacht. Dazu soll eine spezielle AKZ-Ausgabeeinheit mit V.24-Nahtstelle für die Umwandlung des parallelen Datenwortes in ein serielles entwickelt werden. Die 24 Bit jedes Datenwortes werden in 4 Hexaden aufgeteilt. Die einzelnen Hexaden werden zu 8 Bit ergänzt und asynchron, d. h. mit Start-Stop-Schritten versehen, übertragen.

Im Gegensatz zum parallelen Interface, das die Daten vom Rechner durch Anforderung und Quittierung durch eine Rückmeldung abrufen, ist dieses Hand-shake-Verfahren beim seriellen Interface nicht möglich. Die V.24-Nahtstelle sieht eine solche Betriebsart nicht vor.

Die Daten müssen also von der Ausgabesteuerung am Rechner mit einer festen Datenübertragungsrate ausgesendet werden, die sich nach dem Mode mit der längsten Verarbeitungszeit im Steuerwerk richtet. Während beim Hand-shake-Verfahren mit optimaler Übertragungsrate bis zu 30 000 Punkte/sec oder 660 cm Vektoren je sec oder im Mittel 1 800 Zeichen/sec übertragen werden können, beschränkt die längste Vektorverarbeitungszeit bei seriellen Interface die Übertragungsrate auf etwa 100 Datenworte zu 24 Bit je sec, d. h. 100 Vektoren oder Punkte, bzw. 400 Zeichen je sec.

Die Übertragungsrate kann auch hier optimal gewählt werden, wenn über den Eingangskanal von der Ausgabe-Steuerung her ein Quittungssignal als Fertigmeldung zum Rechner gesendet wird, welches von der dort vorhandenen Nahtstellen-Anpassung als Aufforderung für eine erneute Datenwortaussendung interpretiert wird. Ein solches Quittungszeichen darf aber zu keiner Verwechslung mit Eingabedaten führen. Hierfür bieten sich z. B. Steuerzeichen im ASCII an.

Die Dateneingabesteuerung enthält analog ein Eingabe-Steuerwerk und ein Eingabe-Interface.

Das Interface dient hier ebenfalls zur Anpassung des Steuerwerkes an die jeweilige Rechner-nahtstelle. Zunächst ist es auch hier für 24 Bit parallele Datenübertragung zum Anschluß an die P3EG, P3KS bzw. über das AKZ-Register an den AKZ ausgelegt. Es ist aber ebenfalls ein serielles Interface mit V.24-Nahtstelle geplant.

Das Eingabe-Steuerwerk enthält 4 Eingabe-Kanäle für Rollkugel, Alphanumerische Tastatur, Funktionstastatur und graphische Eingabeplatte. Zur Verschlüsselung der Daten der Funktionstastatur oder für die Umsetzung von ASCII-Zeichen in den Siemens-Interncode können Codewandler eingesetzt werden.

Die verschiedenen Darstellungsarten sind folgende:

#### 1. Random Point Plot (freie Punktpositionierung)

Durch Übertragung der X-Koordinate und Y-Koordinate jedes einzelnen Punktes ist jede beliebige Darstellung möglich.

Die Bildaufteilung ist in X-Richtung 1 400 Punkte ( $\cong$  11 Bit) und in Y-Richtung 1 024 Punkte ( $\cong$  10 Bit) bei einem Punktabstand von 0,15 mm. Der Punktdurchmesser beträgt 0,25 mm, so daß benachbarte Punkte überlappen und so geschlossene Linienzüge entstehen.

Der Random Point Plot ist der einfachste Mode, er erfordert nur zwei Register und zwei DAC, sowie eine Hellsteuerschaltung. I.a. sind aber andere Darstellungsarten wesentlich zweckmäßiger und man wird den Random point plot in erster Linie für Grundpositionierungen von Darstellungen in anderen Modes verwenden. Aus diesem Grund kann die Positionierung hier auch dunkel (ohne Helltastung) erfolgen.

#### 2. Sequentielle Punktpositionierung

Der Hauptanwendungsbereich liegt hier bei der Spektrendarstellung.

Es erfolgt nur die Übertragung der Y-Koordinate, die X-Koordinate wird durch Inkrementierung des X-Registers in der Steuerung erzeugt. In einem 24-Bit-Datenwort lassen sich so zwei Y-Koordinaten übertragen.

Dieser Mode erfordert gegenüber dem Random Point Plot keinen zusätzlichen Aufwand, da die X- und Y-Register bereits vorhanden sind. Er gestattet darüber hinaus in X-Richtung eine Dehnung um den Faktor 2,4 oder 8, wodurch Ausschnitte dargestellt werden können.

Mit dieser Darstellungsart sind auch mehrparametrische Darstellungen, d. h. isometrische Darstellungen möglich. Jedoch ist diese Darstellungsart sehr unbefriedigend, da eine Zuordnung der Peaks zur Grundlinie nicht erkennbar ist. Wesentlich klarer wird die Zuordnung bei Anwendung des Vektormodes.

### 3. Vektor-Mode

Die Vektoren sind inkrementell aus Punkten aufgebaut. Der hier eingesetzte Vektor-generator ist ein Binary-Rate-Multiplier, der entsprechend der Vektorlänge und Richtung in bestimmter Weise die X- und Y-Register inkrementiert. Bei jeder Änderung eines Registerinhaltes wird hellgetastet.

Dieses digitale Verfahren hat gegenüber analogen Vektorgeneratoren den Vorteil, daß der Vektor immer streng im Bildraster geschrieben wird.

Neben der klaren Zuordnung der Peaks bei isometrischer Darstellung (Abb. 1) bringt dieser Mode einen weiteren Vorteil:

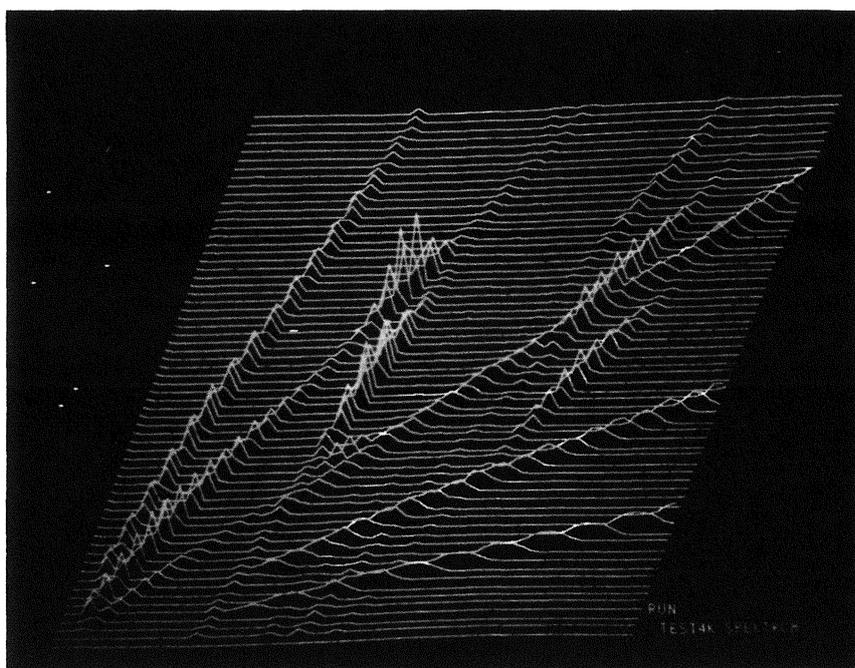


Abb. 1 VEKTOR-Darstellung  
Isometrische Darstellung 64 x 64 Kanäle

Die Positionierung der Vektoren erfolgt durch Ausgabe von *relativen* Koordinaten  $\pm \Delta X$ ,  $\pm \Delta Y$  (je 9 Bit mit Vorzeichen), d. h. die Koordinatenangabe bezieht sich nicht auf den Koordinatenursprung, sondern auf den Endpunkt des vorherigen Vektors. Damit läßt sich allein durch Änderung der Anfangsposition der Vektorzüge der Isometriewinkel und der Zeilenabstand verändern.

Man erreicht damit eine scheinbare Drehung der gesamten Darstellung.

Zur Anfangspositionierung können auch Dunkelvektoren geschrieben werden.

Selbstverständlich sind im Vektormode auch Darstellungen von Einzelspektren möglich. Durch die relative Positionierung ist auch hier die gedehnte Ausschnittsdarstellung softwaremäßig leicht möglich (Abb. 2).

Die Vektoren können als durchgehende Linie oder gestrichelt oder durch Helltastung nur des Vektorendpunktes geschrieben werden. Kombinationen dieser Vektorarten können leicht zur Darstellung von Soll- und Istkurven verwendet werden (Abb. 3).

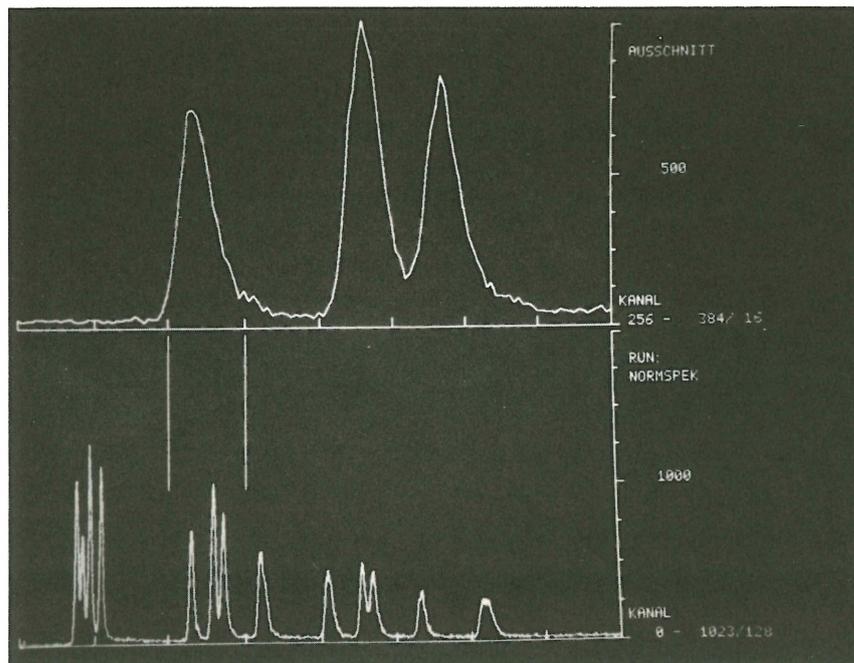


Abb. 2 *VEKTOR-Darstellung*  
 Gesamtspektrum 1 024 Kanäle mit Ausschnittmarkierung  
 u. gedehnter Ausschnitt 128 Kanäle

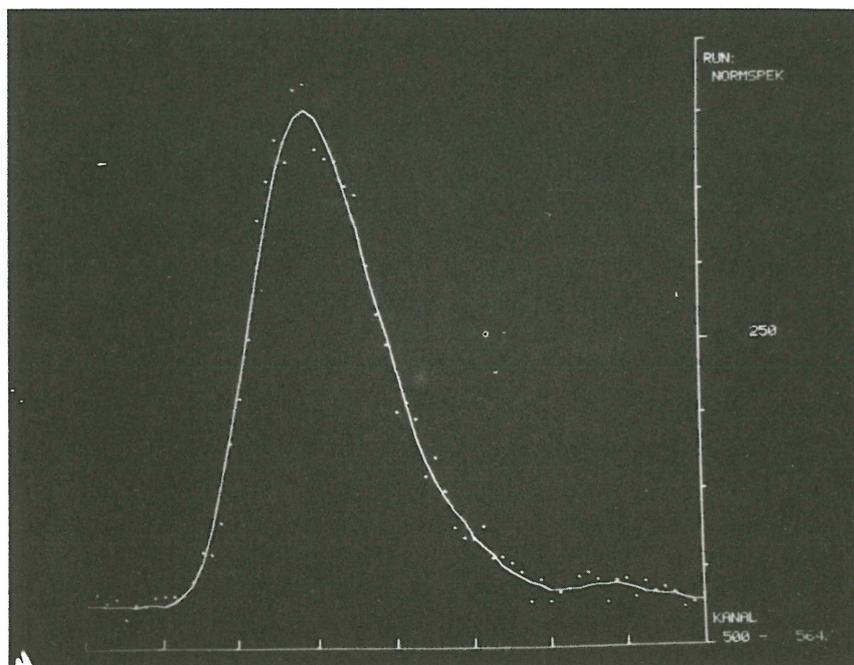


Abb. 3 *Einzelpunktvektor*  
 (Rohdaten) und Vektordarstellung (Geglättetes Spektrum)

Benutzt man im rechteckigen Bildformat für die graphischen Darstellungen i. a. den quadratischen Bereich mit 1 024 x 1 024 Punkten, so steht rechts eine Schriftleiste für Kommentare zur Verfügung.

#### 4. Alphanumerischer Mode (Abb. 4)

Im Prinzip ist die Zeichendarstellung auch im Vektormode effektiv möglich. Es wurde jedoch ein modularer Zeichengenerator vorgesehen, um den Rechner auch von der Zeichengenerierung zu entlasten.

Zur Zeichenauswahl werden vom Rechner 6 Bit/Zeichen übertragen, d. h. ein 24-Bit-Datenwort enthält 4 Zeichen. Mit diesen 6 Bit wird ein integriertes Read-Only-Memory angesteuert. Die Codierung des ROM ist ASCII (6 Bit Subset). Durch einen vorgeschalteten Codewandler ist aber auch jeder andere Code anwendbar, z. B. der Siemens Interncode. Das ROM gestattet die Darstellung von 64 Zeichen, es werden aber nur 53 Zeichen ausgenutzt, die restlichen 11 Codekombinationen werden als Funktionen gewertet, wie Wagenrücklauf, Zeilenwechsel, Zwischenraum, Rückschritt, Zeilenrücksprung usw.

Das Zeichenraster ist eine 5 x 7 Matrix. Die Zeichenhöhe beträgt etwa 2,4 mm, der Zeilenabstand 32 Inkremente, der Zeichenabstand 16 Inkremente. Dies ergibt eine Darstellungskapazität von 32 Zeilen zu je 86 Zeichen, d. h. rund 2 750 Zeichen je Bild. Ferner ist kursive Schrift und doppelte Zeichengröße möglich.



Abb. 4 Alphanumerische Darstellung

86 Zeichen je Zeile, 32 Zeilen je Bild, Klein- u. Großformat, normal u. kursiv

#### 5. Incrementelle Punktpositionierung

Dieser Mode gestattet von einem Punkt ausgehend nur Inkrementschritte in 8 Richtungen um  $\pm X$  und  $\pm Y$ , die hell oder dunkel ausgeführt werden können.

Die Vorteile dieser Darstellungsart sind der sehr geringe Hardware-Aufwand und die Möglichkeit der Anwendung der gesamten CALCOMP-Plotter-Software für das Sichtgerät. Hierfür liegt uns die ausdrückliche Genehmigung der Fa. CALCOMP vor.

Für die Darstellung eines Punktes sind 6 Bit erforderlich, d. h. jedes 24-Bit-Datenwort enthält die Angaben für 4 Punkte. Die Schreibzeit auf dem Bildschirm kann auf etwa 1/200 der Plotzeit herabgesetzt werden. Eventuelle Fehler können so auf dem Bildschirm erkannt und vor dem Plot korrigiert werden.

Neben den vielen Vorteilen der Speicherröhre gibt es aber auch eine Reihe von Nachteilen:

- Es ist keine Intensitätsmodulation möglich. Damit entfällt die helligkeitsgesteuerte MAP-Darstellung bei mehrparametrischen Messungen.
- Wegen der fehlenden Bildwiederholung entfällt auch die Lichtgriffelanwendung.
- Es ist kein Live-Display möglich, wobei Live-Display hier im Sinne eines bewegten Bildes zu verstehen ist. Es kann zwar Bildinformation zu der bereits bestehenden hinzugefügt werden, zur Änderung der Information muß das Bild aber gelöscht und neu geschrieben werden.

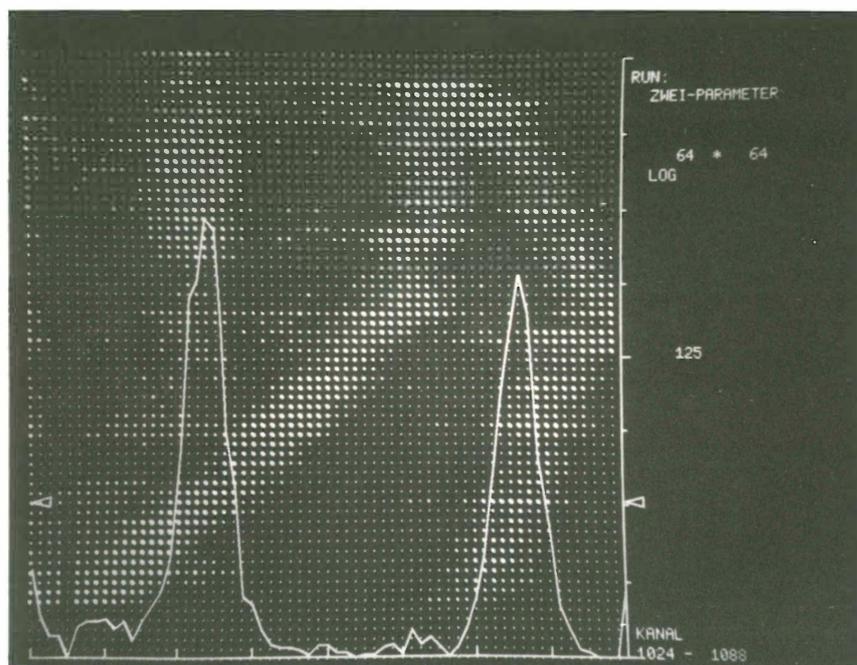


Abb. 5 *MAP-Darstellung*  
64 x 64 Kanäle mit horizontalem Schnitt

Für alle diese Einschränkungen gibt es aber brauchbare technische Lösungen:

- Die MAP-Darstellung erfolgt mit Hilfe eines MAP-Generators, wobei die übertragene Information von 5 Bit/Punkt die Punktgröße in 8 Stufen steuert. Der Eindruck ist von dem einer helligkeitsgesteuerten Darstellung kaum zu unterscheiden. Auch Raster der Größe  $128 \times 128$  ( $\hat{=}$  16 K) sind bei einem Speicherschirm ohne weiteres darstellbar. Zusätzlich kann man in diese Darstellungen auch horizontale oder vertikale Schnitte einblenden (Abb. 5).

An Stelle der MAP-Darstellung kann auch eine Höhenliniendarstellung mit Vektoren gewählt werden (Abb. 6).

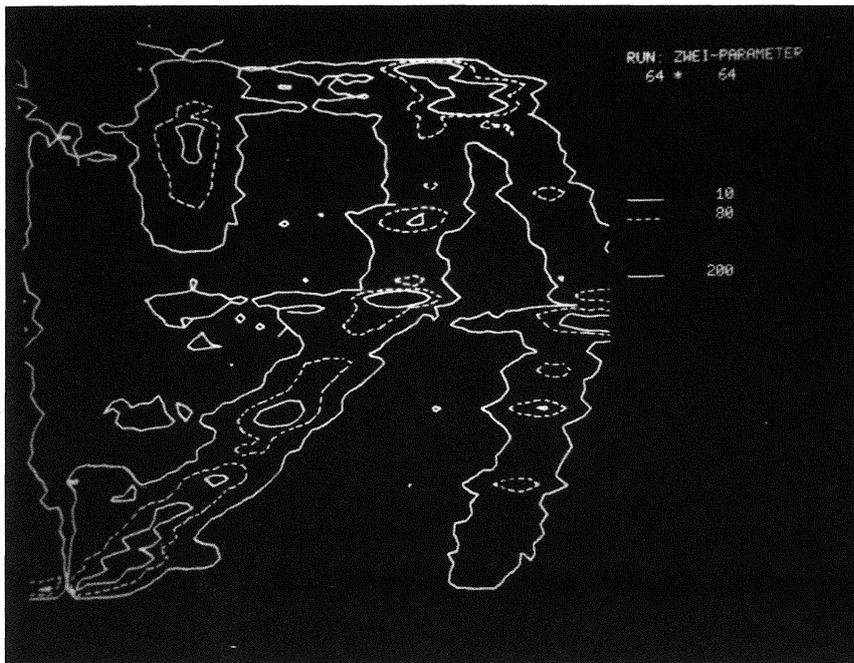


Abb. 6 Höhenlinien-Darstellung  
64 x 64 Kanäle

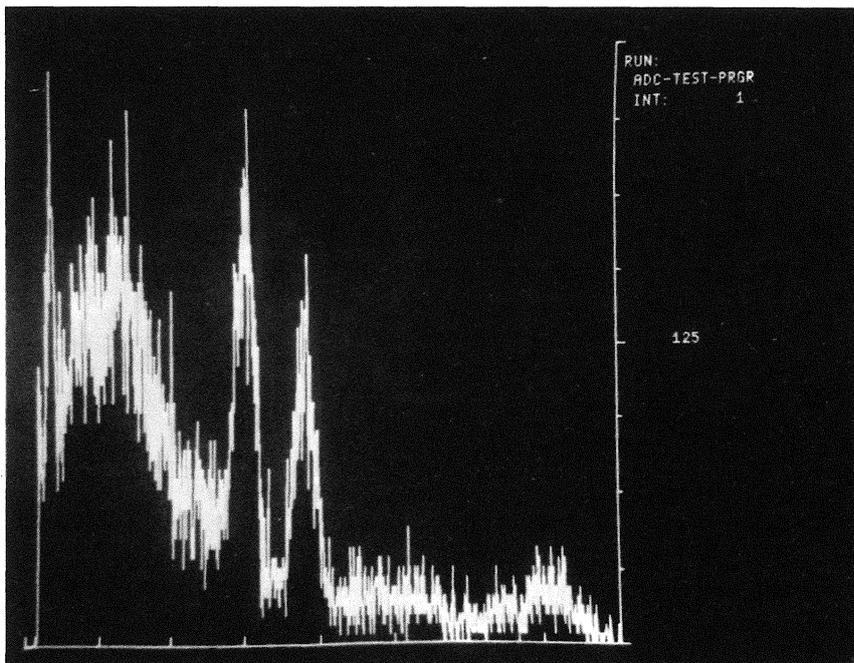


Abb. 7 Realzeitdarstellung  
32 000 Impulse/Intervall bei 25 sec/Intervall Meßzeit. Mittlere Rate nach 1 Intervall

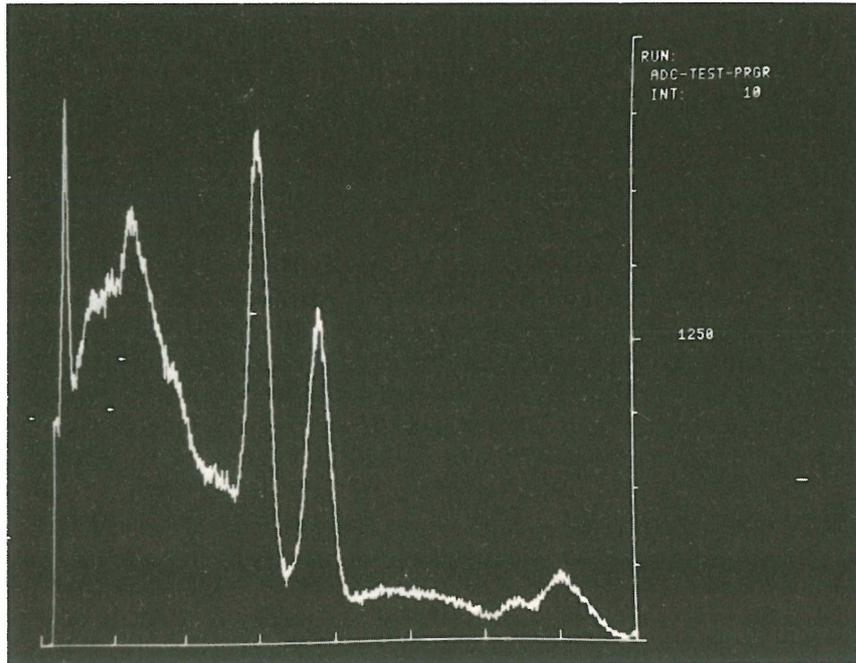


Abb. 8 Mittlere Rate nach 10 Intervallen

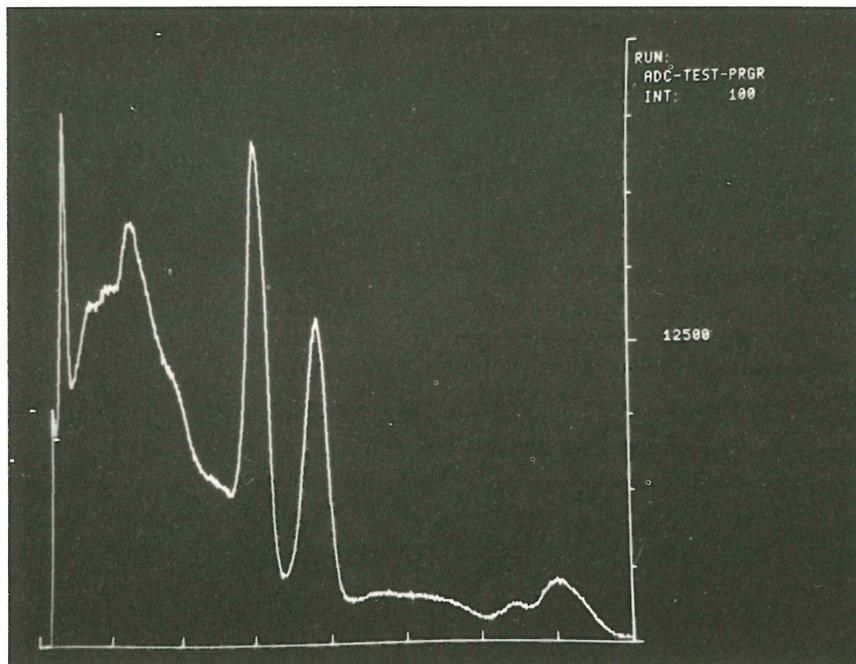


Abb. 9 Mittlere Rate nach 100 Intervallen

- An Stelle eines Lichtgriffels kann man zur Kennzeichnung bestimmter Bildpunkte ohne weiteres eine Rollkugel einsetzen, mit der eine Marke über den Bildschirm verschoben werden kann. Diese Marke besteht aus einem hardwaremäßig erzeugten Kreuz, das im sog. Write-Through-Mode mit geringerer Intensität nichtspeichernd geschrieben wird. An Stelle einer Rollkugel kann auch ein wesentlich billigerer Steuerknüppel eingesetzt werden.
- Die fehlende Möglichkeit des Live-Displays läßt sich dadurch ersetzen, daß man aus einer sich ändernden Informationsmenge, z. B. aus einem wachsenden Spektrum, in gleichlangen Abständen „Schnappschüsse“ darstellt. Diese Abstände können 10 sec oder auch 10 min lang sein, je nach Änderungsgeschwindigkeit. Für die meisten Anwendungen dürfte dieses Verfahren ohne weiteres ausreichend sein. Wenn man nun dabei nicht die Anzahl der Ereignisse in jedem Kanal darstellt, sondern die nach jedem Intervall durch Mittelwertbildung neu errechnete mittlere Rate, d. h. die Anzahl der Ereignisse je Zeitintervall, so ergibt sich der zusätzliche Vorteil, daß ein wachsendes Spektrum nicht ständig aus dem Bildbereich herauswächst, sondern es wird mit stets gleicher Größe und immer geringerer statistischer Streuung dargestellt (Abb. 7 – 9).

*Lit.:* Abend, Klessmann, Zahn u. a.:

Ein graphisches Sichtgerät für nukleare Experimente mit On-Line Rechner

Hahn-Meitner-Institut für Kernforschung Berlin GmbH. Bericht HMI-B 100 (NDV 4), Juli 1970, 96 S., 35 Abb.

## **P. Abend**

Ein modulares Programmsystem für grafische Darstellungen

## Ein modulares Programmsystem für grafische Darstellungen

Beim Einsatz von Prozeßrechnern in Experimentiersystemen oder bei der Laborautomatisierung ist der Einsatz von grafischen Sichtgeräten unerlässlich. Aus diesem Grunde wurde am Hahn-Meitner-Institut Berlin eine Sichtgeräte-Steuerung zum Anschluß eines Speichersichtgerätes an die Prozeßrechner des Systems 300 entwickelt. Es bestand nunmehr die Aufgabe, ein Software-System für dieses Sichtgerät zu erstellen. Dabei ergeben sich eine Reihe von Randbedingungen, die hier kurz aufgeführt werden sollen:

### *1. Randbedingungen*

Das zu erstellende Software-Paket muß unbedingt die Größe der benutzten Prozeßrechner berücksichtigen. Es darf nur einen geringen Teil des gesamten Arbeitsspeichers einnehmen, der für die eigentlichen Prozeßaufgaben frei bleiben muß. Aus diesem Grunde ist die Benutzung der Calcomp-Plotter-Software nicht möglich. Die Größe der Bildausgabepuffer kann wegen der Speichereigenschaften der Bildröhre klein gehalten werden.

Da jedoch auch andere grafische Ausgabegeräte verwendet werden, wie z. B. Digital-Plotter oder elektrostatische Plotter, die eine andere Daten- und Befehlsstruktur besitzen als das genannte Sichtgerät, soll eine gleichartige Benutzung für die genannten bzw. später noch hinzukommenden grafischen Ausgabegeräte erreicht werden. Dabei sollen jedoch die Eigenschaften des jeweiligen Gerätes aus Gründen der Programmlänge, der Programmlaufzeit und einer eventuell bestehenden langsamen Fernübertragung möglichst vollständig ausgenutzt werden.

Da auch das Sichtgerät in seinen Funktionen (wie Vektor, Charakter, Point Plot, Increment usw.) modular aufgebaut und erweiterungsfähig ist, wurde auch die Software modular angelegt. Fehlen beispielsweise bestimmte Funktionen im Sichtgerät, können sie leicht durch ein Programmmodul ersetzt werden.

Die Rohdaten aus den Messungen liegen hier fast ausschließlich als Festkomma-Werte vor. Die Benutzung von Winkelfunktionen und die notwendigen Umwandlungen von Fest- in Gleitkomma und umgekehrt hätten einen bedeutenden Grundaufwand ausgemacht. Es war also notwendig, nach geeigneten Algorithmen zu suchen, um Vektoren, Kreisbögen usw. auf einfachere Weise zu erzeugen. Auch sollte nach Möglichkeit die Gleitkommaarithmetik vermieden werden, da diese auf kleineren Maschinen wie S 301 simuliert werden muß.

Aus diesen Betrachtungen ergibt sich folgender Aufbau des grafischen Programmsystems.

### *2. Aufbau des grafischen Ausgabesystems*

Das System ist hierarchisch in drei Ebenen aufgeteilt worden. Das Kernstück ist dabei die mittlere Ebene, die als Geräteebene bezeichnet werden soll. Sie enthält die Module für die einzelnen grafischen Elemente, wie Position, Gerade, Kreis und einfache Symbole. Die Module setzen die vom Anwender vorgegebenen Daten in die gerätespezifischen Datenformate um und geben diese an die E/A-Ebene ab (siehe Abb. 1). Befehls Worte werden unter

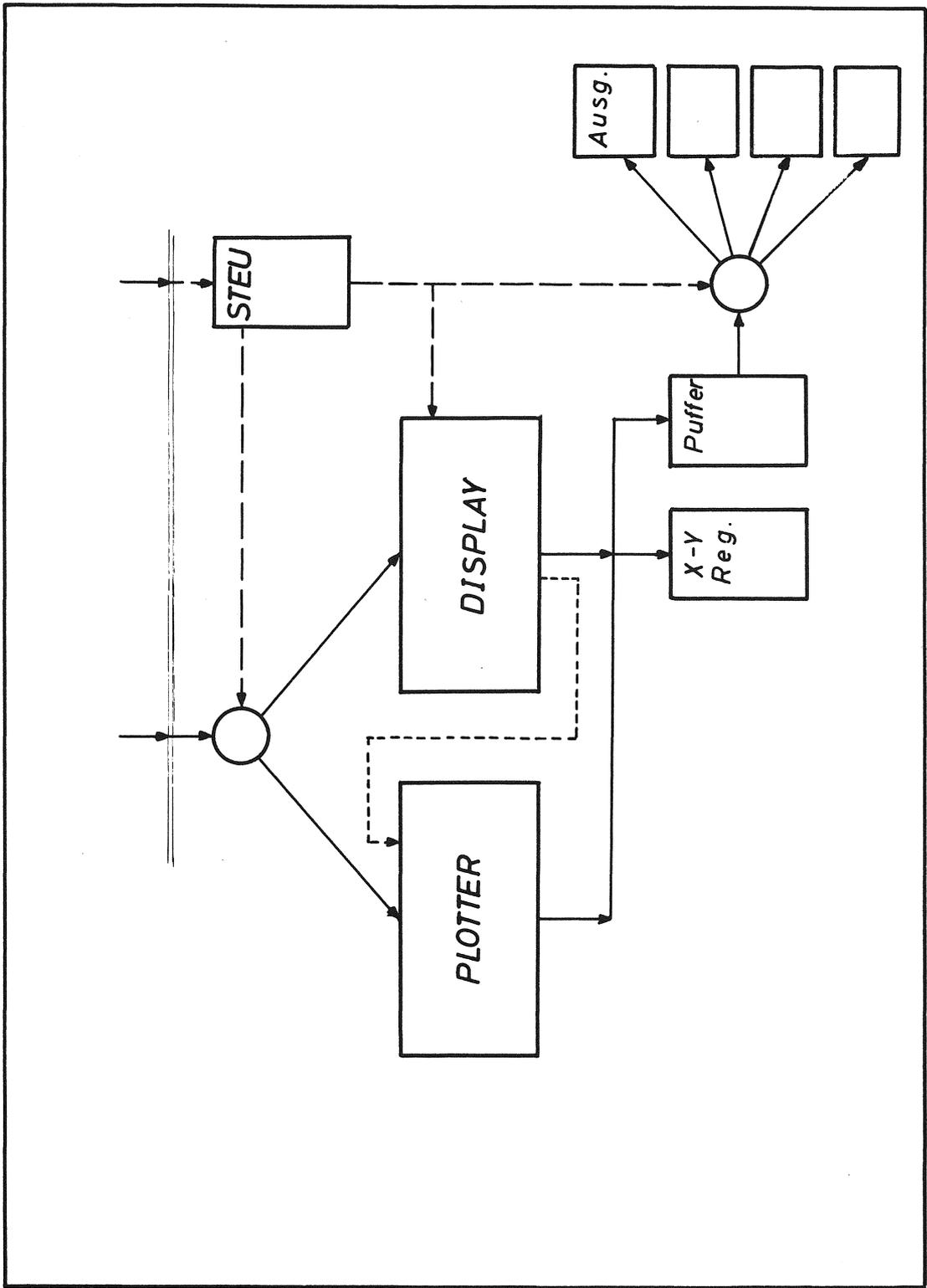


Abb. 1

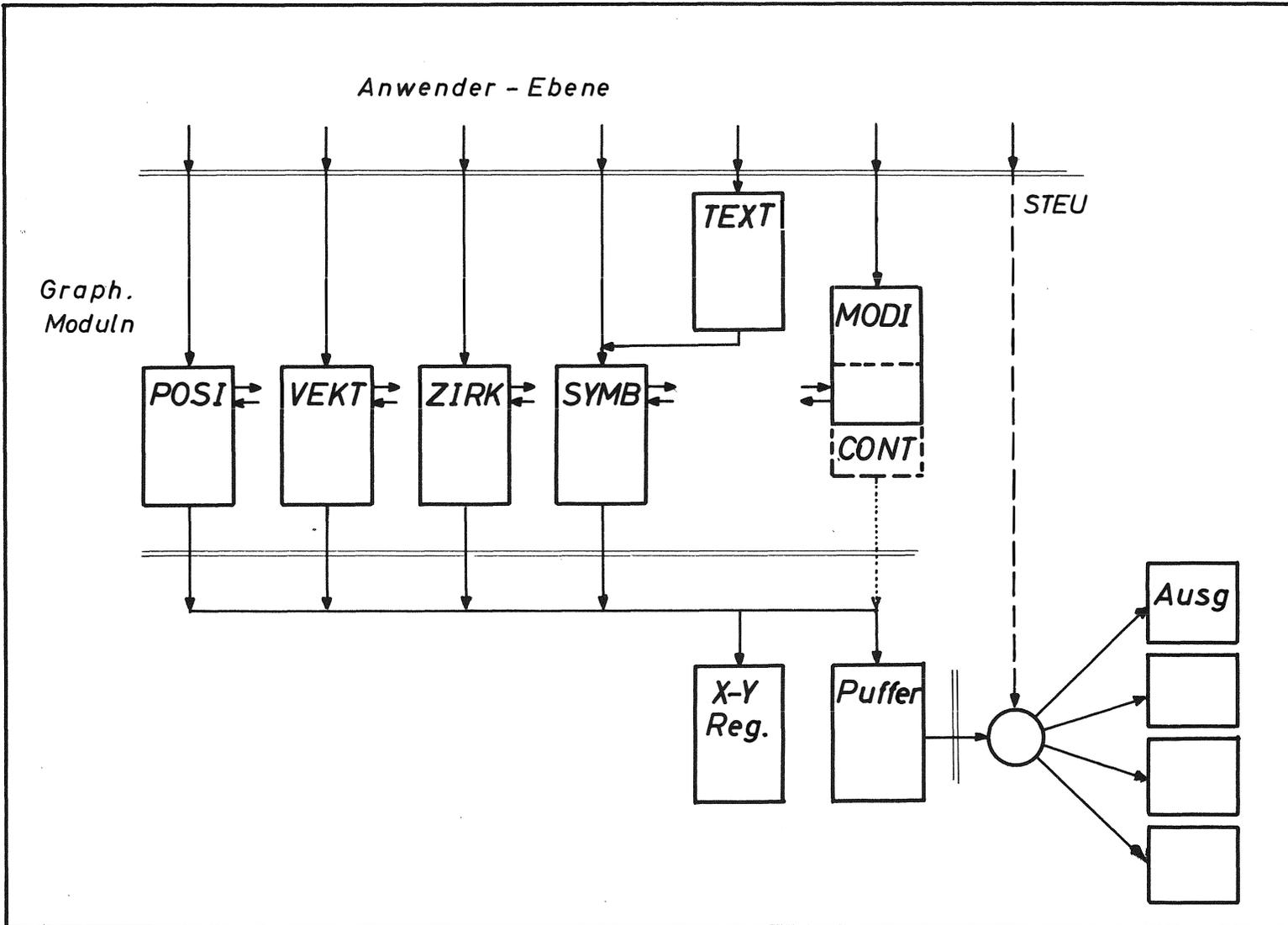


Abb. 2

der Kontrolle des Moduls CONT nur dann abgegeben, wenn ein Wechsel von einem Gerätemode in einen anderen erfolgen muß. Bei Geräten ohne Befehlsstruktur (Calcomp-Plotter) fehlt dieser Baustein. Mit einem weiteren Baustein MODI kann die Vergrößerung oder/und Drehung von Bildern oder Bildteilen durch vorgebbare Faktoren bzw. Winkel erreicht werden. Weiterhin wird über den Baustein X-Y REG die jeweilige Position des Schreibstrahls bzw. -stiftes mitgeführt. Dadurch wird es möglich, nicht nur mit relativen (VEKT), sondern auch mit absoluten Vektoren (POSI) zu arbeiten. Die erreichte Position kann auch vom Anwenderprogramm ausgelesen werden. Ein weiterer Baustein WINDOW (welcher noch nicht implementiert ist) soll für die Darstellung von Bildausschnitten sorgen, so daß das Anwenderprogramm nur die Ausschnittskordinaten anzugeben hat.

Die E/A-Ebene übernimmt nun die Ausgabe der gerätespezifischen Daten an das externe Gerät. Auch hier sind verschiedene Module vorgesehen. So ist es möglich, grafische Ausgabegeräte an verschiedenen Nahtstellen zu betreiben, z. B. das HMI-Sichtgerät parallel über P3K, P4K oder seriell über camac. Auch kann anstelle der direkten Ausgabe an einen Digitalplotter eine Ausgabe auf einen Externspeicher gewünscht werden, um die Verweilzeit des Programmes im Rechner gering zu halten. Die Zeichnung kann dann später off-line über ein sehr kurzes Programm, bestehend aus den Bausteinen EIN u. AUS, erfolgen.

Eine übergeordnete Ebene, die als Funktionsebene bezeichnet werden kann, wird zur Zeit spezifiziert. Sie soll ebenfalls modular sein und Bausteine wie Polygon, Achsen u. ä. enthalten. Diese Module sind geräteunabhängig, sie beziehen sich auf die Nahtstelle der Geräteebene.

Der Anwender hat auf alle Ebenen über normierte Nahtstellen Zugriff. Er kann auch Geräteebene und E/A-Ebene für verschiedene Geräte parallel benutzen (Abb. 2). Die Geräteebenen-Nahtstelle wird dann um eine Weiche, die vom Anwenderprogramm angesteuert werden kann, erweitert. Die Weiche schließt dann an die Nahtstellen der Geräteebenen an.

### *3. Realisierung des Systems*

Das grafische Ausgabesystem ist zunächst für den Calcomp-Plotter realisiert worden. Da die Sichtgerätesteuering im Incrementmode die gleiche Datenstruktur wie der Plotter besitzt, kann das gleiche System auch für das Sichtgerät verwendet werden. Ein gleicher Satz von Moduln für das Sichtgerät wird zur Zeit erstellt. Alle Module sind in Prosa 300 geschrieben. Es sind Unterprogramme, die als Makroerklärungen vorliegen. Der Anwender kann sich dadurch die benötigten Funktionen in Form von Makroaufrufen in sein Programm einbauen. Er kann das grafische Ausgabesystem dadurch an seine Aufgabenstellung anpassen. Der Funktionsaufruf (Unterprogrammssprung und Parameterversorgung) kann ebenfalls als Makro definiert werden.

Ein System für einen Calcomp-Plotter – wie in Abb. 1 dargestellt – hat eine Länge von etwa 650 FW. Ein entsprechendes System für das Sichtgerät wird etwa 400 FW einnehmen.

## **Walter Tenten**

Zentrallabor für Elektronik, Abteilung Nukleare Elektronik  
und On-Line-Datenverarbeitung  
der Kernforschungsanlage Jülich GmbH.

Das Display-Terminal TEKTRONIX 4002 am Rechner Siemens 305

## Das Display-Terminal TEKTRONIX 4002 am Rechner S 305

Der folgende Bericht beschreibt die besonderen Eigenschaften des graphischen Sichtgerätes TEKTRONIX T 4002 und erklärt den Anschluß dieses Terminals an einen Prozeßrechner vom Typ Siemens 305.

Das graphische Terminal T 4002 enthält einen Oszillographen-Bildschirm mit Speicher-Eigenschaft im Format  $210 \times 152 \text{ mm}^2$  als Ausgabemedium und als Eingabeteil eine Tastatur, die außer den normalen Schreibmaschinentasten auch solche für spezielle Funktionen enthält. Weitere Bestandteile des Gerätes sind eine Steuerung mit eingebautem Zeichengenerator, eine Anschaltung für einen Rechner oder ein Fernschreibnetz und einen Zusatzkanal zum Anschluß an oder von Hilfsgeräten. Diese in einem Gerät zusammengefaßten Elemente ermöglichen die folgenden Betriebsarten des Terminals im Zusammenwirken mit einem Rechner.

### 1. *Alphanumerische Arbeitsweise*

Ein- und Ausgabe von Zeichen und Buchstaben, auch Kleinbuchstaben, in zwei verschiedenen Größen, außerdem kursiv oder gerade. Der Zeichengenerator verwendet eine Matrix von  $7 \times 9$  Punkten. Die Kapazität des Bildschirms beträgt 39 Zeilen zu je 85 Zeichen. Eine vierzigste Zeile, die nicht auf dem Bildschirm gespeichert wird, sondern mit Hilfe eines eingebauten MOS-Schieberregisters erneuert wird, kann zum Zusammensetzen von Zeilen ohne eine gleichzeitige Rechnereingabe benutzt werden. Diese Zeile wird auf einen Knopfdruck hin als Ganzes gesendet.

Die graphischen Betriebsarten beruhen alle auf einem gedachten Punktraster von  $760 \times 1024$  Punkten; für die Definition eines Punktes sind im allgemeinen  $2 \times 10$  Bit notwendig.

### 2. *Incremental Plot*

Bei dieser Betriebsart, die von mechanisch arbeitenden Plottern übernommen wurde, ist ein Punkt immer durch seine relative Lage zum vorigen Punkt unter Zuhilfenahme des Rasters definiert. Der Abstand beträgt jeweils einen Rasterpunkt, und es sind acht verschiedene Himmelsrichtungen hell und dunkel möglich.

### 3. *Point Plot*

Hiermit kann ein Punkt durch Angabe eines x-y-Koordinatenpaars auf einen beliebigen Punkt des Rasters frei positioniert werden.

### 4. *Linear Interpolation*

Der Vektor-Modus erlaubt das Zeichnen einer geraden Linie zwischen zwei Punkten, die ähnlich wie im frei positionierbaren Modus beschrieben werden. Ein Vektor ist dabei definiert durch die Angabe eines neuen Endpunktes. Deshalb wird der erste Vektor nach Einschalten dieser Betriebsart immer dunkel geschrieben. Die Länge der Vektoren ist beliebig, doch wird im Interesse einer gleichmäßigen Helligkeit eine Maximallänge von 5 cm empfohlen.

Für den Anschluß des beschriebenen graphischen Terminals an einen Rechner sind im Prinzip folgende Möglichkeiten denkbar:

1. Paralleler Rechner-Anschluß im Nahbereich des Rechners. Diese Anschaltung ist serienmäßig nur für verschiedene PDP-8-Rechner mit Akkumulator-Schnittstelle lieferbar. Eine Anpassung an die Akkumulator-Nahtstelle der 301 dürfte leicht durchzuführen sein.
2. Direkter Anschluß an die interne Nahtstelle des Terminals.  
Diese Anschaltung kann zwar die meisten Möglichkeiten zur vollen Ausnutzung aller Eigenschaften und der maximalen Arbeitsgeschwindigkeit des Sichtgerätes bieten, ist jedoch für ein Einzelprojekt zu teuer wegen der entstehenden Entwicklungskosten.
3. Serieller Anschluß für Datenübertragungsnetze mit und ohne Modem für alle gebräuchlichen Geschwindigkeiten. Diese Anschaltung ist sehr universell, eine serielle Schnittstelle ist an jedem Rechner vorhanden, auch an den Rechnern des Systems 300. Es sind beliebige Entfernungen zugelassen, die Geschwindigkeit kann in Anpassung an die zur Verfügung stehenden Leitungen variiert werden. Aus diesen Gründen wurde im vorliegenden Falle der Anschluß an das Fernschreibelement FSK des Rechners Siemens 305 für das Terminal gewählt.

Die Firma Tektronix liefert für ihr Terminal T 4002 serielle Anschaltungen für Standard-Geschwindigkeiten von 110 bd bis zu 10 kbd. Diese Anschaltungen verwenden den ASCII-Code. Damit sind wir bei einem wunden Punkt der meisten Rechner des Systems 300 angekommen, denn sie erlauben wegen der 6-Bit-Struktur der Standard-Nahtstelle in den meisten Fällen nur den CCIT-Code.

An dieser Stelle sei noch einmal kurz die Arbeitsweise und der Aufbau des FSK wiederholt. Das Fernschreibelement besteht aus bis zu vier simultan arbeitenden Teilkanälen, die im Blockbetrieb Ein- und Ausgabe machen können. Dabei passiert eine automatische Umcodierung der Fernschreibzeichen über eine im Arbeitsspeicher stehende Codetabelle. Bei dieser Ausgabe wird jedes Zeichen des Interncodes (6 Bit) in ein beliebiges 6-Bit-Zeichen extern umcodiert und bei Eingabe geht dieser Weg umgekehrt. Dieses Verfahren funktioniert für den Code CCIT Nr. 2, der wegen seiner Doppeldeutigkeit von fünf auf sechs Bit ergänzt wird. Die Codetabelle übernimmt auch die Auswertung von Sonderfunktionen wie z. B. Blockende und Farbumschaltung.

Der ASCII-Code ist jedoch ein 7-Bit-Code, der auf acht Bit ergänzt wird (z. B. Parity).

Jedes Zeichen des Interncodes wird als relative Adresse in der 64 Worte langen Codetabelle interpretiert. Aus der Zeile der Tabelle, auf die diese Adresse zeigt, wird zuerst ein Steuerzeichen geholt und ausgewertet, dann wird aus der gleichen Zeile, nur verschoben, das Zeichen des Externcodes geholt und ausgegeben. Da im Steuerzeichen ein Bit unbenutzt bleibt, kann dieses als siebtes Code-Bit benutzt werden, zwischengespeichert werden und zusammen mit den sechs Bit des Externcodes und einem Parity-Bit ausgegeben werden. Damit ist das Problem der Ausgabe von 7-Bit-Zeichen halbwegs gelöst.

Für eine hierzu analoge Eingabe müßte man eigentlich die sieben Bit des Externcodes als relative Adresse in einer 128 Worte langen Codetabelle interpretieren und aus dieser Zelle die sechs Bit des Interncodes herausholen. Es existiert aber eine einfachere Möglichkeit der Umcodierung.

Vergleicht man einmal die Zeichenmenge, die sowohl im Interncode als auch im ASCII-Code existiert, so zeigt es sich, daß das sechste und das siebente Code-Bit des ASCII-Zeichens immer ungleich ist. Die Beschränkung auf die gemeinsame Zeichenmenge ist wohl einleuchtend, birgt aber noch andere Konsequenzen in sich. Die beiden Ausnahmen von dieser Ungleichheit bilden die Zeichen mit den Funktionen Wagenrücklauf und Zeilenvorschub. Diese Format-Steuerzeichen sind aber bei Eingabe in den Rechner zu unterdrücken, genau wie

beim CCIT-Code für den Bedienungsblattschreiber. Es genügt also für die Eingabe zusätzlich eine Logik zu bauen, die die Eingabe eines ASCII-Zeichens unterdrückt, wenn Bit 6 und Bit 7 beide gleich sind.

Das beschriebene Verfahren für die Ein- und Ausgabe von ASCII-Zeichen ist anwendbar auf das Lochstreifenelement und auf das Fernschreibelement und ist im Zentrallabor für Elektronik der KFA Jülich seit mehreren Jahren realisiert. Am FSK angeschlossen ist dabei eine Teletype ASR 33 mit 10 Zeichen pro Sekunde. An eine entsprechende Nahtstelle wurde auch das Tektronix-Terminal angeschlossen mit dem Ziel, alle Betriebsarten zu ermöglichen, die das Gerät bietet.

Die alphanumerische Ein-Ausgabe bereitet überhaupt keine Schwierigkeiten. Es sind höhere Geschwindigkeiten als bei einem Blattschreiber möglich; gewählt wurden 2 400 bd, das sind mehr als 200 Zeichen pro Sekunde.

Die graphischen Betriebsarten werden über Umschaltzeichen angewählt. Die entsprechenden Zeichen des Interncodes sind zwar nicht a priori vorhanden, jedoch lassen sich die Codes für Farbumschaltung und die Endezeichen für die Zeichenarithmetik der 3003 hierfür mißbrauchen. Nach dem Einschalten einer graphischen Betriebsart folgt die Ausgabe der Daten wiederum als Fernschreibzeichen. Beim Incremental Plot braucht man drei Bit für die acht verschiedenen Himmelsrichtungen und ein Bit für die Unterscheidung hell – dunkel. Das ergibt eine Liste von 16 Zeichen des Interncodes, die alle möglichen Inkrementvektoren beinhaltet. Eine höhere Zeichengeschwindigkeit und glattere Vektoren ohne Treppen bieten die frei positionierbaren Betriebsarten mit und ohne sichtbare Verbindung der Punkte. Die Form der Datenausgabe ist hierbei die gleiche. Benötigt werden 10-Bit-Koordinaten ohne Vorzeichen, das geht auch bei ASCII nicht als ein Fernschreibzeichen. Angewandt wird eine Aufteilung in  $2 \times 5$  Bit, die ergänzt auf sieben Bit als Fernschreibzeichen ausgegeben werden. Die zwei ergänzenden Bit dienen der Redundanz. Sie kennzeichnen, ob es sich um die höheren oder niederen fünf Bit des x- oder y-Wertes handelt. Dadurch wird es möglich, Kurzvektoren mit weniger als vier Fernschreibzeichen zu senden und bei Übertragungsfehlern, bei denen ein ganzes Zeichen verschluckt wird, kommt es dennoch nicht zu längeren Folgefehlern, weil nach spätestens vier Zeichen wieder Synchronismus in der Bedeutung der Teilkoordinaten besteht.

Wenn man auf die Möglichkeit der Kurzvektoren verzichtet, so kann man bei Ausgabe von nur fünf Bit Information die redundanten Bit durch einen Zähler erzeugen und kommt so wieder auf ein komplettes ASCII-Zeichen. Diesen Zähler kann man am Rechner oder im Terminal stationieren je nach den Forderungen an die Übertragungssicherheit.

Alle beschriebenen Betriebsarten ermöglichen lediglich die Ausgabe von graphischen Daten. Für ein interaktives Arbeiten mit dem Terminal muß aber auch die Eingabe von graphischen Daten in den Rechner möglich sein. Diesen Wunsch erfüllt bei dem vorliegenden Terminal ein Steuerknüppel, der ein Fadenkreuz auf den Bildschirm zeichnet, dessen Position zwei Analogspannungen entspricht, die von zwei im Steuerknüppel eingebauten Potentiometern verstellt werden kann. Dieses Fadenkreuz wird nicht auf dem Bildschirm gespeichert, sondern „durchgeschrieben“. Auf Knopfdruck oder auf Anfrage des Rechners werden die Koordinaten des Schnittpunktes des Kreuzes dem Rechner als Fernschreibzeichen übermittelt. Dabei erfolgt wiederum eine Aufteilung der Koordinate in  $4 \times 5$  Bit. Aus diesem Fernschreibzeichen kann die Position des Fadenkreuzschnittpunktes leicht ermittelt werden.

Zum Abschluß seien die Vor- und Nachteile der beschriebenen Anschaltung, die in dieser Form realisiert ist, erwähnt.

Der große Vorteil ist die Kompatibilität der alphanumerischen Betriebsart, die mit der Grundstellung des Terminals identisch ist. Alle Programme, die auf einen Blattschreiber ausgehen, bzw. die bedienbar sind, laufen auf Anhieb ohne irgendwelche Ergänzungen. Das beste Beispiel hierfür ist das Testhilfsprogramm PROTEST. Hier erlaubt das Tektronix-Terminal die Ausgabe längerer Befehls- und Datenfolgen in kürzester Zeit als auf einem

Blattschreiber. Die Kapazität des Bildschirms beträgt hierbei 155 Befehle, Dezimal- oder Gleitkommazahlen.

Für graphische Arbeiten sind mehrere längere oder kürzere Tabellen zum Umrechnen der graphischen Information notwendig, da auch die graphischen Daten über die Codetabelle gehen müssen.

Die Codetabelle platzt wegen der Beschränkung auf 64 Zeichen aus allen Nähten. Neben den bisher erwähnten Codes für die Betriebsartenumschaltung wären noch wünschenswert Steuerzeichen für die Umschaltung kursiv-gerade, für die zwei verschiedenen Buchstaben-Größen, für das Auslösen einer akustischen Klingel. Hier muß der Benutzer sich auf eine sinnvolle Auswahl beschränken. Implementiert sind in der jetzt vorliegenden Form neben der Umschaltung der Betriebsarten Zeichen für das Löschen des Bildschirminhaltes, für das Ansteuern der alphanumerischen Ausgangsposition (links oben) und für das Ein- und Ausschalten des Fadenkreuzes.

Trotz der erwähnten Nachteile ist das Tektronix Terminal T 4002 ein bewährtes Hilfsmittel zum Auswerten von kernphysikalischen Spektren mit Hilfe des Rechners Siemens 305 geworden.

**H. Prinz**

EDS-Systemtechnik, Aachen

Magnetbandgeräteanschluß

## Magnetbandgeräteanschluß

Vortrag am 27. 5. 71 im Hahn-Meitner-Institut, Berlin

### I. Aufgabenstellung

Im Auftrag von Siemens – Karlsruhe waren zwei Aufgaben zu lösen:

1. Entwicklung einer preiswerten Magnetbandanschaltung für 30 bzw. 60 kHz Bandgeräte.
2. Entwicklung der Software so, daß die vorhandenen Dienstprogramme benutzt werden können.

Es sollten zwei Programm-Ebenen möglich sein:

- a) physikalische Ebene
- b) Datei-Ebene

### II. Lösung der Hardware

Als Rechner Nahtstelle wurde das P3K gewählt, weil es für die ganze Serie der Siemens 300 Prozeßrechner Schnittstellen kompatibel ist.

Die Magnetband-Anschaltung wurde in modifizierter CAMAC-Technik ausgeführt, da diese Technik eine Reihe von Vorteilen aufweist:

- a) mechanische Einheit
- b) ausreichende Kartengröße; eine logische Elektronik-Einheit findet auf einer Karte Platz
- c) Die gesamte Anschaltung für maximal 4 Geräte, einschließlich Netzgerät, kann in einem Rahmen untergebracht werden.

Das führte zu folgendem Konzept: Abb. siehe umseitig.

Durch ein weitgehendes Bussystem konnte auf eine Background-Verdrahtung verzichtet werden.

Im einzelnen besteht die Anschaltung aus 5 verschiedenen Kartentypen:

- a) Befehlsdecoder:

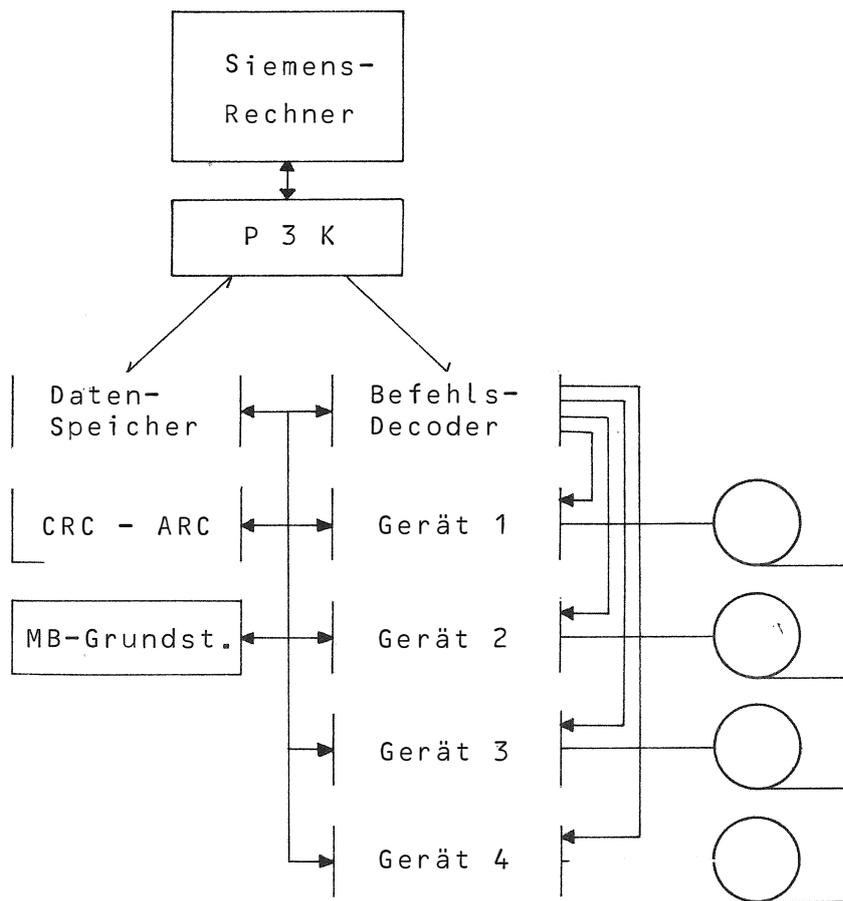
Hier wird der Befehl dekodiert und das MB-Gerät angewählt. Eine fehlende MB-Geräteanschaltung wird als Fehler (MB UNKL) zurückgemeldet.

- b) Datenumsetzer:

Hier werden die Daten, die vom Rechner kommen oder zum Rechner gehen, zwischengespeichert. Außerdem findet die Umsetzung von 24 Bit in 3 x 8 bzw. 3 x 8 in 24 Bit statt.

- c) Magnetbandgrundsteuerung:

Die Steuerung der zeitlichen Abläufe wird koordiniert, der Schreibtakt erzeugt, bzw. der Lesetakt verarbeitet. Außerdem wird der Datentransfer auf dem Bus koordiniert.



d) CRC-LRC-Karte:

Darstellung und Kontrolle von LRC (Longitudinal-Redundancie-Check) bzw. CRC (Cyclik-Redundancie-Check). Gegebenenfalls kann hier auch die Lese- und Korrigierlogik eingebaut werden.

CRC und LRC gewährleisten die IBM-Kompatibilität des beschriebenen Magnetbandes.

e) Geräteanschaltung:

Diese Karte enthält die Pegelumsetzer für die verschiedenen MB-Geräte-Typen AMPEX TMZ bzw. TM9 oder gegebenenfalls auch Geräte anderer Hersteller.

Die Steuerung des P3K und der Bandgeräte geschieht über EAW-, EVS-Befehl, wobei EVO und EV1 für den P3K (Adressen und Pufferlänge) und EV5 für das MB-Gerät gelten:

Der Aufbau des MB-Befehlswortes (EV5) wurde folgendermaßen gewählt:

MB - B	—	MB NR	—	EV5
1 4 5		12 13 14		18 24

Es stehen 16 Befehle zur Verfügung, davon werden jedoch nur 12 ausgenutzt:

1. Band rückspulen
2. Band rückspulen und entladen
3. Band um einen Block rücksetzen
4. Band um einen Block vorsetzen
5. Band um eine AM zurücksetzen
6. Band um eine AM vorsetzen
7. vergrößerte Blocklücke schreiben
8. AM schreiben
9. Block lesen
10. Block korrigiert lesen
11. Block schreiben
12. verkürzten Block schreiben.

### III. Lösung der Software

#### 1. Allgemeines

Da das P3K nicht softwarecompatibel mit der SKU-Steuerung ist, mußte ein neues Programm-Paket erstellt werden.

Die Aufrufe sollen in Verarbeitung und Rückmeldung mit den bestehenden austauschbar sein, um die Siemens Dienstprogramme verwenden zu können.

Im einzelnen heißt das, daß die Makros unverändert übernommen werden müssen.

Um keinen Eingriff ins ORG vornehmen zu müssen, wurde das Programmpaket an die neu-geschaffene EAP-Nahtstelle angeschlossen.

Die EAP-Nahtstelle (Ein-/Ausgabe-Programm) ist eine Möglichkeit, Anwender-Programme ins Betriebssystem einzuschleifen. Die Aufrufe werden in gewissem Sinn umgeleitet und durch ein Anwender-Programm verarbeitet.

Außerdem bietet das EAP folgende Möglichkeiten:

- a) Es kann im allgemeinen mit einem kleineren ORG gearbeitet werden. Nur bei Bedarf wird die MB-Software dazugeladen.
- b) Die MB-Software braucht trotzdem nur einmal vorhanden sein (Multiprogramming für bis zu 23 Benutzer).
- c) Es können logische Geräte-Nr. verwendet werden.

Die Allgemeinheit der EAP-Nahtstelle bringt aber auch Nachteile. Die gesamte Organisation, durch die im ORG die prioritätsgerechte Gerätebenutzung gesteuert wird, muß im EAP wiederholt werden.

#### 2. Magnetbandbenutzung aus physikalischer Ebene

Physikalische Ebene heißt:

Schreiben und Lesen von Daten, Schreiben von AM Bandbewegung (Block vor-rück, AM vor-rück, Rewind). Es können somit alle zur Magnetbandverarbeitung notwendigen Befehle angeführt werden.

Dieses Grundprogramm-Paket umfaßt 500 Kernspeicherplätze einschließlich aller Warteschlangen.

Um die Programmierung zu vereinfachen, wurde folgende Konvention getroffen:

Grundsätzlich muß ein MB-Gerät von einem Benutzer vor dem 1. Aufruf belegt werden. Das Gerät bleibt dann bis zum Freigabe-Aufruf für diesen Benutzer belegt.

Dies ist außerdem notwendig, um zu verhindern, daß mehrere Benutzer gleichzeitig ein und dasselbe Gerät benutzen und somit die Information in willkürlicher Reihenfolge auf dem Band steht.

Im einzelnen sind folgende Aufrufe realisiert:

- a) MA MBAU = adr, adr 1, adr 2  
von Speicherstelle adr 1 bis einschließlich adr 2 einen Block auf das durch adr bezeichnete Band schreiben
- b) MA MBAUE = adr, adr 1, adr 2  
von Speicherstelle adr 1 Bit 9 bis einschließlich adr 2 einen Block auf das durch adr bezeichnete Band schreiben
- c) MA MBEI = adr, adr 1, adr 2  
von Speicherstelle adr 1 bis einschließlich adr 2 einen Block von dem durch adr bezeichneten Band lesen
- d) MA MBAM = adr  
auf das durch adr bezeichnete Band eine Abschnittsmarke schreiben
- e) MA MBLO = adr  
auf dem durch adr bezeichneten Band ein Stück in Länge einer Blocklücke löschen
- f) MA MBVB = adr  
MA MBRB = adr  
MA MBVA = adr  
MA MBRA = adr  
das durch adr bezeichnete Band um einen Block (B) bzw. Abschnitt (A) vorsetzen (V) bzw. rücksetzen (R)
- g) MA MBRS = adr  
das durch adr bezeichnete Band zur Anfangsmarke zurückspulen
- h) MA MBRE = adr  
das durch adr bezeichnete Band zur Anfangsmarke zurückspulen und entladen. (Bei AMPEX Geräten nicht möglich! Daher hat der Befehl die gleiche Wirkung wie MA MBRS.)

Bei der Übertragung können folgende Fehler bzw. Anzeigen auftreten:

a) Schreib- oder Lesefehler

Abschnittsmarke gefunden, Störblock gelesen, lesen im geschützten Bereich versucht, gelesener Block zu lang, Parityfehler, Parityfehler lorryierbar, Bandmarke gefunden, Datenübertragungsfehler.

b) Gerätefehler

Gerät unklar (kein Gerät mit der angegebenen Nr. vorhanden), falscher Befehl, kein Schreibring.

Bei der Fehlerbehandlung wird unterschieden zwischen automatischer und benutzergesteuerter Fehlerbehandlung.

a) automatische Fehlerbehandlung

Bei Parityfehler wird fünfmal der Lese- bzw. Schreibvorgang wiederholt. Bei Datenübertragungsfehler wird solange wiederholt, bis der Block korrekt übertragen wurde. Bandmarken und Abschnittsmarken werden dem Benutzer direkt übergeben. Fehlender Schreibring und Gerät unklar wird dem Operateur mitgeteilt und auf Quittung gewartet.

b) benutzergesteuerte Fehlerbehandlung

Durch Ausblenden einzelner Fehlerbits (Maskenbits) kann die Fehlerbehandlung vom Benutzer durchgeführt werden. Dies ist besonders wichtig bei On-Line-Datenerfassung, bei der es unter Umständen auf die rasche Speicherung von Informationen ankommt.

### 3. Benutzung auf Dateiebene

#### Vorbemerkung

Grundsätzlich sind die auf physikalischer Ebene beschriebenen Bänder von allen Rechnern mit 9 Spur-Bandgeräten lesbar, da sie der IBM-Norm entsprechen.

Diese Norm beinhaltet:

- a) Lage der Bits auf dem Band
- b) generieren und aufschreiben von CRC und LRC
- c) Länge der Gaps
- d) Abschnittsmarke

Die Dateiverarbeitung bringt lediglich eine Vereinfachung der Datensicherung und vereinfachte Kompatibilität unter den Maschinen. Dies bedingt aber auch einen zusätzlichen Programmaufwand (zusätzlich 1 K).

#### Grundlagen der Dateiorganisation

Ein im Datei-Format beschriebenes Band enthält mindestens ein Spulenetikett. Dieses Etikett wird über ein besonderes Dienstprogramm aufgeschrieben.

Auf das Spulenetikett können dann mehrere Daten folgen.



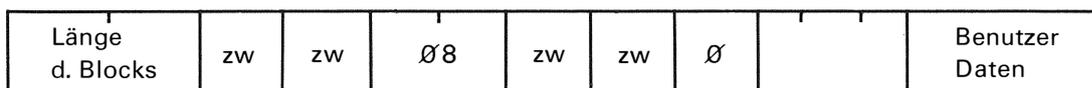
Eine Datei ist gekennzeichnet durch ein Dateianfangsetikett, Datenblöcke und einem Dateiendeetikett.

Das Ende des verwertbaren Teils der Bandaufzeichnung ist durch eine doppelte Abschnittsmarke gekennzeichnet.

Eine Datei innerhalb einer Gruppe kann nicht geändert werden, es sei denn, die restlichen Daten dürfen gelöscht werden.

Im einzelnen sehen die Datei-Teile folgendermaßen aus:

- a) Das Standardspulen-Etikett dient zur Kennzeichnung der Spulen-Nr. und des Eigentümers. Bandanfangsetiketten werden vom Organisationsprogramm gelesen aber nicht geschrieben.
- b) Das Standarddateianfangsetikett dient zur Kennzeichnung des Anfangs einer Datei. Es enthält den Namen und die Datei-Nr. Außerdem Schreib- und Freigabedatum (gegebenenfalls Schreibsperr).
- c) Das Standarddateiendeetikett markiert das Ende einer Datei. Es enthält die gleiche Information wie das Dateianfangsetikett, jedoch zusätzlich die Anzahl der Blöcke.
- d) Das Standardspuleneendeetikett wird nicht geschrieben. Wird es gelesen, so wird es wie eine Abschnittsmarke behandelt.
- e) Ein Datenblock besteht formal aus zwei Sätzen. Eine darüber hinausgehende Blockung und Satzeinteilung muß vom Benutzer vorgenommen werden. Der grundsätzliche Aufbau ist folgendermaßen:



Die ersten 4 Worte werden automatisch erzeugt. Der Benutzer muß jedoch den Platz dafür vor seinen Daten reservieren.

Die einzelnen Etiketten werden anhand der GEDA-Angaben aufgebaut und gegebenenfalls automatisch aufs Band geschrieben.

Es sind folgende Aufrufe vorgesehen:

- a) MA DAER = adr  
Banddatei eröffnen. adr verweist auf die 1. Zelle des GEDA-Feldes.
- b) MA DASL = adr, mod.  
Datei schließen. adr verweist auf die 1. Zelle des GEDA-Feldes MOD bestimmt die Art des Dateiendes:  
MOD = 0 Vorsetzen bis zur nächsten Datei  
MOD = 1 Rücksetzen auf den Anfang der Datei  
MOD = 2 Rückspulen zum Bandanfang

- c) NA STEI, = gedaadr, adr 1, adr 2  
STAU Standard Ein-/Ausgabe von Daten auf das Band.
- Grundsätzlich gilt auch hier, daß vor dem 1. Aufruf das MB-Gerät belegt werden muß.

**L. Gries**

Siemens AG Karlsruhe

**W. Woletz**

Hahn-Meitner-Institut Berlin

Die Kopplung von Rechnern des Systems 300

## Die Kopplung von Rechnern des Systems 300

Kleine On-line-computer, wie z. B. S 301, verfügen im allgemeinen über keine große Peripherie und schnelle externe Speicher. Aus diesem Grunde wurde in Zusammenarbeit mit der Fa. Siemens ein Softwarepaket für das HMI entwickelt, welches die Kopplung zwischen Rechnern des Systems 300 ermöglicht. Im folgenden wird am Beispiel der Kopplung zwischen einer S 305 und einer S 301 näher auf dieses System eingegangen.

Die Kopplung erfolgt über die Prozeßelemente P4KS (S 305) und eine P3KS (S 301). Die Übertragung erfolgt wortweise, d. h. 24-bit parallel und ist voll-duplex. Die Übertragungstrecke kann maximal 30 m betragen. Soll über größere Entfernungen übertragen werden, muß auf beiden Seiten zusätzlich jeweils eine P3UK benutzt werden.

Sinn und Aufgabe der Kopplung ist es, dem On-line-Rechner – in unserem Falle die S 301 – Zugang zu der umfangreichen Peripherie der S 305 zu verschaffen und die größere Rechenkapazität der 305 (Gleitkomma-Arithmetik, keine Simulation) auszunutzen; besonders um große Datenmengen, die in den Experimenten anfallen, schnell auf große Externspeicher zu schaffen. Um diese Aufgaben erfüllen zu können, wurden innerhalb des Koppelsystems vier Funktionen festgelegt:

### 1. Plattenverkehr (Codewort FPSK)

Die S 301 kann die Platte der S 305 benutzen. Sämtliche Plattenfunktionen wie Datei eröffnen, lesen, schreiben usw. sind möglich.

### 2. Programme in der S 305 aktivieren (FAWE)

Die S 301 kann in der S 305 Programme von der Platte (S 305) bereitstellen und starten, die dann unabhängig vom Koppelsystem ablaufen können.

### 3. Bereitstellen (FBRS)

Programme, die auf der Platte der S 305 stehen, werden im ASP der S 301 bereitgestellt.

### 4. Datenaustausch (FDAS)

Die S 305 kann aus ihrem Arbeitsspeicher Daten direkt in den ASP der 301 senden, die dort erwartet werden.

Sollen periphere Geräte der S 305 von der S 301 angesprochen werden, so muß dies mit der Funktion ‚Programm in S 305 aktivieren‘ geschehen. Dies bedeutet, daß zuvor die auszugebenden Daten mit der Funktion FPSK auf die Platte der S 305 geschrieben werden müssen. Der Grund für diesen Umweg liegt im begrenzten Speicherplatz der S 305 bzw. in dem Wunsch der schnellen Freigabe der Koppelstrecke für weitere Datentransfers.

Die vier vorher beschriebenen Funktionen können per Makro-Aufruf von den Anwenderprogrammen angesprochen werden. Diese Makros haben die gleiche Struktur wie die Standard-Eingabe/Ausgabe-Aufrufe des E/A-ORG's und unterteilen sich in zwei Arten:

### 1. Funktionsaufrufe

Sie bewirken den Unterprogrammprung in das Koppelprogramm und damit den Anstoß des Koppelverkehrs. Als Parameter enthalten sie die Adresse des Parameterblockes.

## 2. Beschreibungsaufrufe

Sie definieren die Parameter.

Die Beschreibungsaufrufe unterteilen sich in zwei Gruppen:

- a) Adressparameterblock
- b) Koppelparameterblock

Der Adressparameterblock beinhaltet die Anzeigenzelle, die Adresse des Koppelparameterblocks und den Eingabe/Ausgabe-Bereich im Arbeitsspeicher.

Der Koppelparameterblock enthält die Information, welche Funktion angesprochen werden soll und die speziell erforderlichen Parameter für die Funktion.

Die Funktion FAWE (Aktivieren von Programmen in der S 305) kann ein Daten-senden (FDAS) von der S 305 zur S 301 zur Folge haben. Da dies zu einem beliebigen Zeitpunkt erfolgen kann, existiert ein weiterer Funktionsaufruf (DKWA), der es ermöglicht, auf diese speziellen Daten zu warten, um die Synchronisation von Programmen zu ermöglichen. Sind die Daten noch nicht eingetroffen, wird das aufrufende Programm an der Stelle des Warteaufrufs unterbrochen und erst nach Eintreffen der Daten fortgesetzt.

Das Koppelsystem besteht im wesentlichen aus drei Teilen:

### 1. Koppelhauptprogramm (KP)

Es benötigt eine eigene Programmnummer und besteht im wesentlichen aus einem Eingabe-Aufruf für das Prozeßelement (P4KS bzw. P3KS).

### 2. Koppelunterprogramm (UDKP)

Es hat die Aufgabe, die Prioritätswarteschlange aufzubauen. Dies ist notwendig, da neben dem Koppelprogramm andere Programme simultan ablaufen können und ihrerseits Koppelaufrufe abgeben können. Eine weitere Aufgabe ist die Übergabe eventueller Anzeigen an die Anwenderprogramme.

### 3. Funktionsbausteine (UF . .)

Sie realisieren die einzelnen Funktionen und führen den Koppelnahstellenverkehr durch.

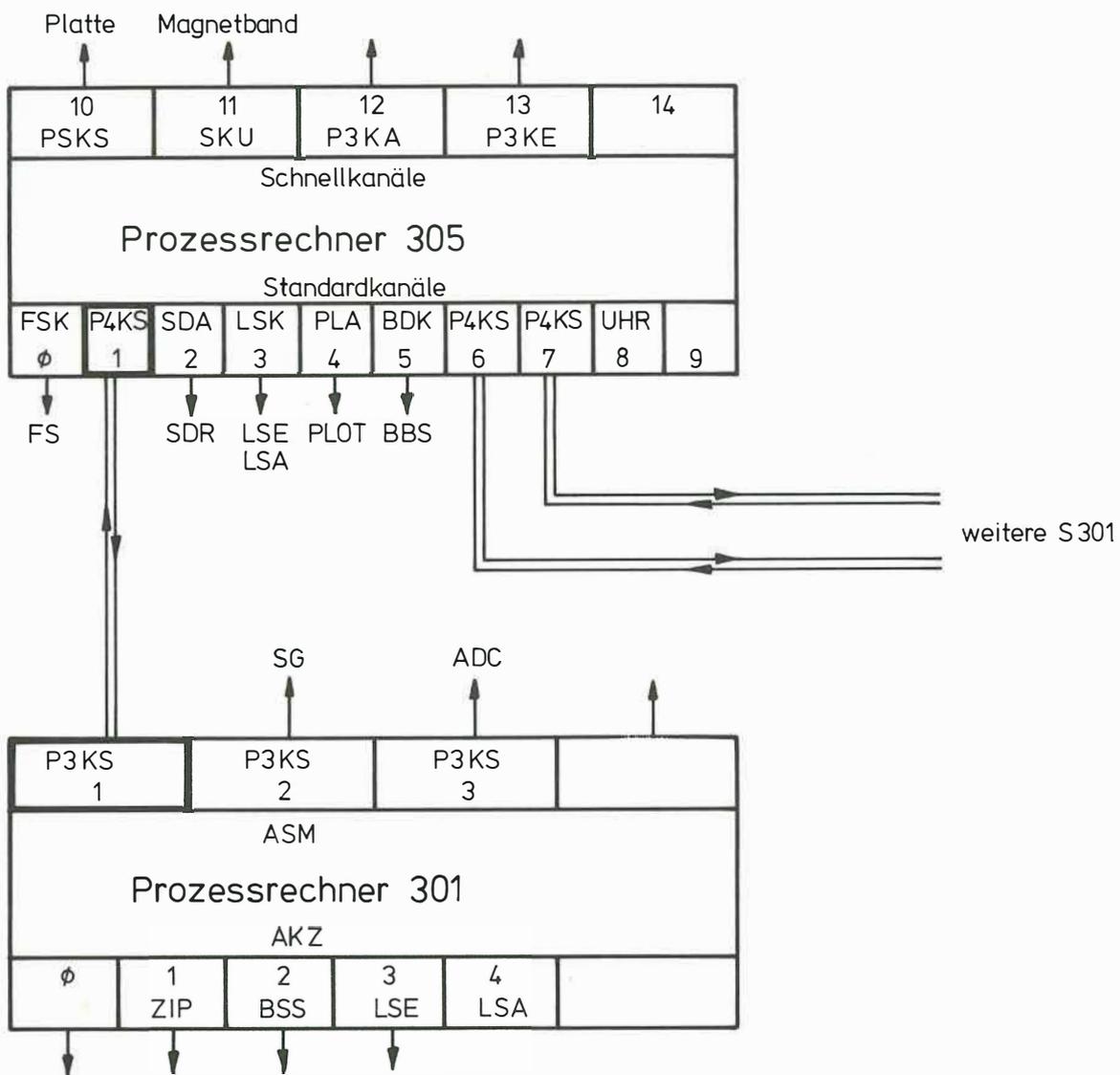
Die Reaktion auf einen Koppelaufruf soll im folgenden kurz erläutert werden:

Bevor die Anwenderprogramme gestartet werden, wird auf beiden Rechnerseiten das Koppelprogramm gestartet. Nach dem Start geht das KP auf beiden Seiten auf Eingabe und wartet auf Anmeldungen der Gegenseite. Das System ist nun bereit, von beiden Seiten Koppelaufrufe entgegenzunehmen.

Wird von einem Anwenderprogramm ein Koppelaufruf abgegeben, so erfolgt ein Unterprogrammssprung in das UDKP; dieses sendet eine Anmeldung zur Gegenseite und leitet damit den Koppelverkehr ein. Gleichzeitig teilt es seinem Koppelhauptprogramm (KP) die Abgabe der Anmeldung mit. Diese Anmeldung gelangt auf der Gegenseite in einen dafür vorgesehenen Puffer des KP's (die Gegenseite stand auf Eingabe). Das KP der Gegenseite erkennt an Hand einer Kennung die Anmeldung und stößt sein UDKP an, außerdem sperrt es seine eigene Eingabe. Nun übernimmt das UDKP bzw. der betreffende Funktionsbaustein die weiteren Aufgaben. Es wird eine Quittung zu der Seite, die den Aufruf erzeugte, gesendet. Diese Quittung gelangt in den Puffer des KP's, welches auf dieser Seite noch immer auf Eingabe steht, aber bereits die Information erhalten hat, daß es eine Quittung zu erwarten hat. Das KP prüft an Hand der Kennung, ob es sich wirklich um eine Quittung handelt. Ist dies der Fall, so sperrt auch auf dieser Seite das KP seine Eingabe, und den weiteren Verkehr übernimmt der angesprochene Funktionsbaustein. Stellt das KP jedoch fest, daß es sich anstatt einer Quittung um eine Anmeldung handelt, so hatten beide Seiten gleichzeitig eine Anmeldung getätigt, und den beiden Anwenderprogrammen wird dieser Konfliktfall mitgeteilt. Im Normalfall jedoch wird der Funktionsbaustein den Koppelverkehr mit Quittungen und Daten-senden bzw. empfangen zu Ende führen. Nach Durchführung eines Aufrufes wird auf beiden Seiten die Eingabe des KP's freigegeben, und die eventuell gefüllte Koppelwarteschlange wird weiter abgearbeitet.

Anmeldung und Quittung bestehen aus einem Block von zehn Festworten. Dieser Block besteht im wesentlichen aus dem Koppelparameterblock, der Kennung, einer internen Anzeige und der Transferlänge für den nachfolgenden Datenverkehr.

Das Koppelsystem ist völlig symmetrisch aufgebaut, und beide Rechner sind völlig gleichberechtigt. Aus diesem Grunde können beide Seiten zu einem beliebigen Zeitpunkt Anmeldungen senden und entgegennehmen und so den Koppelverkehr einleiten. Die Kopplung mehrerer S 301 an eine S 305 ist vorgesehen, und mit der Erweiterung der Kopplungssoftware wurde begonnen.



## Aufbau der Rechnerkopplung

**Dr. Hp. Englmeier**

Abteilung für Röntgenstrukturforschung am Max-Planck-Institut München

Datenaustauschsteuerung zwischen einer Siemens 305 und einer IBM 1130 Rechenmaschine

## **Datenaustauschsteuerung zwischen einer Siemens 305 und einer IBM 1130 Rechenmaschine**

Die IBM 1130 Rechenmaschine ist ein Satellit der Großrechenmaschine IBM 360/91 in München-Garching. Die Siemens 305 Rechenmaschine dient vor allem zur Datensammlung (on line und off line). Da die über die Siemens 305 gesammelten Daten fast ausschließlich auf der IBM 360/91 weiterverarbeitet werden, erschien es günstig, zwischen der Siemens 305 und der IBM 1130 eine Verbindung herzustellen.

Die Datenaustauschsteuerung (Abb. 1) setzt sich zusammen aus einer gemeinsamen Datentransportsteuerung und aus zwei der jeweiligen Rechenmaschine zugeordneten Steuerungen. Letztere bestehen aus der Speichersteuerung mit Adressregister für den Datentransport, dem Externbefehlsdekoder zur Einleitung des Datentransportes, und aus der Interruptsteuerung, die je nach Modifikation der betreffenden Rechenmaschine der Speichersteuerung (305) oder dem Externbefehlsdekoder (1130) angegliedert ist. Die gemeinsame Datentransportsteuerung ist aufgebaut aus dem Datentransportregister (für programmierbaren und cycle stealing Datenaustausch), einem Blocklängenzähler (für cycle stealing), und der eigentlichen Kontrolleinheit.

Die zu übermittelnden Informationen teilen sich in per Programm zu übertragende Codeworte – zum Einleiten und Beenden eines Datentransportes – und in die von Kernspeicher zu Kernspeicher zu transportierenden Datenworte (Abb.2). Bei den Datenworten wird ein 24 bit Wort der 305 auf zwei 16 bit Worte der 1130 aufgeteilt, wobei von einem Wort der 1130 4 bit ungenutzt bleiben. Bei den Codeworten werden von 18 bit (= 3 Teilworte) der 305 16 bit auf beiden Seiten ausgenutzt. Es gibt 4 verschiedene Codeworte. Codewort 0 dient zur An- und Abmeldung von Übertragungsprozeduren. Mit diesem Codewort wird signalisiert, welcher Art die Daten sind, woher sie geholt werden müssen oder was mit ihnen zu geschehen hat, bzw. welche direkte Funktion sie genauer modifizieren. Die Codeworte 1 und 2 dienen zur Einleitung einer einzelnen Datenübertragung (Blocklänge, Übertragungsrichtung) und zum Beenden dieser Datenübertragung (Longitudinalkontrolle). Codewort 3 kommt von der Datenaustauschsteuerung, es meldet Fehler oder das Ende einer einzelnen Datenübertragung.

Der Befehlsvorrat der Externbefehlsdekoder ist bei beiden Rechenmaschinen gleich groß. Fehlt bei einem Externbefehlsdekoder ein Befehl, so ist dieser per Hardware nachgebildet – entsprechend den speziellen Eigenschaften der betreffenden Rechenmaschine (Abb.3). Beide Rechenmaschinen können die Datenaustauschsteuerung belegen bzw. freigeben, und Codeworte an die andere Rechenmaschine senden bzw. von dieser empfangen. Beide Maschinen können eine Datenübertragung auslösen bzw. dazu das zu ihr gehörige Adressregister laden, wenn die andere Maschine eine Datenübertragung auslösen will.

Als Beispiel sei der Ablauf einer Datenübertragung angegeben, bei der die Siemens 305 als aktive Maschine fungiert (Abb.4). Die 305 belegt den Kanal und sendet das Codewort 0 an die 1130. Ist der Übertragungsvorgang möglich oder ermöglicht worden, sendet die

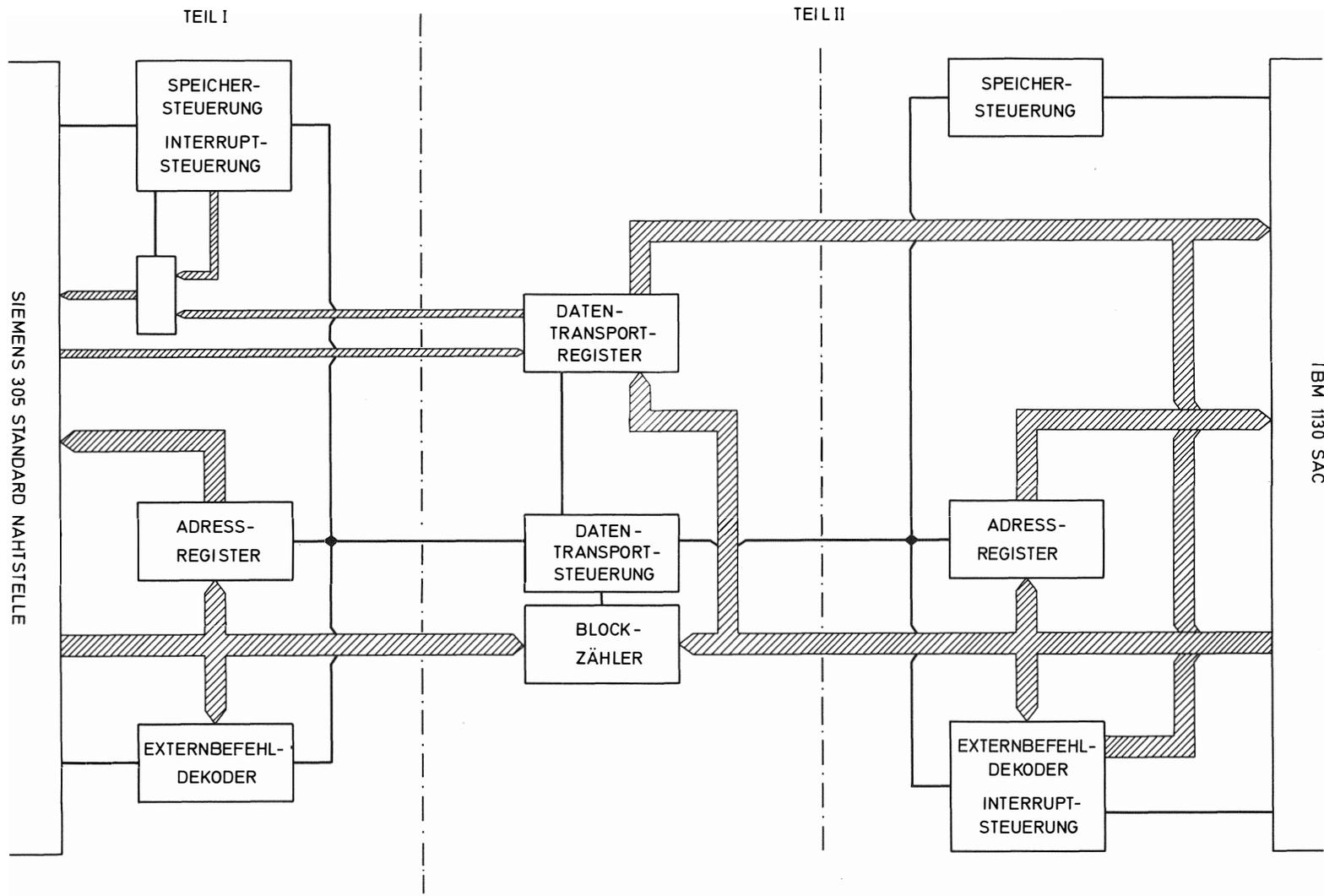
1130 als positive Quittung das Codewort 0 unverändert zurück. Daraufhin sendet die 305 das die eigentliche Übertragung kennzeichnende Codewort 1. Bei positiver Quittung durch die 1130, die dabei vorher ihr Adressregister geladen hat, setzt die 305 ebenfalls ihr Adressregister, anschließend den Blocklängenzähler und startet die Datenübertragung. Am Ende der Datenübertragung wird das Codewort 2 für die Longitudinalkontrolle ausgetauscht und, wenn die Übertragung nach dieser Einzelübertragung abgeschlossen ist, auch das Codewort 0 zur Abmeldung. Bei den Darstellungen der Zustände der Rechenmaschinen (Abb. 4, links und rechts) bedeutet: einfach schraffiert: irgendwelche Arbeiten des Rechners; doppelt schraffiert: Arbeiten für die Übertragung. Bei der Darstellung des Kanalzustandes (Abb. 4, mitte) bedeuten: nicht schraffiert: Kanal frei; einfach schraffiert: Kanal belegt; schwarz: Kanal arbeitet für programmierbare Datenübertragung, oder bildet Externbefehle nach; doppelt schraffiert: reine cycle stealing Datenübertragung. Ein zeitlicher Maßstab kann der Darstellung nicht entnommen werden. Das Beispiel entspricht der Auslösung einer direkten Funktion, da bei einer realen Datenübertragung zwischen An- und Abmeldung viele Einzelübertragungen stattfinden.

Die Datenaustauschsteuerung wurde 1967 entworfen und gebaut. 1968 wurde sie – damals noch mit einer Siemens 303, die 1969 gegen eine 305 ausgetauscht wurde – in Betrieb genommen. Sie diente am Anfang vor allem zur Übertragung der ausgewerteten Meßdaten von off line Diffraktometern (über die Lochstreifeneingabe der 303 bzw. 305). Später kamen Daten vom on line Diffraktometer [1], vom on line Photometer [2] und auf Band aufgezeichnete Daten hinzu. In letzter Zeit wurde auch ein PROSA Übersetzer an der Kombination 305-1130 in Betrieb genommen (Kartenlesen über 1442 Protokollausgabe auf 1403 – Abb. 5).

Zum Schluß noch ein Blick auf die gesamte Anlage (Bild 5). Links die Siemens 305, mit Bedienungsblattschreiber, Lochkarteneingabe, Lochstreifeneingabe, Diffraktometer mit zusätzlichem Bedienungsblattschreiber, Photometer (ein Parallelprozessor, der den Kernspeicher des Zentralprozessors – 305 – mitbenutzt) und Magnetbandgerät. Rechts die IBM 1130, mit Konsolschreiber, Plattenspeicher, Schnelldrucker, Karteneingabe, Plotter und Modem zur Datenübertragung an die IBM 360/91.

#### *Schrifttum*

1. HP. Englmeier und W. Hoppe: Automation der Datensammlung in der Röntgenstrukturanalyse – messtechnik Heft 11 (1969) S. 265–271.
2. HP. Englmeier: Anschluß von Meßgeräten an eine Rechenmaschine über ein externes Leitwerk – messtechnik Heft 7 (1971) S. 145–151.

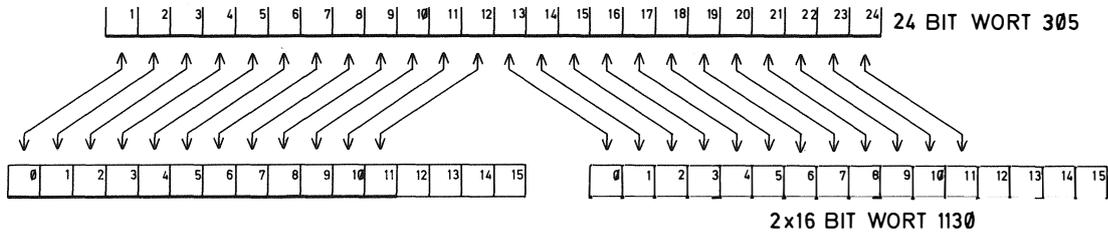


DATENAUSTAUSCHSTEUERUNG ZWISCHEN EINER SIEMENS 305 RECHENMASCHINE UND EINER IBM 1130 RECHENMASCHINE

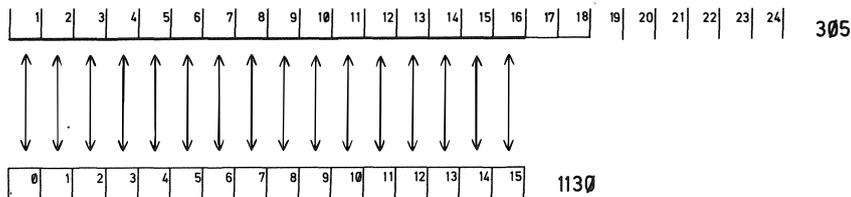
Abb. 1

BIT  BENUTZT  UNBENUTZT  NICHT UEBERTRAGEN

DATENWOERTER



CODEWOERTER



BEDEUTUNG DER CODEWOERTER - 4 TYPEN - KENNZEICHEN BIT 0 UND 1

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	UEBERTRAGUNG AN- ODER ABMELDEN BZW. AN- ODER ABMELDUNG BESTAETIGEN
0	0	R	A	KENNUNGSNUMMER												

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	EINZELUEBERTRAGUNG ANMELDEN BZW. ANMELDUNG BESTAETIGEN
0	1	R	LAENGE IN 12 BIT WORTEN													

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	LONGITUDINALKONTROLLE SENDEN BZW. BESTAETIGEN
1	0	R	LONGITUDINALKONTROLLE													

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	MELDUNG DER DATENAUSTAUSCH- STEUERUNG
1	1	Z	E													

R = RICHTUNG - "1" = 305 → 1130, "0" = 1130 → 305  
A = ANMELDEN - ABMELDEN - "1" = ANMELDEN, "0" = ABMELDEN  
Z = ZUSTAND DER ANDEREN RECHENMASCHINE  
E = FEHLERMELDUNGEN

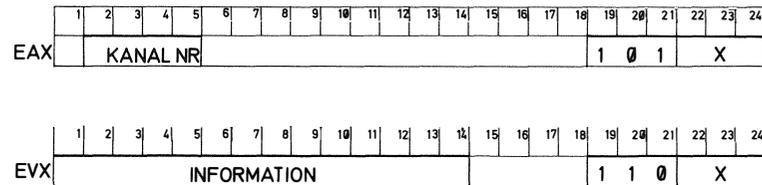
INFORMATIONSDARSTELLUNG EINER DATENAUSTAUSCHSTEUERUNG ZWISCHEN EINER  
SIEMENS 305 RECHENMASCHINE UND EINER IBM 1130 RECHENMASCHINE

Abb. 2

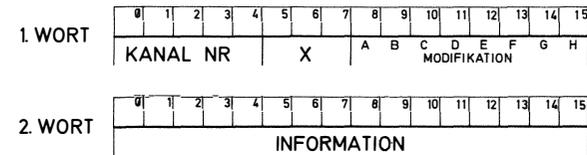
	SIEMENS 305		IBM 1130	
	BEFEHL	INFORMATION	BEFEHL	INFORMATION
KANAL TESTEN UND BELEGEN WENN FREI	EA 0		XI07	
KANAL FREIGEBEN	EA6		XI04-C	
KODEWORT SENDEN UND ANDERE RECHENMASCHINE UNTERBRECHEN	EA2	CODEWORT WIRD AUS ANZEIGENZELLE GEGENDET	XI04-A	CODEWORT
INTERRUPT ABFRAGEN	*	INTERRUPTKENNUNG WIRD IN PU-ZELLE GESCHRIEBEN	XI03	
KODEWORT ANNEHMEN	*	CODEWORT WIRD IN ANZEIGENZELLE GESCHRIEBEN	XI07-H	
ADRESSREGISTER FUER DATENUEBERTRAGUNG LADEN	EA3		XI04-B	ADRESSE
	EV1	ADRESSE		
DATENUEBERTRAGUNG 1130→305 AUSLÖESEN	EA4		XI05	ADRESSE
	EV1	ADRESSE	*	
	EV2	BLOCKLAENGE		BLOCKLAENGE IM ERSTEN WORT DES BLOCKS
DATENUEBERTRAGUNG 305→1130 AUSLÖESEN	EA5		XI06	ADRESSE
	EV1	ADRESSE	*	
	EV2	BLOCKLAENGE		

\* HARDWARELÖSUNG

SIEMENS 305 EA UND EV BEFEHLE



IBM 1130 PARAMETER EINES XI0X BEFEHLS



BEFEHLSDARSTELLUNG EINER DATENAUSTAUSCHSTEUERUNG ZWISCHEN EINER SIEMENS 305 RECHENMASCHINE UND EINER IBM 1130 RECHENMASCHINE

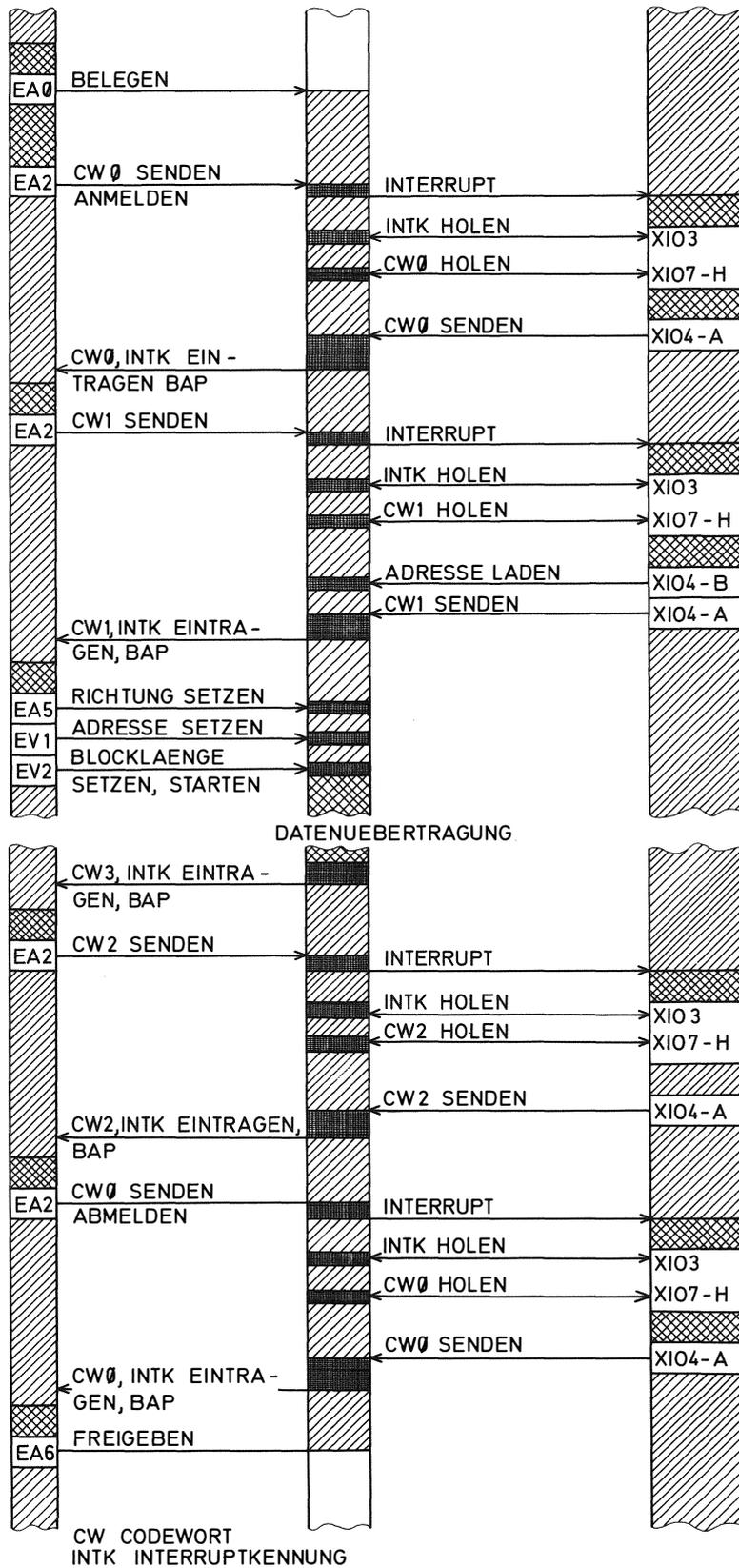
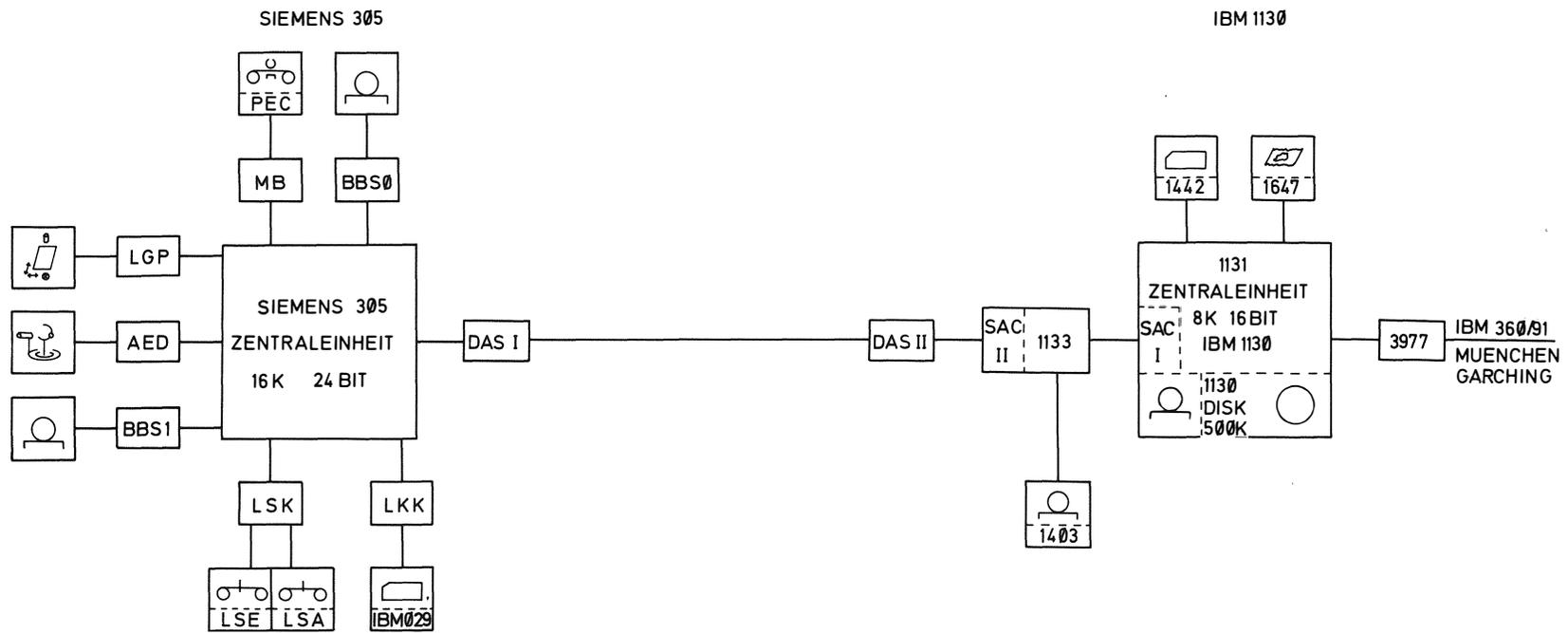


Abb. 4



BBSX BEDIENUNGSBLATTSCHREIBER X  
 PEC PERIPHERAL EQUIPMENT CORPORATION  
 MB MAGNETBANDSTEUERUNG  
 LGP LEITWERKGESTEUERTES PHOTOMETER  
 AED AUTOMATISCHES EINKRISTALL DIFFRAKTOMETER  
 LSK LOCHSTREIFENSTEUERUNG  
 LSE LOCHSTREIFENEINGABE  
 LSA LOCHSTREIFENAUSGABE  
 LKK LOCHKARTENSTEUERUNG

DAS I DATENAUSTAUSCHSTEUERUNG TEIL1  
 DAS II DATENAUSTAUSCHSTEUERUNG TEIL2

SAC STORAGE ACCESS CHANNEL  
 3977 IBM MODEM  
 1647 IBM PLOTTER  
 1442 IBM CARD READER PUNCH  
 1133 IBM CONTROL UNIT  
 1403 IBM PRINTER

DATENVERARBEITUNGSANLAGE  
 MAX PLANCK INSTITUT FUER EIWEISS- UND LEDERFORSCHUNG  
 ABTEILUNG ROENTGENSTRUKTURFORSCHUNG  
 MUENCHEN

Abb. 5

**Dr. H. Mittendorf**

Siemens AG Karlsruhe

FORTRAN IV H-Level für das Siemens-System 300

## FORTRAN IV H-Level für das Siemens-System 300

*Die große Anzahl der verschiedenen Aufgaben, die heute von Prozeßrechnern gelöst werden müssen, macht in zunehmendem Maße die Verwendung höherer Programmiersprachen erforderlich. Hierbei nimmt die Sprache FORTRAN zur Zeit eine dominierende Stellung ein; für die Aufgaben der Prozeßprogrammierung sind überdies zusätzliche Sprachelemente vorgeschlagen, die unter dem Sammelbegriff Prozeß-FORTRAN in die Literatur eingegangen sind. Dem Zuge dieser Entwicklung folgend entwickelt unser Haus daher einen neuen FORTRAN-Compiler, der diesen neuen Anforderungen Rechnung trägt.*

Compiler für die älteren Maschinen des 300-Systems mußten noch mit einer minimalen Anlagenausstattung (z. B. nur LSK als Ein-Ausgabegerät, kein Externspeicher) arbeiten können. Sinnvoll war dies nur möglich, wenn man im Falle von FORTRAN auch den Sprachumfang beschränkte; daher wurde zunächst der E-Level zugrunde gelegt. Erst mit den erweiterten Mitteln der DVA 306 war es gerechtfertigt, den vollen H-Level zu implementieren.

### 1. Eigenschaften der Sprache

Mit FORTRAN 306 ist der Sprachumfang von FORTRAN bezeichnet, den der zur Zeit in Arbeit befindliche Übersetzer für das SIEMENS SYSTEM 300 bietet. Er ist zu unterscheiden von der bisher unter dem Namen FORTRAN 300 verfügbaren Version. Ungeachtet dieser (vorläufigen) Bezeichnung können jedoch sowohl der Übersetzer als auch die von ihm erzeugten Programme auf den Anlagen 301 bis 305 ablaufen, soweit Anlagenausstattung, Speicherplatz und verdrahteter oder simulierter Befehlsvorrat hierfür ausreichen.

FORTRAN 306 umfaßt den vollen Sprachumfang, wie er von der American Standards Association (ASA) genormt wurde. Darüber hinaus sind erlaubt:

- Implizite Spezifikationsanweisung (IMPLICIT)
- Dateninitialisierung bei expliziter Spezifikationsanweisung
- arithmetische Ausdrücke gemischten Typs
- Längenangabe von Variablen mit \*2, \*4, \*8 und \*16
- Komplexe Rechnung mit doppelter Genauigkeit (COMPLEX\*16)
- Felder mit mehr als 3 (7) Dimensionen
- T-Format
- NAMelist-Anweisung
- Apostroph als Begrenzungszeichen von Literalkonstanten ENTRY-ANWEISUNG bei Unterprogrammen

FORTRAN 306 wurde, soweit möglich, an das FORTRAN des SIEMENS SYSTEM 4004 angelehnt. Da es sich bei allen Anlagen dieses Systems um Byte-Rechner handelt, beim SYSTEM 300 dagegen um Wort-Rechner mit 24 bit-Wortlänge, ergeben sich einige Abweichungen, die im folgenden beschrieben werden sollen.

### 1.1 Zeichenvorrat

Der Zeichenvorrat basiert auf dem 6-bit-Code des SYSTEM 300 und ist von dem 8-bit-Code der Rechner des SYSTEM 4004 verschieden. Er ist jedoch identisch mit dem der IBM 7090 und 7094 und kann daher von einer SIEMENS 4004-Anlage mit dem sog. 7090/7094-Zusatz verarbeitet werden. Umgekehrt kann der 306-Übersetzer auch Programme übersetzen, die im Lochkartencode für das System 4004 vorliegen, sofern sie folgenden Zeichenvorrat nicht überschreiten:

Buchstaben     A – Z  
 Ziffern         0 – 9  
 Sonderzeichen   = + - \* / . , ( ) '

Alle anderen Zeichen des Interncodes dürfen nur in Literalen vorkommen. \$ ist ein Sonderzeichen, & ist nicht erlaubt.

### 1.2 Datenstruktur

Zwischen den Bytes des SYSTEM 4004 und den Festworten des SYSTEM 300 wird folgende Zuordnung festgelegt:

2 Byte = 1 FW

Damit ergeben sich folgende Datentypen:

Variablen- typ	SYSTEM 4004 Länge in Bytes	SYSTEM 300 Länge in FW	Bemerkung
INTEGER	2	1	
INTEGER	4	2	Standardlänge
REAL	4	2	Standardlänge
REAL	8	4	
COMPLEX	8	4	Standardlänge
COMPLEX	16	8	
LOGICAL	1	1	= LOGICAL*2
LOGICAL	4	2	Standardlänge

Bei Literalen werden 4 Zeichen je Wort gespeichert.

Zu beachten ist die unterschiedliche Behandlung des Typs LOGICAL\*1, da Halbworte nicht adressierbar sind.

### 1.3 Zahlendarstellung und Wertebereiche

Die Gleichsetzung von 2 Byte mit einem Festwort ergibt nur eine formale Übereinstimmung. Es besteht jedoch eine Abweichung im Zahlenbereich der Variablen:

INTEGER*2 I	$-2^{23} \leq I \leq 2^{23}-1$	$2^{23} = 8\,388\,608$
INTEGER*4 ID	$-2^{46} \leq ID \leq 2^{46}-1$	$2^{46} = 70\,368\,744\,177\,664$
REAL*4 R	} .5E-308 Z 1.0E308	Auflösung 1.2E-7
REAL*8 RD		bzw. 1.16E-10 (306)
		Auflösung 1.4E-14
COMPLEX*8	entspricht 2 REAL*4-Größen	
COMPLEX*16	entspricht 2 REAL*8-Größen	

## 2. Beschreibung und Handhabung des Übersetzers

Der Übersetzer liest Quellprogramme von Lochkarten, Lochstreifen oder vom Plattenspeicher und übersetzt sie in Maschinencode (Abb. 1). Die erzeugten Programm-Moduln müssen mit den benötigten Unterprogrammen aus System- und gegebenenfalls Benutzerbibliotheken gebunden und zum Ablauf in den Arbeitsspeicher geladen werden.

Das Binden kann entweder in linearer Anordnung oder in OVERLAY-Technik erfolgen.

Der eigentliche Übersetzer besteht aus 11 Pässen und benötigt mindestens 4096 FW im Arbeitsspeicher. Für automatisches Binden und für Ausgabe des erzeugten Programms in Assemblersprache PROSA 300 kommen jeweils 2 weitere Pässe hinzu. Zur Beschleunigung des Übersetzungsvorganges kann beim Laden des Systems ein größerer Bereich im Arbeitsspeicher (bis 16 KFW) zugeteilt werden.

Auf dem Externspeicher benötigen Übersetzer und Binder einschließlich der erforderlichen Systembibliothek(en) über 200 KFW. Durch Verzicht auf nicht benötigte Bibliotheksfunktionen läßt sich der Externspeicher-Platzbedarf verringern.

### *2. 1 Laden des Übersetzers*

Der Übersetzer wird mit einem besonderen Ladeprogramm in die Arbeitsform vom Externspeicher geladen. Dabei werden seine Bereitstelladresse und sein Laufbereich festgelegt.

### *2. 2 Übersetzungslauf*

Zur Steuerung des Übersetzungsvorganges stehen wahlweise zwei Steuerprogramme zur Verfügung. Mit deren Hilfe wird die benötigte Information entweder über einen Bedienungsblattschreiber oder nach Art eines Monitors über Steuerlochkarten eingegeben. Folgende Möglichkeiten stehen zur Verfügung:

- a) Protokollierung auf Blattschreiber oder Schnelldrucker. Es werden alle Bedienungsanweisungen mitprotokolliert. Fehlermeldungen erscheinen als ausführlicher, selbsterklärender Text im Protokoll des Quellprogramms unmittelbar nach der fehlerhaften Zeile.
  - b) Eingabe des Quellprogrammes von Lochkarten (BCD-Code = Normalfall oder EBCDIC-Code wie beim SYSTEM 4004), von Lochstreifen oder aus einer wählbaren Plattenspeicher-Datei.
  - c) Ausgabe des erzeugten Programms als Modul oder als fertig gebundenes, selbständig lauffähiges Programm.
  - d) PROSA-Ausgabe des erzeugten Moduls. Im PROSA-Protokoll erscheinen die FORTRAN-Namen des Quellprogramms.
  - e) Einbau von Testhilfen:

Indextrace	Überwachung der Indizes von Feldkomponenten
Valuetrace	Protokollierung aller Wertzuweisungen beliebig wählbarer Variablen.
Sprungtrace	Protokollierung aller Statement-Nummern, die durch Sprung – also nicht im sequentiellen Programmablauf – erreicht werden.
Zeilentrace	Fehlermeldungen zur Programmlaufzeit werden als ausführlicher, selbsterklärender Text mit Modulnamen und Zeilennummer ausgegeben.
  - f) Zusätzliche Protokolle, wie Namensliste, Markenliste und Dimensionsliste.
  - g) Verlängern der Arbeitsdateien, falls deren Standardlänge nicht ausreicht.
  - h) Löschen der Arbeitsdateien nach der Übersetzung, um Externspeicherplatz freizugeben.
- Zwingend notwendig sind nur Angaben über die Protokollierung; alle anderen Wahlmöglichkeiten werden entweder nicht aktiviert oder durch Standardmaßnahmen ersetzt.

### *3. Erzeugung der gebundenen Programme*

Für die Herstellung ablauffähiger gebundener Objekt-Programme (in MC) sind nach der Übersetzung noch zusätzliche Maßnahmen erforderlich.

#### *3. 1. Binden der erzeugten Moduln*

Der Binder erzeugt ablauffähige Programme aus Moduln, die teils vom Übersetzer erzeugt sind, teils in Standardbibliotheken vorliegen. Solche Moduln können auch aus PROSA-Programmen mit Hilfe des PROSA-Übersetzers erzeugt sein.

Durch Steuerkarten kann festgelegt werden

- a) welche Bibliotheken in welcher Reihenfolge durchsucht werden sollen,
- b) welche Moduln aus welchen Bibliotheken verwendet bzw. von der Verwendung ausgeschlossen werden sollen,
- c) bei welchem Adreßpegel ein jeweils anzugebender COMMON-Bereich einzubauen ist.

Zusätzlich für den Overlay-Binder:

- d) welche Segmente überlagert werden sollen (Baumstruktur)
- e) wo ein neuer Verzweigungsbaum beginnt (Region)

Ein Beispiel für eine Overlay-Struktur zeigt Abb. 2.

Die Bindestrategie ermöglicht die Herstellung verschiedener Programmvarianten – z. B. auch für verschiedene Betriebssysteme – ohne Neuübersetzung des Quellprogramms, da der Übersetzer nie unmittelbar Betriebssystem-Aufrufe in Programmen erzeugt, sondern diese stets in den einzubindenden Moduln der Standardbibliotheken stehen.

### 3. 2. *Mischen von FORTRAN- und PROSA-Programmteilen*

Die Moduln, aus denen ein Gesamtprogramm zusammengestellt wird, können selbstverständlich auch in PROSA geschrieben sein, wie z. B. die Standardfunktionen. Ein solcher Modul kann ein Hauptprogramm, eine SUBROUTINE oder eine FUNCTION sein. Es müssen lediglich bestimmte Normierungen bei der Parameterversorgung von Unterprogrammen, bei der Verwendung von COMMON-Bereichen und BLOCK DATA-Unterprogrammen eingehalten werden.

### 3. 3. *Programmkopplung*

Unter Programmkopplung sei hier die Zusammenarbeit mit anderen, in der gleichen Anlage simultan laufenden Programmen verstanden. Hier bestehen folgende Möglichkeiten:

- a) Datenaustausch über Arbeitsspeicher oder Externspeicher mit Hilfe von READ/WRITE, binär oder alphanumerisch, außerdem im Arbeitsspeicher ohne Umspeichern über COMMON-Bereiche.
- b) Koordination durch Organisationsaufrufe an das Betriebssystem mit Hilfe entsprechender CALL-Statements.

## 4. *Prozeß-FORTRAN*

### 4. 1. *Grundsätzliches*

Prozeß-FORTRAN ist eine in der Literatur gebräuchliche Bezeichnung für die Erweiterung des Sprachvorrates von FORTRAN zur Programmierung von Prozeßaufgaben. Bei dem von uns verfolgten Weg wird das vorliegende Konzept des H-Level ausgebaut. Die vorgeschlagenen Erweiterungen betreffen neue Verknüpfungsmöglichkeiten zwischen Variablen, besondere Anweisungen für Textverarbeitung, zusätzliche Standardfunktionen zur Programmkoordinierung und für Prozeß-Ein/Ausgaben (PURDUE-Norm). Außerdem sollen wichtige Standardfunktionen von mehreren Programmen benutzbar sein. Bei Einhaltung bestimmter Normierungen muß die Einfügung von PROSA-Zeilen in das FORTRAN-Programm erlaubt sein.

Im folgenden werden die neuen Sprachmittel kurz vorgestellt.

### 4. 2. *Neue Verknüpfungen zwischen Variablen*

Zunächst wird die syntaktische Trennung von LOGICAL\*2 und INTEGER\*2 (ebenso für die \*4-Größen) aufgehoben.

Damit können nun .AND.,.OR. usw. auch auf INTEGER-Variable und umgekehrt \*,+,/ usw. auch auf logische Variable angewendet werden.

Zusätzlich werden für diese Variablen neue Operatoren eingeführt:

.EOR.        exklusives Oder  
.ISHIFT.K    arithmetische Verschiebung  
.LSHIFT.K    logische Verschiebung  
              K > 0 bedeutet Verschiebung nach links  
              K < 0 bedeutet Verschiebung nach rechts

#### 4. 3. *Neue Hilfsmittel für Textverarbeitung*

Um formatgesteuerte Texte auch im Kernspeicher auf- und abbauen zu können, werden entsprechend READ und WRITE die Anweisungen

EN-CODE        (K, F-Nr., Feld) Liste  
DE-CODE

eingeführt. Damit können K Zeichen ( $K \leq 120$ ) in das Feld „Feld“ geschrieben (EN CODE) bzw. aus ihm gelesen werden (DE-CODE). Mit diesen Funktionen ist eine flexible Textverarbeitung möglich.

Mit zusätzlichen Formatschlüsseln läßt sich das Datum in der Form WT.TT.MM.JJ (z. B. MI. 26.05.71) und die Uhrzeit in der Form HH.MM.SS ausgeben (z. B. 07.15.30).

#### 4. 4. *Execute-Aufrufe*

Zur Kennzeichnung von Unterprogrammen, die Sonderfunktionen des Betriebssystems ausführen, wird ein neues Schlüsselwort eingeführt:

EXECUTE Name (Parameterliste)

Diese Anweisung ist formal mit der Anweisung CALL vergleichbar. Jedoch werden durch das neue Schlüsselwort Doppeldeutigkeiten vermieden, die auftreten würden, wenn ein Anwender eigene Unterprogramme mit demselben Namen belegt. Bedeutet etwa EXECUTE P START (N) den Start eines Hauptprogramms der Priorität N, so kann unabhängig davon ein Anwender ein eigenes Unterprogramm mit CALL P START (N) aufrufen.

Man vermeidet so die Notwendigkeit, eine Reihe von reservierten Namen erfinden zu müssen.

#### 4. 5. *Standardfunktionen für Prozeßanwendungen*

Das „Interim Procedural Language Committee“ des PURDUE-Workshop (Lafayette, Indiana) hat im Oktober 1970 eine Reihe von Unterprogrammen für Prozeßanwendungen zur Normierung vorgeschlagen. Wir werden diese Programme zur Verfügung stellen und aus den schon erwähnten Gründen über EXECUTE aufrufbar machen. Insbesondere gehören hierzu

Starten und Beenden von Programmen nach einer Zeitverzögerung sowie die zeitverzögerte Fortsetzung des aufrufenden Programms.

Bei den Prozeß-Ein/Ausgaben sind zur Normierung vorgeschlagen

Analogeingabe sequentiell,  
Analogeingabe wahlfrei,  
Digitaleingabe – Impulsausgabe  
(alle Funktionen mit und ohne Warteaufruf).

Bestimmte Standardfunktionen werden unabhängig vom aufrufenden Programm in den sogenannten Common-Code-Bereich eingeordnet. Sie besitzen dann reentrant-Eigenschaften.

#### 4. 6. *Einfügen von PROSA-Zeilen in FORTRAN-Programme*

Für die Kopplung von FORTRAN- und PROSA-Programmen gibt es außer der bekannten Möglichkeit über Unterprogramme auch noch das direkte Einfügen von PROSA-Zeilen in das FORTRAN-Quellprogramm. Diese PROSA-Anweisungsfolge beginnt mit dem Schlüsselwort PROSA und wird jeweils durch das Schlüsselwort FORTRAN abgeschlossen.

Für das PROSA-Programm gibt es jedoch einige Einschränkungen:

- a) Es sind nur *lokale* Adreßdefinitionen erlaubt.
- b) Im Operationsteil dürfen nur Befehlsschlüssel stehen (nicht GK, DZ, AP usw.).
- c) Es sind im Adreßteil nur lokale Adressen, Variablennamen des FORTRAN-Programms, Externadressen und Konstante erlaubt.

Diese Einschränkungen sind notwendig, um eine sinnvolle Syntax-Kontrolle auch in den PROSA-Programmteilen durchzuführen.

Mit diesen Hilfsmitteln einer Prozeßversion des FORTRAN IV stehen, so meinen wir, dem Anwender jetzt sehr viel erweiterte Programmierungsmöglichkeiten zur Verfügung. Wir hoffen, daß sie nach Fertigstellung des Compilers auch ausgenutzt werden und somit das Programmieren von Prozeßaufgaben in höheren Sprachen sich weiter durchsetzt.

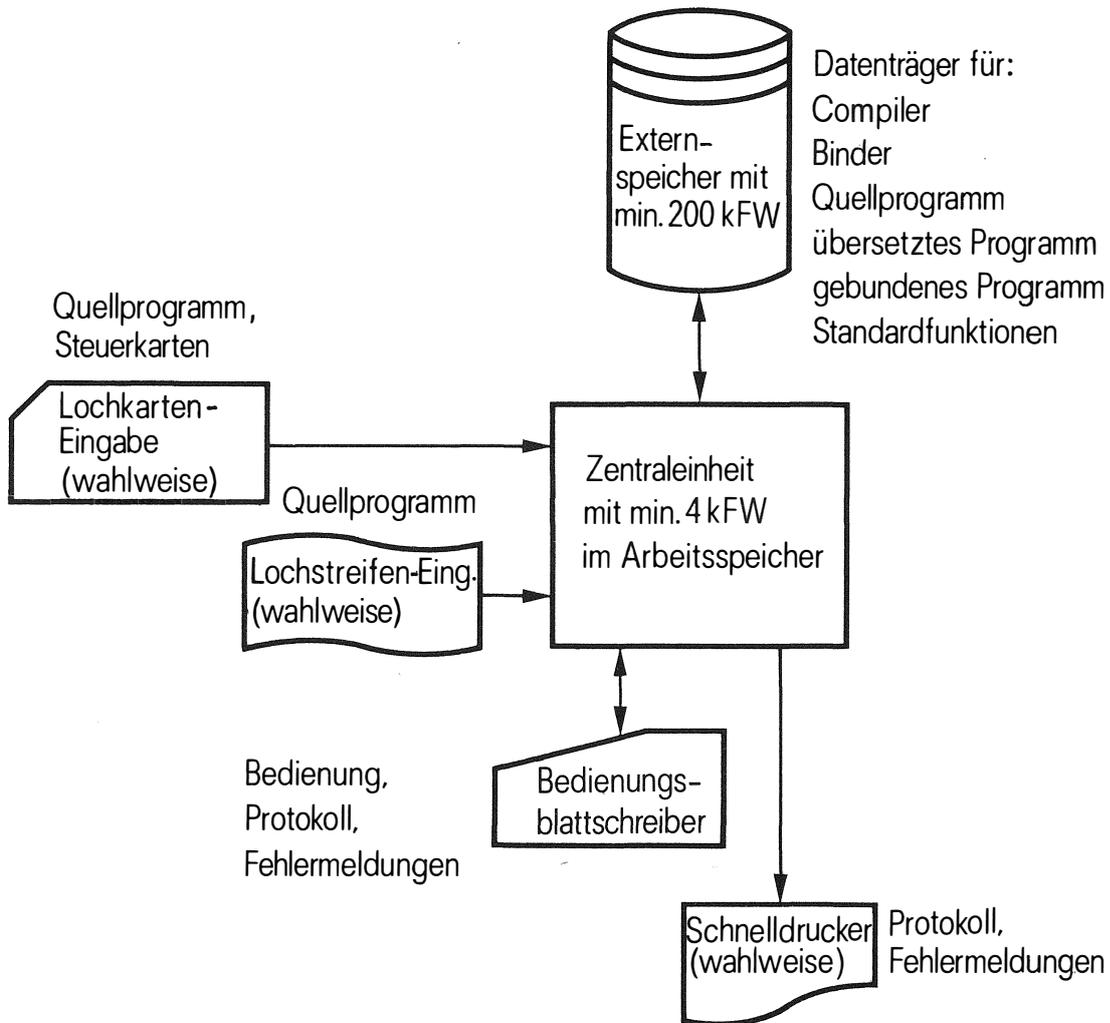
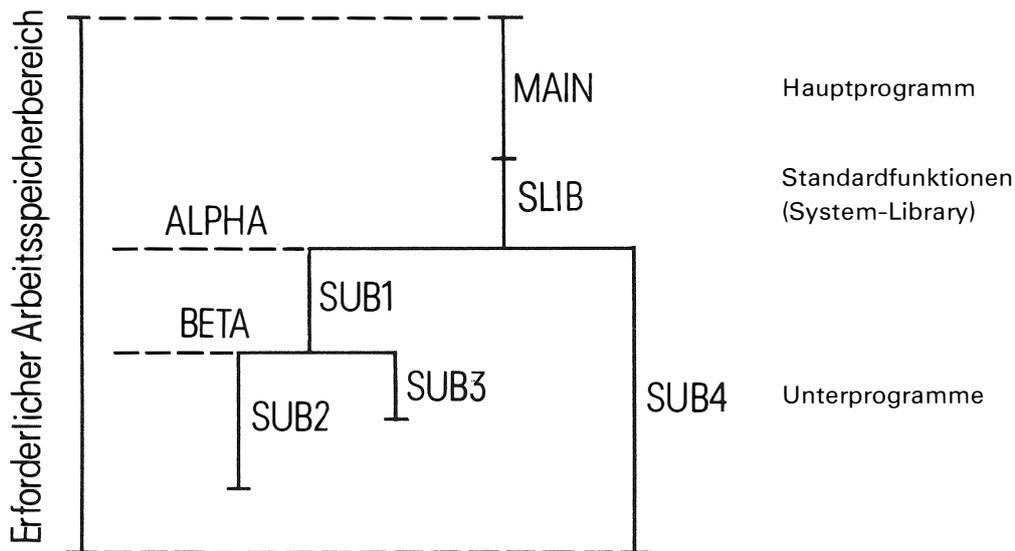


Abb. 1 Anlagenkonfiguration für FORTRAN IV-Level



*Steuerkarten für diese Struktur*

PROGRAM	TEST, 2000	Programm TEST soll ab 2000 bereitgestellt werden
INCLUDE	MAIN	System-Library an MAIN angekoppelt
LIBRARY	SLIB	
OVERLAY	ALPHA	Definition von ALPHA für den
INCLUDE	SUB1	} 1. Hauptzweig
OVERLAY	BETA	
INCLUDE	SUB2	
OVERLAY	BETA	
INCLUDE	SUB3	
OVERLAY	ALPHA	Definition von ALPHA für den
INCLUDE	SUB4	} 2. Hauptzweig
ENTRY		Ende der Steuerkarten

Abb. 2 Beispiel einer einfachen Overlay-Struktur

**Gerd Mrozynski**

TU Berlin (früher Hahn-Meitner-Institut) Berlin

Ein Magnetband-Platten-Editor für das System 300

## Ein Magnetband-Platten-Editor für das System 300

Das Programm ermöglicht es, Quellprogramme, also alphanumerische Zeichenfolgen, die in Dateien auf externen Speichern abgelegt sind, zu ändern, zu ergänzen oder Teile davon zu löschen. Der Anwender kann dabei im Dialogbetrieb von einem der angeschlossenen Fernschreibelemente direkt seine Anweisungen an das Programm geben. Dieser Dialogbetrieb bedeutet gegenüber Lösungen, bei denen die Anweisungen zunächst auf Lochstreifen oder Lochkarten gespeichert sind, eine längere Belegung des Fernschreibelementes. Erfahrungsgemäß ist aber die Anfertigung dieser Korrekturstreifen umso fehleranfälliger, je mehr Möglichkeiten das Programm bietet, so daß der hier beschriebenen Lösung der Vorzug gegeben würde.

Die Quellprogramme werden auf dem Externspeicher so abgelegt, daß der PROSA-Übersetzer in der Externspeicherversion die generierten Dateien übersetzen kann. Der Programmierer ist damit weitgehend von Lochkarten und Lochstreifen unabhängig, wenn er seine Programme ändert und testet.

Das Programm weist eine Struktur auf, in der einzelne Blöcke mit verschiedenen Funktionen durch ein übergeordnetes Decoder-Programmteil verbunden sind. Die einzelnen Blöcke mit festen Funktionen haben Zugriff zu bestimmten Datenpuffern und veranlassen die Datenübertragung zwischen ihnen. Die Anweisungen an das Programm gliedern sich in Bedienungsanweisungen, die eine Änderung der aktuellen Zeile beinhalten (z. B. Löschen eines string, Ersetzen eines string durch einen anderen), Ein-Ausgabeanweisungen, die das Eingabemedium (Externspeicher, Lochstreifenelement, Blattschreiber) festlegen oder die Protokollierung auf einem der angewählten externen Geräte bewirken und Steueranweisungen, die den Programmablauf selbst betreffen (Abbruch, Beenden, Rücksetzen auf Dateianfang).

Auf dem Externspeicher wird programmintern eine zweite Datei zur Zwischenspeicherung von Daten für die Dauer des Programmlaufes eröffnet. Während eines Durchlaufes werden die Zeichen des Quellprogrammes zeilenweise aus der Datei gelesen, in Programmpuffern zur Änderung bereitgestellt und bei weiterführenden Anweisungen in die Hilfsdatei übertragen. An beliebiger Stelle kann der Anwender das Einlesen von der Datei auf dem Externspeicher anhalten und über einen dynamisch zu wählenden Datenpuffer vom Lochstreifenelement einlesen. Auch für dieses Eingabemedium bleiben die Korrekturanweisungen gültig. Beim Beenden des Programmes werden die geänderten Daten unter gleichem Dateinamen auf dem Externspeicher abgelegt. Wurde mit der Platte gearbeitet, gehen die ursprünglichen Daten verloren und die neu generierten stehen im allgemeinen an anderer Stelle auf dem Plattenspeicher. In der Magnetbandversion werden die geänderten Daten in eine Datei mit um 1 erhöhter Versions-Nr. zusätzlich auf das gleiche Band geschrieben.

Nach dem Beenden muß das Programm neu bereitgestellt werden, da ein einleitender Programmteil im Overlay über die Datenpuffer geschrieben wurde und somit beim Lesen verlorengelht.

Das Programm belegt im Arbeitsspeicher 1705 Zellen bei der Plattenversion und 2016 Zellen bei der Magnetbandversion.

Weitere Einzelheiten können dem Bericht B 98 des Hahn-Meitner-Instituts für Kernforschung aus der Reihe Nukleare Datenverarbeitung B-NDV 6 vom August 1970 entnommen werden.

**K. H. Degenhardt**  
**W. Woletz**

Hahn-Meitner-Institut für Kernforschung Berlin GmbH

Ein modulares Programmsystem zur Vielkanalanalyse mit dem System 300

## Ein modulares Programmsystem zur Vielkanalanalyse mit dem System 300

### *Zusammenfassung*

Es wird ein Programmsystem für die S 301 am Beispiel der Vielkanalanalyse beschrieben. Das System besteht aus

- a) einem Steuerprogramm, das die organisatorische Arbeit des Programmsystems übernimmt, z. B. die Verwaltung von Warteschlangen.
- b) einer größeren Anzahl von modular geschriebenen Programmen zur eigentlichen Datenverarbeitung, z. B. Auswerte-Programme, Display-Programme, Ein- und Ausgabe-Programme.
- c) einem Benutzer-Programm, das den logischen Ablauf des Experimentes enthält.

### *Anlagenkonfiguration*

An den Rechner S 301 sind jeweils über ein P3-Element zwei ADC's, ein Speicherdisplay und eventuell eine Platte angeschlossen. Daneben gibt es noch Lochstreifeneingabe und -ausgabe und einen Bedienungsblattschreiber.

### *Ein Beispiel aus der Kernphysik*

Mit einem Detektor wird die Energie einer Teilchenart gemessen. Der Detektor wandelt die Energie in einen Impuls um. Dieser Impuls wird von einem Analog-Digital-Converter in einen Zahlenwert umgewandelt. Man läßt das Experiment, die Messung der Energie von Teilchen, eine Zeit lang laufen und interessiert sich dann für die Verteilung der Energiewerte. Dazu wird die Energieachse in viele Intervalle geteilt. Diese Intervalle nennt man auch Kanäle und daher der Name „Vielkanalanalyse“. Teilt man die Energieachse z. B. in 1 024 Kanäle von 0 bis 1 023 ein und ist ein vom ADC gelieferter Zahlenwert 215, so erhöht man den Inhalt des Kanals 215 um eins, usw. Man erhält auf diese Weise eine Häufigkeitsverteilung der gemessenen Energiewerte, man erhält ein Spektrum.

Es werden Daten gesammelt, von ADC's in Zahlenwerte umgewandelt und dann in Spektren sortiert. Die Programme, die diese Aufgabe übernehmen, heißen ADC-Programme. Die Display-Programme bieten die Möglichkeit, sich Spektren unter verschiedenen Darstellungsarten anzusehen. Zu den Auswerte-Programmen gehören z. B. Programme, die Spektren glätten, 2 Spektren addieren oder subtrahieren usw. Daneben gibt es noch Eingabe- und Ausgabe-Programme. Über diesen Programmen steht ein Steuerprogramm, das die Menge dieser Programme erst zu einem Programmsystem macht.

### *Organisation des Programmsystems*

Die Organisation des Programmsystems, der Aufbau der Verwaltung, ist im wesentlichen unabhängig davon, daß damit Vielkanalanalyse gemacht werden soll.

Jede der genannten Programmgruppen (ADC-, Display-, Auswerte-, Eingabe-, Ausgabe-Programme) hat eine eigene Programmnummer, eine Priorität. Das Steuer-Programm bekommt dabei die relativ höchste Priorität. Das ist einmal deswegen gemacht worden, um

die Möglichkeit der Simultanarbeit zu nutzen. Zum anderen kann die in den Rechner fließende Datenrate so groß sein, daß man es sich nicht leisten kann, alle Programme auf dem gleichen Level laufen zu lassen.

Zu jeder der Programmgruppen gehört eine Anzahl von Programmen. Diese Programme benötigen zu ihrem Lauf in der Regel Parameter, die sie entweder vom Benutzer über eine Blattschreibereingabe oder von anderen Programmen durch einen Programmaufruf erhalten. Die Parameterübergabe muß daher genormt sein. Die Programme sind als Unterprogramme geschrieben:

```

NAME    NOP  O
        SPR  ANFANG
        NOP  n      Parameteranzahl
PARI    NOP
        .
        .
        .
PARn    NOP      n-ter Parameter
ANFANG  TEP  ...
        .
        .

```

Die Parameter müssen in die dafür vorgesehenen Zellen übertragen werden, bevor der Einsprung in das Programm erfolgt. Will ein Programm ein anderes Programm zum Laufen bringen, so geschieht das über den Makro

```
ADR2 MA WS = AUSW, ADR1
```

Dieser Aufruf bewirkt einen Sprung in das Verwaltungsprogramm. Mitgeliefert wird die Adresse ADR1, unter der die Parameter zu finden sind:

```

ADRL    AN  'SUBS'
        NOP  3
        NOP  PUF1
        NOP  PUF2
        NOP  PUF3

```

Zu diesen Parametern gehört auch der Name des Programms, hier SUBS (subtrahiere 2 Spektren), das mit den anderen Parametern laufen soll.

Will man dasselbe Programm von außen über den Bedienungsblattschreiber aufrufen, so geschieht das über die Bedienungsanweisung:

```
AUSG, SUBS, PUF1, PUF2, PUF3;
```

In beiden Fällen gibt AUSG (oder DISP, EING, usw.) die Priorität an, unter der ein Programm laufen soll. Da i. a. mehrere Programme die gleiche Priorität haben, wird für jede Programmgruppe, d. h. für jede Programmnummer, vom Steuer-Programm eine freie Warteschlange geführt. Die Warteschlangen sind ähnlich aufgebaut wie die Warteschlangen, die das ORG für die externen Geräte bildet. Sowohl innere als auch äußere Aufrufe eines Programms werden vom Steuer-Programm in die betreffende Warteschlange eingereiht.

Zur Synchronisation des Programmablaufs ist es erforderlich, daß man an einer beliebigen Stelle eines Programms so lange warten kann, bis ein Auftrag erfüllt ist, d. h. bis ein aufgerufenes Programm beendet ist. Zu diesem Zweck gibt es den Makro

```
MA AUWART = ADR2
```

An dieser Stelle wird so lange gewartet, bis der unter ADR2 stehende Aufruf abgearbeitet ist.

Aufgabe des Benutzers ist es, mit Hilfe der beschriebenen Makros ein Programm zu schreiben, das den richtigen Ablauf seines Experimentes gewährleistet. Zu einem Programmlauf braucht nur dieses Benutzerprogramm übersetzt zu werden, wenn das Steuerprogramm an eine vereinbarte Adresse geladen wird. Die Programme der beschriebenen Programmgruppen werden, wenn eine Platte vorhanden ist, nach Bedarf vom Steuer-Programm geladen.

**K. Weise**

Physikalisch-Technische Bundesanstalt, Braunschweig

Ein Tischrechenprogramm für Siemens-Prozeßrechner des Systems 300

## Ein Tischrechenprogramm für Siemens-Prozeßrechner des Systems 300\*)

### *Zusammenfassung*

Mittels des Programmsystems TIRE können die an einer Rechenanlage des Systems 300 angeschlossenen Blattschreiber simultan in einer Art Time-Sharing-Betrieb als programmierbare Tischrechenmaschinen benutzt werden.

Beim Einsatz von Prozeßrechnern im wissenschaftlichen Bereich, z. B. zur Erfassung von Meßdaten kernphysikalischer Experimente, kommt es häufig vor, daß die Experimentatoren kurzfristig und schnell kleinere mathematische Berechnungen durchführen möchten. Eine ALGOL- oder FORTRAN-Programmierung der Probleme ist jedoch oft nicht sinnvoll, da während des tagelang dauernden On-line-Betriebs bei laufenden Messungen meist nicht genügend freier Arbeitsspeicherraum für die Übersetzung und den Ablauf der Programme zur Verfügung steht und eine Vergabe der Programme an eine Großrechenanlage aus Zeit- und Aufwandsgründen oder wegen zu geringer Bedeutung der Probleme nicht in Frage kommt. Andererseits sind die Rechenaufgaben doch so kompliziert, daß kleinere und auch programmierbare Tischrechenmaschinen zu ihrer Lösung nicht geeignet sind. Neuerdings sind auch Tischcomputer auf dem Markt, die programmierbar sind, Standardfunktionen zu berechnen gestatten und bedingte Programmsprünge ausführen können. Diese sind unter den Rechenmaschinen den anstehenden Problemen am besten angepaßt, ihre Anschaffung erfordert jedoch erhebliche zusätzliche Geldmittel. Auch ist nicht einzusehen, warum nicht doch ein vorhandener, wenn auch stark belasteter Prozeßrechner für diese Zwecke benutzt werden kann. Dieser ermöglicht zudem noch den Zugriff zu automatisch erfaßten Daten und den Eingriff in den Prozeß.

Aus diesen Gründen wurde das Programmsystem TIRE entwickelt, welches bis zu einem weiteren Ausbau gestattet, die zur Bedienung zugelassenen Blattschreiber, die am Bedienungs- oder Fernschreibelement einer Siemens-Rechenanlage des Systems 300 angeschlossen sind, simultan in einer Art Time-Sharing-Betrieb und neben einer laufenden Verarbeitung von Prozeßdaten als programmierbare Tischrechenmaschinen zu benutzen. Für die Lösung der auftretenden mathematischen Probleme durch das Programmsystem wurde eine Programmiersprache entworfen, die sich bei der Darstellung von Ausdrücken an FORTRAN anlehnt und sonst so einfach ist, daß sie gerade noch nicht zu schwierige Aufgaben bequem zu handhaben erlaubt. Besondere Anstrengungen wurden unternommen, um den Arbeitsspeicherbedarf möglichst klein zu halten. Bei der fertiggestellten arbeitsspeicherresidenten Version des Programmsystems benötigt ein zentraler Teil, der ein Vorbereitungs- und Bedienungsprogramm, den interpretativ arbeitenden Übersetzer in Form eines Unterprogramms, die Standardunterprogramme und einen Kellerspeicher für Zwischenergebnisse bei Ausdrücken enthält, fast genau 2 K und jedes der den einzelnen Teilnehmern am Tisch-

\*) Mitteilung aus der Physikalisch-Technischen Bundesanstalt

rechenbetrieb zugeordneten Steuerprogramme einen Bereich wählbarer Länge von wenigstens etwa 250 Speicherplätzen.

Die Arbeitsweise des Programmsystems TIRE und die Syntax der Programmiersprache (s. Anhang B) seien an Hand eines einfachen Beispiels (s. Anhang A) erläutert. Nach der Bereitstellung und dem Start von TIRE wird jedem Teilnehmer eine Programmnummer  $n$ , ein Arbeitsspeicherbereich (Adresse  $a$  bis  $e$ ) und ein beliebiges Kennzeichen  $x$  durch eine Anweisung  $TSH=n,a,e,x$ ; zugewiesen. Danach und nach Abgabe der Programmdefinitionsanweisung  $DEFx$ ; kann dieser nun das Tischrechenprogramm in Form einer Folge von durch Semikolon abgeschlossenen Anweisungen entsprechend der Syntax von Hand oder per Lochstreifen eingeben. Eine zusätzliche Korrekturmöglichkeit zu der üblichen durch das Irrungszeichen  $\#$  besteht darin, daß mittels der uneigentlichen Anweisung  $*$ ; die vorhergehende gelöscht und danach diese wiederholt werden kann. Die Syntax kennt nur 5 Anweisungsarten: Wertzuweisungen, Ein- und Ausgaben, Marken und bedingte Sprünge. Ein einzelnes Semikolon, gleichzeitig Endeanweisung des Tischrechenprogramms, beendet die Eingabe. Durch die Bedienungsanweisung  $LSTx$ ; kann das geladene Programm aufgelistet und ein Lochstreifen davon erstellt werden. Die Bedienungsanweisung  $RUNx$ ; löst den Programmablauf aus, der u. U. abgebrochen werden kann durch  $HLT=n,x$ ; Die Abarbeitung des Programms erfolgt mit Syntaxkontrolle interpretativ ohne Erstellung eines Maschinencodes. Programmierfehler werden angezeigt und Ausdrücke in Anlehnung an Ref.1) mit Hilfe rekursiver Unterprogramme berechnet. Nach jeder Anweisung erfolgt ein zyklischer Programmwechsel zum nächsten Teilnehmer. Bei jedem Programmsprung beginnt die Abarbeitung von vorn, jedoch wird das Programm bis zur Zielmarke überlesen. Während des Ablaufs werden Daten in einer standardmäßigen Kolonnenform, bei der auch die Variablennamen erscheinen, ein- oder ausgegeben. Nach Ablauf des Tischrechenprogramms kann ein weiterer Lauf angeschlossen oder auch ein neues Programm eingegeben werden.

Anhang A

Programmbeispiel

Berechnung der Gammafunktion nach der Formel

$$\Gamma(x) \approx \sqrt{2\pi x} \left(\frac{x}{e}\right)^x e^{1/12x}$$

mit einem relativen Fehler < 3% für  $x \geq 1$ .

Für  $x < 1$  wird  $\Gamma(x) = 0$  gesetzt.

```
BRPS=7,2100,TIRE;;;BR;STRT=7;;           Vorbereitung
TSH=8,4200,4600,Q;;;DEFQ;;;             Q Teilnehmerkennung
X;;Y=X-1;Y'1;GAMMA=0;GO'2;1;           Programmierung
GAMMA=SQRT(2*3.141593*X)*X**X*EXP(-X+1/12/X);
5W;*;2;GAMMA;;                           Korrektur durch *;
/Q;LSTQ;;                                 Liste

X;;                                       Eingabe
Y=X-1;                                   Wertzuweisung
Y'1;                                     Sprung wenn Y ≥ 0 nach Marke 1
GAMMA=0;
GO'2;                                   unbedingter Sprung
1;                                       Marke
GAMMA=SQRT(2*3.141593*X)*X**X*EXP(-X+1/12/X);
2;
GAMMA;                                   Ausgabe
;                                       Ende
/Q;RUNQ;;                                Ablauf

  X = 2;                                 Eingabe
  GAMM = 2.000648;                       Ausgabe
/Q;RUNQ;;

  X = .5;
  GAMM = 0;
/Q;
ENDW=8;;
ENDW=7;;
```

## Anhang B Syntax der Programmiersprache

**1. Programm** Das Tischrechenprogramm besteht aus einer Folge von Anweisungen. Ein einzelnes Semikolon markiert als uneigentliche Anweisung das Programmende.

$\langle \text{Programm} \rangle ::= ; | \langle \text{Anweisung} \rangle \langle \text{Programm} \rangle$

**2. Anweisungen** Das letzte Zeichen jeder Anweisung ist ein Semikolon.

$\langle \text{Anweisung} \rangle ::= \langle \text{Marke} \rangle | \langle \text{Sprung} \rangle | \langle \text{Wertzweisung} \rangle$   
 $\langle \text{Eingabe} \rangle | \langle \text{Ausgabe} \rangle$

$\langle \text{Marke} \rangle ::= \langle \text{Ziffernkette} \rangle;$  (z. B.: 1; 037;)

$\langle \text{Sprung} \rangle ::= \langle \text{Variable} \rangle' \langle \text{Marke} \rangle$  (z. B.: Y'5; GO'10;)

Ist der Wert der Variablen nicht negativ, so wird nach der angegebenen Marke gesprungen.

Sonderfall: Die Variable GO wird nicht abgefragt, sondern ein unbedingter Sprung ausgeführt.

$\langle \text{Wertzweisung} \rangle ::= \langle \text{Variable} \rangle = \langle \text{Ausdruck} \rangle;$

(z. B.: A=B\*\*(-2/3); Z=X-SIN(Y);)

$\langle \text{Eingabe} \rangle ::= \langle \text{Variable} \rangle ;$  (z. B.: PETER;)

Der Wert der Variablen wird nach Ausgabe ihres Namens über Blattschreiber angefordert und ist in der Form  $\langle \text{reelle Zahl} \rangle$  (s.5.) einzugeben (z. B.: PETE = -2.5\$-3;).

$\langle \text{Ausgabe} \rangle ::= \langle \text{Variable} \rangle ;$  (z. B.: OTTO;)

Der Name und der Wert der Variablen werden über Blattschreiber in der Form  $\langle \text{reelle Zahl} \rangle$  (s.5.) ausgegeben (z. B.: OTTO = -2.50000\$- 13;).

$\langle \text{Variable} \rangle ::= \langle \text{Name} \rangle$

Alle Variablen und auch alle anderen Zahlengrößen sind vom Typ Real.

### 3. Zeichen und Zeichenketten

$\langle \text{Ziffernkette} \rangle ::= \langle \text{Ziffer} \rangle | \langle \text{Ziffernkette} \rangle \langle \text{Ziffer} \rangle$

$\langle \text{Name} \rangle ::= \langle \text{Buchstabe} \rangle | \langle \text{Name} \rangle \langle \text{Ziffer} \rangle |$   
 $\langle \text{Name} \rangle \langle \text{Buchstabe} \rangle$  (z. B.: BERLIN 1)

Nur die maximal 4 ersten Zeichen eines Namens oder der Ziffernkette bei einer Marke sind von Bedeutung, die übrigen redundant.

$\langle \text{Ziffer} \rangle ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$

$\langle \text{Buchstabe} \rangle ::= A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T |$   
 $U | V | W | X | Y | Z$

$\langle \pm \rangle ::= + | -$

$\langle * / \rangle ::= * | /$

Zeichen „Zwischenraum“ können beliebig in das Programm eingestreut werden. Sie werden überlesen.

### 4. Ausdrücke Schreibweise wie in FORTRAN

$\langle \text{Ausdruck} \rangle ::= \langle \text{Glieder} \rangle | \langle \pm \rangle \langle \text{Glieder} \rangle |$   
 $\langle \text{Ausdruck} \rangle \langle \pm \rangle \langle \text{Glieder} \rangle$

$\langle \text{Glieder} \rangle ::= \langle \text{Faktor} \rangle | \langle \text{Glieder} \rangle \langle * / \rangle \langle \text{Faktor} \rangle$

$\langle \text{Faktor} \rangle ::= \langle \text{Primausdruck} \rangle | \langle \text{Faktor} \rangle ** \langle \text{Primausdruck} \rangle$

$\langle \text{Primausdruck} \rangle ::= \langle \text{Zahl} \rangle | \langle \text{Variable} \rangle | \langle \text{Ausdruck} \rangle |$   
 $\langle \text{Funktionsaufruf} \rangle$

$\langle \text{Funktionsaufruf} \rangle ::= \langle \text{Name} \rangle ( \langle \text{Ausdruck} \rangle )$

Bei Funktionsaufrufen sind nur die Namen EXP, SIN, COS, ABS, LN, SQRT, ARCTAN (hier genügt ARCT) erlaubt, d. h. nur die der Standardfunktionen.

5. *Zahlen* Alle Zahlen werden als Gleitpunktzahlen behandelt. Sie müssen im erlaubten Zahlenbereich des Systems 300 liegen.

$\langle \text{Zahl} \rangle ::= \langle \text{Mantisse} \rangle \mid \langle \text{Mantisse} \rangle \langle \text{Exponent} \rangle \mid \langle \text{Exponent} \rangle$

$\langle \text{Mantisse} \rangle ::= \langle \text{Ziffernkette} \rangle \mid \langle \text{Ziffernkette} \rangle . \mid$

$\quad . \langle \text{Ziffernkette} \rangle \mid \langle \text{Ziffernkette} \rangle . \langle \text{Ziffernkette} \rangle$

$\langle \text{Exponent} \rangle ::= \$ \langle \text{Ziffernkette} \rangle \mid \$ \langle \pm \rangle \langle \text{Ziffernkette} \rangle$

$\langle \text{reelle Zahl} \rangle ::= \langle \text{Zahl} \rangle ; \mid \langle \pm \rangle \langle \text{Zahl} \rangle ;$

#### *Literatur*

<sup>1)</sup> F.E.J. Kruseman Aretz, Elektronische Datenverarbeitung 6 (1964) 233

**Dipl.-Ing. D. Kollbach**

Hahn-Meitner-Institut für Kernforschung Berlin GmbH

Das Camac-Instrumentierungssystem

# Das CAMAC-Instrumentierungssystem

## 1. Einleitung

Das CAMAC-System ist ein Datenverarbeitungssystem, das vorwiegend im Zusammenhang mit einem Prozeßrechner betrieben wird. Es erlaubt die Übertragung hoher Datenraten zwischen Prozeß und Rechner und ermöglicht den Eingriff in den Prozeß, so daß z. B. auch Regelkreise realisiert werden können, die beliebig komplex sein dürfen.

Das System ist modular aufgebaut, wobei die Instrumentierungsbausteine i. a. nicht speziell auf die Anwendung zugeschnitten sind; sie können häufig bei anderen Prozessen wieder verwendet werden.

CAMAC ist *nicht rechnerspezifisch*.

Das System ist international genormt (bisher Westeuropa und USA). Selbst die hohen Anforderungen in der Kernphysik lassen sich erfüllen, z. B.

- gelegentlich hohe Datenraten
- statistisch anfallende Daten
- die Möglichkeit, daß Geräte, die bedient werden „möchten“, sich beim Rechner melden.

Zur Zeit entsteht in Zusammenhang mit PEARL eine CAMAC-Befehlssprache, die in ihrer Struktur rechnerunabhängig ist.

## 2. Der CAMAC-Branch

Das Bild zeigt eine typische Konfiguration des CAMAC-Systems. Irgendein beliebiger Prozeß liefert und empfängt Daten. Die in Überraahmen steckenden CAMAC-Geräte (modules) empfangen oder liefern die Daten. Über ein 86-poliges Bus-System (Dataway) stehen die Modules mit der Rahmensteuerung in Verbindung. Diese ist ein Interface zum Branch Highway, einem Kabel mit 66 Adernpaaren, das bis zu 7 Überraahmen gleichzeitig mit dem Rechner-Interface, dem „Branch Controller“, verbindet. Der Branch Controller, ein typischerweise rechnerspezifisches Gerät, besitzt einerseits ein Branch-Highway Port (die Nahtstelle des Branch Highways) und andererseits eine Nahtstelle zu einem Rechner.

Im HMI Berlin wurden zwei verschiedene Branch Controller entwickelt; einer benutzt einen Kanal des Arbeitsspeicher-Multiplexers der S 301, der andere verwendet die Akkumulator-zusatz-Nahtstelle der S 301.

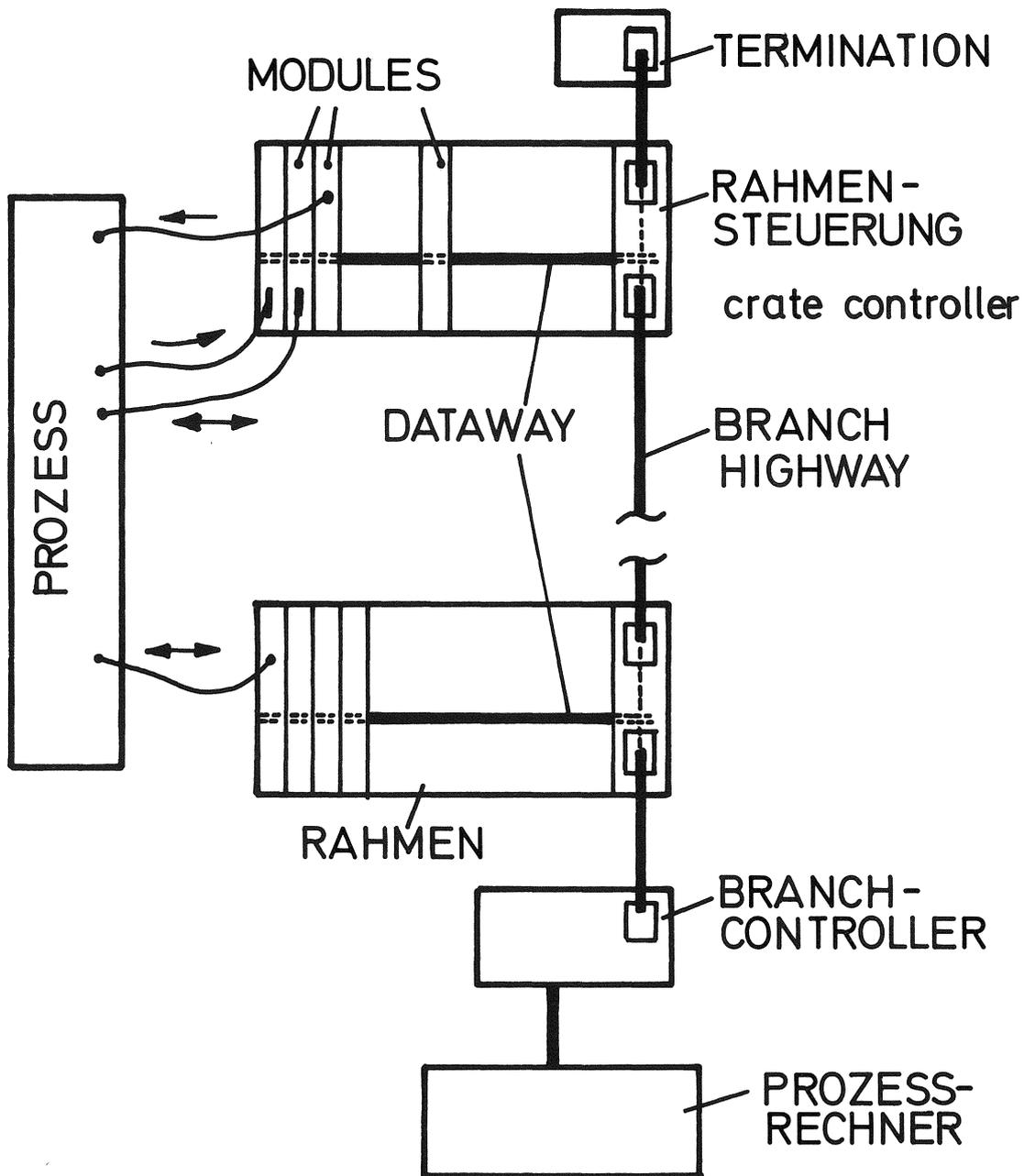
## 3. Branch-Operationen

### 3.1 Befehlsoperationen

Es gibt drei verschiedene Arten von Befehlsoperationen:

- Lesen von Daten. Der Rechner legt die Adresse eines Modules zusammen mit dem Funktionscode (z. B. „Lesen eines Registers“) heraus. Das adressierte Module sendet daraufhin ein Datenwort (24 bit), das der Rechner empfängt.
- Schreiben von Daten. Der Rechner legt zusammen mit der Adresse eines Modules und

# \_CAMAC BRANCH\_



dem Funktionscode, z. B. „Überschreiben eines Reg.“, Schreib-Daten heraus, die das adressierte Module empfängt.

– Steuerbefehle. Ohne daß eine Datenübertragung stattfindet, teilt der Rechner einem Module, das er adressiert, einen Befehl mit, z. B. „Löschen eines Registers“.

Innerhalb eines Modules können bis zu 16 verschiedene Register mit Hilfe des Subadreß-codes angesprochen werden.

### *3.2 Anforderungsbearbeitung*

Modules können beim Rechner melden, daß sie bedient werden möchten (z. B. von einem entsprechenden Programmstück). Das ODER aller Anforderungen liegt auf einer speziellen Leitung des Branch Highways. Der Rechner ermittelt mit besonderen Operationen die Herkunft der Anforderung. Hat der Rechner erkannt, welches Module eine Anforderung gestellt hat, kann er das Module mit einer entsprechenden Routine bedienen (z. B. mit Parametern versorgen, Daten lesen, Daten schreiben).

### *4. Das Timing*

Die Branch-Zyklen werden nach dem Anforderungs-Rückmeldungs-Prinzip (Handshake-Verfahren) abgewickelt, so daß die Laufzeiten auf dem Branch Highway und damit die Länge des Branch-Highways im Prinzip beliebig groß sein können. Ein Branch Zyklus dauert typisch  $1,5 \mu\text{s}$ .

### *Literaturhinweis*

Euratom-Berichte

EUR 4100e: CAMAC – A Modular Instrumentation System for Data Handling

EUR 4600e: CAMAC – Organisation of Multicrate Systems

**Dipl. Phys. Werner Gold**

Siemens AG, Karlsruhe

Datenerfassung und Datenauswertung mit Prozeßrechnern bei Gaschromatographen  
und Massenspektrometern

## Datenerfassung und Datenauswertung mit Prozeßrechnern bei Gaschromatographen und Massenspektrometern

(Kurzfassung eines Vortrages während der Tagung des Siemens-Prozeßrechner-Anwenderkreises in Berlin am 25. Mai 1971)

Gaschromatographen und Massenspektrometer sind in der analytischen Chemie die am meisten verwendeten Geräte. Der Gaschromatograph trennt ein Substanzgemisch auf Grund des verschiedenen Adsorptionsverhaltens der einzelnen Komponenten. Das Gemisch läuft dazu durch eine sogenannte Trennsäule (Rohr), die mit speziellen Trennungsmitteln, meist hochporösen Kunststoffen, gefüllt ist.

Substanzen, die kaum Adsorption am Füllmittel zeigen, erreichen sehr bald das Ende der Trennsäule, d. h. den Detektor, andere Substanzen kommen später. Diese sogenannte Retentionszeit ist charakteristisch für die Substanz. Kennt man also Säulenfüllung und Retentionszeit, so kann man die Substanz identifizieren.

In der Praxis treten nun eine Reihe von Schwierigkeiten auf, von denen nur die wesentlichsten hier erwähnt werden sollen:

- Die Dynamik des analogen Meßsignals beträgt  $1:10^6$ . Man muß also mit mehreren Meßbereichen arbeiten.
- Die Basislinie ist nicht konstant, sie wandert.

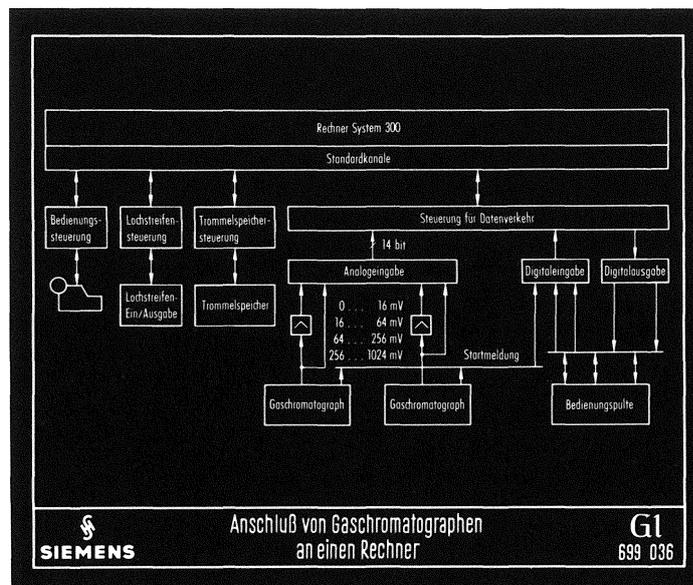


Abb. 1 Anschluß von Gaschromatographen an einen Prozeßrechner

– Die auftretenden Peaks sind fast immer überlappt, d. h. die richtige Peaktrennung ist ein fundamentales Problem

– Die Peakbreite wächst mit zunehmender Analysendauer.

Die Erfassung der Dynamik des Meßsignals ist ein reines Hardware-Problem, während Basislinienkorrektur, Peaktrennung und Peakbreitenkorrektur Software-Probleme sind.

Den Anschluß von Gaschromatographen an einen Prozeßrechner zeigt Abb. 1. Der Ausgang des Gerätes wird in zwei Kanäle aufgesplittet. Durch einen externen Verstärker am Gaschromatographen und einen (im Bild nicht eingetragenen) umschaltbaren Verstärker in der Analogeingabe ergeben sich vier Meßbereiche und eine Dynamik von  $1:10^6$ .

Vor Beginn der Messung werden dem Rechner über ein Bedienungspult die wichtigsten Meßparameter in Form einer Schlüsselnummer mitgeteilt. Solche Parameter sind zum Beispiel:

- Probenbezeichnung
- Analysendauer
- Eichfaktoren
- Peaktrennungsmethoden
- Nummer des Retentionszeit-Kataloges

Nach Beginn der Messung wird der Gaschromatograph zyklisch in festen Zeitabständen, z. B. alle 100 ms, abgefragt, wobei die einzelnen Meßbereiche vom Datenerfassungsprogramm automatisch umgeschaltet werden. Nachdem alle Meßwerte erfaßt sind, erfolgt die Auswertung mit Peakerkennung, Paektrennung, Zuordnung von Retentionszeit und Komponente des eingegebenen Substanzgemisches und Protokollierung. Ein Beispiel für ein Analysenprotokoll gibt Abb. 2. In der letzten Spalte (TR) ist dabei vermerkt, ob eine Peaktrennung vorgenommen wurde.

Dabei bedeutet „B“ Basislinie und „T“ Tal, d. h. Minimum zwischen zwei Peaks. Man erkennt, daß Peak 2,3 und 4 überlappt waren, denn Peak 2 wurde berechnet von der Basislinie zu einem Tal (BT), Peak 3 von einem Tal zum nächsten (TT), Peak 4 von einem Tal zur Basislinie (TB).

In einem Gaschromatogramm können natürlich auch Substanzen auftauchen, die nicht im Retentionszeit-Katalog enthalten sind. In einem solchen Fall wird man von dieser Substanz ein Massenspektrum aufnehmen, indem man Gaschromatograph und Massenspektrometer koppelt.

Abb. 2 Protokoll eines Gaschromatogramms

NR	GEW %	KOMPONENTE	RERET	RET	FLAECHE	TR
20. 10. 70	19.26	5. *P11*SUMPF		00040 N 24.90 0	9 1	48 0
1	0.0015	X	12	.81	1.40700E	3 BB
2	0.4565	PROPYLENOXID	22	1.48	3.14360E	5 BT
3	0.0992	PROPANAL	26	1.73	6.22216E	4 TT
4	0.3270	ACETON	33	2.16	2.43441E5	5 TB
5	0.1210	12.DCLPROPAN	62	4.01	5.85960E	4 BB
6	98.6367	WASSER	100	6.48	9.02649E	7 BT
7	–	EPICLHYDRIN	–	–	–	–
8	0.2547	CL-PROPANOL	201	13.09	1.43037E	5 * TT
9	0.1025	CL-PROPANOL	256	16.62	5.75642E	4 * TB
	100.0000			1.0208	9.11455E	7

Im Massenspektrometer wird die Substanz durch Beschuß mit Elektronen in Bruchstücke zerschlagen und ionisiert. Die Ionen werden mit einigen Kilovolt beschleunigt und in einem Magnetfeld in Ionenteilströme mit gleichem Verhältnis  $m/e$  aufgespalten. Ist das Magnetfeld zeitlich variabel, so treffen die einzelnen Massen seriell auf einem feststehenden Detektor auf. Der Peak mit der höchsten Massenzahl entspricht in der Regel dem einfach ionisierten Molekül und heißt Molpeak.

Man unterscheidet von der Meßgenauigkeit her „niedrigauflösende“ Geräte (Meßgenauigkeit etwa  $1/10$  Masseneinheit) und „hochauflösende“ Geräte (Meßgenauigkeit etwa  $10^{-4}$  Masseneinheiten). Bei Hochauflösung kann die geforderte Genauigkeit nur mit einem Magnetfeld nicht erreicht werden. Der Ionenstrom passiert vielmehr zunächst ein statisches elektrisches Feld, anschließend ein zeitlich veränderliches Magnetfeld. Außerdem wird der zu messenden Probe eine Eichsubstanz Perfluorkerosin (PFK) beigemischt, um Referenzmassen im Spektrum zu erhalten.

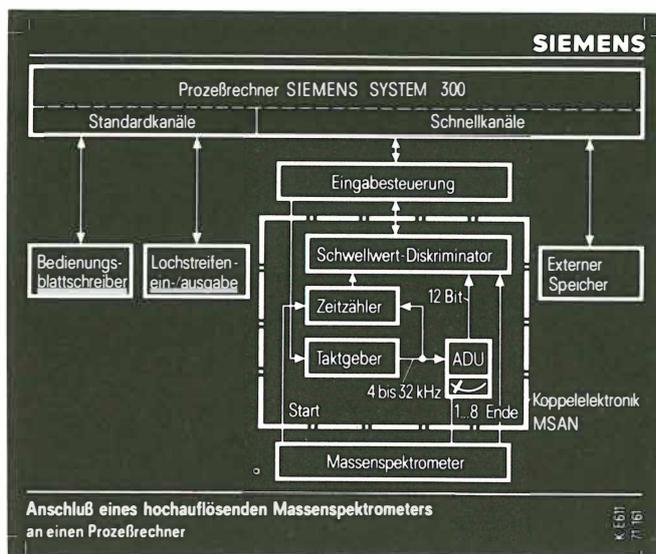


Abb. 3 Anschluß eines hochauflösenden Massenspektrometers an einen Prozeßrechner

Den Anschluß eines hochauflösenden Massenspektrometers an einen Prozeßrechner zeigt Abb. 3. Die Eingabe der Meßparameter erfolgt in diesem Fall über einen Blattschreiber. Die Eingabesteuerung (P3KS) schreibt die Meßwerte kontinuierlich in einen Wechsellpeicher. Von dort transferiert sie das Datenerfassungsprogramm auf den Externspeicher.

Aus den erhaltenen Daten werden die Intensitäten und Lagen der Peaks ermittelt. Die Intensität ist gleich der Fläche des betreffenden Peaks, die Lage ergibt sich aus dem Schwerpunkt. Eine andere Methode, die Bestimmung der Lage eines Peaks durch Aufsuchen des Maximums, liefert ungenaue Ergebnisse, da die Peakform wegen der Ionenstatistik keinen glatten Verlauf hat und ein einzelner Peak mehrere Maxima haben kann.

Der Abszissenwert des Flächenschwerpunktes muß als Gleitkommazahl doppelter Genauigkeit verarbeitet werden, da sonst im Interpolationsprogramm zu große Fehler entstehen. Außerdem ist eine Eigenschaft beim Meßvorgang zu berücksichtigen. Die Messung läuft von hohen zu niedrigen Massen, die Suche der Eichlinien beginnt jedoch zweckmäßigerweise bei kleinen Massen. Die Datensätze mit den Intensitäten und Abszissenwerten werden des-

halb umgedreht. Das Eichprogramm sucht zunächst die PFK-Peaks der Massen 69 und 70, durch Extrapolation findet es die Masse 81 usw. Da der Meßvorgang exponentiell verläuft, erfolgt die weitere Massenbestimmung über die Logarithmen der Massenzahlen.

Zur Interpolation der Probenmassen werden je zwei PFK-Massen oberhalb und unterhalb der Probenmasse herangezogen, die Interpolation erfolgt mit einer Parabel dritter Ordnung. Eine lineare Interpolation, die theoretisch ausreichen müßte, hat sich in der Praxis als ungenügend erwiesen.

Nach dem Aussortieren kleiner Störpeaks, die von PFK herrühren, werden die verbleibenden Peaks auf den größten Peak (Basepeak) normiert und die sogenannte „element map“ erstellt. Dies ist eine Liste, in der zu jeder gemessenen Masse alle möglichen chemischen Zusammensetzungen ausgedrückt werden, die innerhalb eines vorgegebenen Fensters liegen. Abb. 4 zeigt eine element map von Cholesterin ( $C_{27}H_{46}O$ ,  $m=386$ ).

In den einzelnen Spalten sind aufgelistet:

- die relative Intensität
- die gemessene Masse
- die berechnete Masse
- die Differenz in Millimassen
- die Zahl der Atome der genannten Art.

Dem Ausdruck liegt eine Software-Intensitätsschwelle von 15% des Basepeaks und ein maximaler Fehler von  $3.10^{-3}$  Masseneinheiten zugrunde. Bei der Masse  $m=368$  sind zwei Bruttoformeln angegeben, von denen jedoch die mit dem größeren Fehler behaftete falsch ist. Man erkennt, daß der Molpeak und der Basepeak identisch sind. Der große Fehler des Molpeaks rührt von einer schlechten Einstellung des Massenspektrometers her. Anhand dieser Liste kann der Chemiker Aussagen über die Struktur seiner Substanz machen, eine Auswertung über einen Rechner ist in allgemeiner Form heute noch nicht möglich.

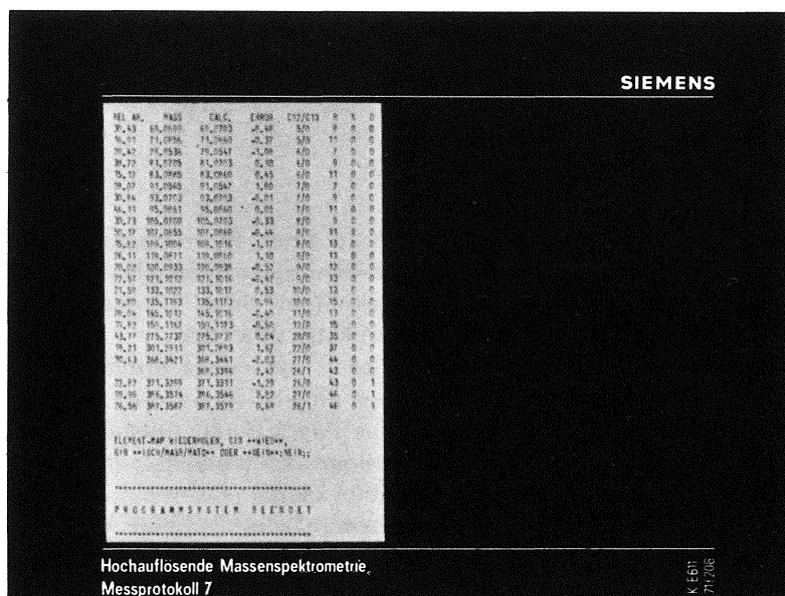


Abb. 4 Element-map von Cholesterin

**W. Lorenz**

SIEMENS AG, Berlin

Externgesteuerte Datenversorgung von Prüfautomaten  
mit SIEMENS-Prozeßrechner 305 im Fg-Prüffeld Berlin

## Externgesteuerte Datenversorgung von Prüfautomaten mit Siemens-Rechner 305 im Fg-Prüffeld Berlin

Im Fg-Prüffeld Berlin wurden bisher Einschübe der Siemens-ESK-Nebenstellentechnik mit einem lochstreifengesteuerten Automaten (ALUP) geprüft.

Die erste Teilaufgabe bei der Prüffeldautomatisierung mit einem Prozeßrechner im Fg-Bereich war die rechnergesteuerte Versorgung der vorhandenen Automaten mit den auf der Platte gespeicherten Prüfdaten im wahlfreien, direkten Zugriff.

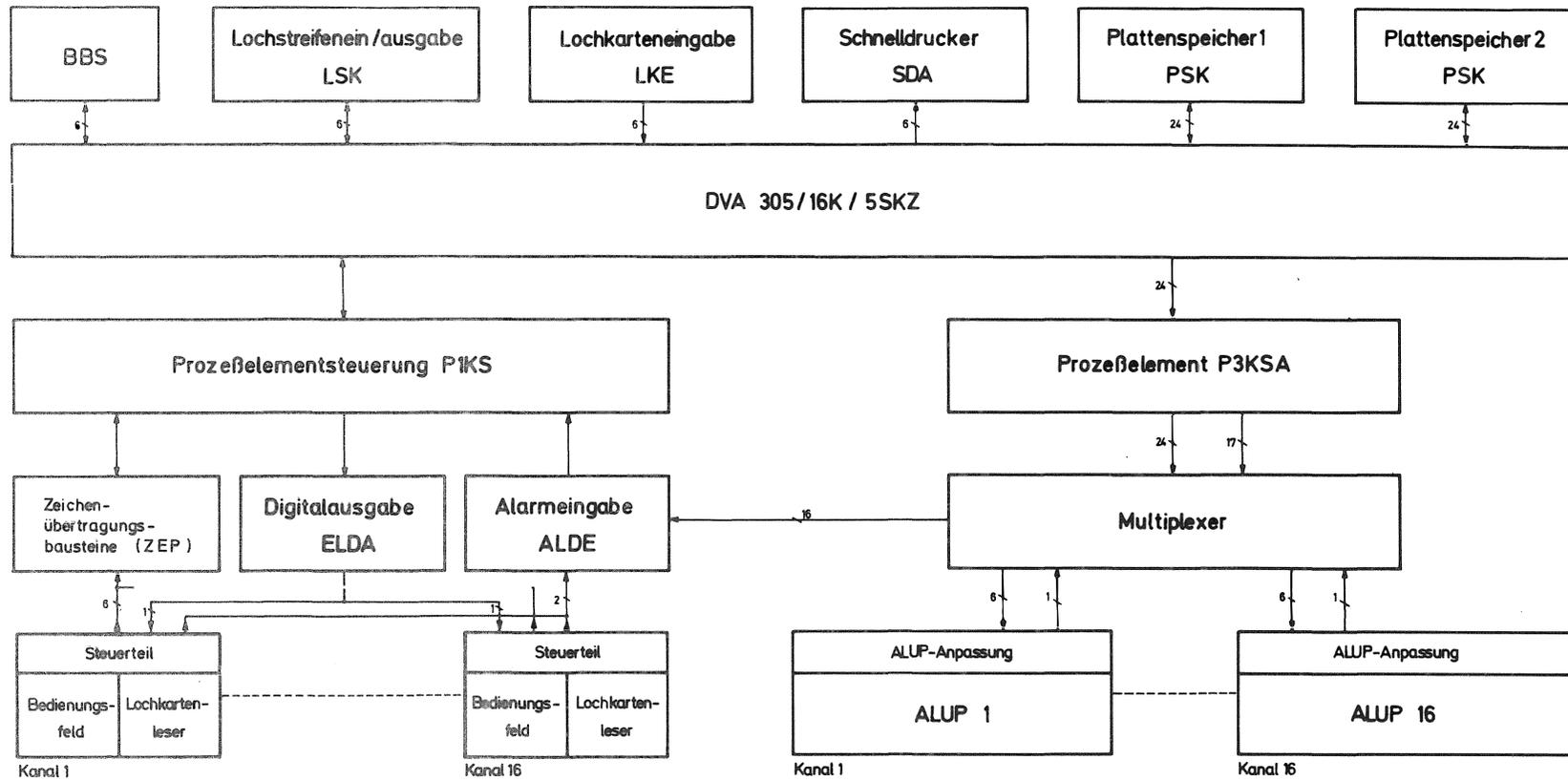
Das hierfür entwickelte Konzept enthält folgende Elemente: Abb. 1

Siemens Prozeßrechner 305 / 16 k / 5 SKZ

Lochstreifenein/ausgabe	LSK
Plattenspeicher	PSK
Prozeßelementsteuerung	P1KS
Zeichenübertragungsbausteine	ZÜB
Alarめingabe	ALDE
Digitalausgabe	ELDA
Prozeßelement	P3KSA
Multiplexer für 16 Kanäle (Eigenentwicklung)	
Lochkartenleser (Forster)	

### 1. Beschreibung des Funktionsablaufs

Die Prüfprogramme werden mit einem Umsetzprogramm auf die Platte geladen. Dabei werden die Daten so aufbereitet, daß ein Festwort (24 Bit) in Stelle 1 – 4 ein Steuerkennzeichen und in Stelle 5 – 24 je 4 ALUP-Zeichen (spezieller ALUP-Code) zu je 5 Bit enthält. Ein ALUP-Prüfprogramm ist im Mittel 2,5 k Festworte lang. Der Name bzw. Suchbegriff für den Aufruf des Prüfprogramms ist eine 8stellige Baunummer. Der Aufruf geschieht mittels Lochkarte über Einzelkartenleser. Nach dem Einlesen der Lochkarte wird vom Anwenderprogramm die Liste der den abgespeicherten Prüfprogrammen zugeordneten Baunummern durchsucht, der erste Teil der Daten in einem Arbeitsspeicher-Pufferbereich bereitgestellt und an der Nahtstelle zum externen Gerät das Übergabebit gesetzt. Falls kein Prüfprogramm unter der angegebenen Baunummer vorhanden ist, erfolgt eine diesbezügliche Meldung über den Bedienungsblattschreiber und eine Anzeige am Prüfplatz. Die Datenausgabe erfolgt externgesteuert über das am Schnellkanal angeschlossene Prozeßelement P3KSA, dessen Ausgangsnahtstelle mit dem nachgeschalteten Multiplexer auf 16 Ausgänge vervielfacht wird. Im Multiplexer werden die Daten außerdem wieder in das Ursprungsformat (5 Bit-Zeichen) umgewandelt. An der Multiplexer-Ausgangsnahtstelle werden der Koppel-elektronik des ALUP-Automaten (DES) ein 5-Bit-Zeichen und ein Übergabebit angeboten. Die Koppel-elektronik gibt nach der Zeichenübernahme ein Quittungsbit an den Multiplexer, mit dem das nächste Zeichen bereitgestellt wird. Die Datenausgabegeschwindigkeit wird vom externen Gerät bestimmt. Ist der Pufferbereich abgearbeitet, dann wird durch einen vom



Prüfdatenversorgung von 16 Prüfautomaten mit Rechner Siemens 305

Steuerzeichen ausgelösten Alarm über das Prozeßelement P1K (das Prozeßelement P3KSA hat keine BAP-Bearbeitung) die nächste Pufferversorgung von der Platte ausgelöst. Das Prüfprogrammende wird auf diese Weise ebenfalls erkannt und beendet die Datenausgabe. Das angestrebte Ziel des vorliegenden Konzeptes war eine minimale Belastung der Rechner-Programmsteuerung, da für die zweite Ausbaustufe des Projektes der eigentliche Schwerpunkt, die rechnergesteuerte Fehlersuche und -Beseitigung, ansteht, die den Rechner relativ stark in Anspruch nehmen wird. Als erstes Ergebnis der Inbetriebnahme ist eine Erhöhung der mittleren Prüfgeschwindigkeit um den Faktor 4,3 zu verzeichnen. Die Zeitdauer von der abschließenden Vorbesprechung über das Projekt bis zum Beginn des Versuchsbetriebs betrug 14 Monate.

## 2. Beschreibung des Multiplexers

Der Multiplexer ermöglicht in Verbindung mit dem Prozeßelement P3KSA (DMA) den direkten Zugriff von 16 externen Geräten zu 16 zugeordneten Pufferbereichen im Arbeitsspeicher. Ein Pufferbereich enthält 256 Festworte. Die Länge ist durch entsprechende Verdrahtung bzw. Schaltereinstellung der Adressdurchschaltung variabel.

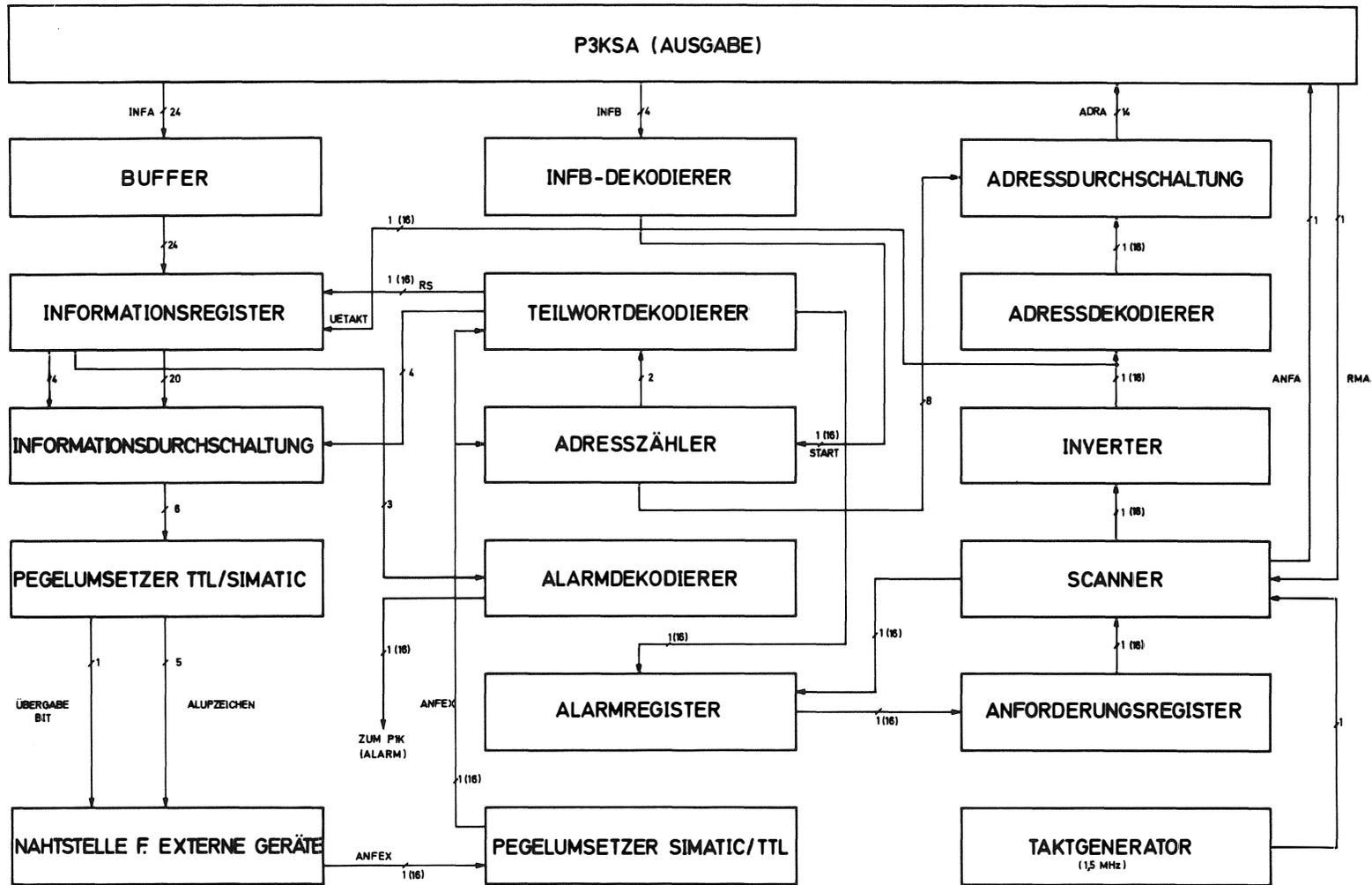
Für jeden Datenausgabekanal ist im Multiplexer ein Adresszähler, ein Informations-Ausgaberegister, ein Teilwortdekodierer und eine Informations- bzw. Teilwortdurchschaltung vorgesehen. Die Bestückung erfolgt entsprechend der gewünschten Zahl der Ausgabekanäle. Letzteres gilt auch für den Pegelumsetzer TTL/Simatic, die Adressdurchschaltung, den Scanner und das Anforderungsregister.

Der Ablauf der Datenausgabe für einen Kanal ist wie folgt (Abb. 2).

Nach der Bereitstellung eines auszugebenden Prüfprogramms wird dem externen Gerät an der Nahtstelle ein Teilwort, bestehend aus Übergabe-Bit und 5-Bit-Zeichen, parallel angeboten. Das Gerät übernimmt das erste 5-Bit-Zeichen und gibt einen Quittungsimpuls (ANFEX) an den Multiplexer. Damit wird das nächste Teilwort durchgeschaltet. Hierbei gelangt das Signal „ANFEX“ über den Pegelumsetzer (Simatic/TTL) zum Adresszähler, der zu Beginn der Datenausgabe rückgesetzt wurde, und erhöht den Zählerstand von 0 auf 1. Am Zählerausgang stehen maximal 14 Bit für die Adressierung von 16 k Festworten zur Verfügung. Hiervon werden die Bitstellen 1 und 2 zur Steuerung der Teilwortdurchschaltung für die jeweils 4 im Informations-Ausgaberegister enthaltenen 6 Bit-Kombinationen mit dem Teilwortdekodierer verwendet.

Die Bitstellen 3 – 10 des Zählerausgangs bilden die laufende Adresse des Pufferbereichs. Die Bitstellen 11 – 14 sind nicht verdrahtet. An der Adressdurchschaltung sind die Bitstellen 1 – 8 mit den Bitstellen 3 – 10 des Adresszählers verbunden, während die Bitstellen 9 – 12 festverdrahtet die relative Anfangsadresse des jeweiligen Pufferbereichs festlegen, der wiederum durch die Bitstellen 13 („0“) und 14 („1“) auf die absolute Anfangsadresse des Gesamtpufferbereichs (8 k – 12 k) fixiert ist.

Nach dem 3. ANFEX-Signal ist das 4. Teilwort bzw. Zeichen durchgeschaltet. Mit der Vorderflanke des 4. ANFEX-Signals wird durch eine Und-Verknüpfung von ANFEX-Signal und Durchschaltesignal (für das 4. Teilwort) das Informations-Ausgaberegister gelöscht. Danach wird mit der Rückflanke von „ANFEX“ der Adresszähler weitergeschaltet. Es erfolgt ein Übertrag in Bitstelle 3, d. h. eine Erhöhung der laufenden Adresse um 1, mit der nun das nächste Informationswort im Arbeitsspeicher angesteuert werden kann. Nach erfolgtem Übertrag sind die Bitstellen 1 und 2 gleich 0 und der Teilwortdekodierer schaltet das 1. Teilwort durch. Mit dem Rücksetzen des Informations-Ausgaberegisters ist aber inzwischen auch die Steuerbitkombination für das Übergabebit gelöscht, so daß die Eingabe weiterer ANFEX-Impulse gesperrt wird. Die Vorderflanke des Durchschaltesignals für Teilwort 1 setzt im Alarmregister ein Bit, das vom Anforderungsregister übernommen wird, damit den Identifizierungsvorgang durch den Scanner auslöst, der mit der Durchschaltung der laufenden



MULTIPLEXER (16 KANÄLE) FÜR EXTERNGESTEUERTE DATENAUSGABE MIT  
SIEMENS RECHNER 305

Adresse (ADRA) mit einer Anforderung auf Datenausgabe (ANFA) zum Prozeßelement P3KSA die Ausgabe des nächsten Informationswortes in das Ausgaberegister (INFA) des P3KSA veranlaßt.

Über den Puffer liegt die Information parallel an den Eingängen aller Register an. Die Übernahme in das Register des identifizierten Kanals geschieht mit der Rückflanke des Adressdurchschaltesignals, das mit dem Signal Rückmeldung nach erfolgter Datenausgabe (RMA) vom P3KSA zum Scanner rückgesetzt wird. Gleichzeitig wird damit auch das Alarmbit gelöscht.

Mit der Übernahme in das Informations-Ausgaberegister wird das Übergabebit an der Nahtstelle gesetzt. Es werden weitere 4 Zeichen ausgegeben und der Vorgang wiederholt sich.

Das letzte Festwort eines Pufferbereichs enthält, wenn eine weitere Puffernachversorgung erfolgen soll, anstelle der Steuerbitkombination 1111 in den Bitstellen 1 – 4 die Kombination 1000. Die Bit 5 – 24 sind gleich 0. Die Steuerbitkombination 1000 wird nach dem Durchschalten des 2. Zeichens zum automatischen Stop der Ausgabe, weil das Übergabebit „0“ wird. Die Steuerbitkombination wird vom Alarmdekodierer erkannt und löst über einen Alarm zum Prozeßelement P1K die Puffernachversorgung aus. Das Ende der Datenausgabe wird in entsprechender Weise mit der Steuerbitkombination 1100 erkannt. Insgesamt können (mit Bit 1 – 3) 8 verschiedene programmgesteuerte Alarmer ausgelöst werden.

Das Rücksetzsignal für den Adresszähler zu Beginn einer Datenausgabe, mit dem gleichzeitig das Alarmbit für die Ausgabe des ersten Festwortes gesetzt wird, wird über den Befehlskanal (INFBE) des P3KSA programmgesteuert als Bitmuster (4 Bit) ausgegeben und vom INFBE-Dekodierer in ein Signal 1 aus 16 umgesetzt. Für die Ausgabe weiterer Signale stehen noch 13 Bit zur Verfügung.

Die Realisierung einer Byte-Nahtstelle zum externen Gerät ist durch einfache Verdrahtungsänderung möglich. Es werden dann jeweils 3 Byte ohne Steuerzeichen oder 2 Byte mit 8-Bit Steuerzeichenkombination ausgegeben.

### *3. Kurzbeschreibung des Anwenderprogramms*

Das Anwenderprogrammsystem ist modular aufgebaut. Es enthält 3 selbständige Programme und insgesamt 16 Unterprogramme.

#### *3.1. Bedienprogramm BEDI*

Mit „BEDI“ werden Daten von Lochstreifen in den Arbeitsspeicher eingelesen, mit der Steuerbitkombination versehen und dann auf der Platte abgespeichert. Die als Suchbegriff verwendete 8stellige Baunummer wird über den Bedienungs-Blattschreiber eingegeben. Mit dem Einlesevorgang wird gleichzeitig eine Baunummernvergleichsliste „BALI“ im Arbeitsspeicher eingerichtet, die außer der Baunummer noch den Anfangsindex der zugehörigen Lochstreifeninformation auf der Platte enthält.

Nach Abschluß des Einlesevorgangs für einen oder mehrere Lochstreifen wird mit dem Codewort „SORT“ die Liste „BALI“ nach aufsteigenden Baunummern sortiert und auf der Platte abgespeichert. Mit dem Codewort „LIST“ kann sie über den Bedienungsblattschreiber für Kontrollzwecke ausgegeben werden. Die sortierte Baunummernliste ermöglicht ein sehr schnelles Durchsuchen der „BALI“. Um den Suchvorgang zusätzlich zu beschleunigen, steht die „BALI“ ständig im Arbeitsspeicher.

Für max. 700 auf Platte geladene Lochstreifen ist sie 2,1 k Festworte lang. Die maximale Länge des Ladepuffers „LAPU“ für das Nachladen (ON-LINE) während des laufenden Datenausgabebetriebs ist 2,8 k. Beim OFF-LINE Laden kann das Lochstreifenprogramm bis zu 6,8 k lang sein. Während des laufenden Datenausgabebetriebs kann der Ladepufferbereich als Laufbereich für Hintergrund-Programme benutzt werden.

K

1	ORG				
2	ORG				
3	ORG	BEDI	INRE	ALBE	DP
4	DP. BALI				
5					
6		DP. LAPU			
7					
8					
9	AUPU 1	2	3	4	
10	5	6	7	8	
11	9	10	11	12	
12	13	14	15	16	
13	UP.				
14	UP.		ORG		
15	ORG				
16	ORG				

Abb. 3

### 3.2. Interruptprogramm „INRE“ für die Alarmauflösung

Es identifiziert die eingehenden Alarme, speichert sie in einem Umlaufpuffer ab und veranlaßt ihre Bearbeitung.

### 3.3. Alarmbearbeitungsprogramm „ALBE“

„ALBE“ wird von „INRE“ gestartet und führt folgende Alarmbearbeitungen aus:

- a) Prüfprogrammaufruf vom externen Gerät, d. h. Lochkarte einlesen, Baunummer suchen, erste Pufferversorgung durchführen, Übergabebit setzen.
- b) Puffernachversorgung
- c) Programmende oder -abbruch
- d) Programmwiederholung ohne Einlesen einer neuen Lochkarte

Diese Bearbeitungsliste kann bei Bedarf entsprechend erweitert werden.

### 3.4. Arbeitsspeicherbelegung

Die verwendete Belegung des ASP zeigt Abb. 3.

**W. Kammerer**

Siemens, München

Das Integrierte Prüfsystem 70  
(Systemphilosophie, Prüfautomaten, Anwendung)

## **Das Integrierte Prüfsystem 70 (Systemphilosophie, Prüfautomaten, Anwendung)**

*Referat für die Tagung des SIEMENS-Prozeßrechner-Anwenderkreises I, Institute und Ausbildungsstätten, im Hahn-Meitner-Institut für Kernforschung Berlin GmbH am 24. – 27. Mai 1971*

### *1. Einleitung*

Das Integrierte Prüfsystem 70 entstand in den letzten 6 Jahren für die Prüfung von vorwiegend elektrischen Geräten in der industriellen Fertigung. Der Einsatz von Prozeßrechnern im Prüfgeschehen der Serienfertigung war damals neu.

Das unbekannte Gebiet bedurfte einer systematischen Analyse und neuer Lösungswege. Das Integrierte Prüfsystem 70 entstand in der fertigungstechnischen Entwicklung des Hauses Siemens, es wurde durch wertvolle Mithilfe diverser Prüf- und Entwicklungsstellen unseres Hauses sehr bereichert. Seit ungefähr 2 Jahren wurde es in das Programm der Meß- und Prozeßtechnik, Siemens Karlsruhe, aufgenommen.

Das System wird laufend ergänzt und ausgebaut. In den letzten beiden Jahren sind insgesamt 26 solcher Automaten bzw. Prüfplätze in unserem Hause zum Einsatz gekommen.

Das System basiert auf einem großen Erfahrungsschatz und hat die Phase der Einführung bzw. praktischen Bewährung bereits überstanden. Es befindet sich z. Z. in der Phase der breiten Anwendung.

Das Prüfsystem berücksichtigt nicht nur die Prüfautomaten selbst, sondern auch die möglichst maschinelle Erzeugung der Prüfdaten, die Adaptierung der Prüflinge und nicht zuletzt zeigt es auch die Grundsätze für einen prüfgerechten Aufbau bzw. prüffreundliche Konstruktion des zu prüfenden Gerätes auf.

### *2. Die praktische Anwendung des Prüfsystems und die Anforderungen an dieses*

Stellen wir uns das Innere eines Rechners vor, so erkennen wir, daß in den ersten Phasen des Fertigungsprozesses folgendes geprüft werden muß:

- die Schaltungsträger der Flachbaugruppen und die Rückwandverdrahtungen,
- die komplett bestückte Flachbaugruppe.

Mit diesem praktischen Hintergrund möchte ich Ihnen das Prüfsystem, seine Philosophie, seinen Aufbau und seine Leistungen vorstellen.

#### *2. 1 Die Verdrahtungsprüfung*

Abb. 1 ZFA 35 293

zeigt eine Multilayerplatte ca. 50 x 50 cm, sie besitzt 15 800 zu prüfende Anschlußpunkte. Die Platte muß von der Vorderseite und von der Rückseite überlappend geprüft werden, das sind insgesamt 40 000 Anschlußpunkte.

In der Minute werden 3000 Anschlußpunkte geprüft, die Prüfzeit beträgt 13 Minuten. Pro Anschlußpunkt muß der Rechner ein 24-bit-Wort ausgeben. Das Prüfdatum für eine Messung beträgt 24 bit in der Grundsprache des Rechners, in der einzugebenden Automaten-sprache, die besser zu lesen und zu schreiben ist, 8 Zeichen. Die gesamte Informationsmenge für diese Platte beträgt 320 000 Zeichen oder 800 m Lochstreifen. Die partielle Adaptierung für diese Leiterplatte zeigt Abb. 2.

Die Platte besitzt eine orthogonale Leitungsführung. Die waagrechten und senkrechten Leiterzüge werden hier in Teilschritten getrennt geprüft. Der Nadelträger ist geschwenkt, er besitzt  $10 \times 132 = 1320$  Nadeln.

Anders ist der Fall, wenn keine orthogonale Leitungsführung vorhanden ist. So z. B. bei den Schaltungsträgern für Flachbaugruppen. Hier muß der Schaltungsträger komplett adaptiert werden.

Abb. 3 zeigt den Nadelträger mit ca. 1000 Kontaktnadeln für eine einfache Flachbaugruppe in Europaformat.

Den kompletten Automaten zeigt Abb. 4

Links die Blattschreibernausgabe. Der linke Schrank enthält die Zentraleinheit des Prüfautomaten mit DVA 101, die Lochstreifeneingabe, das Bedienungsfeld und einige periphere Prüfbausteine. Daran schließt sich der Relaismatrixschrank an. Er ist bis 4096 Anschlußpunkte ausbaufähig. Max. 16 solcher Schränke können je Automat angeschlossen und damit max. 64 000 Anschlußpunkte geprüft werden.

Verdrahtungsprüfung heißt:

Durchgangsprüfung mit bis 3 A Meßstrom, Messung des Durchgangswiderstandes; einstellbare Meßschwelle 1...9 Ohm.

Isolationsmessung mit max. 500 V Meßspannung, Messung des Isolationswiderstandes einer Verbindung gegen alle anderen.

Einstellbare Meßschwelle zwischen 1 und 100 MOhm.

Die Zeit für eine Messung dauert 20 ms, es genügt, daß der Rechner innerhalb 1 ms ein Prüfdatum = 24 bit an die Prüfperipherie ausgibt. Sie erkennen daran, daß die Anforderungen an den Prozeßrechner klein sind, dagegen wird ein preislich billiger Rechner verlangt, weshalb die DVA 101 eingesetzt ist. Neuerdings werden die Prüfdaten maschinell auf der größeren Anlage 4004 aus den Schaltungsunterlagen generiert. Ausgegeben werden die Prüfdaten in Lochstreifen bzw. auf Magnetband. Deshalb muß der Verdrahtungsprüfautomat diese Datenträger, z. B. die der 4004, lesen können.

## 2. 2 Logikprüfung

Die Funktionsprüfungen bestückter Flachbaugruppen können verständlicherweise nicht über einen Kamm geschoren werden, daher werden

- sog. Logikflachbaugruppen mit nur digitalen Schaltkreisen, meist in integrierter Technik
- Flachbaugruppen vorwiegend mit analogem Inhalt und Sinustechnik für Nachrichten- und Übertragungstechnik
- gemischte Flachbaugruppen, für die noch nicht eine übergeordnete technische Ordnung gefunden werden konnte

unterschieden.

Abb. 5 zeigt die Ansicht einer Logikflachbaugruppe. Sie enthält beispielsweise 25 IC.

Die Prüfung dieser Baugruppe besteht aus Aneinanderreihen dynamischer Parallelmessungen. An alle programmierbaren Eingangsstifte des Prüflings wird (über Pegelumsätze) ein digitales Bitmuster angelegt und das jeweilige Bitmuster an Ausgangsstiften automatisch mittels Komparatoren überprüft. Die Informationsmenge der einzugebenden Prüfdaten beträgt im Durchschnitt 3000 Worte je 24 bit.

Abb. 6 zeigt den Logikprüfautomat LOG 10/IP 70

Der linke Schrank enthält den Rechner 301, daran schließen sich die Schränke und das Pult mit den peripheren Prüfbausteinen an. Das Pult enthält die pro Anschlußpunkt vorgesehenen Pegelumsetzer und Komparatoren sowie das Bedienungsfeld. Die Ein- und Ausgabe fehlt auf diesem Bild.

Es wird gefordert, daß ein fehlerfreier Prüfling in 1...2 Sekunden geprüft wird. Dies bedeutet, daß der Rechner spätestens alle 100  $\mu$ s ein Prüfwort ausgeben muß.

Auch hier würde zunächst ein Kleinrechner genügen, doch die umfangreichere Standardperipherie (z. B. Platte) verlangt den Einsatz eines mittleren Rechners wie in diesem Fall die DVA 301, es sind aber auch schon Rechner 304 und 305 für solche Logikprüfungen eingesetzt. Der Grund liegt daran, weil man z. Z. versucht, im Fehlerfall eine sog. automatische Fehlersuche per Software zu realisieren, die den Fehler einigermaßen auf der Flachbaugruppe lokalisieren soll.

### *2.3 Pegelmeßautomaten für Weitverkehrs- und Richtfunktechnik*

Zur Übertragung von z. B. Ferngespräche über größere Entfernungen wird die Tonfrequenz höheren Frequenzen überlagert. Ich meine hier das Gebiet der Weitverkehrs- und der Richtfunktechnik. Für die Bausteine dieser Überlagerungstechnik fallen in der Hauptsache Pegelmessungen an.

Abb. 7 zeigt den entsprechenden Prüfplatz. Dieser ist als Teil eines schon vorhandenen Automatenetzes in den Siemens-Prüffeldern für Weitverkehrstechnik hier in Berlin entstanden. Als Kleinrechner wurde die 101 verwendet, die 3 unabhängige solcher Prüfplätze steuert. Das Bild zeigt den bausteinmäßigen Aufbau. Die einzelnen Bausteine sind

- steuerbare Generatoren (Pegelsender)
- steuerbare Abstecker- bzw. Eichleitungen
- steuerbare Schalter
- steuerbare Pegelmesser
- steuerbare Frequenzzähler

und nicht zuletzt das Bedienungsfeld.

Gemessen wird bis zu einer Genauigkeit von 1  $\text{‰}$  bei einem Dynamikbereich von ca. 100 dB. Oberer Frequenzbereich ca. 100 MHz. Die Arbeitsgeschwindigkeit beträgt ca. 1,3 Sek. je Meß- bzw. Prüfschritt. Wegen der Einschwingvorgänge in den Meßkreisen muß im Schnitt ca. 1 Sek. gewartet werden, bevor nach den Einstellanweisungen gemessen wird.

Das Prüfen solcher Geräte erfordert oft mehr als 100 Meßvorgänge je Prüfling. Zu messen sind vorwiegend:

Gleich- und Wechselspannung, Frequenz, Spannungs- und Leistungspegel, Betriebsumfang, Reflektions-, Nebensprech-, Symmetrie- und Klirrdämpfung, Dämpfungs- und Zeichenverzerrung.

### *3. Grundgedanken für den Einsatz des Prozeßrechners als Prüfrechner*

3.1 Da die Prüfvorschrift bzw. die Prüfdaten sequentiell vom Datenträger bzw. aus dem Arbeitsspeicher gelesen werden, müssen diese Anweisungen pro Prüfbaustein gespeichert werden (Abb. 8).

Die Anschaltung dieser Flipflopspeicher an den Rechner erfolgt über einen Datenbus. Wir nennen diese Nahtstelle „Nahtstelle für periphere Prüfbausteine“ (abgekürzt PBS-Nahtstelle). Sie enthält einen Eingabe-, Ausgabe- und Adreßkanal. Der Rechner wird unabhängig von der Zahl der angeschlossenen peripheren Prüfbausteine, die Adernzahl ist konstant und ein bausteinmäßiger Aufbau daher leichter möglich.

Es ist unbedingt notwendig, daß die Adresse für den Prüfbaustein zusammen mit der dazugehörigen Information (z. B. Einstellwert für den Spannungsgeber) in einem 24-Bit-Wort untergebracht werden müssen. Nur so ist ein blockweiser Verkehr mit dem Prozeßrechner möglich; nur so können die Meßanweisungen in der maximal möglichen Geschwindigkeit bei direktem Verkehr mit dem Arbeitsspeicher ausgegeben werden. Langsamere Messungen erlauben einen wortweisen Betrieb, d. h. bei der Ausgabe bzw. Eingabe jedes Wortes ist das Rechenprogramm eingeschaltet. Der letztere Fall ist aus Kostengründen interessant, weil nun das Anpaßelement zwischen Peripherie des Prüfplatzes und Zentraleinheit des Rechners

nun nicht mehr die Adreßarithmetik übernehmen muß. Dadurch wird das Anpaßelement, d. h. das Interface, wir sprechen hier vom Prozeßelement für numerische Steuerung, abgekürzt NC..K, billiger. Start-Stopp-Signale der PBS-Nahtstelle erlauben eine Zeitentkopplung zwischen Rechner und Peripherie.

Die PBS-Nahtstelle grenzt den Prüfrechner mit seinem Anpaßelement auf der einen Seite ab. Damit die gesamte Zentraleinheit des Prüfautomaten als „Black Box“ erscheint und das Prüfgeschehen unabhängig vom verwendeten Rechner wird, wird der Rechner mit einer 2. Nahtstelle, der Ein- und Ausgabenahtstelle, abgeschlossen.

Das Integrierte Prüfsystem 70 ist so aufgebaut, daß zwischen diesen beiden Nahtstellen der Rechner mit seinem Anpaßelement und seiner Standardperipherie je nach den Anforderungen an das Prüfgeschehen ausgewählt und unabhängig von der Prüfperipherie eingesetzt werden kann.

Die normierte Ein- und Ausgabe der Prüfdaten bzw. Prüfergebnisse bringt die Unabhängigkeit von individuellen Standardein- und -ausgabeelementen des verwendeten Rechners. Für jeden einzustellenden Prüfbaustein werden eine Adresse und eine Information benötigt. Verschiedene Erhebungen zeigten, daß durchschnittlich 3–4 Ziffern, also 12–16 bit als Information bei tetradischer Darstellung genügen.

Dies führte zur Teilung des im Rechner der Serie 300 vorhandenen Maschinenwortes von 24 bit in 6-bit-Adresse und 18-bit-Information. Wir definieren hier eine Eingabenahtstelle in Grundsprache bzw. Automaten-sprache, letztere ist besser schreib- und lesbar.

Diese Normierung deckt sich mit der PBS-Nahtstelle, d. h. im einfachsten Fall können die Prüfdaten sequentiell im Arbeitsspeicher gespeichert werden und von diesem wieder sequentiell ohne Umformungsroutinen ausgegeben werden. Für viele Prüfungen im fertigungstechnischen Prozeß muß dies gefordert werden. Aus allen diesen Gründen waren wir gezwungen, besondere, rechnernahe Prozeßelemente zur Anpassung der Rechnernahtstelle an die PBS-Nahtstelle zu erstellen. Einige solcher NC-Elemente sind bereits lieferbar, andere noch in Entwicklung.

Abb. 9 zeigt das Element NC1K für wortweisen Verkehr. Es setzt die AKZ-Nahtstelle der DVA 301 in die PBS-Nahtstelle um.

Abb. 10 zeigt die Übersicht der NC-Elemente, geordnet nach wort- und blockweisen Verkehr. Kennzeichnende Daten sind die Datenrate über die PBS-Nahtstelle, die Reaktionszeit und der Kostenfaktor.

Diese Elemente für blockweisen Betrieb haben einen höheren Kostenfaktor, sie leisten dafür auch mehr.

Damit die Hardwarebausteine als Prüfautomaten zusammenwirken, muß der Rechner programmiert werden. Wir unterscheiden

- Programme, die rein prüflingsorientiert sind, also das Prüfprogramm bzw. die Prüf Sprachendarstellung und
- „Prozeßführungsprogramme“, die weitgehend unabhängig vom Prüfling sind, also z. B. Störungsaufgaben sowie Ein- und Ausgabefunktionen übernehmen.

Letztgenannte Programme werden in Systemen mit der Bezeichnung BIPROF (Betriebssystem für Prüfautomaten im Off-Line-Betrieb) und BIPRON (Betriebssystem für Prüfautomaten im On-Line-Betrieb) zusammengefaßt. Sie sind aus Teilprogrammen bausteinartig aufgebaut, bestimmte Teilprogramme enthalten prüflingsgebundene Passagen, z. B. für die Meßwertverarbeitung. Es würde zu weit gehen, auf den Programmaufbau weiter einzugehen. Den Prüfrechner mit seiner Standardperipherie, seinem NC-Element und dem BIPROF bezeichnen wir als Zentraleinheit des Prüfautomaten. Diese Zentraleinheiten sind normiert und geben dem Prüfautomatenentwickler, soweit er speziellere Automaten selber entwickeln und keine Standardprüfautomaten verwenden kann, eine beachtliche Vorleistung an Entwicklungskosten. Man bekommt damit auch für speziellere Automaten dieselbe technische

und programmierte Struktur, wie bei den Standardprüfverfahren. Das erleichtert sehr das Arbeiten mit den Prüfautomaten, die Wartung und schließlich auch den organisatorischen Einbau des Automaten in das Betriebs- bzw. Prüfgeschehen.

Solche Faktoren sind oft viel wichtiger als eine ständige Verbesserung und damit Änderung der technischen und programmierten Strukturen.

#### *4. Einbeziehung des Prüfprogramms in das Datenverarbeitungssystem des Betriebes*

Abb. 11 zeigt das in einer Rechnerhierarchie eingegliederte Integrierte Prüfsystem 70 mit folgenden Automatisierungsstufen:

##### *Automatisierungsstufe 1*

bedeutet Prüffeldbetrieb mit automatischer, rechnergestützter Prüfabwicklung wie vordem aufgezeigt.

##### *Automatisierungsstufe 2*

bedeutet die Eingliederung des Prüfprozesses in den Fertigungsprozeß. Die im allgemeinen größere Leistungsfähigkeit des FLR kommt dem Prüfprozeß z. B. bei der automatischen Fehlersuche, bei Abgleichvorgängen, bei der integrierten Qualitätsregelung, zugute.

##### *Automatisierungsstufe 3*

bedeutet das Einbeziehen des Fertigungsprozesses einschließlich Prüfprozeß in den Betriebsprozeß, der vom Betriebsrechner geführt wird.

#### *5. Schlußbemerkungen*

Wenn auch bis jetzt das Integrierte Prüfsystem 70 vorwiegend für die Prüfung moderner elektronischer Geräte eingesetzt ist, so gilt es auch für die übrige Elektrotechnik.

Neben der Wirtschaftlichkeit ist es im wesentlichen eine Frage der analog-digitalen bzw. digital-analogen Umwandlungsprozesse. Bei der Messung bzw. Prüfung nichtelektrischer Größen bedarf es der Umwandlung dieser in elektrische. Schließlich zeichnen sich auch Einsatzmöglichkeiten in Entwicklungs- bzw. Erprobungsstellen ab, in welchen die Prinzipien und Geräte des IP 70 zu Dauerversuchseinrichtungen und ähnlichen benutzt werden können.

In Zukunft wird daher der Prozeßrechnereinsatz auch im Prüfgeschehen sehr zunehmen, und die Prüfanforderungen steigen, weil das Leistungs-/Preisverhältnis dieser immer günstiger wird. Entsprechend ausgebildetes Personal muß hierzu bereitstehen. Bei heute noch etwas fraglichen Objekten sollte an dieses gedacht werden.

Die Abbildungen zu diesem Artikel können auf den folgenden Seiten eingeklebt werden.

