



KERNFORSCHUNGSANLAGE JÜLICH
GESELLSCHAFT MIT BESCHRÄNKTER HAFTUNG

Siemens - Prozeßrechner - Benutzertagung

veranstaltet vom

Zentrallabor für Elektronik / Abt. Nukleare Elektronik

und On - Line - Datenverarbeitung

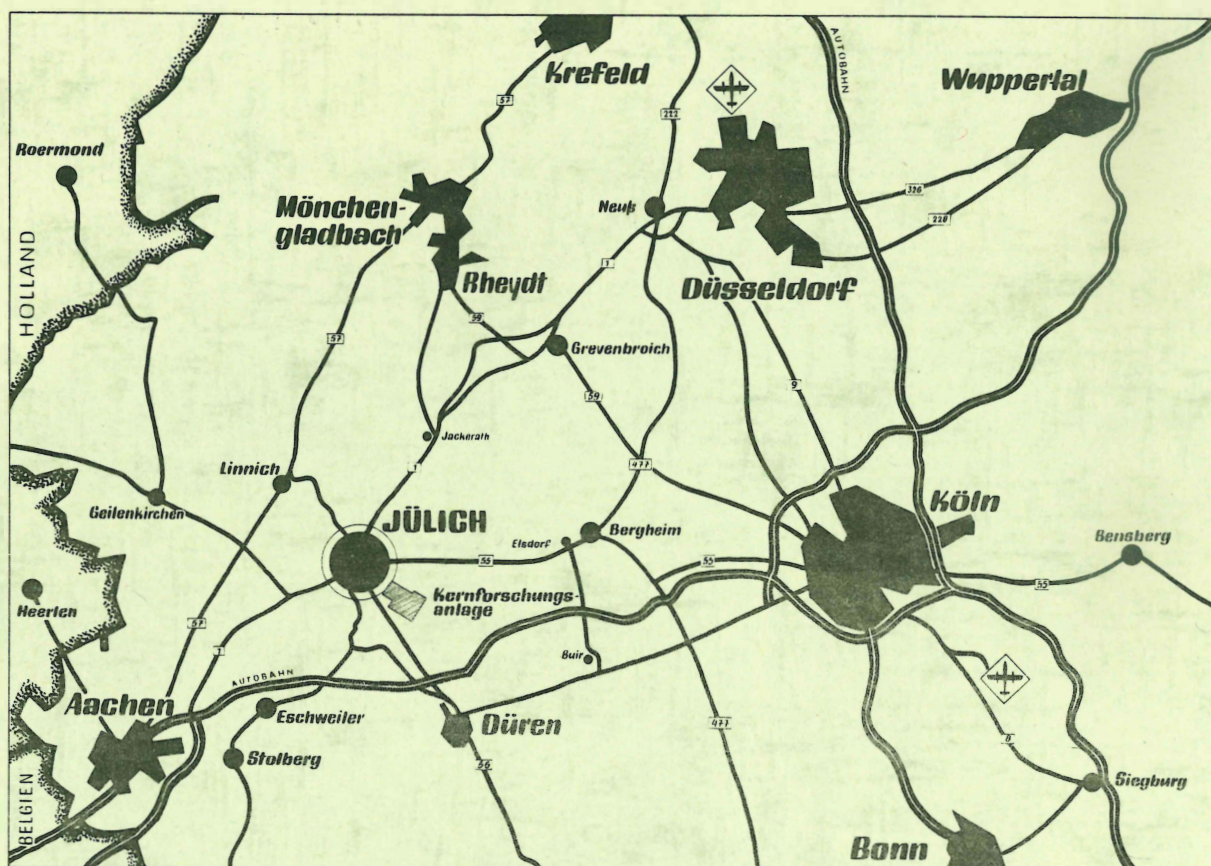
vom 29. Juni bis 1. Juli 1970

Herausgeber

Klaus-Dieter Müller, Walter Tenten

Jül - Conf - 5
März 1971

Als Manuskript gedruckt



Berichte der Kernforschungsanlage Jülich – Jül - Conf - 5

Dok.: Conferences - Germany (Federal Republic) / 1970
Computers - Process Control

Im Tausch zu beziehen durch: ZENTRALBIBLIOTHEK der Kernforschungsanlage Jülich GmbH,
Jülich, Bundesrepublik Deutschland

Siemens - Prozeßrechner - Benutzertagung

veranstaltet vom

**Zentrallabor für Elektronik / Abt. Nukleare Elektronik
und On - Line - Datenverarbeitung**

vom 29. Juni bis 1. Juli 1970

Herausgeber

Klaus-Dieter Müller, Walter Tenten

Vorwort:

Seit Anfang 1969 besitzt das Zentrallabor für Elektronik der Kernforschungsanlage Jülich GmbH. einen Siemens - Prozeßrechner 305. Dieser Rechner wird im wesentlichen für die Datenerfassung und -reduzierung bei kernphysikalischen Experimenten eingesetzt. In diesem Zusammenhang zeigte sich immer klarer, daß ein Erfahrungsaustausch mit anderen Benutzern von Siemens Prozeßrechnern wünschenswert ist, um Doppelarbeit zu vermeiden und einen Programmaustausch einzuleiten.

Deshalb fand mit Unterstützung des Hauses Siemens vom 29.6. bis 1.7.1970 in Jülich eine erstmalige Zusammenkunft statt. An der Tagung nahmen ca. 50 Benutzer von Siemens-Prozeßrechnern, vornehmlich aus dem technisch-wissenschaftlichen Bereich, aus allen Teilen der Bundesrepublik teil. Ziel des Treffens war ein Erfahrungsaustausch über Hardware- und Software-Probleme in Form von Vorträgen und Diskussionen zwischen den verschiedenen Benutzern und dem Hause Siemens.

Unter anderem wurde die Gründung einer offiziellen Benutzer-Organisation diskutiert; entsprechende Verhandlungen wurden eingeleitet, die jedoch noch nicht zu einem konkreten Ergebnis geführt haben.

Der vorliegende Bericht enthält die in Jülich gehaltenen Vorträge, soweit sie uns von den Vortragenden zur Verfügung gestellt wurden. Dafür sei allen Autoren an dieser Stelle nochmals herzlich gedankt.

Wir hoffen, daß dieser Bericht möglichst vielen Prozeßrechner-Benutzern bei ihren weiteren Arbeiten von Nutzen sein wird.

Die Herausgeber: Klaus Dieter Müller

Walter Tenten

Zentrallabor für Elektronik/
Abt. Nukleare Elektronik und
On-Line-Datenverarbeitung

Program

=====

Montag, den 29. Juni

Der Vormittag stand zur Anreise der Teilnehmer zur Verfügung

13,30 Uhr

Begrüßung und Organisatorisches

14,00 Uhr

H. Zander

(Fa. Siemens AG, Karlsruhe)

Das Modell 306 - Eine neue Datenverarbeitungsanlage des Systems 300

15,00 Uhr - 15,20 Uhr

Kaffeepause

15,20 Uhr

Verschiedene Vorträge über Hardware-Entwicklungen von Benutzern:

1. H. Pangritz

(Hahn-Meitner-Institut, Berlin)

Eine digitale Uhr zum Anschluß an eine Standard-nahtstelle der S 305

2. R. Schmitt

(Fa. Messerschmitt-Bölkow, München)

Der Kanalumsetzer KUS- eine schnelle Ein-Ausgabe-Einheit für einen Siemens 300 Rechner mit Schnellkanalzusatz

3. H.-J. Schuster,

(Phys.-Techn. Bundesanstalt, Braunschweig)

Experimentgeordnete und externgesteuerte Datenübertragung beim Prozeßrechner Siemens 305

4. H. Englmeier

(Max-Planck-Institut für Eiweiß- u. Lederforschung, München)

Ein externes Leitwerk als Interface zwischen Meßgerät und Rechenmaschine

5. H. Kevecordes
(Ingenieurschule Paderborn)
Hardware Pläne eines möglichst optimalen Prozeß-
rechnereinsatzes an der Ingenieurschule Pader-
born

6. W. Tenten
(KFA Jülich, Zentrallabor für Elektronik)
Inkrementierungen bei Siemens - Prozeßrechnern

18,00 Uhr

geplantes Ende der Tagung an diesem Tag

Dienstag, den 30. Juni

9,00 Uhr

H. Schlimbach
(Siemens AG, Karlsruhe)
Die neuen Standard-Ein-/Ausgabe-Aufrufe für das ORG
300

10,00 Uhr

R. Schmitt
(Fa. Messerschmitt-Bölkow, München)
Eigene Programme und Systemprogramme für einen Rechen-
zentrumsbetrieb mit der Siemens 305

10,20 Uhr - 10,40 Uhr

Kaffeepause

10,40 Uhr

J. Radünz
(Deutsche Forschungs- u. Versuchsanstalt für Luft-
u. Raumfahrt e.V., Oberpfaffenhofen)
Betriebssystem für den deutschen Forschungssatelliten
AZUR

K. Weise
(Phys.-Techn. Bundesanstalt, Braunschweig)
Ein Programmsystem für die Meßdatenerfassung und
-verarbeitung bei Experimenten am Forschungs- und Meß-
reaktor der Physikalisch-Technischen Bundesanstalt

Chr. Meixner
(KFA Jülich, Zentralinstitut für Reaktorexperimente)
On-Line-Auswertung von Gammaskpektren an Spülkreis-
läufen

18,00 Uhr

Einladung zum Kalten Buffet im Wald-Casino der KFA
Jülich

Mittwoch, den 1. Juli

9,00 Uhr

H. Teuschler
(Fa. Siemens AG, Karlsruhe)
Bericht über den neuen FORTRAN-H-Compiler für das
System 300

1. L. Radermacher
(KFA Jülich, Zentrallabor für Elektronik)
Hardware - Rechnerkopplung Siemens 305 - PDP 8
2. F. Rongen
(KFA Jülich, Zentrallabor für Elektronik)
Dialogverkehr bei der Rechnerkopplung Siemens 305 -
PDP 8

11,00 Uhr - 11,20 Uhr

Kaffeepause

11,20 Uhr

Product - Panel

13,00 Uhr

Mittagspause

14,15 Uhr

K. Bindewald
(Fa. Siemens AG, Karlsruhe)
Sichtgeräte für Prozeßrechner des Siemens Systems 300

W. Ehrenberger
(Technische Hochschule, München)
Alarmbearbeitung in einer 305 und Simultanarbeit in
höheren Sprachen

J. Radünz

(Deutsche Forschungs- u. Versuchsanstalt für Luft-
u. Raumfahrt e.V., Oberpfaffenhofen)

Betriebssystem ORG I - 305 GCC für Stapelverarbeitung

P. Abend

(Hahn-Meitner-Institut, Berlin)

Ein Teil-Programmsystem zur Auswertung kernphysika-
lischer Experimentdaten mit dem Prozeßrechner Siemens
305

13,00 Uhr

Mittagspause

14,15 Uhr

H. Schlimbach

(Fa. Siemens AG, Karlsruhe)

Mehrfachzugriffssysteme

15,10 Uhr - 15,30 Uhr

Kaffeepause

15,30 Uhr,

J. Becker

(Staatliche Ingenieurschule, Paderborn)

Software-Erfahrungen beim Einsatz der 303 an der
Staatlichen Ingenieurschule Paderborn

H. Langemack

(Technische Universität, Braunschweig)

Software für einen Digital - Plotter

W. Tenten

(KFA Jülich, Zentrallabor für Elektronik)

PAL-Assembler 305/8

K.D. Reinartz

(Inst. f. Kristallographie, Techn.Hochschule Aachen)

Syntax von ROSA, Programm mit symbolischen Adressen
(PROSA-Untermenge)

15,30 Uhr

Diskussion über die Gründung einer Benutzer -
Organisation

17,00 Uhr

Ende der Tagung

Inhaltsverzeichnis:

=====

Herr H. Pangritz, Hahn-Meitner Institut, Berlin

"Eine digitale Uhr zum Anschluß an einen Standardkanal der S 305" - S. 0

Herr R. Schmitt, Fa. Messerschmitt-Bölkow, München

"Der Kanalumsetzer KUS, eine schnelle Ein-Ausgabe-Einheit für einen Siemens 300 Rechner mit Schnellkanalzusatz" - S. 5

Herr H.-J. Schuster, Physikalisch-Technische Bundesanstalt, Braunschweig

"Experimentgeordnete und externgesteuerte Datenübertragung beim Prozeßrechner Siemens 305" - S. 24

Herr H. Englmeier, Max-Planck-Institut für Eiweiß- und Lederforschung, München

"Ein externes Leitwerk als Interface zwischen Meßgerät und Rechenmaschine" - S. 33 -

Herr Kevecordes, Ingenieurschule Paderborn

"Hardware Pläne eines möglichst optimalen Prozeßrechnereinsatzes an der Ingenieurschule Paderborn" - S. 48

Herr W. Tenten, ZEL/NE der KFA Jülich

"Inkrementierungen bei Siemens-Prozessrechnern" - S. 54

Herr R. Schmitt, Fa. Messerschmitt-Bölkow, München

"Eigene Programme und Systemprogramme für einen Rechenzentrumsbetrieb mit der Siemens 305" - S. 63

Herr J. Radünz, Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt e.V., Oberpfaffenhofen

"Betriebssystem für den deutschen Forschungssatelliten AZUR" - S. 76

Herr K. Weise, Physikalisch-Technische Bundesanstalt, Braunschweig

"Ein Programmsystem für die Meßdatenerfassung und -verarbeitung bei Experimenten am Forschungs- und Meßreaktor der Physikalisch-Technischen Bundesanstalt" - S. 103

Herr J. Radünz, Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt e.V., Oberpfaffenhofen

"Betriebssystem ORG I - 305 GCC für Stapelverarbeitung" - S. 111

Herr J. Becker, Staatliche Ingenieurschule Paderborn

"Software-Erfahrungen beim Einsatz der 303 an der Staatlichen Ingenieurschule Paderborn" - S. 121

Herr Langemack, Technische Universität Braunschweig

"Software für einen Digital - Plotter" - S. 124

Herr W. Tenten, ZEL/NE der KFA Jülich

"PAL-Assembler 305/8" - S. 137

Herr K.D. Reinartz, Institut für Kristallographie, Technische Hochschule Aachen

"Syntax von ROSA, Programm mit symbolischen Adressen (PROSA-Untermenge)" - S. 143

Herr Chr. Meixner, ZIR der KFA Jülich

"On-Line Auswertung von Gammaspektren an Spülkreisläufen" - S. 148

Herr Teuschler, Siemens AG Karlsruhe

"Bericht über den neuen FORTRAN-H-Compiler für das System 300" - S. 153

Herr L. Radermacher, ZEL/NE der KFA Jülich

"Hardware - Rechnerkopplung Siemens 305 - PDP 8" - S. 160

Herr F. Rongen, ZEL/NE der KFA Jülich

"Dialogverkehr bei der Rechnerkopplung Siemens 305 - PDP 8" - S. 173

Herr K. Bindewald, Siemens AG Karlsruhe

"Sichtgeräte für Prozeßrechner des Siemens Systems 300" - S. 181

Herr F. Gaffal, Technische Hochschule München

"Eine 305 als digitaler Partner in einem Hybridsystem" - S. 187

Herr W. Ehrenberger, Technische Hochschule München

"Alarmbearbeitung in einer 305 und Simultanarbeit in höheren Sprachen"

- S. 214

Heiko Pangritz

Hahn-Meitner-Institut

Berlin

"Eine digitale Uhr zum Anschluß
an einen Standardkanal der S 305 "

Eine digitale Uhr zum Anschluß an einen Standardkanal der S 305

1. Zeitbasis

Die Zeitbasis der digitalen Uhr ist die Netzfrequenz, die eine hohe integrale Genauigkeit in einem längeren Zeitraum aufweist. Um jedoch auch bei Netzausfall die Zeitzählung beibehalten zu können, ist ein quarzgesteuerter Oszillator vorgesehen, der auch bei geringstem Schaltungsaufwand über relativ kurze Zeiten eine ausreichende Frequenzstabilität besitzt. In der praktischen Ausführung wird der ständig schwingende Quarzoszillator (100 kHz) durch die Netzfrequenz synchronisiert.

2. Zeitählung

Die Taktimpulse der Zeitbasis werden in Zehntelsekunden, Sekunden, Minuten und Stunden (jeweils rein binär) gezählt. Der Übertrag von 23.59.59.9 nach 00.00.00.0 wird sowohl in einem Tageszähler als auch in einem Wochentagszähler gezählt. Dabei muß der Tageszähler am Ende eines Monats korrigiert werden.

3. Voreinstellung

Die Uhrzeit und das Datum können voreingestellt werden. Dazu werden die Zahlenwerte für Wochentag, Tag, Monat, Jahr, Stunde und Minute nacheinander in Abhängigkeit von der Stellung eines Drehschalters an sechs Kippschalter mit den Wertigkeiten 1 bis 32 in binärer Form eingestellt und per Tastendruck in die Uhr geladen. (Für die Jahreszahl ist $2^6 = 64$ fest verdrahtet!). Mit dem Voreinstellen der Minute werden Sekunde und Zehntelsekunde auf Null gesetzt. Der Start der Zeitählung muß also immer zur vollen Minute gegeben werden.

4. Anschluß an die Standardnahtstelle

Der Anschluß der Uhr an den Rechner geschieht über eine Steuerung, die die Bedingungen der Standardnahtstelle erfüllt und mit verschiedenen EA- und EV-Befehlen versorgt werden kann.

Bei entsprechender Versorgung der Steuerung können Datum und Uhrzeit als 8 Teilworte in zwei aufeinanderfolgende Kernspeicherzellen übertragen werden. Die Übertragung wird mit einer Fertig-BAP abgeschlossen. Weiterhin kann alle 100 ms oder jede Sekunde eine BAP bei entsprechender Anzeigenübergabe gestellt werden. Die BAP soll im Rechner eine Software-Zeit-zählung auslösen und muß mit einem EA-Befehl quittiert werden.

4.1 EA-Befehl

1	2	5	6	7	8	11	12	14	15	18	19	21	22	24
	Kanal-Nr.	ANZ-TW					Lage	PU-Bit	A	M	S	U	EA	EA-Nr.

Stellen 1 und 8-11 : frei

Stellen 2 bis 5 : Kanalnummer = AAl1-14 der Anzeigenzelle
und = AAl3-16 der Programmunterbrechungszelle

Stellen 6 und 7 : AAl5 und 16 der Anzeigenzelle

Stellen 12 bis 14 : Lage des PU-Bits im Teilwort

Stellen 19 bis 21 : Kennzeichen für EAW (oktal 5)

Stellen 22 bis 24 : Nummer des EAW

Die Stellen 2 bis 7 werden in der Steuerung zwischengespeichert. Zur Adressierung der Anzeigenzelle werden die Stellen 2 bis 5 zu der festverdrahteten Adresse 37740₈ hinzugefügt. Die Stellen 6 und 7 kennzeichnen das Teilwort der ANZ-Zelle. Bei der PU-Zelle werden die Stellen 2 und 3 zu der festverdrahteten Adresse 37760₈ hinzugefügt, und die Stellen 4 und 5 kennzeichnen das Teilwort. Die Bitkombination in den Stellen 12 bis 14 wird decodiert, und das Ergebnis wird für den PU-Bit-Schreibzyklus gespeichert.

4.2 EV-Befehl

ASP-Adresse	A	M	S	U	EV	EV-Nr.
-------------	---	---	---	---	----	--------

Stellen 1 bis 14 : Anfangsadresse eines 2er-Blockes, in den Uhrzeit und Datum eingeschrieben werden sollen

Stellen 19 bis 21 : Kennzeichen für EVS (oktal 6)

Stellen 22 bis 24 : Nummer des EVS

4.3 Mögliche Befehlskombinationen

1. EAØ + EVØ : Eingabe von Uhrzeit und Datum in den Rechner
2. EA1 + EVØ : Eingabe von Uhrzeit und Datum in den Rechner,
Freigabe der 100 ms-BAP und Freigabe einer
Taste (s.u.)
3. EA2 + EVØ : Eingabe von Uhrzeit und Datum in den Rechner,
Freigabe der 1 sec-BAP und Freigabe der Taste.
4. EA3 : Freigabe der 100 ms-BAP.
5. EA4 : Freigabe der 1 sec-BAP.
6. EA5 : Sperren der Zeittakt-BAP und der Taste.

4.4 Anzeigen

Bit 1 : 100 ms (bei einer Zeittakt-BAP)

Bit 2 : 1 sec (bei einer Zeittakt-BAP)

Bit 3 : Fertig (Abschluß der Befehlskombinationen 1 bis 3)

Bit 4 : Zeitfehler (s. 4.5)

Bit 5 : Taste (s. 4.6)

4.5 Zeitfehler

Eine Zeittakt-BAP (100 ms oder 1 sec) muß vom Programm als Zeichen dafür, daß die Software-Zeitzählung durchgeführt wurde, quittiert werden. Als Quittungen dienen EA3 bzw. EA4. Wenn keine Quittung bis zum nächsten Zeittakt erhalten wurde, wird die Anzeige "Zeitfehler" eingeschrieben.

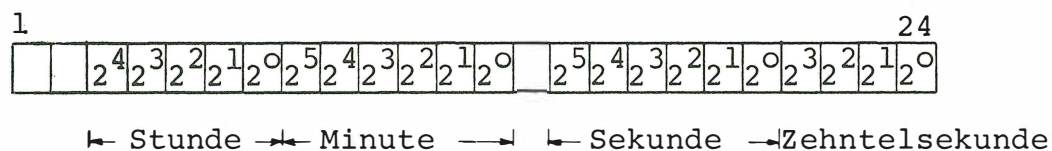
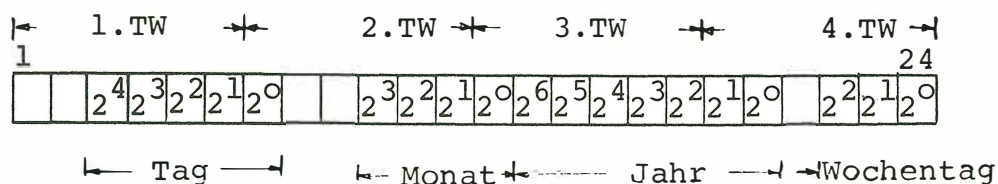
Die Eingabe von Uhrzeit und Datum (Befehlskombinationen 1 bis 3) wird mit dem Zeittakt synchronisiert, um die Wahrscheinlichkeit zu erhöhen, daß die Eingabe bis zum folgenden Zeittakt abgeschlossen ist. Sollte dies jedoch nicht der Fall sein, so wird mit der "Fertig"-Anzeige auch die "Zeitfehler"-Anzeige übergeben. Die Software-Reaktion auf die "Zeitfehler"-Anzeige sollte stets die erneute Übernahme von Uhrzeit und Datum aus der Uhr in den Rechner sein.

Die Anzeigen "Fertig" und "Zeitfehler" verhindern die Anzeigen "100 ms" und "1 sec".

4.6 Taste

Die Taste, die in das Bedienungsfeld des Rechners eingebaut ist, setzt nur das Anzeigenbit 5 bei der folgenden Anzeigenübergabe. Sie bietet die Möglichkeit, auf Wunsch z.B. Uhrzeit und Datum auf dem Bedienungsblattschreiber anzudrucken.

4.7 Bitbelegung der Uhrzeit- und der Datum-Zelle



Bis auf die Wochentagsangabe entspricht diese Bitbelegung derjenigen, die für das ORG und für das Programm ZUDU der SIEMENS-Programmbibliothek vorgeschrieben ist.

(Wochentag: \emptyset = Sonntag; 6 = Samstag)

4.8 Stromversorgung

Die Uhr wird vom Netz betrieben; bei Netzausfall werden der Quarzoszillator und der Zeitzähler aus einer Batterie versorgt, die den Betrieb für ca. 8 Stunden aufrechterhält.

Reinhard Schmitt

Fa. Messerschmitt-Bölkow Blohm

"Der Kanalumsetzer KUS - eine schnelle
Ein-Ausgabe-Einheit für einen Siemens
300 - Rechner mit Schnellkanalzusatz"

Juni 1970

MESSERSCHMITT-BÖLKOW-BLOHM
GESTELLT VON DER FACHABTEILUNG
OTTOBRUNN BEI MÜNCHEN

UK-53-70-8

-3-

Titel: Der Kanalumsetzer KUS - eine schnelle Ein-Ausgabe-Einheit für einen Siemens 300-Rechner mit Schnellkanalzusatz

Verfasser: Reinhard Schmitt

Abteilung: KE11

Datum: 24. 6. 70

Übersicht:

Der Aufbau und die Funktionsweise des von der Firma MBB entwickelten Kanalumsetzers KUS wird besprochen. Der KUS stellt eine allgemeine Prozeßperipherie zur Ein- und Ausgabe von Digitalworten hoher Datenrate in einen Siemens-Rechner der Serie 300 dar. Eine Erweiterung des ORGs ermöglicht es, den KUS systemkonform in Anwenderprogrammen über ORG-Makros anzusprechen und mit der Hilfe von Testprogrammen ist es möglich, den KUS zu warten.

(Auszug aus dem Bericht UK54-70 für einen Vortrag auf der Siemens Prozeßrechner-Benutzer-Tagung vom 29.6. - 1.7.70 an der KFA in Jülich).

1. VERWENDUNG

Der Kanalumsetzer KUS dient als Kommutator bzw. Dekommutator und erlaubt die Ein- oder Ausgabe von max. 14 Digitalworten à 24 Bit über eine Schnellkanalnahtstelle. Die Datenübertragung erfolgt über den Schnellkanal durch direkten Zugriff (cycle stealing) in den Arbeitsspeicher. Der KUS ermöglicht die Übertragung von Daten hoher Rate (max. ca. 160.000 W/sek., bzw. alle 6, μ sec) im Wechselfufferbetrieb. Die Anforderungen auf Datenübernahme oder Datenübergabe können ebenso wie die Kernspeicheradressierung fremdgesteuert, programmgesteuert oder gemischt erfolgen. Bei Fremdadressierung ist auch der Betrieb eines Umlaufpuffers möglich. Die Versorgung des KUS erfolgt über den Programmkanal. Durch Vorschalten eines Alarmzusatzes kann der KUS als Alarmeingabe verwendet werden. Es werden dann bei Änderung eines Alarmwortes das Alarmwort, ein PU-Wort, ein Anzeigenwort und zwei Worte Uhrzeit eingeschrieben, bei gleichzeitigem Stellen einer Programmunterbrechung mittels einer BAP oder einer MAP. Der Anschluß und die programmierte Ansteuerung von AD- und DA-Wandlern ist möglich.

2. BISHERIGE EINSÄTZE

Der KUS wurde bis jetzt zweimal bei 304-Rechnern und einmal bei einem 303-Rechner mit spezieller Anpassung eingesetzt, und zwar zur Eingabe von Telemetriedaten und zur Ein- bzw. Ausgabe von Meßdaten bzw. Stimuli für Check-Out-Aufgaben im Rahmen des Raumfahrtprojektes ELDO.

An einem 305-Rechner fand der KUS gleich drei Mal Verwendung, und zwar als Alarmeingabe- (SEA), Digital-eingabe- (SED) und als Digitalausgabe-Einheit (SAD). Jeweils ein Digitaleingabe- und ein Digitalausgabe-kanal dient über AD- bzw. DA-Wandler der Firma Adage zur Analog-Ein- bzw. Ausgabe. Ein Analogwert kann dabei in den Adage-Geräten noch einmal auf einem von 16 Analogkanälen ausgegeben bzw. von einem von 32 Analogkanälen hereingeholt werden. Diese Anlage dient sowohl für Check-Out-Aufgaben als auch zur Datenerfassung bei Triebwerksversuchen.

3. AUFBAU

Der KUS enthält folgende Register, welche bis auf die Ausgaberegister alle per Programm mittels EVS-Befehl versorgt werden können.

- 3.1 2 Adreßregister: Das Adreßregister, je eines für eine Wechselpufferhälfte, enthält die Adresse der Zelle aus der gelesen bzw. in die geschrieben werden soll. Nach jedem Datentransfer wird der Inhalt des Adreßregisters, sofern es per Programm nicht gesperrt wurde, um 1 erhöht.
- 3.2 2 Blocklängenzähler: Der Blocklängenzähler, je einer für eine Wechselpufferhälfte, enthält die Anzahl der Zellen aus denen noch gelesen bzw. in die noch geschrieben werden soll. Nach jedem Datentransfer

wird der Inhalt des Blocklängenzählers, sofern er nicht per Programm gesperrt wurde, um 1 verringert. Die Umschaltung von 1 auf Null, das Endkriterium eines Blockes, bewirkt eine automatische Umschaltung auf die andere Wechselpufferhälfte, wenn sie nicht per Programm gesperrt wurde. Bei Ende eines Blockes kann eine Programmunterbrechung per BAP oder MAP gestellt werden, um dem Programm eine erneute Versorgung zu ermöglichen.

3.3 1 Adreßmischregister: Das Adreßmischregister gibt an, welche Bits der Adresse für den nächsten Datentransfer von der extern angelieferten Adresse und welche Bits von dem im Augenblick aktiven Adreßregister bestimmt sein sollen.

3.4 1 Sprungregister: Das Sprungregister gibt an, bei welchen Kanälen nur die extern angelieferte Adresse verwendet werden soll. Die interne Adresse und die Blocklänge der augenblicklich effektiven WP-Hälfte werden dabei nicht verändert. Dadurch ist es möglich, Uhrzeiten oder ähnliche Informationen von bestimmten Kanälen in bestimmte Kernspeicherzellen abzusetzen, parallel zu einem Wechselpufferbetrieb mit anderen Ein- oder Ausgabekanälen.

- 3.5 1 Anforderungsregister: Das Anforderungsregister speichert die Anforderungen auf Datentransfers, die für die einzelnen Kanäle gestellt wurden. Diese Anforderungen können entweder programmiert mittels eines EVS-Befehls oder extern vom Prozeß erfolgen.
- 3.6 1 Maskenregister: Das Maskenregister kann verhindern, daß externe Anforderungen das Anforderungsregister setzen. Es bestimmt damit bei welchen Worten Anforderungen von extern erwartet werden und zugelassen sind.
- 3.7 12 Ausgangsregister: (nur bei Ausführung als Ausgabe-Einheit) Das aus dem Kernspeicher gelesene Wort wird im Ausgaberegister des entsprechenden Ausgabekanals gespeichert. Ein Datenübergabesignal zeigt dem Prozeß an, daß die Information zur Ausgabe im Ausgangsregister bereitsteht.

4. VERSORGUNG DER NAHTSTELLE

Ein Rücksetzen der Register in eine unverfängliche Ausgangsstellung kann über die Taste "Anlage Rücksetzen" erfolgen. Die Versorgung der Register mit neuer Information erfolgt über EAW- und EVS-Befehle.

4.1 Der EAW-Befehl hat folgenden Aufbau:

- | | |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Bit 1 | ist unbenutzt |
| Bit 2 bis 5 | bestimmen die Schnellkanalnahtstelle an welcher der KUS angeschlossen ist. |
| Bit 6 | ist unbenutzt |
| Bit 7 | bestimmt, ob ein Startkriterium für Ein- oder Ausgabekanal gesetzt wird, solange wenigstens eine WP-Hälfte zu einem Datentransfer betriebsbereit ist. Das Kriterium wird Null, sobald kein Datentransfer mehr über den WP stattfinden kann (d. h. wenn beide WP-Hälften neu zu versorgen sind).
1 = mit Startkriterium
2 = ohne Startkriterium |
| Bit 8 | bestimmt, ob die WP-Hälfte 1 zum Datentransfer freigegeben wird.
1 = WP-Hälfte 1 freigegeben
0 = WP-Hälfte 1 gesperrt. |
| Bit 9 | bestimmt, ob die WP-Hälfte 2 zum Datentransfer freigegeben wird.
1 = WP-Hälfte 2 freigegeben
0 = WP-Hälfte 2 gesperrt |

MESSERSCHMITT-BÖLKOW-BLOHM GESELLSCHAFT MIT BESCHRÄNKTER HAFTUNG OTTOBRUNN BEI MÜNCHEN	UK-53-70-Ü	-9-
Bit 10	ermöglicht bei einer Alarmeingabe das Einschreiben aller Alarmworte. 1 = Alarmworte einschreiben 0 = Alarmworte nicht einschreiben	
Bit 11	bestimmt, ob der gerade aktive Block- längenzähler zum Weiterzählen gesperrt wird. 1 = Blocklängenzähler freigegeben 0 = Blocklängenzähler gesperrt	
Bit 12	bestimmt, ob der gerade aktive Adreß- zähler zum Weiterzählen gesperrt wird. 1 = Adreßzähler freigegeben 0 = Adreßzähler gesperrt	
Bit 13	bestimmt, ob beim Ende einer WP-Hälfte eine BAP gestellt wird. 1 = BAP 0 = keine BAP	
Bit 14	bestimmt, ob beim Ende einer WP-Hälfte eine MAP gestellt wird. 1 = MAP 0 = keine MAP	
Bit 15 bis 19 haben die gleichen Bedeutungen wie bei allen Befehlen		
Bit 20 bis 21 enthalten die Codierung des EAW-Befehls		
Bit 22 bis 24 enthalten die Nummer (0 bis 7) des EAW- Befehls		

4.2 Der EVS-Befehl hat folgenden Aufbau:

Bit 1 bis 14 enthalten die Versorgungs-Information
für das gewählte Register.

Bit 15 bis 19 haben die gleichen Bedeutungen wie bei
allen Befehlen.

Bit 20 bis 21 enthalten die Codierung des EVS-Befehls

Bit 22 bis 24 enthalten die Nummer (0 bis 7) des EVS-
Befehls.

Die Nummer des EAW-Befehls wird zwischengespeichert,
so daß es möglich ist, diese mit den Nummern der darauf-
folgenden EVS-Befehle zur Auswahl eines zu versorgenden
Registers zu verknüpfen. Da es 8 EAW- und 8 EVS-Befehle
gibt, ergeben sich daraus 64 Möglichkeiten, von denen
jedoch nur ca. 17 mit ca. 5 redundanten (je nach Ausbau
des KUS) Verwendung finden.

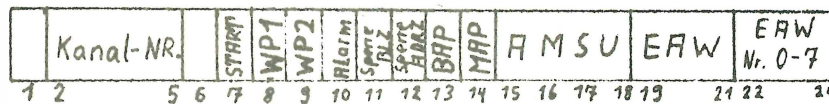
Die nachfolgende Tabelle gibt Aufschluß über die Auf-
gaben der EAW-Befehle mit den entsprechenden EVS-Befeh-
len.

MESSERSCHMITT-BÜLKOW-BLOHM
GESELLSCHAFT MIT BESCHRÄNKTER HAFTUNG
OTTOBRUNN BEI MÜNCHEN

UK-53-70-8

-11-

Aufbau des EAW-Befehls



8x8=64
Möglichkeiten

Aufbau des EVS-Befehls



EVS-Nr.	Aufgabe	EA EA EA EA EA EA EA EA							
		0	1	2	3	4	5	6	7
0	Versorgung des Adreßzählers 1 mit einer Anfangsadresse		X	X					
1	Versorgung des Adreßzählers 2 mit einer Anfangsadresse		X	X					
4	Versorgung des Blocklängenzählers 1 mit einer Blocklänge		X	X					
3	Versorgung des Blocklängenzählers 2 mit einer Blocklänge		X	X					
5	Versorgung des Adreßregisters 1 = dieses Bit wird vom Adreßzähler genommen 0 = dieses Bit wird von der Externadresse genommen		X						
6	Versorgung des Sprungregisters 1 = die Adresse kommt rein extern 0 = die Adresse wird aus der Adreßmischung genommen		X						
5	Interne Anforderung setzen (lang) 1 = für dieses Wort wird eine Anforderung gesetzt 0 = für dieses Wort besteht keine Anforderung					X			
6	Interne Anforderung setzen (kurz) (mit Blockierung d. Programmsteuerung) 1 = für dieses Wort wird eine Anforderung gesetzt 0 = für dieses Wort besteht keine Anforderung					X			

MESSERSCHMITT-BÖLKOW-BLOHM GESELLSCHAFT MIT BESCHRÄNKTER HAFTUNG OTTOBRUNN BEI MÜNCHEN		UK-53-70-Ö								-12-	
EVS-Nr.	Aufgabe	EA	EA	EA	EA	EA	EA	EA	EA		
		0	1	2	3	4	5	6	7		
7	Versorgung des Maskenregisters 1 = die von extern kommende Anforderung ist freigegeben 0 = die von extern kommende Anforderung ist gesperrt		X								
0	Einstellen der Uhr Bit 5 bis Bit 14 geben die Zeit in den Wertigkeiten 2^9 bis 2^{10} in sec an.								X		
1	Einstellen der Uhr Bit 5 bis Bit 14 geben die Zeit in den Wertigkeiten 2^{19} bis 2^{10} in sec an								X		
2	Versorgungsparameter für die Analogeingabe		X	X							

5. ARBEITSWEISE

Werden im Anforderungsregister für einen oder mehrere Kanäle Anforderungen gestellt, entweder per Programm oder bei einer entsprechenden Maske des Maskenregisters von einer externen Anforderungsleitung, so wird an den Abtaster (Scanner) eine Sammelanforderung gestellt. Durch die Sammelanforderung läuft der Abtaster los und fragt das Anforderungsregister in der Reihenfolge der Worte (Prioritätsfolge) auf eine gesetzte Anforderung ab. Wird eine Anforderung gefunden, so geschehen folgende Abläufe quasi gleichzeitig:

- 5.1.1 Es wird der Datenweg vom Datenkanal des Schnellkanalzusatzes zum Daten-Ausgabe-Register des entsprechenden Kanals bei einer Ausgabe oder zum Daten-Eingabe-Stecker bei einer Eingabe durchgeschaltet.
- 5.1.2 Die in der Adreßverarbeitung (Adreßmischung) entstandene Adresse oder, wenn das Sprungregister für dieses Wort gesetzt ist, die Externadresse wird ebenfalls zum Datenkanal des Schnellkanalzusatzes durchgeschaltet.
- 5.1.3 Über den Datenkanal des Schnellkanalzusatzes wird der Zentraleinheit der Modus des Datentransfers mitgeteilt. (Lesen oder Schreiben, jeweils mit oder ohne Löschen).
- 5.1.4 Weiter wird über den Datenkanal des Schnellkanalzusatzes eine Anforderung an die Datensteuerung (ADST) gestellt.

Mit dem zugeteilten Kernspeicherzyklus findet der Datentransfer statt. Dem KUS wird dies über den Datenkanal des Schnellkanalzusatzes mittels des EABF-Signals (Element Abfertigung) mitgeteilt.

Die EABF bewirkt im KUS folgende Vorgänge:

- 5.2.1 Bei der Ausgabe leitet der KUS aus dem EABF-Signal das Triggersignal für das Daten-Ausgabe-Register ab.
- 5.2.2 Bei einem Ausgabekanal wird nach der Datenübernahme in das Daten-Ausgabe-Register das Datenübergabesignal für den Anwender gesetzt.
- 5.2.3 Der Adreßzähler wird, sofern er nicht gesperrt ist, um 1 erhöht.
- 5.2.4 Der Blocklängenzähler wird, sofern er nicht gesperrt ist, um 1 verringert. Wird im Blocklängenzähler durch das Herabzählen der Inhalt Null erreicht, so wird, wenn es per EAW-Befehl freigegeben ist, eine BAP bzw. eine MAP gestellt. Ist der andere Blocklängenzähler ungleich Null und die Steuerung für die andere WP-Hälfte nicht gesperrt, so wird auf die andere WP-Hälfte umgeschaltet.
- 5.2.5 Im Anforderungsregister wird die Anforderung für das gerade bearbeitete Wort gelöscht. Liegen noch weitere Anforderungen vor, so tastet der Scanner die weiteren Kanäle ab; andernfalls begibt er sich wieder in die Nullage und wartet auf eine neue Sammelanforderung.

6. SOFTWARE

Um den KUS über das ORG systemgerecht über Makros anzusprechen zu können, wurden entsprechende Bearbeitungsprogramme geschrieben, welche mit einer Korrekturübersetzung ins ORG eingebaut werden können. Es ergeben sich dann folgende Makros:

- 6.1.1 MA KW1 = n, ADR1, ADR2
MA KW2 = n, ADR1, ADR2

Das ORG baut für jedes Prozeßelement KUS je eine Warteschlange für jede der beiden Wechselpufferhälften W1 und W2 auf. Es trägt den Aufruf in die Warteschlange für W1 bzw. W2 des Elements mit der Nummer n ein, startet die Operation (wenn nicht bereits eine läuft) und setzt das Programm fort. Danach laufen Ein- bzw. Ausgabe und Programm simultan zueinander ab. Das ORG veranlaßt den Datentransfer in den bzw. aus dem Arbeitsspeicher von Zelle ADR1 bis ADR2, je nach dem Typ des angeschlossenen KUS.

Soll auf das Ende der Operation gewartet werden, so ist ein Warteaufruf MA EXWA zu geben.

- 6.1.2 MA KMAR = n, BM

Der Aufruf dient zur Versorgung des Maskenregisters und wird nicht in eine Warteschlange eingetragen. Die Elementbefehlsgruppe wird sofort aufgebaut und das Bitmuster BM als Maske an das Element mit der Nummer n übergeben. Eine 1 im Bitmuster der Maske gibt die externe Anforderungsleitung für das entsprechende Wort frei.

6.1.3 MA KSPR = n, BM

Der Aufruf dient zur Versorgung des Sprungregisters und wird nicht in eine Warteschlange eingetragen. Die Elementbefehlsgruppe wird sofort aufgebaut und das Bitmuster BM als Sprungkennzeichnung für die einzelnen Worte an das Element mit der Nummer n übergeben. Eine 1 im Bitmuster gibt an, bei welchem Wort der Datentransfer nicht über die Wechselpuffersteuerung erfolgt, sondern die Externadresse für den Transfer bestimmend ist.

6.1.4 MA KIAL = n, BM

MA KIAK = n, BM

Der Aufruf dient zur Versorgung des Anforderungsregisters und wird nicht in eine Warteschlange eingetragen.

Die Elementbefehlsgruppe wird sofort aufgebaut und das Bitmuster BM als Anforderungen für die einzelnen Worte an das Element mit der Nummer 1 übergeben.

Eine 1 im Bitmuster gibt an, daß für dieses entsprechende Wort eine Anforderung auf Datentransfer gestellt werden soll.

Der Aufruf KIAK unterscheidet sich vom Aufruf KIAL dadurch, daß die Programmsteuerung solange angehalten wird, bis der Datentransfer aller gestellten Anforderungen stattgefunden hat, während beim Aufruf KIAL der Datentransfer erst bei der nächsten Möglichkeit stattfindet.

6.1.5 MA KAR1 = n, ADR

MA KAR2 = n, ADR

Der Aufruf dient zur Versorgung des Adreßregisters 1 oder 2 und blockiert die Weiterzählung des Adreßregisters, er wird nicht in eine Warteschlange eingetragen.

Die Elementbefehlsgruppe wird sofort aufgebaut und die Adresse ADR dem Adreßregister 1 oder 2 übergeben.

6.1.6 MA KBR1 = n, ADR1, ADR2

MA KBR2 = n, ADR1, ADR2

Der Aufruf dient zur Versorgung des Blocklängenzählers 1 oder 2, er wird nicht in eine Warteschlange eingetragen. Die Elementbefehlsgruppe wird sofort aufgebaut und die Blocklänge als Differenz der Adressen ADR1 und ADR2 dem Blocklängenzähler 1 oder 2 übergeben. Ist die Differenz negativ, so wird der Blocklängenzähler ungleich Null gesetzt und die Weiterzählung blockiert.

Außer diesen Makros wurden einige Testroutinen geschrieben, welche es gestatten, Hardwarefehler schnell einzukreisen. Es sind die Programme:

6.2.1 PRAE Prüfen der Ausgabe und der Eingabe

Verbindet man ein Ausgabewort über eine Steckverbindung mit einem Eingabewort, so gibt das Programm nacheinander sämtliche Bitkombinationen aus und holt sie sofort über die Eingabe wieder herein; der anschließende Vergleich zeigt, ob ein Fehler vorhanden ist. Bei Ungleichheit zwischen ausgegebenem und hereingeholtem Bitmuster werden beide Bitmuster auf dem Bedienungsblattschreiber ausgegeben.

6.2.2 PRAU Prüfen der Ausgabe

Das Programm gibt ein frei wählbares Bitmuster aus einer frei wählbaren Adresse aus. Der Vorgang kann durch Setzen der Bedingung A am Bedienungsfeld zyklisch wiederholt werden, so daß alle Signale mit dem Oszillographen leicht kontrolliert werden können.

6.2.3 PREI Prüfen der Eingabe

Das Programm holt ein anliegendes Bitmuster in eine frei wählbare Zelle herein und gibt das Bitmuster auf dem Bedienungsblattschreiber aus. Durch das Setzen der Bedingung A am Bedienungsfeld kann der Vorgang zyklisch wiederholt werden, so daß alle Signale mit dem Oszillographen leicht kontrolliert werden können. Durch Setzen der Bedingung B wird die Ausgabe auf dem Bedienungsblattschreiber unterdrückt.

6.2.4 PRAA Prüfen der Analog-Ausgabe

Dieses Programm ist nur sinnvoll wenn DA-Wandler der Firma Adage an einen Ausgabe-KUS angeschaltet sind. Für andere Geräte muß es leicht umgeschrieben werden. Das Programm PRAA gibt Spannungen mit höchster Geschwindigkeit in einen frei wählbaren Bereich zwischen +20V und -20V in frei wählbaren Treppenschritten aus, so daß man die Treppenkurven auf dem Oszillographen verfolgen kann. Durch Setzen der Bedingung A am Bedienungsfeld kann der Vorgang zyklisch wiederholt werden, so daß alle Signale mit dem Oszillographen leicht kontrolliert werden können.

6.2.5 PAEI Prüfen der Analog-Eingabe

Dieses Programm ist nur sinnvoll im Zusammenhang mit den entsprechenden AD-Wandlern der Firma Adage.

Das Programm PAEI liest angelegte Spannungen in den Kernspeicher und gibt den Wert auf dem Bedienungsblattschreiber oder dem Schnelldrucker aus. Durch Setzen der Bedingung A am Bedienungsfeld kann der Vorgang zyklisch wiederholt werden, so daß alle Signale mit dem Oszillographen leicht kontrolliert werden können. Durch Setzen der Bedingung B am Bedienungsfeld wird die Ausgabe auf dem Bedienungsblattschreiber bzw. dem Schnelldrucker unterdrückt.

7. TECHNISCHE DATEN

	Eingabe-KUS	Ausgabe-KUS	Alarm-Eingabe-KUS
Rahmen SIVAREP B	zweizeilig	zweizeilig	dreizeilig
Anzahl der Flach- baugruppen in TTL- Technik (je nach Ausbau)	ca. 32	ca. 39	ca. 46
Anzahl der Flach- baugruppen in diskreter Aufbau- technik	4 Kabel- sender 2 Kabel- Empfänger	2 Kabel- Sender 4 Kabel- Empfänger	4 Kabel- Sender 2 Kabel- Empfänger
Anzahl der Ein- bzw. Ausgabe- Datenstecker zur Anwender-Periphe- rie (je nach Ausbau)	max. 14	max. 12	max. 14
Anzahl der Kabel- stecker zur Rechnernahstelle	7	7	7

Stromversorgung

+ 12V	+ 5 %	ca. 9W	ca. 9W	ca. 18W
- 12V	+ 5 %	ca. 12W	ca. 12W	ca. 6W
+ 5V	+ 5 %	ca. 18W	ca. 25W	ca. 30W

Elektrische Eigenschaften der Prozeß-Nahstelle

Signalpegel	L = 2,4 ... 5,5V oder offener Eingang
	O = 0 ... 0,4V

Belastung der Eingangssignale:	ca. 1,6mA
max. Belastung der Ausgangssignale:	ca. 16mA
Invertierung der Information:	keine

H.-J. Schuster
Physikalisch-Technische
Bundesanstalt, Braunschweig

"Experimentgeordnete und extern-
gesteuerte Datenübertragung beim
Prozeßrechner Siemens 305"

Experimentgeordnete und externgesteuerte
Datenübertragung beim Prozeßrechner
Siemens 305

H.-J. Schuster

Physikalisch-Technische Bundesanstalt, Braunschweig

Zusammenfassung

Das Prozeßelement P3K [1] gestattet eine im wesentlichen extern gesteuerte Datenübertragung aus mehreren Primärgeräten in einen gemeinsamen Pufferbereich des Rechners Siemens 305 oder aus einem gemeinsamen Pufferbereich in mehrere Primärgeräte. Daraus, daß mehreren Primärgeräten nur jeweils ein gemeinsamer Pufferbereich zugeordnet ist, ergeben sich viele Einschränkungen und Nachteile für die Leistungsfähigkeit des Prozeßsystems, z.B.:

- Geschwindigkeitsverlust: Die Daten verschiedener Experimente werden untereinander vermischt und müssen im allgemeinen wieder sortiert werden, was Rechenzeit kostet.
- Beeinflussung der Experimente (s.u.).
- Verlust an Informationskapazität durch die notwendige Kennzeichnung der Daten.
- Keine Möglichkeit der Prioritätssteuerung bei der Verarbeitung der Daten (s.u.).
- Einschränkungen beim Anschluß von Ausgabeeinheiten (s.u.).

Aus diesen Gründen wurde durch geringfügige Änderungen der P3K und einige elektronische Zusatzeinrichtungen die Möglichkeit geschaffen, die Datenübertragung in den und aus dem Rechner experimentgeordnet vorzunehmen. Unter experimentgeordneter Datenübertragung wird verstanden, daß jedem an die P3K angeschlossenen Primärgerät jeweils ein eigener Pufferbereich im Arbeitsspeicher zugeordnet ist, über welchen seine Meßdaten gesondert ein- oder ausgegeben werden.

Experimentgeordnete Dateneingabe

Für die experimentgeordnete Dateneingabe werden die beiden als Adreßzähler wirkenden Vorwärtzzähler in der Eingabesteuerung P3ES/N [2] ersetzt durch mehrere Adreßzähler, von denen in jedem Primärgerät jeweils einer vorgesehen ist. Das erfordert, daß zu jeder Eingabeanforderung neben der Wortinformation nun auch die Arbeitsspeicheradresse der Zelle, in die übertragen werden soll, der Eingabesteuerung mitgeteilt werden muß. Die Arbeitsspeicheradresse ist die Summe aus dem Zählerstand des zuständigen Adreßzählers, einer experimenteigenen Basisadresse und der für den gesamten Eingabebereich gültigen Basisadresse. Die vom Primärgerät angebotene Adreßinformation (Summe aus Adreßzählerstand und Experimentbasisadresse) wird parallel zur Wortinformation zur Eingabesteuerung übertragen. Dazu sind in der Zusatzsteuerung analog zu der Einrichtung für die Übertragung der Wortinformation ein Register für jedes Primärgerät, eine Weiche mit 8 Eingängen und einem Ausgang zur Durchschaltung der Adreßinformation und ein Register für die Steuerung vorgesehen. Die Übernahme der Adreßinformation in das Adreßinformationsregister erfolgt mit dem Signal $INFUE_x$, die Adreßinformationsdurchschaltung mit $INFDU_x$ und die Übernahme in das Adreßinformationsregister für die Steuerung mit dem Takt TD2 der P3ES/N [2]. Dem Register für die Steuerung ist ein Adreßaddierer nachgeschaltet, mit dessen Hilfe die Basisadresse zur Adreßinformation addiert wird.

In der Eingabesteuerung wird die Platine des ersten Vorwärtzzählers (Flachbaugruppe 4005) gegen eine Eingabeplatine ausgetauscht, über welche anstelle des Inhalts des Vorwärtzzählers die im Adreßaddierer anstehende Arbeitsspeicheradresse angeboten wird. Außerdem muß in der Steuerung durch Festlegen des Pegels LZA I auf "Eins" das Signal "Rückwärtzzählerinhalt gleich "Null" $\hat{=}$ BAP unterdrückt werden, so daß die Steuerung stets empfangsbereit bleibt und nicht neu versorgt werden muß.

Die beiden Bereichshälften jedes Adreßzählers legen zwei Puffer im Arbeitsspeicher fest, mit denen ein Wechselpufferbetrieb möglich ist. Nach jeder Worteingabe eines Primärgerätes wird der Inhalt des zuständigen Adreßzählers um 1 erhöht. Ist das Ende einer Bereichshälfte erreicht, wird eine Alarmmeldung an das Programm gegeben. Zu diesem Zwecke ist jedem Primärgerät ein Alarmeingang zugeordnet. Diese Alarmmeldung ersetzt die bedingte Anweisung an die Programmsteuerung BAP und teilt mit, daß ein Wechselpuffer gefüllt ist und abgearbeitet werden kann. Der letzte Schritt eines Arbeitsprogrammes ist die letzte Zelle des Wechselpuffers, in den gerade Daten eingespeist wurden, auf z.B. Null zu prüfen, wenn bei der Abarbeitung die Zelleninhalte jeweils gelöscht wurden und Null als Information nicht auftritt. Ergibt die Prüfung einen Wert ungleich Null, ist ein Datenverlust sehr wahrscheinlich und eine Meldung an das Experiment notwendig. Dadurch kann auf die Anzeigenzelle im Versorgungsanruf der P3K verzichtet werden.

Bis auf die Vor- und Rückwärtszähler werden im wesentlichen alle Einrichtungen der P3EG und der P3ES/N auch bei dieser experimentgeordneten Dateneingabe benutzt.

Vorteile der experimentgeordneten Dateneingabe

Bei Datenverlust ist das betroffene Experiment bekannt. Die Datenabarbeitung kann durch Prioritätswahl nach Dringlichkeit erfolgen, so daß bei Überlastung des Rechners zunächst unwichtigere Werte verlorengehen.

Die Gefahr der Beeinflussung der Experimente untereinander ist stark reduziert. Durch programmtechnische Maßnahmen kann verhindert werden, daß wichtige Werte eines Experimentes durch zu großen Datenanfall in einem anderen Experiment unberücksichtigt bleiben.

Auf eine Kennzeichnung der Daten nach Experimentzugehörigkeit kann verzichtet werden, so daß durch das Fehlen von Markierungsbits Informationskapazität eingespart wird.

Dadurch, daß die Meßdaten im Arbeitsspeicher bereits experimentgeordnet vorliegen, kann bei der Abarbeitung erheblich Rechenzeit eingespart werden.

So werden z.B. bei der "On-Line"-Abarbeitung der Daten ca. 10 Befehle pro Wort dadurch eingespart, daß die Prüfung nach Experimentzugehörigkeit und die Anregung des entsprechenden Abarbeitungsprogramms entfällt. Auf folgende Befehle kann verzichtet werden:

LAP	Ø	
TEP'	(X)	Trennung Markierungsbits von der Wortinformation
VDL	3	
VLR'	3	
VLL	10	
ADA	BA	Ermittlungen des Unterprogramms für die Abarbeitung und Sprung zu diesem Unterprogramm
TAS	HZ	
SPR	(HZ)	
UNT	UNTPR	
SPR	(UNTPR)	Rücksprung aus dem Unterprogramm.

Besonders bei 2-parametrigen Experimenten ist es oft sehr zweckmäßig, die Daten vor der Weiterverarbeitung (z.B. Speicherung auf der Platte) zunächst nach Experimenten zu ordnen. Man kann dann z.B. durch einmalige Ausgabe auf eine Speicherröhre über die Ausgabesteuerung bereits vor der Abspeicherung auf die Platte eine sehr einfache Akkumulierung der Daten erreichen. In diesem Fall werden durch die experimentgeordnete Eingabe zu den oben erwähnten Befehlen weitere vier Befehle pro Wort eingespart:

TAS' (ADR)

EHA ADR

ENA ADRP Einsortieren der Daten in einen
 experimenteigenen Puffer

SGN UMLAD

Die eingesparte Rechenzeit entspricht etwa der zur Daten-Übertragung auf die Platte notwendigen Rechenzeit. Das bedeutet, daß in diesem Fall durch die experimentgeordnete Übertragung die Eingaberate verdoppelt oder aber der Eingangspufferbereich verkleinert werden kann.

Experimentgeordnete Datenausgabe

Entsprechend zur Dateneingabe läßt sich auch die Datenausgabe über die P3K experimentgeordnet vornehmen. D.h. jedes Ausgabeprimärgerät bezieht seine Daten aus einem eigenen Pufferbereich im Arbeitsspeicher. Wie bei der Datenausgabe werden dazu die Vorwärtszähler der Ausgabesteuerung P3AS [3] ersetzt durch in den Primärgeräten vorgesehene Adreßzähler. Mit jeder Ausgabeanforderung muß der Steuerung dann zusätzlich die Arbeitsspeicheradresse der Zelle angeboten werden, deren Inhalt ausgegeben werden soll. Wie bei der Dateneingabe ist die Arbeitsspeicheradresse die Summe aus der vom Primärgerät angebotenen Adreßinformation (Summe aus Adreßzählerstand und einer einstellbaren Experimentbasisadresse) und der Basisadresse des gesamten Ausgabebereichs. Im Gegensatz zur Dateneingabe wird als Ausgabegerät nicht das P3AG [3], sondern ein speziell angefertigtes Gerät mit 8 Ausgängen benutzt. Dies erschien vorteilhaft, da viele Einrichtungen des P3AG nicht benötigt werden, wie z.B.: Maskensteuerung, Paritykontrolle, Kettung und Rücksetzen. Auf diesen Steuerungskomfort kann verzichtet werden, da einerseits durch die Datenausgabe der Arbeitsspeicherinhalt nicht verändert wird und außerdem an diese Ausgabeeinrichtung im wesentlichen nur unkritische Wiedergabegeräte (Display) angeschlossen werden (s.u.).

Das Ausgabegerät entspricht im Prinzip einer einfacheren Ausführung des P3AG mit einer zusätzlichen Einrichtung zur Übertragung der Adreßinformation und arbeitet in ähnlicher Weise wie die oben beschriebene Kombination aus P3EG und Zusatzsteuerung.

In der Ausgabesteuerung wurde wieder der erste Vorwärtszähler ersetzt durch eine Eingabeplatine, über die an Stelle des Inhalts des Vorwärtszählers die im Ausgabegerät anstehende Arbeitsspeicheradresse angeboten wird. Durch Festlegen des Pegels LZA I auf Eins wird verhindert, daß die P3AS neu versorgt werden muß (s.o.).

Vorteile der experimentgeordneten Datenausgabe

Die extern gesteuerte und experimentgeordnete Datenausgabe bietet erhebliche Vorteile beim Anschluß von Wiedergabesteuerungen (Sichtgerät, Punktdrucker).

An das oben beschriebene Ausgabegerät lassen sich bis zu acht voneinander völlig unabhängig arbeitende Wiedergabesteuerungen anschließen.

Jede Wiedergabesteuerung besteht im wesentlichen aus einem Taktgenerator, dem oben beschriebenen Adreßzähler mit einer nachgeschalteten Addierstufe zur Addition der Experimentbasisadresse, einem Eingangsinformationsregister und 2 Digital-Analog-Wandlern. Mit jedem Taktsignal des Generators wird der Adreßzähler um Eins weitergeschaltet und eine Anforderung an das Ausgabegerät gestellt, die Wortinformation der durch die Adreßinformation definierten Zelle auszugeben. Mit dem Rückmeldesignal des Ausgabegerätes wird der Zelleninhalt in das Eingangsinformationsregister der Wiedergabesteuerung übernommen. Die Inhalte des Adreßzählers und des Informationsregisters werden jeweils mit Hilfe eines Digital-Analog-Wandlers in analoge Spannungswerte umgesetzt und z.B. an die x- und y-Ablenkplatten einer Braunschen Röhre gegeben. Auf diese Weise

können bis zu 8 Wiedergabesteuerungen, deren Taktfrequenzen völlig unkorreliert sind, gleichzeitig betrieben werden. Durch eine geringfügige Erweiterung der Wiedergabesteuerung und eine vorherige einmalige programmgesteuerte Aufbereitung der Daten ist auch eine 3-dimensionale Darstellung möglich.

Die Datenausgabe erfolgt rein extern gesteuert ohne Mitwirkung der Programmsteuerung. Die Versorgung der Ausgabesteuerung entfällt. Dadurch wird die bestmögliche Ausnutzung der für die Bildausgabe zur Verfügung stehenden Rechenzeit garantiert. Jede Wortausgabe benötigt nur eine Zykluszeit des Rechners.

Die maximal mögliche Auslastung des Rechners durch die Bildausgabe kann mit Hilfe eines Anforderungsunivibrators im Ausgabegerät eingestellt werden. Mit den Anstiegsflanken der Ausgangsimpulse dieses Univibrators werden die Anforderungen an die Steuerung gestellt. Der Mindestabstand zwischen den Anforderungen ist dann durch die einstellbare Impulsbreite des Univibrators gegeben.

Zur Speicherung der Anforderungen an das Ausgabegerät ist jeweils eine Anforderungs-Flip-Flop vorgesehen. Dieses wird von den vom Primärgerät gestellten Anforderungen gesetzt und entweder von der Rückmeldung der Steuerung zurückgesetzt oder nach einer einstellbaren Wartezeit vom Ausgangsimpuls eines Univibrators, wobei die Anforderung verlorenggeht. Dadurch können von der Wiedergabesteuerung, unabhängig von der Auslastung des Computers, zu definierten Zeitpunkten neue Anforderungen gestellt werden, so daß der Arbeitsrhythmus der Wiedergabesteuerung nicht gestört wird. Der Verlust von Anforderungen äußert sich durch unregelmäßiges Ausbleiben von Leuchtpunkten auf dem Bildschirm. Durch die Wahl der Wartezeiten kann außerdem auf die Sicherheit, mit der eine Anforderung bedient wird, Einfluß genommen werden. Z.B. wird man die Wartezeit für eine Punktdruckersteuerung groß wählen, da hier der Verlust einer Anforderung mehr stören würde als bei einer Steuerung für ein Sichtgerät.

Die Adreßzählerstände oder die x-Ablenkung in der Bildröhre stehen immer in fester Beziehung zu den Adressen des auszugebenden Arbeitsspeicherbereichs. Dadurch kann auf Synchronisiereinrichtungen in der Wiedergabesteuerung verzichtet werden.

Literatur:

- [1] Beschreibung des Prozeßelementes P3K für Siemens-Prozeßrechner von 1969.
- [2] Beschreibung der Eingabesteuerung P3ES/N und des Eingabegerätes P3EG für Siemens-Prozeßrechner von 1969.
- [3] Beschreibung der Ausgabesteuerung P3AS und des Ausgabe-gerätes P3AG für den Siemens-Prozeßrechner von 1969.

H. Englmeier
Max-Planck-Institut für
Eiweiss- u. Lederforschung
Abteilung f. Röntgenstruktur-
forschung, München

"Ein externes Leitwerk als Interface
zwischen Messgerät und Rechenmaschine"

Hanspeter Englmeier

EIN EXTERNES LEITWERK ALS INTERFACE ZWISCHEN MESSGERÄT
UND RECHENMASCHINE

Über die gleichen Einrichtungen, die an einer Rechenmaschine für Datenübertragungen vorhanden sind, werden auch chemische und physikalische Meßgeräte an eine Rechenmaschine angeschlossen. Die Durchführung einer Messung kann jedoch schwieriger sein, als eine reine Datenübertragung. Im allgemeinen ist nämlich ein chemisches oder physikalisches Meßgerät zugleich Sender und Empfänger von Informationen. Es empfängt Informationen über die Freiheitsgrade, die für die Messung eingestellt werden müssen, und über zusätzliche Bedingungen, unter welchen die Messung durchgeführt werden soll. Dabei muß häufig die Einstellung der Freiheitsgrade und Bedingungen und die Durchführung der Messung von der Rechenmaschine gesteuert werden. Das Meßgerät sendet die Daten der gemessenen Größen und Informationen über seinen Zustand und etwaige Überschreitungen seiner Grenzen.

Meßgeräte, bei denen die Vorbereitung der Messung und die Messung selbst von der Rechenmaschine gesteuert werden müssen, werden häufig auf folgende zwei Methoden an die Rechenmaschine angeschlossen. Im ersten Fall übernimmt die Rechenmaschine per Programm und Externbefehl die Steuerung des Meßgeräts. Die Externbefehle erhalten dabei ungefähr folgende Funktionen: Erstens übertragen sie Informationen aus Registern der Rechenmaschine an Register des Meßgeräts und umgekehrt. Zweitens können Externbefehle elektronische Uhren starten und stoppen, die, an den Interrupt der Rechenmaschine angeschlossen, diese zur Zeitzählung veranlassen. Drittens ist es möglich, mit Hilfe der Externbefehle spezielle Signale abzugeben, wie z. B. ein Start-Signal für eine Analog-Digital-Konvertierung oder Schritimpulse für Stellmotore. Die zwischen solchen Schritimpulsen nötigen Pausen werden durch Programmschleifen oder Zeitzählung realisiert. Im zweiten Fall wird zwischen das Meßgerät und die Rechenmaschine eine Steuerung geschaltet, welche langdauernde Vorgänge

übernimmt - wie die Abgabe einer gewissen Zahl von Schrittimpulsen mit einem festgelegten Zeitabstand, oder eine Zeitzählung. Diese Steuerung führt nach Übergabe der nötigen Parameter ihre Arbeit selbständig aus, und meldet sich am Ende der Arbeit an die Rechenmaschine zurück.

Der zweite Fall des Betriebs eines von einer Rechenmaschine gesteuerten Meßgeräts ist nur ähnlich der einer Datenübertragung, bei der die Rechenmaschine Parameter übergibt, und die ganze weitere Arbeit der Datenübertragungssteuerung überläßt. Ein echtes Analogon zu dieser Datenübertragung müßte für den Betrieb eines Meßgeräts wie folgt aussehen: Die Rechenmaschine bereitet in einem Teil ihres Speichers Informationen für eine Messung oder Meßreihe vor, und reserviert einen weiteren Teil des Kernspeichers für die bei dieser Meßreihe zu erwartenden Daten; Die Rechenmaschine überträgt die Anfangsadressen der beiden Teile des Speichers an ein spezielles Interface und startet dieses; das Interface verarbeitet die bereitgestellten Informationen, steuert damit die gewünschten Abläufe und Messungen, überträgt die anfallenden Meß- und Zustandsdaten in den angegebenen Teil des Kernspeichers der Rechenmaschine, und unterbricht die Rechenmaschine am Ende der auszuführenden Meßreihe. Ein solches Interface verfügt über zwei Adreßregister: eines für die abzuliefernden Meßdaten, und ein zweites für die zu verarbeitenden Informationen. Weiter muß dem Interface von irgendwoher mitgeteilt werden, was es mit den zur Verfügung gestellten Informationen anfangen soll. Kommen diese zusätzlichen Mitteilungen von außen über fest verdrahtete oder steckbare Ablaufprogramme, so kann man sich leicht vorstellen, daß ein solches Interface nur wenige starre Meßabläufe ausführen kann, oder äußerst aufwendig und kompliziert wird.

Zur Verwirklichung eines Interfaces, bei welchem die Rechenmaschine durch Übertragung von nur zwei Parametern (Adressen) jeden gewünschten Einstellungs- und Meßablauf ausführen lassen kann, wurde an unserem Institut ein etwas anderer Weg gewählt. Um dem Interface mitzuteilen, was es mit den zur Verfügung gestellten Informationen anzufangen hat, besteht

die einfache Möglichkeit, den für den Einstell- und Meßablauf nötigen Informationen einen zusätzlichen Code mitzugeben, dem das Interface entnehmen kann, was mit der speziellen Information zu geschehen hat. Verfolgt man diesen Gedanken weiter, so erscheint es zweckmäßig, dem Meßgerätinterface nicht nur richtungsgekennzeichnete Informationen, sondern ein Programm aus Befehlen und Befehlen plus Information zur Bearbeitung anzubieten. Dazu muß ein Teil des Meßgerätinterfaces in ein Leitwerk umgewandelt werden. In der Rechenmaschinensprache ist ein Leitwerk ein Teil des Rechners, der die Abarbeitung des gespeicherten Programmes steuert. Das Leitwerk des Interfaces - siehe Bild 1 - setzt sich zusammen aus dem Befehlszähler - dem umfunktionierten Adreßregister für die zu verarbeitenden Informationen -, dem Befehlsregister mit -dekodierung und -ausführung, und einem Schleifenzähler für sinnvolle Wiederholungen von Programmteilen. Vor dem Start einer auszuführenden Arbeit wird der Befehlszähler, sowie auch das Meßdatenadreßregister per Externbefehl von der Rechenmaschine geladen. Daraufhin arbeitet das Leitwerk ein Programm ab, das von der Rechenmaschine, an die das externe Leitwerk angeschlossen ist, vorbereitet wurde, und im Kernspeicher dieser Rechenmaschine steht. Der Befehlszähler, der am Anfang von der Rechenmaschine direkt geladen wurde, kann dabei von der Befehlsdekodierung und -ausführung auch umgeladen werden, was einen Sprung von einer Stelle des Programms zu einer anderen bedeutet. Das Leitwerk wird ergänzt durch Verstärker zur Versorgung des Meßgeräts, Signalaufnehmer vom Meßgerät, dem Meßdatenaufnehmer - wenn nötig mit Meßdatenregister -, dem Meßdatenadreßregister und der Speichersteuerung. Die Speichersteuerung versorgt das Befehlsregister, und übernimmt auf Anstoß durch den Befehlszähler das Abspeichern der Daten.

Es soll nun der Befehlscode, mit dem ein Leitwerk ein Meßgerät steuern kann, etwas näher erläutert werden. Prinzipiell gibt es - wie auch die später erwähnten Anwendungsbeispiele zeigen - beliebig viele Möglichkeiten, den Befehlscode eines solchen Leitwerks festzulegen. Im folgenden

wird versucht, ein Codemodell anzugeben, das für Befehle, die viel Information benötigen, auch möglichst viele Informationsbits zur Verfügung stellt - Bild 2. Der Sprung-Befehl - Laden des Befehlszählers - wird angewendet zusammen mit dem Befehl "Schleifenzähler Laden" - Befehl Nr. 2 - und anderen Befehlen, die den Schleifenzähler inkrementieren und abfragen. Der Schleifenzähler - eine Art Indexregister - wird mit dem Zweierkomplement der gewünschten Wiederholungen einer Programmschleife geladen. Der dritte Befehl "Zähle Zeit" dient zur echten Zeitzählung oder zur Bildung nötiger Pausen. Er unterbricht die Arbeit des Leitwerks und startet eine elektronische Uhr. Die Arbeit des Leitwerks wird erst dann fortgesetzt, wenn die im Informationsteil des Befehls als Zweierkomplement dargestellte Zahl von Taktimpulsen eingelaufen ist. Dazu muß der dem Informationsteil des Befehls entsprechende Teil des Befehlsregisters als Zähler aufgebaut sein. Zur exakten Zeitmessung kann bei dem Befehl "Zähle Zeit" ein Pegel abgenommen werden, der mit dem ersten Takt der Quarzuhr nach dem Befehl erscheint, und mit dem ersten Takt nach Nullwerden des Informationsteils des Befehlszählers verschwindet. Bei den folgenden Befehlen - vier bis sieben - dienen die Informationsbits teilweise oder ganz zur Modifizierung des Operationsteils. Mit Sammelbefehl vier können Einzelbefehle entschlüsselt werden, wie z. B. Auslösung einer Analog-Digital-Konvertierung, Rücktransport von Informationen an den Rechner, oder der Befehl "Ende des externen Programms", der zur Unterbrechung der Rechenmaschine führt. Befehl fünf wird zur Ladung gewisser Register des Meßgeräts (z. B. für Meßbedingungen) herangezogen. Der sechste Befehl dient zur Programmverzweigung. Ist eine der im Informationsteil angegebenen kombinierbaren Bedingungen - z. B. Begrenzungskontakte und Nullkontakte von Mechanismen des Meßgeräts, oder der Zustand Null des Schleifenzählers erfüllt, so wird der nächste im Programm folgende Befehl übersprungen. Dem "Überspringe-Befehl" müssen im Allgemeinen zwei Sprungbefehle folgen. Läßt es die Wortlänge des Speichers der Rechenmaschine zu, kann der Befehl UB auch in einen Befehl SB - springe bei Bedingung - umge-

wandelt werden. Der letzte Sammelbefehl - Nummer sieben - wird zur Erzeugung kombinierbarer Befehle und Signale herangezogen, da solch eine Kombinierbarkeit zur Steuerung gewisser Abläufe sehr günstig sein kann. Solche Befehle können sein: Einschalten von Schrittmotoren in die verschiedenen Richtungen, Stellimpulse für diese Schrittmotoren, Inkrementierung des Schleifenzählers usw.

Zur Erläuterung sei ein einfaches Programmbeispiel für ein externes Leitwerk, welches ein Meßgerät steuert, angegeben - siehe Bild 3. Es soll ein Schrittmotor um 1000 Schritte verstellt werden. Der Motor wird eingeschaltet über "KB + 0" und der Schleifenzähler mit dem Zweierkomplement von 1000 geladen (SL - 1000). Anschließend werden der Schleifenzähler und die Endkontakte abgefragt. Ist der Schleifenzähler ungleich Null, und ist kein Endkontakt des Kreises eingeschaltet, so wird der UB folgende "Sprung zum Ende-Befehl" übersprungen. Bei der Weiterbearbeitung des Programms erhält der Schrittmotor einen Impuls zur Weiterstellung, zugleich wird der Schleifenzähler inkrementiert - (KB). Anschließend wird 5 ms lang gewartet. Nach der Pause erfolgt ein Rücksprung nach "NSCHR" (nächster Schritt). Diese Programmschleife wird solange durchlaufen, bis im günstigen Fall der Schleifenzähler zu Null wird, oder im ungünstigen Fall ein Begrenzungskontakt anspricht. Tritt einer dieser Fälle ein, springt das Leitwerk nach "FIN", überträgt die Zustandsinformation des Kreises an die Rechenmaschine (EB), schaltet den Motor aus (KB) und unterbricht die Rechenmaschine (EB).

Man könnte nun einwenden, daß die vom Leitwerk mit Hilfe des oben angegebenen Programms geleistete Arbeit genau so gut von einer Steuerung geleistet werden könnte, bei der die Rechenmaschine den Motor einschaltet, einen Zähler mit dem Zweierkomplement der gewünschten Schrittzahl ladet, und einen Oszillator für die Schritimpulse startet, und bei der die Rechenmaschine von der Steuerung unterbrochen wird, wenn der Zähler, der die Schritimpulse registriert, nach Null gelaufen ist. Was diese Steuerung

nicht kann, jedoch ein Interface mit Leitwerk kann, ist, daß ein Leitwerk beliebig viele und beliebig verschiedenartige solcher Arbeiten nacheinander durchführen kann, ohne den Rechner zu unterbrechen. Im Gegensatz zu obigem Programmbeispiel wird im Normalfall der Rechner nur bei einer Störung unterbrochen, ansonsten das Programm, von dem das Beispiel nur ein kleiner Ausschnitt ist, fortgesetzt. Weiter wird bei der oben genannten Steuerung das Nullwerden des Zählers und der Kontaktzustand des Meßgeräts durch eine starre Hardware verarbeitet. Die Abfrage der Kontakte des Meßgeräts und des Standes des Schleifenzählers per Befehl eines Leitwerks ist demgegenüber logisch wesentlich übersichtlicher und daher auch betriebssicherer.

Es werden nun zwei Anwendungsbeispiele für ein Meßgerät, das über ein Leitwerk an eine Rechenmaschine angeschlossen ist, gezeigt. Die beiden Anwendungsfälle sind zeitlich älter als das aufgeführte Modell, und können - das erste mehr - das zweite weniger - als Vorstufen zu diesem Modell betrachtet werden.

Das erste Anwendungsbeispiel ist der Anschluß eines Röntgendiffraktometers der Fa. Siemens /1/ an eine Rechenmaschine IBM 1130. Das Interface wurde an unserem Institut entwickelt und von der Fa. Graphic display (London) gebaut /2/. Die Anlage ist seit ca. eineinhalb Jahren in Betrieb an der Universität St. Andrews in Schottland - siehe Bild 4. Beim Entwurf dieses Interfaces wurde von einer herkömmlichen Diffraktometersteuerung, die einzelne Operationen selbständig ausführen kann, ausgegangen - links oben in Bild 4. Sie wurde nicht unmittelbar an die Rechenmaschine angepaßt, sondern zwischen Steuerung und Rechenmaschine wurde ein Leitwerk eingefügt. Da die Diffraktometersteuerung der Anlage, an die sie angeschlossen ist, nach Beendigung einer Einzeloperation ein Interruptsignal sendet, wurde dem Leitwerk eine zusätzliche Steuerung namens "Diffraktometerinterrupt" beigelegt. Bei

einem Interrupt seitens der Diffraktometersteuerung prüft diese Steuerung den Zustand des Diffraktometers. Bei positivem Ausgang dieser Prüfung wird das Leitwerk zur Weiterarbeit angestoßen, bei negativem Ausgang die Rechenmaschine unterbrochen. Es werden also bei dieser Anschlußart von der Operationssteuerung mit Kontaktstop und vom "Diffraktometerinterrupt" noch Arbeiten ausgeführt, die eigentlich Aufgaben der Befehlsausführung sind. Weiter müßte dem Modell entsprechend der Operationszähler ein Teil des Befehlsregisters sein. Daher lautet die Befehlsliste dieses Interfaces etwas anders als die des Modells; die vom Modellfall abweichenden Befehle sind durch kleine Buchstaben in den Befehlsabkürzungen markiert. Daß der Befehl Programmende ein separater Befehl ist, und daß mehrere Befehle zum Laden von Registern vorhanden sind, ist nur von nebensächlicher Bedeutung. Der Befehl sop ersetzt den Befehl ZZ (Zähle Zeit), und wird auch zur Abgabe einer gewünschten Zahl von Stellimpulsen für Schrittmotore herangezogen. Durch sop wird der Operationszähler geladen und die Operationssteuerung gestartet. Weiter ist nur ein Sprungbefehl SB - Sprünge, wenn Schleifenzähler ungleich Null, - vorhanden, der als Doppelwortbefehl ausgeführt ist, da alle 16 Bits des Rechnerwortes zur Adressierung benötigt werden. Auch ist ein Befehl zur Inkrementierung des Schleifenzählers nicht vorhanden. Der Schleifenzähler wird bei jedem Interrupt seitens der Diffraktometersteuerung (Ende einer Einzeloperation) um eins erhöht.

Das zweite Anwendungsbeispiel ist der Anschluß eines Photometers zur Auswertung elektronenmikroskopischer Aufnahmen an eine Siemens 305 Rechenmaschine. Das Photometer besteht aus zwei durch Schrittmotore angetriebene Schlitten, mit denen der zu vermessende Film in der X und Y Richtung bewegt werden kann - siehe Bild 5. Die Schwärzungen des Films werden über Verstärker und einen Analog-Digital-Konverter gemessen. Der einzige Unterschied zum Modell eines von einem Leitwerk gesteuerten Meßgerät liegt darin, daß für den X-Motor ein "High speed Preset Indexer" der Fa. Superior

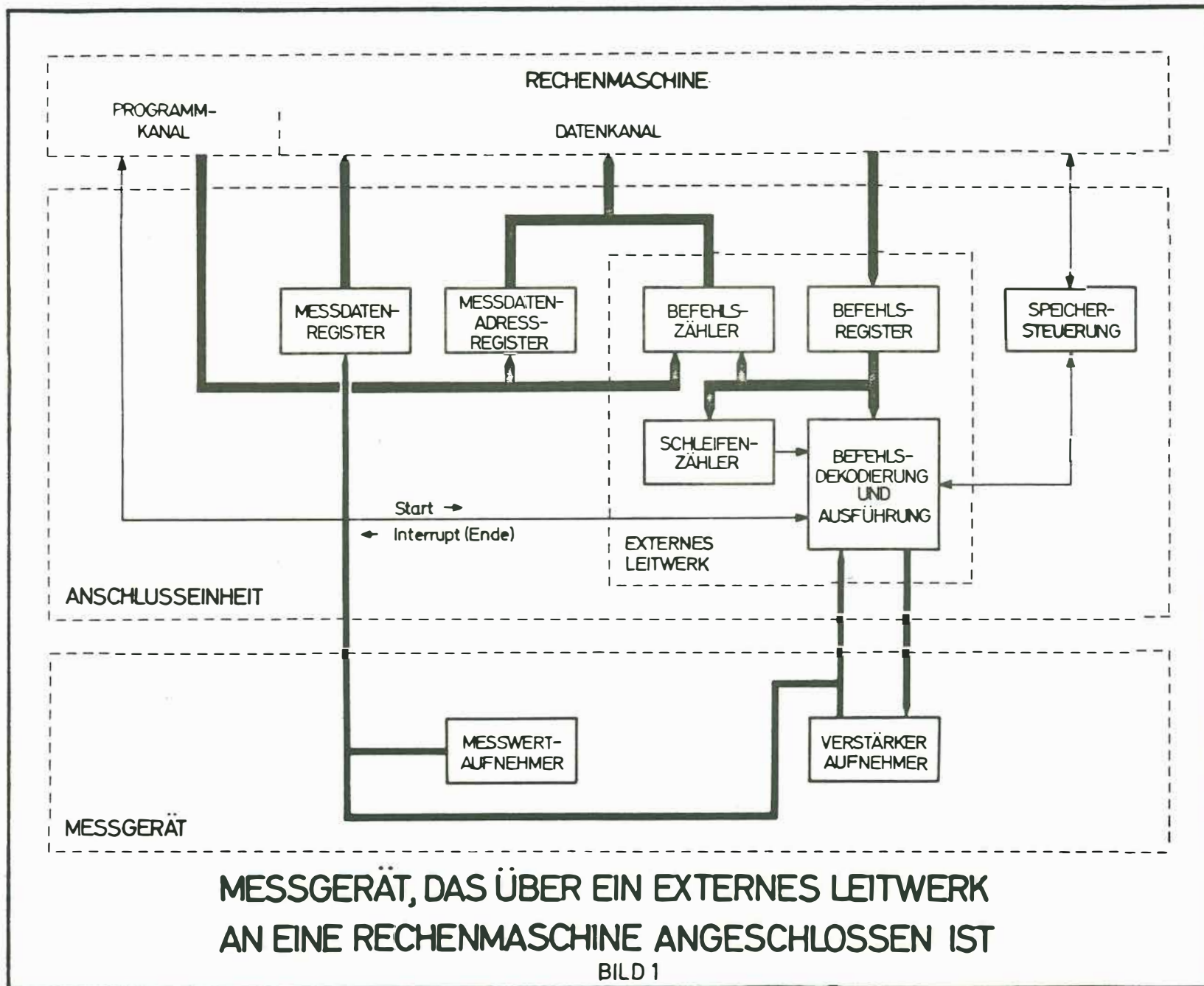
Electric verwendet wird, der es gestattet, einen Schrittmotor mit der Frequenz von 2 KHz zu betreiben, und daß der Schleifenzähler auch als Puffer für diesen Indexer Verwendung findet. Die Messungen und die Abspeicherung der Meßdaten erfolgen entweder über den Befehl KB (Einzelmessung), oder laufen im Zustand "Indexermessung" synchron zu den Schrittpulsen des Indexers. Obwohl mehrere Bedingungen abzufragen sind, wird, da es die Wortlänge der Rechenmaschine erlaubt, nicht der Befehl "Überspringe bei Bedingung", sondern der Befehl "Springe bei Bedingung" eingesetzt. Die zusätzliche Bedingung "Schleifenzähler nicht BCD (binärdekadisch)" wird bei Verwendung des Schleifenzählers als Puffer für den Indexer benötigt, um einen Stop des Indexers zu gewährleisten. Das Meßdatenadreßregister wird bei diesem Interface vom Leitwerk geladen. Vom Programmkanal kommt nur ein Start-Befehl zur Anwendung, wobei das Interface einen Sprung zum Programmanfang aus der beim System Siemens 300 häufig verwendeten Anzeigen-Zelle in das Befehlsregister des Leitwerks ladet. Der Preset Indexer beschleunigt einen Schrittmotor auf 2 KHz, und bremst ihn vor dem Ende der einzustellenden Schrittzahl auf die zum Anhalten nötige Frequenz herab. Durch Einführen von Sprungbefehlen, bei welchen der Inhalt des Schleifenzählers zur Sprungadresse addiert oder von dieser subtrahiert werden kann, wäre es auch möglich, die Beschleunigungs- und Bremsvorgänge, die der Preset-Indexer selbständig ausführt, dem Leitwerk zu übertragen.

Zusammenfassend kann gesagt werden: Die Arbeit des Meßgeräts mit Hilfe des über das Leitwerk abgearbeiteten dynamischen Programms ist genau so flexibel, als wenn das Meßgerät direkt per Externbefehl von der Rechenmaschine gesteuert wird, belastet aber die Rechenmaschine programmseitig in keiner Weise. Auch zeitlich werden andere Arbeiten, die der Rechner auszuführen hat, kaum beeinträchtigt, da viele logische und arithmetische Operationen des Rechenwerks länger dauern als ein Speicherzyklus, und somit Speicherzyklen für das Leitwerk zur Verfügung stehen. Wegen der geringen Belastung der Rechenmaschine durch ein über ein Leitwerk angeschlossenes Meßgerät ergibt sich die Möglichkeit ohne größeren Zeitverlust

für die Messung Arbeitsprogramme für eine größere Anzahl von Meßreihen vorauszurechnen, und diese einschließlich des Programms, das sie errechnete, auf einem Externspeicher abzulegen. Ein Minimalprogramm zur Grobkontrolle der Meßdaten, zum Abtransport der Meßdaten, und zum Bereitstellen neuer Arbeitsprogramme genügt zum eigentlichen Betrieb des Meßgeräts, wodurch der Platzbedarf für die Programme, die das Meßgerät steuern, erheblich reduziert wird. Diffraktometer und Photometer sind relativ einfache Meßgeräte. Man kann dies daran erkennen, daß in den Leitwerken, die sie steuern, nur ein Bruchteil der möglichen Befehle ausgenützt wird. Es ist anzunehmen, daß die Steuerung über ein Leitwerk auch für kompliziertere Meßgeräte und andere Geräte Vorteile bringen kann. Abgesehen davon entspricht die Steuerung eines Meßgeräts über ein Leitwerk als parallel arbeitende Externeinheit dem System der parallel rechnenden Untereinheiten, wonach zur Zeit die modernsten Rechenmaschinen aufgebaut werden.

Schrifttum

- /1/ E. A. Mayer: Über ein neues Gerät zur automatischen Messung der Intensitäten von Einkristallreflexen nach W. Hoppe, Zeitschrift für analytische Chemie, Band 205 (1964), Seite 153 - 164.
- /2/ HP. Englmeier und W. Hoppe: Automation der Datensammlung in der Röntgenstrukturanalyse. messtechnik 11 (1969) Seite 265 - 271.



CODEBAUM

CODE

BEDEUTUNG

ABKÜR-
ZUNGLÄNGE DES
INFORMATION-
TEILES

BIT 1 2 3 4 5 6 ... m

NR.

0	0XXXXX... X	SPRUNG	SP	m - 1	1
1	10XXXX... X	SCHLEIFENZÄHLER LADEN	SL	m - 2	2
11	110XXX... X	ZEIT ZÄHLEN	ZZ	m - 3	3
111	11100X... X	EINZELBEFEHLE	EB	m - 5	4
111	11101X... X	REGISTER LADEN	RL	m - 5	5
111	11110X... X	ÜBERSPRINGE BEI BEDINGUNG	UB(SB)	m - 5	6
111	11111X... X	KOMBINIERBARE BEFEHLE	KB	m - 5	7

BEFEHLSCODEMODELL EINES LETTWERKES FÜR EIN MESSGERÄT

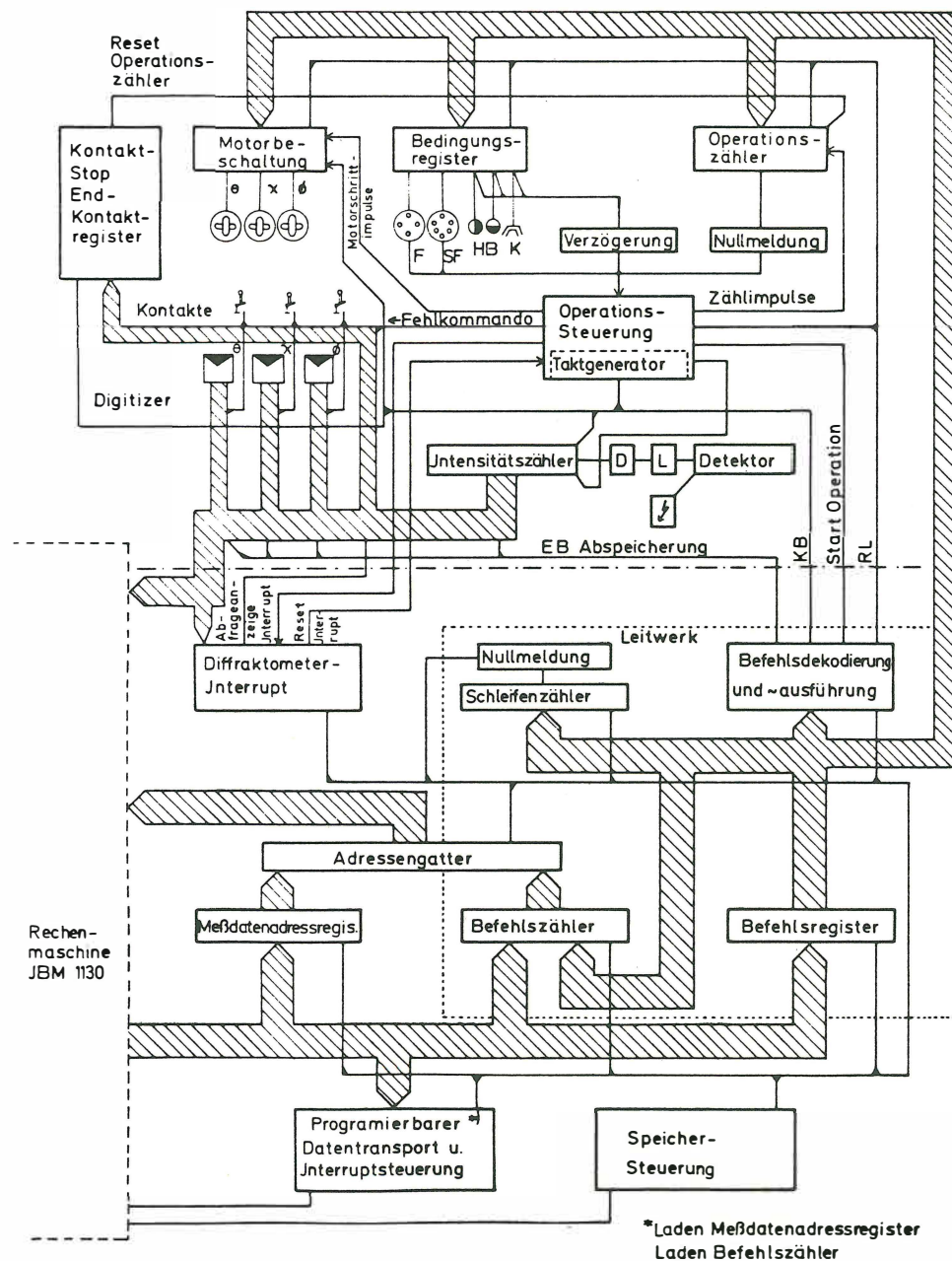
BILD 2

Einfaches Programmierungsbeispiel für ein externes Leitwerk

Verstellung des θ Kreises eines Diffraktometers um 1000 Schritte

	KB	+ θ
	SL	-1000
NSCHR	UB	wenn Schleifenzähler $\neq 0$, kein θ Kreis Endkontakt
	SP	FIN
	KB	Motorschritt θ , increment Schleifenzähler
	ZZ	Pause 5ms
	SP	NSCHR
FIN	EB	θ Kreis Kontaktzustand abspeichern
	KB	θ Motor aus
	EB	Ende Interrupt

BILD 3



Abkürzungen

F Filter
SF Schwächungsfilter
HB Halbblenden
K Kupplung $\theta-2\theta$
D Diskriminator
L Linearverstärker

Röntgendiffraktometer (Siemens)
gesteuert über ein Leitwerk
an einer Rechenmaschine (JBM 1130)

Befehlsliste

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15								
Informationsteil												Operations- teil											
															0	1	0	1	0	end			
															0	1	0	0	1	sop			
															0	1	0	1	0	KB			
															0	1	0	0	0		0	1	0
															0	1	0	0	0		0	1	0
															0	0	1	0	0		0	1	0
															0	0	0	1	0		0	1	0
															0	0	0	0	1		0	0	1
															0	0	1	1		EB			
															0	1	0	0	0		0	1	1
															0	1	0	0	0		0	1	1
															0	0	1	0	0		0	1	1
															0	0	0	1	0		0	1	1
															0	0	0	0	1		0	1	1
															0	1	0	0	0	SB			
															1	0	0	0	0	SL			
															1	0	0	1		RLm			
															1	0	1	0		RLh			
															1	0	1	1		RLf			
															1	1	0	0		RLs			

Programme Interrupt

Start Operation Zeit- oder Winkelinkrement-
zählung

Kombinierbare Befehle

Motorschritt

Endkontaktstop

Nullkontaktstop

Fahren Vorbereitung

Reset Intensitätszähler

Einzelbefehle Abspeicherung

Intensitätszähler Bi+0-11

Intensitätszähler Bi+12-23

Position θ

Position x

Position ϕ

Springe wenn Schleifenzähler $\neq 0$
Adresse im zweiten Wort

Schleifenzähler Laden

Register laden Motorbeschaltung

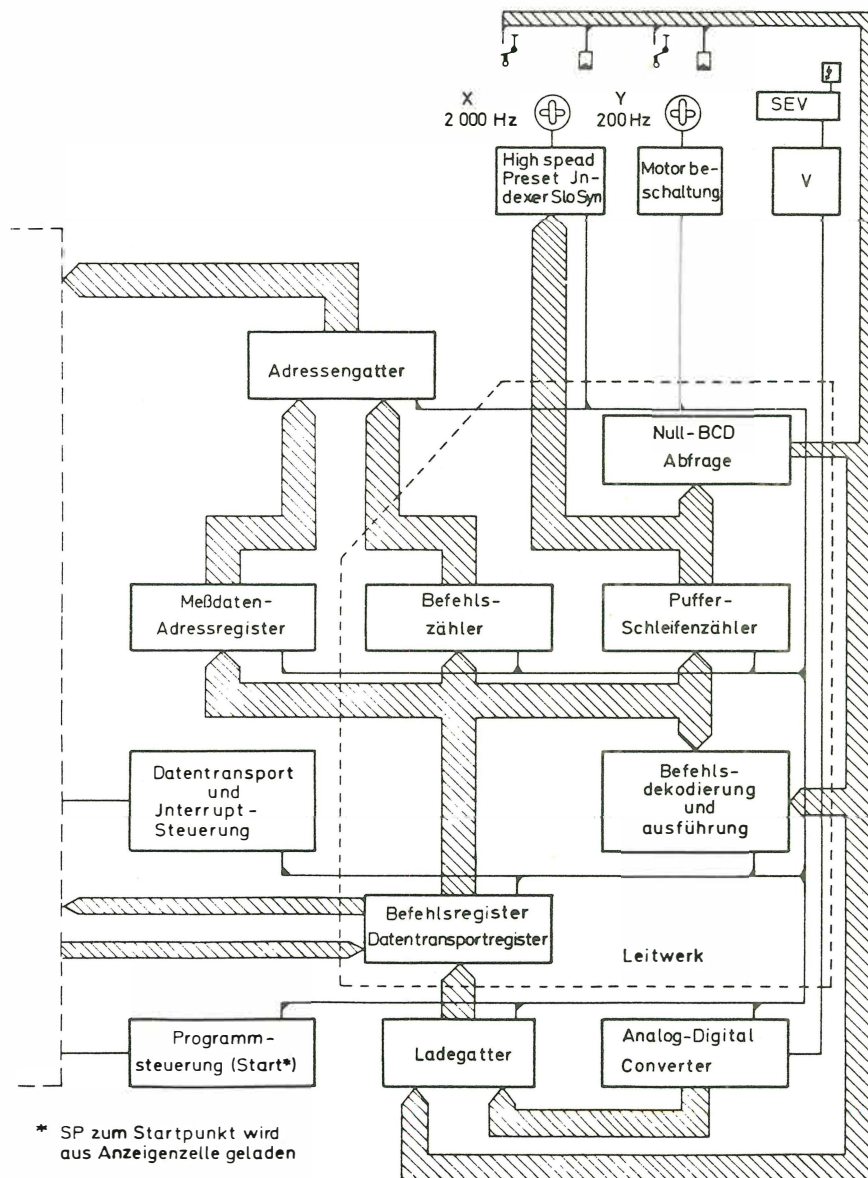
Register laden Halbblenden Kupplung

Register laden Filter

Register laden Schwächungsfilter

*Laden Meßdatenadressregister
Laden Befehlszähler

BILD 4



* SP zum Startpunkt wird aus Anzeigezelle geladen

Photometer gesteuert über ein Leitwerk an einer Rechenmaschine (Siemens 305)

Befehlsliste

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24		
Informationsteil																					Operations- teil				
																					0, 0, 0			end	Programmende Interrupt
x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁																					0, 0, 1			SP	Sprung
x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁																					0, 1, 0			lm	Laden Meßdatenadrefregister
x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁																					0, 1, 1			SL	Schleifenzähler Laden (binär - als Puffer für Indexer binärdekadisch)
x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁																					B, B, B, B, B, B, B, B, 1, 0, 0			SB	Springe bei Bedingung x Adresse B Bedingungen
																					Schleifenzähler nicht BCD Schleifenzähler Null X Motor wählen Y Motor wählen Endkontakt Nullkontakt Nullpunkt				
x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁ x ₁																					1, 0, 1			KB	Kombinierbare Befehle
Einzelmessung Reset Indexer Reset Kontakt flip-flops Messen Vorbereitung Fahren Vorbereitung Increment Schleifenzähler Motorschritt X Motorschritt Y Motoren aus X Motor an + X Motor an - Y Motor an + Y Motor an -																									
x ₁ x ₁ x ₁																					1, 1, 0			ZZ	Zähle Zeit (150µs bis ca 5ms)
																					1, 1, 1			in	Starte Indexer

BILD 5

Kevecordes
Ingenieurschule Paderborn

"Hardware Pläne eines möglichst optimalen
Prozeßrechnereinsatzes an der Ingenieur-
schule Paderborn"

Hardware Pläne eines möglichst optimalen Prozeßrechnereinsatzes an der Ingenieurschule Paderborn

1967 entstanden die ersten konkreten Pläne für den Prozeßrechnereinsatz, nachdem schon etwa 1 bis 2 Jahre vorher vom Kultusministerium in Düsseldorf festgelegt worden war, daß an der Staatl. Ingenieurschule für Maschinenwesen in Paderborn ein Schwerpunkt für den Prozeßrechnereinsatz entstehen sollte. In der Zwischenzeit stand mehrfach eine Zuteilung der zum Aufbau des Systems erforderlichen Mittel bevor, leider kam es jedoch in "letzter Sekunde" immer wieder zu einem zeitraubenden Aufschub bei der Zuteilung.

Im Folgenden möchte ich über das Hardware-Konzept des Prozeßrechneranschlußsystems sprechen, das in diesen Tagen im Neubau unserer Schule installiert wird. Zu unserem Bedauern kann aus oben angeführten Gründen der für dieses System erforderliche Großprozeßrechner gleichzeitig noch nicht mit eingebaut werden. Wir können daher nur an einer kleinen, allerdings großzügig vom Haus Siemens überbrückungsweise zur Verfügung gestellten Prozeßperipherie der 303 arbeiten.

Zu Beginn der Planung für die Errichtung des Prozeßrechnerschwerpunktes an unserer Schule war zunächst eine Bedarfsabschätzung in Bezug auf folgende Punkte durchzuführen:

- 1) Einsatzgebiete eines Prozeßrechners im Rahmen der Möglichkeiten unserer Schule;
- 2) Anzahl der an jeder einzelnen Anwendungsstelle an den Prozeßrechner anzuschließenden Prozeßein-/ausgabewerte;
- 3) Gleichzeitigkeitsfaktor für einen Simultanbetrieb des Prozeßrechensystems.

Zum Punkt 1) ergaben sich an zunächst 18 Stellen Einsatzmöglichkeiten für eine Prozeßführung. Aus der folgenden Tabelle sind die Einsatzgebiete und jeweils an diesen Stellen zunächst in Angriff zu nehmende Probleme zu ersehen:

Lack Großlabor	Steuerung einer Lackiereinrichtung und eines Infrarot-Umluft-Durchlaufofens
Kunststofflabor	Extrudersteuerung
Herstellungstechnologie	Steuerung einer Lackstraße; Steuerung einer elektrophoretischen und einer elektrostatischen Lackauftragsanlage

Labor für Kraft- und Arbeitsmaschinen	u.a. Steuerung eines Ottomotorenprüfstandes (Prüfstand vorhanden)
Werkzeugmaschinen-Labor	NC-Steuerungen; (später auch on line)
Physik-Labor	Kennlinienaufnahme; statistische Auswertung von Versuchen
Isotopen-Labor	Meßverfahren unter Einsatz von Isotopen; Spektrometer-Analysen
Chemie Infrarot-Gaschromatographie	Automatische Analysensteuerung und Auswertung
Labor für Regelungstechnik	(besonderer Prozeßrechner-Schwerpunkt) Regelmodelle; Prozeßsimulationen, Optimierungsprobleme, DDC-Versuch
HF-Labor	Antennennachlaufsteuerungen
Meßtechnik-Labor	Datenerfassungsprobleme, Erd-schleifen und Schirmungsprobleme
Theoretische Elektrotechnik	komplizierte Berechnungen im on line open loop-Betrieb
NF-Labor	off line Probleme; on line open loop-Probleme
Labor für Elektrische Maschinen	Laborversuche; Kennlinien- und Kenndatenermittlung
Labor für Elektronische Antriebe	Maschinentests
Labor für Prozeßrechensysteme	Spezielle Untersuchungen und Demonstration von Prozeßrechensystemen. Kopplung mit Analogrechner
Konstruktionsraum	Plotter

Nach den Einsatzpunkten nun zu 2), den erforderlichen E i n / A u s - g a b e w e r t e n an den jeweiligen Nahtstellen. Da aus Kosten-

gründen eine Einheitsnahtstelle angestrebt werden mußte, war eine Anpassung an den Bedarf und eine entsprechende Koordination unerläßlich. Folgender Bedarf schälte sich heraus:

- 60 Binäreingänge (Untermenge von 20 als Interrupts)
 - 40 Analogeingänge
 - 20 Binärausgänge
 - 12 Analogausgänge
 - 1 Spezialerdungsanschluß
 - 1 BBS-Anschluß
- zusätzlich eine bedingte Rangiermöglichkeit,
um flexibel bei bestimmten Mehranforderungen
zu sein.

Die Abschätzung des Gleichzeitigkeitsfaktors nach Punkt 3) zeigte folgendes Ergebnis: Mit einem Simultanbetrieb an 6 von 18 Einsatzpunkten ist ein zufriedenstellender Betrieb zu erreichen. Eine entsprechend zu organisierende Koordination und interne Absprache wird zu einer guten zeitlichen Durchmischung aller 18 Verbraucher führen. Aus Kostenersparnisgründen mußte der Gleichzeitigkeitsfaktor so niedrig wie nur eben möglich gehalten werden. Keinerlei Einschränkung in dieser Hinsicht wäre uns natürlich lieber gewesen.

Aus den aufgeführten Bedingungen für den Prozeßrechnereinsatz ergab sich folgendes Konzept:

Einsatz eines Prozeßrechners in einem Rechenzentrum konventionellen Stils, wo alle Prozeßaufgaben körperlich zum Rechenzentrum kämen, war unmöglich.

Die Alternative - der Einsatz vieler z.T. kleiner Prozeßrechner an den Anwenderstellen - schied aus Kostengründen ebenfalls aus.

Es galt also, eine Möglichkeit für eine Art Prozeßrechner time-sharing-Betrieb zu finden. Dazu haben wir u.a. das hier zu beschreibende hardware Prozeßrechner-Anschlußsystem entwickelt. Es besteht aus den 18 Terminals, die die unter Punkt 2) aufgeführten Prozeß Ein/Ausgaben an den Rechner anschließen. Für die Einheitsnahtstelle wurde ein Standard-Terminal, die sogenannte Rechneranschlußtafel, entwickelt. Dieses nach Art von Experimentiertafeln aufgebaute Terminal mit den Abmaßen 100 x 60 cm der Frontplatte, enthält in übersichtlicher Form die Anschlüsse für den Rechner. Die 80 Binär- bzw. Alarm-Ein/Ausgänge sind doppelpolig ausgeführt und können somit potentialfrei übertragen werden.

Um 4-Leiterschaltungen möglich zu machen und eine Schirmerdung vornehmen zu können, sind alle Analogeingaben 5-polig ausgeführt; die Analogausgabewerte sind zweipolig durchgeschaltet. Außerdem enthält die Tafel den Anschluß für die Spezialerde und den Bedienungsblattschreiber. Die Frontplatte der auf die Wand aufbaubaren Tafel ist

schwenkbar montiert, alle Werte sind übersichtlich angeordnet und mit entsprechenden Adressen versehen. Eine 220 V Stromversorgung ist nicht in die Tafel eingebaut; die BBS werden 220 V-seitig nicht von der zentralen Rechnerstromversorgung betrieben. An diese Tafeln kann in labormäßigem Aufbau ein Prozeß angeschlossen und vom Rechner gesteuert werden.

Für die Realisierung des Anschlusses der Rechner-Anschluß-Tafeln an das Prozeßbleitelement (PLE) des Prozeßrechners gab es grundsätzlich drei Möglichkeiten.

1. Die direkte Verkabelung jeden Wertes an das PLE

Vorteil: geringste Störanfälligkeit, gute Übertragungsbedingungen, feste Adressierbarkeit, dauernder Anschluß am Rechner.

Nachteil: bei Entfernungen von bis zu jeweils 150 m hohe Kabelkosten; großer Ausbaugrad des PLE

2. Rechneranschluß der Tafeln über ein Fernwörksystem

Vorteil: Minimum an Kabelkosten, keine Probleme bei Erweiterungen, geringe Störungen durch Einstreuungen bei der Datenübertragung.

Nachteil: hohe Kosten für die Übertragungsgeräte, da hohe Datenübertragungsraten erforderlich sind;
Koordination an den Nahtstellen bei Zusammenarbeit mit dem Rechner;
eine hohe zu erwartende Störanfälligkeit der Geräte

3. Rechneranschluß über ein Kabelverteilersystem

Voraussetzungen:

Gleichzeitigkeitsfaktor < 1
eine Einteilung in Gruppen von Benutzern
muß möglich sein,
je Gruppe kann nur ein Benutzer mit dem Rechner verbunden sein,
Simultanarbeit nur für jeweils einen Benutzer jeder Gruppe möglich

Vorteil: günstiger Kostenkompromiß, übersichtlicher als Lösung 2, geringerer Ausbaugrad des PLE

Nachteil: eingeschränkte Flexibilität gegenüber 2 und 1 (wirkt sich nicht so sehr aus, da zumindest zunächst der Gleichzeitigkeitsfaktor sicher weit unter 1 sein wird);
Umschaltungszeit; Organisation erforderlich

In der Auswahl der Verfahren haben wir uns beraten (vor allem auf technische Durchführbarkeit unserer Ideen) durch das Haus Siemens, das das Verteilersystem liefert, aus Kostengründen für eine Lösung nach 3 entschlossen.

Das Verteilersystem besteht aus einem an das PLE angeschlossenen und im Prozeßrechenzentrum stationierten Hauptverteiler (HV). In diesem HV, der in Schränken des Systems 300 untergebracht ist, erfolgt durch

Rangieren mehrpoliger Steckverbindungen die sternförmige Verteilung der Anschlüsse an das PLE.

Die nach Analog- und Binärwerten getrennten, abgeschirmten und paarweise verdrehten Sammelkabel führen zu 5 Unterverteilern (UV).

Diese UV bestehen wie der HV aus in Rechnerschränken untergebrachten Rangiervorrichtungen durch manuell wählbare Steckverbindungen. An jedem UV hängt prozeßseitig je eine Gruppe von Terminals. Die Zuordnung dieser Terminals zu einem UV wurde nach gebäudetechnischen, belastungsmäßigen und kabelkostensparenden Gesichtspunkten vorgenommen. Durchgeschaltet wird am UV genau wie am HV durch Einlegen bestimmter Steckverbindungen.

Um zusätzliche Steckverbindungen innerhalb der UV zu sparen, ist die Rangierung konstruktiv so gelöst, daß nur jeweils eine Steckverbindung pro UV und durchzuschaltende Leitung zustandekommt. Das ist unbedingt erforderlich, da trotz der Verwendung von Goldkontakten die durch Kontaktwiderstände und Thermospannungen auftretenden Störungen auf ein Minimum reduziert werden müssen. An den vielpoligen Steckern ist ein flexibles Kabel angelötet, das - unterstützt durch spezielle mechanische Vorrichtungen - ein übersichtliches Durchschalten der Terminals gewährleistet.

Sollte der Bedarf an Ein/Ausgaben der Standardtafel für ein spezielles Problem nicht ausreichen, so kann man an dem entsprechenden UV und dem zugehörigen Teil des HV in gewissen Grenzen die Zahl der Anschlüsse variieren.

Mit Hilfe des eben geschilderten Hardware-Rechneranschlußsystems wollen wir den Prozeßrechner in seinen vielfältigen Einsatzmöglichkeiten verwenden für

- 1) den Vorlesungsbetrieb (für Demonstrationsversuche)
- 2) für den Laboratoriumsbetrieb
- 3) (hauptsächlich) für das ingenieurmäßige Arbeiten

Die gemachten Erfahrungen in hard- und software sollen auch anderen Schulen zur Verfügung gestellt werden. Außerdem ist an Kurse für interessierte Dozenten gedacht, wie sie jetzt bereits für den Rechnereinsatz im technisch-wissenschaftlichen Bereich (für Prozeß nur am Rande) stattgefunden haben.

Walter Tenten
ZEL/NE der KFA Jülich

"Inkrementierungen bei
Siemens - Prozeßrechnern"

Inkrementierungen bei Siemens-Prozessrechnern

W.Tenten, KFA ZEL/NE

Zusammenfassung: Es wird über den Einbau von Inkrementierungseinrichtungen in verschiedene Typen von Siemens-Prozessrechnern berichtet. Dabei wird zuerst der Anwendungsbereich erläutert und anschließend die praktische Ausführung einer solchen Einrichtung am Beispiel der Rechnertypen Siemens 305 und 301 beschrieben.

Um den Sinn und Zweck einer Inkrementierungseinrichtung oder Inkrementierung zu erklären, geht man davon aus, daß die Forderung besteht, Ereignisse in ihrer Häufigkeit zu erfassen. Dabei besteht die Nebenbedingung, daß das Ergebnis der Zählung für die Weiterverarbeitung, aber auch schon während des Zählens zur Überwachung für einen Rechner zugänglich sein soll.

Die Ausführungsform einer solchen Zählapparatur hängt von der Geschwindigkeit, mit der gezählt werden soll, ab. Bei sehr niedrigen Zählraten besteht die Möglichkeit, bei jedem Zählimpuls den Rechner im Programm zu unterbrechen, damit per Programm der Inhalt einer vereinbarten Kernspeicherzelle um eins erhöht (inkrementiert) wird. Je nach Ausführung und Geschwindigkeit von Hardware und Software des vorhandenen Rechners liegt die obere Grenze für dieses Verfahren bei einer Frequenz von einigen kHz. Darüber ist der Rechner derart belastet, daß er keine anderen Aufgaben mehr erfüllen kann; außerdem besteht die Gefahr, daß Impulse verschluckt werden. Diese Methode wird als Interrupt-Methode bezeichnet. Kennzeichnend ist, daß die Programmsteuerung für die Ausführung eines Programms mit mehreren Befehlen benötigt wird. Kann die eigentliche Erhöhung auch bei vielen Rechnertypen mit einem einzigen Befehl veranlaßt werden, dauert es jedoch ziemlich lange, bis es soweit ist. Im Falle eines Siemens-Rechners nehmen BAP-Bearbeitung und Alarmauflösung einen großen Teil der Zeit in Anspruch, zusätzlich ist man von Unterbrechbarkeitsstellen im gerade laufenden Programm abhängig.

Damit sind die Grenzen der Software-Methode aufgezeichnet, und es bleibt wie in vielen Fällen der Ausweg zu einer Hardware-Lösung. Die verschiedenen Hardware-Lösungen werden weiter unten beschrieben.

In vielen Fällen will man nicht nur eine Art von Ereignissen zählen, sondern eine größere Anzahl von Zählkanälen unterscheiden. Ein bekanntes Beispiel aus dem Bereich der Kernphysik ist die Vielkanalanalyse oder Impulshöhenanalyse. Ziel der Impulshöhenanalyse ist es, die Verteilungsfunktion von Impulsen zu ermitteln, die aus einem kernphysikalischen Vorgang stammen. Dabei interessiert in der Regel die Amplitude der Impulse, deren Verteilung ein Amplitudenspektrum ergibt.

Geräte zum Messen solcher Spektren nennt man Vielkanalanalysatoren. (Blockschaltbild Bild 1). Die in analoger Form vorliegenden Spannungsimpulse werden einem Analog-Digital-Konverter zugeführt, der einen Zahlenwert ermittelt, der der Eingangspulshöhe entspricht. Dieser Zahlenwert wird an das Adressregister eines Kernspeichers übergeben. Die spezifizierte Zelle des Kernspeichers wird angesteuert, und ihr Inhalt wird um eins erhöht. Die Erhöhung findet im Datenregister des Kernspeichers statt, und zwar in der Mitte des Kernspeicherzyklusses zwischen dem zerstörenden Lesen und dem Zurückschreiben der Information. Eine solche Arbeitsweise des Kernspeichers wird als Split-Cycle-Betrieb bezeichnet. Eine Darstellung des ermittelten Spektrums kann auf einem x-y-Oszillografenschirm gemacht werden. Dabei werden an die beiden Achsen des Oszillographen die über Digital-Analog-Umsetzer konvertierten Werte von Kanalzahl und Kanalinhalt angelegt. Nach oben ist der Inhalt des jeweiligen Kanals aufgetragen, wobei die einzelnen Kanäle nach rechts nebeneinander gesetzt werden. Typische Vielkanalanalysatoren haben Kanalzahlen von etwa 200 bis zu 8000 (8 bis 13 bit). Die Kapazität eines Kanals liegt bei einer halben Million bis zu mehreren Millionen Ereignissen (19 bis 24 bit).

Der beschriebene Analog-Display ist natürlich nicht die einzige Ausgabemöglichkeit eines Vielkanalanalysators. In vielen Systemen findet man Ausgabe auf Schreibmaschine, Lochstreifen oder Magnetband; vielfach existieren auch Wieder-Eingabemöglichkeiten. Daneben ist es möglich, einfachere arithmetische Operationen wie Addieren, Subtrahieren oder Integrieren durchzuführen.

Wegen dieser vielfältigen Anforderungen ist man in den vergangenen Jahren dazu übergegangen, als Vielkanalanalysatoren arbeitende Kleinrechner einzusetzen, deren Speicher zum Teil für die Aufnahme des Spektrums verwendet wird und zum anderen Teil Programme enthält. Besonders bewährt hat sich bei solchen Einsätzen der Kleinrechner Typ PDP-8/I. Einer der Gründe dafür ist neben dem günstigen Preis die Tatsache, daß dieser Rechner eine Hardware-Inkrement-Einrichtung besitzt. Man hat die Möglichkeit, beim direkten Speicherzugriff von außen her einen Inkrement-Modus zu wählen. Dabei wird der Inhalt der von außen spezifizierten Adresse in einem Zyklus gelesen, um eins erhöht und in den Kernspeicher zurückgeschrieben. Eine vergleichbare Hardware-Lösung besteht im System 300 der Firma Siemens im Einsatz des Prozess-Elementes P3E in der Inkrementausführung. Beim P3E wird über den Schnellkanal von außen ein Lesezyklus eingeleitet, der den Inhalt der spezifizierten Arbeitsspeicherzelle in ein Register in der P3E-Steuerung übernimmt. In diesem Register wird die Inkrementierung vorgenommen, und in einem zweiten Kernspeicherzyklus wird der erhöhte Register-Inhalt in dieselbe Arbeitsspeicherzelle, aus der er entnommen wurde, zurückgeschrieben.

Der Gesamtvorgang dauert drei Kernspeicherzyklen, wovon zweie die Datensteuerung belegen; ein gerade laufendes Programm wird um zwei Kernspeicherzyklen verzögert.

Nun hat jeder Ringkernspeicher die Eigenschaft, daß die gespeicherte Information nicht zerstörungsfrei gelesen werden kann.

Im Normalfall muß die gerade gelesene Information in die Zelle zurückgeschrieben werden, damit sie für weitere Anwendungen verfügbar ist. Aus dieser Arbeitsweise ergibt sich der Begriff des Arbeitsspeicherzyklus. Der Kernspeicherzyklus besteht zu etwa gleichen Teilen aus dem Lesevorgang und dem Schreibvorgang. Zwischen diesen beiden Teilen befindet sich eine geringe zeitliche Lücke. Bei einem Kernspeicher mit Split-Cycle-Betrieb kann diese Lücke beliebig verlängert werden. Die Kernspeicher des Systems 300 erlauben diese Betriebsart nicht. Deshalb werden beim Inkrementieren über P3E zwei bzw. drei Arbeitsspeicherzyklen benötigt.

Die im folgenden beschriebenen Hardware-Änderungen ermöglichen es, die Inkrementierung in einem Zyklus durchzuführen. Dabei wurden leicht unterschiedliche Lösungswege beschritten.

Das Datenregister des Kernspeichers befindet sich örtlich im Steuerwerk und heißt Lese-Schreib-Register (LS). Über dieses Register läuft sämtlicher Verkehr mit dem Arbeitsspeicher ab, und aus diesem Register wird die Information für das Rückschreiben in der zweiten Hälfte des Kernspeicherzyklus entnommen. Während eines Kernspeicherzyklus im Modus Lesen läuft die Information aus dem Arbeitsspeicher in das LS-Register und wieder zurück in den Kernspeicher. (Bild 2). Auf diesem Wege wird die Information bei den vorliegenden Hardware-Modifikationen verändert, um eine Inkrementierung in einem Zyklus durchzuführen.

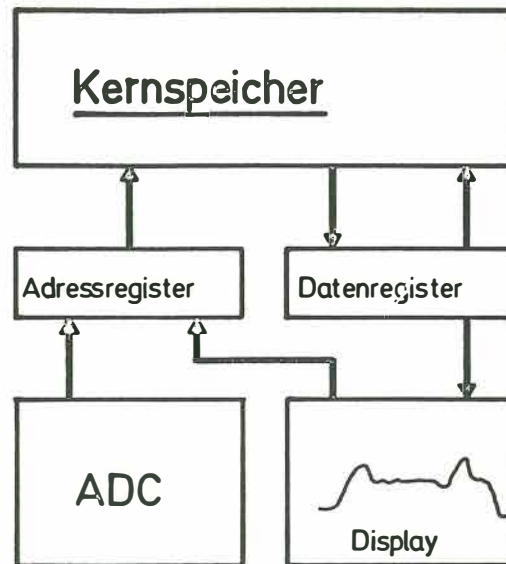
Das erste Beispiel für eine funktionierende Lösung ist ein Rechner Typ Siemens 305. Bei der 305 befindet sich das LS auf vier Steckkarten zu je sechs Flipflops. Die Steckkarten der 305 haben 31-polige Steckleisten und sind deshalb nur zu einem Drittel ihrer Fläche ausgenutzt. Die Flipflops des LS wurden so untereinander verbunden, daß das Register auch sehr schnell zählen kann. Eine zusätzliche Karte mit etwas Logik sorgt dafür, daß im richtigen Augenblick ein Zählimpuls erzeugt wird. Die Erweiterung des LS-Registers wurde innerhalb der einzelnen Karten vorgenommen.

Die ursprünglichen Funktionen des LS blieben uneingeschränkt erhalten; außerdem ist es jederzeit möglich, Originalkarten an die entsprechenden Plätze zu stecken und damit den Lieferzustand der 305 wiederherzustellen. Für den Anwender der Inkrementierung wurde am Schnellkanal eine zusätzliche Leitung geschaffen, die den Modus Inkrementieren auswählt.

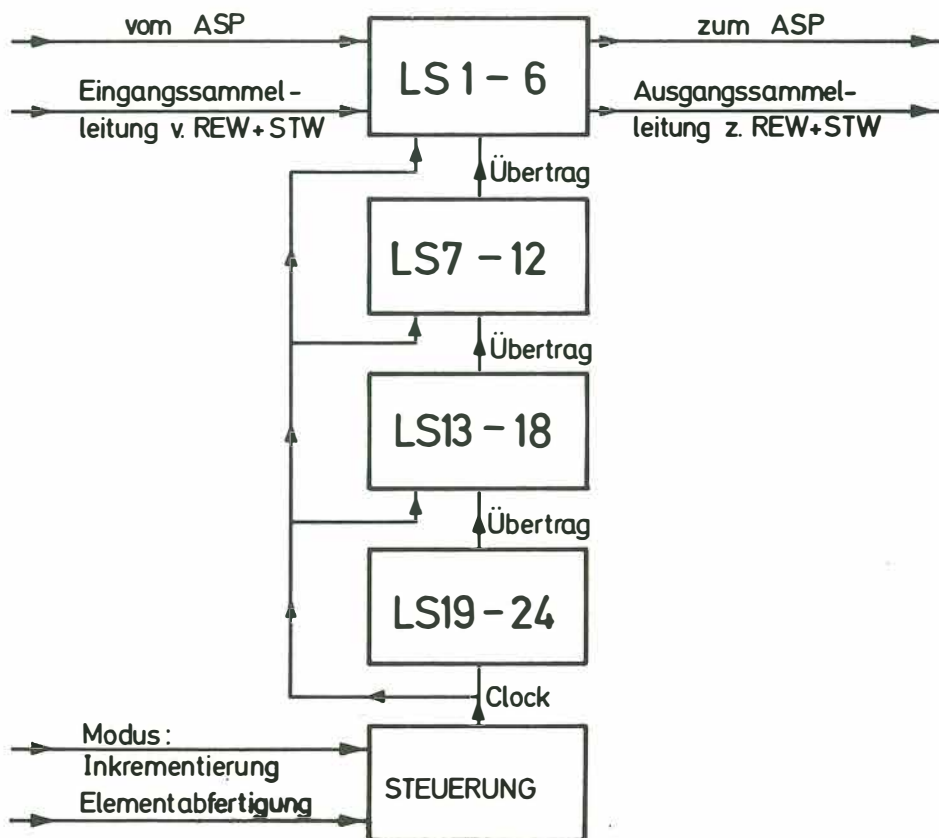
Nachdem der nachträgliche Einbau einer Inkrementierung in eine 305 trotz heftiger Warnungen von kompetenter Seite nach den üblichen Anfangsschwierigkeiten seit Mitte des Jahres 1969 störungsfrei funktioniert, wurde Anfang 1970 eine ähnliche Einrichtung in eine 301 eingebaut. Hier wurde ein etwas abweichender Weg beschritten. In den Weg vom LS zurück zum Arbeitsspeicher wurde ein Netzwerk von Addierern eingesetzt, daß an der niedrigstwertigsten Stelle die Addition einer eins ermöglicht. Auch diese Änderung funktioniert einwandfrei.

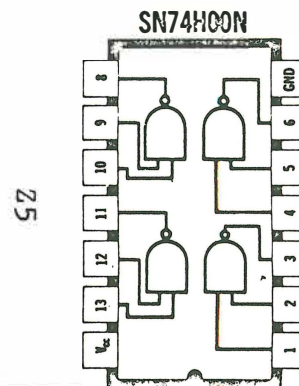
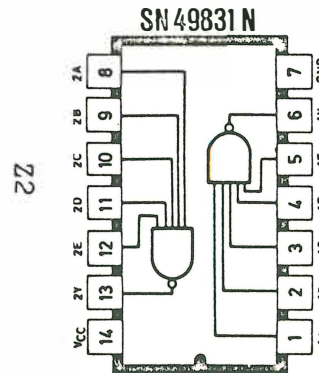
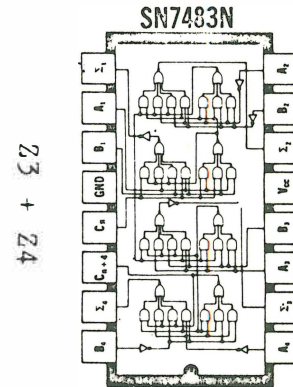
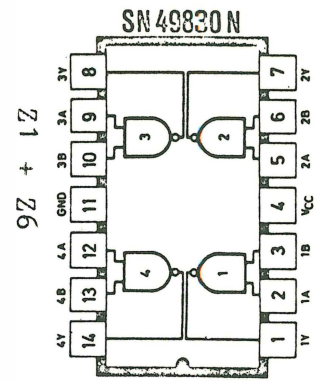
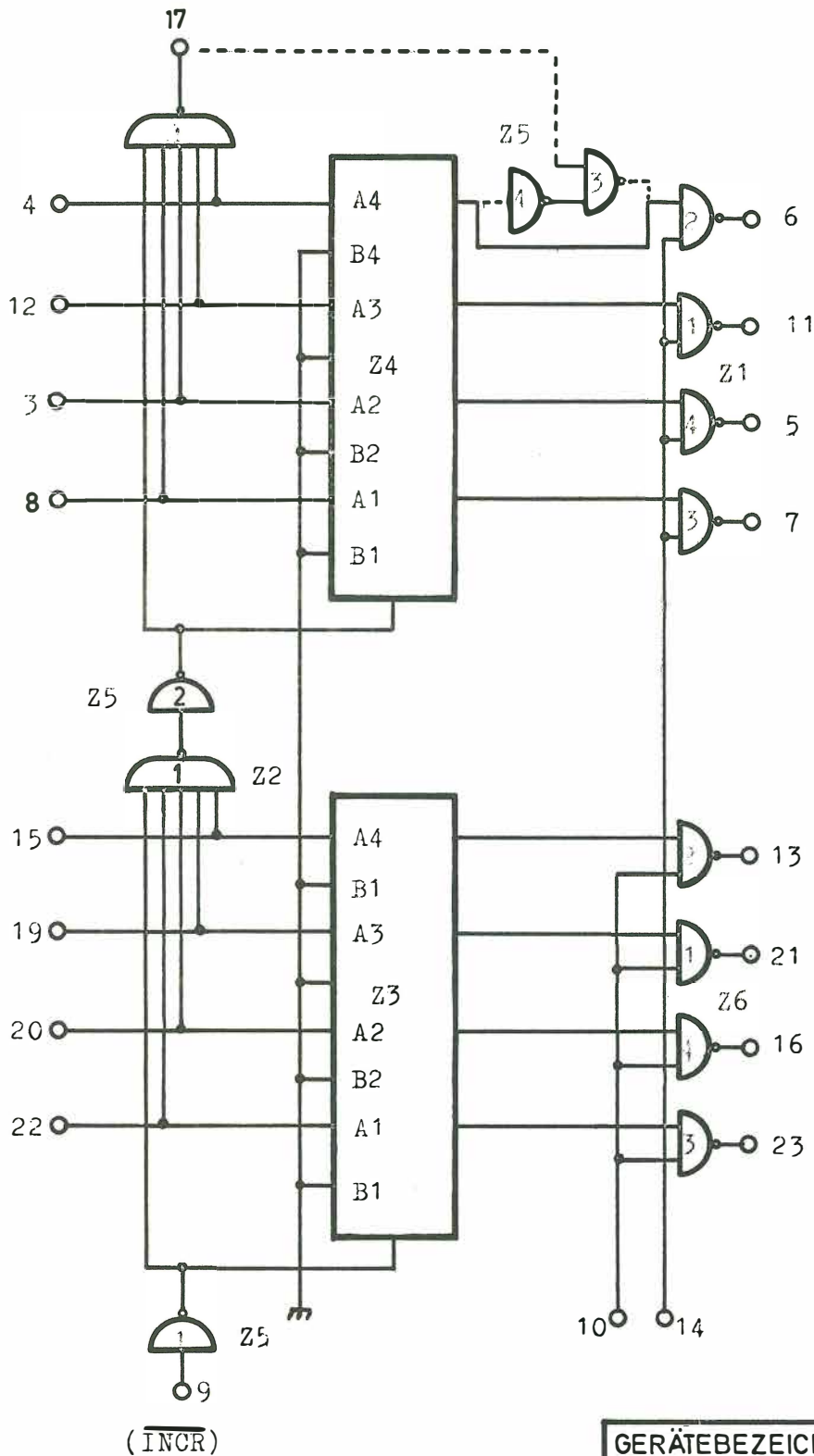
Die möglichen Nachteile solcher Lösungen seien zum Abschluß genannt. Der Inkrement-Modus läuft bei der 305 ab wie ein Lese-Zyklus, deshalb ist der Arbeitsspeicherschutz unwirksam. Eine Basisadresse, ab der inkrementiert werden soll, ist nur über einen Schaltersatz und nicht per Programm einstellbar. Das Starten und Stoppen der Datenaufnahme wurde nur bei der 301 über zwei Koppelbefehle verwirklicht; bei der 305 ist kein vollautomatischer programmgesteuerter Betrieb vorgesehen.

Das Ziel der nachträglichen Modifizierungen an der Hardware eines Rechners war es, den Rechner mit wenig Aufwand für eine bestimmte Aufgabe geeigneter zu machen, als er es ab Fabrik war. Dieses Ziel wurde in beiden Fällen erreicht.



VIELKANAL ANALYSATOR (Prinzip)





GERÄTEBEZEICHNUNG:

Incrementierung Siemens 301

KFA/ZLE-NE

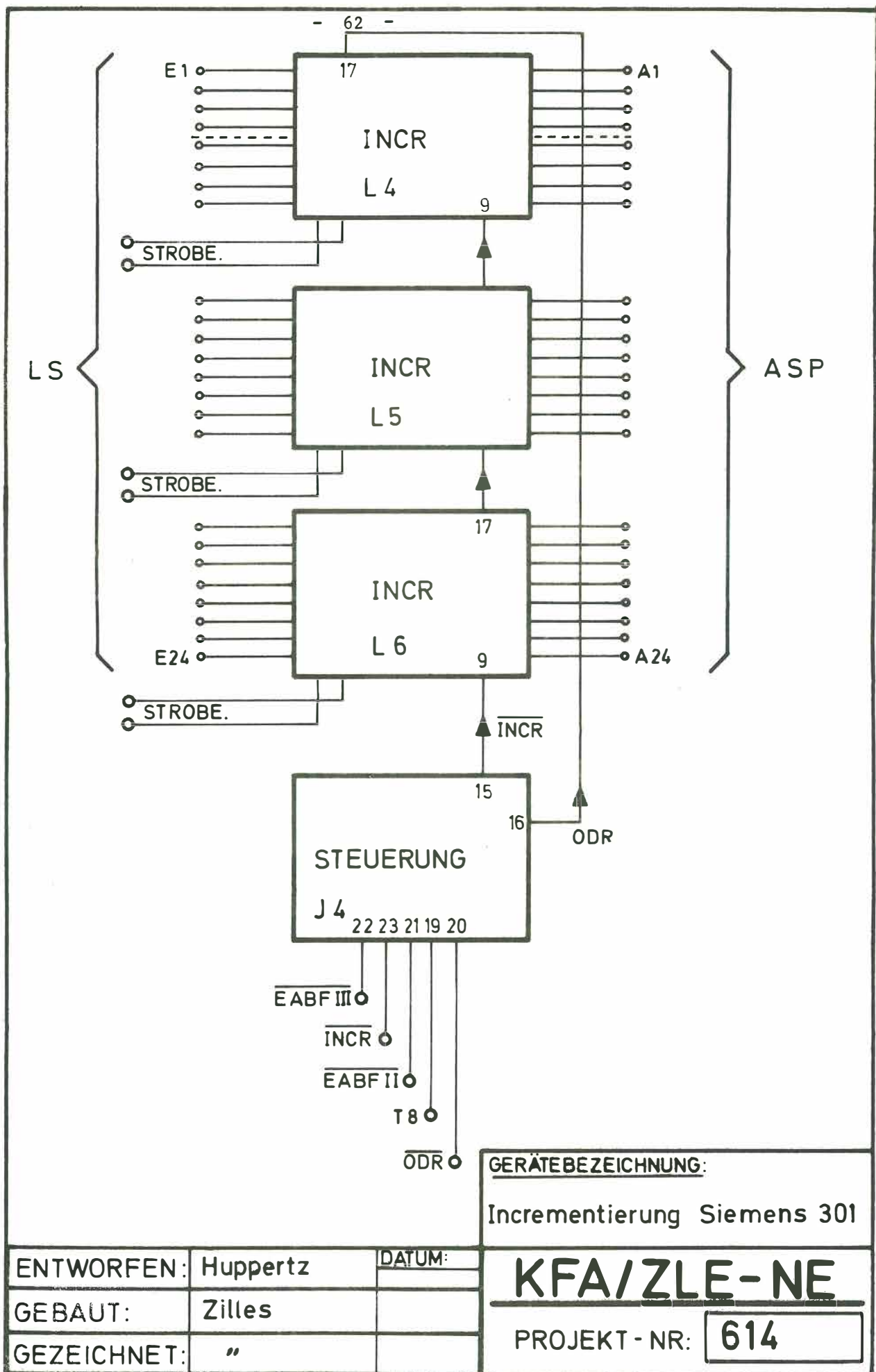
PROJEKT - NR: 614

ENTWORFEN:

DATUM:

GEBAUT:

GEZEICHNET:



GERÄTEBEZEICHNUNG:

Incrementierung Siemens 301

ENTWORFEN: Huppertz

DATUM:

GEBAUT: Zilles

KFA/ZLE-NE

GEZEICHNET: "

PROJEKT - NR: **614**

R. Schmitt
Fa. Messerschmitt-Bölkow-
Böhm, München

"Eigene Programme und Systemprogramme
für einen Rechenzentrumsbetrieb mit
der Siemens 305"

Titel: Eigene Programme und Systemprogramme für einen
Rechenzentrumsbetrieb mit der Siemens 305

Verfasser: Reinhard Schmitt Abteilung: KE11

Datum: 24. 6. 70

Übersicht

Ein Open-Shop-Rechenzentrumsbetrieb ohne speziellen Operator bringt einige organisatorische Schwierigkeiten mit sich. Im Anfang wird besprochen, wie diese durch die Erstellung von besonderen Systemprogrammen und eine entsprechenden Organisation des Rechenzentrums soweit als möglich behoben werden können. Als Beispiele werden die Abrechnung der anfallenden Rechenzeiten und die Wartung der Dateiorganisation auf der Platte aufgeführt. Im weiteren werden einige erstellte Systemprogramme für die Lochkartenbearbeitung, einige spezielle Testhilfen und interessante Unterprogrammgruppen erwähnt.

1. Abrechnung der Rechenzeiten

Eine der Schwierigkeiten bei Rechenzentrumsbetrieb ist die Erfassung der Rechenzeiten der einzelnen Benutzer und die Verrechnung auf die einzelnen Projekte. Nach einigen Versuchen hat man sich für folgende Lösung entschieden:

Der Rechner ist an eine Stempeluhr angeschlossen, so daß er nur über diese ein- bzw. ausgeschaltet werden kann. Die Uhr ist so ausgelegt, daß pro Karte nur 2 Stempelungen, eine für ein und eine für aus, erfolgen können. Jeder Benutzer erhält vom Betreuer der Anlage eine bestimmte Anzahl mit Name und Projektnr. vorgelochter Karten, mit welchen er die Rechenanlage in Betrieb nehmen kann. Die gestempelten Karten werden anschließend in einen Kasten geworfen. Da die Stempelungen von der Uhr gezählt werden, kann man bei Bedarf kontrollieren, ob auch alle Karten eingeworfen wurden. Sollte es sich als notwendig erweisen, kann man auch die Anzahl der ausgegebenen Karten kontrollieren.

Die gestempelten Karten werden täglich vom Betreuer der Anlage gelocht, um dann am Ende des Monats mit dem Programm LOGB (Logbuch) abgerechnet zu werden.

Das Programm (LOGB) läßt es zu, daß auch Kommentarkarten eingestreut werden können. Dies ist dann interessant, wenn Anlagenfehler, Ausfälle, Wartung oder sonstige Besonderheiten der Anlage miterfaßt werden sollen. Man erhält durch diese zusätzlichen Kommentare zu den Benutzerzeiten eine Art Logbuch. So ist es möglich, Unterlagen über Ausfälle usw. über größere Zeiträume zu sammeln.

Das Programm LOGB listet die Karten auf dem Schnelldrucker auf und rechnet die Zeiten nach Benutzern und nach Projektnr.geordnet zusammen.

Durch eine Sonderlochung in den Zeitkarten kann deren Zeit als Hardwarefehler, Systemsoftwarefehler, Rüstzeit oder unter einer anderen Rubrik eingeordnet werden. Nach dieser Methode fallen nun nur noch ca. 10 min pro Tag zum Lochen der Karten und 10 bis 15 min pro Monat für den Durchlauf des Abrechnungsprogrammes an, außerdem kann jederzeit sehr schnell Zwischenbilanz gezogen werden. Dieses Abrechnungsverfahren hat sich sehr gut bewährt, und es ergeben sich keine Meinungsverschiedenheiten mehr mit den Benutzern.

2. Wartung der Dateiorganisation auf der Platte

Da bei Rechenzentrumsbetrieb und besonders bei sehr vielen verschiedenen Benutzern die Auslastung der Anlage stets sehr stoßweise ist, ist es interessant, möglichst alle Programme auf Platte umzustellen. Ein typisches Beispiel ist der Prosa-Übersetzer. Es ist sehr lästig bei Übersetzungen die Karten für 2 Durchläufe in die Karteneingabe einzulegen, außerdem hält das Einlesen von Karten sehr auf. Der Übersetzer wurde dahingehend umgearbeitet, daß er über die Lochkarteneingabe einzulesende Karten gleichzeitig in eine frei zu wählende Datei auf der Platte ablegt oder daß er die Prosa-Karten aus einer Datei auf der Platte einliest. So ist es möglich, den 1. Durchlauf von Karten einzulesen und den 2. Durchlauf von der Platte her zu verarbeiten. Bei sehr langen Programmen ist es noch besser, sich eine Prosa-Datei anzulegen, welche mit dem Siemens-Programm KGAM leicht korrigiert werden kann und dann sowohl für den 1. als auch 2. Durchlauf der Übersetzung die Prosa-Karten von der Platte her einzulesen. Dadurch ist es möglich, die einzelnen Anlagenbenutzungszeiten um die langsamen Externverkehrszeiten zu verringern und damit die Stoßzeiten etwas zu entlasten. Um nicht durch häufiges Plattenwechseln wieder zusätzliche Ausfallzeiten zu erhalten, ist es günstig, wenn alle Benutzer die gleichen Plattenstapel verwenden. Lediglich Teams mit größeren Projekten sollten ihre eigenen Plattenstapel benutzen.

Durch die Benutzung eines Gemeinschaftsplattenstapels ist auch der Service der Systemprogramme leichter durchzuführen, d.h., die Systemprogramme auf der Gemeinschaftsplatte sind stets auf dem neuesten Stand.

Als Problem ergibt sich jetzt jedoch die Verteilung der Dateinamen und das Löschen alter unbenutzter Dateien, um den Speicherplatz wieder freizugeben. Dieses Problem wird dadurch immer größer, daß durch die Umstellung von immer mehr Systemprogrammen auf die Platte immer mehr Hilfsdateien angelegt werden, welche nicht automatisch gelöscht werden.

Um die Platte von Zeit zu Zeit von alten Dateien säubern zu können, wurde das Programm ESLO (Externspeicher listen und löschen) geschrieben. Es arbeitet ähnlich wie das Siemens-Programm ESLI (Externspeicher listen), löscht aber zusätzlich, bevor es die Dateien protokolliert, alle Dateien, welche nicht in einer Kartei verzeichnet sind. In der Kartei müssen außer dem Dateinamen noch Plattennummer, Benutzername, Abteilung, Telefonnummer und eventuelle zusätzliche Bemerkungen wie Zeitpunkt, ab wann die Datei gelöscht werden kann oder Aufgabe und Zweck der Datei abgelocht werden. Diese zusätzliche Information wird mit auf dem Schnelldrucker aufgelistet, so daß man am Protokoll erkennen kann, welcher Benutzer welche Dateinamen verwendet hat. Dateien, welche nicht in der Kartei verzeichnet sind, werden mit ihren Namen auf dem Bedienungsblattschreiber ausgegeben und gelöscht. Die Entscheidung, ob die nicht aufgeführten Dateien automatisch gelöscht werden sollen oder erst nach einer positiv quitierten Anfrage, kann mit einem Codewort gewählt werden.

Ein Vorteil des Programmes ESLO ist es gleichzeitig, daß es auch Dateien mit verstümmelten Namen löschen kann, welche oft bei Tests von Programmen entstehen können.

Diese Eigenschaft hat sich als sehr zweckmäßig erwiesen, da verstümmelte Namen meist nicht abdruckbare Zeichen enthalten und damit ihre Namen nicht über Bedienungsblattschreiber oder ein anderes externes Gerät eingegeben werden können; das Programm ESLO holt sich den Namen jedoch direkt aus der Namensliste.

Um gezielt nur einzelne Dateien zu löschen, wurde das Programm DALO (Dateien löschen) geschrieben. Sie können damit einzelne Dateien löschen oder auch bis zu 10 Dateinamen gleichzeitig eingeben. Da sowohl über ein MA BEWA als über ein MA BSEI Dateinamen eingegeben werden können, ist es möglich, auch Namen mit Blocks an beliebigen Stellen zu löschen. Außer Dateien löschen, kann das Programm DALO auch Dateien beliebiger Länge auf der Platte anlegen. Das ist dann wichtig, wenn man Dateien an eine bestimmte günstige Stelle legen möchte. Man kann den unerwünschten Bereich vorher durch Dummy-Dateien füllen oder einen gewünschten Bereich durch eine Dummy-Datei belegen und damit freihalten. Ein anderer Grund kann sein, daß ein Programm nur mit wiedereröffneten Dateien arbeitet.

Bei sehr starker Benutzung der Platte und gleichzeitigem Testbetrieb kommt es leicht vor, daß die Dateiorganisation zerstört wird; dies kann durch folgende Tatsachen verursacht werden:

- 1 Benutzung der Taste "Unbedingte Anforderung". In dem Siemens-Laborbericht ZL/LAB/184/7026/HZG/ZK wird behandelt, welche Fehler durch die Benutzung der Taste entstehen können. Gerade bei Testbetrieb wird die UAP-Taste gerne verwendet.

- 2 Benutzung der Taste "Anlage Rücksetzen" in ungünstigen Momenten, z.B. wenn gerade ein Plattentransfer stattfindet.
- 3 Ein unausgetestetes Anwenderprogramm schreibt in die im Kernspeicher stehende Dateiliste bzw. zerstört das Bearbeitungsprogramm für Dateiorganisation.
- 4 Ein unausgetestetes Anwenderprogramm überschreibt durch eine eigene falsche Errechnung des Schreibindex die Dateiliste auf der Platte.
- 5 Es wird die Platte gewechselt, ohne das ORG neu einzulesen. Es kann dann die im Kernspeicher befindliche Dateiliste des alten ORG's einen Teil der Dateiliste des neuen ORG's überschreiben und damit zerstören.

Mancher dadurch in der Dateiorganisation entstandene Fehler ist nicht sofort erkennbar, bewirkt aber eine doppelte Belegung des Externspeicherplatzes bzw. zerstört durch falsche Daten die Organisation der Dateien erst richtig bei weiterer Benutzung.

Um solche Fehler möglichst früh zu erkennen, wurde das Programm PSLI (Plattenspeicherlisten prüfen) geschrieben. Es prüft die Zylinder- und die Namensliste auf formale Fehler, anschließend wird die Zuordnung von Namensliste und Zylinderliste untersucht und es wird auf eventuelle doppelt belegte Sektoren geprüft. Weiter wird ausgegeben: die Anzahl der freien Sektoren, der belegten Sektoren und der unbenutzten, aber nicht verwendbaren Sektoren. Aus der Anzahl der nicht verwendbaren Sektoren kann geschlossen werden, ob es notwendig ist, durch einen Aufruf des Komprimierungs-Makros (MA PSDK) die unbrauchbar gewordenen Sektoren wieder freizulegen.

Zeigt das Programm PSLI Fehler auf, so ist es notwendig, sich den Inhalt der Namensliste und der Zylinderliste ausgeben zu lassen. Anhand der beiden Listen kann dann sehr schnell entschieden werden, welche Korrekturen an der Namensliste bzw. der Zylinderliste unternommen werden müssen, um die Dateiorganisation zu retten. Die Korrekturen können im allgemeinen leicht mit dem Siemens-Programm EXBE oder UWØØ (Uewam) durchgeführt werden. Ein spezielles Korrekturprogramm soll noch geschrieben werden.

Die Ausgabe der Namensliste und deren Prüfung auf formale Fehler kann durch das Programm NALI (Namensliste ausgeben) erfolgen. Die Ausgabe der Zylinderliste und deren Prüfung auf formale Fehler kann durch das Programm ZYLI (Zylinderliste ausgeben) erfolgen. Es hat sich als sehr günstig gezeigt, das Programm PSLI in regelmäßigen Abständen (z.B. 1 Woche) laufen zu lassen.

3. Systemprogramme zur Lochkarten-Bearbeitung

Damit alle Programmbeschreibungen auf Lochkarten abge-
locht und dann beliebig oft aufgelistet werden können
und auch sonst viele Arbeiten über Lochkarten erfolgen,
wurde auch für die Lochkarten-Bearbeitung eine Reihe
Programme erstellt.

Das Programm COPY dient zum Doppeln von Lochkarten. Soll
die Lochkarteneingabe beim Doppeln von Karten nur sehr
kurzzeitig belegt werden, um das Doppeln von Karten als
Hintergrundsarbeit durchzuführen, so ist das Programm
UMSE (Umsetzen) günstig zu verwenden. Es liest Lochkar-
ten in eine Datei beliebigen Namens ein bzw. gibt Dateien
mit alphanumerischem Inhalt auf Lochkarten aus. In einem
Arbeitsgang können die zu doppelnden Karten auf der Plat-
te in eine Datei abgesetzt werden und dann kann diese
Datei im Hintergrund zu anderen Programmen beliebig oft
ausgegeben werden. Das Programm UMSE soll bei uns noch er-
weitert werden, daß es wie das Siemens-Programm MCUM
(Maschinencode umsetzen) als Programm ANUM (Alphanumeri-
schen Inhalt umsetzen) alphanumerischen Inhalt in Form
eines Kartenäquivalentes von einem Externspeicher-Medium
auf ein anderes umsetzen kann. An einem entsprechenden
Programm für binären Inhalt BIUM (Binären Inhalt umset-
zen) wird gearbeitet.

Mit dem Programm LKPR (Lochkartenprüfen) können gedoppel-
te Lochkarten mit dem Originalstapel verglichen werden.
Das Programm LKDR listet Lochkarten auf dem Schnelldruck-
er auf. Schnelldrucker und Lochkarteneingabe arbeiten
mit einem Umlaufpuffer, so daß das langsamste Gerät die
Geschwindigkeit bestimmt, während das andere parallel da-
zu arbeitet. Über Steuerkarten kann man erreichen, daß

das Programm LKDR eine gewünschte Anzahl von Zeilen- oder Blattvorschüben durchführt. Eine Überschrift, welche auf jeder Seite erscheint, kann per Bedienungsblattschreiber eingegeben werden. Zeilen können gegenüber anderen Zeilen verschoben werden. Zeilen, Karten und Spalten können auf Anweisung durchnumeriert werden. Durch diese vielen Varianten ergibt sich für die Benutzung des Programmes LKDR ein breites Spektrum z.B. Lochkarten auflisten, Erstellen von Programmbeschreibungen oder sonstigen Anweisungen auf dem Schnelldrucker ausgeben.

Mit dem Programm LKNR (Lochkarten numerieren) können Lochkarten in beliebigen Spalten, ab beliebigen Zahlen und in beliebigen Sprüngen durchnumeriert werden.

Mit dem in Arbeit befindlichen Programm LKSO (Lochkarten sortieren) können Karten nach gewünschten Kriterien aussortiert werden.

4. Testhilfen

Das Programm ZEME (Zeitabstände messen) ermöglicht es, echte Laufzeiten von Programmen oder Programmteilen zu messen. Dieses Programm erfordert jedoch eine bestimmte Prozeßperipherie. Das Programm LARA (linker Akku, rechter Akku) ermöglicht das Abspeichern der Akkuinhalte an gewünschten BZ-Ständen in einen vorgegebenen Kernspeicherbereich, um sie später mit einem Arbeitsspeicherabzug auszugeben. Diese Methode ist wichtig für Programmuntersuchungen an Echtzeitprogrammen, welche nur möglichst wenig aufgehalten werden dürfen. Sehr nützlich hat sich das Programm SUCH erwiesen, welches einen gewünschten Kernspeicherbereich nach einem gewünschten Zelleninhalt absucht. Es gibt die Adresse der Zellen mit dem gesuchten Inhalt auf dem Bedienungsblattschreiber aus. Dieses Programm ist besonders günstig beim Testen von Code-Prozeduren für Algol- und Fortranprogramme. Mit dem Programm SUCH kann die MC-Fassung des Algol- bzw. Fortran-Programmes auf die Lage der Code-Procedur abgesucht werden. Weiter kann man Befehle auf eine bestimmte Operandenadresse hin absuchen und zwar sowohl auf die substituierte als auch auf die unsubstituierte Adresse.

Als günstig hat sich das Programm SYLE (Lesen symbolidentifizierter Daten verschiedenen Typs von Lochkarten in den Arbeitsspeicher) erwiesen. Es ermöglicht es ähnlich wie bei Fortran Daten anhand einer "Namelist" in den Kernspeicher von Karten aus einzulesen. Dadurch ist es möglich, Testdaten für ein zu testendes Programm an beliebiger Stelle im Arbeitsspeicher aufzubauen. Mit den ersten Karten wird das einzugebende Datenfeld definiert und es werden die entsprechenden Namen festgelegt. Im Anschluß an diese Karten folgen die Daten selbst.

5. Unterprogramme

Einige besonders interessante Unterprogramme sollen noch vorgestellt werden.

Das bereits erwähnte Hauptprogramm SYLE (Lesen symbolidentifizierter Daten verschiedenen Typs von Lochkarten in den Arbeitsspeicher) besteht im wesentlichen aus dem Unterprogramm SYLK. Dieses Unterprogramm soll das Schreiben von Testprogrammen vereinfachen und ermöglicht es, Daten per "Namelist" einzulesen.

Weiter interessant sind die drei Programme UPUF, UMPU, UPFE, welche einen Umlaufpuffer organisieren. Das Programm UPUF organisiert einen zeilenorientierten Umlaufpuffer mit 2 Zeigern, welcher wortweise beschrieben werden kann. Das Programm UMPU organisiert einen wortweise organisierten Umlaufpuffer mit 2 Zeigern und das Programm UPFE organisiert einen zeilenorientierten Umlaufpuffer mit beliebig vielen Zeigern, welcher wortweise beschrieben werden kann. Eine weitere Gruppe von UP's, die Programme AUSG, ZEI, Z2F, BIMU, DAN, TEX1, TEX2, dienen dazu, anhand von Formatlisten auf dem Schnelldrucker Information in beliebigen Formaten auszugeben. Das Programm AUSG ruft die anderen Programme anhand der Formatlisten auf, bis eine Zeile Information druckreif aufgebaut ist und gibt diese aus. Das UP AUSG eignet sich dazu, eine in einem Umlaufpuffer gespeicherte Rohinformation, welche aus Zeitgründen nicht direkt ausgegeben werden konnte, zu einem späteren Zeitpunkt abzuarbeiten und auszugeben.

J. Radünz
Deutsche Forschungs- und
Versuchsanstalt für Luft-
und Raumfahrt e.V.,
Oberpfaffenhofen

"Betriebssystem für den deutschen
Forschungssatelliten A Z U R"

26.6.1970

I. Einleitung

Es gibt wohl kaum ein Gebiet der modernen Technik, das so sehr von der elektronischen Datenverarbeitung abhängig ist und diese wiederum so nachhaltig beeinflußt hat, wie die Weltraumfahrt. Ganz besonders gilt dies für die Betriebsphase der Raumflugprojekte. Stellt man sich nur vor, daß äußerst Komplexe wissenschaftliche und technische Systeme in Abständen zwischen 200 km und 300 Mio km von der Erde entfernt mit Geschwindigkeit von 28 bis 100 km/sec um die Erde oder die Sonne fliegen und von der Erde aus betrieben werden müssen, so kann man die Schwierigkeiten ahnen, die sich aus diesem Betrieb ergeben. Anhand des Beispiels des 1. deutschen Forschungssatelliten AZUR, der am 8. November 1969 von Kalifornien aus mit einer 4-stufigen Scout-Rakete gestartet wurde und vom Satelliten-Kontrollzentrum in Oberpfaffenhofen b.München betrieben wird, soll auf die Datenverarbeitungsaufgaben sowie seine Funktionsüberwachung und -Steuerung näher eingegangen werden.

II. Aufgaben eines Kontrollzentrums

Die Aufgaben eines Kontrollzentrums teilen sich in zwei Hauptgebiete, zum einen die Kontrolle des gesamten Bodenbetriebssystems und zum anderen die Funktionsüberwachung des Satelliten.

Alle Einrichtungen am Boden, die für den Empfang von Satellitendaten erforderlich sind, also die Telemetriebodenstationen, gehören zum Bodenbetriebssystem. Das Bodenstationsnetz für AZUR setzt sich zusammen aus 4 deutschen Telemetriebodenstationen, 8 Stationen der NASA, 3 Stationen der ESRO, 3 Stationen der CNES (französische Welt-
raumorganisation) und einer brasilianischen Station. So findet z.B. ein stetiger Versand von Analogbändern zu den Empfangsstationen und umgekehrt von den Stationen zum Kontrollzentrum statt. Außerdem müssen die Stationen mit Kontaktübersichtstabellen, also zu welchem Zeitpunkt der Satellit empfangen werden kann und Antennenlochstreifen zur automatischen Antennensteuerung versorgt werden. Die richtige Arbeitsweise aller Geräte gehört ebenfalls zu den Kontrollfunktionen.

Um eine einwandfreie Funktion des Satelliten gewährleisten zu können, ist es erforderlich, alle Daten die an der deutschen zentralen Bodenstation in Weilheim empfangen werden, direkt in Echtzeit im Kontrollzentrum zu verarbeiten und zu analysieren, um anhand dieser Daten den Status des Satelliten zu überwachen und so mögliche Gefahrenzustände durch sofort veranlaßte entsprechende Kommandogabe abzuwenden.

III. Telemetriesystem und Datenempfang

Die wissenschaftlichen Meßwerte der Experimente und die Daten des Funktionsüberwachungssystems werden -falls sie nicht schon in digitaler Form als Zahlenwerte vorliegen- in einer ganz bestimmten Reihenfolge abgetastet und digitalisiert. Abtasthäufigkeit und Quantisierungs-genauigkeit (Zahl der binären Stellen) hängen von den Anforderungen der Wissenschaftler und Techniker ab. Anschließend werden diese Meßwerte an Bord aufbereitet und in Serie Bit für Bit über Funk im Takt eines sehr genauen Oszillators zum Boden übertragen. Jeder Meßwertzyklus oder "Rahmen" wird mit einem bekannten Bitmuster, dem Synchronisationswort, begonnen. Wir sprechen hier von sogenannten RTT-Daten (ReaT-Time-Telemetrie-Daten), denn die Daten werden direkt gesendet. Sehr häufig werden die anfallenden Daten auch an Bord auf ein Magnetband gespielt und dieses bei Kontakt einer Bodenstation auf ein Kommando hin abgespielt. Hier sprechen wir dann von TT-Daten (Tape-Telemetrie-Daten). Im Falle AZUR bekommen wir alle 500 ms einen RTT-Rahmen.

Die am Boden empfangenen Telemetriesignale werden sofort, zusammen mit Zeitinformation (Zeitkode) und Funktionsüberwachungsgrößen der Bodenstation, auf ein Analogband aufgenommen, und später in dem Prozeß der Datenauswertung wissenschaftlich verarbeitet.

Über zwei verschiedene Datenstrecken zwischen der deutschen zentralen Bodenstation und dem Kontrollzentrum können die empfangenen Daten direkt in Echtzeit von der Bodenstation zum Kontrollzentrum übertragen werden. Normalerweise laufen die Daten über eine Kabelstrecke, sollte diese unterbrochen sein, wird automatisch auf eine Richtfunkstelle umgeschaltet. Im Kontrollzentrum werden die Daten sofort aufbereitet, analysiert und ausgegeben. Noch während des gleichen Kontaktes mit dem Satelliten, der von einigen Minuten bis zu einer halben Stunde dauern kann, können die verantwortlichen Satellitenoperateure bei aufgetretenem Fehlverhalten die entsprechenden Abhilfemaßnahmen ergreifen.

IV. Anlagenfiguration

Welche Anforderungen bestanden nun für die Auslegung des Echtzeitdatenverarbeitungssystems im GCC?

Die Anlage mußte in Echtzeit verarbeiten können:

- Pulscodemodulierte Satellitendaten (wie sie bei AZUR anfallen)
- Analoge Daten
- Analoge Daten, auf Unterträgern frequenzmoduliert
- Zeitkodes
- Digitale Daten, gespeichert auf herkömmliche Datenträgern.

Dabei sollten die PCM-Daten über Datenleitungen von der Bodenstation oder von Magnetband oder von einem anlageninternen Simulator in das System eingespeist werden können mit einer max. Bitrate von ca. 1 Mbps.

Analoge Daten sollten prinzipiell von Magnetband abgespielt werden, während Zeitkodes sowohl von Band oder einer internen Digitaluhr als auch über eine Datenleitung von der Bodenstation zur Verarbeitung anliegen können.

Die digitale Dateneingabe sollte in herkömmlicher Weise über Magnetband, Lochkarte oder Lochstreifen erfolgen, jedoch auch über eine speziell zu entwickelnde Funktionstastatur und Datensichtgeräte möglich sein.

Sämtliche Datenausgabeelemente mußten darauf abgestimmt werden, daß es sich beim Einsatz der Anlage um die Überwachung teilweise schnell veränderlicher Größen handelt, die größtenteils in Echtzeit verarbeitet und angezeigt werden müssen.

So kamen als Ausgabegeräte nur in Frage

- Datensichtgeräte
- Analogschreiber
- Schnelldrucker und
- Digitalmagnetbandgeräte.

Bei der Planung des Systemkonzeptes der Anlage mußte davon ausgegangen werden, daß das System in der Lage sein müsse, nach vorgegebenen Programmen kontinuierlich und teils mit hoher Geschwindigkeit anfallende Telemetriesignale in Echtzeit zu verarbeiten, wobei im Gegensatz zur herkömmlichen Datenverarbeitung die Verarbeitungsgeschwindigkeit mit der von außen vorgegebenen Datengeschwindigkeit Schritt halten muß. Dementsprechend wurde auf höchstmögliche Betriebs- und Bediensicherheit Wert gelegt. Zudem bestand die Forderung, daß in kürzester Zeit die Versorgung der einzelnen externen Geräte des Systems mit Betriebsparametern sowie die Schaltung anderer Signalwege möglich sein muß, um kurzfristig verschiedene Satellitenprojekte bedienen zu können.

Und schließlich mußte eine automatische Steuerung und Überwachung des Gesamtsystems eingeplant werden, um Fehlerquellen, die zu Ausfall oder zur Verfälschung der Resultate führen könnten, möglichst frühzeitig entdecken und beheben zu können.

Als Konsequenz aus all diesen Forderungen ergab sich die Notwendigkeit eines vollautomatischen, programmgesteuerten Datenverarbeitungssystems.

Der Kern des Systems besteht aus einer Prozeßrechenanlage mit 16K Kernspeicher, 24 Bit Wortlänge und 1,5 usec Zykluszeit. Umfangreiche periphere Standardgeräte sind an die Rechenmaschine angeschlossen: 3 Digitalbandgeräte, 1 Platte, Lochstreifen-Ein/Ausgabe, Lochkarteneingabe, Schnelldrucker und Fernschreiber. Die zusätzliche Sonderperipherie, die in speziellen Arbeitskonsolen untergebracht ist, umfaßt 3 voneinander unabhängige Sichtgeräte mit insgesamt 4 Parallelgeräten für alphanumerische und graphische Darstellung, 16 Digital-Analogwandler mit angeschlossene Schreibern sowie Tastenfelder für die manuelle Eingabe von 1-Bit-Informationen zur Programm- und Ausgabesteuerung während des Satellitenbetriebes.

Dem Rechner vorgeschaltet ist eine Signalverarbeitungsanlage. Der FM/Analogteil mit insgesamt 8 Kanälen ermöglicht die Verarbeitung von frequenzmodulierten analogen Daten und unmodulierten analogen Daten.

Der PCM-Teil umfaßt Bitsynchronisierer und Rahmensynchronisierer für die Aufbereitung von pulscodemodulierten Daten sowie einen PCM-Simulator.

Über einen Zeitkodegenerator - Übersetzer, eine Art Digitaluhr, die entweder intern oder fremdgesteuert wird, kann eine Zeitinformation in den Rechner eingespeist werden.

Die zu verarbeitenden Daten werden entweder von Analogbändern an den 2 vorhandenen 7-Spur-Maschinen abgespielt oder in Echtzeit dem System zugeführt. Die Signalverteilung erfolgt mittels einer Schaltkonsole, wo auch die wichtigsten Kontrollgeräte zur Signalüberwachung untergebracht sind. Die Schaltung der Signalwege und die Parametereinstellung kann programmgesteuert oder manuell

erfolgen. Zur automatischen Überwachung der Anlagen sind von den einzelnen Geräten Status- und Betriebsbereitschaftsmeldungen zum Rechner geführt.

V. Software-Programmsystem

Das gesamte Programmsystem für AZUR wurde für diese Anlage im Kontrollzentrum entwickelt und programmiert. Welche Anforderungen wurden nun an dieses Programmsystem gestellt.

- Aufbereitung aller Satellitendaten
- Dekommutieren, Kalibrieren und Normieren der aufbereiteten Daten
- Ausgabe der Resultate auf Schnelldrucker, Digitalbandgeräte und Sichtgeräte, wobei während des Prozesses die Ausgabeformate sowohl auf den Sichtgeräten als auch auf dem Schnelldrucker geändert werden können.
- Kommandoeingabe und -verifizierung
- automatische Überwachung der gesamten Anlage
- einfache Bedienung durch den Operateur, falls erforderlich;
die Bedienungen müssen kontrolliert werden und eine falsche Bedienung darf nicht zur Programmunterbrechung führen
- die Datenverarbeitung muß völlig getrennt von der Datenaufbereitung ablaufen.

Der gesamte Programmablauf mußte sich nach der Geschwindigkeit der ankommenden Satellitendaten richten, und es mußte ein Höchstmaß an Betriebssicherheit gewährleistet sein. Zur Bewältigung dieser Aufgaben standen im Kernspeicher aber nur 12 K zur Verfügung, die bei diesem Programmsystem bei weitem nicht ausreichen konnten.

Das gesamte Programmsystem benötigt ca. 48 K und mußte deshalb aufgeteilt werden in Abschnitte, die ständig zur Verfügung stehen müssen und in solche die nur auf Anforderung benötigt werden. Aus Gründen der Speicherplatz- und Zeitoptimierung wurde als Programmiersprache PROSA gewählt. Auf die wesentlichen Programme soll hier nun etwas näher eingegangen werden.

VI. Programm "BEDI"

Bei der Datenaufbereitung für AZUR unterscheiden wir vier Verfahren

- die Echtzeit-Datenverarbeitung, "Quick-Look A" genannt:

hier werden die RTT-Daten noch während des Kontaktes mit dem Satelliten aufbereitet und analysiert

- der "Quick-Look B":

er unterscheidet sich vom "Quick-Look A" dadurch, daß er bereits aufbereitete Daten vom Digitalband verarbeitet

- der "Quick-Look C":

er verwendet keine RTT-Daten sondern TT-Daten (Tape-Telemetrie-Daten)

- die Simulation:

sie dient vor dem Start des Satelliten hauptsächlich zur Schulung der Satellitenoperateure. Bei diesem Verfahren werden die Daten über Lochkarten eingegeben. Man kann damit auf einfache Weise definierte Satellitenzustände herstellen, die der Satellitenoperator dann richtig analysieren muß.

Alle diese verschiedenen Verfahren unterscheiden sich hinsichtlich der Datenaufbereitung, aber nur sehr unwesentlich hinsichtlich der Datenverarbeitung. So lag es also nahe ein Bedienungsprogramm zu erstellen, das nach Eingabe des Aufbereitungsmodus das eigentliche Programm zusammenstellt.

Diese Aufgabe übernimmt das sogenannte "BEDI".

Es läuft unter der Programmpriorität 2 und besteht aus einem residenten Teil, der immer im Kernspeicher steht und Parameter für die weiteren Programme enthält, und einer Anlaufroutine die hinterher aus Platzersparnis überschrieben wird.

Nach dem Start des Programms wird zur Anlaufroutine verzweigt und eine Lochkarteneingabe verlangt.

Ist die LKE unklar, erscheint eine entsprechende Meldung auf dem Bedienungsblattschreiber (BBS).

Diese Meldung kann nun vom Operateur mit plus oder minus quittiert werden. Wird ein Minuszeichen eingegeben, deklariert das Programm die LKE als unklar, und sie wird während des gesamten Programmlaufs nicht mehr angesprochen. Die Parametereingaben müssen dann über den BBS vom Operateur eingegeben werden. Danach werden die externen Geräte wie Schnelldrucker, Digitalmagnetbänder, Sichtgeräte und Lochstreifenausgabe auf die gleiche Art und Weise getestet. Anschließend wird das Programmsystem zusammengestellt, die zulässigen Anwenderprogramme werden bestimmt und die Dateien eröffnet. Nun wird das Programm zum Testen der Aufbereitungsanlage bereitgestellt und gestartet, auf dieses Programm wird später noch näher eingegangen. Ist die Aufbereitungsanlage auch versorgt und getestet, werden die erforderlichen Programme gestartet und das Programm "BEDI" läuft auf einen BEWA-LISTE. Es wird erst wieder am Ende der Aufbereitung benötigt.

VII. Programm "ALBE"

Das Programm "ALBE" dient ausschließlich der Bearbeitung von Alarmen und läuft unter der Priorität 3.

Folgende Alarme können eintreffen:

1. Der Sekundenalarm von der Zeitanlage:

er wird benötigt einmal, um auf einem Display in Minuten und Sekunden die Dauer bis zum Kontaktfang anzugeben, und zum anderen anschließend die Zeit bis zum Kontaktende, damit der Satellitenoperator die noch verbleibende Kontaktzeit kennt, um ein erforderliches Kommando noch rechtzeitig absetzen zu können.

2. Der Alarm vom Rahmensynchronisierer (RS):

jedesmal wenn der RS ein Synchronisationswort erkannt hat, wenn also ein neuer Rahmen beginnt, wird der Zeitübernahmepuffer gesperrt und ein Alarm über das Pl abgesetzt. Mittels dieses Alarms wird das Aufbereitungsprogramm immer wieder neu gestartet.

3. Alarme von den 3 Digitaleingabetastaturen mit je 24 Bit:

die erste Reihe ist für die Kommandoeingabe vorgesehen. Beim AZUR unterscheiden wir 70 verschiedene Kommandos. Den einzelnen Tasten ist eine bestimmte Wertigkeit (10,20,30-70 und 1,2,3-9) zugeordnet. Wird nur die Kommando-Nr. eingegeben so heißt das, das Kommando soll vorbereitet werden, wird auch die "GO"-Taste dazu gedrückt, soll das Kommando gesendet werden.

Eine weitere Taste ist für die Kommandoverifizierung vorgesehen, denn jedes Kommando bewirkt im Satelliten eine Änderung der Schaltzustände. War z.B. der Sender I ausgeschaltet und wurde ein entsprechendes Kommando gegeben um den Sender I wieder einzuschalten, muß der neue Schaltzustand auf Sender I "EIN" stehen, wenn nicht, wurde das Kommando nicht erkannt und muß erneut gesendet werden. Alle gesendeten Kommandos werden auf einem Fernschreiber (FS) mit der Vorbereitungs- und mit der Sendezeit protokolliert.

Die zweite Tastatur dient zur Auswahl der Sichtgeräte und welches Ausgabeformat darauf erscheinen soll. Außerdem ist eine weitere Taste vorhanden, um alle Sichtgeräte-Formate auch auf dem Schnelldrucker auszugeben. Die dritte Tastatur ist zur Auswahl der Anwenderprogramme vorgesehen. Das "ALBE"-Programm läuft nach jeder Alarmbearbeitung wieder auf einen MA-PlWA.

Wird die Kontaktendezeit erreicht, setzt das Programm "ALBE" eine sogenannte "Endzelle", die von allen Programmen abgefragt wird. Das hat zur Folge, daß die Enderoutinen bearbeitet werden, und die Programme auf einen WA-ENDE laufen.

VIII. Das Programm "AUBE"

Das Programm "AUBE" nimmt die eigentliche Datenaufbereitung vor und läuft unter der Programmpriorität 9. Das Programm besteht aus 2 Abschnitten der RTT- und der TT-Datenaufbereitung. Wie schon beim Programm "AUBE" erwähnt, wird das Programm bei jedem Alarm des RS gestartet, das heißt, wenn der RS ein Synchwort erkannt hat. Das Synchwort kennzeichnet den Beginn eines neuen Rahmens. Das P3 übernimmt wortweise die Daten vom Rahmensynchronisierer. Ist nun ein Puffer mit einem RTT-Rahmen gefüllt, wird auf den nächsten Puffer umgeschaltet. Zum neuen Rahmen wird die Zeit eingelesen und abgespeichert, während vom RS das Statuswort für den vorherigen Rahmen übernommen wird. Im Statuswort sind Angaben über den Rahmen selbst enthalten, z.B. ob im Synchwort Bitfehler waren, ob die Rahmenlänge zu kurz oder zu lang war usw. Während der neue Rahmen eingelesen wird, wird der vorherige auf Digitalband geschrieben und zur Verarbeitung freigegeben, dabei verzweigt das Programm zum zweiten Abschnitt.

Ein gesamter Meßzyklus umfaßt 20 RTT-Rahmen und beinhaltet alle Meßdaten der Experimente und die Funktionsdaten (Housekeeping-Daten) des Satelliten. Ein RTT-Rahmen enthält jeweils 8 Wortnummern mit den entsprechenden Meßwerten des TT-Rahmens. Der erste enthält z.B. die Wortnummern 0-7 und ihre Inhalte, der zweite dann 8-15 usw. Steht nun ein vollständiger RTT-Rahmen zur Verfügung, untersucht der zweite Teil des Programms zunächst die Wortnummer.

Es müssen wenigstens 2 Wortnummern aus dieser Gruppe erkannt werden, um diesen RTT-Rahmen richtig im TT-Rahmen plazieren zu können. Wir sagten vorhin der erste RTT-Rahmen enthält die Wortnummer 0-7. Erkennt das Programm also 2 Ziffern aus dieser Reihe, z.B. 2 und 5, so wird der Rahmen an der entsprechenden Stelle des TT-Puffers abgespeichert, in diesem Fall also auf die ersten 8 Plätze. Die Wortnummern allein genügen allerdings noch nicht, denn durch schlechten Empfang oder Störungen können einige Rahmen ausfallen. Daher wird die Zeit zur Identifizierung ebenfalls verwendet. Wie vorher schon berichtet, bekommen wir alle 500 ms einen RTT-Rahmen, da aber zu einem TT-Rahmen 20 solcher RTT-Rahmen gehören, entspricht das einer TT-Rahmenlänge von 10 sec. Macht die Zeit zwischen 2 RTT-Rahmen einen Sprung von 10 oder mehr Sekunden, so gehört der letzte RTT-Rahmen bereits zum nächsten TT-Puffer. In diesem Falle wird der vorherige Puffer sofort zur Verarbeitung freigegeben und der nächste TT-Puffer wird aufbereitet. Genauso wird verfahren, wenn die letzte Wortgruppe 152-159 im RTT-Rahmen erkannt wurde. In beiden Fällen wird das "KORD"-Programm gestartet und wieder zum ersten Teil des Programms "AUBE" verzweigt, wo es auf einen MA P3 WA läuft.

IX. Das Programm "KORD"

Das Programm "KORD" koordiniert den Ablauf der Anwenderprogramme und läuft unter der Priorität 10. Vom "BEDI" wurde eine Zelle "Zulässige Anwenderprogramme" gesetzt und vom "ALBE" eine Zelle "Geforderte Anwenderprogramme". Beide Zellen werden mittels eines logischen UND verknüpft, und die tatsächlich möglichen Anwenderprogramme werden vom "KORD" nacheinander in den Kernspeicher geholt und gestartet, wobei sie alle unter der gleichen Programmpriorität laufen. Jedes Programm gibt seine Programmnummer auf einem Display aus, sodaß man sofort sehen kann, welches Programm z.Zt. bearbeitet wird.

Sämtliche Ausgaben werden nicht von den Anwenderprogrammen, sondern vom "KORD" selbst abgewickelt, denn nur das "KORD" weiß, ob noch weitere Anwenderprogramme benötigt werden. Ist das nicht der Fall, wird der TT-Puffer sofort wieder zur Aufbereitung freigegeben. Werden nämlich alle Anwenderprogramme benötigt, so dauert die Verarbeitung länger als 10 sec. Dadurch ergibt sich, daß der eine Puffer schon wieder aufgefüllt ist, während der andere noch nicht abgearbeitet ist. In diesem Fall wird die TT-Aufbereitung bis zur Freigabe des TT-Puffers unterdrückt, die RTT-Rahmen werden aber weiter übernommen und aufs Digitalband geschrieben.

Wenn nun alle Anwenderprogramme durchfahren worden sind, wird der gesamte Pufferbereich normalisiert, d.h. Bit 1 eines jeden Wortes gesetzt (MINUS-Bit). Bei der TT-Aufbereitung wurde erwähnt, daß der RTT-Rahmen mit Hilfe der ermittelten Wortnummer an der entsprechenden Stelle des TT-Puffer abgespeichert wird.

Das heißt, das Minus-Bit wird wieder überschrieben, während es überall dort, wo kein RTT-Rahmen vorhanden war, erhalten bleibt. Die Anwenderprogramme fragen nach "Minus" bei der Verarbeitung ab. Ist das der Fall, wird der Wert als nicht vorhanden angesehen, und es erscheint eine entsprechende Kennung im Ausgabeformat.

X. Das Programm "TEST"

Das Programm "TEST" besteht ebenfalls aus zwei Abschnitten und dient zum Testen der Aufbereitungsanlage. Das Programm "TEST" wird vom Programm "BEDI" bereitgestellt und gestartet.

Es trägt zunächst das "BEDI" aus, merkt sich den Rücksprung und trägt sich selbst wieder unter der Programmpriorität ein. Danach werden die Parameterkarten eingezogen, aus denen die Datenart (Quick-Look A,B,C) sowie die Einstellungen an den Analogbändern, der Schaltkonsole, dem Bit- und Rahmensynchronisierer hervorgeht. Sind alle Geräte versorgt worden und betriebsklar, wenn nicht, wird eine entsprechende Meldung auf dem BBS ausgegeben, wird die gesamte Anlage getestet. Von der deutschen zentralen Bodenstation in Weilheim wird ein ganz bestimmter Datenrahmen über die Kabelleitung zum Kontrollzentrum gesendet. Es werden 1000 Rahmen eingelesen und Bit für Bit untersucht. Anzahl der Fehler wird laufend auf dem Schnelldrucker ausgegeben.

Nach den 1000 Rahmen wird eine Fehler-Statistik auf dem BBS ausgegeben. Falls die Strecke unbrauchbar ist, kann auf die Richtfunkstrecke umgeschaltet werden. Nach diesem Test ist die Anlage betriebsbereit.

Unter der Programmpriorität 22 wird nun ein Teil des Programmes "TEST" bereitgestellt und gestartet. Er fragt zyklisch alle Geräte ab und meldet auftretende Fehler sofort. Anschließend wird das "BEDI"-Programm fortgesetzt.

Bei Kontaktende wird vom "ALBE" eine ENDEZELLE gesetzt. Diese Zelle wird von allen Programmen immer wieder abgefragt, ist sie ungleich Null laufen die Programme sofort auf einen MAENDE oder sie starten vorher noch die Enderoutinen.

Natürlich konnte hier nicht auf alles eingegangen werden, die Anwenderprogramme z.B. blieben ganz unberücksichtigt, obwohl gerade sie genügend Anlaß zur Entwicklung von vielen Unterprogrammen gaben. Vielmehr sollte hier ein Lösungsweg für ein Programmsystem gezeigt werden, das gerade bei Prozeßrechnern vielseitig angewendet werden konnte.

Fig. 4.4: Real Time Telemetry Subframe

WORD NO.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0-3	EI 95 (PCU 2.2)													EI 102 (PCU 2.3) TAP 3			EI 95 (EXP)		EI 15/ATTITUDE MEAS X (EXPERIMENT)					
	THR 1 (EXP)	TAP 1	TAP 2	TAP 3	TAP 4	TAP 1	TAP 2	TAP 3	TAP 4	TAP 1	TAP 2	TAP 3	TAP 4	CH 1	CH 2	CH 3	THR 2	THR 3	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰
4-7	SAME AS LINE 1																EI 95 (EXP)		EI 15/ATTITUDE MEAS Y (EXPERIMENT)					
																	THR 2	THR 3	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰
8-11	SAME AS LINE 1																EI 102 CH 1		WORD NO. OF TT-FRAME (DTU)					
																	TAP 1	TAP 2	0	0	0	0	2 ⁷	2 ⁶
12-15	SAME AS LINE 1																EI 102 CH 2		WORD NO. OF TT-FRAME (DTU)					
																	TAP 1	TAP 2	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰
16-19	SAME AS LINE 1																EI 102 CH 3		1ST WORD OF TT-FRAME (DTU)					
																	TAP 1	TAP 2						

1 ROW = 24 BITS = 12.5 MS

1 SUBFRAME = 5 LINES = 120 BITS = 62.5 MS

BIT RATE: 1920 BITS PER SECOND

1 FRAME = 3 SUBFRAMES = 360 BITS = 0.5 SEC

THE FIRST FOUR WORDS OF EACH RT FRAME ARE THE SYNCHRONIZATION WORDS

SYNCHRONIZATION WORD

0-3	0	0	0	0	0	1	0	1	0	0	0	1	1	0	0	1	1	0	1	1	1	1	1
-----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

01206337	0 00	00	00 000	00 00	0 00	00	00 000	00 00	0 00	00	00 000	00 00	0 00	00	00 000	00 00	0 00	00	00 000	00	76
0 00	00	00 000	00 00	0 00	00	00 000	00 00	0 00	00	00 000	00 00	0 00	00	00 000	00 01	0 00	00	00 000	00	57	
0 00	00	00 000	00 00	0 00	00	00 000	00 00	0 00	00	00 000	00 00	0 00	00	00 000	00 02	0 00	00	00 000	00	14	
0 00	00	00 000	00 00	0 00	00	00 000	00 00	0 00	00	00 000	00 00	0 00	00	00 000	00 03	0 00	00	00 000	00	40	
0 00	00	00 000	00 00	0 00	00	00 000	00 00	0 00	00	00 000	00 00	0 00	00	00 000	00 04	0 00	00	00 000	00	62	
0 00	00	00 000	00 00	0 00	00	00 000	00 00	0 00	00	00 000	00 00	0 00	00	00 000	00 05	0 00	00	00 000	00	00	
0 00	00	00 000	00 00	0 00	00	00 000	00 00	0 00	00	00 000	00 00	0 00	00	00 000	00 06	0 00	00	00 000	00	70	
0 00	00	00 000	00 00	0 00	00	00 000	00 00	0 00	00	00 000	00 00	0 00	00	00 000	00 07	0 00	00	00 000	00	03	

the first of 20 rtt-frames including sync pattern
in octal and decimal notation.

output equipment: line printer
keyboard program selection: 6

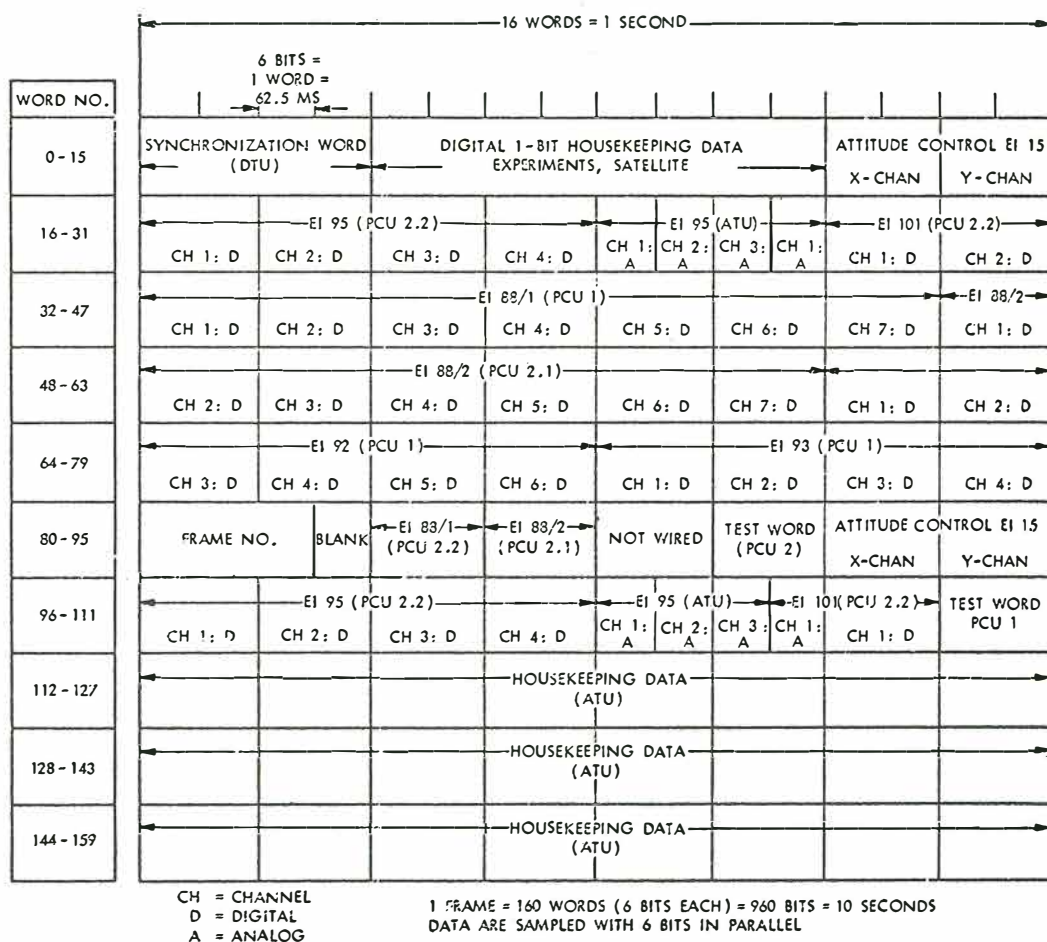


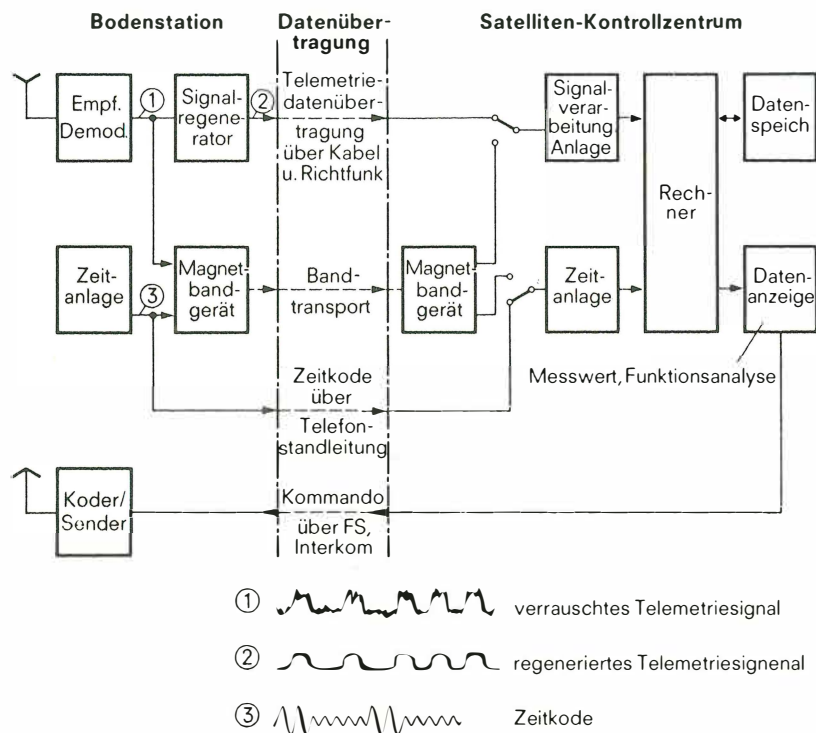
Fig.4.5: Tape Telemetry Frame

76571440		62	00	70	03	76	40	07	63		0		0
524E3 1	524E3 1	524E3 1	524E3 1	524E3 1	524E3 1	00	00	00	00	524E3 1	524E3 1	524E3 1	524E3 1
1EO 1	1EO 1	1EO 1	1EO 1	1EO 1	1EO 1	1EO 1	1EO 1	1EO 1	1EO 1	1EO 1	576EO 1	576EO 1	576EO 1
576EO 1	576EO 1	576EO 1	576EO 1	576EO 1	576EO 1	576EO 1	115E1 1	1EO 1	1EO 1	1EO 1	1EO 1	1EO 1	1EO 1
1EO 1	1EO 1	1EO 1	1EO 1	1EO 1	1EO 1	1EO 1	1EO 1	1EO 1	1EO 1	1EO 1	1EO 1	1EO 1	1EO 1
57 267		00	524E3 1	224E1 1	1EO 1	0EO 1				0			0
524E3 1	524E3 1	524E3 1	524E3 1	524E3 1	00	00	00	00	00	524E3 1	0EO 1	0EO 1	0EO 1
65 05	41 42	53 01	62 15	57 60	35 41	47 77	70 00						
36 35	36 66	64 64	65 62	64 64	64 64	64 64	63 63						
46 65	00 00	00 00	55 00	64 00	00 00	36 00	00 00						

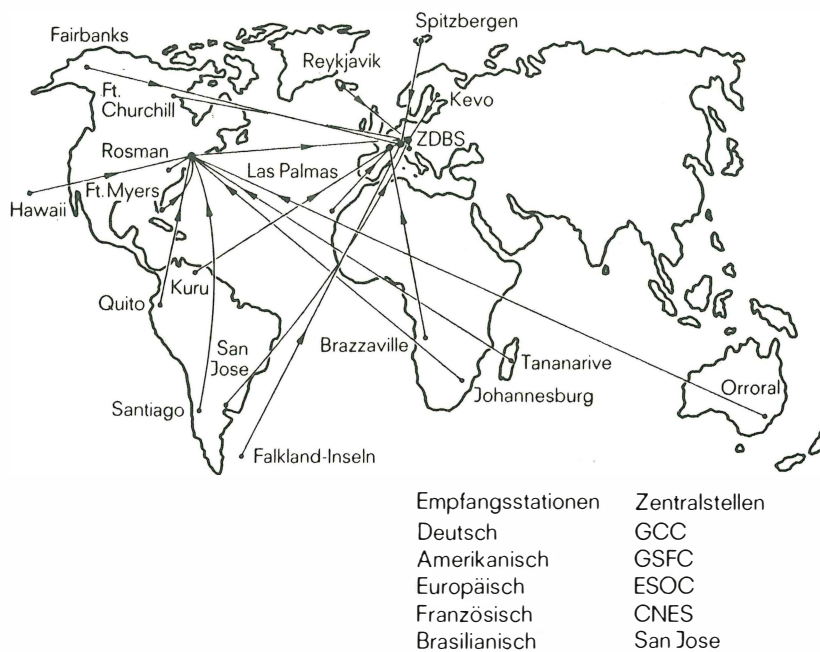
tt-frame all channels including sync pattern in octal
and decimal notation.

output equipment: line printer

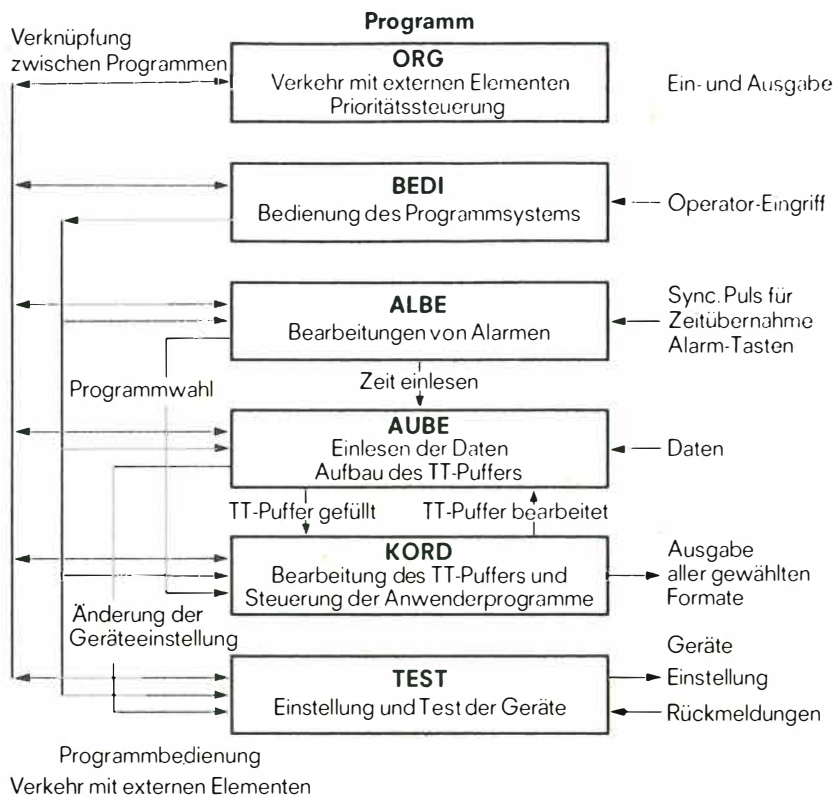
keyboard program selection: 7



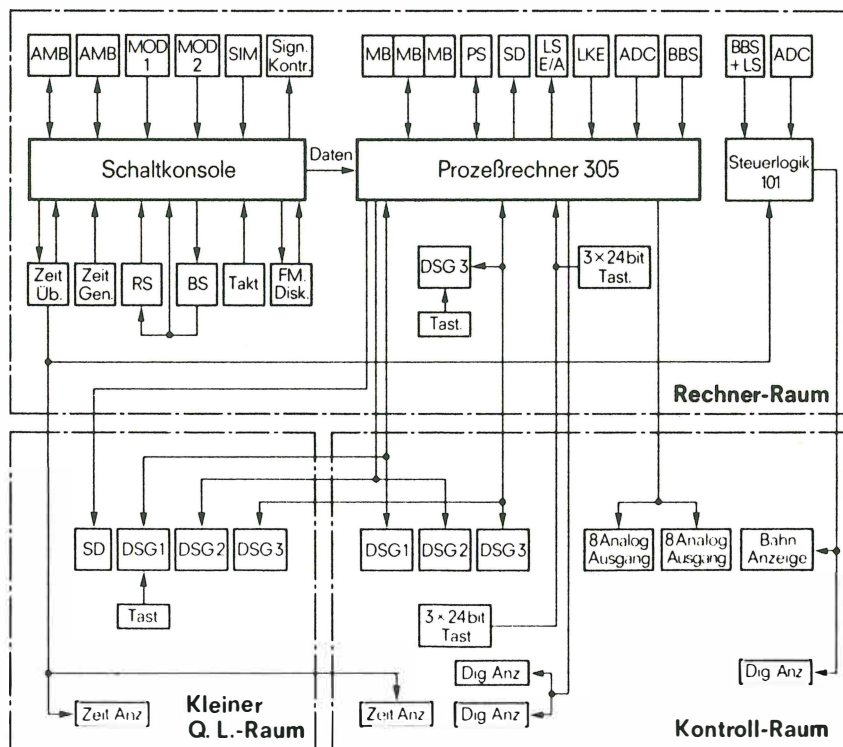
Schematische Darstellung eines Bodensystems zur Satellitenüberwachung



Bodenbetriebssystem für Satellit AZUR



Programmsystem AZUR



Datentechnische Anlage des GCC I. Ausbaustufe

K. Weise

Physikalisch-Techni-
sche Bundesanstalt,
Braunschweig

"Ein Programmsystem für die Meßdaten-
erfassung und -verarbeitung bei Experi-
menten am Forschungs- und Meßreaktor
der Physikalisch-Technischen Bundesan-
stalt"

Ein Programmsystem für die Meßdatenerfassung
und -verarbeitung bei Experimenten am
Forschungs- und Meßreaktor der Physikalisch-
Technischen Bundesanstalt

K. Weise

Physikalisch-Technische Bundesanstalt, Braunschweig

Seit Juni 1968 ist am Forschungs- und Meßreaktor der Physikalisch-Technischen Bundesanstalt in Braunschweig ein Siemens-Prozeßsystem für die automatische Erfassung, Überwachung, Verarbeitung und Archivierung der Meßdaten physikalischer Experimente eingesetzt. Dieses System besteht aus einer Zentraleinheit 305 mit 16 K-Arbeitsspeicher, Bedienungsblattschreiber BBS und zwei weiteren Blattschreibern an einem Fernschreibelement FSK für die Bedienung und Datenausgabe im Klartext, weiterhin einem Lochstreifenelement LSK, einem Plattenspeicher PSK als Programm- und Datenzwischenspeicher und einer Zwillingsmagnetbandeinheit SKU/MBK für die Archivierung der Meßdaten und deren Vorbereitung für eine Weiterverarbeitung auf einer anderen Anlage. An Prozeßelementen für die Datenerfassung sind vorhanden ein Element P1K für die programmgesteuerte Digital-Ein- und -Ausgabe mit 30 x 24 Digitaleingängen einer GRODE, 4 x 24 Digitalausgängen einer ELDA und 24 Alarmeingängen einer ALDE, außerdem drei Elemente P3K für die externgesteuerte Digital-Ein- und -Ausgabe, nämlich eine Digitaleingabesteuerung P3ES/N, eine Inkrementsteuerung P3ES/I für die Vielkanalanalyse und eine Digitalausgabesteuerung P3AS. Die anhängenden Eingabegeräte P3EG besitzen Eingänge für je 4 Worte, das Ausgabegerät P3AG Ausgänge für nur ein Wort. Eine Vergrößerung der Anzahl der Inkrementeingänge auf 8 Worte zu 14 bit ist geplant. Auch die Digitalausgabe der P1K soll auf 8 Worte und die Alarmeingabe auf 48 Alarmeingänge ausgebaut werden.

An dieses Prozeßsystem sind an Experimentiereinrichtungen und sonstigen Geräten z.Z. angeschlossen: über das Prozeßelement P1K ein 10 Hz-Impulsgeber als Uhr für Zeitüberwachungszwecke, ein Meßstand für Neutronenflußdichtebestimmungen, ein 4 K x 20 bit

Vielkanalanalysator, der von der Zentraleinheit auch als externer Kernspeicher benutzt werden kann, und ein sogenannter Befehlsgeber, ein Gerät, von dem gleich noch die Rede sein wird, zur Bedienung der Anlage vom Experiment aus; bei den Elementen P3K hängen an der Eingabe eine schnelle Uhr mit Zwischenspeicher für ein $\gamma\gamma$ -Koinzidenz-Experiment zur Messung von Lebensdauern angeregter Kernzustände, an der Inkrementeinrichtung schnelle Analog-Digital-Konverter für die Vielkanalanalyse bei der Spektrometrie von Neutronen und Gammastrahlung im Rahmen verschiedener Experimente an Strahlrohren des Reaktors, weiterhin an der Ausgabe ein Sichtgerät für die Darstellung des Inhalts von Arbeitsspeicherbereichen. Mittels eines Punktdruckers kann das Bild des Sichtgerätes auch aufgezeichnet werden.

Geplant ist in nächster Zeit der Anschluß von insgesamt 4 Sichtgeräten, die zusammen mit der gleichen Zahl von Befehlsgebern als Kommandostände für die Experimentatoren dienen sollen, von denen aus diese ihr Experiment bedienen und überwachen und auch den Rechner ansprechen können. Die Wechselpufferausgabe der P3K hat sich als nicht gut geeignet für die Bildausgabe bei mehreren Sichtgeräten erwiesen, da die Zentraleinheit dabei mehr als sinnvoll belastet wird. Wie Herr Schuster in seinem Vortrag beschreibt, werden wir daher die Ausgabesteuerung P3AS verändern und die Adresssteuerung experimentgeordnet extern durchführen. Ein Programm für die Bildausgabe entfällt dabei. Entsprechendes ist vorgesehen für die schnelle Dateneingabe. Außerdem sollen noch an das Prozeßelement P1K eine Ausgabeschreibmaschine für eine entfernte Station, ein optischer Schnelldrucker und ein weiterer Vielkanalanalysator als Hilfsspeicher angeschlossen werden, damit die Daten einiger arbeitsspeicherverschlingenden Experimente möglichst extern akkumuliert werden können.

In der ersten Zeit wurde das Prozeßsystem noch nicht "on line" betrieben. Die Meßdaten fielen zur Verarbeitung mit Hilfe von ALGOL-Programmen hauptsächlich in Form von Lochstreifen aus Vielkanalanalysatoren an. Leider erfolgt das Ausstanzen der

Zahlenwerte bei jedem Gerät in einem anderen Code oder in einer anderen Anordnung auf dem Streifen, so daß es notwendig war, ein PROSA-Programm zu erstellen für die automatische Code- und Formaterkennung, die Datenentschlüsselung und deren Bereitstellung auf einem ALGOL-Feld vor Ablauf des ALGOL-Verarbeitungsprogramms. Mit diesem Programm können 7 verschiedene Vielkanallochstreifentypen eingelesen, ggf. auch auf einem Blattschreiber ausgegeben oder auf Lochstreifen im ALGOL-Format umgesetzt werden. Dieses Programm wurde im Laufe der Zeit von uns ergänzt durch weitere Datentransferprogramme in PROSA für die Datenausgabe auf Blattschreiber und Sichtgerät, die Daten- und Programmübertragung auf und von Lochstreifen, Plattenspeicher und Magnetband, und für die Dateienorganisation auf Magnetband und Platte, d.h. Etikettieren, Listen, Löschen, Kopieren, Archivieren und Umcodieren in die Zahlendarstellung der Siemens 4004, die für die Weiterverarbeitung im Rechenzentrum der Physikalisch-Technischen Bundesanstalt zur Verfügung steht. Einige wichtige Datentransfers können auch vom Experiment her mit Hilfe des erwähnten Befehlsgebers ausgelöst werden. Die Programme sollen, wenn die neuen Standard-Ein- und -Ausgabeaufrufe zur Verfügung stehen, noch so weit ausgebaut werden, daß durch Bedienung, "on line" oder auch durch ALGOL-Programme praktisch jede wichtige und sinnvolle Daten- oder Programmübertragung in bequemer Weise mit Hilfe eines einzigen Programmsystems ausgeführt werden kann.

Fast alle unsere Meßdatenauswerteprogramme sind in ALGOL geschriebene mathematische Programme, die, wie sich gezeigt hat, häufig geändert oder gar durch neue ersetzt werden. Da nun eine Korrektur von Lochstreifen über Blattschreiber sehr mühselig ist, haben wir ein Korrekturprogramm für ALGOL-, PROSA- und einige Datenlochstreifen entwickelt. Auch die oft durchzuführenden ALGOL-Übersetzungen haben wir uns etwas vereinfacht mit Hilfe eines speziellen Steuerprogramms für den ALGOL-Compiler (z.Z. ist dies allerdings ein älterer Compiler). Hiermit ist es u.a. möglich, auf der Platte stehende ALGOL-Quellenprogramm-Stücke

nach Wahl zu binden. Gewisse Prozeduren, etwa die Ein- und Ausgabe-Prozeduren, können auch standardmäßig automatisch in die Programme eingefügt werden. Leider beanspruchen die ALGOL-Programme bei der Übersetzung und beim Ablauf relativ viel Arbeitsspeicherraum, so daß es uns bisher leider nur selten möglich war, mathematische Rechnungen und den oft tagelang dauernden Experimentierbetrieb simultan durchzuführen oder sogar zu koppeln, wie es außerordentlich wünschenswert wäre. Dies liegt daran, daß das Organisationsprogramm 5K und auch die Meßdatensammelpuffer mitunter bis zu 8K Speicherplätze benötigen. Leider ist aber der Arbeitsspeicher der 305 nicht ausbaufähig. Eine Segmentierung der ALGOL-Programme hilft auch meist nicht viel oder verbietet sich wegen der Rechenzeit. Ein Externspeicherelement KSK ist auch nicht geeignet, da darin nicht gerechnet und keine Vielkanaldaten akkumuliert werden können.

Andererseits liegt noch recht viel Rechenkapazität in manchen Nächten und an Wochenenden brach, zu Zeiten, an denen aber kein Bedienungspersonal zur Verfügung steht. Um diese Zeiten auch ausnutzen zu können, planen wir die Programmierung eines ALGOL-Monitors, mit dem es möglich ist, ohne die Mitwirkung eines Operateurs, völlig automatisch eine Anzahl von ALGOL-Programmen sequentiell zu übersetzen und ablaufen zu lassen. Alle Anweisungen für den Ablauf der Programmfolge und Zwischenfälle sowie alle zu bindenden Quellenprogrammstücke, Codeprozeduren und die Daten für die Rechnung müssen dann natürlich vorbereitet auf der Platte oder auf Magnetband stehen. Da die Ausgabe von Ergebnissen auf Blattschreiber sehr zeitraubend sein kann, soll diese stark über die Platte gepuffert und von einem speziellen, simultan zu den ALGOL-Programmen laufenden Ausgabeprogrammen übernommen werden. Für einen solchen Monitor ist es nicht notwendig, die vorhandenen Compiler oder die von diesen erstellten Maschinencodeprogramme zu verändern, es genügt im wesentlichen, durch einen kleinen Eingriff des Monitors in das Organisationsprogramm ORG, alle Organisationsaufrufe mit UNT(5) der zu überwachenden

Programme während des Ablaufs abzufangen, zu erkennen, ggf. durch eine bestimmte Tätigkeit des Monitors zu ersetzen und wieder zurückzuspringen ins aufrufende Programm oder auch ins ORG, wenn nichts ersetzt wurde. Auf diese Weise können z.B. leicht alle Blattschreiberaufrufe durch entsprechende Programmstücke für den Plattenspeicher ersetzt werden. In diesem Zusammenhang sind auch die neuen Standard-Organisationsaufrufe von besonderem Interesse.

Sehr wichtig für die an unser Prozeßsystem angeschlossenen Versuche ist das von uns erstellte Betriebsprogrammsystem für die automatische Meßdaten-Organisation bei Reaktorexperimenten, abgekürzt: AMOR. Dieses Programm ermöglicht u.a. die Bedienung der Anlage für die Experimente über Blattschreiber und über die schon mehrfach erwähnten Befehlsgeber. Diese bestehen aus 48 Binärschaltern, an denen die Bits von 2 Wörtern eingestellt werden können, d.h. die Experimentnummer (3 bit), ein Befehlscode (7 bit), Bereichsadressen (14 bit), relativ zur Anfangsadresse des jedem Experiment zugeordneten Arbeitsspeicherbereichs, Dateinamen und andere Parameter. Die Befehlsgabe erfolgt nach der Einstellung über einen Alarmknopf. Wird bei der Befehlsverarbeitung ein Fehler festgestellt, so leuchtet eine Signallampe auf. Vom Experimentator können Befehle gegeben werden zum An- und Abmelden des Experiments, zum Starten und Beenden von Messungen, für verschiedene Datentransfers, z.B. in eine bestimmte Plattendatei oder auf Blattschreiber, zur Bedienung des Sichtgeräts, die bald allerdings direkt an der Steuerung erfolgen soll, weiterhin zum Löschen, natürlich nur des eigenen Puffers, für die Ein- und Ausgabe von Parametern, z.B. Dateinamen, für die Ausgabe der Fehlerart beim Auftreten eines solchen, zum Einleiten und Beenden automatischer Meßzyklen und insbesondere zum Anstoßen beliebiger, vorbereiteter Programme auf der Platte für bestimmte Zwecke. Solche Befehle oder Anweisungen werden von einem Teil des Alarmprogramms oder vom Blattschreiberbedienungsprogramm des AMOR oder auch von einem beliebigen anderen angekoppelten Programm in Form eines Aufrufs über ein spezielles Unterprogramm an ein Befehlssteuerprogramm weitergereicht. Dieses prüft die Befehle, erledigt sie z.T.,

leitet die Protokollierung ein, wenn gewünscht, holt die entsprechenden Befehlsausführungsprogramme von der Platte und startet sie nach der Übergabe der Parameter. Dieses Weitergeben der Befehle geschieht ähnlich wie beim ORG mit Hilfe von Aufrufen. Z.Z. können nur maximal drei Befehle gleichzeitig bearbeitet werden. Dieser Nachteil soll jedoch verbessert werden durch eine Befehlsaufrufkettung wie bei den Standard-Organisationsaufrufen, wobei noch besonders dringliche Befehle an den Kopf der Warteschlange gesetzt werden können. Die meisten Tätigkeiten des AMOR werden über solche automatisch oder von Hand ausgelösten Befehlsaufrufe gesteuert. Für die Ausführungsprogramme, die beliebig ergänzt werden können und nur bei Bedarf in den Arbeitsspeicher geholt werden, stehen zwei feste Programmpuffer zur Verfügung. Dies ist bei Berücksichtigung der ebenfalls festen Meßdatenpuffer der Experimente ein Nachteil. Zukünftig soll die gesamte räumliche und zeitliche Speicherplatzorganisation von einem speziellen Programmteil übernommen werden, so daß der viel zu kleine Arbeitsspeicher nach Möglichkeit immer optimal und den schnell wechselnden Anforderungen entsprechend ausgenutzt werden kann. Ein Problem hierbei ist allerdings noch nicht gelöst: nämlich das Komprimieren laufender Programme im Arbeitsspeicher, hauptsächlich wegen der Organisationsaufrufe, die leider nicht verschoben werden können.

Bei den Wissenschaftlern, die ihre Experimente mit Hilfe unserer Anlage durchführen, besteht das dringende Bedürfnis, auch während des Meßbetriebes gewisse Daten aus den Meßwerten rechnerisch ermitteln zu können, z.B. Vorergebnisse für Überwachungszwecke. Diese Rechenwünsche entstehen meist plötzlich, sollen möglichst sofort erfüllt werden und ändern sich sehr schnell, umfassen aber im allgemeinen nur einfache arithmetische Rechenoperationen. Jedesmal ein ALGOL-Programm zu erstellen, ist recht unrationell, auch muß der Wunsch meist mit großem Zeitverlust um wenigstens einige Stunden zurückgestellt werden, nicht so sehr wegen der zeitlichen Auslastung der Anlage, als vielmehr mangels Arbeitsspeicherraums für Übersetzung und Rechnung. Wünschenswert wäre

deshalb ein Programm für das Prozeßrechnersystem 300 für Rechenzwecke, das von Blattschreibern, evtl. sogar in einer Art "timesharing" bedient werden kann und wenigstens die Möglichkeiten einer komfortablen programmierbaren Tischrechenmaschine bietet. Es sollte Standardfunktionen enthalten und insbesondere auch eine Wechselwirkung mit dem Prozeß erlauben, etwa wie bei der Programmiersprache BASIC, ergänzt durch spezielle Anweisungen für den Prozeß, z.B. wie bei den Hewlett-Packard-Rechnern. Höchstens 1-2K Speicherraum darf das Programm beanspruchen. Auf Kosten der Rechenzeit, die aber in vielen Fällen nicht sehr kritisch ist, wird es deshalb aus vielen Segmenten, die nur bei Bedarf im Arbeitsspeicher liegen, bestehen müssen. Nach unseren noch nicht ausgereiften Überlegungen erscheint die folgende Organisation des Programmsystems sinnvoll: Ein Programmteil codiert die Programmieranweisungen, die am Blattschreiber gegeben werden, zu einem Quellenprogramm, das auf der Platte gespeichert wird und aus einer Folge von Anweisungen besteht. Jeder möglichen Anweisung ist ein vorprogrammierter, in Maschinencode vorliegender Baustein, ebenfalls auf der Platte zugeordnet. Beim Rechenlauf, der auch vom Prozeß, z.B. durch Alarmer ausgelöst werden kann, interpretiert ein Steuerprogrammteil jede durchlaufene Anweisung des Quellenprogramms, holt den jeweils zugeordneten Baustein von der Platte, falls er sich noch nicht im Arbeitsspeicher befindet, und durchläuft diesen als Unterprogramm. In Verbindung mit dem Betriebsprogramm AMOR und speziellen Programmen für besondere Tätigkeiten würde solch eine Möglichkeit zur "on line"-Programmierung und -Rechnung ein außerordentlich flexibles Prozeßprogrammsystem darstellen, das schnell wechselnden Bedürfnissen und sich ändernden Versuchseinrichtungen leicht auch während des laufenden Experimentierbetriebes angepaßt werden kann.

J. Radünz

Deutsche Forschungs- und
Versuchsanstalt für Luft-
u. Raumfahrt e.V.
Oberpfaffenhofen

"Betriebssystem ORG I - 305 GCC
für Stapelverarbeitung"

1. Aufgabe

Das Betriebssystem ORG I - 305 wird so erweitert, daß die Steuerung des Programmablaufes durch Lochkarten vorgenommen wird.

Der Zweck der Erweiterung ist:

1. Erfassung der Rechen- bzw. Stillstandszeit am Siemens-Rechner 305.
2. Vereinfachung der Bedienung. Die gewünschten Programme werden automatisch bereitgestellt und gestartet.

Die Steuerung erfolgt durch eine JOB-CONTROL-Karte (siehe Seite 6).

2. Betriebsarten

Das erweiterte Betriebssystem sieht 3 Betriebszustände vor:

1. Automatisches Bereitstellen und Starten von Programmen (NORM).
2. Bereitstellen und Starten von Programmen über den Bedienungsblattschreiber (SPEZ).
3. Rechenanlage wird nicht benutzt. Bereitstellen und Starten von Programmen ist nicht möglich (PAUS).

Die Betriebsart NORM wird durch die Lochung §JOB im Feld 1 (Spalten 1 bis 4) eingestellt.

Durch die Lochung §JCE im Feld 1 der JOB-CONTROL-Karte können die Betriebsarten SPEZ und PAUS gewählt werden.

3. Aufbau der JOB-CONTROL-Karte

Die Lochkarte ist in 23 Felder eingeteilt. Die Bedeutung ist in der folgenden Tabelle zusammengestellt.

<u>Feld</u>	<u>Bedeutung</u>	<u>Loch- vorschrift</u>	<u>Bemerkungen</u>
1	Kartenart	7	'§JOB' für NORM '§JCE' für SPEZ und PAUS
2	Programmträger	4	'/P//' = Platte '/K//' = Karte '/S//' = Streifen
3	Programmname	7	'PAUS' in Verbindung mit §JCE stellt den Zustand PAUS ein
4	Name des Benutzers	6	Name muß linksbündig eingetragen werden
5	Name des Projektes	6	
6	Voraussichtliche Lauf- zeit in Minuten	6	Eintrag rechtsbündig
7 bis 23	Verwendete Extern- Geräte	4	J rechtsbündig, wenn entsprechendes Gerät benötigt wird.

Die Lochvorschrift besagt, daß die entsprechenden Felder bei bestimmten Betriebszuständen ausgefüllt werden müssen, damit die JOB-CONTROL-Karte akzeptiert wird. Die angegebenen Zahlen ergeben sich durch Addition der folgenden Kennzahlen:

4	Zustand NORM
2	Zustand SPEZ
1	Zustand PAUS.

4. Betriebsart NORM

Das Programm, das im Feld 3 benannt ist, wird von dem im Feld 2 angegebenen Programmträger unter der Programmnummer 2 ab Zelle 2000 bereitgestellt und gestartet. Die Liste der Externgeräte und die Laufzeit des Programmes werden auf dem Bedienungsblattschreiber ausgegeben.

Weiterhin wird die erste freie Zelle nach dem Programm ausgegeben, um eventuell zur gleichen Zeit weitere Programme verarbeiten zu können, die den Ablauf des automatisch gestarteten Programmes nicht stören.

5. Betriebsart SPEZ

Bei der Betriebsart SPEZ können beliebige Programme vom Blattschreiber aus bereitgestellt und gestartet werden.

Es ist zu beachten, daß die Programmnummern 1 und 2 belegt sind.

Nach Abschluß aller Programme ist das Lochwort "§JOB" am Bedienungsblattschreiber einzugeben, um den normalen Betriebszustand wieder herzustellen.

6. Betriebsart PAUS

Während des Betriebszustandes PAUS ist keine Programmbearbeitung möglich. Der normale Betriebszustand kann durch Eingabe des Codewortes "§JOB" über den Bedienungsblattschreiber wieder hergestellt werden.

7. Abrechnung der Rechenzeit

Am Beginn und Ende des Programmlaufs wird die Zeit protokolliert.

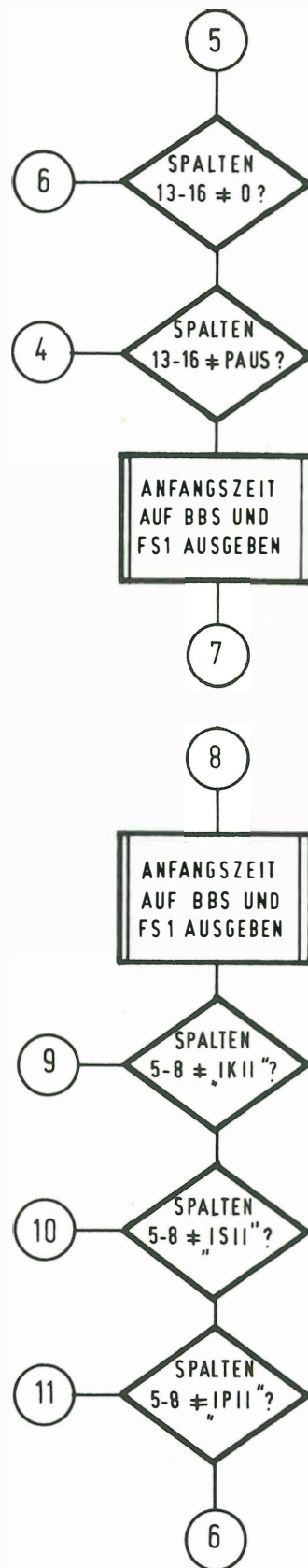
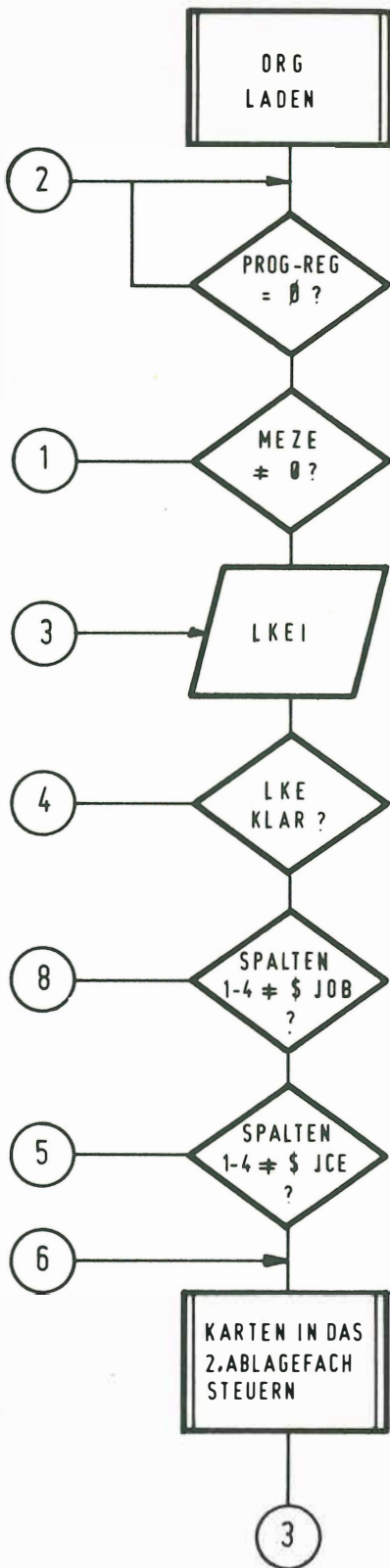
Die Betriebszustände SPEZ und PAUS werden abrechnungsmäßig wie Programmläufe behandelt.

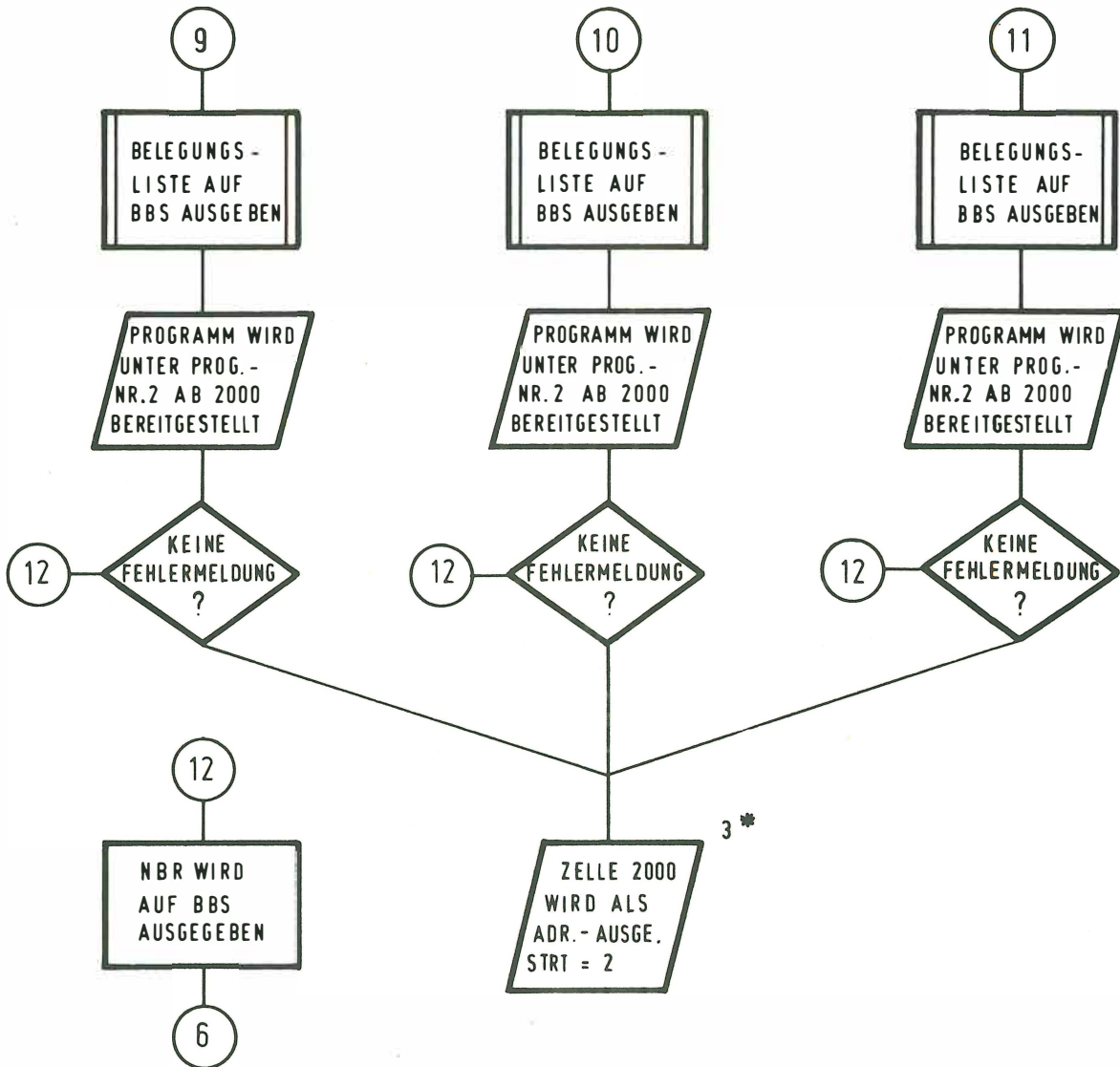
8. Programmnormierungen

- Alle Programme sollen im Adreßteil der ersten Zelle die Adresse der ersten freien Zelle hinter dem Programm enthalten.
- Bei Programmen, die mehrere Programmnummern belegen, ist eine Angabe der belegten Programmnummern auf dem Bedienungsblattschreiber erforderlich.



Projekt: _____
Datum: 8.6.70
Seite: - 6 -





- 1* Beim Zeiteinlesen wird die Zelle "MEZE" auf Null gesetzt und eine Kontrolle durchgeführt, ob alle Parameter wie Programm-Name, Benutzer und Projekt vorhanden sind, wenn eine Angabe fehlt wird die Karte angesteuert.
- 2* Das Programm gibt den Text "Programm 2 belegt Job-Control beendet, Start durch \$ Job" Auf BBS Aus
- 3* In der ersten Zeile steht die letzte ADR des gesamten Programms.

#ANF15. 7.70 GMT 8H 2M 46S N:KONT P:0000 B:KORNDORFER

DAUER DES JOB:0020MIN

BENUTZTE GERAETE

PS1

LKE

ZTA

MB1

FREI AB: 11209

BELEGTE PROG.-NUMMERN SIND 2 3 4

PROG.-ENDE

#END15. 7.70 GMT 8H 2M 21S

#ANF15. 7.70 GMT 8H 2M 29S N:LSAB P:0107 B:SCHROETER

DAUER DES JOB: 15MIN

BENUTZTE GERAETE

PS1

LKE

SD1

MB1

FREI AB: 16379

GIB LISTENLAENGE (4-STELLIG);1000;

UEBERSETZERENDE ZELLE 11597;

GIB +(WEITER)/ -(WIEDERHOLEN);+;PAUE;;PSTT;;

DL1-ENDE;

DL2-ENDE;PENE;;

#END15. 7.70 GMT 8H 5M 51S

A U S D R U C K D E S J O B - C O N T R O L A U F D E M B B S .

#END15. 7.70 GMT 7H 5M 34S
#ANF15. 7.70 GMT 7H 5M 41S N:XYZX P:NEIN B:PAUS
#END15. 7.70 GMT 7H 20M 23S
#ANF15. 7.70 GMT 7H 20M 31S N:PROT P:SYST B:RADUENZ
#END15. 7.70 GMT 7H 27M 48S
#ANF15. 7.70 GMT 7H 31M 1S N:LSAB P:0000 B:KORNDORFER
#END15. 7.70 GMT 7H 42M 55S
#ANF15. 7.70 GMT 7H 43M 2S N:KONT P:0000 B:KORNDORFER
#END15. 7.70 GMT 7H 45M 45S
#ANF15. 7.70 GMT 7H 47M 39S N:PSMB P:0000 B:KORNDORFER
#END15. 7.70 GMT 7H 50M 34S
#ANF15. 7.70 GMT 7H 50M 43S N:NEIN P:NEIN B:PAUS
#END15. 7.70 GMT 7H 54M 14S
#ANF15. 7.70 GMT 7H 54M 21S N:KONT P:0000 B:KORNDORFER
#END15. 7.70 GMT 7H 58M 1S

A U S D R U C K D E R R E C H E N Z E I T A U F D E M
F S 1 U N D L O C H S T R E I F E N.

J. Becker

Staatl. Ingenieurschule
Paderborn

"Software-Erfahrungen beim Einsatz
der 303 an der Staatlichen Ingenieur-
schule Paderborn"

Software-Erfahrungen beim Einsatz der 303
an der Staatlichen Ingenieurschule Paderborn

Die Staatliche Ingenieurschule Paderborn hat im Frühjahr 1967 eine Siemens 303 in folgender Ausstattung erhalten: 8 K Kernspeicher, Lochstreifen-Ein- und -Ausgabe LSK, Lochkarten-Ein- und -Ausgabe LKK IBM 26, Bedienungslochstreifenleser BLL, Bedienungsblattschreiber T 100. Organisationsprogramm und PROSA-Übersetzer wurden sofort, ALGOL-Übersetzer wenig später, FORTRAN-Übersetzer geraume Zeit danach geliefert. Von Anfang an konnte der 305-Befehlsvorrat verwendet werden, doch mangelte es bei der Programmierung in PROSA an zuverlässigen Konvertierungsroutinen. Andererseits gilt in Paderborn der Grundsatz, daß kein Student seine Ausbildung beendet, ohne wenigstens problemorientiert programmieren zu können (3. Semester, 2-stündig FORTRAN), und daß die Studierenden der Fachrichtung Elektrotechnik zusätzlich PROSA lernen (1. Semester, 4-stündig; 2. Semester, 1-stündig). Zu Anfang wurden von den Studenten die fehlenden Umwandlungs- und anderen Hilfsprogramme erstellt. Wegen der geringen Kernspeichergröße gestattet nämlich der PROSA-Übersetzer keine Makro-Verarbeitung und keine Verwendung von Extern-Namen, so daß die Verwendung der Organisationsaufrufe und das Binden von Anwenderprogrammen mit vorübersetzten Unterprogrammen sehr erschwert ist. Daher wurden Programme entwickelt, die das Binden beliebiger Routinen vom Bibliotheksband (Lochstreifen) und die Verwendung der Standard-Makro-Aufrufe gestatten. Dabei erfolgt gleichzeitig die Übertragung des PROSA-Programms von den für den Programmierer bequemer Lochkarten auf den schnelleren Lochstreifen und eine fast zeitlose Auflistung des Quellprogramms. Das Bibliotheksband enthält sowohl eigene wie die von Siemens gelieferten, teilweise korrigierten und modifizierten Unterprogramme. Die Aufrufe insbesondere der Funktionsgeneratoren sind vereinheitlicht worden. Die Beschreibung "Unterprogramm-bibliothek" enthält alle erforderlichen Informationen.

Die Paderborner Ingenieurschule verfügt daneben über eine ganze Reihe von Anwenderprogrammen aus den verschiedensten Gebieten, z.B. Netzplantechnik, Linearprogrammierung, Statistik, Elektrotechnik, Farbmatrik, Verwaltung. Ein Verzeichnis der Programme ist ab Oktober 1970 verfügbar. Zu erwähnen ist der in Paderborn entstandene Formelübersetzer, der die Eingabe einer beliebigen ALGOL-mäßig geschriebenen Formel über LKK oder BBS gestattet. Die Berechnung erfolgt sofort nach Anforderung der verwendeten Parameter. Danach kann die bisherige Formel erneut verwendet werden oder eine neue eingegeben werden.

Der Vorrat an FORTRAN-Programmen ist sehr bescheiden, da auf der 8 K - Anlage nur sehr kleine Programme gebunden werden können. ALGOL lernen in Paderborn nur die Studierenden der Abteilung Informationsverarbeitung (6.Semester, 2-stündig).

Den Besuchern des Paderborner Rechenzentrums werden eigens dafür geschaffene Programme vorgeführt, die z.B. den Simultanbetrieb zeigen, eine Stichwort-Unterhaltung ermöglichen, Kalenderdaten errechnen und die Reaktionszeit des Gastes messen.

Die Paderborner Anlage ist für die dortigen Belange viel zu klein und zu langsam. Sie ist praktisch pausenlos in Betrieb, auch wenn das Logbuch - wegen fehlender Eintragungen - im Mai 1970 nur 649 Rechenstunden ausweist. Die Bestellung der Rechenzeit geschieht eine Woche im voraus über eine spezielle Liste, die stets eine Stunde nach Auslegen komplett ist. Je Semester werden 270 Studierende in FORTRAN, 120 Studierende in PROSA und 25 Studierende in ALGOL ausgebildet. Jeder Student muß in jeder von ihm erlernten Sprache abschließend ein kleineres Programm selbständig erstellen. Hier bietet sich anderen 300-Benutzern die Chance, in Paderborn derartige Programme anfertigen zu lassen. Daneben ist es auch möglich, ein größeres Problem von einem oder mehreren Studenten als Abschlußarbeit programmieren zu lassen. Entsprechende Anfragen möge man an das Rechenzentrum der Staatlichen Ingenieurschule Paderborn richten.

Langemack
Technische Universität
Braunschweig

"Software für einen
Digital - Plotter"

Software für einen Digital-Plotter

Das Institut für Regelungstechnik hatte schon vor der Installation des Prozeßrechners Siemens 305 am Hochschulrechenzentrum Erfahrungen mit Digitalrechnern gesammelt. U.a. stellte sich heraus, daß für die Auswertung, Veröffentlichung und Diskussion von Rechenergebnissen eine graphische Darstellung der Ausgabedaten in fast allen Fällen unerläßlich ist. Daher wurde für die 305 als zusätzliches Externgerät ein Digital-Plotter vorgesehen. Es handelt sich hierbei um einen Trommelplotter der Firma Benson-France.

Der Plotter hat 4 steuerbare Schrittrichtungen, durch Kombination sind Schritte unter 45^0 möglich, so daß insgesamt 8 Richtungen zur Verfügung stehen (Bild 1)

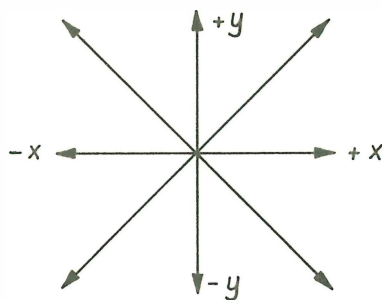


Bild 1

Die Schrittweite beträgt $1/10$ mm; es sind 700 Schritt/sec möglich.

Der Plotter ist über ein Interface am Lochstreifenausgabekanal angeschlossen; es kann also jeweils nur der Plotter oder der Lochstreifenstanzer angesprochen werden. Die Lochstreifensteuerung begrenzt die Zeichengeschwindigkeit auf 300 Schritte/sec.

Für die Ansteuerung stehen 6 Befehle zur Verfügung, 4 Befehle für die Schreibrichtungen und 2 für das Heben und Senken des Zeichenstiftes. Die 6 Befehle sind den 6 bit eines Teilwortes zugeordnet; ein Wort kann also 4 Befehle enthalten. Die Ausgabe der Befehle erfolgt im Modus "Binäre Lochstreifenausgabe".

Jede zu zeichnende Kurve muß in 1/10 mm Schritte aufgelöst werden. Hierfür sind Unterprogramme entworfen worden. Die wesentlichen Unterprogramme seien hier kurz erläutert.⁺)

Wichtig ist zunächst die geradlinige Verbindung von Kurvenpunkten. Bild 2 zeigt die Auflösung einer Geraden in die einzelnen Schritte. Der maximale Fehler ist hierbei die Hälfte der Schrittweite.

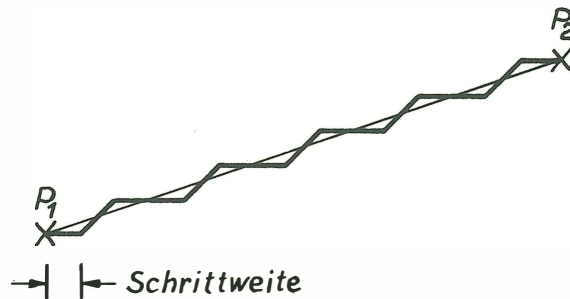


Bild 2:

Eine weitere Variante ist die krummlinige Verbindung von Kurvenpunkten. Durch die Punkte (P_0, P_1, P_2) wird ein Kurvenzug 3. Ordnung gelegt, der sich mit dem Parameter t als $\varphi = \alpha_0 + b_0 \cdot t + c_0 \cdot t^2 + d_0 \cdot t^3$ schreiben läßt (Bild 3).

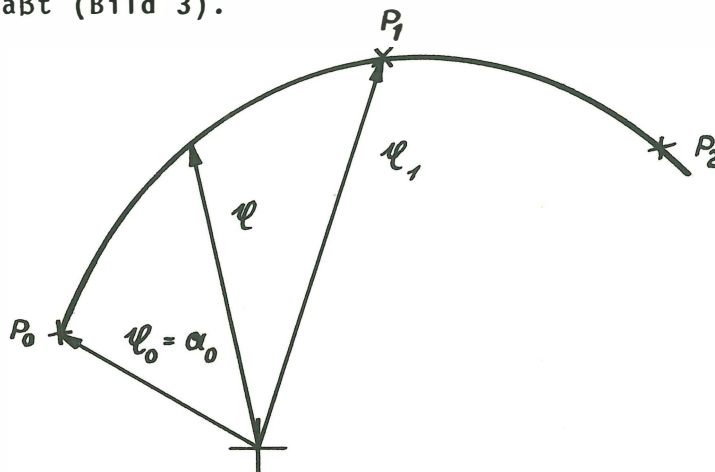


Bild 3:

⁺) Die Programme wurden zum Teil von Studenten erstellt.

Für $t=0$ ist $\varphi = \varphi_0$, für $t=1$ ist $\varphi = \varphi_1$. Mit den Punkten P_0, P_1, P_2 und der Bedingung gleicher Tangenten in P_0 lassen sich für den Bereich P_0-P_1 die Koeffizienten α_0, b_0, c_0, d_0 bestimmen.

Letztlich wird der Kurvenzug durch Geradenstücke angenähert. Dabei werden über den Parameter t Stützstellen gerade so ausgewählt, daß eine Abweichung ε nicht überschritten wird. Das bedeutet, daß im Bereich geringer Krümmung nur wenige, im Bereich starker Krümmung dagegen mehr Stützstellen berechnet werden (Bild 4).

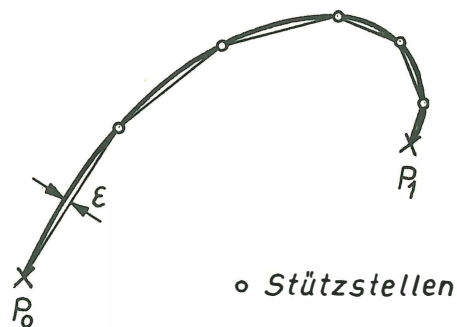


Bild 4:

Ein weiter erwähnenswertes Unterprogramm ist das zum Beschriften von Diagrammen. Die Schriftzeichen sind der DIN-Norm entsprechend festgelegt. Die charakteristischen Koordinaten jedes Schriftzeichens sind abgespeichert. Sie werden krumm- oder geradlinig verbunden. Beliebige Schriftgrößen sind möglich, die Schreibrichtungen entsprechen den 4 Schrittrichtungen des Plotters.

Da die meisten Benutzer ihre Probleme in ALGOL programmieren, sind die Zeichenunterprogramme in der Form von Code-Prozeduren abgefaßt. Dem Anhang ist eine genauere Ausführung der Prozeduren und der erforderliche Speicherbedarf zu entnehmen. Das anschließende Beispiel verwendet sämtliche Plotteraufrufe.

2.3 Code - Prozeduren

Code-Prozeduren sind Prozeduren, deren Rumpf bereits in übersetzter Form vorliegt. Im Algol-Programm erscheint anstelle des Prozedurrumpfes das Wort 'CODE'.

Z.B.:

```
'PROCEDURE' SINH (A);
```

```
'VALUE' A; 'REAL' A; 'CODE';
```

Während der Übersetzung des Algol-Programmes erfolgt die Meldung:

GIB CODE:

Sie ist zu beantworten mit

CODELS;

Der Code-Lochstreifen wird in Schalterstellung 1 eingelesen.

Enthält das Programm mehrere Code-Prozeduren, so wiederholt sich die Meldung entsprechend oft. Die Reihenfolge, in der die Code-Prozeduren eingelesen werden, ist belanglos.

2.3.1. Plotter - Prozedur

Die Plotter-Prozedur PLOT hat 3 Parameter:

```
'PROCEDURE' PLOT (K, X, Y);
```

Es dürfen für K nur Variable, Konstante oder Ausdrücke vom Typ 'INTEGER' eingesetzt werden.

Für $0 \leq K \leq 11$ sind X und Y in der Prozedur als 'VALUE' X, Y; 'REAL' X, Y; spezifiziert.

Für $12 \leq K \leq 15$ und $20 \leq K \leq 29$, $K \neq 25$ sind X und Y als 'VALUE' X, Y; 'INTEGER' X, Y; spezifiziert.

Für $K = 25$ ist X als 'VALUE' X; 'INTEGER' X; , Y als 'STRING' Y; spezifiziert.

Im ALGOL-Programm genügt folgende Vereinbarung:

'PROCEDURE' PLOT;

'CODE';

Abhängig vom Parameter K führt die Prozedur folgende Funktionen aus:

PLOT (0,X,Y); Festlegung der Maßstabsfaktoren.

Der Zahlenwert 1.0 wird in X bzw. Y mm dargestellt.

PLOT (1,0,0); Beginn des Plotterbetriebes. Dieser Aufruf ist auf jeden Fall vor Zeichenbeginn zu geben.

Das Programm stoppt und meldet sich mit:

PLOTTER: UM-AUTO-PLON:

Nach dem Umschalten von Lochstreifenstanzer auf Plotter, Einstellen auf Schalterstellung 2, Festlegen des Nullpunktes und Einschalten der Automatik wird mit

PLON;

quittiert. Hierauf wird weitergerechnet.

Eine Quittierung mit

PEND;

beendet das ALGOL-Programm.

PLOT(2,X,Y); Geradliniges Fahren zum Kurvenpunkt (X,Y).

PLOT (3,X,Y); Krummliniges Fahren zum Kurvenpunkt (X,Y).

Der gleiche Kurvenpunkt sollte nicht direkt nacheinander aufgerufen werden.

PLOT (4,0,0); Krummliniges Fahren zum letzten Kurvenpunkt.

Dieser Aufruf muß unbedingt am Ende des krummlinigen Zeichnens gegeben werden. Dieser Aufruf darf erst erfolgen, wenn vorher mindestens zwei Aufrufe PLOT (3,X,Y) gegeben worden sind.

PLOT (5,0,0);	Zeichenstift heben.
PLOT (6,0,0);	Zeichenstift senken.
PLOT (7,0,0);	Dieser Aufruf bewirkt die Ausgabe der Pufferbereiche für die Plotterbefehle. Der Zeichenstift wird gehoben. Der Aufruf sollte am Ende des Programms stehen.
PLOT (8,0,0);	Zwischenstop. Die Pufferbereiche werden ausgegeben. Der Zeichenstift bleibt gehoben. Das Programm stoppt und meldet sich mit STOP: PLON: Nach Quittierung mit PLON; wird weitergerechnet. Eine Quittierung mit PEND; beendet das ALGOL-Programm.
PLOT (9,X,Y);	Änderung des Koordinatensystems. Dem Standort des Zeichenstiftes werden die Koordinaten (X,Y) zugeordnet.
PLOT (10,X,Y);	Zeichnen von Markierungsquadraten. Der Stift wird gehoben, fährt zum Punkt X,Y und zeichnet um den Punkt ein Quadrat von 1 mm Seitenlänge. Der Stift bleibt gehoben.
PLOT (11,X,Y);	Zeichnen von Markierungskreuzen. Entspricht PLOT (10,X,Y);
PLOT (12,X,Y);	Zeichnen von gestrichelpunktierten Linien für die Aufrufe K = 2,3,(4). Die Anzahl der Striche wird durch X bestimmt ($ X \geq 1$), die Anzahl der Punkte durch Y . Bei X = 0 wird eine ununterbrochene Linie gezeichnet. Der Zeichenstift wird gesenkt. Während des Strich-Punktierten Zeichnens behalten die Aufrufe Zeichenstift-Heben und - Senken ihre Wirkung.

PLOT (13,o,o);	Beenden des Strich-Punktiierten Zeichnens. Der Zeichenstift wird gehoben.
PLOT (14,X,o);	Zeichnen eines Rechtecks im Format DIN A 4. Das Rechteck wird in +x und +y-Richtung gezeichnet, d.h. Bezugspunkt ist die linke untere Ecke. X = 1 Längsformat, X = 0 Hochformat. Senken und Heben des Zeichenstiftes erfolgt selbsttätig.
PLOT (15,X,Y);	Zeichnen eines Pfeiles von 6 mm Länge. Bezugspunkt ist die Pfeilspitze. Senken und Heben des Zeichenstiftes erfolgt selbsttätig. Angabe der Pfeilrichtung: X = 1; Y = 0: X-Richtung X = -1; Y = 0: -X-Richtung X = 0; Y = 1: Y-Richtung X = 0; Y = -1: -Y-Richtung
PLOT (2o,X,Y);	Bestimmung der Normschrift. X = 0: Gerade Normschrift X = 1: Schräge Normschrift Y = 0: Normalschrift (Mittelschrift) Y = 1: Engschrift Y = 2: Breitschrift
PLOT (21,X,Y);	Angabe der Schreibrichtung: X = 1; Y = 0: X-Richtung X = -1; Y = 0: -X-Richtung X = 0; Y = 1: Y-Richtung X = 0; Y = -1: -Y-Richtung
PLOT (22,X,0);	Festlegung der Schrifthöhe. X gibt die Höhe der Zeichen (Großbuchstaben) in mm an.

Für die folgenden Aufrufe sind die Aufrufe K = 20,21,22 Voraussetzung. Fehlende Aufrufe werden durch die Meldung "Aufruf K fehlt" ausgeschrieben. Es erfolgt dann eine willkürliche Festlegung auf: Gerade Normschrift, bzw. X-Richtung, bzw. 10 mm.

- PLOT (23,0,0); Einstellen einer neuen Zeile.
Der neue Zeilenanfangspunkt liegt senkrecht unter dem Anfangspunkt der letzten Eingabe.
(Zeilenabstand: 1.7 x Schrifthöhe)
- PLOT (24,X,0); Einstellen einer neuen Zeilenhöhe. Es wird ein Zeilensprung von $X/10$ x Schrifthöhe ausgeführt.
- PLOT (25,X,"Y"); Eingabe der zu zeichnenden Schriftzeichen.
Dabei haben die Zeichen [,] , 10 folgende Bedeutung:
[- Umschaltung von Klein- auf Großbuchstaben
] - Umschaltung von Groß- auf Kleinbuchstaben
 10 - Zwischenraum (entspricht einer Buchstabenbreite)
- Einschränkung: Die Zeichen [,] , 10 können nicht mitgezeichnet werden.
X = 0 : Zeichnen der Zeichenfolge "Y" normal.
X = 1 : Zeichnen der Zeichenfolge "Y" als Indices.
- PLOT (26,X,Y); Zeichnen des Wertes einer Variablen.
Der Wert der Zahl Y ('INTEGER') wird gezeichnet.
X = 0 : Zeichnen von Y normal.
X = 1 : Zeichnen von Y als Index.
- PLOT (27,0,0); Die Zeichenfolge der letzten Eingabe wird einmal unterstrichen.
Dieser Aufruf darf nur nach einem vorherigen Aufruf PLOT (25,X,"Y"); erfolgen.
Standpunkt nach dem Aufruf: Anfang der Zeichenfolge auf der Höhe des Striches.

PLOT (28,0,0); Die Koordinaten des Standortes der Feder werden zwischengespeichert. (Anwendung beim Zeichnen eines Bruchstriches, s. PLOT (29,0,0);).

PLOT (29,0,0); In der Höhe des durch den Aufruf PLOT (28,0,0); zwischengespeicherten Koordinatenpunktes wird vom Standort der Feder bis zum gemerkten Punkte eine Gerade gezogen (Anwendung beim Zeichnen eines Bruchstriches).

Die Plotter-Prozedur liegt folgenden Versionen vor:

1. Code Plot

Alle Aufrufe

Diese Prozedur umfaßt alle beschriebenen Aufrufe. Länge: 3006 FW

2. Code Plot

Krummlinig

Diese Prozedur umfaßt die Aufrufe K = 0 bis 15
Aufrufe K > 15 werden übersprungen.
Länge: 1327 FW

3. Code Plot

Geradlinige Schrift

Die Aufrufe K = 3,4 fehlen.
Die Aufrufe für Schriftzeichen werden geradlinig ausgeführt.
Aufrufe K = 3,4 werden wie K = 2 interpretiert. Beim ersten Aufruf K = 3 erfolgt die Meldung K = 3,4 NV.
Länge: 2350 FW

4. Code Plot

Geradlinig.

Wie 3., jedoch ohne Schriftzeichen.
Länge: 661 FW

5. Code Plot

Mini Plot

Diese Prozedur enthält nur die
Aufrufe K = 0,1,2,5,6,7.
Aufrufe mit anderen K-Werten
führen zur Zerstörung des Programms.
Es erfolgt beim Aufruf PLOT(1,0,0);
keine Blattschreibermeldung; d.h. der
Plotter muß vor dem Programmstart
zeichenbereit sein.
Länge: 277 FW

Etwa 10 bis 15 Minuten vor Zeichenbeginn sollte der Plotter
eingeschaltet werden, damit das Getriebeöl der Schrittmotoren
auf Betriebstemperatur erwärmt wird.

'BEGIN'

'COMMENT' PLOTTERBEISPIEL 15.9.77;

'REAL' X;

'INTEGER' I;

'PROCEDURE' PLOT;

'CODE';

START:

PLOT(1,0,0); PLOT(22,4,0); PLOT(21,1,0);
 'COMMENT' HIER FEHLT AUFRUF PLOT(20,X,Y), NOETIG FUER K=25,27,23;
 PLOT(25,0,'[R]EISPIEL:'); PLOT(27,0,0);
 PLOT(23,0,0); PLOT(28,0,0); PLOT(25,0,'[G]ERADE,[M]ITTELSCHRIFT');
 PLOT(23,0,0); PLOT(20,1,0); PLOT(25,0,'[S]CHRAEGE,[M]ITTELSCHRIFT');
 PLOT(23,0,0); PLOT(20,0,1); PLOT(25,0,'[G]ERADE,[E]NGSCHRIFT');
 PLOT(23,0,0); PLOT(20,1,1); PLOT(25,0,'[S]CHRAEGE,[E]NGSCHRIFT');
 PLOT(23,0,0); PLOT(20,0,2); PLOT(25,0,'[G]ERADE,[R]EITSCHRIFT');
 PLOT(23,0,0); PLOT(20,1,2); PLOT(25,0,'[S]CHRAEGE,[R]EITSCHRIFT');
 PLOT(23,0,0); PLOT(0,0,0); PLOT(20,1,0); PLOT(22,6,0); PLOT(23,0,0);

PLOT(25,0,'[A]BCDEFGHIJKLMNOPQRSTUVWXYZ'); PLOT(23,0,0);
 PLOT(25,0,'[a]bcdefghijklmnopqrstuvwxyz'); PLOT(23,0,0);
 PLOT(25,0,'[01234567890](*~/',.-:;'); PLOT(23,0,0); PLOT(23,0,0);

PLOT(25,0,'[f]'); PLOT(25,1,'[r]'); PLOT(25,0,'[=]'); PLOT(24,3,0); PLOT(28,0,0); PLOT(24,4,0);
 PLOT(25,0,'[v]'); PLOT(25,1,'[r]'); PLOT(25,0,'[f]'); PLOT(29,0,0);
 PLOT(24,-12,0); PLOT(25,0,'[t]'); PLOT(25,1,'[1]'); PLOT(25,0,'[p+1]'); PLOT(25,1,'[2]');
 PLOT(25,1,'[2]'); PLOT(25,0,'[p+1]'); PLOT(0,100,100); PLOT(0,0.3,1.2);

'FOR' I:=0 'STEP' 1 'UNTIL' 10 'DO' PLOT(10,0,1-0.1);
 PLOT(6,0,0); PLOT(2,0.25,1); PLOT(15,1,0); PLOT(0,-0.25,0); PLOT(12,1,2);
 'FOR' X:=0.5 'STEP' 0.5 'UNTIL' 3.1 'DO' PLOT(3,-COS(X)/4,-SIN(X)/4);
 PLOT(3,0.25,0); PLOT(4,0,0); PLOT(13,0,0); PLOT(0,0.75,1); PLOT(15,-1,0); PLOT(6,0,0); PLOT(2,1,1);
 'FOR' I:=10 'STEP' -1 'UNTIL' 0 'DO' PLOT(11,1,1-0.1);
 PLOT(6,0,0); PLOT(2,0,0); PLOT(5,0,0); PLOT(22,3,0);
 'FOR' I:=0 'STEP' 1 'UNTIL' 10 'DO'

'BEGIN'

PLOT(2,-0.1,1-0.1); PLOT(26,0,-1);

'END';

PLOT(2,0.5,0.4); PLOT(0,0,0);

PLOT(21,1,0); PLOT(25,0,'[R]ICHTUNG,[X]'); PLOT(2,0,0);
 PLOT(21,0,1); PLOT(25,0,'[R]ICHTUNG,[Y]'); PLOT(2,0,0);
 PLOT(21,-1,0); PLOT(25,0,'[R]ICHTUNG,-[X]'); PLOT(2,0,0);
 PLOT(21,0,-1); PLOT(25,0,'[R]ICHTUNG,-[Y]'); PLOT(2,-0.5,-0.5); PLOT(0,0,0);

PLOT(12,-3,3); PLOT(2,0.4,0); PLOT(5,0,0); PLOT(2,0.7,0); PLOT(6,0,0);
 PLOT(2,1,0); PLOT(13,0,0); PLOT(2,-0.7,-0.2); PLOT(14,0,0);
 PLOT(7,0,0);

'END'

Beispiel:

Gerade Mittelschrift

Schraege Mittelschrift

Gerade Engschrift

Schraege Engschrift

Gerade Breitschrift

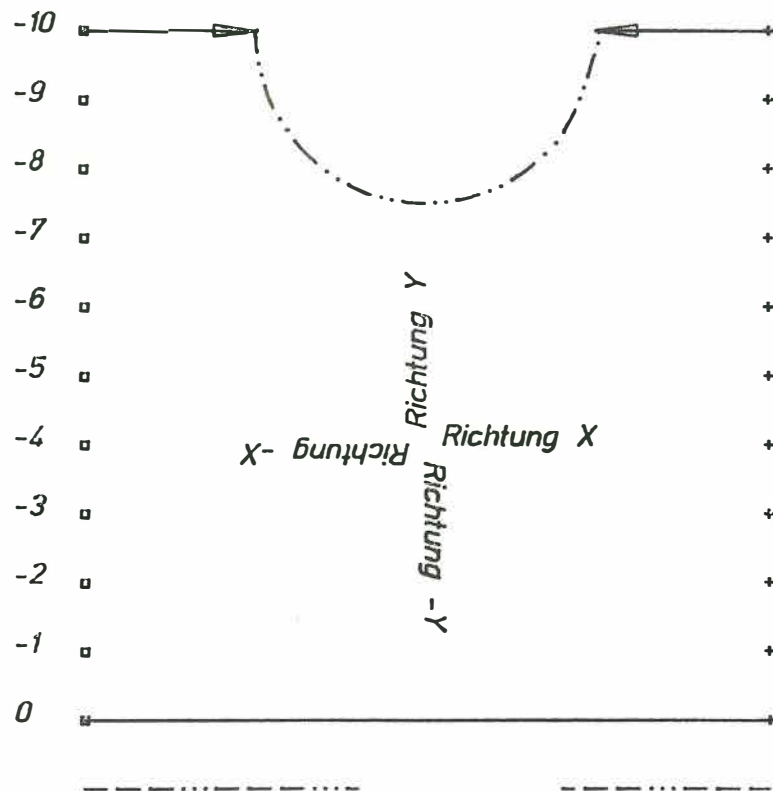
Schraege Breitschrift

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

1234567890()+x/','-:=;

$$F_R = \frac{V_R}{(T_1 p + 1) \times (T_2 p + 1)}$$



Walter Tenten

ZEL/NE der KFA Jülich

"PAL - Assembler 3o5/8"

PAL-Assembler 305/8

W.Tenten, KFA ZEL/NE

Zusammenfassung:

Der nachfolgende Bericht befaßt sich mit dem Assemblieren von PDP-8-Programmen auf einem Rechner Siemens 305. Es wurde ein Assembler geschaffen, der es ermöglicht, PDP-8-Programme auf einem anderen Rechner eines anderen Fabrikats zu assemblieren. Dieser Fremd-Assembler ist kompatibel zu dem PDP-8-eigenen PAL-Assembler in dem Sinne, daß er beim Schreiben des Quell-Programms den gleichen Formalismus voraussetzt und aus diesem Quell-Programm einen Binär-Lochstreifen erzeugt, der mit dem Standard-Ladeprogramm der PDP-8, dem Binary Loader, in den Kleinrechner eingelesen werden kann.

In der Kernforschungsanlage Jülich sind gegen Mitte des Jahres 1970 etwa ein Dutzend Kleinrechner vom Typ PDP-8 installiert. Diese Kleinrechner haben fast alle nur eine Teletype als Standard-Peripherie, weil für Datenerfassungs- und Experiment-Steuer-Aufgaben keine andere Standard-Peripherie benötigt wird. Insbesondere sind DEC-Tapes, Platten und schnelle Lochstreifengeräte sehr selten, wo Lochkartengeräte und Schnelldrucker gar nicht vorhanden sind.

Auf der anderen Seite befindet sich im Zentrallabor für Elektronik der Kernforschungsanlage ein Prozessrechner vom Typ Siemens 305. Dieser Prozessrechner verfügt über einen leistungsfähigen Schnelldrucker, eine mittelschnelle Lochkarteneingabe und eine schnelle Lochstreifenausgabe, und erfüllt damit fast alle Anforderungen, die man an einen Rechner für schnelle Assemblierungen stellen kann. Mit dieser Maschinen-Konfiguration ergeben sich nun folgende Vorteile für das Assemblieren von PDP-8-Programmen. Durch die Verwendung von Lochkarten als Träger des Quell-Programms wird eine hohe Einlese-Geschwindigkeit erzielt. Auf Lochkarten ist ein eventuell vorgeschriebenes festes Format leichter einzuhalten als auf Lochstreifen.

Handelsübliche Lochkartenstanzer für Handbedienung haben fast immer eine Tabulations-Einrichtung, während die für die Erstellung von Quell-Lochstreifen benutzten Teletypes keinen Hardware-Tabulator besitzen. Ist kein festes Format vorgeschrieben, ist der Programmierer dennoch dazu geneigt, ein solches zu benutzen, da auf diese Weise die Lesbarkeit des Quell-Programms erhöht wird. Die immer notwendigen Änderungen und Ergänzungen im Programm sind leichter als auf einem Lochstreifen durchgeführt. Dieser Tatbestand fällt besonders bei Gelegenheits-Programmierern ins Gewicht: viele Experimentatoren lassen es sich nicht nehmen, ihr Experiment-Programm selber zu erstellen. Auf einer Lochkarte steht üblicherweise nicht mehr als ein Befehl, wodurch etwa ein Viertel der Karte ausgenutzt ist. Der verbleibende Platz kann großzügig für Kommentare benutzt werden. Dagegen ist man beim Programmieren mit Lochstreifen immer bestrebt, den Streifen möglichst kurz zu halten.

Die Verwendung eines schnellen Lochstreifenstanzers für die Ausgabe des Objekt-Programms (Binär-Streifen) hat außer einer höheren Geschwindigkeit eine höhere Zuverlässigkeit zur Folge, weil der gestanzte Streifen in einem Arbeitsgang kontrollgelesen wird.

Die Ausgabe eines Assemblierungs-Protokolls auf einem Schnelldrucker ergibt, wie der Name schon sagt, eine höhere Druckgeschwindigkeit. Das Protokoll kann ausführlicher gestaltet werden, ohne daß die Druckgeschwindigkeit im entsprechenden Maße sinkt. Durch den Einsatz von getrennten Ausgabe-Medien für das Protokoll und den Binär-Streifen entfällt die Notwendigkeit eines dritten Passes beim Assemblieren. Schließlich ergeben sich noch Vorteile dadurch, daß durch Verwendung eines anderen Rechners der On-Line-Betrieb des Kleinrechners ungestört weiterlaufen kann.

Der vorliegende Fremd-Assembler ist in der Assembler-sprache PROSA geschrieben und in einem Kernspeicherbereich von weniger als 2000 Worten ablauffähig. Er heißt zur Unterscheidung von dem PDP-Assembler PAL 305/8.

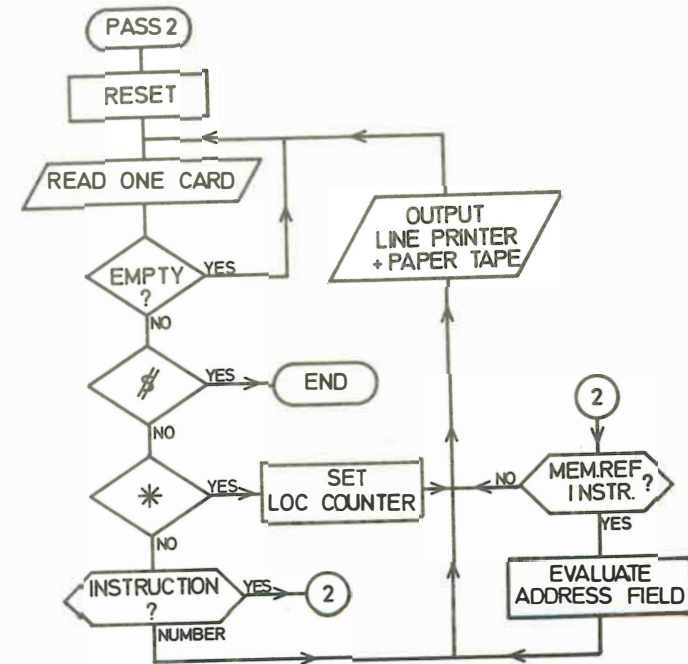
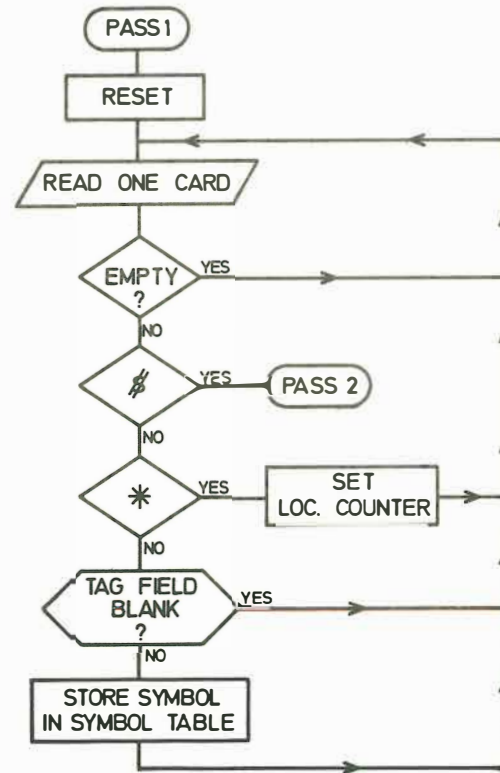
Die hervorstechendste Eigenschaft ist die hohe Geschwindigkeit. Bei vorsichtiger Abschätzung ergibt sich ein Geschwindigkeitsfaktor von mehr als einer Größenordnung. Bei dieser Abschätzung sind nur die reinen Ein-Ausgabe-Zeiten herangezogen worden, da die eigentliche Rechenzeit entweder vernachlässigbar ist oder durch Simultanarbeit nicht wirksam wird.

Beim formalen und logischen Aufbau wurde von zwei Vorbildern ausgegangen. Das erste ist der Siemens-PROSA-Assembler; ihm wurde das Format der Lochkarten und das Schema der Bedienung während des Ablaufs nachempfunden. Als zweiter Anhaltspunkt diente ein in FORTRAN geschriebener PAL-Assembler.

Die bisherigen Betriebserfahrungen mit dem PAL-305/8-Assembler sind als gut zu bezeichnen. Das Programm wird in zunehmendem Maße von PDP-8-Benutzern verwendet. Dabei werden nicht nur neue Programme geschrieben, sondern es kommt vor, daß bereits seit längerer Zeit vorhandene umfangreiche Datenerfassungs- und Experimentsteuerprogramme neu auf Lochkarten getippt werden, um immer wieder hinausgeschobene Änderungen endlich berücksichtigen zu können. Zum Abschluß sei die Bemerkung gestattet, daß es sich auf lange Sicht gesehen lohnt, einen Fremd-Assembler zu erstellen, da selbst unter der Voraussetzung, daß nur ein paar Kleinrechner vorhanden sind, die die Vorteile des Fremd-Assemblers ausnützen, die Arbeitersparnis beim Programmieren und Assemblieren den Zeitaufwand für die Herstellung des Fremd-Assemblers mehr als aufwiegt.

SPEED COMPARISON

	PAL III (TTY)	PAL 305/8	FACTOR
LOAD PROGRAM	5min	35sec PAPER TAPE 10 sec CARDS	9 30
PASS 1 PASS 2 INPUT	1sec/ 10char.	90msec/ card	11
PASS 2 BIN OUTPUT	200msec/ 2char.	20msec/ 2char.	10
PASS 3 LISTING	2.5sec/ 25char.	80msec/ Line	31
LISTING + BIN OUTPUT	27sec	80msec	34



PAL - ASSEMBLER 305 / 8

Bedienungsanweisungen

Der Assembler meldet sich nach dem Bereitstellen und Starten mit folgendem Text:

```
PAL-ASSEMBLER 305/8
PROGRAMMLAENGE XXXX ZELLEN
GIB LISTENLAENGE (LILE) .
```

Danach ist zu bedienen mit

```
LILE YYY; (Listenlänge) ,
```

wobei YYY eine mehrstellige Dezimalzahl (<4096) ist, die den für den Aufbau des Adreßbuches zur Verfügung stehenden Platz angibt. Der tatsächlich benötigte Platz ist drei Zellen pro Symbol.

Der Assembler antwortet mit

```
PROGRAMMENDE IN ZELLE ZZZZZ .
```

Danach kann die Listenlänge durch erneute Eingabe von LILE noch korrigiert werden.

Vor Beginn des ersten Passes, zwischen Ende des ersten und Beginn des zweiten Passes, sowie vor Beginn eines zusätzlichen (zweiten) Passes können folgende Bedienungen gemacht werden:

```
PLMP; (PAL mit Protokoll)
PLOP; (PAL ohne Protokoll)
PLMS; (PAL mit Streifen-Ausgabe)
PLOS; (PAL ohne Streifen-Ausgabe) .
```

Der Anfangszustand des Assemblers ist PLMP und PLMS.

Für den Start des ersten Passes muß mit

```
PST1; (PAL Start Paß 1)
```

bedient werden; entsprechend werden der zweite und alle zusätzlichen Pässe gestartet durch

```
PST2; (PAL Start Paß 2) .
```

Die Ausgabe des Adreßbuches wird veranlaßt durch

```
PLAB; (PAL Adreßbuch ausgeben) .
```

Der Assembler muß, bevor er durch ein anderes Programm überschrieben wird, beendet werden mit

```
PLEX; (PAL Exit) .
```

K.D. Reinartz
Inst. f. Kristallographie
Technische Hochschule
Aachen

"Syntax von ROSA, Programm mit
symbolischen Adressen.
(PROSA-Untermenge)"

Syntax von ROSA, Programm mit symbolischen Adressen
(PROSA - Untermenge)

K.D. Reinartz

Institut für Kristallographie, Techn. Hochschule
Aachen

```
< Programm > := < Vorspann > < Datenfolge >;
< Datenfolge > := < leer > / < Datenfolge > < Date > 'WR'
< Date >      := 'ZWR' < Kommentar > / < Datenadresse > < Wort >
1  < Kommentar > := < leer > / < Kommentar > < Kommentarzeichen >
.1  < Kommentarzeichen > := < Buchstabe > / < Ziffer > / < sonstige
                                   Zeichen >
...1 < Buchstabe > := A/B/..../Z
...2 < Ziffer > := < Oktalziffer > /8/9
...1 < Oktalziffer > := 0/1/2/ ... /7
...3 < sonstige Zeichen > := < Symbolsonderzeichen > / < allgemeine
                                   Zeichen > /= / ) /[
...1 < Symbolsonderzeichen > := +/ -/x/ :/ 'ZWR'/'Wer da'/' ]/_10/_./,/'/(
...2 < allgemeine Zeichen > := 'ZL'/'BU'/'ZI'/'ZWR'
< alle Zeichen > := < Buchstabe > / < Ziffer > / < sonstige Zei-
                                   chen > / 'WR' /;
2  < Datenadresse > := < leer > / ( < Symbol > )
.1  < Symbol > := < Buchstabe > / < Symbol > < Symbolzeichen >
...1 < Symbolzeichen > := < Buchstabe > / < Ziffer > / < Symbolsonder-
                                   zeichen >
3  < Wort > := < leer > / < OZ > / < DZ > / < GK > / < Befehl >
.1  < OZ > := OZ < Oktalzahl >
...1 < Oktalzahl > := < leer > / < Vorzeichen > < Oktalzahl ohne Vor-
                                   zeichen >
...1 < Vorzeichen > := < Plus > / -
....1 < Plus > := < leer > / +
...2 < Oktalzahl ohne Vorzeichen > := < Oktalziffer > / < Oktalzahl
                                   ohne Vorzeichen > < Oktalziffer >
.2  < DZ > := DZ < Dezimalzahl >
...1 < Dezimalzahl > := < leer > / < Vorzeichen > < Dezimalzahl ohne
                                   Vorzeichen >
...1 < Dezimalzahl ohne Vorzeichen > := < Dezimalziffer > / < Dezimal-
                                   zahl ohne Vorzeichen > < Dezimalziffer >
```

- .3 < GK > := GK < Gleitkommazahl >
- ..1 < Gleitkommazahl > := < leer > / < Vorzeichen > < Dezimalzahl ohne
Vorzeichen > < Exponententeil >
- ...1 < Exponententeil > := < leer > /₁₀ < Vorzeichen > < Dezimalzahl
ohne Vorzeichen >
- .4 < Befehl > := < Operationsteil > < Kennzeichen > < Operandenad-
se >
- ..1 < Operationsteil > := ADD/EA0/.../EA7/EPR/EVO/.../EV7/KPL/MLT/NOP/
ODR/SAM/SAP/SGN/SKU/SPR/STP/SUB/SUL/SUM/TAS/
TEL/TEM/TEP/UGL/UND/UNT/VAL/VAR/VDL/VDR/VLL/
VLR/VSE
- ..2 < Kennzeichen > := < Akkukennzeichen > < Substitution > < Unter-
brechbarkeit >
- ...1 < Akkukennzeichen > := < leer > / '
- ...2 < Substitution > := < leer > / :
- ...3 < Unterbrechbarkeit > := < leer > / .
- ..3 < Operandenadresse > := < leer > / < Oktalzahl ohne Vorzeichen > /
(< Unterprogrammnummer > < Platznummer >) /
< Datenadresse > < Oktalzahl > /x < Oktal-
zahl >
- ...1 < Unterprogrammnummer > := < Oktalzahl ohne Vorzeichen >
- ...2 < Platznummer > := < leer > / , < Oktalzahl ohne Vorzeichen >

Erläuterungen zum Assembler

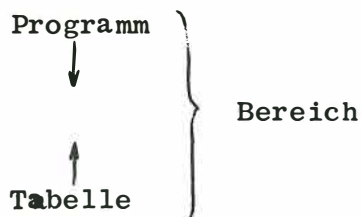
1. < Vorspann > := 'WR' 'ZL' < Nr > : 'WR' 'ZL'

DZ: < Anfangsadr. > - DZ : < Endadresse > - DZ :
< P-Nr >

< NR > ist die Nummer des assemblierenden Hauptprogramms

< Anfangsadr. > ,

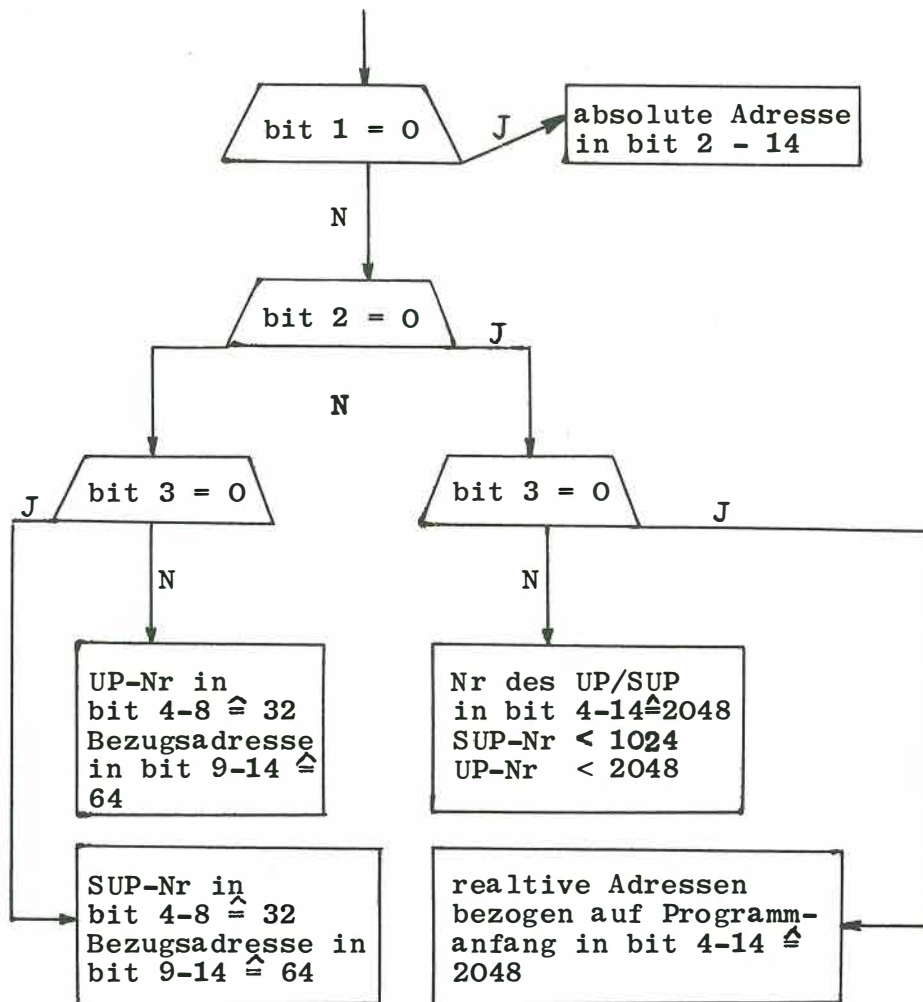
< Endadresse > geben den Bereich an, in dem das zu assemblierende Pro-
gramm einschliesslich seiner Symboltabelle aufgebaut
wird



< P-Nr > Hauptprogramm - Nr des zu assemblierenden Programms.
Ist das Programm ein UP/SUP, so ist die P-Nr = 0.

2. Programmaufbau

- 1) Die erste Zelle des Programms muss als erstes < Wort >
NOP < Nummer > enthalten
< Nummer > = 0 bei Hauptprogramm
≠ 0 bei UP/SUP
- 2) Die verschiedenen Operandenformen und ihre Darstellung im Maschinewort



- 3) Beim letzten relativen Befehl eines Programms ist die Markierungs-
stelle besetzt.
Dadurch entsteht ein voll relatives Programm, das über die

Programmnummern mit anderen UP/SUP verbunden ist. Das Organisationsprogramm baut beim Laden der einzelnen Programme die Nummern ab, so dass eine relative Form ohne Nummern-Verbindungen entsteht. Erst bei Start wird diese relative Form aufgehoben.

4) Die Eingabe und Ausgabe eines Zeichens erfolgt im Assembler durch je einen UNT-Aufruf.

5) Fehlermeldungen erfolgen on-line in der Form 'ZWR' F < Fehler-nummer > 'WR' 'ZL'

- F1 < Operationsteil > oder Zahltyp falsch
- F2 < Gleitkommazahl > enthält unzulässige Zeichen
- F3 < Befehl > " " "
- F4 < Symbol > beginnt nicht mit einem Buchstaben
- F5 Programm + Symbolliste werden zu land
(Ende des Programms)
- F6 < Symbol > mehrfach definiert
(erstes Symbol gilt)
- F8 Das Programm besitzt mehr als 512 Symbole
(weitere Befehle mit < Datenadresse > ≠ < leer > werden
überlesen)
- F9 Symbole nicht alle definiert
(Ende des Programms)
- F10 Dezimalzahl enthält mehr als 8 Ziffern

Normalerweise kann nach einer Fehlermeldung die fehlerhafte < Date > neu eingegeben werden, da sie nicht abgespeichert wird.

Das Irrungszeichen [lässt alle vorhergehenden Zeichen der < Date > übergehen.

6) Gleitkommazahlen werden in zwei aufeinanderfolgenden Worten dargestellt.

- 1. Wort Mantisse bit 1-24
- 2. Wort dualer Exponent bit 1-14

3. Der Assembler ist eingebettet in ein eigenes Betriebssystem für einen Kernspeicher mit maximal 8K Worten : Rechner z.B. 303 P.

Chr. Meixner
ZIR der KFA Jülich

"On-Line Auswertung von
Gammaspektren an Spülkreis-
läufen"

On-line Auswertung von Gammaspektren an Spülkreisläufen

Chr. Meixner

Zentralinstitut für Reaktorexperimente K F A Jülich

Zur schnellen Datenerfassung und Datenverarbeitung wird in dem Reaktor in Mol (Belgien) für 6 Spülkreisläufe ein Siemensrechner vom Typ 301 installiert werden. Die Spülkreisläufe gehören zu Bestrahlungsexperimenten von beschichteten Teilchen (coated particles), deren Verhalten unter Reaktorbedingungen untersucht wird. Insbesondere ist ihr Rückhaltevermögen für die radioaktiven Spaltgase Krypton und Xenon von Interesse.

Die beschichteten Teilchen werden während der Bestrahlung mit Helium oder einem Helium-Neon-Gemisch gespült. Die radioaktiven Gase, die durch Diffusion an die Oberfläche gelangen oder durch Bruch der Teilchen freigesetzt werden, werden von dem Spülgas mitgenommen. Das Gasgemisch wird mit einem Germaniumzähler gammaspektrometrisch untersucht und aus der Aktivität der einzelnen Komponenten das Rückhaltevermögen der Teilchen berechnet. Ein Maß für das Rückhaltevermögen ist das R/B-Verhältnis, das ist das Verhältnis von freigesetzter zu erzeugter Aktivität.

Abbildung 1 zeigt vereinfacht das Rohrleitungssystem des

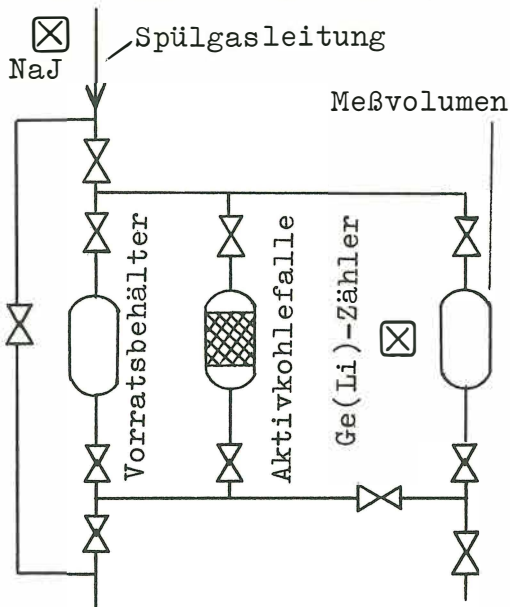


Abb. 1

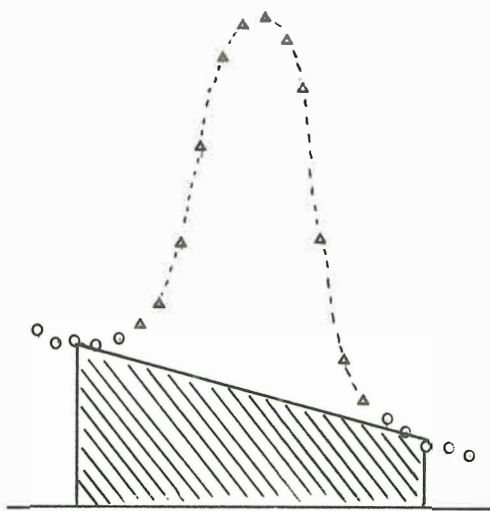
Spülkreislaufes im Bereich der Aktivitätsmeßstelle. Das Spülgas wird zuerst mit einem NaJ-Szintillationszähler monitiert, d.h. es wird rein qualitativ das zeitliche Verhalten der gesamten Spülgasaktivität überwacht. Ein der Aktivität proportionales Signal gibt ein Ratemeter an den Rechner. Das Signal wird benutzt, um den Ablauf der quantitativen Aktivitätsmessung zu steuern. Ein plötzliches Ansteigen der Aktivität deutet z.B. auf den Bruch

eines Teilchens hin und man erwartet in diesem Fall eine andere Zusammensetzung des Spaltgasgemisches. Abhängig von der Größe des Monitorsignals und einigen anderen Faktoren strömt das Gas durch drei verschiedene Zweige: einen Vorratsbehälter, ein Aktivkohlefilter oder direkt in das Meßvolumen vor dem Germaniumzähler. Aus dem Vorratsbehälter bzw. dem Aktivkohlebehälter wird das Gas dann rechnergesteuert nach fest vorgegebenen Abklingzeiten in das Meßvolumen überführt.

Die nukleare Elektronik für den Germaniumzähler ist konventioneller Art. Ein Teil des Rechners (4K) wird zur Speicherung des Spektrums verwendet. Das Interface zwischen ADC und Rechner ist eine Entwicklung des Zentrallabors für Elektronik der K F A Jülich.

Für die Auswertung der Spektren wird angenommen, daß die Lage der Photopeaks bereits bekannt ist, d.h. es wird im Programm fest angegeben, bei welchen Kanälen mit einem Peak zu rechnen ist. Diese Annahme ist bei dem vorliegenden Problem gerechtfertigt, da nur Krypton- und Xenonisotope und ihre Folgeprodukte als radioaktive Substanzen im Spülgas auftreten können. Aus dem gleichen Grunde kann man das Auftreten von Doppellinien ausschließen. Ein aufwendiges Peak-sucheprogramm ist daher nicht notwendig. Es muß allerdings bei jedem Spektrum geprüft werden, ob bei den angegebenen Kanalzahlen auch tatsächlich Photopeaks vorhanden sind. Das Auftreten einer Linie hängt nämlich sowohl von der ursprünglichen Zusammensetzung des Gasgemisches als auch von der Abklingzeit ab. Kurzlebige Isotope wird man nur finden, wenn das Gasgemisch direkt durch das Meßvolumen strömt, langlebige Isotope aber erst nach einer längeren Wartezeit, da deren Aktivität zu niedrig ist, um sie neben den kurzlebigen hochaktiven Isotopen zu messen. Die Prüfung wird nach dem von MARISCOTTI (Nuc.Instr.Meth.72 (1969) 125-142) angegebenen Verfahren durchgeführt. Dabei wird geprüft, ob der Wert der 2. Ableitung der Meßkurve an den angegebenen Stellen signifikant über ihrem statistischen Fehler liegt.

Zur Berechnung der 2. Ableitung werden die Kanalinhaltel lediglich mit Gewichtungsfaktoren multipliziert und addiert, das gleiche gilt für den statistischen Fehler der Ableitung. Die Gewichte ergeben sich aus der zweiten Ableitung einer Gaußkurve mit der gleichen Halbwertsbreite wie die zu prüfende Linie. Da die Linienbreite etwas variieren darf, ohne daß sich die Gewichtungsfaktoren stark ändern, wird angenommen, daß alle Linien eine Breite von 3 keV haben. Die Gewichte sind ebenfalls im Programm angegeben. Zur Bestimmung der Peakfläche werden die Kanalinhaltel von mindestens 21 Kanälen um die Peakmitte herum aufaddiert und ein linearer Untergrund abgezogen. Die Höhe des Untergrundes wird aus je 5 Kanälen an den auslaufenden Flanken des Peaks bestimmt. Der Inhalt dieser 5 Kanäle wird jeweils durch eine Parabel angenähert; der Wert der Parabel in der Mitte (siehe Abb. 2) wird als rechter bzw.



linker Schnittpunkt der Untergrundgeraden angenommen. Die Berechnung der Ausgleichsparabel wird wiederum durch eine gewichtete Addition der Kanalinhaltel vorgenommen. Zur Berechnung der Aktivität der verschiedenen Komponenten des Spaltgases müssen die Peakflächen zunächst auf eine Einheitsmeßzeit (1 sec) normiert werden, wobei die Totzeit des Systems zu berücksichtigen ist, und anschließend für unterschiedliche geometrische

Abb. 2

Ausbeute, Linienintensität und die Empfindlichkeitskurve des Germaniumzählers korrigiert werden. Da die letztgenannten Korrekturen teilweise für jede Linie verschieden sind und sich nicht auf einfache Weise berechnen lassen, wurden sie zu einem Faktor für jede Linie zusammengefaßt und fest im Rechner gespeichert. Das Ergebnis ist schließlich die Aktivität der freigesetzten Spaltgase.

Zur Bestimmung des R/B-Verhältnisses ist noch die Kenntnis der erzeugten Aktivität notwendig. Diese wird aus der Masse des Urans zu Beginn der Bestrahlungsperiode, dem Neutronenfluß, der Neutronenfluenz (das ist der bis zum Meßzeitpunkt integrierte Neutronenfluß), dem Spaltquerschnitt von U 235 und der Spaltausbeute bestimmt. Bis auf den Neutronenfluß bzw. der Fluenz sind alle Größen konstant und können fest im Programm angegeben werden. Der Neutronenfluß muß aus der jeweiligen Reaktorleistung bestimmt werden; ein der Reaktorleistung proportionales Signal wird von der Reaktorinstrumentierung an den Rechner gegeben und vom Meßwertverarbeitungsprogramm bzw. einem speziellen Unterprogramm aufgearbeitet.

Das Program ist in FORTRAN geschrieben. Es wurde bisher an der IBM 360/75 und an der SIEMENS 301 getestet und zeigte zufriedenstellende Ergebnisse. Einzelheiten über das gesamte Experiment sind in den internen Berichten ZIR 69-2, 69-1 zu finden.

Teuschler
Fa. Siemens AG
Karlsruhe

"Bericht über den neuen
FORTRAN-H-Compiler für
das System 300 "

V O R T R A G

gehalten am Mittwoch, den 1. 7. 1970
bei der Siemens-Prozeßrechner-Benutzer-Tagung
in Jülich

Thema: Bericht über den neuen FORTRAN-H-Compiler
für das System 300

Bevor ich über den neuen FORTRAN-Compiler berichte, möchte ich kurz die historische Entwicklung der FORTRAN-Compiler für das System 300 aufzeigen.

1. Historische Entwicklung

Im Jahre 1966 begann man mit der Erstellung eines ALGOL und eines FORTRAN-Compilers für die DVA 3003 bzw. 303. An diese Compiler wurden folgende Anforderungen gestellt:

Die Compiler müssen auf allen Rechnern des Systems 300 zu betreiben sein, wenn ein ausreichender Kernspeicherbereich zur Verfügung steht (4 bis 8 K Worte). Sie müssen mit einer minimalen Anlagenausstattung arbeiten können (z. B. LS-Ein-/Ausgabe).

Aus dieser Forderung heraus wählte man bei ALGOL den Sprachumfang ALGOL 60 IFIP-Subset und bei FORTRAN den E-Level, ohne Double-Precision und ohne Magnetband-Ein-/Ausgabe, allerdings mit adjustable Dimension.

1.1. Kernspeicherversion

Die 1. Stufe der Implementierung war die Erstellung einer Kernspeicherversion, d. h. zum Kompilieren wird kein Externspeicher benötigt. Damit man mit einem relativ kleinen Kernspeicherbereich auskommt, wurde der Compiler in 10 Pässe (Durchläufe) segmentiert, jeder Pass hat ca. 2 K Programm, der Rest des Speichers kann als Datenpuffer benutzt werden.

Die 10 Pässe führen folgende Teilaufgaben aus:

Pass 1 bis 6	Syntax-Analyse und Adressierung der Variablen
Pass 7	Makrogenerierung
Pass 8	Prosa-Ausgabe
Pass 9	Adressierung des MC
Pass 10	MC-Ausgabe

Die einzelnen Pässe wurden der Reihe nach von Lochkarten bzw. Lochstreifen bereitgestellt.

Der Compiler liefert vom Quellprogramm sogenannte Module, die mit den Standardfunktionen zu einem ablauffähigen Programm zusammengebunden werden. Diese Aufgabe übernimmt der Binder.

Der FORTRAN-Binder für die Kernspeicherversion bindet das Hauptprogramm und eventuell die dazugehörigen Unterprogramme (Subroutine, Function) zusammen mit den benötigten Standardfunktionen und setzt im Kernspeicher ein ablauffähiges Programm ab, das fest adressiert ist.

1.2. Externspeicherversion

Nachdem für das System 300 als Externspeicher Trommel und Platte zur Verfügung standen, wurde 1967/68 aus der Kernspeicherversion eine Externspeicherversion erstellt. Dabei wurde an dem Konzept der 10 Pässe nichts geändert. Eine Änderung der Bereitstellaufrufe hätte jedoch einen sehr langsamen Compiler ergeben, deshalb wurden die einzelnen Pässe festadressiert vom Externspeicher in den Kernspeicher transferiert. Dabei belegt der Compiler einen Laufbereich von maximal 4 K im Arbeitsspeicher.

Zur Ausgabe der Module auf LK und LS kam nur noch die Ausgabe aus PSK und TSK.

Die Externspeicherversion erhielt auch einen neuen Binder, der die gebundenen Programme in relativierter Form liefert.

Mit der Erstellung der Externspeicherversion stieg das Interesse am FORTRAN-Compiler und es wurden auch sehr bald Erweiterungs- und Änderungswünsche angemeldet.

Z. B.:

Adressierung der Integerwerte

1 Wert /Wort

Einfügung des logischen If

Apostroph im FORMAT an Stelle von nH...

READ/WRITE für magnetische Datenträger

Segmentierungsmöglichkeiten

Alle diese Wünsche konnten jedoch im alten Compiler nicht realisiert werden, da er nur für den E-Level ausgelegt ist und die kleinste Änderung praktisch zu einer Umprogrammierung in allen Pässen führte.

Deshalb erschien es sinnvoll, einen neuen FORTRAN-Compiler zu erstellen, mit dem Sprachumfang von Full-FORTRAN.

2. FORTTRAN-H-Level-Compiler

2.1. Sprachumfang /1/

Bei der Fixierung des Sprachumfanges des neuen FORTRAN Compilers für das System 300 ging man vom FORTRAN IV der DVA 4004 aus. Nun ist ja die 4004 ein Byt-Rechner, so daß für die Anpassung bestimmte Normierungen zu treffen sind.

Für die unterschiedlichen Datenformate gilt folgende Zuordnung:

2 Byte = 1 FW

INTEGER * 2	bedeutet	INTEGER 1 FW	
INTEGER * 4		INTEGER 2 FW	SL
REAL * 4		REAL 2 FW	SL
REAL * 8		REAL 4 FW	
COMPLEX * 8		COMPLEX 4 FW	S.L.
COMPLEX * 16		COMPLEX 8 FW	
LOGICAL * 4		LOGICAL 2 FW	SL
LOGICAL * 1		LOGICAL 1 FW	

Die mit S.L. gekennzeichneten Datenformate stellen die Standardlänge dar, d. h. bei INTEGER * 4 wird die Arithmetik über Unterprogramme mit 48 Bit Doppelwortarithmetik gelöst.

REAL * 4 wird in 2 Versionen zur Verfügung stehen:

- a) für die DVA 301 bis 305 mit 24 Bit Mantisse
- b) für die DVA 306 mit 34 Bit Mantisse

REAL * 8 wird zur Zeit mit 48 Bit Mantisse und 10 Bit Exponent realisiert, kann jedoch später bis auf 82 Bit Mantisse erweitert werden, da zur Speicherung 4 FW zur Verfügung stehen.

/1/ Beschreibung FORTRAN IV des SIEMENS
SYSTEMS 4004 (3. Ausgabe August 1970,
Bestell-Nr. D 14/4025)

Die Anmerkungen für REAL * 4 und * 8 gelten entsprechend auch für COMPLEX * 8 und * 16.

Die Zuordnung 2 Byt = 1 FW wird nur für LOGICAL * 1 durchbrochen, da der Adressierungsaufwand für halbe Festworte nicht sinnvoll ist.

Eine weitere Abweichung gegenüber FORTRAN IV der DVA 4004 ergibt sich aus der unterschiedlichen Zeichenstruktur.

Es kann nur der Zeichenvorrat des Systems 300 benutzt werden, d. h. \$ ist kein alphanumerisches Zeichen und & ist im System 300 nicht verfügbar.

2.2. Compiler

Der Compiler ist ein Externspeicher-Compiler und arbeitet ebenfalls mit mehreren Durchläufen.

Es benötigt als Minimalausstattung:

- 4 K freien Kernspeicherbereich
- 1 Bedienungsblattschreiber
- 1 Plattenspeicher bzw. Trommel mit min. 128 K LSE bzw. LKE zur Eingabe des Quellprogrammes

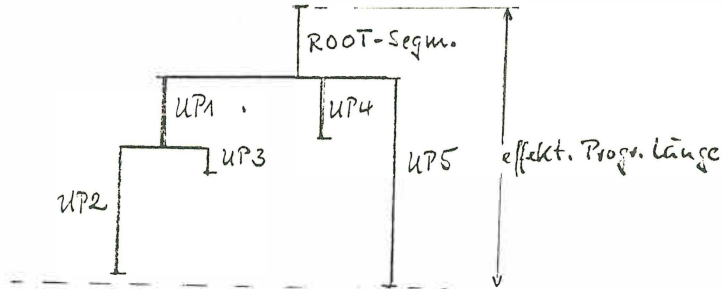
Der Compiler liest das Quellprogramm vom Externspeicher; es muß vorher über LS oder LK auf Externspeicher geladen werden.

Der Compiler erzeugt Module nur auf dem Externspeicher. Protokolle und Fehlermeldungen können über Schnelldrucker ausgegeben werden.

Bei einer Erweiterung des Laufbereiches, z. B. auf 6 bis 8 K je nach Programmgröße, kann die Namensliste im Speicher geführt werden. Dadurch ergibt sich eine höhere Compilergeschwindigkeit um den Faktor 2 bis 3.

2.3. Overlay-Binder

Mit dem Externspeicherbinder können Programme mit Overlay gebunden werden. Voneinander unabhängige Unterprogramme können so eingebunden werden, daß sie erst zum Zeitpunkt des Aufrufes vom Externspeicher in den Kernspeicher transferiert werden. Das setzt Programme voraus, die eine sogenannte Baumstruktur besitzen.



UP2 und UP3 kann vom Root-Segment von UP1 aufgerufen werden, UP1, UP4 und UP5 nur vom Root-Segment.

Mit dieser Technik lassen sich eine Reihe von Segmentierungswünschen lösen.

2.4. Adressierungsvolumen

Bei der DVA 301 bis 305 ist das Adressierungsvolumen auf 16 K begrenzt, bei der DVA 306 auf 32 K.

2.5. Termine

Anfang 1971 wird der Prototyp des Compilers zur Verfügung stehen, d. h. der Compilerkern, der Binder und die wichtigsten Standardfunktionen einschließlich komplexer Arithmetik. Spezielle Funktionen der Ein-/Ausgabe, wie READ, WRITE für Speicher mit wahlfreiem Zugriff können erst im Herbst 1971 implementiert werden.

Termin

Leonard Radermacher
ZEL/NE der KFA Jülich

Hardware - Rechnerkopplung

Siemens 305 - PDP 8

Nov. 1970

Hardware - Rechnerkopplung Siemens 305 - PDP 8

Es soll die hardware einer Rechnerkopplung des Prozeßrechners Siemens 305 mit mehreren Digital Equipment Kleinrechnern PDP-8 beschrieben werden. Ziel dieses Projektes ist es, Erfahrungen für ein on-line Datenverarbeitungsnetz in der Kernforschungsanlage Jülich zu gewinnen. Hier werden mehrere Experimente mit der PDP-8 betrieben. Einige dieser Experimentrechner sollen gekoppelt werden mit dem Prozeßrechner Siemens 305 im Zentrallabor für Elektronik.

Die Kopplung am Siemens Rechner soll über die langsame sog. Standardnahtstelle erfolgen. Über diese kann der Datentransfer teilwortweise 6 bit parallel maximal alle 6 μ s stattfinden. Es sollen auf Grund der 6 bit Teilwortstruktur an der Standardnahtstelle maximal sechs PDP-8 Rechner auf der Experimentseite angeschlossen werden. Diese können bis zu ca. zwei Kilometer vom Siemens Rechner entfernt sein. Das Interface an der Siemens - Nahtstelle soll - zusammen mit dem entsprechenden Steuerprogramm im Siemens Rechner - die Gesamtsteuerung der Rechnerkopplung übernehmen. Die zugehörige Koppel elektronik an der PDP muß so einfach wie möglich ausgelegt werden können, da sie ja für jeden PDP - Anschluß benötigt wird (Bild 1).

Beide Rechnertypen werden nicht gleichberechtigt gekoppelt, sondern hierarchisch. Die Initiative einer gewünschten Datenübertragung muß also immer von einer PDP-8 ausgehen. Die Siemens 305 soll nicht für ihre ureigenen Daten eine Ablage bei einer PDP suchen, sondern die PDP am Experiment entlasten. Im Siemens Rechner können die Daten dann nach bestimmten Kriterien ausgewertet und abgelegt oder zur PDP zurück übertragen werden.

Der Standardkanal der Siemens - Nahtstelle wird sozusagen aufgespalten, in sechs Teilkanäle für den Anschluß von bis zu sechs PDP-8 Rechnern oder evtl. auch Vielkanalanalysatoren. Diese Teilkanäle sollen allerdings nicht gleichzeitig innerhalb eines Übertragungsblocks arbeiten können, sondern nur nacheinander. Simultarbeit ist in einer weiteren Ausbaustufe möglich durch den Anschluß eines weiteren Interface an einen zweiten Siemens - Standardkanal. PDP-8 Anforderungen können aber zu jeder beliebigen Zeit beim Interface an der Siemens gestellt werden.

Ein Prioritätsnetzwerk speichert hier diese Requests ab. Vom Steuerprogramm werden diese dann zu geeigneter Zeit zugelassen. Die Prioritätssteuerung gibt dann die PDP - Anforderung frei, die zu diesem Zeitpunkt gerade die höchste Priorität hat.

Die PDP-Anforderung, die zugelassen wurde, überträgt nun ihren sog. Steuerblock in den Siemens Rechner und spezifiziert damit genau die Datenübertragung, die sie wünscht. Übrigens wird diese Steuerblockübertragung natürlich genau so behandelt wie eine Übertragung der wirklichen Experimentdaten. Im derzeitigen Konzept besteht dieser Steuerblock aus einem 24 bit - Wort. Es handelt sich also sozusagen um ein Steuerwort in Bezug auf die Siemens - Wortlänge. Das Steuerprogramm im Siemens Rechner wertet dieses Wort aus. Es enthält verschlüsselt die Übertragungsrichtung - also Eingabe oder Ausgabe aus dem Siemens Rechner -, zwei Dateinamen für die Daten und das Verarbeitungsprogramm, dann die Blocklänge der Übertragung und, ob eine sog. kurze oder lange Verarbeitung der Daten gewünscht wird. Je nach Inhalt des Steuerwortes kann eine Siemens Antwort auf diese PDP - Anforderung vom Steuerprogramm zeitlich im Anschluß an die zugehörige Steuerwortübertragung eingeleitet werden, oder die Beantwortung wird vom Steuerprogramm zurückgestellt, und es läßt eine neue PDP - Anforderung zu. Die Koordinierung der Zulassung von PDP-Anforderungen und deren Beantwortung führt also das Steuerprogramm durch. Eine Siemens Antwort auf eine PDP-Anforderung stellt dann die gewünschte Datenübertragung dar.

Der gesamte zu übertragende Datenblock wird in Teilblöcke aufgeteilt von z.B. 128 Siemens Worten, um bei einem evtl. in der Übertragung auftretenden Fehler nicht den gesamten Block wiederholen zu müssen. Besonders wichtig ist aber, daß eine PDP bei Auftreten eines Fehlers definiert zum Beginn eines Teilblocks zurückspringen kann.

Vor jeder Datenübertragung wird zur Auswahl des Rechnerkopplungselementes aus anderen externen Elementen und zu dessen Versorgung mit den nötigen Parametern eine sog. Eingabe- bzw. Ausgabebefehlsgruppe verwendet. Diese Befehlsgruppe ist nach einem festen Schema aufgebaut. In einem ersten Elementauswahlbefehl wird das Interface ausgewählt und ihm die Information

"folgende Eingabe oder Ausgabe" mitgeteilt. Dann folgen zwei Elementversorgungsbefehle, die das Rechnerkopplungselement mit der Anfangsadresse und der Blocklänge, des im folgenden zu übertragenden Datenblocks versorgen.

In Bild 2 ist dargestellt, wie mit Hilfe zweier weiterer Elementauswahlbefehle eine neue PDP - Anforderung oder eine Siemens Antwort auf eine schon dekodierte PDP-Anforderung im Rechnerkopplungselement zugelassen wird. Der Elementauswahlbefehl "Prüf" erkennt den Zustand der Steuerung und fragt ab, ob sie belegt ist oder nicht. Unter "belegt" soll verstanden werden, daß die Steuerung gerade eine Datenübertragung durchführt oder vorbereitet. Ist die Steuerung nicht belegt, so wird sie mit dem "Prüf"-Signal für eine Siemens Antwort an eine PDP belegt. Ein "Skip"-Signal meldet dann dem Steuerprogramm den Belegungszustand zurück. Dieser Skip kann dann per Programm alles weitere veranlassen. Es wird die schon erwähnte Eingabe- bzw. Ausgabebefehlsgruppe dem externen Element übergeben und der entsprechende Verarbeitungsablauf eingeleitet. Die Steuerung bleibt so lange für eine Siemens Antwort an eine PDP belegt, bis sie durch den Elementauswahlbefehl "Freigabe" zurückgenommen wird. Die Steuerung soll nun eine neue PDP-Anforderung zulassen. Im Prioritätsnetzwerk für die PDP - Requests wird gefragt: Ist eine PDP-Anforderung da? Wenn ja, so wird der entsprechende Verarbeitungsablauf der Steuerwort-Eingabe in den Rechner eingeleitet. Wenn nein, so ist die Steuerung sozusagen frei nach beiden Seiten, d. h. kommt jetzt eine PDP-Anforderung, so wird die zugehörige Steuerwort-Eingabe sofort eingeleitet, kommt aber der Elementauswahlbefehl "Prüf", so wird die Steuerung für die schon besprochene Siemens - Antwort an eine PDP belegt.

Bild 3 zeigt zunächst den Verarbeitungsablauf eines PDP-Requests mit der entsprechenden Steuerwort-Eingabe. Nach Zulassung der PDP durch das Steuerprogramm wird die Kenn-Nummer in eine vereinbarte Kernspeicherzelle des Rechners eingeschrieben. Dann wird ein spezielles sog. Programm-Unterbrechungsbit eingeschrieben zur Kennzeichnung des folgenden Interrupt. Das Steuerprogramm versorgt daraufhin das Rechnerkopplungselement mit der schon erwähnten Eingabebefehlsgruppe. Nach dieser Versorgung gibt das ex-

terne Element das Signal "Abweisend" zurück. Hiermit wird die Rechnerkopplungssteuerung gegen jede weitere Versorgung per Programm verriegelt. Der Verarbeitungsablauf einer "Siemens-Antwort" an eine PDP beginnt mit der Eingabe - bzw. Ausgabebefehlsgruppe und dem darauf folgenden "Abweisend". Dann wird die PDP-Kenn-Nummer aus der schon erwähnten vereinbarten Kernspeicherzelle ausgelesen. Die eigentliche Datenübertragung, die jetzt beginnt, läuft nach dem "shake-hand" - Prinzip ab, d. h. der ausgebende Rechner sendet mit den Daten ein sog. "Ready for Transfer"-Signal. Die Übernahme der Daten wird vom empfangenden Rechner mit einem Quittingssignal "Transfer-Accepted" bestätigt. Bei einer Dateneingabe in den Siemens - Rechner wird dessen Eingabebereitschaft mit dem sog. "Select"-Signal der PDP mitgeteilt. Hiermit wird die PDP aus anderen ausgewählt und kann mit dem "Ready"-Signal das erste 6 bit Teilwort dem Siemens Rechner übergeben. Danach wird im Rechnerkopplungselement die Datenadresse um ein Teilwort höhergeschaltet, der Blocklängenzähler um eins heruntergezählt und ein "Transfer-Accepted"-Signal der PDP zurückgegeben. Bei einer Ausgabe aus dem Siemens Rechner wird die Siemens - Ausgabebereitschaft mit dem "Ready"-Signal und dem ersten Datenteilwort der PDP direkt mitgeteilt. Ein "Transfer-Accepted" von der PDP zählt im Rechnerkopplungselement den Adresszähler um eins höher, den Blocklängenzähler um eins herunter. Erreicht der Blocklängenzähler Null, so wird ein sog. "Normaler Befehlsabschluß" eingeleitet. Es wird ein spezielles Programmunterbrechungsbit zur Kennzeichnung des folgenden Interrupt eingeschrieben und der beschriebene "Abweisend"-Zustand zurückgenommen. Tritt bei der Datenübertragung ein Fehler auf, so wird ein sog. "vorzeitiger Befehlsabschluß" eingeleitet, dh. es wird zusätzlich in einer festen Kernspeicherzelle (Anzeigenzelle) eine Anzeige eingeschrieben, die den Fehler näher kennzeichnet. Außerdem wird in diese Anzeigenzelle die Restadresse (Stand-des Adresszählers bei Auftreten des Fehlers) eingeschrieben. Fehler, die zu einem vorzeitigen Befehlsabschluß und damit zum Abbruch der Datenübertragung führen, sind erstens Zeitfehler, dh. nach einem "Ready"-Signal kommt nach einer festen Zeit kein "Transfer-Accepted", oder nach einem "Select" kommt kein "Ready"-Signal. Zweitens, wenn das externe Element in den geschützten Arbeitsspeicherbereich einzuschreiben versucht, wird eben-

falls die Datenübertragung sofort abgebrochen; drittens bei erreichter Blocklänge Null und einem danach noch erscheinenden "Ready", und viertens, wenn bei der Datenübertragung ein Parity-Fehler auftritt, wird eine Übertragung sofort beendet.

In Bild 4 sind sämtliche Daten- und Steuerleitungen (14 verdrehte Telefonkabel) dargestellt, die zum Anschluß einer PDP benötigt werden. Sechs Daten- und zwei Parityleitungen können in der Übertragungsrichtung umgeschaltet werden. Auf der "Request"-Leitung wird die PDP-Anforderung dem Prioritätsnetzwerk im Interface am Siemens Rechner mitgeteilt. Die "Direction"-Leitung teilt der PDP-Nahtstelle die Richtung der folgenden Übertragung mit. Das "Select"-Signal wählt eine der sechs PDP-Rechner für die Übertragung aus. Die "Ready for Transfer" - und "Transfer-Accepted" - Leitungen werden je nach Übertragungsrichtung sozusagen in ihrer Bedeutung umgeschaltet. Die "Mistake"-Leitung ist bidirektional ausgelegt, d. h. sie kann unabhängig von der Übertragungsrichtung einen auftretenden Fehler beiden Rechnernahtstellen sofort mitteilen. Stehen mehrere PDP-Rechner nahe beieinander, so kann für diese PDP-Gruppe eine gemeinsame Übertragungsleitung benutzt werden, jedoch müssen die "Request"- und "Select"-Leitungen separat vorgesehen werden. Dadurch können erhebliche Kabelkosten eingespart werden.

Bild 5 zeigt den Steuerteil der PDP-Nahtstelle. Der Datenteil mit der Umsetzung des 12 bit PDP-Wortes in 2 Wörter zu 6 Bit ist fortgelassen worden. Dieses Interface arbeitet an der Akku-Nahtstelle für den programmierten Datentransfer. Kommt ein "Select"-Signal vom Siemens Rechner, so wird für die Übertragungsrichtung: Eingabe in die Siemens 305 das erste in den externen Registern des PDP-Interface vorbereitend gespeicherte Datenteilwort mit einem "Ready" sofort zum Siemens - Rechner gesendet. Die Quittung "Transfer Accepted" nach Übernahme in der Siemens 305 setzt im PDP-Interface das "Data Ready" Flip-Flop zurück und löst entweder ein "Input Output Skip" (IOS) aus, wenn per Programm mit einem sog. "IOT-Befehl" abgefragt wird, oder es wird ein "Program-Interrupt" (PI) ausgelöst, wenn per Programm das "Interrupt-Enable" Flip-Flop gesetzt wurde. Daraufhin wird mit Hilfe eines weiteren IOT-Befehls das "Data Ready" Flip-Flop wieder gesetzt (Ready for Transfer) und das nächste Datenteilwort aus dem Akkumulator in das externe Register

übernommen und zum Siemens Rechner gesendet usw.

Bei der Übertragungsrichtung: Ausgabe aus der Siemens 305 wird mit dem "Ready for Transfer" wieder entweder ein IOS oder PI ausgelöst und mit einem IOT-Befehl das anstehende Datenteilwort in den Akkumulator übernommen. Gleichzeitig wird über einen monostabilen Multivibrator ein "Transfer Accepted" an die Siemens 305 zurückgegeben. Entsteht bei der Datenübertragung ein Fehler, wird dieser über den IOS oder PI der PDP mitgeteilt. Außerdem kann per Programm die Übertragungsrichtung und das "Not Select" als "End of Transfer" abgefragt werden.

Die Bilder 6,7 und 8 zeigen einige Testergebnisse der verwendeten Sender und Empfänger für die Übertragungsstrecke. Es wurden die Typen 75107 und 75110 von Texas Instruments verwendet. Die Sender arbeiten mit einer Konstantstromeinspeisung von 12 mA. Dieser Strom wird je nach logischem Eingangspegel des Senders entweder in den einen oder den anderen Draht des verwendeten verdrehten Telefonkabels eingespeist. Der Spannungsabfall am Kabelabschlußwiderstand wird dem Differenzverstärkereingang des Empfängers zugeführt. Am Ausgang des Empfängers steht dann ein logischer TTL-Pegel zur Verfügung. Die Oszillogramme zeigen eine Übertragung über zwei Kilometer. In der oberen Zeile ist das Eingangssignal des Senders, in der mittleren die beiden Eingangssignale des Empfängers, in der unteren Zeile das Empfänger-Ausgangssignal dargestellt. Die Laufzeit der Signale auf dem Kabel beträgt ca. 5 μ s/km. Überlappen sich die Empfänger-Eingangssignale nur um wenige Millivolt, so wird das entsprechende logische Ausgangssignal erzeugt. Eventuell müssen aus Gründen der Störsicherheit Teilerwiderstände am Empfängereingang vorgesehen werden.

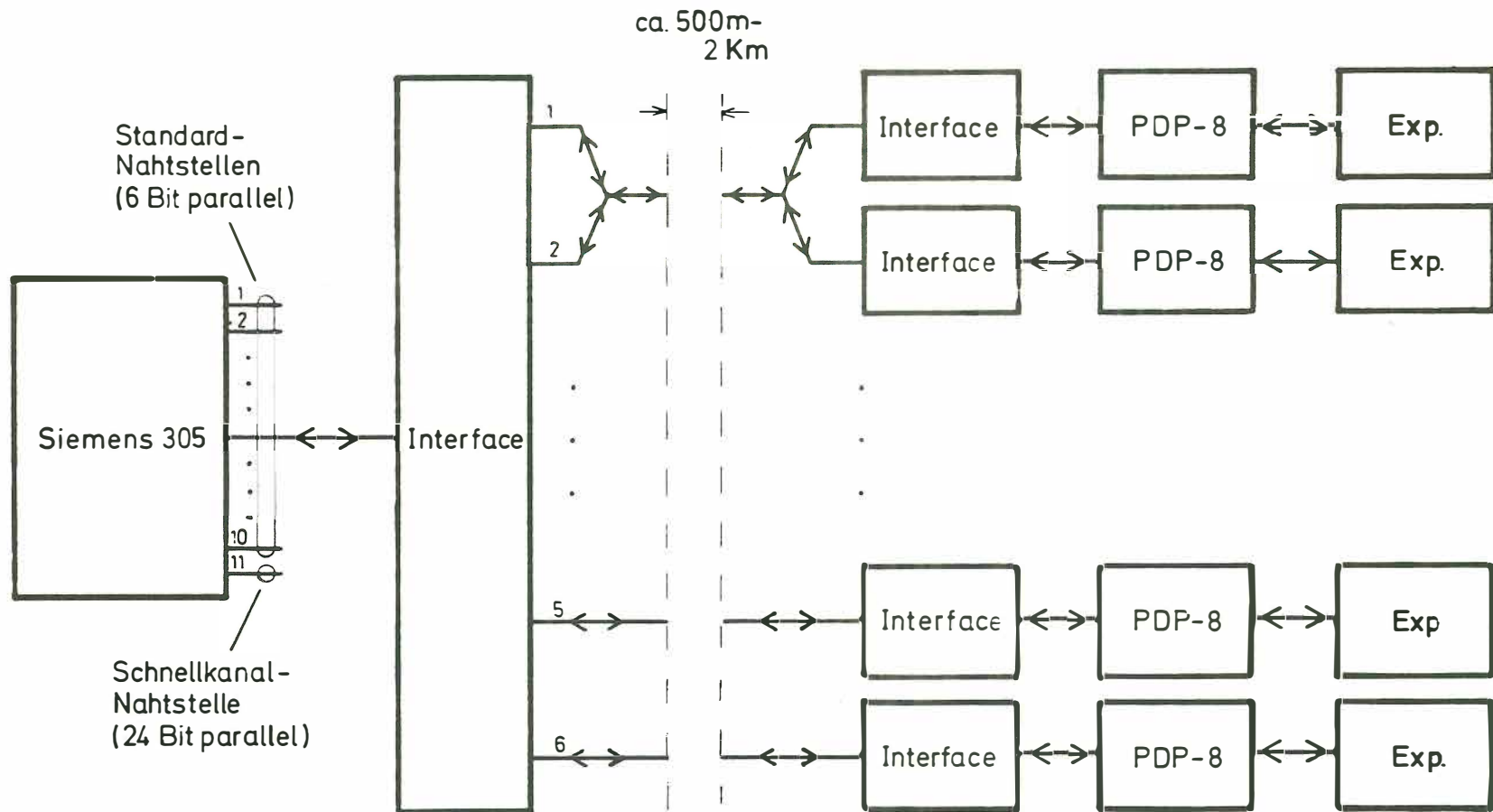


Bild 1

Koppelung von max. 6 PDP-8
mit einer Siemens 305

KFA-ZLE-NE

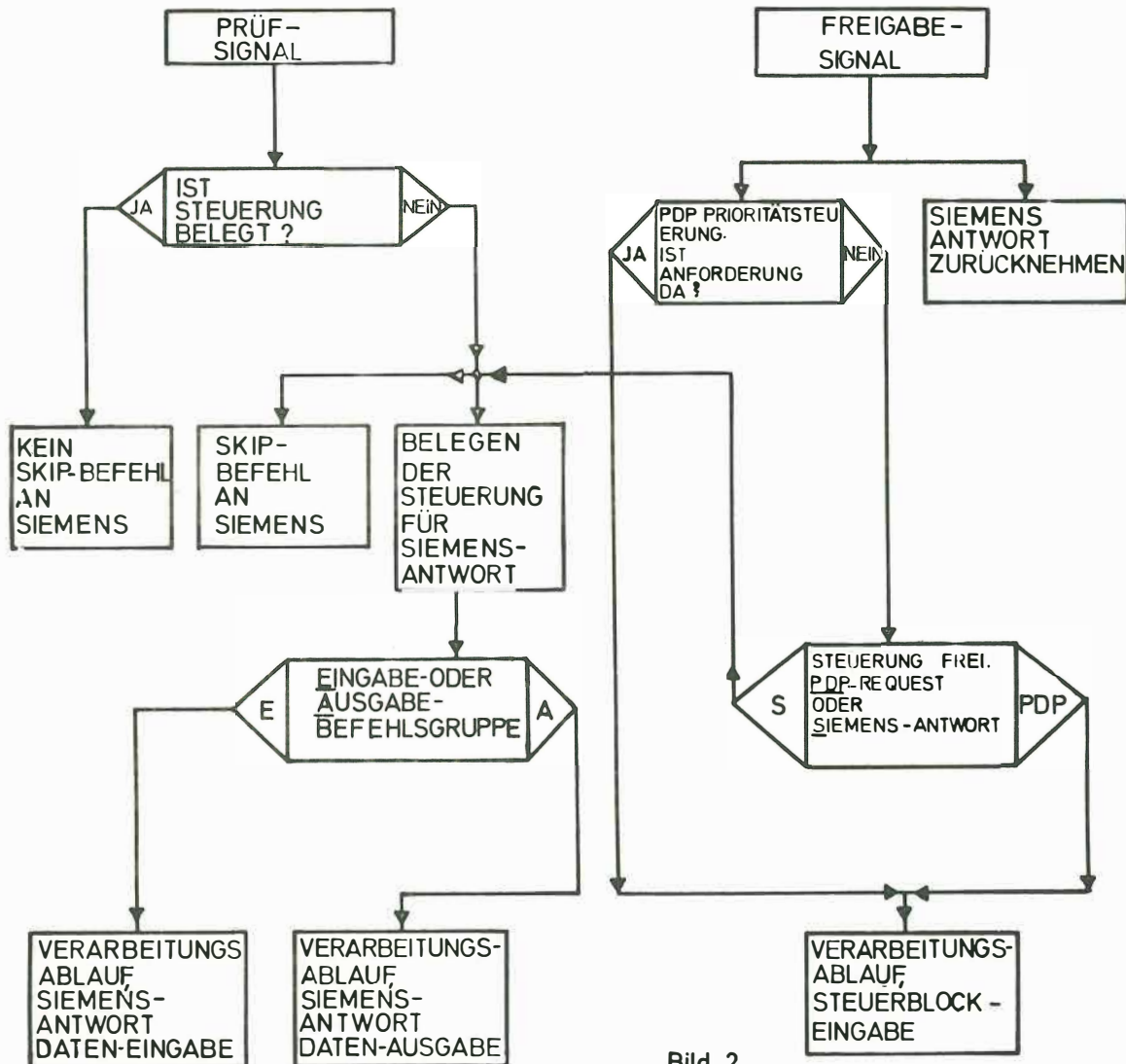
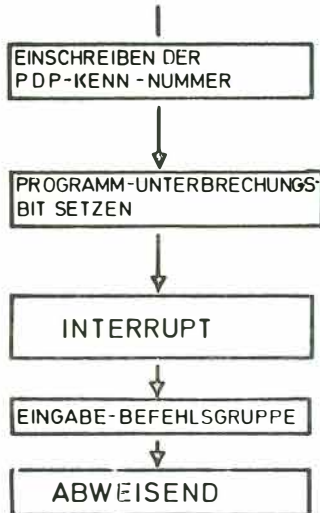
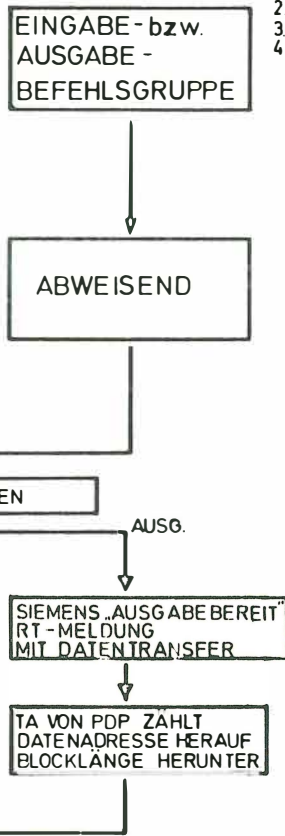


Bild 2
KFA/ZLE-NE

VERARBEITUNGSABLAUF
PDP-REQUEST
STEUERWORT-EINGABE



VERARBEITUNGSABLAUF
SIEMENS-ANTWORT
DATEN/EIN-AUSGABE



NORMALER/VORZEITIGER-BEFEHLSAB-
SCHLUSS

- 1.) ZEITKREIS-RT → TA-SELECT KEIN RT
- 2.) GESCHÜTZTER ASP-BEREICH
- 3.) PARITY-FEHLER
- 4.) BEI BLOCKLÄNGE NULL KOMMT-RT

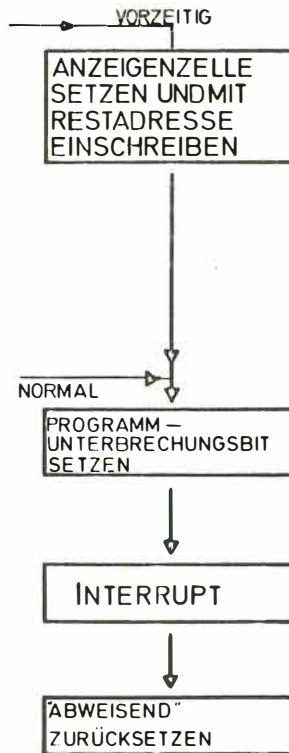


Bild 3
KFA/ZLE-NE

Signalleitungen (Übertragungsstrecke)

verdrilltes Telefonkabel
Laufzeit: 5ns/m oder
5µs/km

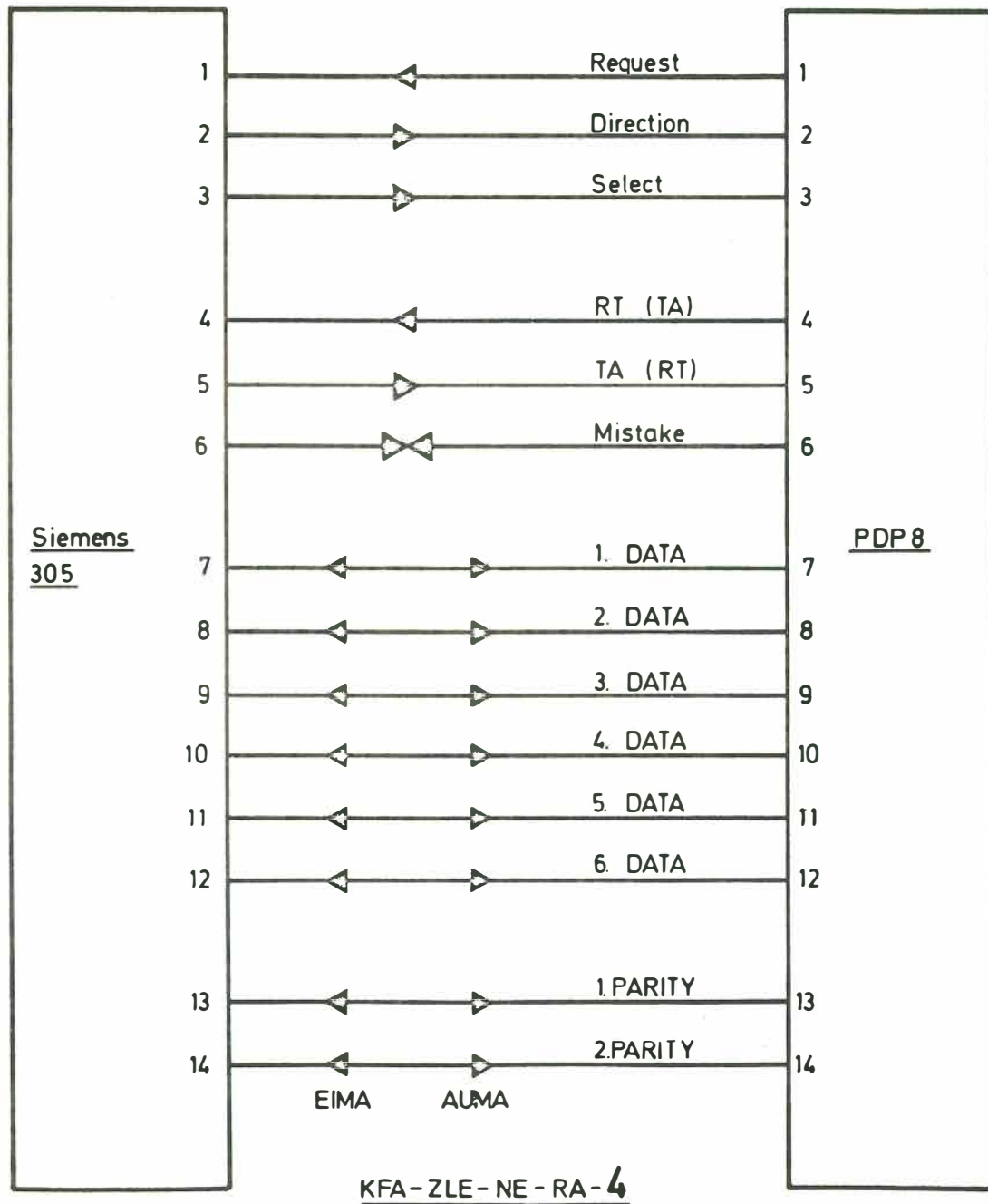


Bild 4

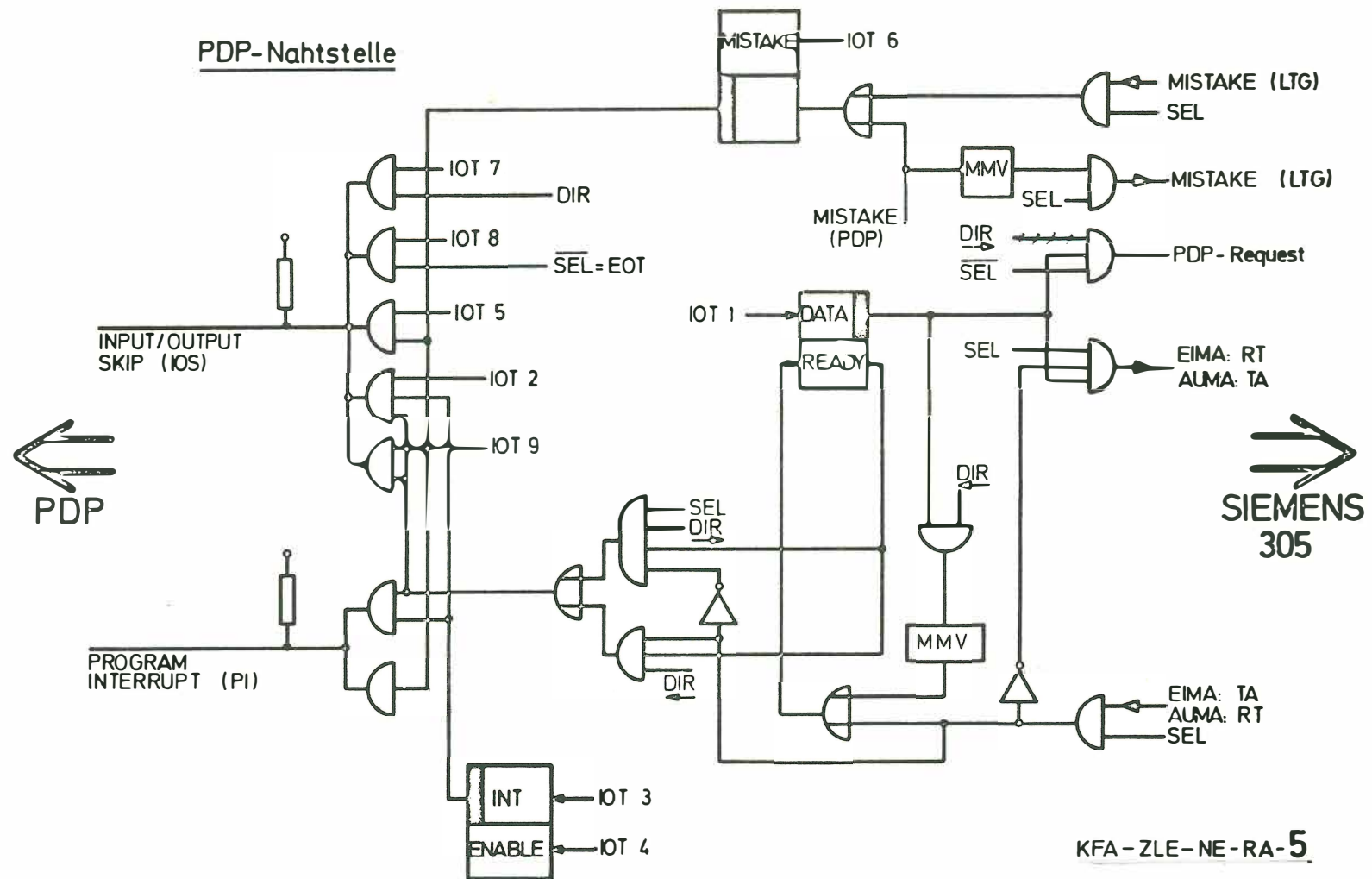
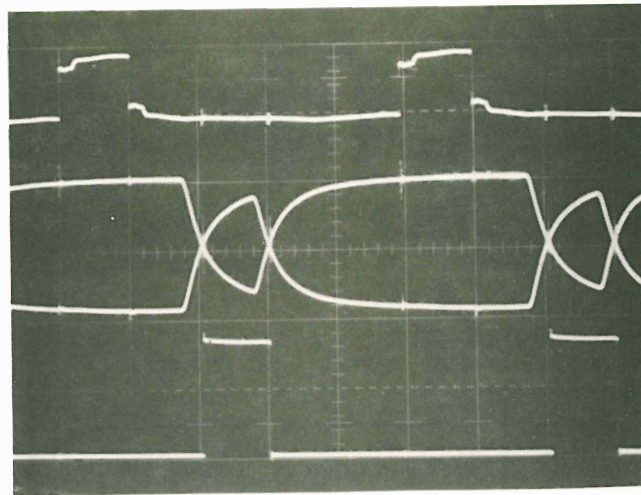


Bild 5

Bild 6



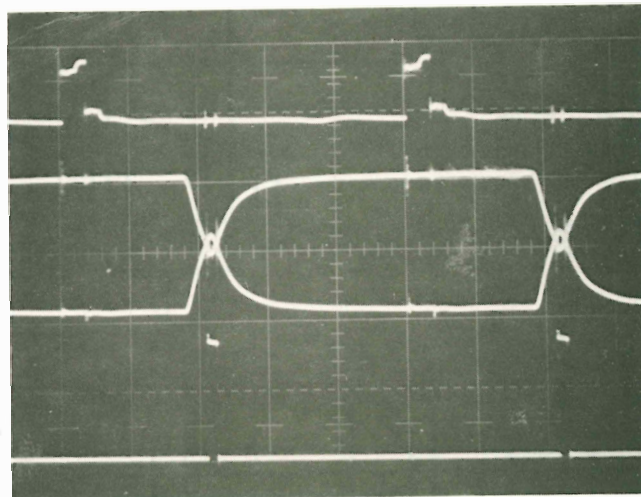
0,5V/cm

0,2V/cm

2V/cm

5 μ s / cm

Bild 7



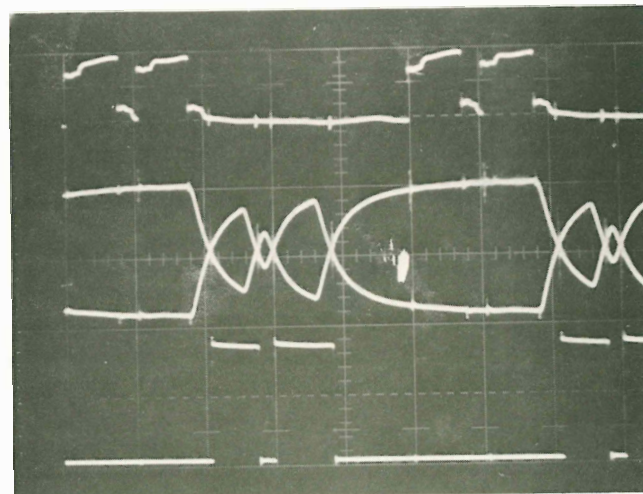
0,5V/cm

0,2V/cm

2V/cm

5 μ s / cm

Bild 8



0,5V/cm

0,2V/cm

2V/cm

5 μ s / cm

Dialogverkehr

bei der Rechnerkopplung Siemens 305 - PDP-8

F. R o n g e n

Kernforschungsanlage Jülich, Zentrallabor für
Elektronik/Abtl. Nukleare Elektronik

Dialogverkehr bei der Rechnerkopplung Siemens 305 - PDP-8

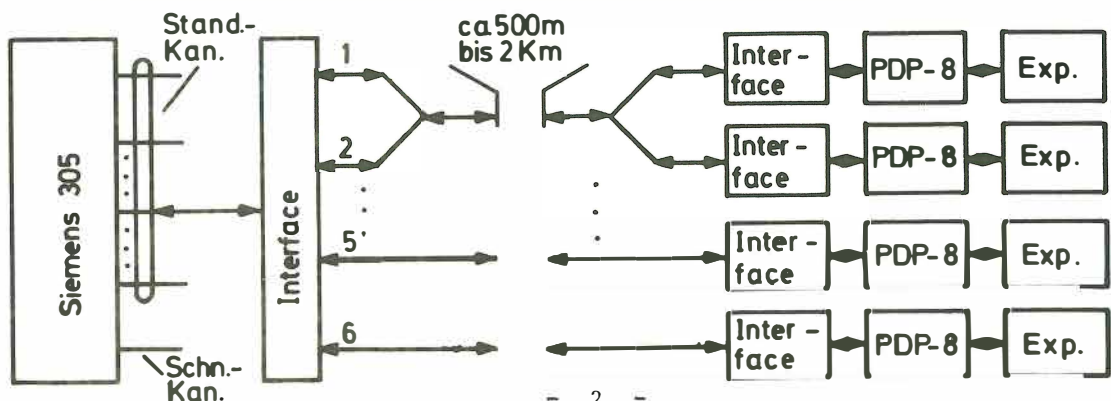
1. Einleitung

In verschiedenen Instituten der Kernforschungsanlage Jülich werden Experimente betrieben, die von einem Kleinrechner, z.B. einer PDP-8, mit z.T. sehr bescheidener Ausstattung, überwacht oder gesteuert werden. Einige dieser Experimente liefern sehr hohe Datenraten, bis zu 40kHz statistisch, und somit auch große Datenmengen. Die gesammelten Daten müssen nach einer bestimmten Meßphase durch umfangreiche Programme ausgewertet werden. Zu diesem Zweck werden die Daten häufig auf Lochstreifen, meist über eine Teletype Fernschreibmaschine, ausgegeben. Die Auswertung erfolgt dann später auf einer IBM 360/75. Von den Ergebnissen hängt aber in vielen Fällen die sinnvolle Weiterführung des Experimentes ab. Somit kann diese Methode sicher nicht als optimal angegeben werden.

Weiterhin besteht bei rechnergeführten Experimenten häufig die Forderung, Parameter, in Abhängigkeit von Experimentdaten, zur Einstellung bestimmter Geräte zu ermitteln. Die Methoden hierfür sind oft aufwendig und kompliziert. Folglich ist auch hiermit ein Kleinrechner meist überfordert.

Da nun in der KFA Jülich derzeit eine Studie über ein Rechnernetz erstellt wird, hat man sich, als ersten praktischen Beitrag, zu einer Koppelung der Experimentrechner mit einer im Zentrallabor für Elektronik vorhandenen Siemens 305 entschlossen. Die Entfernung zwischen der Siemens 305 und den Experimentrechnern liegt zwischen 500m und 2 km. Durch diese Koppelung wird es möglich, mit gegebenen Anlagen und Anlagenteilen für eine Reihe von Experimenten einen wesentlich größeren Komfort zu erlangen.

1.1 Möglichkeiten der hardwaremäßigen Zusammenschaltung



Aus Abb. 1 ist zu ersehen, welche Möglichkeiten der Zusammenschaltung zwischen den Experimentrechnern und der Siemens 3o5 vorgesehen wurden. Alle derzeit möglichen 6 Experimentrechner verkehren mit der Siemens 3o5 über eine Nahtstelle. Die eigentliche Übertragungsstrecke kann wahlweise aus gemeinsamen oder getrennten Leitungsbündeln bestehen. Hierdurch besteht die Möglichkeit, Experimentrechner in einem Ballungsgebiet zusammenzufassen und über nur ein Leitungsbündel zu bedienen. Die Aufspaltung einer Nahtstelle der Siemens 3o5 in 6 Teilkanäle hat für die erforderliche Software erhebliche Konsequenzen. Echte Simultanarbeit der Teilkanäle untereinander ist nicht möglich, jedoch scheint unter bestimmten Voraussetzungen eine geschachtelte Übertragung geeignet, womit man eine "quasi Simultanarbeit" erreichen könnte. Ob die Voraussetzungen hierfür geeignet sind, werden weitere Untersuchungen zeigen müssen.

Bei Bedarf und Eignung der Siemens 3o5 ist an den Anschluß von mehr als 6 Experimentrechnern über mehrere Nahtstellen gedacht, wobei dann auch echte Simultanarbeit möglich wird. Die Tatsache, daß mehrere Experimentrechner nur eine Übertragungsleitung benutzen, ist für die erforderlichen Programme vollkommen bedeutungslos.

1.2 Rechnerhierarchie

Das vorliegende Konzept dient, wie schon gesagt, dazu, die Kleinrechner am Experiment zu entlasten. Aus dieser Tatsache geht hervor, daß es sich hierbei nur um eine Rechnerhierarchie handeln kann, d. h. die Siemens 3o5 soll keinesfalls die Möglichkeit erhalten, selbst bei den Experimentrechnern Entlastung zu suchen. Die Initiative zu einem Datenaustausch muß vom Experimentrechner ausgehen. Diese Vorschrift gilt aber nur für den Austausch von Experimentdaten. Im Gegensatz hierzu sind Fälle denkbar, bei denen z.B. der Operateur an der Siemens 3o5 über den Bedienungsblattschreiber Meldungen an die Operateure der Experimentrechner geben will und umgekehrt, oder aber in den Funktionsablauf eingreifen möchte. In diesen Fällen kann und muß die Initiative von der Siemens 3o5 ausgehen.

2. Organisation des Dialogverkehrs.

2.1 Erfordernisse

Nun kann man sich überlegen, was für das Zustandekommen eines Dialogverkehrs alles erforderlich ist. Sicherlich ist bei einer TK-Meldung die Übermittlung einer Kennung notwendig, um eindeutig festzustellen, wer überhaupt Verkehr mit der Siemens 3o5 wünscht.

Weiterhin muß ein Steuerblock übertragen werden, aus dem eindeutig die Wünsche des Experimentrechners hervorgehen.

Zum Aufbau und zur Abarbeitung von Warteschlangen, zur Aktivierung der Auswerteprogramme, für den Externspeicherverkehr usw. sind umfangreiche Steuerfunktionen erforderlich. Leistungsfähige Ein-Ausgaberoutinen sollten als Selbstverständlichkeit angesehen werden.

Und schließlich muß die Auswertung und Bearbeitung von Anzeigen sichergestellt sein.

Es gibt noch eine Menge beachtenswerter Punkte, auf die hier aber nicht näher eingegangen werden soll.

Zur TK-Kennung

Die TK-Kennung wird bei einer TK-Meldung von der Steuerung erzeugt und hardwaremäßig in eine fest vereinbarte Zelle des ASP der Siemens 3o5 eingeschrieben. Ein Programm sorgt dann für die Sicherstellung und Bearbeitung dieser Kennung.

Zum Steuerblock

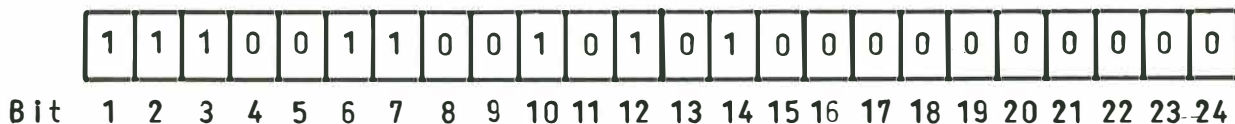


Abb. 2

Abb. 2 zeigt nun, wie ein solcher Steuerblock formal aufgebaut sein muß. Er besteht derzeit aus 24 Bit, ist aber bei Bedarf beliebig erweiterbar. Den einzelnen Bits kommt dabei folgende Bedeutung zu:

- Bit 1 : Übertragung PDP-8 → Siemens 305
- Bit 2 : Verarbeitung der Datei X mit dem Programm
- Bit 3 : Übertragung Siemens 305 → PDP-8
- Bit 4-7 : Dateiname X des Verarbeitungsprogrammes
- Bit 8-11 : Dateiname Y der Daten
- Bit 12 : Kurze/lange Verarbeitung
- Bit 13-24 : Blocklänge der Daten in 24 Bit - Worten.

Die möglichen Wünsche sind beliebig kombinierbar, wobei darauf zu achten ist, daß der Dateiname "0" nicht zulässig ist.

Es scheint noch erwähnenswert, daß im Prinzip auch ein Datenaustausch der Experimentrechner untereinander möglich ist.

Zu den Steuerfunktionen:

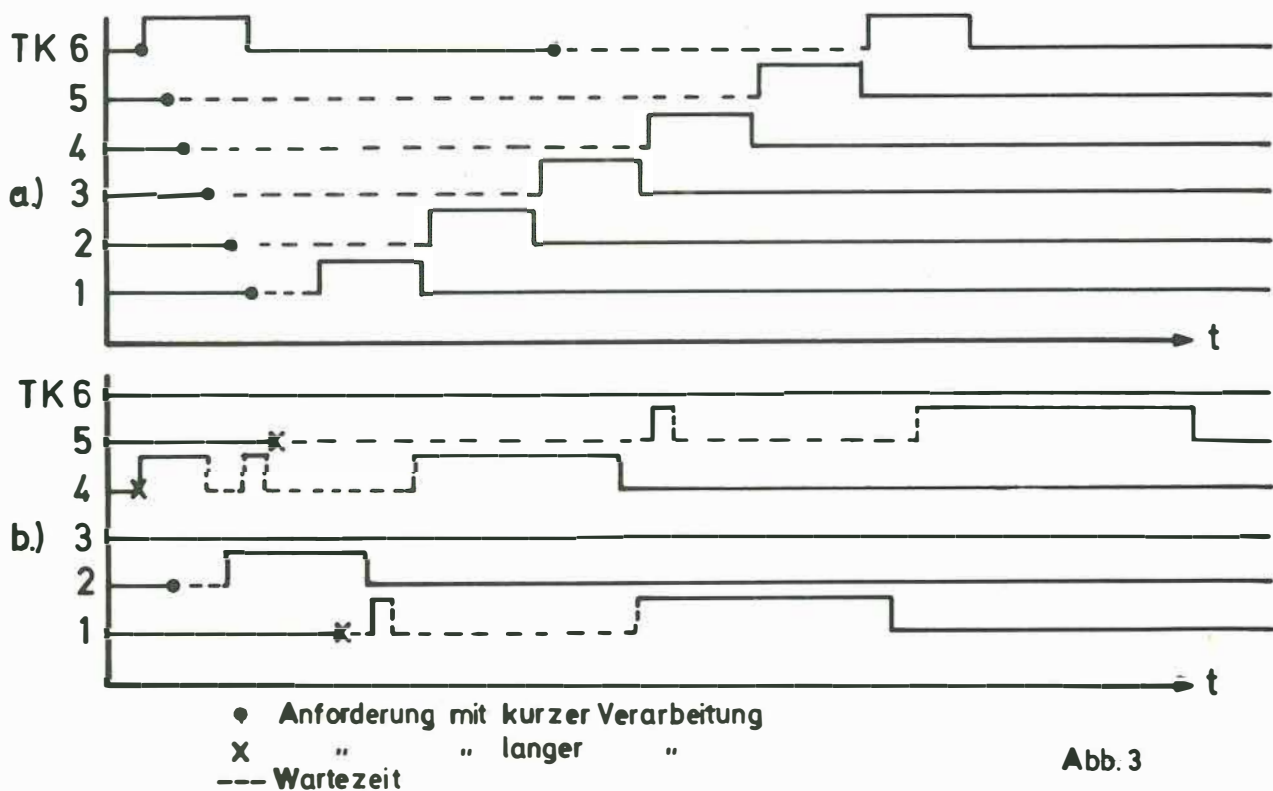


Abb. 3

In Abb. 3a und b sind nun zwei Fälle stark vereinfacht dargestellt, wie etwa die Anforderungen der verschiedenen Teilkanäle zusammen treffen können und abgearbeitet werden. Wie schon erwähnt, kann jeder Experimentrechner zwischen einer langen und einer kurzen Verarbeitung wählen. Wird nun vom Experimentrechner eine Übertragung mit Verarbeitung und Rücktransfer gewünscht, so liegt sicher nicht in jedem Falle die Notwendigkeit vor, die Ergebnisse schnellstens zurückzubekommen, wodurch die Steuerung für die gesamte Zeit der Verarbeitung, aus organisatorischen Gründen, belegt bleiben müßte.

Eine kurze Verarbeitung würde also bedeuten, daß der Experimentrechner schnellstens Ergebnisse benötigt und keine langen Wartezeiten in Kauf nehmen kann. Bei einer langen Verarbeitung wird hingegen nach Übertragung der Daten in die Siemens 305 die Steuerung wieder freigegeben. Ein weiteres Programm nimmt nun die Auswertung im Simultanbetrieb vor und wird selbständig, bei auftretenden Wartezeiten, die Ergebnisse zurückschicken.

Zu Ein-Ausgaberoutinen

Um den Programmierer am Experimentrechner nicht unnötig zu belasten, wird ihm nur ein einziger Abruf für jeden beliebigen Auftrag an die Siemens 305 zur Verfügung gestellt. Alles Weitere übernimmt dann selbständig ein Systemprogramm, welches während des Datenverkehrs die Programmsteuerung zeitweise für den Anwender wieder freigibt. Bei dem Systemprogramm kann auf bestehende Speicherbelegung und Prioritätsverteilung anfallender Interrupts Rücksicht genommen werden.

Das Systemprogramm der Siemens 305 dient zur Koordinierung des Verkehrs der 6 Teilkanäle, der Steuerblockdecodierung, der Fehlerbehandlung und der Aktivierung der Auswerteprogramme. Ferner stellt er die Verbindung zum Betriebssystem her. Als Befehlsgruppen bzw. Befehle, mit denen die Steuerung angesprochen werden kann, stehen zur Verfügung: EINGABE, AUSGABE, FREIGABE und PRÜFEN.

Die Befehlsgruppen EINGABE und Ausgabe sind bei der Firma Siemens verfügbar und für die Nahkoppelung zweier gleichberechtigter Rechner über eine Nahtstelle konzipiert. Für die hier beschriebene Rechnerhierarchie mußten diese Aufrufe leicht modifiziert werden. Die Befehle FREIGABE und PRUEFEN sind zu Koordinierungszwecken neu geschaffen worden.

Der Befehl FREIGABE dient dazu, die nach Abschluß einer Übertragung noch belegte Steuerung für weitere Anforderungen wieder freizugeben, während mit dem Befehl PRUEFEN zu jedem beliebigen Zeitpunkt der Zustand der Steuerung abgefragt werden kann.

Zur Anzeigenbehandlung

Damit überhaupt Fehler vom Programm erkannt und ausgewertet werden können, schreibt die Steuerung ein bestimmtes Bit, in eine fest vereinbarte ASP-Zelle und veranlaßt eine Programmunterbrechung. Hiernach erfolgt die Bearbeitung der entsprechenden Anzeige durch das Programmsystem und z.T. auch durch das Betriebssystem der Siemens 305. Die wichtigsten Anzeigen sind: PARITY-FEHLER, GESCHUETZTER BEREICH, ZEITFEHLER und BLOCKLAENGEN FEHLER.

Bei einigen Anzeigen, wie z.B. beim Parityfehler, versucht das Programm, durch mehrfache Wiederholung doch noch eine fehlerfreie Übertragung zustande zu bringen. Gelingt das nicht, so wird die Übertragung abgebrochen und eine entsprechende Blattschreibermeldung ausgegeben. Notfalls wird der entsprechende Teilkanal für jede weitere Anforderung gesperrt, da vermutlich ein Hardwarefehler vorliegt. (Leitungsbruch, defekte Bauelemente usw.).

2.2 Teilblockorganisation

Es sei noch erwähnt, daß die Datenübertragung in Teilblöcken vorgesehen ist, um somit die Möglichkeit leistungsfähiger Fehlerprüf- und Substmöglichkeiten zu erhalten. Über zweckmäßige Teilblocklängen werden derzeit noch Untersuchungen angestellt. Es scheint jedoch jetzt schon eine Teilblocklänge von 64 bzw. 128 PDP-8-Worten als sinnvoll, um die Übertragungszeit bei sehr häufig auftretenden Fehlern in sinnvollen Grenzen zu halten. Tritt ein Übertragungsfehler auf, so wird der gesamte Datenteilblock erneut übertragen. Ebenfalls ist die Anzahl der zulässigen Übertragungsfehler innerhalb eines Teilblockes Gegenstand laufender Untersuchungen. Auch hierbei würde eine zu große Zahl den Dialogverkehr in erheblichem Maße stören.

3. Schlußbetrachtungen

Praktische Erfahrungen mit der hier beschriebenen Rechnerkopplung liegen leider noch nicht vor, da sich dieses Projekt augenblicklich noch im Teststadium befindet. Der praktische Betrieb wird aller Voraussicht nach in einigen Wochen aufgenommen werden, wobei zunächst Erfahrungen mit einem Experimentrechner die Grundlage für den weiteren Ausbau schaffen sollen. Im Prinzip ist jedoch heute schon der Verkehr mit 6 Experimentrechnern möglich.

Sichtgeräte für Prozeßrechner des Siemens Systems 300

von

Klaus Bindewald

Seit einigen Jahren werden bei kommerziellen Datenverarbeitungsanlagen Sichtgeräte eingesetzt. In letzter Zeit werden sie auch bei Prozeßrechnern in zunehmendem Maße verwendet.

Sichtgeräte, oder wie man auch sagt, Datensichtstationen oder Display, gibt es in vielen Ausführungen. Alle Sichtgeräte haben aber folgende Hauptbestandteile:

Kathodenstrahlröhre, auf deren Schirm die Information in Form von alphanumerischen Zeichen, Punkten oder Linien dargestellt wird;

Ablenkverstärker und Hochspannungsteil (oft mit der Bildröhre konstruktiv zu einer Einheit zusammen gefaßt);

Bildaufbausteuerung zum Interpretieren der einzelnen Datenworte und Einschalten der entsprechenden Bildgeneratoren;

Datenübertragungssteuerung zur Verbindung des Sichtgerätes mit der Datenverarbeitungsanlage;

Bildwiederholungsspeicher zur Regenerierung des Bildes;

Eingabegeräte, wie Tastatur, Rollkugel oder Lichtgriffel.

Sichtgeräte bilden heute ein ideales Mittel für den Dialog mit dem Rechner, da sie genügend schnell sind, um die Möglichkeiten des Menschen in Verbindung mit einer Datenverarbeitungsanlage auszunützen.

Nach einem groben Schema lassen sich Sichtgeräte in zwei Gruppen einteilen:

1. Universalsichtgeräte:

Diese bieten unbegrenzte Möglichkeiten der Informationsdarstellung durch in Hardware realisierte Zeichen-, Kreis- und Vektorgeneratoren, durch einen meist als Kernspeicher mit wahlfreien Zugriff ausgeführten Bildwiederholungsspeicher, sowie einem großen Spektrum an Eingabegeräten und -möglichkeiten.

Ein Nachteil dieser Geräte ist aber der meist sehr hohe Preis.

2. Spezialsichtgeräte:

Es handelt sich um relativ preiswerte Geräte mit Spezial Eigenschaften. Beispiele sind

alphanumerisches Sichtgerät
Analog Sichtgerät
Speichersichtgerät.

Für Prozeßrechner des Siemens Systems 300 sind folgende Geräte lieferbar:

1. Alphanumerisches Sichtgerät:

Datensichtstation 8150 . Hierbei handelt es sich um ein Gerät zur Darstellung von Buchstaben, Zahlen und Sonderzeichen, im Prinzip wie beim Blattschreiber. Gegenüber diesem hat es aber den Vorteil, daß es lautlos arbeitet, eine wesentlich höhere Datenübertragungsrate hat (bis 240 Zeichen/sec.) und einen eigenen Datenspeicher besitzt. Letzterer dient zwar primär der Bildwiederholung (zur flimmerfreien Darstellung wird das Bild ca. 60 mal in der Sekunde neu geschrieben), bietet aber sekundär die Möglichkeit, den Bildaufbau

von der Datenübertragung zu trennen, d. h. erst wird der zu übertragende Text durch Bedienung einer Tastatur zusammengestellt, dann, wenn notwendig, korrigiert (Überschreiben falscher Zeichen durch die Richtigen, Einfügen von etwa vergessenen Zeichen) und erst nach Betätigung einer Übertragungstaste zur DVA übertragen. Damit werden Fehlbelegungen der DVA weitgehend vermieden.

Das Sichtgerät kann überall da eingesetzt werden, wo bisher schon Blattschreiber eingesetzt wurden, dh. zur Bedienung, zur Einsichtnahme und Korrektur von Programmen, zur Ausgabe von Stör- und Alarmmeldungen.

Die Datensichtstation 8150 gestattet die Darstellung von 1080 Zeichen in 20 Zeilen zu je 54 Zeichen. Übertragungsgeschwindigkeit wählbar zu 120 oder 240 Zeichen/sec.

Der Anschluß der Datensichtstation erfolgt an einem Teilkanal des Fernschreibeelementes FSK über eine Anschaltung BIKA-S mit folgenden Möglichkeiten:

- a) BIKA-S Ausbaustufe D1, Anschluß an Fernschreibelement FSK (DVA 302 - 306) für drei Sichtgeräte nur zur Ausgabe von Daten.
- b) BIKA-S Ausbaustufe 03, Anschluß an Zeichenübertragungsbaustein ZAP 301 (DVA 301) für drei Sichtgeräte nur zur Ausgabe von Daten von der DVA zur Datensichtstation. In beiden Fällen werden die einzelnen Datensichtstationen durch Adreßzeichen einzeln angesprochen.
- c) BIKA-S Ausbaustufe D12, Anschluß an Fernschreibelement FSK (DVA 302 - 306), ein Sichtgerät zur Eingabe und Ausgabe von Daten zur und von der DVA. Bei dieser Datensichtstation ist der Anschluß eines Ausgabeblattschreibers zur Protokollierung des Bildschirminhaltes möglich.

Entfernung zwischen Anschaltung und Datensichtstation max. 200 m, mit GDN-Einrichtung (Gleichstromdatenübertragung mit niederem Pegel) 20 km und mit Modem beliebig.

Eine große Anzahl dieser Geräte ist in Kraftwerken, Pipelines und in Satellitenkontrollzentren eingesetzt.

2. Analoysichtgerät ASG 34:

Der Steuerteil dieses Sichtgerätes übernimmt aus dem Arbeitsspeicher der DVA Daten in digitaler Form und setzt diese über Digital-Analog-Umsetzer in Analogspannungen zur Ansteuerung von X-Y-Monitoren um. Um die Hardware preisgünstig zu gestalten, verwendet dieses Sichtgerät den Arbeitsspeicher der DVA als Bildwiederholungsspeicher.

Das Sichtgerät besteht aus Sichtgerätanschaltung ASGA 34, Bedienfeld ASGB 34 und ein bis drei Bildschirmseinheiten. Hier kann jeder handelsübliche Oszillograf verwendet werden, der eine Eingangsspannung von 1 V für die Ablenkung über den ganzen Bildschirm benötigt; besonders bewährt haben sich die X-Y-Monitoren von Siemens M 07103-A1 (Bildfläche 8 x 10 cm) oder Hewlett-Packard 1300 A (Bildfläche 20 x 25 cm).

Da das Sichtgerät den Arbeitsspeicher der DVA als Bildwiederholungsspeicher benutzt, erfolgt der Anschluß über das Prozeßelement P 3 AS bzw. KS mit direktem Arbeitsspeicherzugriff.

Die Anschaltung ASGA 34 stellt alle 8 μ s eine Anforderung auf ein neues 24 Bit-Wort. Dieses kann entweder ein Steuerwort (zur Einstellung der Betriebsart des Sichtgerätes) oder ein Datenwort (mit Bildinformation) sein.

Folgende Betriebsarten sind möglich:

- a) X-Y-FREI- POSITIONIERT. Von den 24 Bit werden je 10 Bit für die X- bzw. Y-Ablenkung und 3 Bit für die Hellsteuerung des Punktes in 8 Stufen ausgewertet.
- b) EINZELKURVENDARSTELLUNG. Mit jeder Anforderung wird ein interner Zähler um eine Stelle erhöht. Ein nachgeschalteter Digital-Analog-Umsetzer erzeugt daraus die Ablenkspannung in X-Richtung. Es können Kurven mit 64, 128, 256, 512 und 1024 Punkten jeweils über den gesamten Bildschirm gespreizt dargestellt werden. Von den 24 Bit des Datenwortes werden 22 Bit für Y ausgewertet, wobei jeweils 10 aufeinanderfolgende Bit über einen Meßbereichswähler ausgewählt und dargestellt werden können. Über einen Schalter am Bedienfeld

läßt sich jeder 16. Punkt besonders hell darstellen. Über eine Drucktastengruppe kann in der aus 1024 Punkten bestehenden Kurve ein beliebiges Viertel markiert und über den ganzen Bildschirm gespreizt dargestellt werden.

- c) ISOMETRISCHE DARSTELLUNG. Hier werden abhängig von der Information des Steuerwortes 32 Kurven mit 32 Punkten auf 64 Kurven mit 64 Punkten dargestellt, wobei nach jedem Durchlauf einer Kurve ein in gewissen Grenzen wählbarer Anteil x und y zur Grundausslenkung addiert wird. Dabei entsteht bei entsprechenden Daten ein räumlich wirkendes Bild.
- d) HELBIGKEITSGESTEUERTE DARSTELLUNG. Hier wird ein unsichtbares Raster von 32×32 oder 64×64 Punkten geschrieben. Diese Rasterpunkte werden in acht Stufen hellgetastet. Die dazu nötigen 3 Bit können aus den 22 Informationsbit über einen Meßbereichwähler ausgewählt werden.

Durch einen auch noch nachträglich einbaubaren Zusatz können die Möglichkeiten des Sichtgerätes noch erweitert werden:

Über ein Schalterfeld im Bedienteil kann in die Einzelkurve eine blinkende Marke eingeblendet werden. Die Koordinaten des markierten Punktes können über das Prozeßelement P 1 K oder P 3 ES bzw. KS in die DVA eingegeben werden.

Über das selbe Schalterfeld kann durch die isometrische Darstellung ein Schnitt gelegt und dieser anschließend über den ganzen Bildschirm gespreizt dargestellt werden. Über einen weiteren Schalter kann an Stelle des linearen Verstärkers in Y-Richtung ein logarithmischer Verstärker eingeschaltet werden.

Bis zu 2500 Punkte können flimmerfrei auf dem Bildschirm mit einer Bildwiederholungsfrequenz von 50 Hz dargestellt werden. Bei mehr als 2500 Punkten wird auf 25 Bildwechsel umgeschaltet. Eine Darstellung von mehr als 5000 Punkten pro Bild ist nicht sinnvoll, da das Bild dann schon sehr stark flackert.

Zusätzlich ist zum Analoysichtgerät ASG 34 eine Rollkugelsteuerung ROKS lieferbar. Über eine in einer Tischplatte versenkt eingebaute Rollkugel kann eine Marke frei über den Bildschirm verschoben und die X - und Y-Koordinaten der Marke auf Wunsch oder automatisch in die DVA über ein Prozeßelement P3 ES bzw. KS eingegeben werden. Die Rollkugelsteuerung erlaubt eine bitgenaue Markierung des Punktes, da bei exakter Positionierung der Marke auf einen Punkt, die Marke besonders hellgesteuert dargestellt wird.

Einsatzbeispiele für das Analoysichtgerät finden sich in der Kernphysik zur Beobachtung von Experimenten, in Prüffeldern zur raschen Anzeige der Meßergebnisse, in der Nuklearmedizin (z.B. Darstellung von Szintigrammen), bei der Auswertung der Meßdaten von Wetterballons, in Forschung und Entwicklung, kurz überall da, wo ein schneller Zugriff zu Daten in grafischer Darstellung notwendig ist.

3. Speichersichtgerät SGS:

Zur raschen Bildauswahl in Zusammenhang mit einem Trommelplotter Calcomp 565 wurde das Speichersichtgerät SGS unter Verwendung des Tektronix Speicherbildschirmes 611 entwickelt. Die Steuerung SAS dieses Gerätes wird über eine Umschalteneinrichtung an eine Universalsteuerung UAN angeschlossen und kann wahlweise an Stelle des Plotters betrieben werden.

Entsprechend der Schrittweite (0,1 mm) des Plotters bei einer Gesamtbreite von 278 mm in der Y-Richtung können 2780 Punkte in der Y-Richtung dargestellt werden, während in der X-Richtung (beim Plotter max. 36 m) eine Beschränkung auf 3580 Punkte eingeführt wurde, um das Format des Bildschirmes optimal zu nutzen. Die Schreibgeschwindigkeit beträgt einige 100 μ s pro Punkt; damit ergibt sich eine gegenüber dem Plotter um den Faktor 15 geringere Bildausgabezeit.

Der Vorteil des beschriebenen Sichtgerätes liegt in der Möglichkeit der raschen Vorauswahl von Daten, wobei anschließend nur die dokumentierungswürdigen Daten über einen Plotter auf Papier ausgegeben werden. Dabei findet die selbe Software Verwendung.

Gaffal
Technische Hochschule
München

"Eine 3o5 als digitaler Partner
in einem Hybridsystem"

Teil 1:

Organisation des Datenverkehrs zwischen einer 305
und einem angekoppelten Analogrechner

F. Gaffal

Zusammenfassung

Am Institut für Meß- und Regelungstechnik der
TU München wurde ein Hybridrechenzentrum aufge-
baut.

Hierzu wurde eine Siemens 305 mit einem Analog-
rechner und einer Siemens 4004/55 gekoppelt.
Die Verbindung 305-Analogrechner erfolgte über
P3-Elemente. Es wird der gerätetechnische Aufbau
und die Organisation des Datenverkehrs im Organi-
sationsprogramm beschrieben.

Inhalt

- 1. Einleitung

- 2. Gerätetechnischer Aufbau
- 2.1. Die Steuer- und Kontrollkopplung
- 2.1.1. Aufgaben
- 2.1.2. Befehlsstruktur
- 2.2. Die Datenkopplung
- 2.2.1. Aufgaben
- 2.2.2. Anforderungen an den Datenverkehr
 in einem Hybridsystem
- 2.2.3. Ablauf des Datenverkehrs

- 3. Programmorganisation
- 3.1. Organisationsprogramm ORG I
- 3.2. Schnelle Wechsellpufferorganisation
- 3.3. Einpufferorganisation
- 3.4. Beispiel

Anhang

1. Einleitung

Am Institut für Meß- und Regelungstechnik der TU München wurde ein Hybridrechenzentrum eingerichtet um bei der Simulation von Kernreaktoren und Reaktorkomponenten unter Betriebs- und Unfallbedingungen als Hilfsmittel zu dienen. Hierzu müssen

umfangeiche digitale Berechnungen,
Simulationen am Analog- bzw. Hybridrechner,
Aufbereiten analoger Daten zur digitalen
Verarbeitung
durchgeführt werden.

Das Hybridsystem besteht aus drei elektronischen Rechenanlagen:

einem Analogrechner EAI 8800
dem Prozeßrechner Siemens 305 und
dem Digitalrechner Siemens 4004/55

Alle drei Rechenanlagen sind miteinander gekoppelt. Die 305 und die 4004/55 tauschen Digitalwerte über ihre Arbeitsspeicher mit Hilfe der Datenaustauschsteuerung DAK aus. Für den Hybridbetrieb ist die 305 der unmittelbare Partner für den Analogrechner. Diese beiden Rechner tauschen über das Hybridkoppelsystem HKS Daten und Steuersignale aus. Auf diese Kopplung soll im folgenden näher eingegangen werden.

2. Gerätetechnischer Aufbau

Sowohl die 305 wie auch die angeschlossenen Prozeßelemente mußten den speziellen Erfordernissen des hybriden Rechnens angepaßt werden. Dabei sind eine Reihe von Problemen aufgetaucht, die durch Hard- und Softwaremaßnahmen gelöst wurden.

Das Hybrid-Koppelsystem HKS (Bild 2) gliedert sich in zwei nahezu unabhängige Teile, die Steuer- und Kontrollkopplung und die Datenkopplung.

2.1. Die Steuer- und Kontrollkopplung

2.1.1. Aufgaben

Die Steuer- und Kontrollkopplung verarbeitet die Befehle des Digitalrechners an den Analogrechner. Diese Befehle dienen dazu

die Betriebszustände des Analogrechners zu steuern,

die Rechengeschwindigkeit des Analogrechners auszuwählen,

Potentiometer (für Koeffizienten) einzustellen,

die Werte aller Rechenkomponenten auszulesen,

Kontrollinformation über den Zustand des Analogrechners und der Datenkopplung auszulesen,

digitale Information in Form von Einzelbits oder 16-bit Worten in beiden Richtungen auszutauschen

Alarme durch Fehlerfälle, Zeitregister, Tasten und das Analogprogramm zu registrieren und an die 305 zu melden (zur Verarbeitung dieser Alarme siehe Beitrag von Herrn Ehrenberger).

und den Datenverkehr zwischen Analogrechner und 305 anzustoßen.

2.1.2. Befehlsstruktur

Alle Befehle die sich an die Bearbeitung für die schnellen Alarme und die Datenkopplung richten, sind unmittelbar EAW-Befehle und werden ohne weiteren Speicherzugriff ausgeführt.

Für Befehle an den Analogrechner wird durch EAW-Befehle die Adresse der Arbeitsspeicherzelle an das HKS übergeben, in der das zugehörige Befehlswort steht. Dieses Befehlswort wird durch einen weiteren Arbeitsspeicherzugriff (SEX-Zustand der 305) in das Informationsausgaberegister der P3KS/A gelesen und durch das HKS dekodiert. Je nach Art des Befehls wird die Information des Befehls an den Analogrechner übergeben oder durch einen weiteren Arbeitsspeicherzugriff die gewünschte Information vom Analogrechner in den Digitalrechner über das Informationseingaberegister der P3KS/A eingeschrieben.

2.2. Datenkopplung

2.2.1. Aufgaben

Die Datenkopplung steuert den Datenverkehr zwischen Analogrechner und Digitalrechner und übergibt Daten von einem Rechner zum anderen.

2.2.2. Anforderungen an den Datenverkehr in einem Hybridsystem

Während der Simulation werden Daten zwischen dem Analogrechner und dem Digitalrechner ausgetauscht. Dabei kann es sich handeln um

- . einzelne Werte einer Variablen
- . viele Werte einer Variablen
- . einzelne Werte mehrerer bel. Variablen
- . viele Werte mehrerer Variablen

Deshalb müssen die Wandler speziell dafür geeignete Betriebszustände haben.

Diese sind:

- . Wahlweiser Zugriff
- . Fortlaufende Ein-Ausgabe eines Kanals
- . Sequentielle Arbeitsweise

Bei wahlweisem Zugriff werden die Adressen für die gewünschten Kanäle über die P3AS an die Steuerung für die Wandler übergeben (vgl. Bild 3, 4). Die Adressen können beliebig gewählt werden. Beim Digital-Analogwandler wird die Adresse gemeinsam mit dem Digitalwert für den Wandler übergeben.

Bei der sequentiellen Arbeitsweise läuft der Wandler selbständig zyklisch vom ersten zum letzten Kanal. Es brauchen deshalb keine Adressen übergeben werden, wodurch Zeit beim Aufbereiten des Puffers bzw. Speicherplatz für den Puffer gespart wird.

Der Betriebszustand fortlaufende Ein- bzw. Ausgabe bewirkt, daß kontinuierlich Worte für den AD-oder DA-Wandler ausgegeben werden.

Die Kopplung erlaubt auch, einen Block der Kanäle anzusprechen und gemeinsam zu bearbeiten oder über einen Kanal mit maximal möglicher Geschwindigkeit zu arbeiten, wobei nur zu Beginn des Datenverkehrs eine Adresse übergeben werden muß. Alle nötigen Steuervorgänge laufen dann automatisch ab.

Das Einstellen dieser Betriebszustände und die Steuerung der Wandler wird durch Steuerbits erreicht, die in den Worten enthalten sind, die über die P3AS an die Steuerung übergeben werden (Bild 2, 3, 4). Hierbei wird die Grundeinstellung der Steuerung durch ein eigenes Steuerwort vorgenommen, während spezielle Steuerbits für die Wandler auch oder nur in den Worten enthalten sind, die an die Wandler adressiert werden.

Die über die P3AS angelieferten Worte werden demnach in

Steuerworte

Worte an den Digital-Analog-Umsetzer

Worte an den Analog-Digital-Umsetzer

dekodiert.

Der allgemeine Aufbau des Steuerworts und der Worte an den A/D-Umsetzer ist in Bild 3, der Aufbau der Worte an den D/A-Umsetzer in Bild 4 dargestellt. Die Bedeutung bzw. Wirkung der einzelnen Bits wird im Anhang beschrieben.

Für die Organisation des Datenverkehrs muß außerdem gefordert werden:

durch Programm steuerbar,
einfache Programmierung,
kurze Programmlaufzeiten für die Organisation des Datenverkehrs,
schnelle Übertragung der Daten,
kontinuierliches Arbeiten auch bei
schneller Übertragung großer Datenmengen.

Diese Forderungen wurden bei der vorhandenen Hardware nur zum Teil erfüllt. Abhilfe wurde geschaffen durch zusätzliche Verdrahtung von EAW-Befehlen und eine schnellere Organisation des Datenverkehrs im ORG (Punkt 3).

2.2.3. Ablauf des Datenverkehrs

Am Beispiel der fortlaufenden Wandlung beliebiger A/D-Kanäle soll das Einstellen der Betriebszustände und die Steuerung des Datenverkehrs geschildert werden (Bild 3 und 5). Es sind folgende Tätigkeiten notwendig:

Aufbereiten der Adressen und Steuerworte
in einem Puffer der P3AS

Versorgen der P3AS und P3ES

Anstoßen des Datenverkehrs

Warten auf das Ende des Datenverkehrs

Das Aufbereiten des Puffers kann im Anwenderprogramm erfolgen oder wird einem ORG-MAKRO übertragen. Dann werden die Prozeßelemente versorgt und der Datenverkehr durch einen Befehl an die Steuer- und Kontrollkopplung angestoßen. Diese Aufgabe ist im allgemeinen dem ORG übertragen.

Als erstes wird dann ein Steuerwort über die P3AS ausgegeben und damit der Zustand der Wandler eingestellt. Das hier gewünschte Steuerwort hätte die Bits FE und ZE gesetzt (Bild 3). Daraufhin wird die Adresse des ersten zu wandelnden A/D-Kanals ausgelesen, der Multiplexer eingestellt und der Wert konvertiert. Am Ende der Konversion stellt die Steuerung selbständig eine Anforderung an die P3ES/N und der Wert wird über das Informations-eingaberegister der P3ES/N in den Arbeitsspeicher geschrieben. Der eingestellte Modus bewirkt, daß fortlaufend neue Worte aus den Puffer der P3AS gelesen werden bis ein neues Steuerwort den Modus ändert. Sowohl die Konversion wie auch das STORE der Wandler kann vom Analogrechner her dabei verriegelt werden. Dadurch wird die Synchronisation zwischen den beiden Rechnern erreicht.

3. Programmorganisation

3.1. Organisationsprogramm ORG I

Für die Bearbeitung des Datenverkehrs über die Blockkanäle der Datenkopplung ist das Organisationsprogramm vorgesehen. Es hat die hier wichtigen Eigenschaften:

Simultanarbeit von bis zu 2⁴ Programmen

Bilden von Warteschlangen für die externen Elemente

Warten auf das Ende einer Externoperation

Bearbeiten der Anzeigen der externen Elemente.

Hierfür muß natürlich ein gewisser Programmieraufwand getrieben werden. An unserer Anlage wurden folgende Programmlaufzeiten für das Organisationsprogramm gemessen:

Versorgen eines Puffers des Blockkanals und Anstoßen des Datenverkehrs	ca. 530 μ sec
Bearbeiten einer Fertigmeldung	ca. 650 μ sec
Warten auf die Fertigmeldung zusätzlich	ca. 400 μ sec

Sollten z. B. 4 Digitalwerte ausgegeben werden, so dauert diese Ausgabe von der Hardware der Wandler her ca. 70 μ sec. Das bedeutet, die Programmlaufzeit zum Einleiten des Datenverkehrs ist ca. 22 mal länger als der angestoßene Datenverkehr dauert.

Nimmt man den Fall an, daß diese vier Werte in Modus wahlweiser Zugriff ausgegeben werden sollen, und zwischen zwei Ausgaben für Berechnungen im Digitalprogramm noch genau so viel Zeit benötigt wird wie für den Anstoß des Datenverkehrs, so beträgt die Zeit zwischen zwei Ausgaben für jeweils 4 Werte ca. 3,1 msec und erreicht so eine Ausgabefrequenz von ca. 1,3 kHz.

Wie dieses kleine Beispiel zeigt, ist die eben beschriebene Verfahrensweise nur dann anwendbar, wenn

- . der hybride Takt niedrig ist
(der h. T. ist die Zeit in der ein Zyklus des Hybridprogramms, bestehend aus einem Analog- und Digitalprogrammstück, abgelaufen ist)
- . sehr viele Daten auf eine Versorgung hin ausgetauscht werden, so daß die oben angegebene Programmlaufzeit in einem erträglichen Verhältnis zur Zeitdauer des angestoßenen Datentransfers steht.

Nach den Abschätzungen für die angegebenen Zeiten ist diese Versorgung brauchbar etwa bei

- . Der Ausgabe von \geq 400 Digitalwerten
 - . Der Eingabe von \geq 150 Analogwerten
 - . \geq 250 Werten
- bei der Ein-Ausgabe von Werten in
einem hybriden Loop.

Eine bessere Organisation des Datenverkehrs muß deshalb darauf abzielen, unter Verzicht auf die Vorteile des Prozeßrechner-Organisationsprogramms auf Geschwindigkeit Wert zu legen. Dies erfordert

geringen Organisationsaufwand bei der Versorgung der Prozeßelemente, möglichst unmittelbares Verwenden der Elementbefehle.

Schnellere Bearbeitung der Fertigmeldungen für den Datenverkehr

Wahlweises Unterdrücken der Fertigmeldungen

Die beiden ersten Punkte wurden durch Softwaremaßnahmen erreicht, der letzte durch eine Hardwareänderung.

Es wurden zwei Organisationsformen entwickelt:

3.2. Schnelle Wechsellpufferorganisation

Hierzu werden die Parameter ohne zusätzlichen Organisationsaufwand unmittelbar an die Prozeßelemente übergeben und der Datenverkehr angestoßen. Die Bearbeitung der Fertigmeldung wird vorrangig behandelt und aus der Standard-Bearbeitung durch das ORG ausgeblendet.

Die Programmlaufzeiten hierbei sind

Versorgen der Geräte und Anstoß des Datenverkehrs	12 μ sec
Bearbeiten der Fertigmeldung	63 μ sec
Warten auf die Fertigmeldung	$(7,5 + n \cdot 40,5)$ μ sec

Wobei $n=0$ (1) N die Anzahl der Abfragezyklen für die Fertigmeldung ist (N beliebig). Wenn auf die Fertigmeldung gewartet werden soll, so findet dieses Warten unmittelbar nach der Versorgung statt.

Bei langdauernden Datenverkehren kann der Wechsellpufferbetrieb dazu genützt werden, die Wartezeit für das laufende Programm auszunutzen. Zu diesem Zweck wird ein Puffer versorgt, seine Fertigmeldung ausgeblendet und der Datenverkehr angestoßen. Dann wird im Programm weitergearbeitet. Vor der Verarbeitung der Daten wird der zweite Puffer versorgt und der abschließende Datenverkehr durchgeführt. Danach wird gewartet, bis die Fertigmeldung für den gesamten Datenverkehr eintrifft. Die erste Wartezeit kann auch für ein anderes Programm verwendet werden, wenn es nicht die Datenkopplung anspricht.

3.3. Einpufferorganisation

Bei dieser Organisation wird nicht mehr im Wechsellpufferbetrieb gearbeitet. Der Datenverkehr wird direkt im Anwenderprogramm ohne Ansprechen des Organisationsprogramms bearbeitet. Die Realisierung ist nur mit dem oben erwähnten Hardwarezusatz möglich. Die Fertigmeldungen brauchen deshalb nicht mehr wie unter 3.1 und 3.2. bearbeitet werden. Stattdessen werden die Prozeßelemente abweisend, d.h., nach der Versorgung durch das Programm kann der Blockkanal erst wieder angesprochen werden, nachdem der Datenverkehr vollständig abgeschlossen ist. Dieser Zustand kann mit einem Befehl an beliebiger Stelle im Programm geprüft werden. Die Zeit, bis der Blockkanal wieder neu angesprochen werden kann, läßt sich für das laufende oder andere Programme mit der gleichen Einschränkung wie unter 3.2 ausnutzen.

Die Programmlaufzeiten sind:

Versorgung des Blockkanals und	
Anstoßen des Datenverkehrs	12 μ sec
Warten auf die Fertigmeldung	4,5 (1 + n) μ sec

wobei $n = 0$ (1) N wieder die Anzahl der Abfragezyklen ist.

Diese Art der Organisation eignet sich ganz besonders für den Austausch von Einzelwerten.

Hier weiß man genau wie lange der Datenverkehr laufen wird. Es braucht also nicht auf eine Fertigmeldung gewartet werden. Die Programmlaufzeit reduziert sich dabei auf 1/100 gegenüber 3.1.

3.4. Beispiel

Es sollen 5 Digitalwerte ausgegeben und 5 Analogwerte eingelesen werden. Die Steuerworte, Werte und Adressen seien bereits für die Ausgabe vorbereitet. Der Verkehr über den Datenkanal läuft dann ca. 180 μ sec. Die Zeit, die vom Einleiten des Datenverkehrs im Programm bis zur Verarbeitung der Daten vergeht, (vgl. Bild 5) ist für die

Standard-Organisation	ca. 2700 μ sec
Schnelle Wechselpufferorg.	ca. 300 μ sec
Einpufferorganisation	ca. 245 μ sec

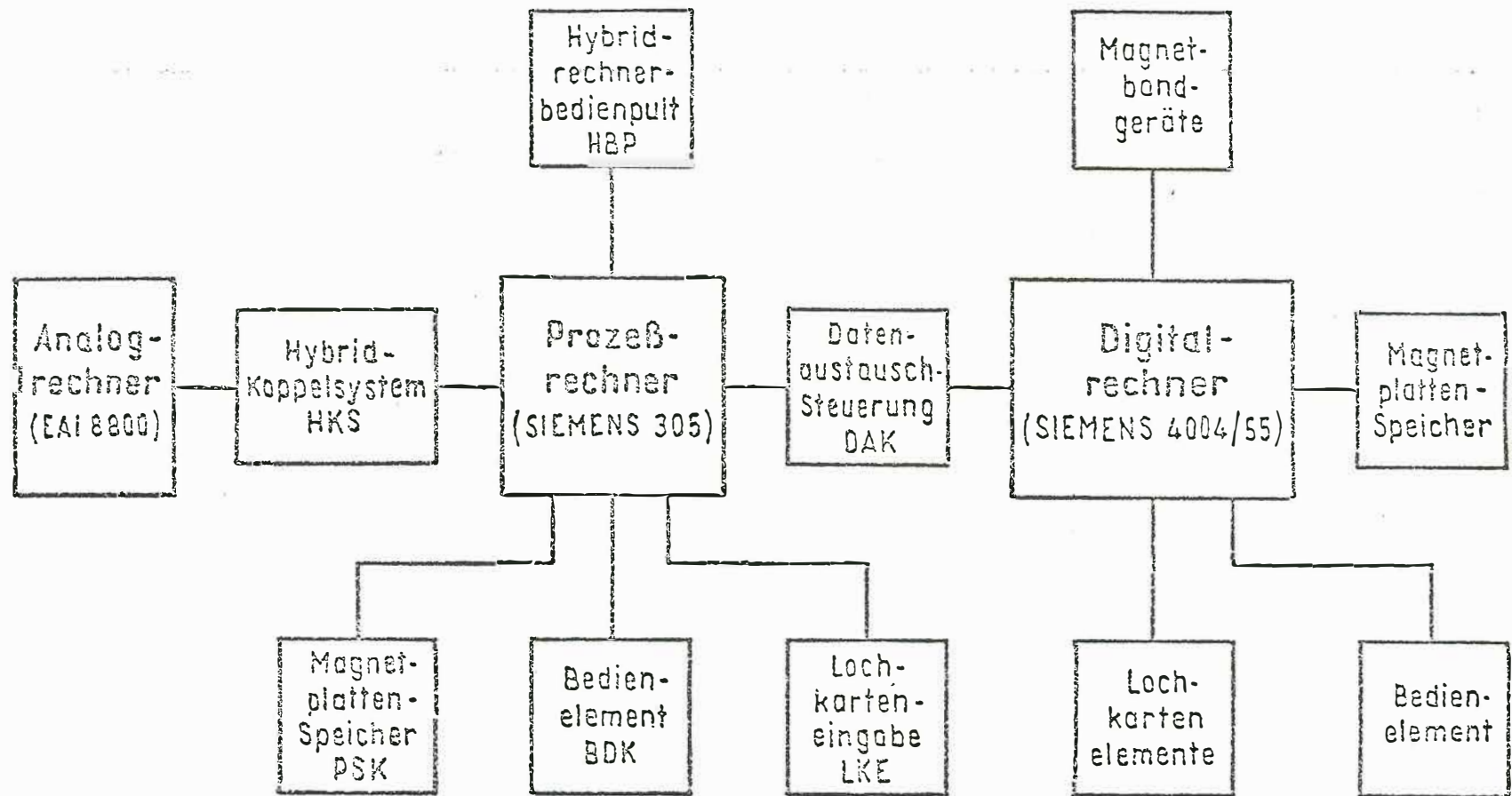
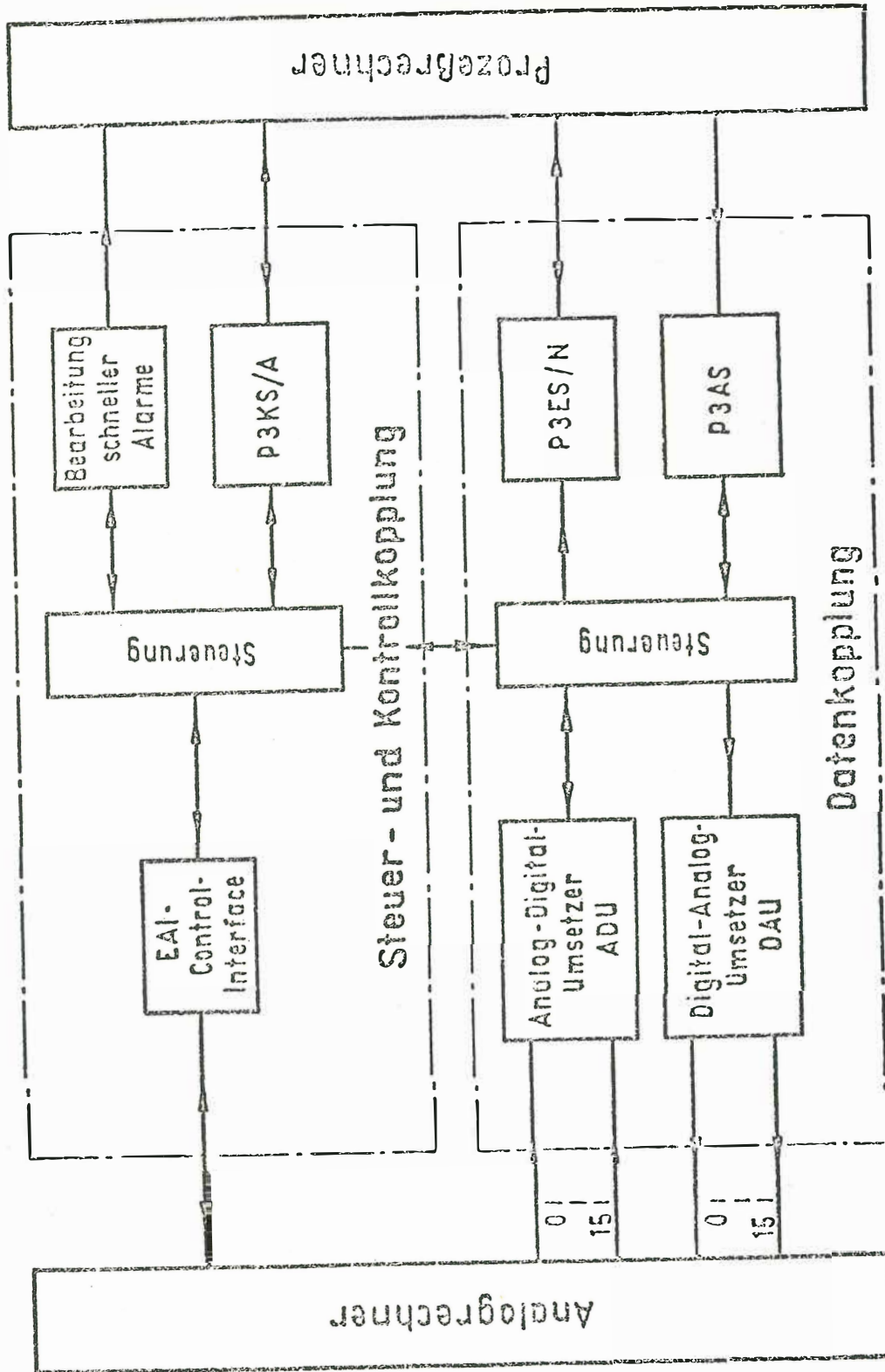


Bild 1

Aufbau des Hybridrechenzentrums am IMR



Aufbau des Hybridkopplensystems HKS

Bild 2

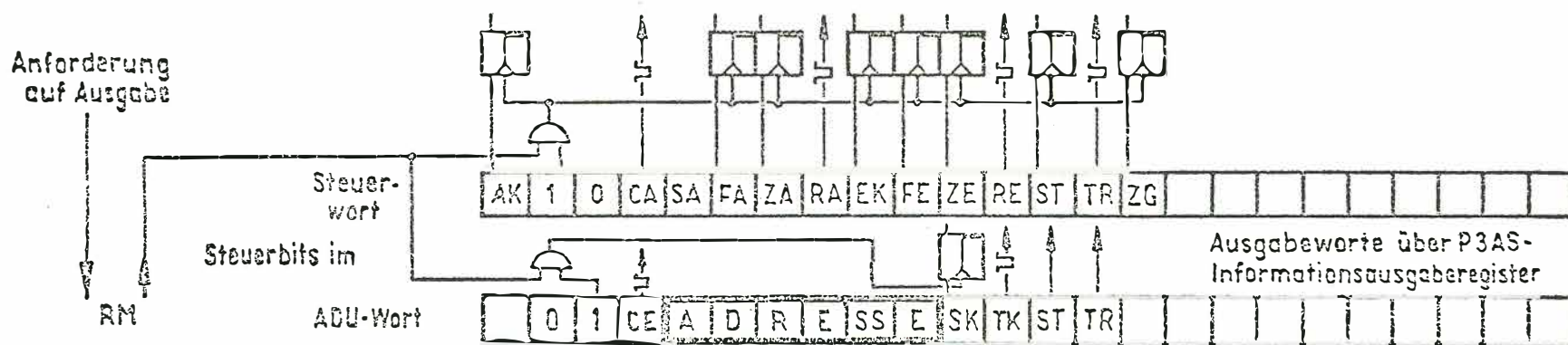
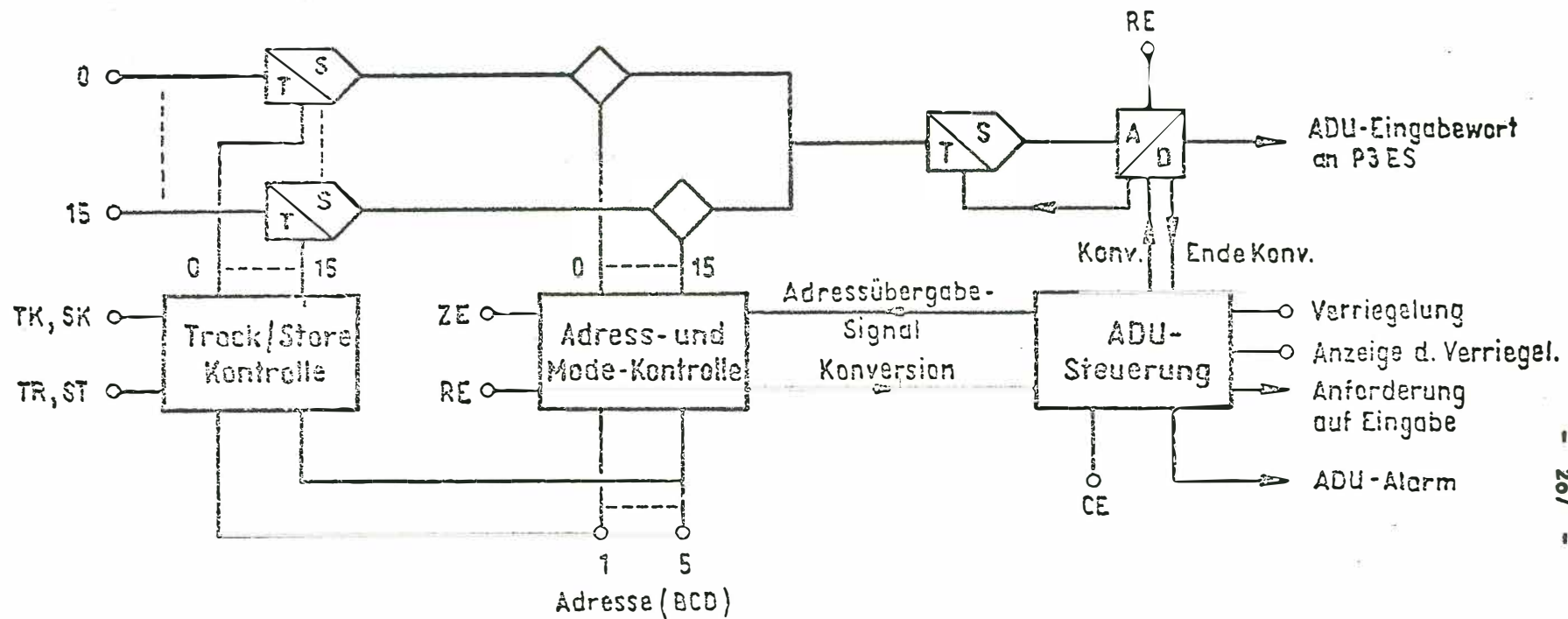
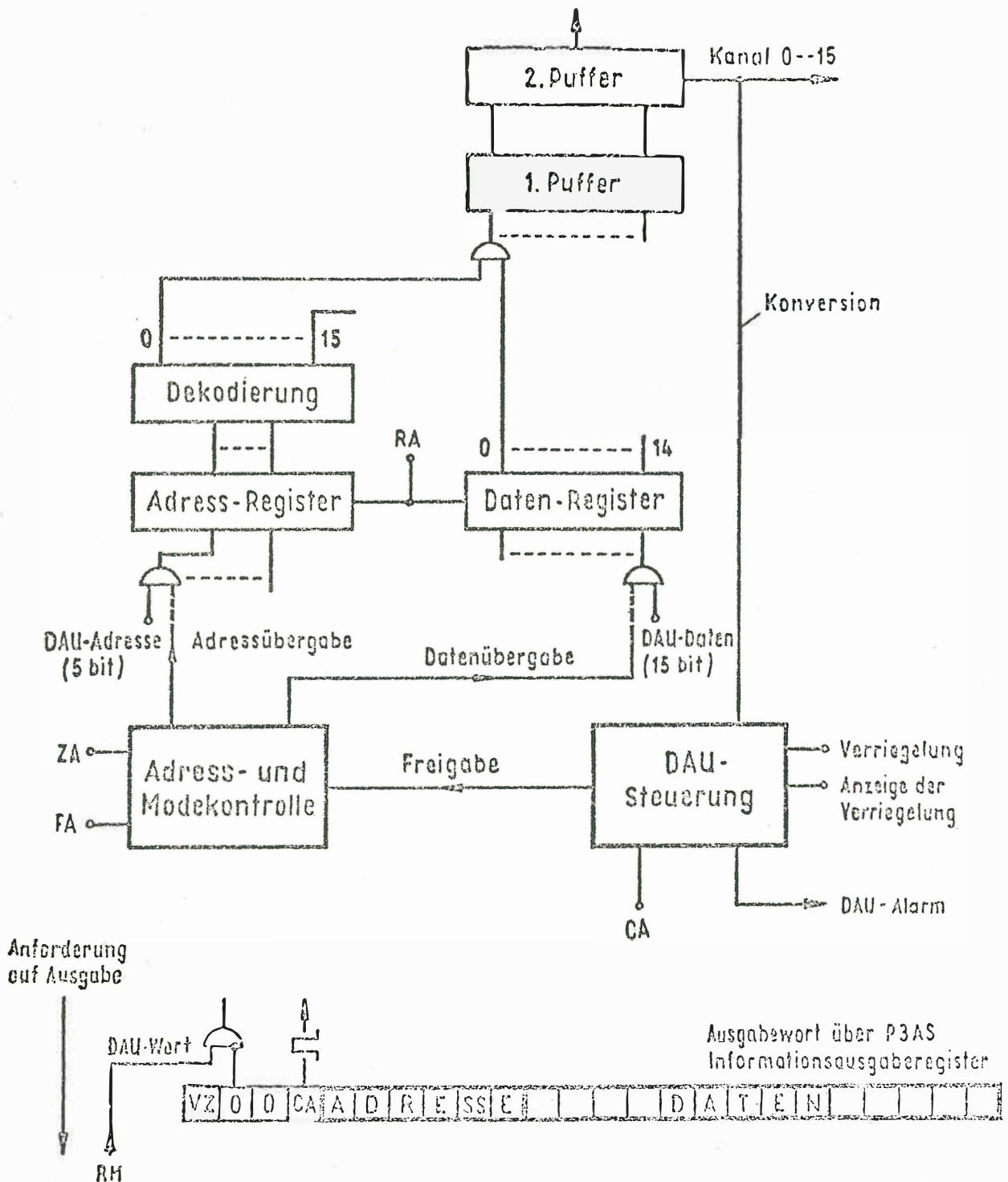
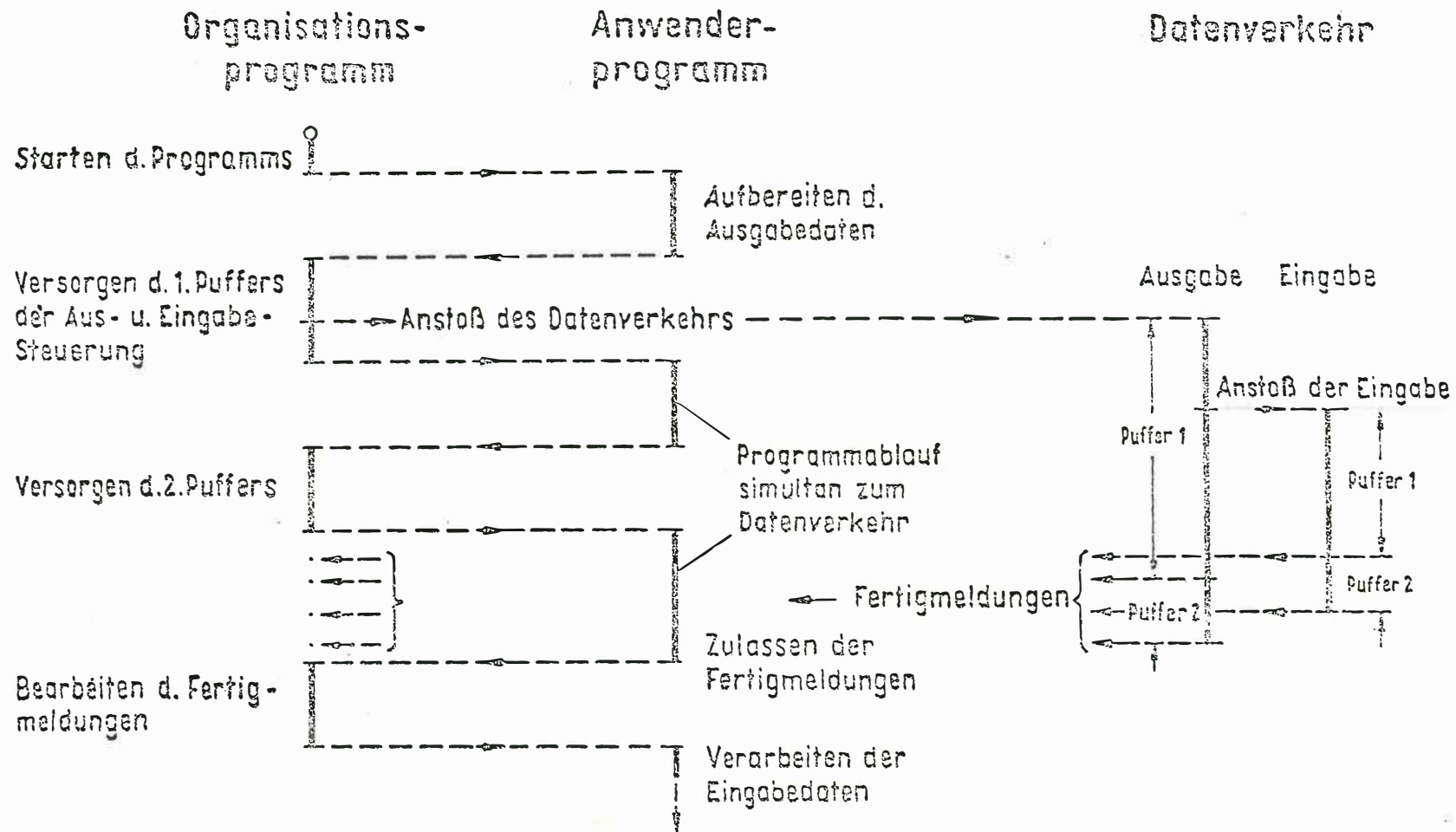


Bild 3 Vereinfachtes Schaltbild der Analog - Digital - Wandlung



Vereinfachtes Schaltbild der Digital-Analog-Wandlung
Bild 4



Ablauf des Datenverkehrs über die Datenkopplung im allgemeinen Fall

(nicht maßstäblich)

Bild 5

Anhang

Zusammenstellung der verwendeten Schaltzeichen und Abkürzungen (soweit nicht in den Bildern enthalten)



Track- and Store - Verstärker



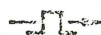
Digital gesteuerter Schalter zum Durch-
schalten einer analogen Größe



Analog-Digital-Umsetzer



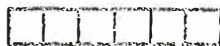
Bistabile Kippstufe



Übergabe eines Steuerimpulses



Und-Gatter
(· = Negation)



digitales Speicherregister

STEUERBITS:

AK	Fortlaufende Ausgabe über denselben Kanal
CA	Konvertierung des DAU einleiten
SA	Das nächste Wort aus dem Arbeitsspeicher holen

FA	Sobald der DAU frei, d.h. die Adresse übergeben oder die Konversion abgeschlossen ist, wird das nächste Wort aus dem Arbeitsspeicher geholt
ZA	Random/Sequential-Umschaltung für DAU
RA	Rücksetzen des DAU, die wichtigen Register des DAU werden mit Null geladen
EK	Fortlaufende Eingabe über denselben Kanal
FE	Sobald der ADU die Konversion beendet hat, wird das nächste Wort aus dem Arbeitsspeicher geholt
ZE	Random/Sequential-Umschaltung für ADU
RE	Rücksetzen des ADU
ST	Alle ADU-Kanäle auf STORE schalten
TR	Alle ADU-Kanäle auf TRACK schalten
ZG	Zeitglied für ADU-Alarm-Überwachung ausschalten
CE	Konvertierung des ADU einleiten
SK	Vier ADU-Kanäle auf STORE schalten
TK	Vier ADU-Kanäle auf TRACK schalten
VZ	Vorzeichen der DAU-Daten

Daten der Wandler

D/A - Wandler

Bereich der analogen Ausgangs-
spannung

: $\pm 100 \text{ V}$

Anzahl der Bits

: $14 + 1 \text{ Vorzeichen}$

Zeit für Übergabe eines
Digitalwertes

: $5 \mu \text{ sec}$

Einschwingzeit der Ausgangs-
verstärker

: $30 \mu \text{ sec (0,01 \%)}$

Anstiegszeit 10 % - 90 % bei 200 V : $6 \mu \text{ sec}$

Genauigkeit bezogen auf

Vollanschlag

: $\pm 0,01 \%$

Codierfehler

: $\pm 1/2 \text{ LSB}$

(LSB = least significant bit)

A/D - Wandler,

Vorgeschalteter T/S in jedem Kanal,

Bandbreite des Eingangsfilters : 300 kHz (- 3db)

Eingangsspannung : ± 100 V bis 50 kHz (- 3db)

Einschwingzeiten:

Sprung bei Track : 7 μ sec
bei 0,01 %

T/S : 10 μ sec

Aperturzeit : 50 nsec

Spannungsabfall des T/S : $\pm 2,5$ μ V/ μ sec

Multiplexer:

Umsetzzeit für 15 bit : 22,5 μ sec

Genauigkeit bezogen auf
Vollanschlag : $\pm 0,01$ %

T/S

Genauigkeit : $\pm 0,04$ %

Einschwingzeit : 5 μ sec auf 0,01 %

Aperturzeit : 100 nsec

Storezeit : 50 μ sec für Fehler $\leq 0,01$ %

Die Gesamtgenauigkeit der A/D-Wandler wird vom Hersteller mit 0,0345 % angegeben.

W. Ehrenberger
Technische Hochschule
München

"Alar**m**bear**h**eitung in einer
3o5 und Simult**a**narbeit in
höheren Sprachen"

**Alarmbearbeitung in einer 305 und
Simultanmarbeit in höheren Sprachen**

W. Ehrenberger

12. 8. 70

Zusammenfassung:

Die folgenden Ausführungen beziehen sich auf die Hybridanlage am Institut für Meß- und Regeltechnik an der TH-München. Für manche Hybrid-Anwendungen ist der über das Org 300 abgewinkelte Externverkehr zu langsam. Eine zusätzlich eingebaute Alarmbearbeitung ermöglicht eine rasche Reaktion auf Anrufe an die Zentraleinheit. Um aus Alarmprogrammen Externverkehr abwickeln zu können, muß das Org etwas verändert werden. Die Prozeßelemente P3AS und P3ES/N sind mit einem Zusatz auszurüsten. In höheren Sprachen ist zur Zeit keine Simultanmarbeit zwischen einem Programm und dessen eigener Ein-Ausgabe möglich. Codeprozeduren können eine solche Simultanmarbeit gestatten, indem sie den Ein-Ausgabeaufruf von seinem EXWA trennen und in einem Vorlauf Org-Makros absetzen.

1. Alarmbearbeitung

1.1 Problemstellung

Nehmen wir an, wir wollen für den Analogrechner einen z-Wert zu einem vorgegebenen Paar von x-y-Werten bestimmen.

Bild 1a, b

Wir müssen folgendes tun:

1. Ein Wertepaar xy einlesen
2. den z-Wert durch Interpolation ermitteln
3. den z-Wert ausgeben.

Dies soll sehr schnell vor sich gehen, um den Zeitfehler klein zu halten.

Der Vorgang soll auch unabhängig von einem anderen Hybridprogramm in der 305 ablaufen können.

Wie man leicht einsieht, ist die durch das Org 300 geleistete Organisationsarbeit dazu zu zeitraubend. Ebenso ist die selbst organisierte BAP-Bearbeitung zu langsam, weil immer auf eine BAP-unterbrechbare Stelle gewartet werden muß und weil das Org nicht BAP unterbrechbar ist.

1.2 Einbau einer Alarmunterbrechungsmöglichkeit in eine 305

Wir haben deshalb das Steuerwerk etwas erweitert, um ein rasches Reagieren der Zentraleinheit zu ermöglichen. Diese Alarmbearbeitung hat folgende Eigenschaften:

1. Sie erlaubt eine Programmunterbrechung nach jedem Befehl, der kein Elementbefehl ist.

2. Sie setzt das Programm an einer durch Schalter einstellbaren Adresse fort.
3. Sie stellt nicht in Zelle 0 sicher, sondern in einer ebenfalls einstellbaren Zelle.
4. Sicher gestellt wird nicht nur der Befehlszähler, sondern auch das Elementauswahlregister, die Überlaufbits bei der Akkus und beide Bits 0 der Akkus (Bild 2)
5. Die Priorität dieser Unterbrechung liegt zwischen der MAP u.d. BAP.
6. Die MAP bleibt in ihrer bisherigen Funktion voll erhalten.
7. Die Alarmbearbeitung verhindert eine BAP solange irgendein Alarm ansteht.

Um diese Punkte zu erfüllen, mußte das Steuerwerk mit Zubauten versehen werden (Bild 3).

2 Platten und 1 Stecker waren neu einzufügen.
An 5 Einbauplätzen wurden Veränderungen nötig.

Im Rechenwerk wurden 2 Einbauplätze neu belegt.

1.3 Reaktionsgeschwindigkeit der Hardware

Die Reaktionsgeschwindigkeit wurde durch den Einbau bedeutend erhöht. Die maximale Reaktionszeit der Hardware liegt bei der Dauer eines Befehls, also unter 40µs. Ist der Befehl MAP-unterbrechbar gewesen, kommt u.U. noch eine PUN-Laufzeit = 4,5µs dazu. Bei kurzen Befehlen, dringt der Alarm bereits nach 1,5µs durch.

Selbstverständlich ist auch das Org unterbrechbar.

1.4 Beginn und Ende von Alarmprogrammen

Nachdem die Unterbrechung hardwaremäßig stattgefunden hat, müssen noch die Registerinhalte sichergestellt werden. Man hat also die Befehle zu durchlaufen:

	SPR	()	Einsprung über die Sprungleiste
ALPROGR	TAS'	SI	Sicherstellung des RA
	TEP'	1183	Holen der Rücksprunginformation
	EA5	Zulassen von Alarmen höherer Priorität
	TAS'	SI+1	Sicherstellen der Rücksprunginf.
	TAS	SI+2	" des LA
	TAX	SI+3	" des ER

Mit dem EA5 kann man Unterbrechungen höherer Priorität zulassen.

Am Ende der dann zu rechnenden Unterbrechungsroutine müssen aus dem Bereich SIZE die Registerinhalte wieder restauriert werden, die Bits 0, die Überlaufbits und das Elementauswahlregister in den alten Zustand gebracht werden; dann erst kann man zurückspringen an die Stelle, an der man unterbrochen hatte.

Der Schluß eines Alarmprogramms hat die Befehle:

EA3	Abtaster freigeben, auch Alarme niedrigerer Priorität werden möglich
EPR	{ }	Restaurieren des Elementauswahlregisters
SPR	{ SI+1 }	Rücksprung in das unterbrochene Programm
SPR	{ SI+1 }	

Nachdem hinter Elementbefehlen keine Unterbrechung durch Alarm möglich ist, ist diese Befehlsfolge vom Wiederzulassen von Alarmen bis zum Rücksprung nicht unterbrechbar.

Die gesamte Zeit für das Wiederherstellen der Registerinhalte usw am Alarmende beträgt 69µs.

1.5 Externverkehr aus Alarmprogrammen

Aus einer in der vorhin beschriebenen Weise eingeleiteten Alarmroutine kann man natürlich keinen Externverkehr machen zu Elementen, die über das Org angesprochen werden. Das Org ist ja u.U. selbst unterbrochen worden! Wenn man das Element über EA-, EV-Befehle versorgt, so gerät man in die Gefahr ein bereits versorgtes Element neu zu versorgen und die alte Versorgung zu zerstören, löst aber immer eine Übergabe von PU-Bits und BAP aus, was dann wieder zu einer Org-Bearbeitung führt, Zeit kostet und im allgemeinen auf eine UAP führt.

Man muß also auch im Org selbst noch etwas tun, nämlich verhindern, daß die eigentliche Versorgung des Elements unterbrochen werden kann. Dies erreicht man durch Hintereinandersetzen der EA-EV-Befehle; denn Elementbefehle haben wir ja ununterbrechbar gemacht.

Im Element selbst müssen wir auch etwas verändern. Wir müssen auf Anfrage das Signal "Abweisend" geben können und wir müssen dieses Signal nach Ende des jeweiligen Puffers wieder zurücknehmen. Wir müssen ferner in der Lage sein, die BAP zu verhindern. Diese Forderungen füllt ein Zusatzeinbau mit den Eigenschaften:

1. Ein EA1 bringt ABW, wenn das Element versorgt ist.
2. Ein EA2 verhindert die BAP am Pufferende.
3. ABW verschwindet, sobald beide Puffer abgelaufen sind.

Vor jeder Elementversorgung muß mit einem EPR festgestellt werden, ob das Element zugänglich ist. Dies wurde ebenfalls in den die P3-Elemente versorgenden Org-Teil aufgenommen.

Im Alarmprogramm muß der auf Prosa-Ebene programmierende Anwender selbst auf „Abweisend“ prüfen. Mit einem EA1-Befehl kann er das Abweisend hervorrufen, falls eine Org-Versorgung vorher stattgefunden hatte, Der Anwender kann

nach Wunsch aber auch eine bereits gegebene Versorgung überschreiben.

1.6 Alarmbearbeitung in höheren Sprachen

Wenn man nun in einer höheren Sprache mit Alarmen arbeiten möchte, dann muß man für jeden der verschiedenen Alarme je 1 Codeprozedur vorsehen, die den Alarmbeginn und das Alarmende ausführt. Für alle Alarme nur 1 oder 2 Codeprozeduren zu verwenden ist bei einer 305 nur auf Kosten der Rechenzeit oder größeren Hardwareaufwands möglich. In einem Vorlauf muß die richtige Einsprungadresse in das Alarmprogramm eingetragen werden. Alarmprogramme in einer höheren Sprache sind als Subroutinen der Coderoutinen möglich. Bei Verwendung von Standardfunktionen ist Vorsicht geboten. Ein-Ausgaben der üblichen Art sind nicht erlaubt. Bild 4.

Nachdem jede BAP-Bearbeitung per Hardware verhindert wird, solange irgend ein Alarm ansteht, entstehen keine Schwierigkeiten durch die BAP-unterbrechbaren Stellen eines durch einen Compiler erzeugten Programmstücks.

2. Simultanarbeit in höheren Sprachen

2.1 Problem

Bei der Digitalisierung von Analogbändern tritt folgende Aufgabenstellung auf:

Während ein Puffer über ein P3-Element gefüllt wird, muß der andere Puffer versorgt und weggeschafft werden. Der Datenfluß vom Analogrechner zum Digitalrechner muß kontinuierlich sein; er darf nicht durch Organisationsrechenzeiten unterbrochen werden. Bild 5.

Diese Aufgabe ist in Prosa leicht lösbar.

In einer höheren Sprache aber bekommt man Schwierigkeiten. Standardfunktionen in höheren Sprachen pflegen gleich hinter den Ein-Ausgabe-Aufruf den Warteaufruf zu enthalten. Es ist keine Simultanarbeit zwischen Externverkehr und dem betreffenden Anwenderprogramm möglich. Probleme von der Art des aufgeworfenen können nur durch mehrere simultan arbeitende Programme in einer anwenderorientierten Sprache gelöst werden.

In dem angesprochen Fall könnte man zudem wünschen, sich im Digitalrechner zur Zwischenpufferung eines Ringspeichers zu bedienen. Ein solcher Speicher, der aus mehreren Einzelpuffern besteht und von einer Seite ständig gefüllt und von der anderen geleert wird, ist in einer höheren Sprache ebenfalls nicht programmierbar.

Schließlich möchte man je nach Schärfe der Geschwindigkeitsforderungen die Möglichkeit haben, die eingelesenen Daten sofort umzucodieren oder sie erst nach einer Abspeicherung auf einem Massenspeicher zur Verarbeitung aufzubereiten.

2.2 Forderungen an die höhere Sprache

Aus dem vorhergehenden Punkt lassen sich für eine anwenderorientierte Sprache für einen Hybridrechner die Wünsche angeben:

1. Die Ein-Ausgabeaufrufe müssen von den Warteaufrufen getrennt werden.
2. Warteschlangenbildung ist zuzulassen. Dazu müssen so viele Org-Makros geschaffen werden können, wie eine eventuelle Warteschlange innerhalb eines Programms Glieder haben soll.
3. Die Umcodierung soll von der Ein/Ausgabe unabhängig sein.

Zur Erreichung dieser Ziele wurden Codeprozeduren erstellt.

2.3 Schema der Codeprozeduren

Die Ein/Ausgabeaufrufe zur Simultanarbeit müssen sich zunächst einmal die benötigten Org-Makros schaffen.

Dazu bedient man sich eines Vorlaufs. In diesem Vorlauf trägt die Codeprozedur den gewünschten Org-Makro zusammen mit seinem Exwa in einem vorher zu deklarierenden Feld ein. Nach der Zahl dieser Felder und Eintragungen bestimmt sich die Maximallänge einer möglichen Warteschlange.

Im Rechenlauf wird der abgesetzte Aufruf ausgeführt. Dabei gibt es zwei Möglichkeiten:

1. Die Parameter, die das Externe Element betreffen, werden verändert.
2. Die das EXE betreffenden Parameter bleiben unverändert.

Ein Wartelauf schließlich erlaubt das Geben des Exwas. Bild 6.

2.4 Ein Beispiel

Nehmen wir die Codeprozedur TYDT, Type Data. Sie wird deklariert:

```
TYDT(NR, ADR, DIST, PLA).,  
'VALUE' NR, DIST.,  
'INTEGER' NR, DIST.,  
'INTEGER' 'ARRAY! ADR, PLA.,  
'CODE'.,
```

ADR bezeichnet das bereits umcodierte Feld aus dem ab der Entfernung DIST vom Beginn ausgegeben werden soll. PLA muß auch deklariert worden sein als:

'INTEGER' 'ARRAY' PLA (/ 1.. 10/),,

Ist NR = 0, so wird in PLA eingetragen:

PLA	UNT	(5)
	003	19
	NOP	0
	NOP	ADR + DIST
	NOP	0
	SPR	(RUECK)
	UNT	(5)
	001	17
	NOP	PLA
	SPR	(RUECK)

Ist NR = 2, wird auf PLA (/1/) gesprungen. Ist NR = 3 oder größer, wird auf PLA (/7/) gesprungen. Ist NR = 1, wird entsprechend der neuen aktuellen Parameter ADR u. DIST die entsprechende Eintragung in PLA (/4/) korrigiert und anschließend auf PLA (/1/) gesprungen.

2.5 Vorhandene Aufrufe

In der eben geschilderten Weise wurden folgende Aufrufe für Algol zugänglich gemacht.

RDCD (NR, ADR, DIST, N, PLA),,
Read cards, Lochkarteneingabe;

TYDT (NR, ADR,,DIST, PLA),,
Type Data , Blattschreiberausgabe von Daten;

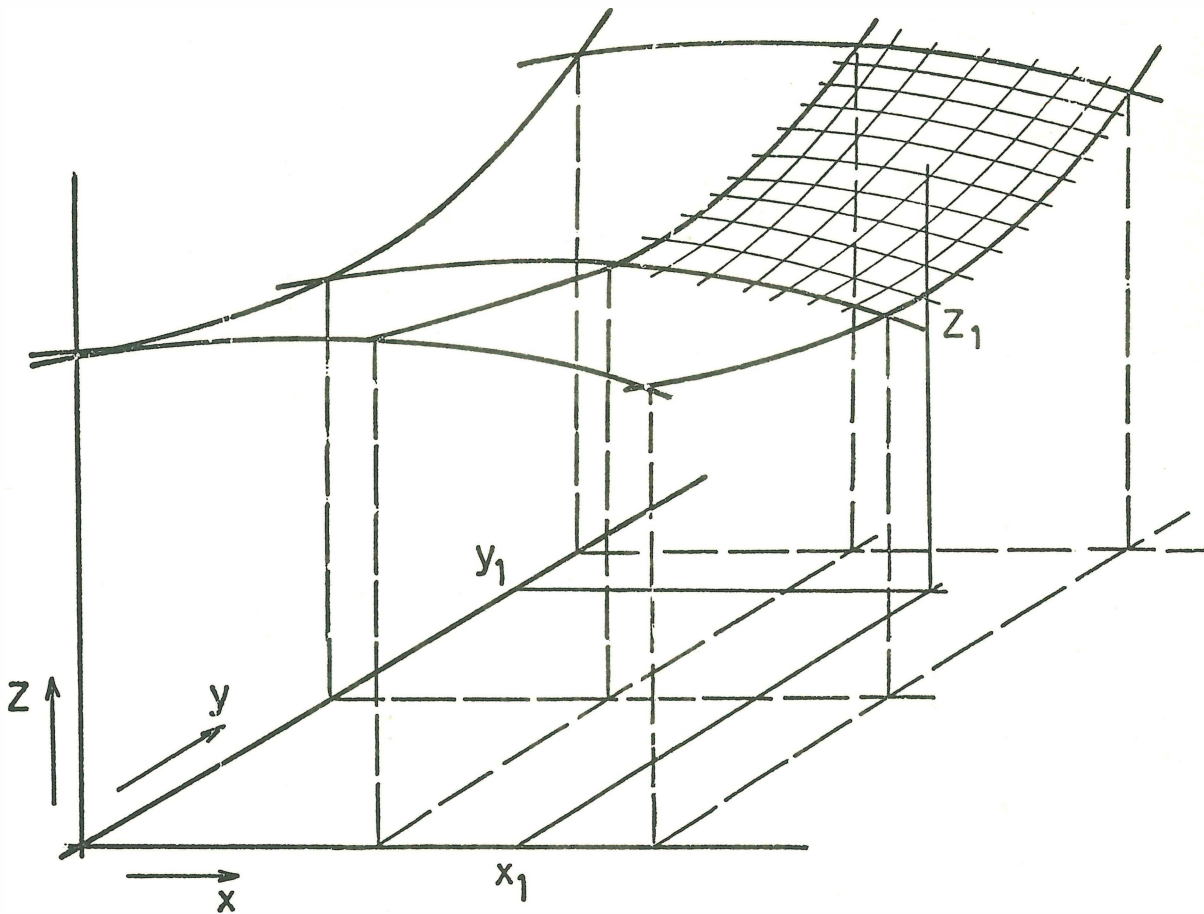
TYTX (NR, TEXT, PLA),,
Type Text , Blattschreiberausgabe von Text;

PRIN (NR,ADR, DIST, LINE, PLA),,
Print , Schnelldruckerausgabe

TAPE (NR, MODE, TNR, ADR, DIST, N, PLA).,
Magnetband Ein-Ausgabe, Bandsteuerung;

COPL (NR, MODE, ADR, N, PLA).,
Copplung zweier Digitalrechner;

DAAD (NR, BLOCK, DISTBL, NBLOCK, ADR, DIST, N, REG, PLA).,
Digital-Analog-Analog-Digital, simultane Ein-Aus-
gabe zum Analogrechner.



Zu einem Wertepaar (x, y) wird der
Z - Wert errechnet

Bild 1a

Räumliche Darstellung der Interpolation eines
Wertes z_1 anhand eines vorgegebenen Wertepaars
 x_1, y_1 , wenn ein festes Raster von Werten x, y, z
gespeichert ist

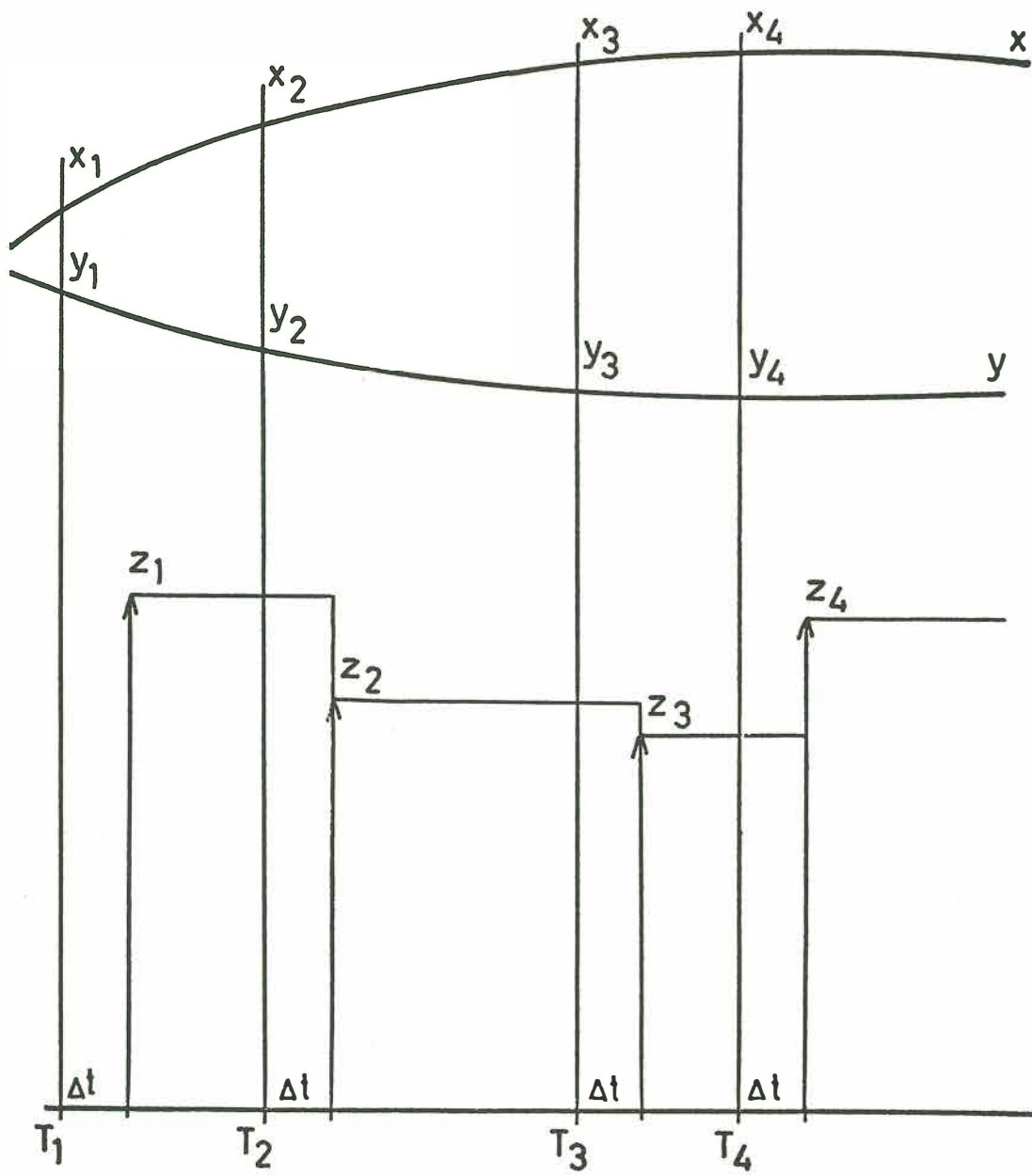


Bild 1b
Zeitliche Darstellung der Interpolation von z -Werten
aufgrund einzulesender x, y -Werte

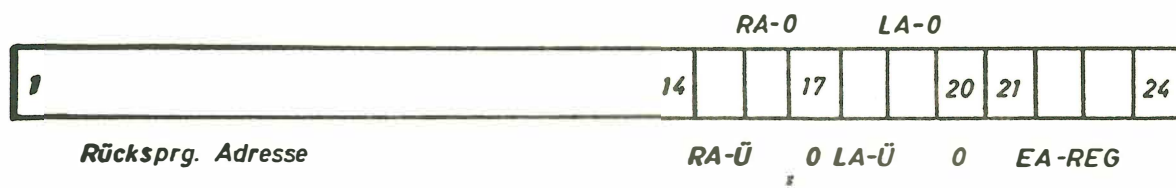


Bild 2
Rücksprunginformation

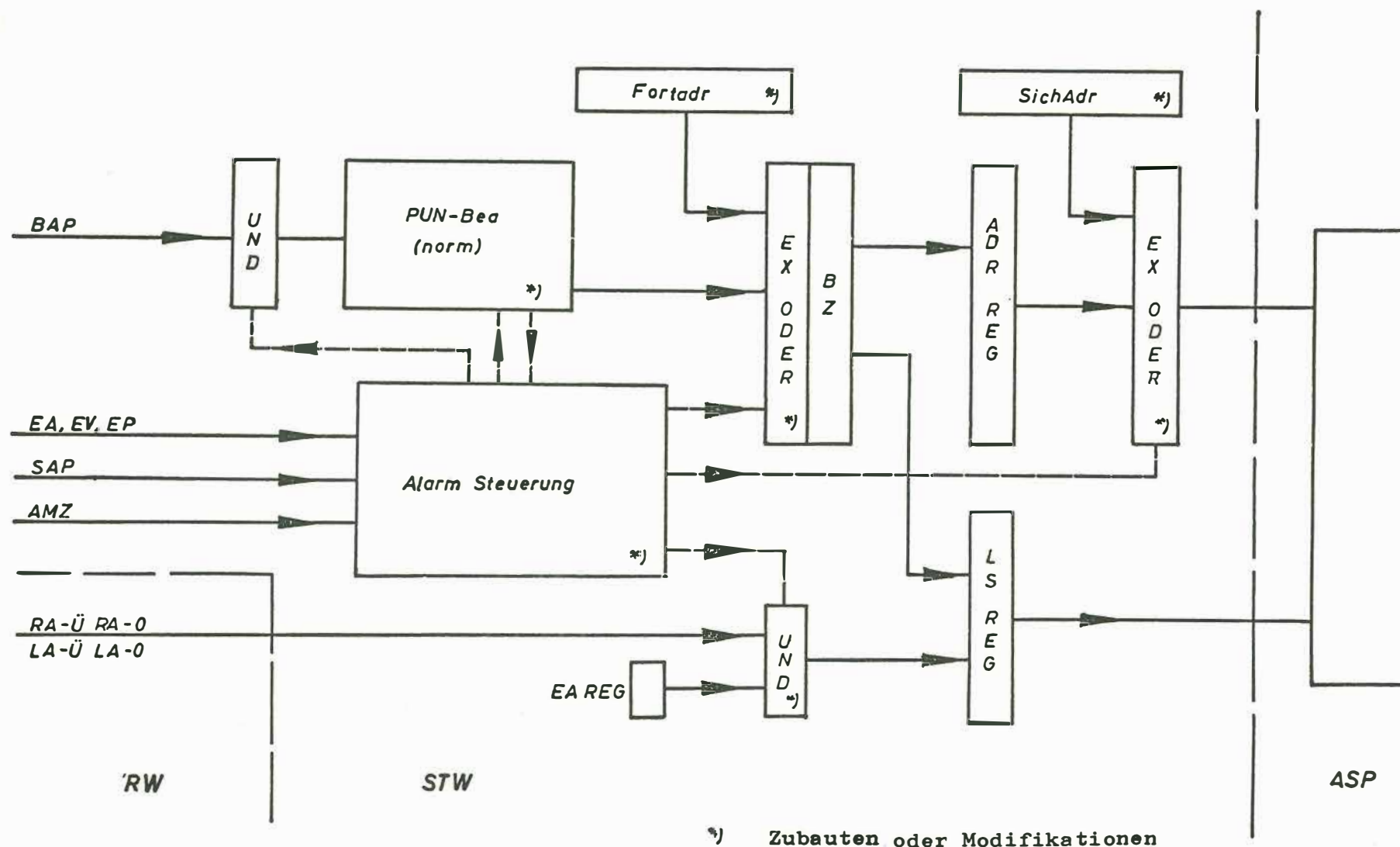


Bild 3
Veräinfachtes Blockbild des Steuerwerks mit den Erweiterungen

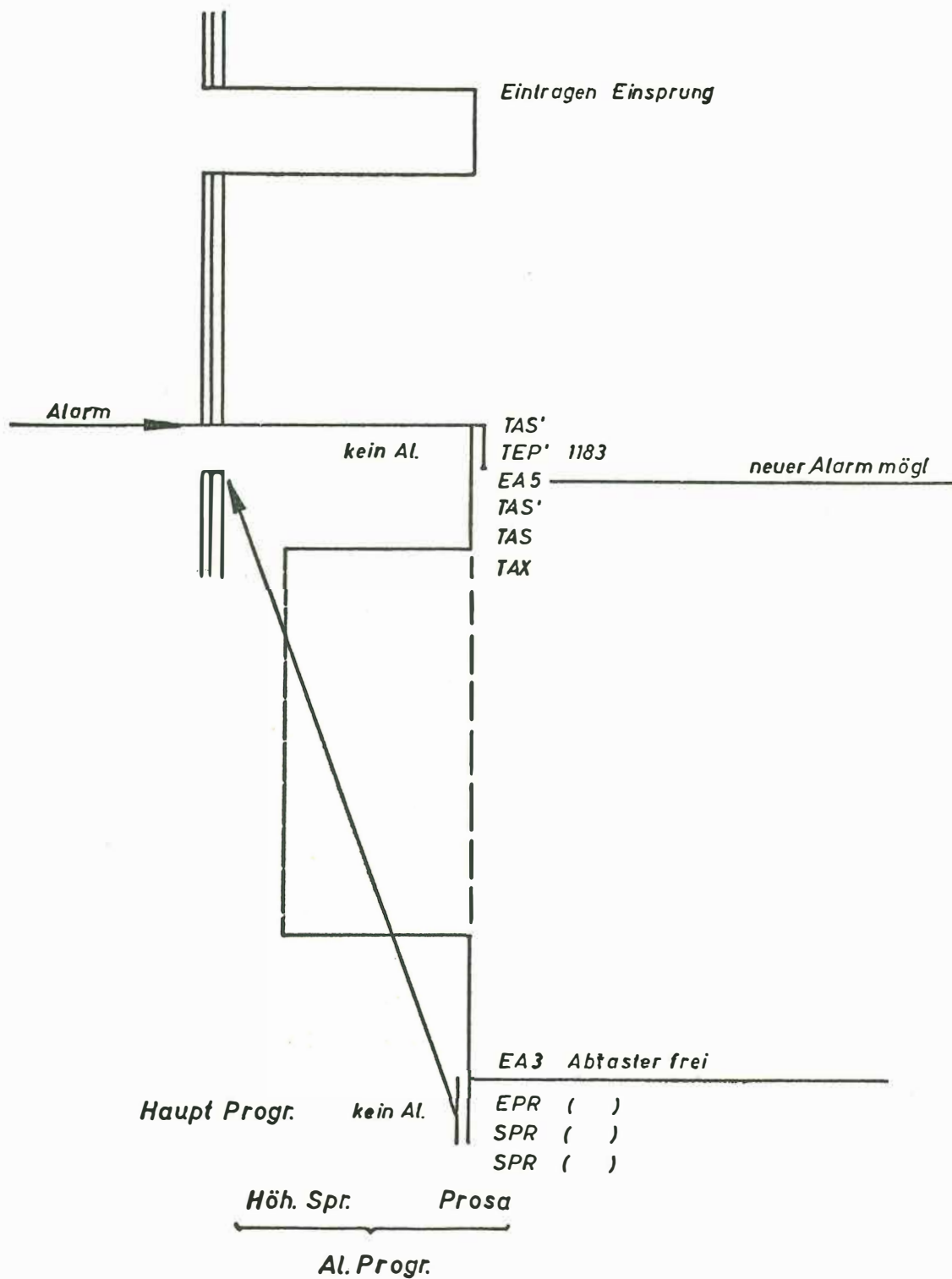


Bild 4
Alarmprogramm in einer anwendernahen Sprache

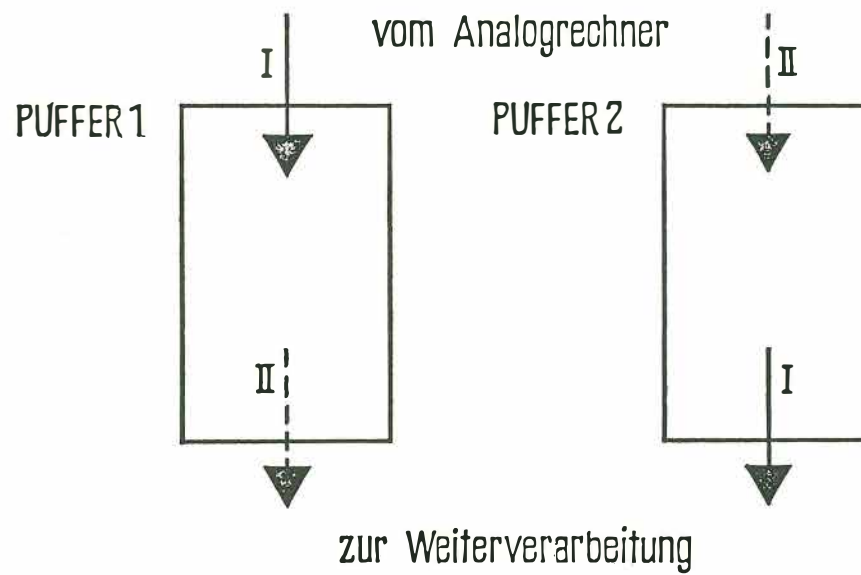


Bild 5

Schema des kontinuierlichen Datenverkehrs über
einen Rechner

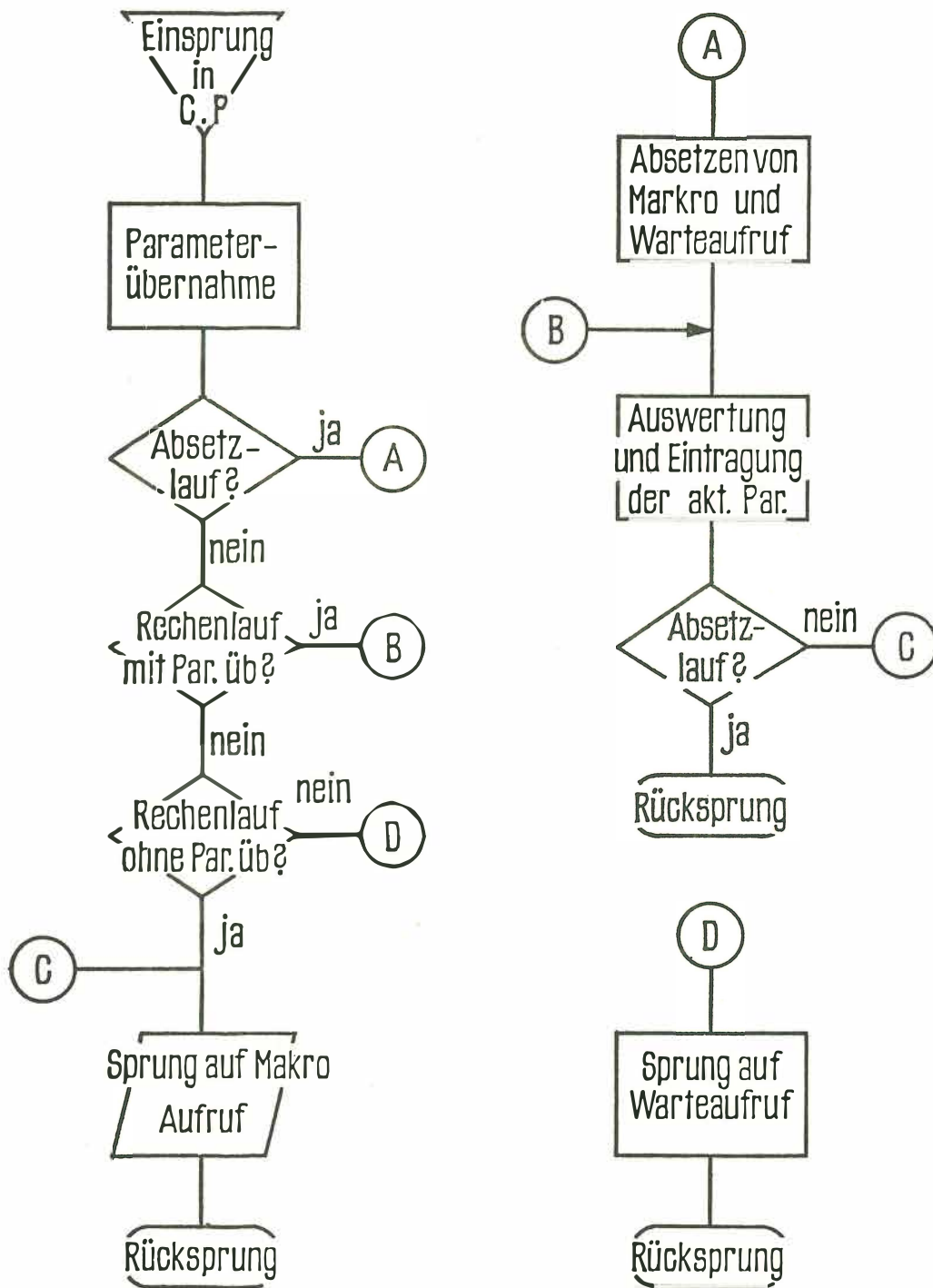


Bild 6

Schema einer Codeprozedur zur Simultanarbeit von Externverkehr und auslösendem Programm in einer anwendungsnahen Sprache

