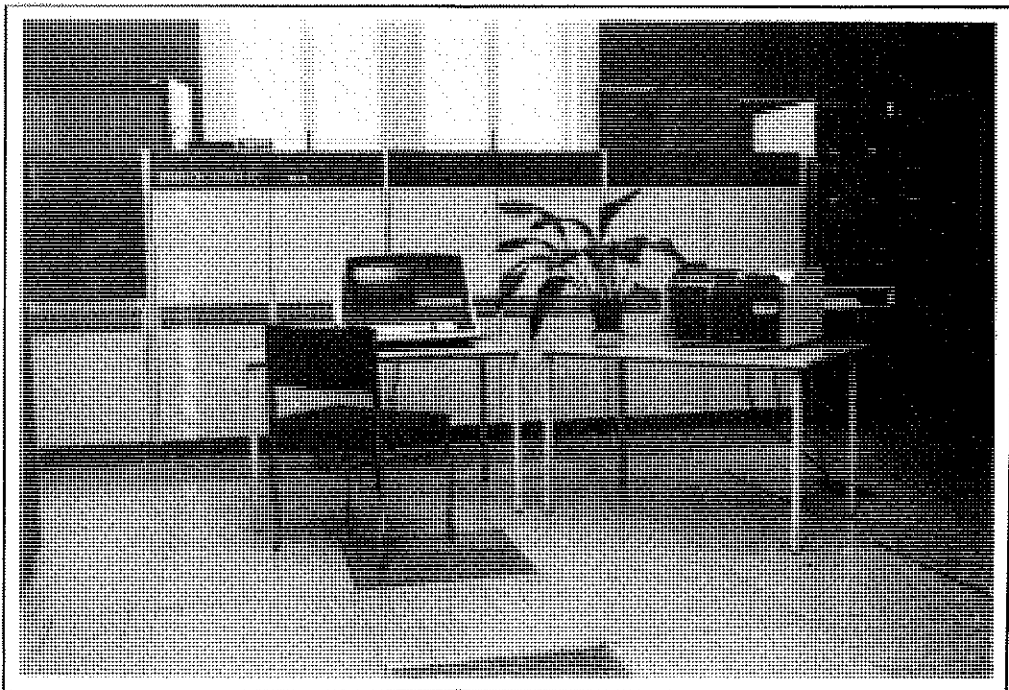

FAKEDBACK

HEFT 5

JUNI 1978



DECsystem-20

Zentraler Unterstützungsrechner der Prozeßrechenanlage



Herausgegeben von der Abt. Prozessrechenanlage des
EDV-Zentrums der Technischen Universität Wien.
1040 Wien, Gusshausstrasse 25

FEEDBACK

Herausgeber: EDV-Zentrum der Technischen
Universität Wien, Abt. Prozeßrechenanlage

Leitung: Dr. M. Paul

Redaktion: Dipl. Ing. W. Formanek

Lay-out: E. Kaspar, H. Brauner

alle 1040 Wien, Gußhausstraße 25

Druck: HTU Wirtschaftsbetriebe Ges.m.b.H.
1040 Wien, Argentinierstraße 8



INHALTSVERZEICHNIS:

	Seite
Seite der Redaktion	5
Aus der Arbeit des PRZ	7
Ein Jahr neue Rechenanlagen des Prozeßrechenzentrums - Bereich Gußhausstraße	9
Online-Datentransfer zwischen Satelliten (PDP-11/04, LSI) und zentralem Unterstützungsrechner DECsystem-20	13
Datentransfer zwischen TP-Rechner (PDP-11/34) und DECsystem-20 über Terminal	16
Magnetband-Kompatibilität zwischen den Betriebs- systemen TOPS-20 (DECsystem-20) und RSX-11M bzw. RSX-11D (PDP-11 Rechner)	17
CAMAC-Ein-Ausgabe: Frequenztest	20
Terminalsimulation für Satellitenrechner	22
Graphics am DECsystem-20	23
Pseudographics am DECsystem-20	25
IEE-488-Bus am zentralen Prozeßrechner des PRZ	26
Realisierung von H D L C auf Prozeßrechnern vom Typ PDP-11	33
Steuerung einer Parabolantenne mit Hilfe eines Prozeßrechners nach vorgegebenen Positionierungs- daten	35
Programmierung und Anwendung der Fourier-Analyse in Prozeß-Satelliten	38
Prozeßrechnereinsatz für Analyse und Simulation biologischer Systeme	39

	Seite
RSX-11 oder: dafür müßte es doch irgendetwas geben!	41
Computersimulation von sonnenbeheizten Anlagen	43
Das IBM System/7 auf dem Weg aus der "Isolation"	45
Jahresbericht 1977	51
Auswertung biorhythmischer Funktionsschwankungen	53
Prozeßrechnersteuerung einer Elektronenstrahlanlage	54
Pulsmagnetometer und Pulsfeldauswertung	55
Massenspektrometrie	56
Entwicklung von Schraubenverdichtern	56
Desorption	57
Analysemethoden	57
Auger-Spektrometer und Auger-Auswertung	58
Prozeßidentifikation	58
Spektroskopische Untersuchungen i. Bereich 10-2500 Å	59
Kristallinitätsbestimmung	59
Vibrationsmagnetometer	60
Geschwindigkeitsspektrometer	60
Mikroprozessor-Cross-Simulator	61
Einsatz d. Prozeßrechners z. Unterstützung d. Vortragenden f. d. Vorlesungen Allgemeine Elektrotechnik 1 und 2	61
Meßdatenauswertung	62
CAMAC-Modul Entwicklung	62

	Seite
Rauschuntersuchungen an Transistoren	63
Leitungsschutz	63
Digitalrechnermethoden f. Planung und Betrieb v. Elektroenergiesystemen	64
Organisatorisches	65
Leitung	67
Bereich Gußhausstraße	68
Bereich Hauptgebäude	70
Bereich Getreidemarkt	72

SEITE der REDAKTION

Nach einem Jahr des "Erfahrungssammelns" an den neuen Rechnern der Prozeßrechenanlage wollen wir diese Erfahrungen im vorliegenden "FEEDBACK" unseren Kunden und allen interessierten Lesern weitergeben.

Neben den Artikeln -"Aus der Arbeit des PRZ"- finden Sie auch wieder eine Auswahl an Kurzbeschreibungen von Projekten, die im Berichtsjahr 1977 an der Prozeßrechenanlage der TU-Wien durchgeführt wurden.

Wir möchten Sie einladen, wenn Sie daran interessiert sind, eigene Artikel über die Prozeßrechenanwendung im "FEEDBACK" zu veröffentlichen, sich mit der Redaktion in Verbindung zu setzen (W. FORMANEK, GuBhausstraße 25, Tel. 65 37 85 Kl. 344).

Erstmals wollen wir mit diesem Heft einen über unsere eigene Universität hinausgehenden Leserkreis erreichen. Auch diese Leser möchten wir einladen zum vorliegenden Heft Stellung zu nehmen, bzw. an uns Wünsche und Fragen heranzutragen.

W. F.

AUS DER ARBEIT DES PRZ

EIN JAHR NEUE RECHENANLAGEN DES PROZESSRECHENZENTRUMS - BEREICH GUSSHAUSSTRASSE

H. Havas

AUSGANGSPUNKT

- Die ersten Rechnerinstallationen
- Vorteile und Beschränkungen

Im Jahre 1970 wurde als "Keimzelle" des Prozeßrechenzentrums (PRZ) der Prozeßrechner IBM 1800 installiert. Dieser Prozeßrechner versorgte über ein Ringleitungssystem die Elektrotechnischen Institute im Institutsgebäude Gußhausstraße sowie eine Reihe von Instituten im Hauptgebäude am Karlsplatz.

1972 führte der konzentrierte Bedarf an Prozeßrechnerkapazität an den Chemie- und Maschinenbauinstituten am Getreidemarkt zur dortigen Installation des Prozeßrechners IBM S/7. Auch hier wurde den einzelnen Instituten der Zugriff zum Prozeßrechner zunächst über ein Verteilungssystem für die Anschlüsse an die Prozeßperipherie ermöglicht.

Den Abschluß der Installation einzelner zentraler Prozeßrechner für eine definierte Institutsgruppe bildete im Jahre 1973 der Prozeßrechner PDP-11/45 an den Physikalischen Instituten im Hauptgebäude am Karlsplatz (Abb.1)

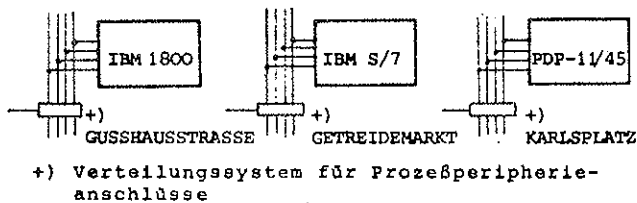


Abb.1 Verteilung der zentralen Prozeßrechner des PRZ bis zum Jahre 1977.

Zur Zeit der Installation der ersten Rechner des Prozeßrechenzentrums war ein Rechneinsatz mit sehr hohen Kosten verbunden. Der gemeinsame Zugriff zu zentral installierten Prozeßrechnern war daher für die Hochschulinstitute vielfach die einzige Möglichkeit, Prozeßrechner für die Automatisierung von Prozeßabläufen einzusetzen. Ebenso konnten die installierten Anlagen durch den gemeinsamen Zugriff wirtschaftlich genutzt werden.

Neben den Vorteilen einer gemeinsamen Nutzung der zentral installierten Anlagen brachte diese Betriebsform natürlich auch ihre spezifischen Probleme mit sich. Vor allem genügte die Sicherheit der Prozeßdatenübertragung über das relativ weit verzweigte Verteilungssystem für die Prozeßperipherieanschlüsse (Ringleitung) nicht immer den gestellten Anforderungen. Ebenso waren die teilweise sehr langen Leitungen des Verteilungssystems für die Übertragung hoher Datenraten nur bedingt geeignet.

WEITERE ENTWICKLUNG

- Einsatz von Kleinprozeßrechnern
- Vorteile und Beschränkungen

Auf dem Kleinrechnersektor ist seit geraumer Zeit eine rapide Kostensenkung zu beobachten. So wird auch die Verwendung von Kleinprozeßrechnern mit zunehmender Senkung der Hardwarekosten wirtschaftlich immer attraktiver. Dies um so mehr, als der Einsatz von Kleinprozeßrechnern eine Reihe von Vorteilen mit sich bringt. Da der mit Prozeßperipherie ausgestattete Kleinrechner in unmittelbarer Umgebung des Prozesses aufgestellt werden kann, ist im Vergleich zur zentralen Lösung eine wesentliche Erhöhung der Zuverlässigkeit der Prozeßdatenübertragung gegeben. Außerdem kann das Prozeßsteuerprogramm im Kleinrechner ungestört von eventuell parallel laufenden Aufgaben anderer Benutzer ablaufen, wodurch eine weitere Erhöhung der Verfügbarkeit der Prozeßsteuerung erreicht wird. Letztlich ist durch den Kleinprozeßrechner im Vergleich zu den fix installierten Anschlußpunkten der Ringleitung eine wesentlich größere Flexibilität des Prozeßrechnereinsatzortes in den einzelnen Labors eines Institutes zu erreichen.

Den genannten Vorteilen der Kleinrechner stehen jedoch einige bedeutende Nachteile bezüglich des wirtschaftlichen Einsatzes an Hochschulinstituten gegenüber. So bleibt der Kaufpreis des Kleinrechners im allgemeinen nur so lange interessant, als auf die Konfiguration von größeren Arbeitsspeicherkapazitäten und insbesondere auf die Installation von leistungsfähigen Massenspeichern verzichtet wird. Bei billigen und daher gering ausgebauten Kleinrechnern ist der Aufwand für die Erstellung von Programmen (Editieren, Übersetzen) verhältnismäßig groß. Dies gilt sowohl hin-

sichtlich des Zeitbedarfes für die Erstellung am Rechner als auch hinsichtlich des erforderlichen Ausbildungsniveaus des Rechnerbenützers (höhere Programmiersprache). Gerade den letztgenannten Punkten kommt durch die am Hochschulinstitut dominierende Rolle der Entwicklungsphase einer Prozeßapplikation und der oft erst im Anfangsstadium der Ausbildung stehenden Rechnerbenützer große Bedeutung zu. Geringer Ausbau von Kleinrechnern bringt darüber hinaus eine sehr empfindliche Beschränkung der Einsatzmöglichkeiten von umfangreicheren Auswerteprogrammen für die erfaßten Prozeßdaten mit sich (Arbeitsspeicher- und Massenspeicherkapazität, Programmiersprache). Weiters ist die permanente Speicherung größerer Datenmengen für eine spätere Auswertung wegen der bei Kleinrechnern im allgemeinen nicht verfügbaren Massenspeicher nicht möglich.

RECHNEREINSATZKONZEPT

- Hierarchisches Rechnernetzwerk
- Gliederung in Funktionsebenen

Das Rechnereinsatzkonzept der neuen Anlagen des PRZ im Bereich Gußhausstraße sieht die Aufteilung der geforderten Funktionen auf die Rechner eines hierarchisch gegliederten Rechnernetzwerkes vor.

Die Hierarchiestruktur des Rechnernetzes besteht aus den in Abb.2 gezeigten drei Funktionsebenen.

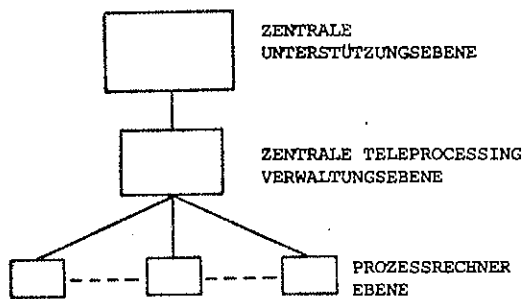


Abb.2 Funktionsebenen des Rechnernetzwerkes

a) Zentrale Unterstüztungsebene

In diesen Ebenen werden teure Ressourcen, wie große Arbeitsspeicher, große Massenspeicher, schnelle Drucker, Plotter usw. zur Verfügung gestellt.

Die Institute können sowohl über Terminals als auch über die zentrale TP-Verwaltungsebene (Punkt b) von den in den Instituten installierten Kleinprozeßrechnern aus auf die zentrale Unterstüztungsebene zugreifen. Damit besteht die Möglichkeit, die Programmerstellung für Kleinprozeßrechner sowohl zeiteffizient (leistungsfähige Editoren, Compiler) als auch bedienungsfreundlich (höhere Programmiersprache, Steuersprache eines Großrechners) durchzuführen. Die über Terminal erstellten Programme können von der zentralen Unterstüztungsebene aus in die Prozeßkleinrechner geladen werden.

Weiters können die im Prozeßrechner anfallenden Prozeßdaten über die zentrale TP-Verwaltungsebene (Punkt b) an die zentrale Unterstüztungsebene weitergeleitet werden, um dort für arbeitsspeicher-und/oder massenspeicheraufwendige Auswerteprogramme als Eingabedaten zu dienen. Die gewonnenen Resultate können wieder in den Prozeßkleinrechner gesendet werden, wodurch eine Rückkopplung auf den Prozeß in gewissen Zeitgrenzen möglich wird.

b) Zentrale Teleprocessing-Verwaltungsebene

Diese Ebene kontrolliert den gesamten Datenfluß zwischen der zentralen Unterstüztungsebene und der Ebene der Prozeßrechner. Ebenso können hier Daten zwischengespeichert werden, wodurch die Verfügbarkeit des Gesamtsystems wesentlich erhöht wird.

c) Prozeßrechnerebene

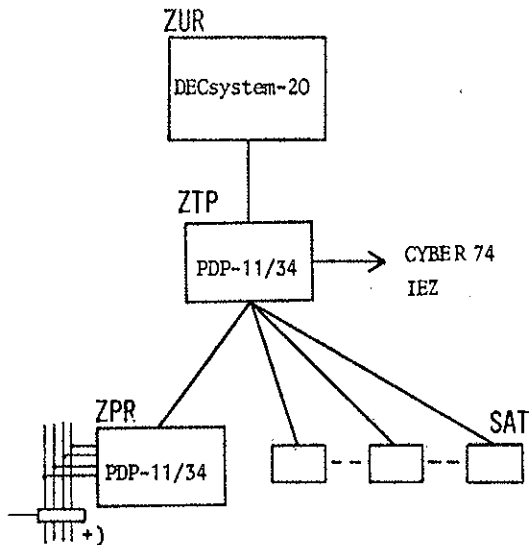
In dieser Ebene werden die Prozeßprogramme exekutiert. Diese Funktionsebene führt also die unmittelbare Erfassung der Prozeßdaten und/oder die direkte Beeinflussung des Prozesses (Steuerung, Regelung) durch.

REALISIERUNG DES RECHNERKONZEPTES

- Neue Anlagen im Bereich Gußhausstraße
- Eingliederung in das Gesamtnetzwerk

Durch die Neuinstallation der Rechner im Bereich Gußhausstraße war es in diesem Bereich möglich, die Aufgaben der einzelnen Funktionsebenen jeweils durch einen gezielt hierfür ausgewählten und konfigurierten Rechner zu erfüllen (Abb.3).

Abb.3 Rechnernetzwerk - Bereich Gußhausstraße



- ZUR Zentraler Unterstützungsrechner
- ZTP Zentraler Teleprocessing-Verwaltungsrechner
- SAT Prozeßrechner (Satelliten)
- +) Verteilungssystem für zentrale Prozeßperipherie

Die Funktionen der zentralen Unterstützungsebene werden von dem neu installierten Rechner DECsystem-20 übernommen. Der Rechner ist mit einem Arbeitsspeicher von 128k Worten (à 36 bit), zwei Plattentürmen à 50 M Byte, einer 9-Spur Magnetbandstation, einem schnellen Drucker, einem Kartenleser, einem Plotter sowie einer Reihe von Timesharing-Terminals ausgestattet. Die Anschlüsse für die Terminals können entweder fix durchgeschaltet oder über Telefon angewählt werden.

Zentraler Teleprocessing-Verwaltungsrechner

Die Aufgabe der Verwaltung des gesamten Datenverkehrs zwischen dem zentralen Unterstützungsrechner und den Prozeßrechnern im Bereich Gußhausstraße wird von den zentralen TP-Verwaltungsrechnern PDP-11/34 übernommen.

Die Teleprocessing-Anschlußeinheiten erlauben dzt. eine asynchrone Datenverbindung mit 9,6 kBd. An einer Verbesserung dieser Datenrate wird gearbeitet.

Eine Festkopfflatte und eine Wechselplatte erlauben die Zwischenpufferung der Daten. Dadurch wird sowohl eine Steigerung der Sicherheit des Datentransfers als auch eine Entlastung der Verbindung zum zentralen Unterstützungs-

rechner erreicht.

Darüber hinaus erwies es sich während des ersten Installationsjahres als günstig, einzelne Funktionen im Zuge der Programmerstellung bereits an diesem Rechner durchzuführen. Damit konnte eine gleichmäßigere Belastung der zentralen Unterstützungsrechner erreicht werden. Ebenso konnte durch diese Verteilung der Aufgaben eine wesentliche Erhöhung der Verfügbarkeit der zentralen Unterstützungsfunktionen erreicht werden.

Prozeßrechner

Der Hauptanteil der Rechner in der Funktionsebene "Prozeßrechner" im Bereich Gußhausstraße wird von Prozeßkleinrechnern gestellt, die direkt in den Labors der einzelnen Institute installiert sind. Diese Prozeßkleinrechner sind i.a. ohne externe Massenspeicher bzw. nur mit relativ kleinen Arbeitsspeichern ausgestattet. Dadurch kann der Anschaffungspreis für das Institut sehr gering gehalten werden.

Zwei Prozeßkleinrechner des Typs PDP-11/04 stehen dem Prozeßrechenzentrum für die vorübergehende Vergabe an die Institute zur Verfügung.

Für jene Institute, für die der Einsatz eines Prozeßkleinrechners dennoch zu aufwendig ist, steht der zentral installierte Prozeßrechner PDP-11/34 (Zentraler Prozeßrechner ZPR) zur Verfügung. Dieser Rechner erfüllt im Bereich Gußhausstraße im kleineren Rahmen die Aufgaben der Vorgängermaschine IBM 1800. Die Institute haben über die bereits vorhandene Ringleitung die Möglichkeit, über Anschlußtafeln zu der zentral installierten Prozeßperipherie zuzugreifen.

In Anbetracht der weiter rapid sinkenden Kosten für Kleinprozeßrechner verliert diese Betriebsform zunehmend an Bedeutung und wird in absehbarer Zeit durch Institutsprozeßsatelliten ersetzbar sein.

In den Bereichen Getreidemarkt und Karlsplatz erfüllen die dort installierten Prozeßrechner (Zentrale Prozeßrechner ZPR in Abb.4) entsprechend der historischen Entwicklung dieser Anlagen gleichzeitig mehrere Funktionen. Das heißt, die einzelnen Funktionen der Funktionsebenen lassen sich hier nicht auf getrennte Rechner aufteilen.

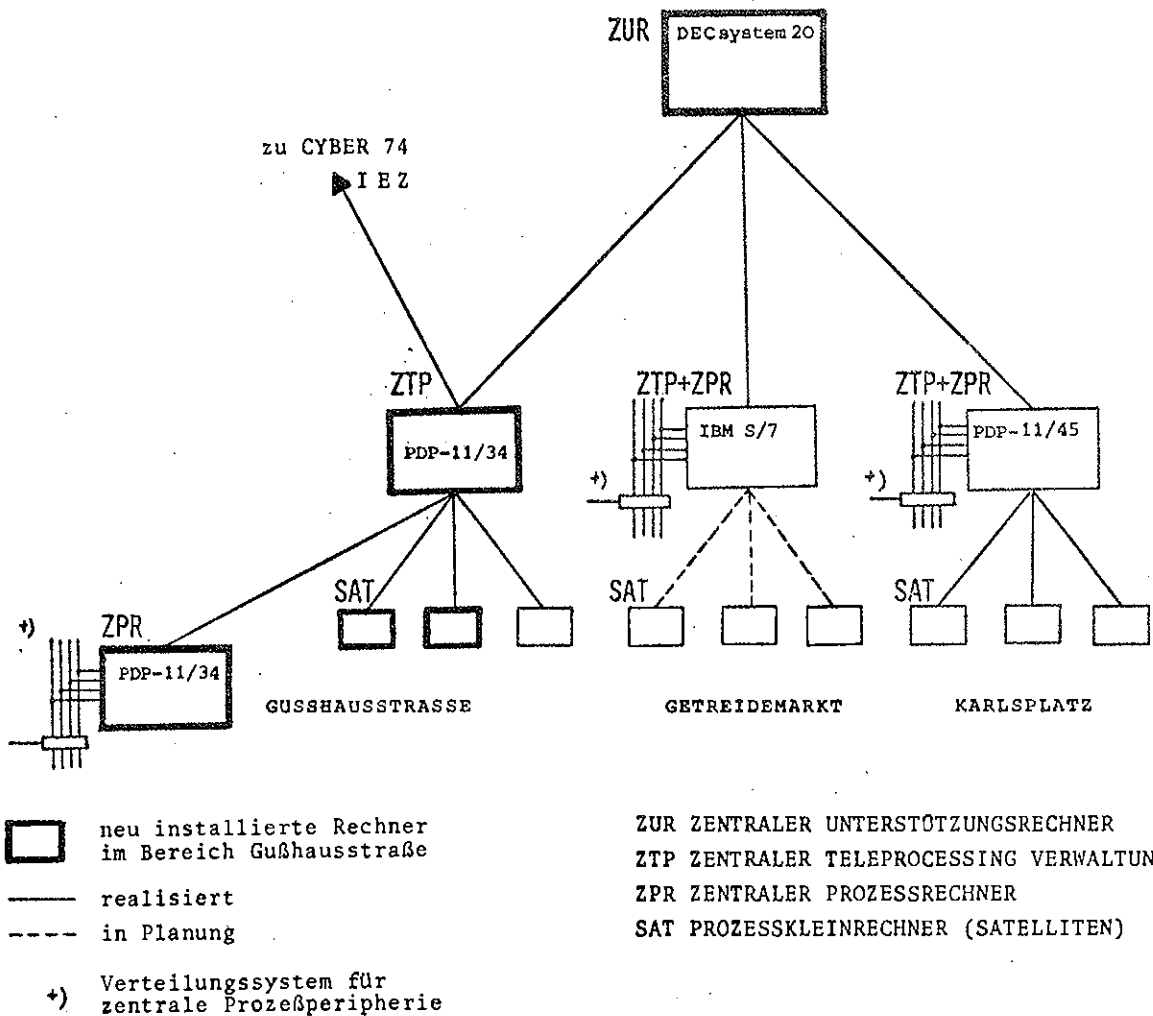


Abb.4 Rechnernetzwerk des PRZ

So übernimmt z.B. der Rechner PDP-11/45 gleichzeitig die Funktionen eines zentralen Prozeßrechners und die der zentralen Verwaltung der von den angeschlossenen Kleinprozeßrechnern kommenden Daten. Weiters wird von diesen Rechnern eine Reihe von zentralen Unterstützungsfunktionen übernommen. An das zentrale Unterstützungssystem DECsystem-20 werden nur jene Unterstützungsfunktionen weitergeleitet, die an diesem Rechner besser durchgeführt werden können. (Z.B. Auswerteprogramme mit extremem Arbeitsspeicherbedarf, Speicherung extrem großer Datenmengen u.ä.).

rechnerfunktionen innerhalb eines hierarchisch gegliederten Rechnernetzwerkes. Hierbei werden teure Ressourcen in wirtschaftlicher Weise zentral für alle Institute zur Verfügung gestellt. Aufgaben der direkten Prozeßsteuerung werden von denselben billigen Kleinprozeßrechnern übernommen, wodurch für kritische Prozeßaufgaben ein hohes Maß an Verfügbarkeit erreicht wird. Die Realisierung dieses Konzeptes wurde im Bereich Karlsplatz (Physikinstitute) begonnen und bei den neuen Anlagen im Bereich Gußhausstraße weiter ausgebaut.

ZUSAMMENFASSUNG

Grundlinie des Rechnereinsatzes des Prozeßrechenzentrums ist die Aufteilung von Prozeß-

ONLINE-DATENTRANSFER ZWISCHEN SATELLITEN (PDP-11/04 , LSI) UND ZENTRALEM UNTERSTÜTZUNGSRECHNER DECSYSTEM-20

W.Lauber

Das am PRZ installierte DECSYSTEM-20 dient vor allem dazu, komplexe Auswertungen großer Datenmengen, die von Satelliten (PDP-11/04, LSI) erfaßt und vorverarbeitet wurden, durchzuführen. Der Datentransfer zwischen Programmen, die in den Satelliten laufen, und dem DECSYSTEM-20 wird über zwei Programmpakete abgewickelt, deren Kopplung im Knoten des TP-Netzwerks Bereich Gußhausstraße, dem TP-Rechner PDP-11/34, durchgeführt wird (Abb.1).

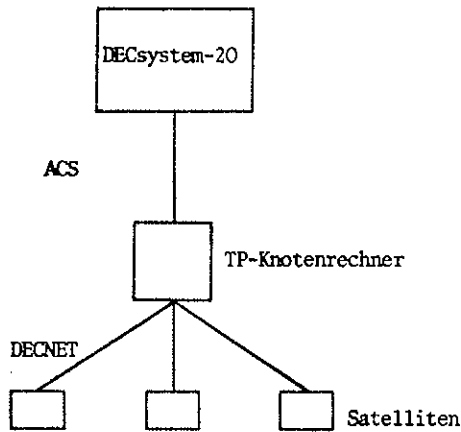


Abb.1 Software für Kommunikation zwischen Satelliten, TP-Knotenrechner und Zentralem Unterstützungsrechner DECSYSTEM-20

DATENTRANSFER ZWISCHEN SATELLITEN UND TP-RECHNER

Sowohl der File-Transfer als auch der Online-Datentransfer zwischen diesen Rechnern wird über das Programmpaket DECNET durchgeführt. Dabei kann eine Datenverbindung prinzipiell zwischen zwei beliebigen Rechnern des Netzes, also auch zwischen zwei Satelliten, hergestellt werden; von zentraler Bedeutung ist aber natürlich die Verbindung zum TP-Knotenrechner, der die weitere Kommunikation zum Zentralen Unterstützungsrechner ermöglicht. Eine online-

Verbindung zwischen Satelliten und DECSYSTEM-20 kann (a) über eine Zwischenspeicherung auf den Datenplatten des TP-Rechners erfolgen oder (b) über eine direkte Kommunikation zwischen drei aktiven Tasks in Satelliten, TP-Rechner und DECSYSTEM-20. Fall (a) ist besonders dann sinnvoll, wenn der Datentransfer nur zur Abspeicherung großer Datenmengen auf Massenspeicher benötigt wird, eine Benutzung des DECSYSTEM-20 für umfangreiche Auswerteprogramme jedoch nicht vorgesehen ist.

Ist ein weiterer Online-Transfer der Daten zwischen TP-Rechner und DECSYSTEM-20 vorteilhaft bzw. nötig, wie etwa für zeit- und speicherintensive Auswerteprogramme, erfolgt dies (c) über ein vom Satelliten aus im TP-Rechner gestartetes Programm und das Programmpaket ACS. (Der vom Terminal aus gesteuerte Offline-Transfer zwischen TP-Rechner und DECSYSTEM-20 wird in einem weiteren Beitrag [Formanek] beschrieben.)

Im folgenden werden die Aufrufe beschrieben, die mit diesen Software-Paketen zur Verfügung stehen. Programmbeispiele werden bewußt nicht angeführt, da es aufgrund der vielen Funktionen (Routinen) kein typisches Beispiel geben kann. Da sich in den am PRZ zugänglichen Manuals detailliertere Angaben sowie zahlreiche Programmbeispiele befinden, ist vor eigenen Programmierversuchen ein Studium dieser Unterlagen bzw. eine Rücksprache mit der PRZ-Programmerberatung unbedingt zu empfehlen.

a) Zugriff auf Files

Folgende Routinen dienen dieser Funktion:

- NFOPRW öffnet einen File zum Lesen
- NFOPAW öffnet einen File zum Erweitern
- NFOPWW öffnet einen File zum Schreiben
- NFGETW liest einen Record
- NFPUTW schreibt einen Record
- NFCLSW schließt einen File
- NFDELW löscht einen File
- NTINIT reserviert einen Buffer für den DECNET-Transfer

Die Reihenfolge der Aufrufe ist daher: NTINIT, NFOPRW oder NFOPAW oder NFOPWW, NFGETW oder NFPUTW, NFCLSW.

Format der Aufrufe:

```
CALL NFOPRW }
CALL NFOPAW } (lun,istat,node,/idinfo/,/ifile/,
CALL NFOPWW } /ichar/,/len/,/iblock/)
```

Die zwischen Schrägstrichen /.../ angeführten

Argumente sind Optionen.

lunlogische Nummer, die dem File zugeordnet wird

istat ...2-Wort-Statusblock; enthält Completion-Code

node6-Character ASCII-String für den Namen des Knotens, der den File enthält (hier also des TP-Knotenrechners)

idinfo ..falls nötig, Password des angesprochenen Knotens (ASCII)

ifile ...File-Specification entsprechend RSX-Regelung

ichar ...damit wird mittels eines maximal 3 Characters enthaltenden ASCII-Strings der Mode (A für ASCII, I für binary-image), Record-Type (F für feste Record-Länge, V für variable) und Carriage-Control (F für Fortran-C.C, T für Terminal C.C, N: kein Carriage-Control) festgelegt.

lenRecord-Länge in Bytes

iblock ...File-Größe in Blocks (1 Block=256 Worte)

```
CALL NFGETW }
CALL NFPUTW } (lun,istat,ibytes,iarray)
```

lun, istat ... wie oben

ibytes ...Zahl der zu lesenden bzw. zu schreibenden Bytes

iarray ...Feld, das den zu schreibenden bzw. gelesenen Record enthält

CALL NTINIT (istat, iwords, iarray)

istat ... wie oben

iwords .. Integer=14 + 21n + m, wobei n die Maximalzahl gleichzeitig verwendeter logischer Verbindungen für die Task-zu-Task Verbindung (siehe unten Punkt b) bedeutet, m die maximale Recordlänge (bei File-Transfer muß m mindestens 50 gewählt werden).

iarray... Feld der Länge iwords

b) Kommunikation zwischen zwei Tasks

Für diese Funktionen stehen folgende Routinen zur Verfügung:

NTINIT siehe Punkt (a)

NTCON/W/ stellt logische Verbindung mit dem aufrufenden bzw. aufgerufenen Task her. Im Gegensatz zur NTCON wartet NTCONW die Durchführung dieser Funktion ab, entsprechendes gilt für die übrigen Funktionen.

NTCGT/W/ beschafft die zur Herstellung einer Verbindung unter Umständen (d.h. wenn nicht bekannt) nötige Information

NTRCJ weist Anforderung einer logischen Verbindung zurück

NTSND/W/ sendet Daten

NTRCV/W/ empfängt Daten

NTSNI/W/ sendet Interrupt

NTDIS/W/ löst logische Verbindung, die von NTCON bzw. NTCONW hergestellt wurde

NTWAIT suspendiert den rufenden Task

NTINFO stellt dem rufenden Programm den Namen seines eigenen Knoten zur Verfügung.

Die Reihenfolge der Aufrufe ist daher:

- a) rufender Task
NTINIT, NTCON/W/, NTSND/W/ bzw. NTRCV/W/ bzw. beide, NTDIS/W/
- β) gerufener Task (dieser Task, der am entsprechenden Knoten nur installiert sein muß, wird erst aufgrund des Aufrufes NTCON/W/ des rufenden Tasks aktiv)

Format der Aufrufe:

CALL NTINIT siehe oben, Punkt (a)

CALL NTCON/W/(lun,istat,icon,node,/task/,/iobj/, /iuic/,/iwords/,/iarray/,/astsub/)

CALL NTCGT/W/(istat,icon,node,/task/,/iobj/, /iuic/,/iwords/,/iarray/)

CALL NTRCJ (lun,istat,icon,node,task,iobj, iuic,ibytes,iarray)

CALL NTSND/W/ }
CALL NTRCV/W/ } (lun,istat,ibytes,iarray)
CALL NTSNI/W/ }

CALL NTDIS/W/ (lun,istat,/iwords/,/iarray/)

CALL NTWAIT (lun,/istat/)

CALL NTINFO (lun,istat,ibytes,iarray)

Die zwischen Schrägstrichen /.../ angeführten Argumente sind Optionen.

lun,istat,node,...siehe oben Punkt (a)

icon.....Nummer der logischen Verbindung; beim Aufruf NTCON/W/ des aufrufenden Tasks muß dieser Wert Null sein. Der gerufene Task muß den Wert mittels NTCGT/W/ ermitteln und im Aufruf von NTCON/W/ angeben.

task....Name des Tasks, mit dem kommuniziert wird, in ASCII
 iobj....spezielle Information des Systems
 iuic....2-Wort Integerfeld mit den Werten des UIC des Tasks, mit dem kommuniziert wird
 iwords, iarray...iwords gibt die Anzahl der Worte an, die im Feld iarray übergeben werden (für spezielle Information)
 ibytes, iarray...ibytes gibt die Anzahl der Bytes an, die im Feld iarray übergeben werden.
 astsub...Name einer Routine, die bei Interrupt exekutiert werden soll

Entsprechend der Funktion der jeweiligen Routine sind die Parameter Ein- bzw. Ausgabe-parameter.

c) Start eines Tasks in einem beliebigen Knoten

Diese Funktion wird mittels CALL NCRUNW(lun, node, istat, task) durchgeführt, der Abbruch eines Programmes erfolgt über CALL NCABOW(lun, node, istat, task).
 Die Parameter entsprechen den unter Punkt (a) bzw. (b) angegebenen.

DATENTRANSFER ZWISCHEN TP-RECHNER UND DECsystem-20

Die Datenkommunikation zwischen diesen beiden Rechnern erfolgt über das Programmsystem ACS. Im folgenden werden nur jene Funktionen des ACS erläutert, die für eine online-Verbindung der Rechner zur Verfügung stehen.

Am TP-Rechner sind dies die folgenden QIOs:

CALL QIO(KODE, LUN, /IEFN/, /PRI/, IACST, IPARAM, /IDS/)

Die in Schrägstrichen/.../ angegebenen Parameter sind Optionen.

KODE....dieser Kode gibt an, ob am DECsystem-20 ein Programm gestartet werden soll, ob Daten einem bereits derart gestarteten Programm gesendet oder von diesem empfangen werden sollen.

KODE="2400 Starten eines Programms
 KODE="6000 Senden von Daten
 KODE="5400 Empfangen von Daten

LUN Nummer der logischen Verbindung

IEFN ..Event Flag, das durch den Aufruf gesetzt und nach Durchführung gelöscht wird. Kann mit CALL WAITFR(IEFN) abgefragt werden.
 /PRI/ .. Priorität
 IACST... 2-Wort-Integer-Feld, das Completion Kode enthält
 IPARAM ..6-Wort-Integer-Feld, das Angaben über weitere Daten enthält:
 IPARAM(1): Adresse jenes Feldes IFELD, das im Fall der Funktion "Programmstart" den Namen des zu startenden Programmes in ASCII enthält, im Falle "Senden" bzw. "Empfangen" die zu sendenden bzw. empfangenen Daten. In FORTRAN wird die Adresse mittels CALL GETADR(IPARAM(1), IFELD(1)) angegeben.
 IPARAM(2): Anzahl der Bytes der in IFELD enthaltenen Information.
 IPARAM(3)="377 beim Aufruf für Start, Senden, Empfangen
 ="376 für Senden eines EOF
 IDS enthält u.U. Information über Interrupt-Routine

Im Task, der mittels QIO(KODE_{start}, ...) auf dem DECsystem-20 aufgerufen wird, können folgende Aufrufe zum Empfangen resp. Senden von Daten zum aufrufenden Task verwendet werden:

CALL RECV(JFELD, ICNT, IFLAG) empfängt auf dem Feld JFELD Datenbytes der Anzahl ICNT. IFLAG enthält einen Return-Kode und ist mittels CALL COMMERR(IFLAG) abfragbar.

CALL SEND(JFELD, ICNT, IFLAG) sendet ICNT Bytes des Feldes JFELD.

Wie bereits erwähnt, sollen die obigen Angaben einen Einblick in die Möglichkeiten der Kommunikations-Software für online-Übertragung vermitteln. Für genauere Angaben über die einzelnen Parameter, die verwendeten Systembibliotheken sowie Programmierbeispiele stehen Manuals sowie die Programmberatung zur Verfügung.

DATENTRANSFER ZWISCHEN TP-RECHNER (PDP-11/34) UND DECsystem-20 OBER TERMINAL

W. Formanek

Mit dem Rechner DECsystem-20 wird allen Benutzern des Prozeßrechenzentrums ein leistungsfähiger zentraler Unterstützungsrechner (ZUR) angeboten. Den Kunden stehen mit der Rechnerverbindung DECsystem-20 - TP-Rechner alle Ressourcen des ZUR wie z.B. Kartenleser oder Lineprinter zur Verfügung (Siehe auch Artikel Havas: "Ein Jahr neue Rechenanlagen des PRZ"). Um diesen Hintergrundsrechner voll ausnützen zu können, ist es nötig, einiges über die Möglichkeiten eines Datentransfers zwischen DECsystem-20 und dem TP-Rechner zu wissen. Ergänzend zum vorangehenden Beitrag über Datentransfer von Programmen aus, soll hier der über Terminal aktivierte Datentransfer beschrieben werden.

Für die Verbindung PDP-11/34 mit dem DECsystem-20 wird das Programmpaket ACS benutzt. Ein Transfer kann sowohl von einem Terminal, das an den TP-Rechner angeschlossen ist, als auch von einem Terminal des zentralen Unterstützungsrechners aus gestartet werden (Abb.1).

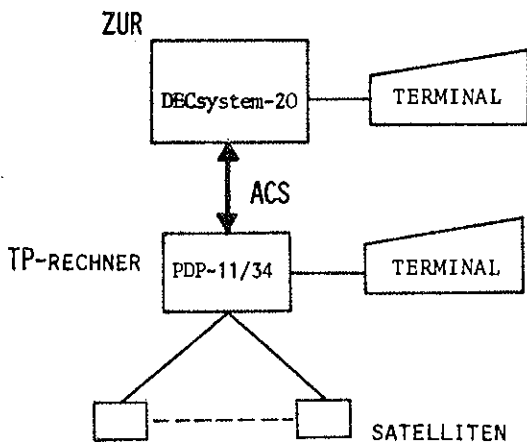


Abb.1

Zunächst ein Überblick über die für den Benutzer wichtigsten ACS-Funktionen, die vom TP-Rechner-Terminal aus durchgeführt werden können. Nach der Kurzbeschreibung jeder Funktion folgt jeweils ein Beispiel, das die genaue Anwendung zeigen soll:

FILSND sendet einen File vom TP-Rechner zum DECsystem-20

```
>RUN FILSND
ENTER TYPE [ 0=ASCII, 1=BIN ] :
>0
FILENAME ON REMOTE NODE : <K353152>ACS.TST
FILENAME ON OWN NODE : DK31[200,200]ACS.TST
..DONE..
```

FILREC holt einen File vom DECsystem-20 zum TP-Rechner

```
>RUN FILREC
ENTER TYPE [ 0=ASCII, 1=BIN ] :
>0
FILENAME ON REMOTE NODE : <K353152>ACS.TST
FILENAME ON OWN NODE : DK01[353,152]ACS.TST
..DONE..
```

OPR schickt eine Message zur Operator-Konsole des DECsystem-20

```
>OPR
MESSAGE : NACHRICHT AN OPERATOR DER 20ER
..DONE..
```

Wie aus den Beispielen zu entnehmen ist, wird der Benutzer über das Terminal nach den jeweils notwendigen Angaben gefragt.

Bei der Benutzung eines Terminals des DECsystem-20 schaut ein Transfer etwas anders aus. Zunächst muß das Programm REMOTE aufgerufen werden, dann werden die einzelnen ACS Funktionen und die dazu nötigen Angaben gesetzt.

Die wichtigsten ACS Funktionen auf dem DECsystem-20 lauten:

SEND sendet einen File vom DECsystem-20 zum TP-Rechner

```
@REMOTE
REMOTE>SET 20-FILE ACS.TST
REMOTE>SET FE-FILE
ACS.ISI
REMOTE>SET SEND
REMOTE>00
[SEND]
[***REQUEST EXECUTED***]
REMOTE>[Acs20 - Done ]
```

RECEIVE holt einen File vom TP-Rechner
zum DECsystem-20

```

REMOTE>SET FE-FILE
ACS.TST-----
REMOTE>SET 20-FILE ACS.TST
REMOTE>SET RECEIVE
REMOTE>CH

      NODEINDEX:      0
      MODE:           ASCII
      FUNCTION:       [RECEIVE]
      20-FILE:       <WILLI>ACS.TST
      FROM:
      FE-FILE:       ACS.TST

REMOTE>DO
[RECEIVE]
[***REQUEST EXECUTED***]
REMOTE>[ACS20 - Done ]

```

SUBMIT exekutierte einen Command-File
im TP-Rechner

```

@REMOTE
REMOTE>SET SUBMIT
REMOTE>SET FE-FILE
ACSISI.CMD-----
REMOTE>CH

      NODEINDEX:      0
      MODE:           ASCII
      FUNCTION:       [SUBMIT]
      PROGRAM:       ACSTST.CMD

REMOTE>DO
[SUBMIT]
[***REQUEST EXECUTED***]
REMOTE>E
[EXIT]
@

```

Beim Datentransfer von einem DECsystem-20 Terminal aus empfiehlt es sich, vor dem DO-Kommando, das die ACS Funktionen startet, noch ein Check(CH) zu machen. Dabei werden die für den Transfer notwendigen und bereits eingetippten Angaben in übersichtlicher Form wiederholt und eventuell noch fehlende urgieren. Mit E (CR) steigt man aus dem REMOTE Programm wieder aus. Für genauere Informationen (auch über weitere ACS-Funktionen) wird auf Manuals bzw. die Programmberatung hingewiesen.

MAGNETBAND-KOMPATIBILITÄT ZWISCHEN DEN BETRIEBS- SYSTEMEN TOPS-20 (DECsystem-20) UND RSX-11M BZW. RSX-11D (PDP-11 RECHNER)

P. Lorenz

EINLEITUNG

Der Rechner DECsystem-20 als zentraler Unterstützungsrechner des Prozeßrechenzentrums gestattet die Implementierung bzw. Entwicklung von umfangreichen Auswerteprogrammen einerseits und die Speicherung von Daten in größerer Menge andererseits.

Zum Teil waren nun diese Auswerteprogramme schon auf dem System CYBER 74 vorhanden, zum Teil wurde die Entwicklung solcher Programme in Erwartung des Zentralen Unterstützungsrechners des PRZ hintangestellt und die bereits anfallenden Daten vorerst auf diversen Speichermedien abgelagert. Gleichzeitig ist der Bedarf entstanden, Programme, die auf dem DECsystem-20 entwickelt werden, auf anderen Rechnern, sei es nun die CYBER 74 oder PDP-11 Maschinen, zu implementieren.

So entstand der Bedarf nach einer Datenübertragung in beiden Richtungen, d. h.: Zwischen den Zentralen Prozeßrechnern bzw. Satelliten und den Systemen DECsystem-20 bzw. CYBER 74.

Dabei wurde an zwei Wege gedacht:

1. Datenübertragung mittels Datenleitung zwischen den oben genannten Rechnern (darüber wird in den Beiträgen auf S. 13 u. S. 16 berichtet),

2. Datenübertragung mittels Magnetband. Der zweite Weg ist für sehr große Datemengen besser geeignet. Die Schwierigkeit, die hierbei auftritt, ist, daß jede dieser Maschinen das Magnetband in einem anderen Format beschreibt. Es hat somit die Notwendigkeit bestanden, an geeigneter Stelle ein Programm zu entwickeln, welches ein auf einem anderen Rechner verwendetes Datenformat schreiben und lesen kann.

Als Standardformat wurde hierfür der "ANSI MAGNETIC TAPE STANDARD" zugrunde gelegt.

Hierin ist eine klare File-Struktur definiert und bereits die Möglichkeit vorhanden, auf dem System CYBER 74 solche Bänder zu erstellen.

PROGRAMMBESCHREIBUNG

Auf folgende Art ist es nun möglich, auf dem System DECsystem-20 ein Magnetband zu bedienen, das in oben erwähntem Format beschrieben worden ist bzw. beschrieben werden soll:

Aufruf des Programmes:

TAKE DMP 11 ... Initialisierung und Programm aufruf.

Es "promptet" das Programm mit:

DMP 11>

und erwartet folgende Befehle:

TAPE MTAN: ordnet der Magnetbandstation MTAN: eine Job-File-Number zu

INIT volumename initialisiert das Magnetband

SAVE filename schreibt Daten, mit Filename bezeichnet, auf das Band

RESTOR filename schreibt Daten, mit Filename bezeichnet, von Band auf die Datenplatte

DIR listet die auf dem Band vorhandenen Datenfiles unter ihrem Filename auf Terminal

REWIND fährt zum Ladepunkt zurück

UNLOAD spannt das Magnetband aus

EXIT beendet das Programm

Andere oder falsche Befehle werden mit Fehlermeldungen abgefangen. Einige Fehler-situationen, die beim Arbeiten entstehen können, werden ebenfalls abgefangen, fatale Fehlersituationen können, falls sie auftreten, aber auch zur Beendigung des Programmes führen.

BESCHREIBUNG DES DATENFORMATES "ANSI MAGNETIC TAPE STANDARD":

1. Organisation des Bandes: "Multi File Single Volume"
 BOT, VOL1, HDR2 * --Daten-- * EOF1, EOF2 *
 HDR1, HDR2 * ---Daten--- * EOF1, EOF2 * *
 * bedeutet Tape Mark
 ** bedeutet END OF TAPE
 BOT bedeutet Ladepunkt
 , bedeutet eine physikalische Recordbegrenzung
2. Volume und File Labels
 VOL1 HDR1 HDR2 EOF1 EOF2

Volume Label Format

VOL1

CHARACTER POSITION	FIELD NAME	LENGTH IN BYTES	CONTENTS
1-3	Label identifier	3	VOL
4	Label number	1	1
5-10	Volume identifier	6	Volume label. Any alphanumeric or special character in the center four columns of the ASCII code table.
11	Accessibility	1	Any alphanumeric or special character in the center four columns of the ASCII code table. A space indicates no restriction. All volumes produced by IAS or RSX-11 have a space in this position.
12-37	Reserved	26	Spaces
38-51	Owner identification	14	The contents of this field are system-dependent and are used for volume protection purposes. See Section G.1.1.1 below.
52-79	Reserved	28	Spaces
80	Label standard version	1	1

Header 1 Label Format

HDR1

CHARACTER POSITION	FIELD NAME	LENGTH IN BYTES	CONTENT
1-3	Label identifier	3	HDR
4	Label number	1	1
5-21	File identifier	17	Any alphanumeric or special character in the center four columns of the ASCII code table.
22-27	File set identifier	6	Volume identifier of the first volume in the set of volumes.
28-31	File section number	4	Numeric characters. This field starts at 0001 and is increased by 1 for each additional volume used by the file.
32-35	File sequence number	4	File number within the volume set for this file. This number starts at 0001.
36-39	Generation number	4	Numeric characters.
40-41	Generation version	2	Numeric characters.
42-47	Creation date	6	yyddd (indicates space) 00000 if no date.
48-53	Expiration date	6	Same format as creation date.
54	Accessibility	1	Space
55-60	Block count	6	000000
61-73	System code	13	The three letters DEC followed by name of system that produced the volume. See Section G.1.1.1. Examples: DECFILE11A DECSYSTEM10 Pad name with spaces.
74	Reserved	7	Spaces

Header 2 Label Format
HDR2

CHARACTER POSITION	FIELD NAME	LENGTH IN BYTES	CONTENT
1-3	Label identifier	3	HDR
4	Label number	1	2
5	Record format	1	F - fixed length D - variable length S - spanned U - undefined
6-10	Block length	5	Numeric characters
11-15	Record length	5	Numeric characters
16-50	System-dependent information	35	Positions 16 through 36 are spaces. Position 37 defines carriage control and can contain one of the following: A - first byte of record contains FORTRAN control characters, space - line feed/carriage return is to be inserted between records, M - the record contains all form control information. If DEC appears in positions 61 through 63 of HDR1, position 37 must be as specified above. Positions 38 through 50 contain spaces.
51-52	Buffer offset	2	Numeric characters. 00 on tapes produced by Files-11. Not supported on input to Files-11.
53-80	Reserved	28	Spaces

ZUSAMMENFASSUNG

Der Zentrale Unterstützungsrechner des PRZ DECsystem-20 und die Zentralen Prozeßrechner PDP-11 in den einzelnen Teilbereichen des PRZ sind bereits bzw. werden mit Magnetbandstationen ausgestattet. Der Datenaustausch zwischen diesen Rechnern mittels Magnetband stößt auf Schwierigkeiten, da die einzelnen Rechner das Magnetband mit unterschiedlichen Formaten beschreiben. Ein am PRZ entwickelter Programmodul ermöglicht den Datenaustausch über Magnetband durch entsprechende Umformierung in ein gemeinsames Standardformat.

EOF1 und EOF2 (End of file labels) haben das gleiche Format wie die entsprechenden Header.

3. Format der Daten

Die Daten sind in Blöcke zu 512 Bytes a 8 Bit aufgeteilt, am Ende eines Blockes steht ein Up-Arrow (Oktal 136).

Ein Block ist in beliebig viele Records aufgeteilt. Ein Record besteht aus dem Recordbytecount zu Beginn, dann folgen die Daten in Bytes. Zu beachten ist, daß der Recordbytecount aus 4 Bytes besteht und in dezimalen Zeichen geschrieben ist. Außerdem wird er mitgezählt: Also Anzahl der Datenbytes + 4. Records können nicht über Blockgrenzen geschrieben werden.

Beispiele:

0004 bedeutet eine Leerzeile
0006AA bedeutet zweimal den Buchstaben
 A
0010DMP11> bedeutet DMP11>

CAMAC-EIN-AUSGABE: FREQUENZTEST

P. Tinkl

HINLEITUNG

Im Rahmen von Voruntersuchungen für Institutsprojekte wurden die maximalen Geschwindigkeiten der CAMAC-Prozeßperipherie der in der Gußhausstraße installierten PDP-11/34 ermittelt. Die Aufgabenstellung gliederte sich in zwei Zielrichtungen:

- a. Die maximale theoretische Geschwindigkeit
Dazu wurden sowohl von der Programmierung als auch vom hardwaremäßigen Versuchsaufbau her optimale Wege beschrritten, die im tatsächlichen Kundenbetrieb oft nicht erreicht werden können.
Deshalb wurde auch versucht,
- b. die reale, praxisnahe Geschwindigkeit zu ermitteln, die jedoch sehr von der jeweiligen Anwendung abhängt, sodaß nur überschlagsmäßig Richtlinien angegeben werden können.

VERSUCHSAUFBAU

Der Frequenztest gliederte sich in vier Teile: Digitalein- und Digitalausgabe sowie Analog-ein- und Analogausgabe.

Zur Eingabemessung wurde folgendermaßen vorgegangen: Ein Frequenzgenerator liefert eine Rechteckspannung, deren Frequenz solange erhöht wird, bis die höchste Einleserate erreicht ist.

Die Ausgaberate wird an einem direkt an den Ausgabebuchsen angeschlossenen Oszillographen gemessen.

(Abb. 1)

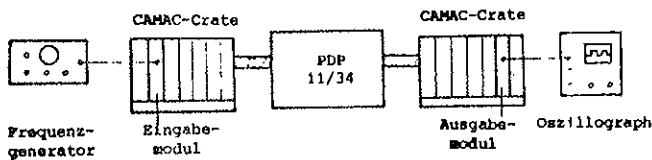


Abb.1 Versuchsanordnung

Bei den Versuchen wurde direkt am Stecker des CAMAC-Moduls bzw. sogar unter Umgehung der Eingangsbeschaltung (z. B.

Optokoppler) des Moduls gemessen. Weiters wurde auch die Geschwindigkeit am Interface zur Haus-Ringleitung gemessen.

PROGRAMMIERUNG

Auch hier wurden zwei verschiedenartige Programmstrukturen verwendet. Zur Erzielung maximaler Geschwindigkeiten wurden Programme in Assembler entwickelt, die nur die absolut notwendigen Befehle enthalten, um den Modul anzusprechen und die Werte schnellstmöglich im Hauptspeicher abzulegen, bzw. an den Modul zu transferieren. In diesen Ein- und Ausgabezyklen wurde auf Rechenoperationen verzichtet, was in der Praxis oft nicht möglich ist.

Andererseits wurden auch Programme in FORTRAN erstellt, die die Ein/Ausgabegeschwindigkeiten auf der Ebene höherer Programmiersprachen testen sollen.

ERGEBNISSE

Die Tests erbrachten folgende Ergebnisse:

- a. Digitale Ausgabe
Die maximale Ausgaberate mit Assemblerprogrammen beträgt hier 70 kHz. Diese Größe wird nicht von Hardwaregeschwindigkeiten bestimmt, sondern ergibt sich aus der Exekutionszeit der notwendigen Befehle. Weiters wurde ein praxisnahes Programm in FORTRAN erstellt; es ergab sich eine Ausgaberate von 5 kHz. Generell kann gesagt werden, -das gilt für alle weiteren Punkte und wird dort nicht mehr gesondert angeführt- daß die Ein/Ausgabeprogrammierung auf FORTRAN-Niveau einen Geschwindigkeitsverlust gegenüber Assembler-Programmen von überschlagsmäßig einer Zehnerpotenz ergibt.
- b. Digitale Eingabe:
Die Grenzfrequenz der Optokopplereingänge der auf dieser Anlage verwendeten DI-Module liegt bei ca. 10 kHz. Diese Maximalrate konnte jedoch erhöht werden, indem die Eingangsbeschaltung überbrückt wurde, was jedoch von der Prozeßrechenanlage nur für Spezialanwendungen durchgeführt wird. Die dabei

erzielten Werte liegen bei ca. 60 kHz.

c. Analoge Ausgabe:

Die Ausgabefrequenz von analogen Signalen ist wiederum durch die Anzahl der benötigten Assemblerbefehle begrenzt. Es sind daher Ausgaberraten bis zu 70 kHz möglich.

d. Analoge Eingabe:

Die Geschwindigkeit der Analogeingabe ist bei Assemblerprogrammen auch von der Konversionszeit des ADC abhängig, dessen Auflösung 12 bit beträgt. Diese Verzögerung beträgt bei dem verwendeten Modell 16 µsec, die Eingaberate beträgt ca. 20 kHz.

FORTTRAN-Programmierung verlängert die Exekutionszeiten um etwa eine Zehnerpotenz.

Die angeführten Raten erwiesen sich bisher auch für zeitkritische Anwendungen als ausreichend, sodaß CAMAC derzeit die gesamten Prozeßanwendungen in der Gußhausstraße abdeckt.

ERGÄNZENDE HINWEISE

Die im obigen Punkt angegebenen Geschwindigkeiten sind folgendermaßen zu verstehen: Die angeführten Maximalraten sind nur bei Beschränkung auf die absolut notwendige Anzahl von Assemblerbefehlen zu erreichen. Bei Einfügen von zusätzlichen Befehlen (etwa zur Zwischenverarbeitung) sind pro Assemblerstatement überschlagsmäßig 4 µsec zu veranschlagen.

Außerdem ist zu beachten, daß das Interface zur Haus-Ringleitung eine Grenzfrequenz von ca. 2 kHz hat und somit zeitkritische Anwendungen nur in Rechnernähe sinnvoll sind.

ZUSAMMENFASSUNG

In übersichtlicher Form sieht das Ergebnis folgendermaßen aus:

Werte in kHz	Maximale Rate	Hardware-revision	FORTTRAN	Ringleitung
Digital Aus	70	-	6	2
Digital Ein	10	60	6	2
Analog Aus	70	-	6	2
Analog Ein	20	-	4	2

Man erkennt: Die maximalen Geschwindigkeiten der Ein/Ausgabe betragen in der Regel (außer Analog-Ein) 60 bis 70 kHz und sind i. a. von der Programmierung abhängig.

TERMINALSIMULATION FÜR SATELLITENRECHNER

P. Tinkl

Die Installation zentraler Unterstützungsrechner am PRZ ermöglicht den Instituten, billige Kleinprozeßrechner zu betreiben, ohne gleichzeitig auf wichtige Funktionen großer Rechenanlagen (wie z. B. höhere Programmiersprachen, Speicherung großer Datenmengen, usw.) verzichten zu müssen. Zur Inanspruchnahme der angebotenen zentralen Unterstützungsfunktionen ist es für die Benutzer der Satellitenrechner nötig, u. a. auf den übergeordneten Rechner PDP-11/34 zugreifen zu können (bezüglich des Konzepts der Funktionsaufteilung im Rechnernetzwerk des PRZ siehe Aufsatz auf Seite 9).

Dieses Ziel kann über eine zusätzliche Terminalleitung und eventuell über ein zweites Terminal, das direkt an der PDP-11/34 angeschlossen wird, erreicht werden. Diese Lösung bringt jedoch einen großen hardwaremäßigen Aufwand, sodaß ein anderer Weg beschrritten wurde:

Die Rechnerverbindungsleitung zwischen Satelliten und TP-Rechner wird nur relativ kurze Zeit zum "Downline-Laden" von Systemen und Benutzerprogrammen und eventuell zum Datentransfer von Anwenderprogrammen verwendet. Die restliche Zeit, vor allem, wenn der Benutzer am zentralen Rechner und nicht am Satellit arbeitet, liegt sie brach. Es ist daher nahelegend, diese Leitung sowohl als Rechnerverbindung, als auch als Terminalleitung zur PDP-11/34 zu benützen.

Die Verwirklichung dieses Gedankens ist derzeit für das Betriebssystem RSX-11M an der PDP-11/34 in Arbeit.

Die Aufgabe gliedert sich in drei Teile:

1. Der Kommunikationsdriver von RSX-11M (XLDRV) behandelt im Grundzustand alle Funktionen auf der Leitung. Um jetzt den Terminalbetrieb für den Satellitenrechner zu ermöglichen, muß der Kommunikationsdriver in der PDP-11/34 weggeschaltet werden und der Terminaldriver (TTDRV) an diese Leitung ge-

schalten werden.

Diese Leitung wird jetzt von der PDP-11/34 wie ein Terminal behandelt.

2. Um wieder in den Grundzustand zurückzukehren, ist der umgekehrte Vorgang auf der PDP-11/34 nötig: die beiden Driver sind wieder auszutauschen.
3. Weiters ist ein Programm am Satellitenrechner nötig, das die von der PDP-11/34 empfangenen Signale an das Konsolterminal des Kleinrechners weitergibt bzw. umgekehrt alle am Terminal ausgelösten Zeichen zur PDP-11/34 schickt. Außerdem ist eine Anpassung der verschiedenen Geschwindigkeiten der Verbindungsleitung zur PDP-11/34 einerseits und vom Konsolterminal andererseits nötig.

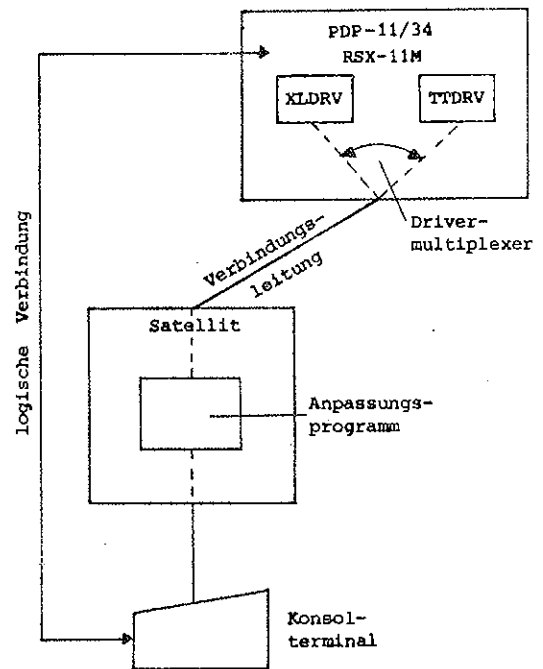


Abb.1 Schema der Simulation eines Terminals der PDP-11/34 an der Konsole eines Satelliten

Ziel der Arbeiten ist somit, nach Laden des Simulationsprogrammes am Satellitenrechner und nach Exekution eines Programmes auf der PDP-11/34, das die Driver in beschriebener Weise multiplext, die Konsole des Kleinrechners als ein Terminal des TP-Rechners benützen zu können.

Durch die Wahl dieser Lösung kann der Hardware- und Kostenaufwand für den Zugriff auf

zentrale Anlagen erheblich verringert werden, wobei sich jedoch keinerlei Einbußen in funktioneller Hinsicht ergeben.

GRAPHICS AM DECSYSTEM-20

F. Brichacek

PROBLEMSTELLUNG

Durch die Verlagerung der Prozeßauswerteprogramme auf den zentralen Unterstützungsrechner DECSYSTEM-20 wurde die Erstellung einer Graphicsoftware und die Installation eines Plotters am zentralen Rechner notwendig.

Dazu kam der Wunsch des Hybridrechenzentrums (HRZ) nach einer Möglichkeit, das dort vorhandene Tektronix-Graphics-Terminal auch in Verbindung mit dem DECSYSTEM-20 verwenden zu können.

Es stellten sich somit eine Reihe von Aufgaben:

1. Erstellung eines Softwarepaketes, welches die Ansteuerung der Graphicsperipherie über Zeichen und Sonderzeichen erlaubt, die der Terminaldriver nicht vollständig zuläßt.
2. Erstellung von Programmen, die den für jede Graphicsperipherie speziellen Charactercode generieren.
3. Realisierung einer Hardwareverbindung mit dem im HRZ vorhandenen Graphics-Terminal.
4. Anpassung der in Macro geschriebenen Graphicsdriver an die Fortranstandardplotterroutinen der Hybridrechenanlage.
5. Auswahl eines für die zentralen Aufgaben geeigneten Plotters.

REALISIERUNG

Als erste Stufe der Realisierung wurde das Softwarepaket zur Ansteuerung der Graphicsperipherie mit Zeichen und Sonderzeichen zusammen mit einem Paket zur Generierung der für das Tektronix-Graphics-Terminal notwendigen Characters erstellt. Parallel hierzu wurde durch Verlegung eines Kabelstranges eine Verbindung mit dem im Hybridrechenzentrum vorhandenen Graphics-Terminal geschaffen.

Diese Softwarepakete zusammen mit der

Hardwareverbindung wurden nun in Zusammenarbeit mit dem HRZ an dem dort vorhandenen Graphics-Terminal mit Hilfe kleiner Fortranprogramme erfolgreich ausgetestet.

Als nächster Schritt werden nun die Fortranstandardplotterroutinen der Hybridrechenanlage, welche schon seit einiger Zeit zum Standard des EDV-Zentrums der TU-Wien geworden sind, an das Macrosoftwarepaket angepaßt und, ebenso wie die im folgenden Artikel beschriebenen Pseudographicsroutinen, allen Kunden zugänglich gemacht.

Die Auswahl eines für die zentralen Aufgaben geeigneten Plotters fiel auf den 4-Farbplotter 9825 A von Hewlett Packard. Ausschlaggebend für die Auswahl dieses Plotters war u. a.:

1. Die Möglichkeit der Erstellung vierfarbiger Graphiken, welche besonders für die Übersichtlichkeit der Prozeßdatenauswertung notwendig ist.
2. Die hohe "Intelligenz" des Plotters (zeichnet und rechnet "in Gedanken" weiter, wenn das Papier zu Ende ist, um dann an genau der richtigen Stelle fortzusetzen, -ob ausgezogen oder strichliert- immer so, wie es das Programm dem eingebauten Mikroprozessor befiehlt).
3. Der -für sein Können- erstaunlich niedrige Preis.

Der Plotter kann Graphiken in der maximalen Größe von A3 erstellen und beinhaltet zur Beschriftung einen eigenen Zeichengenerator.

AUSBlick

Der Plotter, der voraussichtlich Ende Juni geliefert werden wird, kann nach Installation und Anpassung der Steuerzeichenprogramme bis spätestens September von allen Benutzern der PRA verwendet werden. Eine entsprechende Kundmachung wird zum gegebenen Zeitpunkt im Loginfile der 20er zu finden sein.

Eine ausführliche Beschreibung der Plotterroutinen mit ihren neuen Möglichkeiten der Farbauswahl finden Sie im nächsten "Feedback". Wir hoffen, unseren Kunden damit eine weitere

Möglichkeit zur Erleichterung Ihrer Probleme auf dem Gebiet der Prozeßdatenauswertung bieten zu können und stehen zu diesbezüglichen Anfragen gerne zur Verfügung.

PSEUDOGRAPHICS AM DECSYSTEM-20

A. Herrmann

Aus dem Bestreben, dem Kunden alle Möglichkeiten unserer Peripheriegeräte möglichst bequem zugänglich zu machen, und dem häufig geäußerten Kundenwunsch nach flexibler Ausgabe auf dem Bildschirm, entstanden neben den im Aufsatz S.23 beschriebenen Graphicroutinen die im folgenden beschriebenen Pseudographicunterstützungsroutinen für das Bildschirmterminal ADDS Regent 100. Dieses Terminal der neuen Generation bietet, von einem Microprocessor gesteuert, dem Benutzer weit mehr Möglichkeiten als herkömmliche Gerätetypen. So besteht z. B. die Möglichkeit auf einzelnen Stellen den Hintergrund von hell auf dunkel und umgekehrt umzuschalten. Diese Hardwareeigenschaften auszunützen, was bisher nur in Assembler realisierbar war, ist somit auch in Fortran möglich. Diese neuen Fortranmöglichkeiten seien nun kurz umrissen.

1. SEQUENTIELLE AUSGABE:

Bisher war es in Fortran, auf Grund des einheitlichen Bildschirmdisplays, schwierig, einzelne Textstellen besonders hervorzuheben.

Nunmehr ist es dem Programmierer sehr bequem möglich, anstatt der herkömmlichen, einheitlichen Ausgabeform besondere Stellen mit 16 verschiedenen Schreibformen zu betonen. Es bieten sich Blinken, invertierter Hintergrund, halbe Schreibintensität und Unterstreichen sowie alle daraus bildbaren Kombinationen an, um eine Ausgabe stärker zu differenzieren.

2. CURSOR:

Oft stellt sich dem Programmierer das Problem, auf der dargestellten Bildschirmseite an früherer Stelle etwas einfügen oder verändern zu wollen, was ja nicht direkt möglich war, und somit mußte zu Tricks gegriffen werden. Die nunmehr geschaffene direkte Cursoradressierung, die hier abhilft, bietet dem Benutzer die Möglichkeiten

- a. nur den Cursor zu positionieren
(hierbei erfolgt die Positionierung bequem durch Angabe von Zeilen- und

Spaltennummer)

- b. einen Text ab einem bestimmten Punkt mit einem bestimmten Schreibmodus anzuschreiben.

3. PSEUDOGRAPHIC:

Unter Graphic versteht man im allgemeinen die Darstellung von beliebigen Kurven und Punkten, welche das Regent 100 zweifellos nicht bieten kann. Gewisse Einschränkungen sind gegeben. Das Gerät kann jedoch horizontale und vertikale Linien und jegliche Arten von Kreuzungspunkten in 4 verschiedenen Zeichenarten (normal, halbtensiv, blinkend und halbtensiv blinkend) darstellen, unabhängig vom zusätzlich veränderbaren Untergrund (siehe Punkt 1).

Es stehen hierzu Routinen zum Zeichnen von Ketten solcher Zeichen mit frei wählbarer Länge ab einer bestimmten, wählbaren Cursoradresse, in horizontaler wie auch in vertikaler Richtung zur Verfügung.

Ebenso stehen Routinen zur Verfügung, mit denen man das Keyboard sperren und wieder lösen kann.

Die folgende Kurzbeschreibung der Routinen befindet sich im Directory <SUBSYS> des DECSYSTEM-20, zu lesen mit dem Befehl: HELP TTYLIB.
Für spezielle Informationen bitte das PRZ kontaktieren.

ANHANG:

Printout des Files TTYLIB. HLP

PSEUDOGRAPHIC ON SYS20-REGENT 100

Linking instructions:

```
LOAD XXXX,SYS:TTYLIB/SEARCH      OR
EXECUTE XXXX,SYS:TTYLIB/SEARCH
```

```
also possible :
instead of TTYLIB/SEARCH ...@TTYLIB
```

Available routines:

```
VIDEO(Mode)
change video mode

VLOCK.          Lock keyboard
UNLOCK         Unlock keyboard
ONLINE         Set terminal into online-mode
```

```

CURSOR(Line,Column)
    position cursor

STRING(Line,Column,Mode,'Text')
    display 'Text'

HLINE(Line,Column,Number,Gra-mod,Sign)
    display horizontal graphic line L -> R

HRLINE(Line,Column,Number,Gra-mod,Sign)
    display horizontal graphic line L <- R

VLINE(Line,Column,Number,Gra-mod,Sign)
    display vertical graphic column
    Top -> Bottom

VRLINE(Line,Column,Number,Gra-mod,Sign)
    display vertical graphic column
    Top <- Bottom

STATUS(Line,Column,Char,Stat1,Stat2)
    returns current cursor position
    and there displayed character

FF                makes a formfeed on TTY;

STB
    Waits for one byte to be entered

WAITMS(Time)
    Waits for Time milliseconds

```

Range and Meanings of the Parameters:

- Mode 1 .. 16
- 1 ... normal
 - 2 ... underline
 - 3 ... blink
 - 4 ... blink, underline
 - 5 ... reverse (background)
 - 6 ... reverse, underline
 - 7 ... reverse, blink
 - 8 ... reverse, blink, underline
 - 9 ... half (intens)
 - 10 ... half, underline
 - 11 ... half, blink
 - 12 ... half, blink, underline
 - 13 ... half, reverse
 - 14 ... half, reverse, underline
 - 15 ... half, reverse, blink
 - 16 ... half, reverse, blink, underline
- Line 1 .. 24
- Column 1 .. 80
- Gra-mod 1 .. 4
- 1 ... normal
 - 2 ... half intens
 - 3 ... blink
 - 4 ... half intens, blink
- Sign 1 .. 11
- 1 ... Lower right corner
 - 2 ... Lower left corner
 - 3 ... Upper right corner
 - 4 ... Upper left corner
 - 5 ... Bottom intersect
 - 6 ... Left intersect
 - 7 ... Right intersect
 - 8 ... Top intersect
 - 9 ... Horizontal line
 - 10 ... Vertical line
 - 11 ... Cross lines

For more detailed information read
HLP:TTYLIB.MEM

IEE-488-BUS AM ZENTRALEN PROZESSRECHNER DES PRZ

F. Brichacek

1. EINFÜHRUNG

1.1 Was ist der IEE-488-Bus

Der IEE-488-Bus, auch IEC-Bus genannt, ist ein genormtes Meßgerätebusssystem, welches mit dem des bekannten Meßgeräteherstellers Hewlett-Packard kompatibel ist. Es erlaubt den Anschluß von maximal 15 Meßgeräten an den aus 8 Daten- und 8 Steuerleitungen bestehenden Bus.

Ein Vorteil gegenüber dem im Feedback Heft 4 beschriebenen Camac-System besteht in der Möglichkeit der verteilten Aufstellung der Meßgeräte im Meßraum und in der durch die wegfallende räumliche Begrenzung durch das Crate möglichen aufwendigeren Gestaltung (Eingangsspannungsschutz, variable Meßbereiche etc.).

1.2 IEE-488 am PRZ

Auf Wunsch mehrerer Benützer wurde für den Zentralen Prozeßrechner PDP-11/34 im Bereich Gußhausstraße ein IEC-Businterface zusammen mit einem Digitalvoltmeter 3455 A von Hewlett Packard installiert und die dazugehörige Software entwickelt.

Damit steht für alle Kunden des Prozeßrechenzentrums ein Muster der Meßwerterfassung über ein IEC-Bussystem zur Verfügung.

1.3 Allgemeine Beschreibung

Das IEC11-A Interface erlaubt die Kommunikation zwischen dem PDP-11 UNIBUS und Geräten mit IEC-Busanschluß. Das IEC11-A entspricht den Spezifikationen "IEC 66.22" des "International Electrotechnical Committee" (IEEE Standard 488-1975). Der Controller besteht aus einem Hexboard, welches nur einen SPC-Slot einer DD-11 Systemeinheit benötigt. Die Verbindung zum IEC-Bus erfolgt durch einen der zwei BERG-Stecker des Controllers. Der andere Stecker ist für das DMA-Interface oder ein Testgerät vorgesehen. Ein spezielles Adapter-

kabel bildet den Übergang von BERG-Stecker zum Standard-IEC-Busstecker (Amphenol 17-20250).

Das IEC11-A Interface erfüllt folgende IEC-definierte Funktionen:

- a. System Controller
- b. Controller-in-charge
- c. Talker
- d. Listener

Jedes IEC-Bussystem darf nur einen einzigen Systemcontroller haben, welcher IFC und/oder REN auf dem Bus aktivieren kann. Mit dem Signal IFC (Interface Clear) kann der Controller den Bus in den Grundzustand versetzen.

Ein IEC-Bussystem kann mehrere Controller haben, aber nur einer von ihnen darf als "Controller-in-charge" erklärt sein. Dieser kann im Aktivzustand sowohl Kommandos als auch Adressen auf den Bus ausgeben.

Die "Talker-Funktion" kann durch Controllerkommandos aktiviert werden. Wenn sie aktiviert ist, sendet der "Talker" Datenbytes vom IEC11-A zum IEC-Bus.

Die "Listener-Funktion" kann durch den "Controller-in-charge" ebenfalls aktiviert werden. Wenn sie aktiv ist, überträgt der "Listener" Datenbytes vom IEC-Bus zum IEC11-A. Das IEC11-A kann "Parallel Poll" (PP) durchführen und sowohl auf "Service Request" (SR) reagieren, als auch SR aktivieren. Zusätzlich kann das Interface die Funktionen "Remote/Local" (RL), "Device Clear" und "Device Trigger" (DT) in den Einheiten, welche mit dem Bus verbunden sind, aktivieren. Die IEC-Businterfacefunktionen, welche durch das IEC11-A benützt werden, sind in 1.5 angeführt.

Das IEC11-A verständigt sich mit der PDP-11 unter Programmkontrolle mit Interrupts. UNIBUS-adress, Interruptvector und IEC-Geräteadresse sind mit Mikroschaltern einstellbar.

1.4 System Konfiguration

Abb. 1 zeigt die Systemkonfiguration des IEC11-A. An den Bus können bis zu 15 Geräte (inklusive IEC11-A) mittels 25-poligem Standardstecker angeschlossen werden.

Abb. 2 und Tabelle 1 zeigen die IEC-Bus-signale und ihre Funktionen.

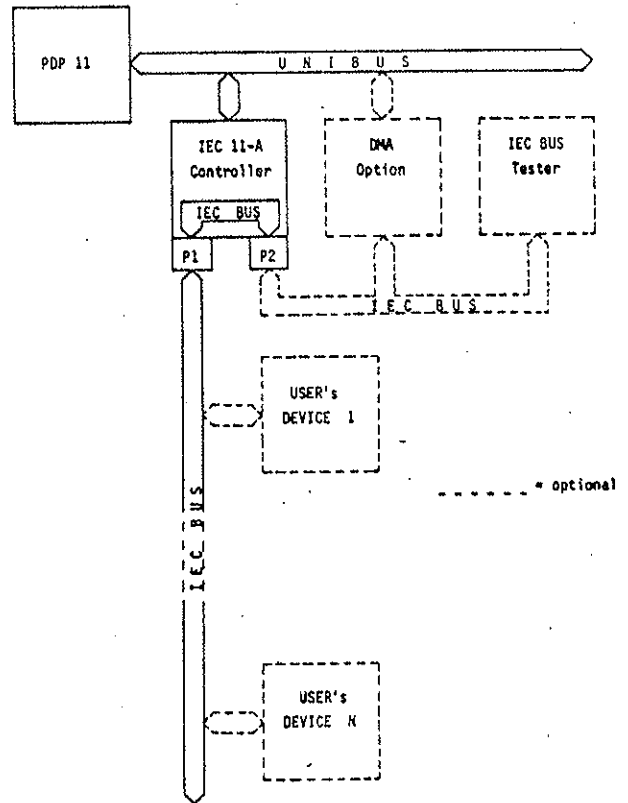


Abb. 1 System Konfiguration

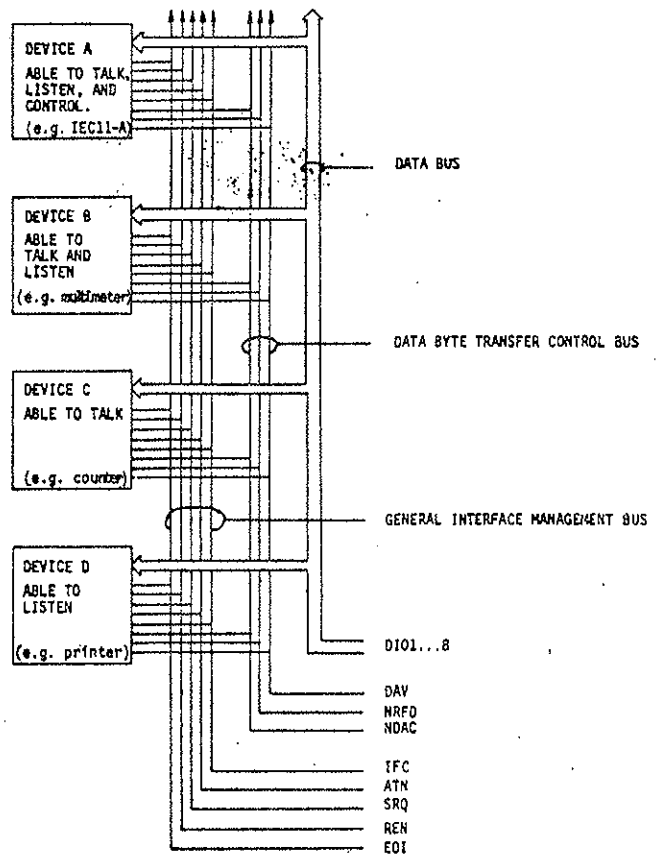


Abb. 2 IEC Bus Struktur

Tabelle 1 IEC-Busleitungen

MNEMONIC	GRUPPE	FUNKTION
DIO1 DIO8	Datenbus (8 Leitg.)	Dienen zur Übertragung aller Daten, Adressen und Steuerbefehle.
DAV	Daten Transfer Kontroll- Bus (3 leitg.)	Data Valid. Zeigt das Vorhandensein und Gültigkeit von Daten auf den DIO's.
NRFD		Not Ready For Data. Zeigt Bereitschaft der Geräte für Daten an.
NDAC		Not Data Accepted. Zeigt die beendete Übernahme der Daten durch Device an.
IFC	Allgem. Interface Manage- ment Bus (5 Leitg.)	Interface Clear. Kann nur vom Controller-in-charge gesetzt werden, um alle Interfaces in Grundstellung zu bringen.
ATN		Attention. Kennzeichnet Adressen und Steuerbefehle auf den DIO's.
SRQ		Service Request. Kann von einem Device gesetzt werden, um den Controller-in-charge zu einem Interrupt zu veranlassen.
REN		Remote Enable. Wird in Verbindung mit anderen Befehlen verwendet, um Geräte in ihren Remotezustand zu versetzen. (Sperrung Gerätefrontplattenbedienung) Nur durch den Controller-in-charge!
EOI		End or Identify. Dient zur Kennzeichnung des Endes einer Blockübertragung oder zur Identifizierung von Geräten, die eine SRQ-Anfrage gestartet haben. Muß in Verbindung mit ATN betrachtet werden.

Tabelle 2 ATN-EOI Verknüpfung

ATN	EOI	DIO
∅	∅	Normales Datenbyte
∅	1	Letztes Datenbyte eines Blockes (END)
1	∅	Adresse, Universalbefehl, Adressierter Bef.
1	1	Aufforderung zur Identifizierung nach SRQ.

1.5 Spezifikationen

a. Mechanisch

Boards: 1 Hex Board
 Platz: 1 SPC Slot in einer DD11 System Unit

Arbeitstemperatur: 0°C bis 55°C

Relative Feuchtigkeit: 95 % (ohne Kondensation)
 Gewicht: 1 kp
 Buslänge: maximal 20 m

b. Elektrisch

Strombedarf: 2.5 A /+5 V
 Logic Pegel: TTL
 Unibus Loads: 1
 Anzahl maxim. IEC-Geräte: 15

c. Operationen

Register: Vier 16-bit Register
 Operating Mode: Programmübertragung mit Interrupt

Interrupt

Level: wählbar (6, wenn nicht anders spezifiziert)

IEC-Interface Talker (T2)

Funktionen Listener (L1)

(Subsets): Acceptor Handshake (AH1)
 Source Handshake (SH1)
 Control (C1, 2, 3, 4, 5)
 Service Request (SR1)

1.6 Ergänzende Literatur

- a. PDP-11 Peripherals and Interfacing Handbook
- b. IEC 66.22 Specifications (siehe Einleitung)

1.7 Adaptierung

Zu Beginn müssen die UNIBUS-Adresse sowie der Interruptvektor und die IEC-Device-Adresse eingestellt werden. Ein anderer Microschalter erlaubt die dauernde "Listener" Einstellung (siehe 2.2.1).

2. OPERATIONEN UND PROGRAMMIERUNG

2.1 Kontrollschalter

Neben den Wahlschaltern für die Adressen

und den Interruptvektor gibt es noch einen Schalter, welcher die Einstellung "Listen only" erlaubt (siehe Tab. 4).

2.2 Unibusregister

Das IEC11-A enthält vier programmierbare Register, deren Basisadresse frei wählbar ist (siehe 1.5). Die relativen Adressen zwischen den Registern sind allerdings fix. Nur die erste Registeradresse muß mit den Wählschaltern eingestellt werden. Tabelle 3 zeigt die verfügbaren Register.

ADRESSE	REGISTER
76xxx0	Control & Interrupt Register CIR
76xxx2	State & Message Register SMR
76xxx4	Input/Output Register IOR
76xxx6	Vector Switch Register VSR

Tabelle 3 IEC11-A Register

Die folgende Zeichnung zeigt, wie die UNIBUS-Adresse gebildet wird.

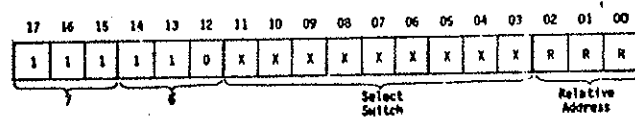


Abb. 3 UNIBUS-Adressenarchitektur

Im folgenden wird jedes Registerbit durch folgende Kurzzeichen im Detail beschrieben:

- R = Read-only Bit; kann nicht gesetzt werden.
- W = Write-only Bit; wird immer als 0 gelesen.
- R/W = Read/Write Bit; kann gesetzt und gelesen werden.
- R/C = Read/Clear Bit; wird nach dem Lesen gelöscht, kann nicht gesetzt werden.
- N.A. = Not applicable.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
DATA ACC	SQR	EOI	ILL MSGE	NO LAC	BLOCK SRQ	NRFD LINE	STATE CHGE	INT	INT ENB	MC	RSV	BLOCK DAC	LAST BYTE	LON	SACS
R/C	R	R/C			R/W	R	R/C	R	R/W	W		R/W	W	R	R/W

Das CIR beinhaltet alle Interrupt-Bedingungen und einige spezielle control-Bits.

Tab. 4 CIR Bit Zuordnung

MNEMONIC	BIT	FUNKTION
SACS	φ	System Control Active State. Wenn es gesetzt ist, kann das Interface SIAS und/oder SRAS (SMR) aktivieren. SACS wird durch ein UNIBUS INIT oder MC (Bit 5) gelöscht. SACS darf während eines Transfers nicht geändert werden.
LON	1	Listen only. Zeigt die Stellung des internen LON-Schalters an. Dieser Schalter kann gesetzt sein, um das Interface in den dauernden "Listener"-Zustand zu versetzen, ohne als "Listener" adressiert worden zu sein.
LAST BYTE	2	Last Byte. Wenn der Talker aktiv ist (TACS), kann LAST BYTE vor dem Laden des letzten Bytes gesetzt sein, um die EOI line des IEC Busses zu setzen. Dies zeigt dem Controller-in-charge an, daß das letzte Datenbyte vom Talker gesendet wurde.
BLOCK DAC	3	Block DAC. BLOCK DAC verhindert ein Entstehen von DATA ACC (Bit 15) durch Interrupts (Bit 7). Es kann z. B. benützt werden, wenn der Listener aktiv ist (LACS), um TCS (SMR) zu erlauben, wenn keine Daten in den Speicher übertragen werden sollen. Zusätzlich darf der Inputbuffer nicht gelesen werden, um ein "acceptor-handshake" des Listeners zu verhindern, da BLOCK DAC ein internes "rdy" (ready for next Byte) simuliert. BLOCK DAC wird durch ein UNIBUS INIT oder MC (Bit 5) gelöscht.
RSV	4	Request Service. Versetzt die SRQ-Sendung des IEC-Busses in ihren Aktivzustand, wenn es gesetzt ist. Die Leitung bleibt aktiv, bis SPAS (SMA Bit 15) gesetzt wird.
MC	5	Master Clear. Erlaubt das selektive Zurücksetzen des IEC11-A's ohne irgendein anderes Device am UNIBUS zu beeinflussen. Im Controller hat MC denselben Effekt wie ein UNIBUS-INIT Signal.
INT ENB	6	Interrupt Enable. Wenn es gesetzt ist, kann INT (Bit 7) einen Programminterrupt hervorrufen. INT ENB wird durch das INTERRUPT DONE Signal nach jedem Interrupt zurückgesetzt. Dadurch muß es jedesmal neu gesetzt werden, um weitere Interrupts zu erlauben. Außerdem wird INT ENB durch UNIBUS INIT oder MC gelöscht.
INT	7	Interrupt. Dieses Bit ist die ODER-Verbindung von allen Bits im "high-order Byte des CIR" (ausgenommen Bit 9 und φ). Ist es gesetzt, verursacht es einen Programminterrupt, vorausgesetzt das INT ENB ist gesetzt.
STATE CHGE	8	State Change. Zeigt eine Veränderung im "high order byte" des SMR durch eine lokale oder IEC-Bus Meldung an. STATE CHGE wird durch ein UNIBUS INIT oder ein MC gelöscht.
NRFD LINE	9	Not Ready For Data. Zeigt den Zustand der NRFD-Leitung am IEC-Bus an.
BLOCK SRQ	10	Block SRQ. Es verhindert INT durch ein SRQ (Bit 14). Durch ein UNIBUS INIT oder MC wird es gelöscht.
NO LAC	11	No Listener Active. Zeigt an, daß kein Listener aktiv ist, nachdem ATN durch den Controller-in-charge weggenommen wurde, sodaß ein Talker keine Daten senden kann. NO LAC wurde durch UNIBUS INIT oder MC gelöscht.
ILL MSGE	12	Illegal Message. Wird gesetzt, wenn versucht wird, eine lokale Meldung ins SMR zu bringen, während der verwendete Interfacestatus nicht aktiv ist. ILL MSGE wird durch UNIBUS INIT oder MC gelöscht.
END	13	End of Block. Zeigt, daß ein Talker die EOI-line während des letzten Datentransfers am IEC-Bus aktiviert hat. END wird durch UNIBUS INIT oder MC gelöscht.
SQR	14	Service Request. Zeigt an, daß ein Device die SRQ-line am IEC-Bus aktiviert hat, um eine Beantwortung durch "Serial Polling" zu erreichen. SRQ wird durch das anfragende Device, nachdem der Controller-in-charge ein "Serial Poll" durchgeführt hat, gelöscht. SRQ kann INT (Bit 7) setzen, wenn BLOCK SRQ (Bit 10) nicht gesetzt ist.
DATA ACC	15	Data Accepted. Zeigt an, daß ein IEC-Bus Zyklus beendet ist. Zu diesem Zeitpunkt kann ein neues Byte unter Berücksichtigung des Interfacestatus im SMR vom IOR geladen oder gelesen werden.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	
SPAS	SRAS	SIAS	LACS	TACS	CPPS	CSBS	CACS	SRE	SIC	LUN	LTN	RPP	GTS	TCA	TCS	
R								R/W	W			R/W	W			
interface Status								local Message								
Tab. 5 SMR																
MNEMONIC	BIT	FUNKTION.														
TCS	0	Take Control Synchronously. Wenn es gesetzt wird, bringt es den Controller, wenn LACS aktiv ist, aus dem CSBS-Zustand in den SACS (Controller Active State) Zustand.														
TCA	1	Take Control Asynchronously. Wenn es gesetzt wird, bringt es den Controller in den CACS-Zustand, wobei LACS nicht aktiviert sein muß, wohl aber CSBS. Bei der Benützung von TCA kann ein Datentransfer am IEC-Bus gestört werden, wenn TCA aktiviert wird.														
GTS	2	Go to Standby. Das Setzen dieses Bits veranlaßt den Controller in den Standbyzustand überzugehen (CSBS). Es kann nur gesetzt werden, wenn CACS aktiv ist, ansonsten wird ILL MSGE im CIR gesetzt.														
RPP	3	Request Parallel Poll. Wenn es gesetzt wird, veranlaßt es den Controller den "Parallel Poll State" (CPPS) zu aktivieren. Es kann ebenfalls nur aktiviert werden, wenn CACS aktiv ist, ansonsten wird ILL MSGE im CIR gesetzt. RPP wird durch UNIBUS INIT oder MC gelöscht.														
LTN	4	Listen. Durch Setzen von LTN wird der Listener direkt adressiert, ohne daß die Listenadresse am IEC-Bus gesetzt werden muß. LTN kann nur gesetzt werden, wenn CACS gesetzt ist, ansonsten wird ILL MSGE im CIR gesetzt.														
LUN	5	Local Unlisten. Durch Setzen von LUN wird die Adressierung des Listeners gelöscht, ohne daß am IEC-Bus die Unlistenadresse gesetzt werden muß. LUN kann nur gesetzt werden, wenn CACS gesetzt ist, ansonsten wird ILL MSGE im CIR gesetzt.														
SIC	6	Send Interface Clear. Durch Setzen von SIC wird SIAS aktiviert. SIC kann nur gesetzt werden, wenn CACS gesetzt ist, ansonsten wird ILL MSGE im CIR gesetzt.														
SRE	7	Send Remote Enable. Durch Setzen von SRE wird SRAS aktiviert. SRE kann nur gesetzt werden, wenn CACS gesetzt ist, ansonsten wird ILL MSGE im CIR gesetzt.														
CACS	8	Controller Active State. Wenn es gesetzt ist, zeigt es an, daß der Controller im Aktivzustand ist und die ATN Leitung gesetzt hat. Wenn CACS gesetzt ist, kann der Controller Adressen und Befehle durch Laden des Output-Buffers über den IEC-Bus senden. CACS kann sowohl durch TCA als auch durch TCS als auch durch die Übertragung von TCI (Take Control) gesetzt werden. Der Controller verläßt den CACS-Zustand, wenn GTS gesetzt wird. CACS wird durch UNIBUS INIT und MC gelöscht.														
CSBS	9	Controller Standby State. Wenn es gesetzt ist, zeigt es an, daß das IEC11-A der "Controller in-charge" ist, aber es aktiviert nicht die ATN Leitung am Bus. In CSBS-Zustand kann das Interface nur Daten senden, wenn der Talker adressiert ist. CSBS wird aktiviert, nachdem GTS gesetzt wurde. Der Controller verläßt den CSBS-Zustand durch Setzen von TCA oder TCS. CSBS wird durch UNIBUS INIT oder MC gelöscht.														
CPPS	10	Controller Parallel Poll State. Wenn es gesetzt ist, zeigt es, daß der Controller ein "Parallel Poll" durchführt, dazu aktiviert er ATN & IDY am IEC-Bus und empfängt ein PPR (parallel poll response) von einem Device. Der PPR wird am Inputbuffer des IOR geladen. Während CPPS aktiv ist, darf keine Meldung gesendet werden. Der Controller geht durch Setzen von RPP in den CPPS-Zustand über und verläßt ihn durch Löschen von RPP wieder.														
TACS	11	Talker Active State. Wenn TACS gesetzt ist, ist der Talker im Aktivzustand. TACS wird durch Übertragung der Talkeradresse am IEC-Bus und anschließenden Übergang des "Controller-in-charge" in den "standby state" aktiviert.														
LACS	12	Listener Active State. Wenn LACS gesetzt ist, befindet sich der Listener in seinem Aktivzustand. LACS wurde durch Übertragung der Listeneradresse über den Bus oder durch Setzen von LTN im CACS-Zustand aktiv.														
SIAS	13	Interfaces Clear Active State. Wenn es gesetzt ist, zeigt es, daß der Controller die IFC-Leitung am Bus gesetzt hat. SIAS wird durch Setzen von SIC für ca. 100 ns aktiviert.														
SRAS	14	Remote Enable Active State. Zeigt an, daß der Controller die REN-Leitung aktiviert hat. SRAS wird ebenso durch UNIBUS INIT als auch durch MC gelöscht.														
SPAS	15	Serial Poll Active State. Zeigt an, daß der Talker ein Statusbyte senden muß. SPAS wird aktiviert, nachdem der Talker die SPE (Serial Poll Enable) Meldung und seine Talker übergeht. SPAS wird durch IFC, UNIBUS INIT, MC und eine andere Talkeradresse gelöscht.														

UNTERSTÜTZUNGSSOFTWARE

FILE: IEC11X.MAC
UIC: E20,1333
DATE: 06-DEC-77

VERS/EDIT: V0,001
EXEC VERS: RSX11M V03
SOURCE LNG: MACRO-11

WRITTEN: F.J.BRICHACEK

PURPOSE: FORTRAN CALLABLE IEE-488 SUPPORT

MIN CONFIG.: PDP-11 (RSX11M)

TRANSLATION AND LINKING INSTRUCTION:

MAC>IEC11X,II:=IEC11X

>TKB
TKB>XXXX=XXXX,E20,1333||IEC11X
TKB>/
ENTER OPTIONS:
TKB>COMMON=CACOM:RW
TKB>//

EIGHT IEC-BUS FUNCTIONS FOR THE BUS-CONTROLLER

- FMT(0) SIC (SEND INTERFACE CLEAR)
- (1) ATN LINE LOW (CMD-MODE)
- (2) ATN LINE HIGH (DATA-MODE)
- (3) REN LINE LOW (REMOTE-MODE)
- (4) REN LINE HIGH (LOCAL MODE)
- (5) CONTROL FLAG TRUE (CONTROLLER MODE)
- (6) CONTROL FLAG FALSE (NON CONTROLLER-MODE)
- (7) ATN LINE LOW (CMD-MODE, SYNCHRONOUSLY)
- (8) MCL (MASTER CLEAR OF INTERFACE)

ALSO POSSIBLE INSTEAD OF:

- FMT(0)....SIC (SEND INTERFACE CLEAR)
- FMT(1)....TCA (CMD-MODE, ASYNCHRONOUSLY)
- FMT(2)....GTS (GO TO STANDBY, DATA-MODE)
- FMT(3)....SRE (SEND REMOTE ENABLE)
- FMT(4)....NRE (NOT REMOTE ENABLE)
- FMT(5)....SAC (SET SYSTEM CONTROL ACTIVE STATE)
- FMT(6)....NSA (RESET SACS)
- FMT(7)....TCS (TAKE CONTROL SYNCHRONOUSLY)
- FMT(8)....MCL (MASTER CLEAR OF INTERFACE)

READ BYTE AND WRITE BYTE SUBROUTINE'S

RDB(N,IF) N...NUMBER OF BYTES TO BE READ INTO OR
WTB(N,IF) WRITE ON ARRAY IF.

REALISIERUNG VON HDLC AUF PROZESSRECHNERN VOM TYP PDP-11

W.Kunft

In den letzten Jahren nahm die Entwicklung von Computernetzen in Europa einen ungeheuren Aufschwung. Unter den verschiedenen Methoden, Nachrichten in einem Rechnernetz zu transportieren, hat vor allem das sogenannte Packetswitching besondere Bedeutung erlangt. Bei diesem Verfahren werden Nachrichten in Pakete aufgespaltet. Diese werden auf dem günstigsten Weg durch das Netz zum Empfänger transportiert und dort wieder zur ganzen Nachricht zusammengefügt.

Damit öffentliche paketschaltende Netze der Post auf nationaler und internationaler Ebene von den Teilnehmern benützt werden können, ist eine Standardisierung notwendig geworden.

CCITT hat in der Empfehlung X.25 eine Schnittstelle zu einem öffentlichen paketschaltenden Computernetz definiert. Der Level 2 von X.25 enthält die Definition der Leitungsprozedur (Link Access Procedure LAP) zwischen Teilnehmer (DTE) und Netz (DCE). LAP verwendet ausschließlich Elemente der High Level Data Link Control Procedure (HDLC) von ISO. LAP sieht eine Übertragung im Duplex-Mode vor, verwendet den Asynchronous Response Mode (ARM) von HDLC und definiert ein Subset der in HDLC zur Verfügung stehenden Commands und Responses zur Regelung des Informationsaustausches zwischen DTE und DCE.

Für Prozeßrechner des Typs PDP-11 steht zwar Hardware zur Verfügung, mit der das bitorientierte Leitungsprotokoll HDLC abgewickelt werden kann, jedoch existiert für diesen Zweck derzeit noch keine Software von DEC.

Daher wurde am PRZ HDLC nach den Vorschriften der X.25 Link Access Procedure auf einer PDP-11/34 unter RSX-11M und auf einer PDP-11/04 unter RSX-11S implementiert.

Bei der Entwicklung der Software wurden folgende Ziele angestrebt:

- 1) X.25 - LAP
- 2) Unterstützung von bis zu 16 Leitungen
- 3) Empfang soll zu jedem beliebigen Zeitpunkt möglich sein

- 4) möglichst geringer Systemoverhead
- 5) minimale Zeiten, in denen das Interruptsystem blockiert ist und damit gutes Echtzeitverhalten
- 6) Realisierung der Software auf Rechnern mit und ohne Memory Management Hardware
- 7) modularer Aufbau

Zur Erfüllung der Forderungen 1) und 7) wurden aus der Definition von X.25 Level 2 einzelne Zustände des Kommunikationsprozesses abgeleitet. Diesen Zuständen wurden Zustandsmodule zugeordnet, die tabellengesteuert abgearbeitet werden. [1] Jeder Zustandsmodul muß drei logische Funktionen erfüllen:

- a) Erkennungsfunktion
- b) Ausgabefunktion
- c) Übergangsfunktion

Die Erkennungsfunktion erfordert eine Selektion eines eintretenden Ereignisses aus der Menge der in diesem Zustand möglichen Ereignisse. Die Ausgabefunktion besteht in der Durchführung bestimmter, dem jeweiligen Ereignis zugeordneter Aktivitäten (z.B. Ausgabe einer bestimmten Kontrollnachricht, Starten eines Timers usw.) Die Übergangsfunktion regelt schließlich einen durch ein bestimmtes Ereignis notwendig gewordenen Zustandswechsel.

Der übliche Weg, solche Zustandsmodule zu implementieren, setzt die Entwicklung eines Linedrivers nach den Konventionen des Betriebssystems voraus. Dieser enthält eine Interruptservice-routine und wird von den Zustandsmodulen unter Verwendung der durch das Betriebssystem gegebenen Synchronisationsmechanismen zur Ein-Ausgabe von Nachrichten über die Leitung herangezogen.

Da die Datenstruktur der Betriebssysteme RSX-11M und RSX-11S für die Realisierung eines Communicationsdrivers für Duplexbetrieb schlecht geeignet ist und durch eine solche Lösung die Forderungen 3) und 4) nur unzureichend erfüllt werden können, wurde ein anderer Weg beschritten.

Die gesamte Abwicklung der Link Access Procedure nach X.25 wird durch einen privilegierten Task vorgenommen, der die erforderlichen Zustandsmodule und auch die Interruptservice-routinen für Senden und Empfang enthält und die

Routinen des Betriebssystems sowie die Dynamic Storage Region (DSR) direkt benützt. Die einzelnen Zustandsmodule können unter Verwendung des sogenannten "Forkprocessings" von RSX-11M bzw. RSX-11S direkt im Systemstatus als Systemprozesse und damit gleichberechtigt mit allen anderen Betriebssystemfunktionen abgewickelt werden. Die Technik des Forkprocessings bietet die Möglichkeit, weniger zeitkritische Teile von Interruptserviceroutinen, die meist auch Zugriffe auf gemeinsame Datenstrukturen enthalten, in eine Warteschlange zu stellen und voll unterbrechbar, völlig asynchron zu den auslösenden Ereignissen noch vor dem Task-scheduling abzuarbeiten. Durch diese "Serialisierung" der Verarbeitung der eintretenden Interrupts wird auch eine Synchronisation der Zugriffe auf gemeinsame Datenstrukturen erreicht.

Durch die Anwendung dieser Technik zur Realisierung der Zustandsmodule können die Forderungen 4) und 5) erfüllt werden.

Eine einschneidende Zielvorstellung ist die Forderung 3). Sie kann nur mit Einschränkungen erfüllt werden, da in jedem Computer die Pufferkapazität beschränkt ist. Für den Fall, daß eintreffende Nachrichten nicht angenommen werden können, weil kein Platz für einen weiteren Empfangspuffer verfügbar ist, sind die "Receiver Not Ready" (RNR) bzw. "Receiver Ready" (RR) Mechanismen von LAP anzuwenden. Stehen jedoch Puffer zur Verfügung, dann müssen sie auf schnellstem Wege der Interruptserviceroutine für den Empfang zugänglich gemacht werden.

Dieses Problem wurde so gelöst, daß der "Auftraggeber-Task", der über eine bestimmte HDLC-Leitung kommunizieren will, einen "Buffer-Pool" zur Verfügung stellt. Dieser Pool ist durch eine "Free-Buffer-Queue" strukturiert. Die Zustandsmodule sorgen dafür, daß die Interruptserviceroutinen immer mindestens einen Empfangspuffer aus diesem Pool zur Verfügung gestellt bekommen. Die Zustandsmodule können direkt auf das Poolmanagement des Auftraggebertasks zugreifen.

So kann der konventionelle Weg über den Mechanismus der Asynchronous System Traps (AST) oder Eventflags, ein anschließendes Task-scheduling und ein neuerlicher Aufruf des Drivers

umgangen werden und die Zeit, die es dauert, bis ein neuer Empfangspuffer zur Verfügung gestellt werden kann, wesentlich verkürzt werden. Erst wenn dieser Buffer Pool erschöpft ist, wird vom aktuellen Zustandsmodul eine RNR-Response an den Partner abgegeben.

Die Forderung 2) wird in analoger Weise wie bei RSX-11M oder RSX-11S Drivers durch eine entsprechende Gestaltung der Datenstruktur, die für jede Leitung getrennte Arbeitsbereiche vorsieht, erfüllt.

Besonderes Augenmerk wurde auch auf die Erfüllung der Forderung 6) gerichtet. HOST-Rechner, die als DTE nach X.25 über ein paketschaltendes Netz (DCE) mit einem anderen HOST kommunizieren, sind in der Regel gut ausgebaute Rechner mit Memory Management Hardware und mehr als 28kW Speicherkapazität. Der unmittelbare "Gesprächspartner" auf der anderen Seite der X.25 Schnittstelle, der gleichsam als Eintrittstor zum paketschaltenden Netz fungiert, hat lediglich die Aufgabe des Informationstransportes und wird in der Regel ein weniger ausgebautes und damit billigerer Computer sein, der keine Memory Management Hardware besitzt.

Diese beiden Möglichkeiten wurden bei der Softwareentwicklung durch Ausnützung des Conditional Assembly des Macroassemblers der PDP-11 berücksichtigt. Bei Rechnern ohne Memory Management Hardware (z.B. PDP-11/04) ist jeder einzelne Speicherplatz direkt adressierbar. Ist jedoch Memory Management Hardware eingebaut, muß diese entsprechend programmiert werden, daß jederzeit die Umsetzung von virtuellen Adressen in physikalische Adressen richtig durchgeführt wird. Dies ist in den Interruptserviceroutinen und Zustandsmodulen zu berücksichtigen, die ja den Adreßraum eines anderen Tasks ansprechen. Außerdem ist zu beachten, daß alle Zustandsmodule wie die Interruptserviceroutinen als Systemprozesse im sogenannten "Kernelmode" ablaufen und daher die Adreßumsetzung über einen gesonderten Registersatz erfolgt.

Die neuentwickelte Software bietet nicht nur die Möglichkeit, innerhalb des Rechnernetzes des PRZ duplex mit geringem Protokolloverhead Informationen auszutauschen, sondern sie stellt auch die Basis für die Implementation von X.25 Level 3 und von Protokollen noch höherer Hierarchiestufen zur Kommunikation zwischen Prozes-

son in verschiedenen Rechnern über ein dazwischenliegendes paketschaltendes Netzwerk dar.

Literatur:

- [1] P.JILEK, H.-J. SEIDEL: Realisierung von Datenübertragungsverfahren auf der Grundlage von Zustandsdiagrammen sequentieller Prozesse; Elektron. Rechenanl. 16 (1974), H.2, S.52-60

STEUERUNG EINER PARABOLANTENNE MIT HILFE EINES PROZESSRECHNERS NACH VORGEGEBENEN POSITIONIERUNGSDATEN

P. Lorenz

PROBLEMSTELLUNG

Mittels des Prozeßrechners PDP-11/34 des PRZ wurde die Positionierung einer Parabolantenne des Instituts für Hochfrequenztechnik nach zuvor berechneten Positionierungsdaten vorgenommen. Zweck der Einrichtung ist der Empfang der Daten von Wettersatelliten der NOAA-Serie. Eine Optimierung des Empfangs ergibt sich durch eine Optimierung der erreichbaren Positionierungsgenauigkeit.

Im Zuge des Austausches der IBM 1800 hat sich die Notwendigkeit ergeben, das bereits existierende Antennensteuerprogramm dem neuen System anzupassen.

Weiters bestand nun auch die Möglichkeit, das vorhandene Programm ZIPSAT (Berechnung der Positionierungsdaten) von der Cyber 74 auf das DECSYSTEM-20, dem zentralen Unterstützungsrechner des PRZ zu übernehmen.

DURCHFÜHRTE ARBEITEN

1. Bau eines Verbindungsinterfaces zwischen CAMAC und Ringleitung.
2. Auftrennung des Steuerprogrammes in seine logisch getrennten Teile.
3. Anpassung der eigentlichen Steuerung an die PDP-11/34.
4. Implementierung des Programmes ZIPSAT auf dem DECSYSTEM-20.
5. Datenübertragung von DECSYSTEM-20 über den Zentralen Teleprocessing Rechner (TP-Rechner) zum Zentralen Prozeßrechner (ZPR).

Ad 1.: Das Interface ist so gebaut worden, daß man nun mittels Leuchtdioden feststellen kann, ob ein Digital-Input für ein bestimmtes Bit anliegt oder nicht. Man kann also die Position der Antenne am ZPR im BCD-Code ablesen.

Ad 2.: Das Steuerprogramm, das im Zuge der Diplomarbeit Krömer am Institut für

Hochfrequenztechnik entstanden ist, ist aufgetrennt worden.

Jener Teil, der die endgültigen Positionierungsdaten unter Berücksichtigung des möglichen Drehradius der Antenne (110° - 800°) berechnet, ist nun auf dem DECsystem-20 installiert.

Ad 3.: Das Steuerprogramm DRIVE ist komplett umgeschrieben worden. Die Steuerung wird nach folgendem System durchgeführt:

- a. Vorgabe von Positionierungsdaten für Höhe und Seite zu einer bestimmten Uhrzeit = Sollpositionswerte.
- b. Einlesen der Positionsdaten für Höhe und Seite = Istpositionswerte.
- c. Vergleichen dieser Werte.
- d. Überschreitet die Absolutdifferenz einen gewissen Toleranzwert, werden die Steuerschleifen für Höhe und Seite angesprungen. Danach folgt wieder Punkt b.
- e. Ist die Absolutdifferenz unter dem Toleranzwert oder sind die Steuerschleifen schon zu oft angesprungen worden, wird der Positionierungsvorgang abgebrochen ("Timeout").
- f. Nach jedem Positionierungsvorgang werden Istpositions- und Sollpositionswerte, sowie die Ist- und Sollzeit und eine Genauigkeitsaussage für diesen Positionierungsvorgang abgespeichert. Dann folgt wieder Punkt a, bis keine Daten mehr vorhanden sind.

Ursprünglich sind drei Geschwindigkeiten zur Positionierung der Antenne benutzt worden: schnelle, mittlere und langsame Bewegung. Die schnelle Bewegung der Antenne ist aufgelassen worden, da sich herausgestellt hat, daß sie fast nie benötigt wird.

Es ist nötig gewesen, auf Programmbasis ein Schwanken der Antenne, d. h. das direkte Umschalten der Motoren von Hoch auf Tief, bzw. Links auf Rechts zu vermeiden, da der hierzu vorgesehene Elektronikteil der

Antennenmotoraussteuerung defekt ist.

Da die Antennenmotoren auf ein und dieselbe Steuerimpulsdauer unterschiedlich reagieren, insbesondere bei niedrigen Temperaturen, ist ein "Timeout" eingebaut worden. Es ist dies hier ein Zähler, der inkrementiert wird, da die ursprünglich verwendete Zeitabfrage zuviel Zeit in Anspruch genommen hat. Wenn ein solches "Timeout" eintritt, ändert sich zwangsläufig die Absolutdifferenz zwischen Istpositionswert und Sollpositionswert und damit schließlich auch die Dauer der Aussteuerimpulse.

Dadurch ist im Anschluß ein Hardwarefehler im Bereich der Antennenmotorenansteuerung klar festgestellt worden, der intermittierend aufgetreten ist und sich durch Testen über mehrere Monate hinweg als defekter Fehlerstromschalter herausgestellt hat.

Um im Anschluß an jede Steuerung für einen Satellitendurchlauf genauere Aussagen über die Positionierungsgenauigkeit machen zu können, sind die "Statistiklisten" (Punkt f) angelegt worden.

Diese sind wichtig, um den temperaturabhängigen Impulsdauerwert bestimmen zu können. Dieser muß zu Beginn einer Steuerung eingegeben werden und ist ein Erfahrungswert. Durch alle genannten Änderungen ist es nun möglich, den Empfang von Satelliten auch in der kalten Jahreszeit durchzuführen.

Ad 4.: Die Berechnung der Positionierungsdaten durch das Programm ZIPSAT auf der Cyber 74 ist aus folgenden Gründen nicht optimal gewesen:

1. Die Daten waren auf Lochkarten verfügbar (kein anderes kompatibles Medium vorhanden), dadurch unkomfortable Handhabung
2. Die Beschaffung der Daten hat aus diesem Grund sehr lang gedauert und ist für einen großen Zeitraum durchgeführt worden.

Da nun eine Berechnung auf dem Zentralen Unterstützungsrechner DECsystem-20 möglich geworden ist, hat das Institut für Hochfrequenztechnik durch einen Studenten dieses Programm auf letzterem

System installieren lassen.
Man kann nun auf Grund einer
Übersichtsberechnung einen speziellen
Satellitendurchlauf auswählen.
Dies ist bei guter Kenntnis der
Bedienung des gesamten Programm-
pakets bis unmittelbar vor den
Empfangsbeginn möglich.

Ad 5.: Die Übertragung der Daten geschieht
mittels der Übertragungssoftware ACS
zwischen DECsystem-20 und TP-Rechner,
mittels DECNET zwischen TP-Rechner
und ZPR.

Weiters wäre eine ständige Korrektur der
Steuerimpulsdauer mit Hilfe einfacher sta-
tistischer Methoden und der "Statistikliste"
in Betracht zu ziehen.
Auch eine Art Verfolgungssteuerung ist denk-
bar, die allerdings nach einem anderen Prin-
zip funktionieren müsste. Eine sinnvolle Kom-
bination beider Steuerungen könnte dann
eventuell eine wesentlich höhere Empfangs-
sicherheit garantieren.

Es ist damit dieses Projekt ein gutes Bei-
spiel für die Verwendungsmöglichkeiten und
das Zusammenspiel der verschiedenen Rechner
des PRZ, wie man aus Abb. 1 ersehen kann:

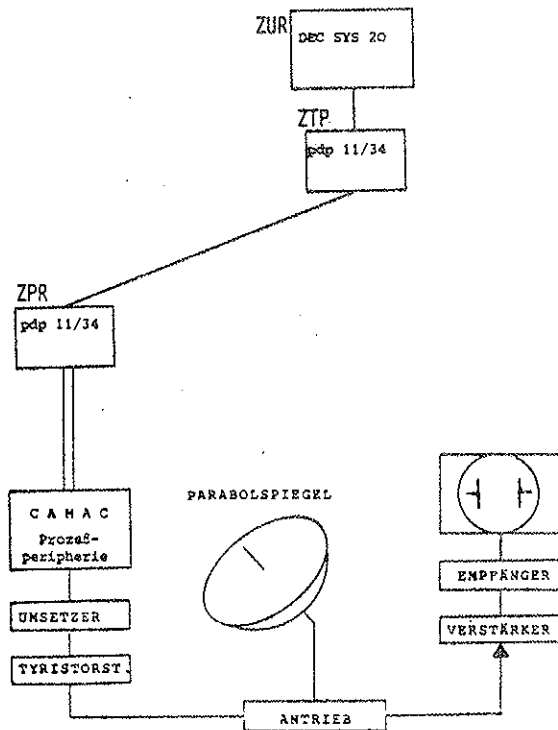


Abb. 1 Antennensteuerung

AUSBLICK

Als Weiterentwicklung der derzeitigen Steuerung
kann man sich eine automatische Ermittlung
der temperaturabhängigen Impulsdauerwerte
aus den "Statistiklisten" eines repräsen-
tativen Durchlaufes denken, den man vor dem
aktuellen Durchlauf startet.

PROGRAMMIERUNG UND ANWENDUNG DER FOURIER-ANALYSE IN PROZESS-SATELLITEN

W. Lauber

PROGRAMMENTWICKLUNG

Am Institut für Physikalische Elektronik war es in Zusammenhang mit mehreren Untersuchungen notwendig, Frequenzanalysen von sehr tiefen Frequenzbereichen - bis 10^{-2} Hz - durchzuführen. In diesem Frequenzbereich ist die Verwendung von Hardware-Analysatoren kaum möglich.

Das am Prozeßrechenzentrum entwickelte Programmpaket sieht die Datenerfassung in einem Satelliten PDP-11/04 vor. Die Daten werden mittels DECNET an den TP-Rechner PDP-11/34 geschickt, von dort mittels ACS-Inter-task-Communication an das DECsystem-20. (Abb. 1). Hier wird die Fourieranalyse durchgeführt, eventuell weitere Auswertungen vorgenommen und von Plotterprogrammen die Berechnung von Ausgabedaten für die CAMAC-Steuerung des Plotters durchgeführt (z.B. Kurvenwerte der Eingangsdaten, des Leistungsspektrums der Fourierdaten etc.). Diese Daten werden mit ACS wieder an den TP-Rechner zurückgesendet und dort auf Platte abgespeichert; an den Satelliten wird eine Erfolgsmeldung weitergesendet. Der Satellit kann die Daten jederzeit über DECNET vom TP-Massenspeicher abberufen und über die Peripherie des Satelliten (CAMAC) an einem Plotter oder TV-Gerät ausgehen.

ANWENDUNG

Die Anwendungsfälle dieser Analyse sind vielfältig, da eine Erweiterung der Auswerteprogramme am DECsystem-20 jederzeit leicht möglich ist.

Am Inst. für Physikalische Elektronik werden mehrere Projekte mit Anwendung der Fourier-Analyse bearbeitet: Für MOS-Transistor-Modelle soll Schwellspannung, Subthreshold-Verhalten, Kennlinien im ohmschen Bereich und in der Sättigung mittels Programm-Simulation ermittelt und mit experimentellen Daten verglichen werden. Ein weiteres Projekt soll die online-Analyse von Signalen im Audiofrequenzbereich (10 Hz bis 10 kHz) ermöglichen.

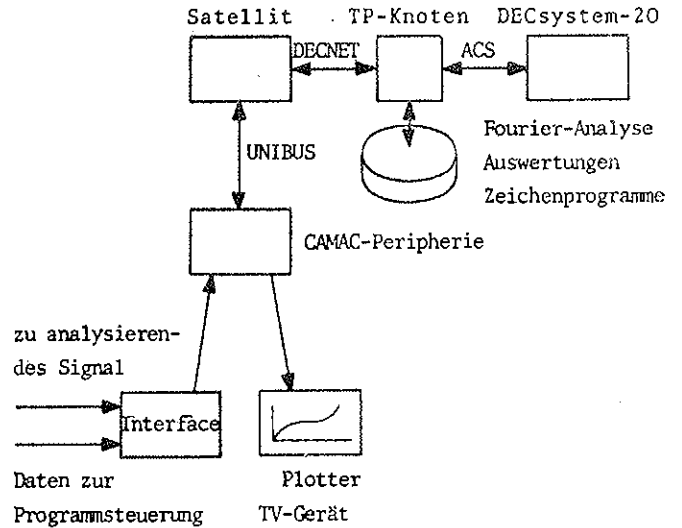


Abb.1 Datenverlauf zwischen Satellitenperipherie und DECsystem-20

Für die am Institut untersuchten akustischen Oberflächenfilter wird die Fourieranalyse zur Ermittlung der Übertragungsfunktion verwendet. Aufgrund dieses Ergebnisses können die physikalischen Daten des Filters derart variiert werden, daß ein gewünschtes Übertragungsverhalten erreicht wird. Durch einen Ausbau der Auswerteprogramme im DECsystem-20 ist eine Ermittlung dieser Daten über den Rechner möglich.

Eine weitere Anwendung ist eine Laborübung über Fourier-Analyse, in der auch ein Vergleich Spektrumanalysator mit Rechner durchgeführt werden soll.

PROZESSRECHNEREINSATZ FÜR ANALYSE UND SIMULATION BIOLOGISCHER SYSTEME

W. Lauber

GRÜNDE DES RECHNEREINSATZES

Für die Auswertung biomedizinischer Signale werden immer mehr Rechenanlagen eingesetzt, da sie im Gegensatz zu den übrigen, bisher verwendeten Auswerteverfahren -Überprüfung des Kurvenbildes durch den geübten Fachmann auf bestimmte optisch eindrucksvolle Kennzeichen oder Ausmessung mit Millimeterpapier bzw. Integrator -sowohl die Verwertung riesiger Datenmengen als auch die Anwendung komplexer mathematischer Verfahren ermöglichen.

Die Entwicklung geht dabei von der technischen Seite aus: Durch die rasch billiger werdenden kleinen Prozeßrechner-einheiten ist eine dezentrale Versorgung medizinisch-biologischer Meßgeräte und damit eine Online-Analyse großer Datenmengen möglich. Andererseits wird die Theorie von Fortschritten im Bereich der Systemwissenschaften vorangetrieben, indem zunehmend für technische Systeme entwickelte Verfahren im Bereich biologischer Systeme benutzt werden. Freilich führte in der Vergangenheit auch die Lösung von Problemen in der Analyse biologischer Systeme zu Ansätzen, die dann im technischen Bereich weiterverwendet wurden.

ENTWICKLUNG DER METHODEN

Seit 1965, dem Zeitpunkt der Veröffentlichung der schnellen Fouriertransformation (Fast Fourier Transformation FFT) von Cooley und Tukey, ist eine deutliche Zunahme der wissenschaftlichen Arbeiten festzustellen, die sich mit der Analyse biologischer Signale befassen. Inzwischen wurden weitere Fast-Fourier-Verfahren entwickelt, die nicht mehr die restriktiven Bedingungen -bei Cooley und Tukey mußte die Anzahl der der Transformation zugrundeliegenden Abtastwerte eine Potenz von 2 sein- der ersten FFT enthalten. Besonders in der Erforschung des EEG (Elektroencephalogramm), das seit etwa 50 Jahren bekannt ist, wurde die Fouriertransformation in den letzten 10 Jahren dazu verwendet, neue Parameter zur Kennzeichnung eines bestimmten Einzel-EEG zu ermitteln. Dabei wurde bewußt nicht von den Theorien über die Entstehung dieses Signals ausgegangen, sondern dieses als Realisation ei-

ner Zufallsvariablen aufgefaßt. Die Ergebnisse dieser Arbeiten sind mager: durch die Überlagerung der Signale aus vielen einzelnen komplexen Entstehungsmechanismen geht im Summensignal ein großer Teil der Information verloren.

Die meisten Arbeiten befassen sich daher mit der Analyse sogenannter ereignisbezogener Signale, das sind Signale, die in Zusammenhang mit einem dem biologischen System "äußeren" oder "inneren" Ereignis stehen. Die dabei entstehenden Signale werden zumeist nach vorgegebenen, dem bisherigen Stand der Kenntnis entsprechenden Kriterien untersucht. Ein häufiger, beim EKG (Elektrokardiogramm) bereits weit verbreiteter praktischer Einsatz ist die ärztliche Diagnose. Für die Überwachung, z.B. im klinischen Bereich in Intensivstationen, ist die Lösung des umgekehrten Problems, nämlich die Feststellung eines bestimmten Ereignisses aufgrund bestimmter Signalparameter, nötig. Bei all diesen Situationen ergibt sich das Problem der Reliabilität und der Validität. Während das Problem der Validität, d.h. eines ausreichend gesicherten Zusammenhangs zwischen einer bestimmten Signalkonstellation und dem interessierenden Ereignis, durch die automatische Signalanalyse kaum beeinflusst wird (die Kriterien des Zusammenhangs sind ja vorgegeben), betrifft die Reliabilität, d.h. die ausreichend sichere Erkennbarkeit einer bestimmten Signalkonstellation, die Qualität des Analysealgorithmus. Daher wird auf diesem Gebiet durch Erforschung zuverlässiger Parameter noch in hohem Ausmaß "Grundlagenforschung" betrieben.

Die verwendeten Analyseverfahren stützen sich zumeist auf die Fourier-Analyse, besonders auf Leistungsdichtespektrum und Kreuzkorrelation. Häufig liefert auch schon die Intervallanalyse, die den zeitlichen Verlauf zweier aufeinanderfolgender Signal-Wellen statistisch auswertet, ausreichende Informationen. Generell können hier alle Verfahren zur Mustererkennung verwendet werden.

ARBEITEN AM PRZ

Am Prozeßrechenzentrum des EDV-Zentrums der TU wurden bzw. werden in diesem Zusammenhang mehrere Projekte durchgeführt:

In Rahmen einer Dissertation am IV-Institut

für Mathematik analysierte Schuster Atemverlauf, EKG und Hautwiderstand, die während Konzert- und Theateraufführungen auf Band aufgenommen worden waren. Ziel der Arbeit war es, in den Signalen wiederkehrende Verhaltensmuster, inter- oder intraindividuelle Zusammenhänge zu finden und die Abhängigkeit der Signale vom dargebotenen Sprach- bzw. Musiksingnal zu ermitteln. Die gefundenen Zusammenhänge wurden mit parameterfreien Tests auf Signifikanz geprüft.

Im Rahmen des Projektes "Nichtinvasive diagnostische Methoden zur Bestimmung von Kreislaufparametern mittels Ultraschall-Doppler-Prinzip" des Inst. f. Allg. Elektrotechnik zusammen mit der Abt. für Magnetohydrodynamik (Jacobovicz) wird ein hydrodynamisches Modell für die Nachbildung verschiedener Strömungsformen aufgebaut. Die mit Ultraschall-Doppler-Gerät gemessenen Strömungsformen sollen mit dem Modell verglichen werden.

In einer Untersuchung der II. Chirurgischen Universitätsklinik führte Lobodzinski eine Auswertung biorhythmischer Funktionsschwankungen durch. Ziel des Projekts war vorerst die Untersuchung der während bestimmter Umweltsituationen ermittelten physiologischen Signale bezüglich signifikanter Parameter. Als ein Ergebnis dieser Untersuchung wird ein lineares Systemmodell der Aorta erstellt, das u.a. mittels Fourieranalyse an die vorhandenen Daten angepaßt werden soll. Während die Daten von einem Prozeßrechner PDP-11/34 erfaßt und komprimiert werden, wird die weitere Analyse sowie der Systemvergleich auf dem DECsystem-20 durchgeführt.

Am Inst. f. Arbeitswissenschaft werden häufig physiologische Signale in Zusammenhang mit Arbeitsuntersuchungen wie z.B. Belastungs- und Vigilanzuntersuchungen erfaßt. Für eine umfangreiche Studie, in der mehrere psycho-physiologische Größen gleichzeitig über mehrere Stunden im Feld über Telemetriegeräte erfaßt werden, ist geplant, die Daten in einem Prozeßrechner PDP-11/34 zwischenspeichern und eine vorläufige Auswertung durchzuführen, und sie für umfangreiche Auswertungen dem DECsystem-20 zuzuführen.

Da im Rahmen des "Biomedical Engineering"

die Zusammenarbeit zwischen medizinischen und technischen Hochschulinstituten zunimmt, wird auch am PRZ der beschriebene Arbeitsbereich als Schwerpunkt betrieben werden.

RSX-11 ODER: DAFÜR MUSSTE ES DOCH IRGEND- ETWAS GEBEN!

W. Wöber

Ist es Ihnen nicht auch schon so ergangen? Sie haben ein Problem, und es sieht gar nicht so schwierig aus, dafür eine Lösung zu finden. Ja - wenn man eine findet oder zumindest eine Idee hat, wo man suchen könnte. Ich glaube, jeder ist schon einmal vor dieser Situation gestanden und war froh, daß er einen Tip bekommen hat, der nachher viel Zeit gespart hat. Deshalb möchte ich versuchen, einen kurzen Überblick über die Programme und Hilfsmittel zu geben, die unter einem Mehrzweckbetriebssystem wie RSX-11 zur Verfügung stehen. Als mögliche Gliederung bietet sich dabei folgende Struktur an:

1. Verarbeitung von Texten, Zeichen und Daten im weitesten Sinn;
2. Verwaltung von Datenbeständen und Datenträgern;
3. Programmiersprachen und Sprachübersetzer.

1. ZEICHENVERARBEITUNG

Wenden wir uns zunächst den Standard-Editierprogrammen zu. Als einfachstes, aber durchaus komfortables Programm steht SLP zur Verfügung. Die Abkürzung "SLP" steht für Source Language Input Program und gibt bereits einen Hinweis auf die Eigenschaften dieses Programms: ein Quellenprogramm, aber auch jeder andere Text, wird von einem Eingabemedium entgegengenommen und normalerweise auf einem Massenspeicher als Datei abgelegt. Das Eingabegerät ist sehr oft ein Benutzerterminal, kann aber im Prinzip jedes Gerät sein, welches dem Betriebssystem bekannt ist. Allerdings ist mit dem einfachen Einlesen von Text die Funktionalität noch nicht erschöpft, denn als Editierhilfe bietet SLP einen ganzen Satz von Kommandos, die eine Änderung eines vorhandenen Textes erlauben.

Dabei ist folgendes zu beachten: SLP ist strikt zeilen-(record)orientiert und eine

Zeile ist durch ihre Position im Datenbestand (Record Nummer) gekennzeichnet. Dabei braucht man den Inhalt einer Zeile nicht zu kennen, nur ihre Nummer. Und damit man nicht zählen muß - wofür hat man schließlich einen Computer - ist SLP imstande, eine Liste der Zeilen mit den Positionsnummern zu erstellen.

Weiters bietet SLP die Möglichkeit, Veränderungen in der Datei mit einer Zeichenkombination zu markieren, damit man Verbesserungen zurückverfolgen kann. Zusammenfassend kann man sagen, daß SLP primär für die Bearbeitung von relativ unstrukturierten Textbeständen geeignet ist, auf die man über die Position zugreifen möchte, aber nicht über den Inhalt einer bestimmten Zeile.

Nun gibt es aber nicht nur unstrukturierte Texte, wie etwa Zahlenkolonnen, sondern auch Dateien, die z. B. Programmtext beinhalten. Und wenn man an Programmen etwas ändern möchte, dann meist an einer Stelle, die durch ihren Inhalt gekennzeichnet ist, wie etwa bei einer Programm-Marke, bei einer Statement-Nummer oder bei einem ganz bestimmten Formatstring. Genau dafür wurde der zweite Standard-Editor geschaffen, er wird unter dem Namen "EDI" aktiviert. EDI ist nun, im Gegensatz zu SLP, zeichenorientiert. Der Grundmechanismus bei EDI besteht darin, eine Zeichenkombination zu suchen. Damit wählt man eine bestimmte Zeile aus, die mit Hilfe weiterer Kommandos modifiziert werden kann.

Was man zu EDI noch sagen könnte?

Nun, er ist nicht gerade genial, er ist auch nicht immer ganz konsequent, aber für die meisten Aufgaben ist er eigentlich recht gut geeignet.

So weit nun zu den beiden normalen Editor-Programmen. Aber wußten Sie auch, daß es für komplexere Aufgaben ein noch leistungsfähigeres Werkzeug gibt? Das Programm heißt "TECO" (Text Editor and Corrector) und bietet vom Konzept her alle Funktionen für eine interpretative Programmiersprache. TECO ist streng zeichenorientiert und bietet über die Funktionen von EDI hinaus die Möglichkeiten, einen Satz von Kommandos zu verketteten, abzuspeichern und wiederzuverwenden, sowie als Suchargumente ganze Zeichengruppen als Treffer zu akzeptieren, wie "Ziffern" oder "Buchstaben + Ziffern" oder

"alle Zeichen". Übrigens: TECO war oft schon die letzte Rettung, wenn einem Datenbestand irgendetwas zugestoßen ist!

Zum Abschluß des Kapitels über Zeichenverarbeitung noch ein kurzer Hinweis auf eine Möglichkeit, professionell formatierte Textseiten zu erzeugen. Das Programm "RUNOFF" liest als Eingabe Rohtext mit Steuerinformationen und produziert daraus fertig formatierte Textseiten in Groß- und Kleinbuchstaben. Weiters realisiert RUNOFF Einrückungen, Absätze, Unterstreichungen und Inhaltsverzeichnisse. Als bestes Beispiel für die Leistungsfähigkeit von RUNOFF kann man wohl die Manuals der Firma DEC betrachten.

2. DATENVERWALTUNG

Eines der wichtigsten Programme für die Datenverwaltung ist wohl "PIP" (Peripheral Interchange Program). Mit PIP kann man Files kopieren, löschen, mit neuen Namen versehen, Directories drucken, usw. Aber PIP kann auch andere Informationen für die Benutzer zugänglich machen. Etwa wieviele Files in einem Directory abgespeichert sind und wieviel Platz die Dateien benötigen. Oder wieviel Platz auf einer Platte oder auf einem Band noch frei ist. Noch ein Tip: mit PIP kann man die Zugriffsrechte auf bestimmte Daten so definieren, daß sie gegen unabsichtliches Löschen geschützt sind. Diese Schutzfunktion sollte man bei wichtigen Daten auch gegen sich selbst anwenden.

Einen ähnlichen Funktionsumfang wie PIP hat auch "FLX" (File Exchange Program), allerdings ist Filex auf die Verarbeitung fremder Filestrukturen spezialisiert. Im speziellen stehen Kommandos für die Bearbeitung von DOS-11 (Disk Operating System) und RT-11 (Real Time System) zur Verfügung.

Wenden wir uns nun einem anderen Aspekt der Dateiverwaltung zu. Jeder vorsichtige Computerbenutzer legt sich doch Kopien (Backups) seiner Programme und Daten auf externen Speichern, wie Bänder oder Platten an. Dabei ergibt sich oft die Notwendigkeit, zu

verifizieren, daß die Daten nicht nur geschrieben wurden, sondern daß sie auch lesbar sind. Für diesen Zweck steht ein eigenes Systemprogramm zur Verfügung: "VFY" (Verify). Als weitere Funktion ist VFY imstande, sicherzustellen, daß die Filestruktur an sich in Ordnung ist oder VFY meldet genau jene Stellen, wo Probleme existieren.

Benützen Sie eigentlich DECTapes? Wenn ja, dann sollten Sie auch über "BAD" (BADblocklocator), "INI" (INIvolume) und "UFD" (User File Directory) Bescheid wissen. BAD versucht alle physikalischen Blöcke auf einem DECTape zu schreiben und zu lesen. Damit kann sichergestellt werden, daß eine Spule in Ordnung ist und die Formatierung richtig vorgenommen wurde. Vorsicht: BAD zerstört alle Information, die auf Band gespeichert war! INI dient zur Initialisierung eines Bandes und erlaubt die Angabe eines Volume-Labels (Spulen Name), sowie die Auslegung der Filestruktur für ein Band. Da die Verarbeitungsgeschwindigkeit sehr von der Filestruktur abhängt, sollte man die Programmierberatung kontaktieren, um möglichst optimale Parameter zu finden. Was für INI gilt, trifft auch auf UFD zu. Eine ungünstige Initialisierung der User File Directories führt oft zu unnötigen Verzögerungen bei Bandzugriffen.

3. SPRACHPROZESSOREN

Da RSX-11 (wußten Sie eigentlich, daß diese Abkürzung Realtime System Executive heißt?) als Echtzeitsystem optimiert ist, steht auf der untersten Programmierungsebene ein ungewöhnlich leistungsfähiger Assembler zur Verfügung. Unter anderem bietet dieser Übersetzer Macromöglichkeit, bedingte Übersetzung, Kontrolle des Listenumfangs und Listenformats, Symboltable und Crossreference. Darüber hinaus erlaubt "MAC" (MACro Assembler) echte, strukturierte Programmierung auf möglichst maschinennaher Ebene. Für verschieden umfangreiche Aufgaben werden mehrere Assemblerversionen konfiguriert, die sich durch Speicherbedarf und Durchsatz unterscheiden. Wenn man zeitkritische Anwendungen oder Systemkommunikation realisieren muß, zahlt es sich oft aus, die psychologische Hürde "Assembler" zu überspringen.

Wer allerdings nicht zeitkritische Programme schreiben muß, ist wahrscheinlich mit FORTRAN besser beraten. Für diese Sprache stehen zwei Übersetzer zur Verfügung, die sich dadurch unterscheiden, daß sie auf verschiedene Anwendungen optimiert sind. "FOR" (FORtran) ist jener Compiler, der keine besonderen Hardwarezusätze zu einer normalen PDP-11 für die übersetzten Programme voraussetzt. FOR ist bei der Übersetzung sehr schnell, braucht wenig Hauptspeicher und ist mit dem Compiler unter RT-11 kompatibel. Als Nachteile nimmt man dafür eher magere bis nicht vorhandene Fehlermeldungen und einen nur wenig optimierten Objectcode in Kauf. Im Gegensatz dazu glänzt "F4P" (Fortran-4-Plus) mit ausführlichen Fehlermeldungen und einem optimierten Objectcode, der Gleitkommahardware ausnützt. Da allerdings jede Leistung ihren Preis hat, benötigt F4P für die Übersetzung eines Programms relativ viel Zeit und belegt ein ziemlich großes Feld im Hauptspeicher. F4P unterstützt unter RSX-11D außerdem noch die Multiuser-Möglichkeit und Hardware-Schreibschutz für Programmteile, die nicht verändert werden sollen.

Zusätzlich zu den beschriebenen Übersetzern für Assembler und Fortran sind unter RSX-11D (PDP-11/45) Sprachsysteme für BASIC und ALGOL auf interpretativer Basis in Testversionen einsetzbar. Interessenten sind eingeladen, diese Programme zu testen, Firmenunterstützung besteht allerdings nicht, da es sich um Benutzerentwicklungen handelt.

ZUSAMMENFASSUNG

Die vorangegangenen Kurzbeschreibungen der RSX-11 Systemprogramme waren nur als Denkanstöße gedacht, um einen Hinweis zu geben, was für welche Aufgabe im Prinzip einsetzbar ist. Klarerweise kann ein solcher Beitrag nie das ausführliche Lesen der Firmendokumentation oder eine Programmberatung ersetzen. Ich hoffe, damit einen möglichst informativen Überblick gegeben zu haben, der zumindest einigen Benutzern Zeit spart und Lösungswege aufzeigt.

COMPUTERSIMULATION VON SONNENBEHEIZTEN ANLAGEN

G. Schaffar

Durch die hohe Komplexität sonnenbeheizter Anlagen unter Rückwirkung der Verbrauchsstruktur von Warmwasser und Heizungsenergie auf die Wirkungsgrade der einzelnen Komponenten, im speziellen der Solarkollektoren, ergibt sich die Notwendigkeit, die auftretenden komplizierten Zusammenhänge mittels einer Digitalrechenanlage zu simulieren. Dies wurde im Projekt "Sonnenhaus" durchgeführt.

Es ergab sich die Notwendigkeit, verschiedene Programmpakete zu entwickeln:

1. Das Sonnenhaussimulationsprogramm
2. Ein Programm, das für das oben genannte Simulationsprogramm die Ein-/Ausgabe verwaltet. Verschiedene Tasks, die Kostenstrukturen und Nutzenstrukturen berücksichtigen.
3. Ein Programmpaket zur Extremalisierung beliebiger Ausgangsparameter in Abhängigkeit von beliebigen Eingangsparametern,
4. sowie zwei Programme, die einerseits eine graphische Darstellung oder eine numerische Darstellung derselben erlauben.

Das Problem stellt sich daher in zwei Ebenen, erstens auf der Ebene der Simulation, zweitens auf der Ebene der wirtschaftlichen und technischen Optimierung.

SIMULATION:

Auf der Ebene der Simulation sind die physikalischen, mechanischen und technischen Eigenschaften der einzelnen Komponenten zu berücksichtigen. Die auftretenden Zusammenhänge sind jedoch von solcher Kompliziertheit, daß die Rechnung mit Mittelwerten nicht mehr zulässig ist. Es ergibt sich daher die Aufgabe, bei der Berechnung solcher Anlagen die Benützungszeiträume, z. B. ein Jahr, in Stundenintervallen zu durchlaufen. Um einen Überblick über die Komplexität der zu berücksichtigenden Eigenschaften zu geben, sei eine kurze Übersicht über die zu berücksichtigenden Eigenschaften gegeben: Jahresgang der mittleren Temperatur, Temperaturextremwerte, Jahresgang der Tem-

peraturamplitude, astronomische Sonnenhöhe, astronomischer Sonnenazimuth, Sonnenscheinwahrscheinlichkeiten, absorbierender Einfluß der Atmosphäre, Gewichte für die terrestrische Solarkonstante, Speichermindesttemperatur, Speicherüberlauf, Zusatzenergiebedarf, Speicher, Wärmeinhalte, Fresnelverluste an den Kollektoren, Berücksichtigung von gegenseitiger Abschattung bei "Shed"-Anordnung, Wirkungsgrad und Kennlinien von Kollektoren, Pumpgeschwindigkeit des Mediums, Leistungsverluste in Wärmetauschern, Kollektorneigungen, Kollektorazimuth, Wärmedurchgangszahl des betrachteten Hauses, Horizonthöhen, Initialtemperaturen usw.

OPTIMIERUNG UND WIRTSCHAFTLICHE GESICHTSPUNKTE:

Das wirtschaftliche und technische Szenarium kann in einem eigenen Unterprogramm den jeweiligen Bedingungen leicht angepaßt werden. Diese Bedingungen dienen als einschränkende Bedingungen für die Optimierung.

Da die Optimierungsrechnungen an einem so komplexen System unter den verschiedenen Gesichtspunkten durchgeführt werden können, ist in dem von uns entwickelten Programmpaket vorgesehen, beliebige Systemparameter oder deren Verknüpfungen in Abhängigkeit von beliebigen Eingangsparametern des Systems zu optimieren. Es handelt sich also hierbei um eine Extremwertaufgabe auf einer n-dimensionalen Hyperfläche.

Programmiertechnik:

Da auf der uns zur Verfügung stehenden Prozeßrechenanlage der maximale Adressenbereich 32 k Worte beträgt, die Komplexität des Problems jedoch größere Speichervolumina in Anspruch nimmt, wurde nach einer Lösung dieses Problems gesucht. Die Lösung dieses Problems stellte sich in der Verwendung der Teilung des Programms in mehrere Teilprogramme, wobei das eigentliche Simulationsprogramm als eigener selbständiger RSX-11-Task läuft und von den asynchron ablaufenden Unterstützungsprogrammen her über Systemdirektiven gestartet und mit Daten versorgt und entsorgt wird. Diese Lösung erwies sich als sehr praktisch, da einerseits

Änderungen am Simulationsprogramm oder an den bedienenden Programmen vorgenommen werden können, ohne alle anderen Teile neu linkeln zu müssen, und andererseits auch in die Programme interaktiv eingegriffen werden kann. Auch ist die Möglichkeit der Teilung des Speicherbereiches der einzelnen Programme in read-only und read-write Bereiche von großem Vorteil für die Programmsicherheit und für das Laden bei großer Speicherbelegung durch fremde Tasks.

Durch die oben erwähnten Kniffe kann in unserem Fall ein so großes Problem in fünf Teile zerlegt werden:

- a. Der Read-only-Bereich des Simulationsprogrammes
- b. Der Read-write-Bereich des Simulationsprogrammes
- c. der Read-only-Bereich des Optimierungsprogrammes
- d. Der Read-write-Bereich des Optimierungsprogrammes
- f. Die fortranresidente Library

EINSATZMÖGLICHKEITEN FÜR DAS VON UNS ENTWORFENE VERFAHREN:

Mit dem von uns entwickelten Programmpaket können Brauchwassererwärmungen mit Solar Kollektoren sowie solare Heizungen, Zusatzheizungen, Kollektoruntersuchungen und Schwimmbadheizungen gerechnet werden. Durch die Erweiterung des Bedienungsprogramms mit einer Kollektorsimulation ist es möglich, aus rein konstruktiven Parametern eines solaren Kollektors auf dessen gesamtphysikalische Eigenschaften im System zurückzurechnen. Umgekehrt ist es natürlich genau so möglich, die konstruktiven Parameter eines Kollektors z. B. Rohrdurchmesser, Isolationsdicke, Scheibenanzahl usw. durch Kombination mit diversen Kostenfaktoren für ein gegebenes System nach bestimmten Gesichtspunkten zu optimieren, respektive kostengünstig zu gestalten. Die Kontrolle mit dem Experiment zeigte, daß die errechneten Kollektorkennlinien mit den experimentell ermittelten übereinstimmen. Abschließend kann man sagen, daß mit dem von uns entwickelten Programmpaket ein vollautomatisches Entwurfssystem für Solaranlagen realisiert wurde.

DAS IBM SYSTEM/7 AUF DEM WEG AUS DER "ISOLATION"

G. Wehrberger

Dem am Getreidemarkt von der Abt. Prozeßrechenanlage betreuten Prozeßrechner IBM S/7 wurde mit dem Abbau der IBM-1800 in der Gußhausstraße und dem damit verbundenen Abbruch der TP-Verbindung zur IBM-1800 und weiter zum System/370 der IBM am Schwedenplatz gewissermaßen die "Nabelschnur" durchtrennt. Um lebensfähig bleiben zu können, war es notwendig, den Hauptspeicher des S/7 auf 28k Worte zu erweitern und einen Zeilendrucker zu installieren. In dieser Ausbaustufe konnte das S/7 zufriedenstellend als "Stand-alone"-Rechner eingesetzt werden und es war auch in der Lage, über die Prozeßrechneranschlußtafeln Prozesse in den Laboratorien zu betreuen. Die hard- und softwaremäßige Unterstützung jener Institute, die über eigene Kleinrechner verfügen, scheiterte an den Kosten der für den Anschluß von Satellitenrechnern bzw. Terminals erforderlichen Hardware-Erweiterungen am System/7.

Um den Benützern des S/7 dennoch die Möglichkeit eines TP-Anschlusses bieten zu können, wurde am Prozeßrechenzentrum ein Gerät gebaut, welches über einen Teil der Prozeßperipherie des Rechners eine 20 mA Stromschleife treibt, also eine serielle Schnittstelle zum System/7 liefert. Unter Verwendung dieser verbreiteten Schnittstelle ist nun der Anschluß peripherer Geräte mit serieller Schnittstelle möglich.

Dadurch konnte nicht nur der Anschluß eines Bildschirm-Terminals und eines Plotters an das S/7 realisiert, sondern auch die TP-Verbindung zwischen dem S/7 und der PDP 11/45 am Karlsplatz ermöglicht werden (Siehe Aufsatz auf S.48). Die Zeiten eines isolierten S/7 sind am Getreidemarkt somit überwunden, und dem S/7-Benützer stehen wieder sämtliche EDV-Ressourcen der Prozeßrechenanlage von Gußhausstraße und Karlsplatz zur Verfügung.

DER SERVOGOR PLOTTER 211

Dem Wunsch vieler S/7-Benützer nach einer Möglichkeit der graphischen Datenausgabe konnte von der Prozeßrechenanlage durch den Ankauf des SERVOGOR PLOTTERS 211 der Firma Goerz entsprochen werden. Dieser Plotter besitzt ein eingebautes Rechenwerk, welches den Datenfluß vom Rechner im Gegensatz zu den gebräuchlichen Inkrementalplottern auf ein Minimum beschränkt. Der SERVOGOR 211 kann als "Digitalplotter mit analogem Aufzeichnungssystem" bezeichnet werden, da im Digitalteil aus den vom Rechner gelieferten Steuer- und Datenbefehlen die Bahnkurve berechnet, in Spannungen umgewandelt und dem analogen Schreibsystem zugeführt wird.

Die maximale Zeichenblattgröße hat das Format DIN A3. Als Zeichengerät stehen ein Tuschezeichner und ein Kunststoff-Faserstift zur Verfügung. Das Zeichenformat kann sowohl in der Arbeitsweise mit festem Maßstab als auch mit variablem Maßstab verändert werden. Bei festem Maßstab beträgt die Einheitslänge 0,1 mm und bei variablem Maßstab liegt sie zwischen 0,01 mm bis 0,11 mm. Die Gesamtgenauigkeit liegt im Bereich von $\pm 0,2\%$ vom Endwert. Ein Beispiel dafür geben die Eckpunkte der Quadrate in Abb.1, welche durchwegs auf die Diagonalen des größten Quadrates zu liegen kommen.

Der SERVOGOR 211 kennt vier Arbeitsweisen:

- VEKTOR: zur linearen Interpolation
- DIREKT: zur absoluten Positionierung
- SYMBOL: zur Beschriftung
- INKREMENT: für spezielle, in der Grundsoftware gespeicherte Symbole, wie Kreise, Ausgleichskurven usw.

Dem S/7-Benützer, der den Plotter zur graphischen Datenausgabe verwendet, steht ein vom PRZ erstelltes Plottersoftwarepaket zur Verfügung. Dieses besteht aus FORTRAN-kompatiblen Unterprogrammen, welche es dem Anwender ersparen, sich um den Code des entsprechenden Plotterbefehls und um die Aufbereitung der zugehörigen Daten zu kümmern. Die im Grundsoftwarepaket des PRZ vorhandenen Unterprogramme gestatten es dem Benützer, die Funktionen bzw. Funktionsgruppen des Plotters mit einem symbolischen Namen anzusprechen, um die notwendigen Daten als Parameter übergeben zu können.

03.MA1 1978 11,57

PLOTTER
TEST

PRA
S/7

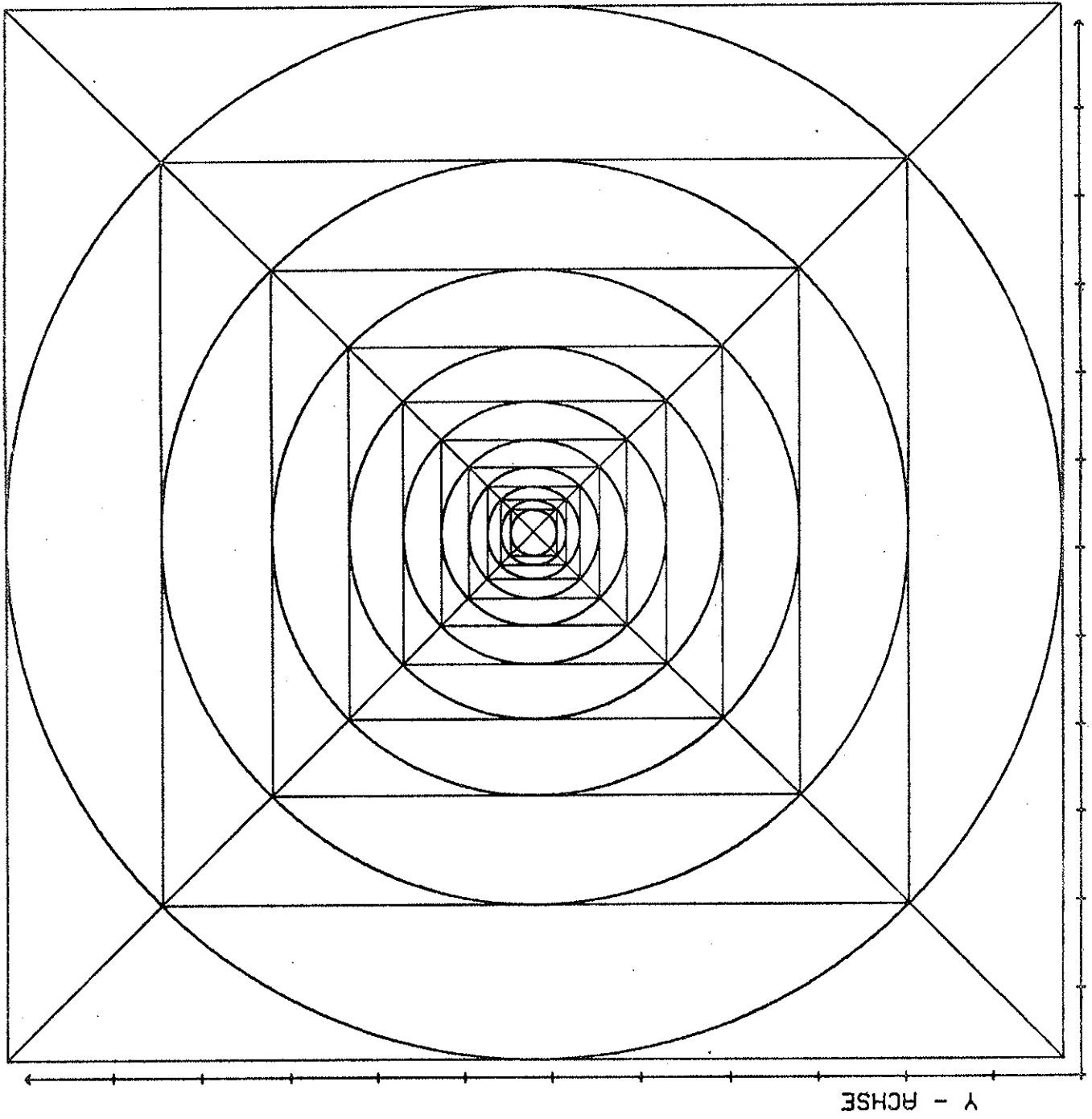


Abb. 1

Einige Beispiele dafür sind:

FPLOT(I,X,Y)

Führt die Zeichenfeder in gehobenem bzw. gesenktem Zustand zum gewünschten Zielpunkt.

I Federsteuerung

X,Y Zielkoordinaten

FGRID(TYPE,X,Y,LENGTH,COUNT,SIZE)

zeichnet eine Gerade mit Unterteilungen, wobei bei jeder Unterteilung eine Kennzeichnung vorgenommen wird. (Achsenkreuze mit Maßstabseintragungen, Raster)

TYPE Zeichenrichtung

X,Y Koordinaten, in denen die Zeichnung begonnen wird

LENGTH Länge einer Teilstrecke

COUNT Anzahl der zu zeichnenden Teilstrecken

SIZE Größe eines Unterteilungsstriches in 16 Stufen (0,...,15)

CIRCL(X,Y,R,START,LENGTH)

Zeichnet einen Kreisbogen

X,Y Koordinaten des Kreismittelpunktes

R Radius des Kreises

START Winkel im Bogenmaß von der X-Achse weg gemessen, von dem der Kreisbogen begonnen werden soll.

LENGTH Länge des Kreisbogens im Bogenmaß

SQUAR(X,Y,HEIGHT,LENGTH,ANGLE)

Zeichnet ein gedrehtes Rechteck

X,Y Koordinaten des Anfangspunktes

HEIGHT Höhe des Rechteckes

LENGTH Länge des Rechtecks

ANGLE Drehwinkel im Bogenmaß von der X-Achse weg.

FCHAR(X,Y,SIZE,PHI)

Ermöglicht die Ausgabe von alphanumerischen und Sonderzeichen. Die eigentliche Ausgabe erfolgt durch das FORTRAN-Statement WRITE(18,...) mit dem dazugehörigen FORMAT-Statement.

X,Y, Punkt, bei dem die Ausgabe der Zeichen beginnen soll.

SIZE Höhe der Zeichen in 16 Stufen (0,...,15)

PHI Drehung des Zeichensatzes bei der Ausgabe in 90°-Schritten im Uhrzeigersinn (0,...,3)

DAS BILDSCHIRMTERMINAL REGENT 100

Der zentrale Unterstützungsrechner des Prozeßrechenzentrums, das DECsystem-20, ist von seiner Größe und Konzeption her als Time-Sharing-Rechner und zur Verarbeitung von speicherintensiven Programmen prädestiniert. Wegen der Bedeutung eines Großrechners insbesondere für Chemiker wurde im Rechenraum des IBM-S/7 am Getreidemarkt ein Terminal zum DECsystem-20 installiert. Dadurch konnte ermöglicht werden, daß die Kunden des PRZ die Programmerstellung und Auswertung von S/7-Daten am DECsystem-20 vom Rechenraum am Getreidemarkt aus durchführen können.

Durch die Installation des am DECsystem-20 angeschlossenen Terminals und die auf der S/7-Seite vorhandene serielle Schnittstelle konnte aber auch der lange gehegte Wunsch der S/7-Benutzer nach einem leisen und komfortablen Dateneingabegerät erfüllt werden. Ein Schalter, der es gestattet, das REGENT 100 fallweise mit dem S/7 bzw. dem DECsystem-20 zu verwenden, bietet im Zusammenhang mit der vom PRZ erstellten Software zur Bedienung des Terminals vom S/7 aus die Möglichkeit, neben der IBM-Bedienerstation (Teletype) von einem zweiten Terminal aus lautlos Eingaben zu tätigen.

Die PRZ-Software für die Benützung des REGENT 100 vom S/7 aus sieht sechs wesentliche Funktionen vor. Durch gleichzeitiges Drücken zweier nachfolgend angeführter Tasten wird erreicht:

- CNTR O Stoppt die Ausgabe eines Files
- CNTR S Bewirkt seitenweise Ausgabeform(24 Zeilen)
- CNTR Q Bewirkt die Ausgabe der nächsten Seite
- CNTR R Zurücksetzen in die kontinuierliche Ausgabeform.
- CNTR U Löscht die laufende Zeile sowohl am Bildschirm als auch im Puffer.
- HT Löscht das zuletzt eingegebene Zeichen sowohl am Bildschirm als auch im Puffer.

Der Anschluß eines Bildschirmterminals an das S/7, der aufgrund der eingangs erwähnten, vom PRZ entwickelten Hard- und Software ermöglicht wurde, hat sich in der Praxis bereits bestens bewährt. Die Bedienerstation des S/7 steht nunmehr ausschließlich für Systemmeldungen sowie zur Kommunikation mit Prozeßprogrammen zur Verfügung. Das REGENT 100 wird, sofern es nicht an das DECsystem-20 angeschlossen ist, wegen

seiner geringfügigen Geräuschentwicklung sowie komfortablen Ein- und Ausgabe in erster Linie zur Programmentwicklung am System/7 verwendet. Durch diese Aufgabenteilung konnte vor allem erreicht werden, daß Anwender während der Programmerstellung nicht ständig durch System- bzw. Prozeßmeldungen unterbrochen werden.

Durch das steigende Interesse am DECsystem-20 steht das Terminal leider viel zuwenig dem eigentlichen S/7-Benützer zur Verfügung. Die Terminalsoftware sowie die spezielle Anschlußeinrichtung würden die Installation eines weiteren Eingabegerätes, welches ausschließlich dem S/7 dient, ermöglichen, und wir hoffen, im Interesse aller S/7-Benützer bald ein solches Terminal vorstellen zu können.

EIN BABY WIRD ERWACHSEN

(DATENVERARBEITUNG MIT DEM IBM SYSTEM/7)

L. Tauer

Ich gebe zu, es ist eine recht ungewöhnliche Überschrift für einen Aufsatz, der von einem Prozeßrechner handelt. Vergewöhnen wir uns aber die Entwicklungsgeschichte des am Getreidemarkt installierten Prozeßrechners IBM S/7, so zeigen sich doch deutliche Parallelen.

Es begann mit dem Abbau der IBM 1800, die mit dem S/7 mit einer "Nabelschnur", nämlich einer Datenübertragungseinrichtung und einer Telefonleitung, verbunden war. Damit war es möglich, wenn auch ohne Komfort, Daten auszutauschen bzw. die E/A-Peripherie der 1800 von dem S/7 aus mitzubedenutzen. Auf sich allein gestellt (Abb.2a), vermochte das S/7 zwar den dringendsten Bedarf zu decken, war aber aufgrund seines beschränkten Hauptspeichers und Massenspeichers nicht imstande, umfangreichere Auswertungen von größeren Datenmengen durchzuführen. Dies ließ auf Seiten der Benutzer den Wunsch entstehen, das S/7 auf irgend eine Weise an einen leistungsfähigeren Rechner heranzuführen. Mit der Fertigstellung der seriellen Schnittstelle (20mA Schleife) und der Software für Plotter und Terminal konnte die Verbindungssoftware in Angriff genommen werden.

Aufbauend auf dem von den Herren Selos und Wöber an den PDP-11/45 entwickelten Leitungsprotokoll wurde von mir ein Softwarepaket entwickelt, das es dem Benutzer ermöglicht, auf den Rechner PDP-11/45 zuzugreifen. Folgende Funktionen stehen zur Verfügung:

- PUT Übertragen eines Files vom S/7 zur PDP-11/45
- GET Übertragen eines Files von der PDP-11/45 zum S/7
- RJB Starten eines Batch-files auf der PDP-11/45
- XEQ Starten eines Tasks in der PDP-11/45
- OPR Senden einer 1-zeiligen Mitteilung an den Operator der PDP-11/45
- TLK 2-Weg Kommunikation mit einem PDP-11/45 Benutzer
- REX Abbruch der Verbindung durch Beendigung des Übertragungsprogrammes in der PDP-11/45

Alle diese Funktionen sind im Betriebssystem MPOS/7 verfügbar und können parallel zu anderen Aufgaben (Real-Zeit-Programme, Batchprogramme) ausgeführt werden.

Welche Konsequenzen ergeben sich daraus?

Einerseits ist es mit diesen TP-Funktionen möglich, die gesamte Peripherie der PDP-11/45 und deren Hauptspeicher zu benützen. Dies schafft eine größere Verfügbarkeit von Datenmengen auf dem S/7, was sich sonst nur durch mehrmaliges Plattenwechseln ermöglichen ließe. Ein Weiterarbeiten laufender Programme während dieses Zeitraumes ist aber nicht möglich, wodurch sich für den laufenden Betrieb Einschränkungen ergeben. Dies wird durch die Rechnerverbindung hintangehalten.

Andererseits aber lassen sich die Daten über eine weitere Verbindung von der PDP-11/45 an den Unterstützungsrechner DECsystem-20 weiterleiten (Abb.2b). Damit eröffnen sich für den in seinen Möglichkeiten beschränkten S/7 Benutzer neue Dimensionen der Datenverarbeitung. Gewonnene und vorverarbeitete Daten können von dem S/7 über die PDP-11/45 an das DECsystem-20 gebracht werden und dort weiterführenden Auswertungen unterworfen werden. Große Massenspeicher und nahezu unbeschränkte Programmlängen gestatten die Ablage großer Mengen an Versuchsdaten und deren Verknüpfung zu Ergebnissen, die auf dem gleichen Transportweg an das

S/7 rückgesendet werden können.

Wir sind nun an einem Punkt angekommen, wo wir sagen können, das S/7 ist wie ein Halb-wüchsiger. Es kann selbständig agieren und pflegt Konversation, braucht aber doch die starke Hand eines Erwachsenen (PDP-11/45), um mit größeren Problemen fertig zu werden.

Wie aber sieht die Zukunft aus?

Um es vorweg zu nehmen, die Zukunft hat schon begonnen! In Kürze werden mehrere Kleinrechner am Getreidemarkt an das S/7 angeschlossen und von ihm unterstützt. Es wird Bindeglied zwischen kleinen Prozeßsatelliten und dem DECSYSTEM-20 werden, ohne daß die PDP-11/45 dazwischen geschaltet werden muß (Abb.2c). Damit steigt der Benützungskomfort für die vom PRZ betriebenen Rechner, und Rechenleistung wird noch näher an die Institute herangetragen werden können. Dann wird das S/7 erwachsen geworden sein!

Die folgenden Beispiele mögen zeigen, wie einfach die Funktionen des vorgestellten Softwarepaketes zu verwenden sind.

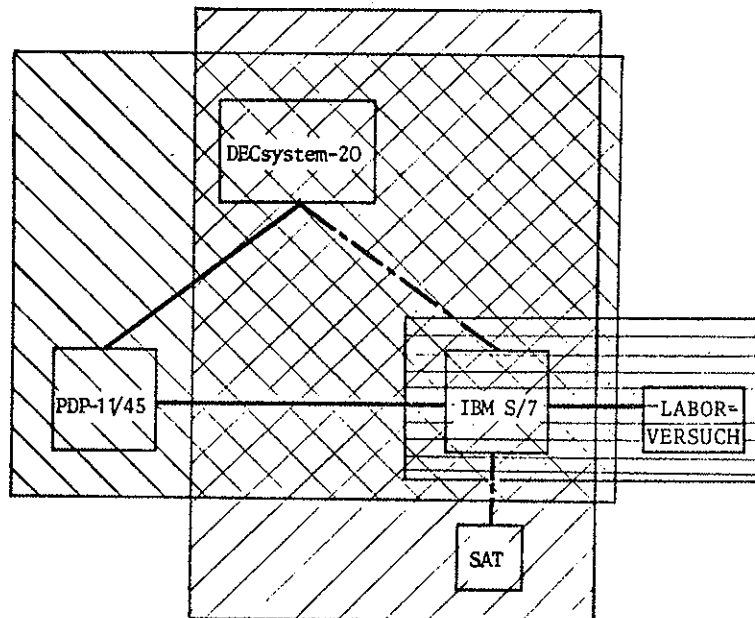
OP1:MCR

```
MCP>GET $SYLP=DK1:CONFILE.LST
MCP>PUT MT:MSDATA.DAT/EN=VOL156,DATALIB,SCAN25
MCP>RJE DK1:(300,300)INSTALL.EAT
MCP>OPR BITTE LEGEN SIE BAND V15602 EIN
MCP>XEC CONVRT
MCP>BYE
```

Ich hoffe, ich konnte Sie für die neuen Möglichkeiten an der S/7 interessieren und lade Sie ein, davon Gebrauch zu machen. Sollte dabei das eine oder andere einmal nicht funktionieren, bedenken Sie bitte, daß ein Programm bestenfalls so gut sein kann wie sein Schöpfer! Und ich bin auch nur ein Mensch!

Auf Wiedersehen bis zum nächsten FEEDBACK.

Abb.2: Schrittweise Realisierung der Datenfernverarbeitung am Getreidemarkt



- a) IBM S/7 als Stand-alone-System am Getreidemarkt
- b) Rechnerverbund IBM S/7 - DECSYSTEM-20 mit PDP-11/45 als Schnittstelle
- c) Rechnerverbund Laborsatellit - DECSYSTEM-20 mit IBM S/7 als Schaltstelle
- Rechnerverbund
- - - Rechnerverbund projiziert



JAHRESBERICHT 1977

KURZBESCHREIBUNG EINER AUSWAHL VON PROJEKTEN, DIE IM BERICHTS-
JAHR 1977 AM PROZESSRECHENZENTRUM DURCHGEFÜHRT WURDEN.

AUSWERTUNG BIORHYTHMISCHER FUNKTIONSSCHWANKUNGEN
(II. Chirurgische Universitätsklinik)

Inhalt des Projektes ist die Analyse des Einflusses von äußeren Ereignissen auf bestimmte biologische Funktionen des Menschen. Hierfür wurden entsprechende Funktionen am Publikum in Theatern und Opernhäusern während der Vorstellung aufgenommen und nachfolgend am Prozeßrechner ausgewertet. Ziel der Auswertung ist die Möglichkeit, den durch die Wahrnehmungsorgane bedingten Kontakt zwischen Künstler und Publikum durch eine objektive Maßzahl (zeitabhängige Variable) zu beschreiben.

Die biologischen Funktionen werden zunächst analog vorverarbeitet und die Ergebnisse von einem 7-Kanal-Analogbandspeicher registriert. Für die synchrone Aufnahme von Daten und Zeit wurde ein Programm entwickelt. Das Datenmaterial wurde auf 3 Datenplatten gespeichert. Die vorgesehenen Indizes wurden computergerecht umgeformt, sodaß im ersten Auswerteschritt von den insgesamt 11 Parametern zunächst 7 Parameter ausgewertet und auf dem Lineprinter grafisch dargestellt wurden; die graphische Darstellung bezieht sich auf jeweils ein originales Tonband mit einer Sprechdauer von 50 Minuten. Parallel zur graphischen Registrierung wurden alle Daten in Minutenintervallen geprintet.

Im zweiten Auswerteschritt wurden die restlichen Parameter sowie ein summatorischer Index gebildet. Auch diese Daten stehen graphisch sowie in Form einer Zahlenmatrix zur Verfügung.

PROZESSRECHNERSTEUERUNG EINER ELEKTRONENSTRAHL-FEINBEARBEITUNGS-
ANLAGE
(Inst. für Industrielle Elektronik)

Es wurde ein Software-Paket erstellt, das die Bearbeitungsanweisungen für das Schweißen und Gravieren mit Elektronenstrahlen verarbeitet und die nötigen Daten für die numerische Steuerung bereitstellt.

Der Schwerpunkt der Arbeiten im abgelaufenen Forschungsjahr lag bei der Entwicklung der Software für die Steuerung der Elektronenstrahl-Bearbeitungsanlage mit Hilfe einer numerischen Steuerung, die ihrerseits durch einen Kleinrechner (PDP11/10) geführt wird.

Es wurde eine Programm-Bibliothek erstellt, die im wesentlichen aus den folgenden Teilen besteht:

- Dateneingabe über Terminal.
- Datenausgabe an Terminal und/oder Interface zur numerischen Steuerung
- Daten-Konversionsroutinen
- arithmetische Routinen(+,-,x,:)mit der für die speziellen Berechnungen nötigen Genauigkeit (64 bit, Fixkomma)
- textinterpretierende Routinen für frei wählbare Syntaxelemente, das heißt die Schaffung der Möglichkeit, die Bearbeitungsanweisungen nicht in einer fixen Syntax einzugeben, sondern mittels Sprachelementen, die dem jeweiligen Anwendungszweck angepaßt werden können.

PULSMAGNETOMETER UND PULSFELDAUSWERTUNG
(Inst. für Experimentalphysik)

Das Pulsmagnetometer ist ein Gerät, das Messungen in Feldern bis zu 20 G im Temperaturbereich von 4,2 K bis 300 K erlaubt. Die anfallenden Daten werden on-line in digitaler Form zum Computer geleitet, wo sie integriert sowie differenziert werden und so dem Experimentator einen raschen Überblick über den Verlauf eines Versuches ermöglichen. Auf diese Weise wurde die Anisotropie an CO , an TbCu_2 u. GdCu_2 gemessen. Außerdem wurde mit der Bestimmung der Molekularfeldkoeffizienten $n(3d-4f)$ am System $\text{Er}_6(\text{Fe, Mn})_{23}$ begonnen. Eine Diplomarbeit untersucht mit der oben beschriebenen Anlage das System $\text{Zr}(\text{Fe, Al})_2$, um es mit einer theoretischen Modellvorstellung zu vergleichen.

Zur Auswertung der on-line erfaßten Meßdaten wurde ein Programm erstellt, das eine sehr vielseitige und allgemeine Behandlung der vom Pulsmagnetometer gelieferten Werte für dM/dt (M =Magnetisierung) und dH/dt (H =Feld) zuläßt. Die wesentlichsten zu berechnenden Kurven sind: $M(H)$, $M(1/H^n)$, $M^2(H/M)$. $M(H)$ dient der Feststellung der Magnetisierungskurve. $M(1/H^n)$ läßt die Bestimmung der Sättigungsmagnetisierung (Extrapolation $H \rightarrow \infty$) zu. $M^2(H/M)$ stellt einen sogenannten "Arrot-Plot" dar, der die Prüfung von theoretischen Modellvorstellungen zuläßt. Daneben stehen noch sehr genaue Integrations- sowie Differentiationsverfahren zur Verfügung. Dieses als Frage-Antwort geschriebene Programm erlaubt dem Experimentator, experimentsspezifisch einen ganz bestimmten Programmablauf zusammenzustellen und so die Auswertung dem physikalischen Problem anzupassen.

MASSENSPEKTROMETRIE (Inst.für Physik. Chemie)

Zunächst wurde ein flexibles, allgemein verwendbares Aufnahmeprogramm erstellt, welches die durch das Quadrupol-Massenspektrometer erzeugten Spektren auf verschiedene Arten aufnimmt und speichert. Gleichfalls wurde das Auswerteprogrammpaket soweit komplettiert, daß die prozeßrechnerunterstützte Meßmethodik in die Katalysforschung des Institutes voll integriert werden konnte. Zur weiteren Datenaufbereitung wurden einige Plotter-Programme erstellt, um die Meßergebnisse in anschaulicher und ablagegerechter Form ausgeben zu können. Weiters wurden Vorarbeiten geleistet in Hinblick auf eine Kombination mit der ebenfalls prozeßrechnergesteuerten Desorptionsapparatur, um die Messungen automatisieren zu können.

ENTWICKLUNG VON SCHRAUBENVERDICHTERN (Inst. für Maschinenelemente)

Am Institut für Maschinenelemente ist ein Schraubenverdichterprüfstand aufgebaut, dessen Ziel es ist, Wirkungsgradverbesserungen an einspritzölgekühlten Schraubenverdichtern zu bewirken. Es werden die Einflüsse der einzelnen Betriebsparameter auf bestimmte Kenngrößen ermittelt, um durch Parameteroptimierung eine Verbesserung des Wirkungsgrades zu erzielen. Da diese Parameter jedoch nicht in allen Fällen die am Versuchsstand in elektrischer Form anfallenden Meßgrößen sind, müßten langwierige Berechnungen zwischen zwei Meßreihen durchgeführt werden um die Kenngrößen zu bestimmen. Die händische Auswertung zwischen zwei Meßreihen während des Versuchsablaufes ist aber praktisch unmöglich, sodaß auch wegen der Fülle der anfallenden Daten

ein Anschluß an den Prozeßrechner IBM-S/7 durchgeführt wurde. Der Rechner übernimmt die Steuerung des Meßzyklus, die Auswertung der Meßdaten und die Anfertigung des Meßprotokolls. Die ausgewerteten Meßergebnisse stehen jederzeit am Bedienungspult des Prüfstandes zur Verfügung und werden überdies, falls gewünscht, auf Magnetplatte für weitere Auswertungen gespeichert.

DESORPTION

(Inst. für Physikalische Chemie)

Für die Temperaturprogrammierte Desorption (TPD) wurde eine prozeßrechnergesteuerte Meßapparatur gebaut. Der Rechner übernimmt dabei sowohl Regelung der maximal 5 Öfen, als auch Datenaufnahme + Speicherung und Sicherheitskontrollen (= wichtig für Nachtbetrieb!). Die mit Trägergas und GC-Detektoren arbeitende Anlage arbeitet zufriedenstellend. Im zweiten Halbjahr wurde eine Kombination der Anlage mit dem ebenfalls prozeßrechnergesteuerten Quadrupol-Massenspektrometer vorgenommen.

ANALYSEMETHODEN

(Inst. für Physikalische Chemie)

Die hauptsächlich in BASIC und FORTRAN geschriebenen Programme umfassen derzeit iterativen Polynomialausgleich (insbesondere von Spektralkurven), Pattern recognition von IR- und GC-Daten und Datenweiterverarbeitungsprogramme für thermodynamische Datenerrechnung von Gaschromatogrammen. Die Programme wurden mit manuell eingegebenen Daten erprobt, eigentliche Anwendung werden sie erst nach Fertigstellung der Verbindung Getreidemarkt S/7 - Gußhausstraße DEC 20 finden.

AUGER-SPEKTROMETER UND AUGER-AUSWERTUNG (Inst. für Allgemeine Physik)

Im laufenden Jahr wurde das von der Prozeßrechenanlage erstellte Programm zur Prozeßsteuerung und -überwachung und Datenerfassung getestet. Seit März dieses Jahres wird es routinemäßig bei etwa 50% der Messungen am Auger-Spektrometer verwendet. Von wesentlichem Vorteil ist dabei, daß die auf diese Weise auf Massenspeicher abgelegten Meßergebnisse zu einem späteren Zeitpunkt jederzeit wieder ausgewertet werden können. Weiters wird durch die Prozeßsteuerung und -regelung die Messung der Proben halbautomatisiert und einmal eingegebene Meßparameter können durch Aufruf immer wieder aktiviert werden.

Zur Auswertung der Meßergebnisse wurden Programme in Fortran geschrieben. Es werden dabei mit der Least-Square-Fit Methode Eichspektren an das gemessene Spektrum angepaßt. Ein wesentlicher Faktor ist dabei die Wahl des korrekten Untergrundes. Es soll dabei die chemische Zusammensetzung der Oberfläche der analysierten Probe bestimmt werden. Die verschiedenen Auswerteprogramme werden an Proben bekannter Zusammensetzung getestet. Das Ziel ist, alle am Auger-Spektrometer erhaltenen Meßergebnisse vollautomatisch auszuwerten.

PROZESSIDENTIFIKATION

(Inst. für Elektrische Regelungs-, Steuerungs- und Antriebstechnik)

Ziel des Projektes war die Entwicklung eines Programmpaketes zur vollautomatischen Modellbildung eines stetigen Prozesses. Das Ergebnis der Modellbildung ist ein lineares Abtastmodell 1. bis 4. Ordnung. Ist die Ordnung des untersuchten Prozesses nicht bekannt, werden alle in Frage kommenden Strukturen berechnet und die optimale Struktur an Hand statistischer Kriterien bestimmt.

Die Kopplung des Rechners mit dem zu untersuchenden Prozeß erfolgt on-line. Ein im Rechner erzeugtes Testsignal wird dem Prozeß zugeführt und simultan am Prozeßein- und -ausgang abgetastet und eingelesen. Die Vorbearbeitung der Daten erfolgt noch im Abtastzyklus, die Parameterschätzung anschließend im off-line Betrieb.

Das Gesamtprogramm beansprucht rund 16 k-Worte des Prozeßrechners PDP 11/34 unabhängig von der Anzahl der zur Modellbildung verwendeten Prozeßdaten.

SPEKTROSKOPISCHE UNTERSUCHUNGEN IM WELLENLÄNGENBEREICH 10-2500Å: TEILPROJEKT: LANGWELLIGE RÖNTGENSPEKTROSKOPIE (Inst. für Experimentalphysik)

Im abgelaufenen Jahr konnte nach anfänglichen Justierproblemen etwa in der Mitte des Jahres mit den kontinuierlichen Messungen begonnen werden, die nur durch maschinelle Störungen in der Vakuumpumpe unterbrochen wurden. Dabei hat sich die eingesetzte Prozeßsteuerung mit der PDP 11/10 vor allem in jenem Bereich, der den Meßablauf kontrolliert, im 24^h Betrieb, bewährt. Für das Jahr 1978 ist vor allem bei der Auswertung der Meßdaten mittels Glättungs- und Entfaltungungsverfahren eine intensive Benützung der PDP 11/45 zu erwarten. Dabei ist die direkte Verbindung PDP 11/10 - PDP 11/45, die im vergangenen Jahr installiert wurde, vorteilhaft.

KRISTALLINITÄTSBESTIMMUNG (Inst. für Angewandte Physik)

In Fortsetzung des "Forschungsprojektes Hochpolymere" wurden eine Reihe von Kristallinitätsbestimmungen an isotaktischem Polypropylen durchgeführt. Außerdem wurden am Quellprogramm eine Reihe von Veränderungen und Verbesserungen vorgenommen. In Vorbereitung eines weiteren Kunststoffprojektes wurde das Programm auch auf eine Verwendung für Polyäthylenuntersuchungen erweitert.

VIBRATIONSMAGNETOMETER
(Inst. für Angewandte Physik)

Es wurden magnetische Messungen im Temperaturbereich von 4,2 bis 300 K bei Feldstärken bis 70 kG an der quasibinären Mischkristallreihe $Y(\text{Fe}_x\text{Co}_{1-x})_2$ (MgCu₂-Typ) durchgeführt. Im Bereich um $x=0,167$ ist eine starke Abnahme sowohl des magnetischen Moments als auch der Curietemperatur zu beobachten. Für $x=0,12$ ist bei 4,2 K keine magnetische Ordnung gegeben. Aus der Extrapolation der Magnetisierungswerte der Fe-enthaltenden Proben gegen verschwindende Fe-Konzentration folgt, daß die Co-Atome in $Y\text{Co}_2$ kein magnetisches Moment besitzen. Die magnetischen Eigenschaften dieser Verbindung werden jedoch durch mechanische Deformation sehr stark beeinflusst. Der Verlauf des mittleren 3d-Momentes über der Fe-Konzentration ergibt, daß magnetische Ordnungszustände erst ab Konzentrationen auftreten, für die eine Wechselwirkung zwischen direkt benachbarten Fe-Atomen möglich ist.

GESCHWINDIGKEITSSPEKTROMETER
(Inst. für Allgemeine Physik)

Für ein Geschwindigkeitsspektrometer wurde vom Institut eine Datenerfassung mittels eines LSI 11 Mikroprozessors geplant. Die Lieferzeit der Datenerfassungsanlage wurde dadurch überbrückt, daß auf dem analogen System des Prozeßrechenzentrums die wichtigsten Assemblerprogramme entwickelt wurden. Dazu wird einerseits die 11/45 als Editier- und Assembliereinrichtung, andererseits die 11/03 zum Testen der Programme verwendet. Weiters wurde eine Datenverbindung zwischen 11/03 und 11/45 hergestellt, die es ermöglicht, Programme von der 11/45 in die 11/03 zu laden und Daten von der 11/03 auf die 11/45 zu überspielen.

MIKROPROZESSOR-CROSS-SIMULATOR
(Inst. für Allgemeine Elektrotechnik)

Im Rahmen dieses Projekts wurde für den am Institut als Standardmikroprozessor eingesetzten Typ F8 ein Simulationsprogramm erstellt, das auf Grund von verschiedenen Direktiven den vom Assembler gelieferten hexadezimalen Code in gleicher Weise wie der Prozessor interpretiert. Dadurch kann auf übersichtliche Art und Weise der Ablauf eines Programms verfolgt werden. Der Prozeßrechner bot sich vor allem deshalb an, weil der Zusammenhang zu den peripheren Bausteinen bzw. Systemen und dem ablaufenden Programm nur hier möglich wäre. Eine Weiterführung des Projekts in Richtung Echtzeitsimulation ist geplant.

EINSATZ DES PROZESSRECHNERS ZUR UNTERSTÜTZUNG DES VORTRAGENDEN
FÜR DIE VORLESUNGEN ALLGEMEINE ELEKTROTECHNIK 1 UND 2
(Inst. für Allgemeine Elektrotechnik)

Die Arbeiten im Rahmen dieses Projektes standen im Jahr 1977 unter dem Zeichen der Umstellung von IBM 1800 auf die PDP11/34, die ausschließlich durch Personal der Prozeßrechenanlage durchgeführt wurde. Die umgestellten Programme erwiesen sich als wesentlich ablaufsicherer als auf der alten IBM-Maschine. Für das nächste Jahr ist die Erweiterung der Programmbibliothek geplant, sowie die Entwicklung eines universell einsetzbaren Programmpakets - in BASIC -, das die Eingabe von physikalischen Formeln beliebiger Komplexität auch während der Vorlesung gestattet, ohne umständlich kompilieren zu müssen.

MESSDATENAUSWERTUNG

(Inst. für Elektrische Meßtechnik)

Programmsystem für Meßdatenauswertung:

Eine einfache Testform der Arbeit befindet sich im Endstadium. Um größtmögliche Benutzerfreundlichkeit zu erreichen, wurde ein einfacher Befehlsvorrat von 8 Grundbefehlen gewählt, die den Aufruf und Abruf der I/O- und numerischen Routinen verwalten.

Die Unterprogramme sind, um eine leichte Erweiterbarkeit des Programmpakets zu erreichen, als eigene Files abgespeichert. Aus zusätzlichen Informationsfiles ist ersichtlich, welche Parameter in welcher Form ein spezielles Unterprogramm erwartet. Die Parameter werden durch das System dem Unterprogramm übergeben. Dadurch ist es dem Systemprogramm möglich, alle Parameter ohne Aufruf des entsprechenden Unterprogrammes abzufragen, was die Erstellung von Batch-Files gestattet. Diese speichern quasi einen kompletten Meßablauf mit allen Parametern und können mit einem Befehl abgerufen werden. Nächstes Ziel ist, eine endgültige und ausgefeilte Version des Systems an der Prozeßrechenanlage zu erstellen.

CAMAC-MODUL ENTWICKLUNG

(Inst. für Elektrische Meßtechnik)

Im Jahr 1977 wurden Prüf- und Testprogramme für zwei am Institut für Elektrische Meßtechnik entwickelte und konstruierte CAMAC-Module erstellt. Mit diesen in BASIC geschriebenen Programmen ist es möglich, die digitalen Funktionen des Moduls DUDAC (Dual-Digital-Analog-Converter) sowie sämtliche Funktionen des Moduls CCCT (Camac-Colour-Curve-Tracer) automatisch zu testen. Weiters wurden prinzipielle Überlegungen angestellt, ein Programmpaket zu erstellen, das die einzelnen Module in einem Crate aufgrund ihres Befehlssatzes identifiziert. Für die nähere Zukunft ist geplant, weitere modulspezifische Testprogramme zu schreiben sowie das oben erwähnte Programmpaket in einer Form zu implementieren, die einen schrittweisen Ausbau ermöglicht. Die Programmerstellung soll dabei immer stärker zur Prozeßrechenanlage verlagert werden.

RAUSCHUNTERSUCHUNGEN AN TRANSISTOREN (Inst. für Physikalische Elektronik)

Im Frequenzbereich unter 1 kHz zeigen bipolare Transistoren, Sperrschicht- und MOS-Feldeffekttransistoren eine $1/f$ Abhängigkeit von der Frequenz. Die Messung der eingangsbezogenen Leerlaufrauschspannungen im Frequenzbereich unter 10 Hz ist durch analoge Meßmethoden unmöglich. Die Anwendung eines Samplingverfahrens bringt hier weitgehende Abhilfe. Die diskretisierten Rauschspannungswerte werden vom Prozeßrechner eingelesen und dort verarbeitet. Fouriertransformation und Berechnung des Leistungsspektrums liefert die gesuchten Ergebnisse. Mit dieser Methode ist es möglich, die Spannungswerte bis zu Frequenzen von 0,01 Hz zu ermitteln. Dabei hat sich gezeigt, daß für diesen Frequenzbereich die $1/f$ Abhängigkeit des Rauschspektrums weiter gilt, wie es auch von der Theorie gefordert wird.

Durch die Umstellung der Prozeßrechneranlage entstand eine Unterbrechung in der Kontinuität unserer Messungen und es mußten für den Ersatz der selbstgebauten Interface-Geräte durch Camac-Module ausgedehnte Beratungsgespräche zwischen dem Institut für Physikalische Elektronik und den Fachleuten des Prozeßrechenzentrums geführt werden. Nach der Umstellung ist jedoch eine höhere Flexibilität und Leistungsfähigkeit bei unseren Meßauswertungen zu erwarten.

LEITUNGSSCHUTZ

(Inst. für Elektrische Anlagen und Hochspannungstechnik)

Für die Berechnung der Fehlerentfernung bei Leitungen aus dem Einschwingvorgang bei Kurzschluß sind eine Reihe von Algorithmen angegeben. Im Rahmen einer Diplomarbeit wurden zwei der bekanntesten Algorithmen programmiert, um einen Vergleich zu ermöglichen. Beide Algorithmen gehen von der Differentialgleichung aus. Sie unterscheiden sich in der Methode der Verbesserung der Koeffizienten für das R und L bestimmende Gleichungssystem.

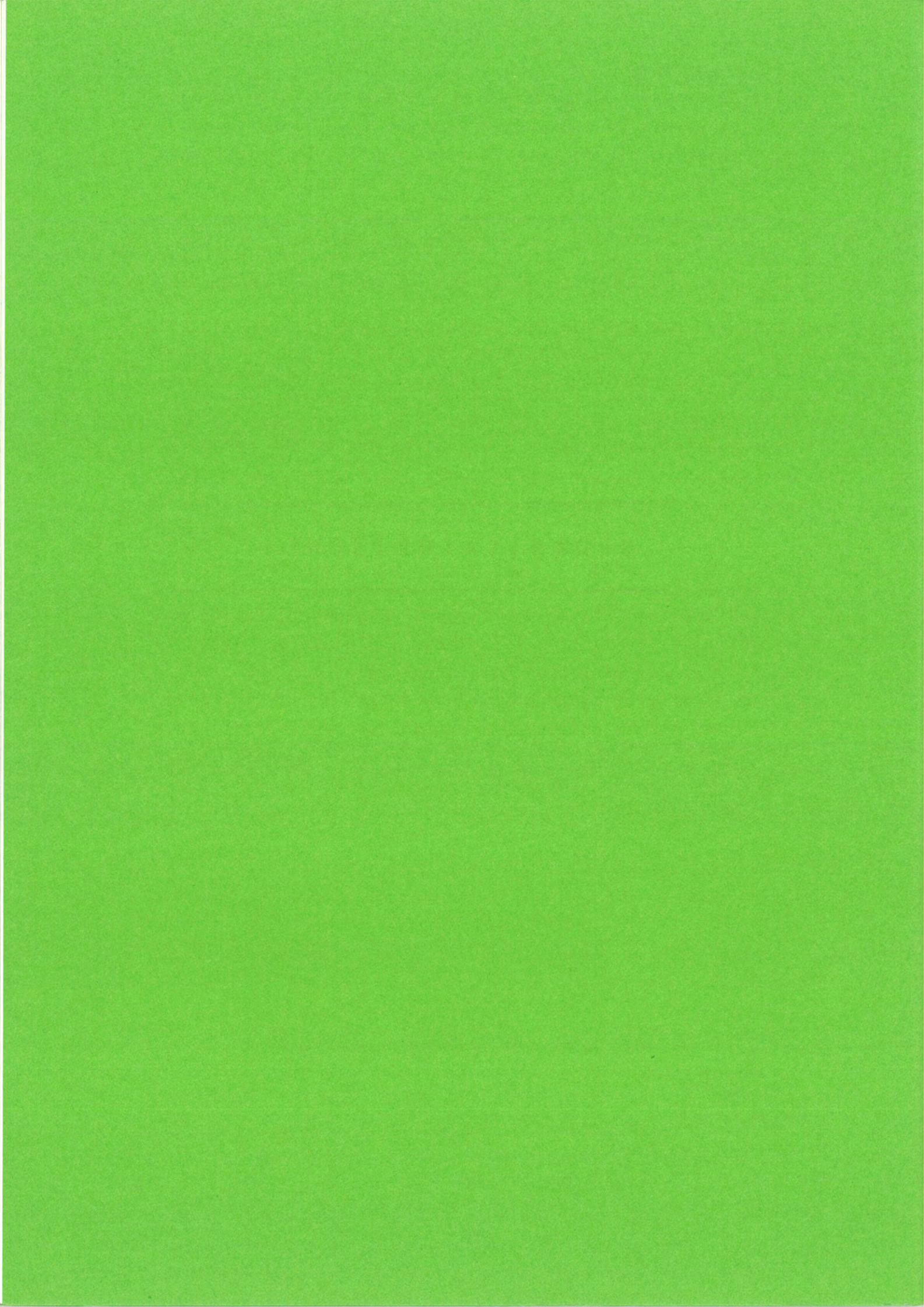
Eine Hauptaufgabe bestand in der Erfassung der Meßwerte durch Abtasten eines Einschwingvorganges, der an einem Leitungsmodell simuliert wurde. Diese Meßwerte liefern die Ausgangsdaten für die beiden Algorithmen. Die Auswertung und Diskussion der Ergebnisse wird derzeit durchgeführt. Ein weiterer Schritt ist die Programmierung der Algorithmen auf einem Mikroprozessor.

DIGITALRECHNERMETHODEN FÜR PLANUNG UND BETRIEB VON ELEKTROENERGIE-
SYSTEMEN, TEILGEBIET: ON-LINE LASTFLUSS-SCHÄTZUNG
(Inst. für Elektrische Anlagen und Hochspannungstechnik)

Im Rahmen des Forschungsprojektes "Digitalrechnermethoden für Planung und Betrieb von Elektroenergiesystemen" wurden auf dem Teilgebiet der on-line Lastflußschätzung folgende Arbeiten durchgeführt:

Zu einem bereits vorhandenen State Estimation Programm nach dem Admittanzmatrizenverfahren wurde eine weitere Version erstellt, welche mit einem Funktionalmatrizenalgorithmus arbeitet. Mit dem Vergleich dieser beiden Verfahren hinsichtlich ihres Verhaltens bei verschiedenen Meßgerätegenauigkeiten, Meßgerätekonfigurationen und eventuellen groben Datenfehlern wurde begonnen.

ORGANISATORISCHES



DAS PRZ STELLT SICH VOR:

VORSTAND: o. Univ.Prof. Dipl.-Ing. Dr. techn. Herbert STIMMER

LEITUNG:

Dipl.-Ing. Dr. techn. Manfred PAUL

Sprechstunde

Mo - Fr

n.V.⁺⁾

Zi. Nr.

339^{o)}

DOKUMENTATION:

Elfriede KASPAR

8-12

338^{o)}

Helga BRAUNER

8-12

337^{o)}

Erika MORAVEC

8-12

337^{o)}

Telefonverzeichnis

Prof. Dr. Herbert STIMMER

362

Dr. Manfred PAUL

376

Elfriede KASPAR

733

Helga BRAUNER

740

Erika MORAVEC

740

⁺⁾ Anmeldung bei Fr. E. KASPAR Kl. 733

^{o)} siehe Orientierungsplan, Gußhausstraße

BEREICH GUSSHAUSSTRASSE:

Betriebszeiten:

Operatorbetrieb: Mo. - Fr. 7.30 - 19.00 Uhr
(nach Vereinbarung bis 22.00 Uhr)

Rechenbetrieb: 0.00 - 24.00 Uhr bei den zentralen Anlagen DECsystem-20
und TP-Rechner (PDP-11/34)

Programmberatung: Mo. - Fr. 10.00 - 11.00 Uhr

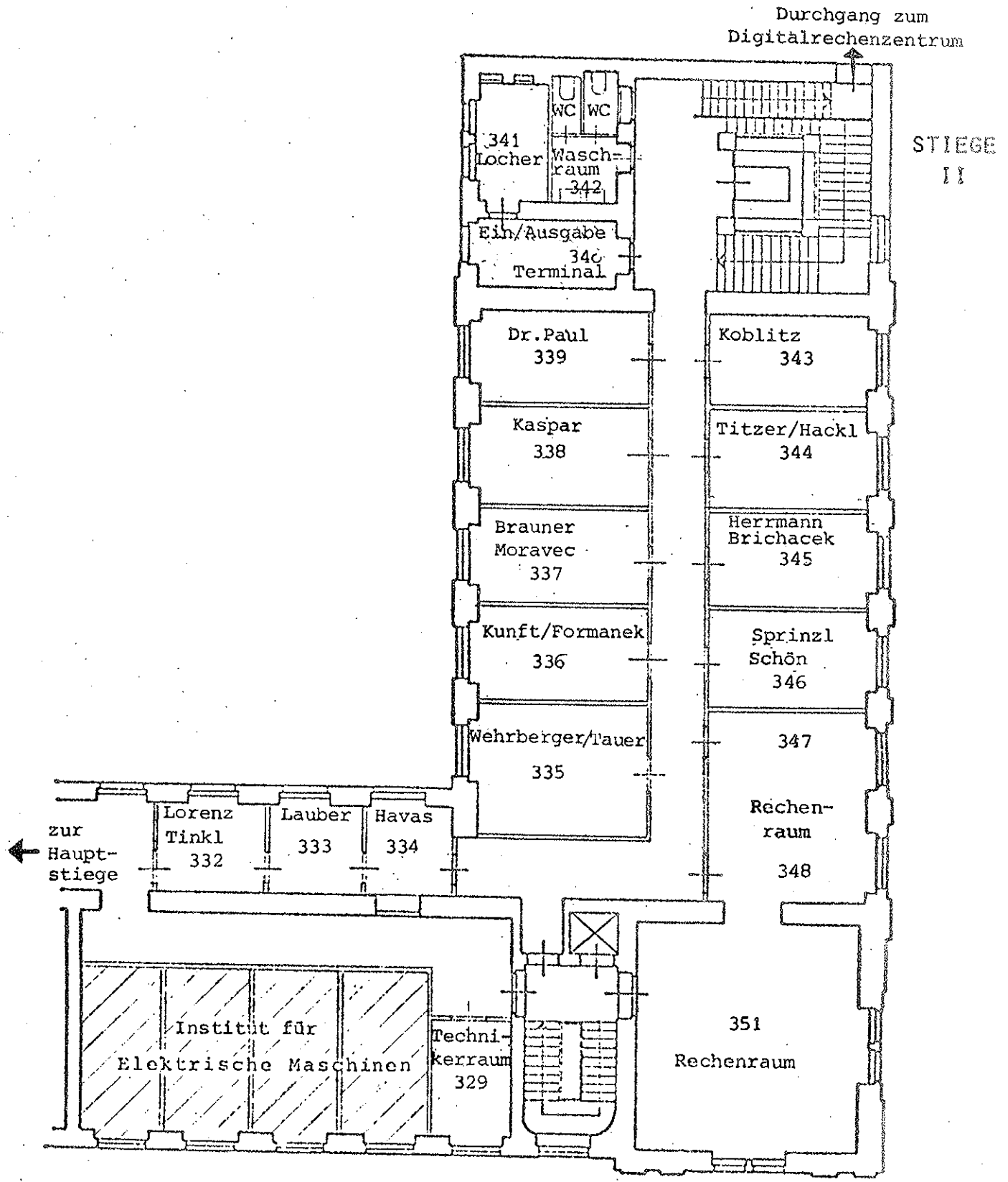
Wer ist zuständig für ... ?

	Zentraler Unterstützungs- rechner DECsystem-20	Prozeßrechner PDP-11/34 PDP-11/04 Teleprocessingrechner PDP-11/34
ORGANISATION	Havas	
OPERATING	{ Schön Sprinzl Moravec	
BETRIEBSSYSTEM		
Wartung	Sprinzl	Brichacek
Programmierung	Lorenz	{ Brichacek Tinkl
PROJEKTBETREUUNG U. PROGRAMMBERATUNG	{ Lauber Lorenz	{ Brichacek Lauber Tinkl
TECHNISCHE EIN- RICHTUNGEN U. PROZESS- PERIPHERIE		{ Brichacek Herrmann
TELEPROGESSING		{ Kunft Formanek Tinkl

Telefonverzeichnis

Rechenraum	105, 548	Dipl.-Ing. Wilhelm FORMANEK	344
Dipl.-Ing. Helmut HAVAS	629	Ing. Franz BRICHACEK	750
Dipl.-Ing. Dr. phil. Wolfgang LAUBER	691	Alfred HERRMANN	750
Peter LORENZ	691	Anton SPRINZL	105, 548
Peter TINKL	691	Peter SCHÖN	105, 548
Dipl.-Ing. Walter KUNFT	344	Erika MORAVEC	740

Orientierungsplan Gußhausstraße



G U S S H A U S S T R A S S E 2 5

BEREICH HAUPTGEBÄUDE:

Betriebszeiten:

Operatorbetrieb: Mo. - Fr. 8.00 - 12.00 u. 13.00 - 17.00 Uhr
Rechenbetrieb: 0.00 - 24.00 Uhr nach Vereinbarung
Programmberatung: Mo, Di, Do, Fr 10.00 - 11.00 Uhr

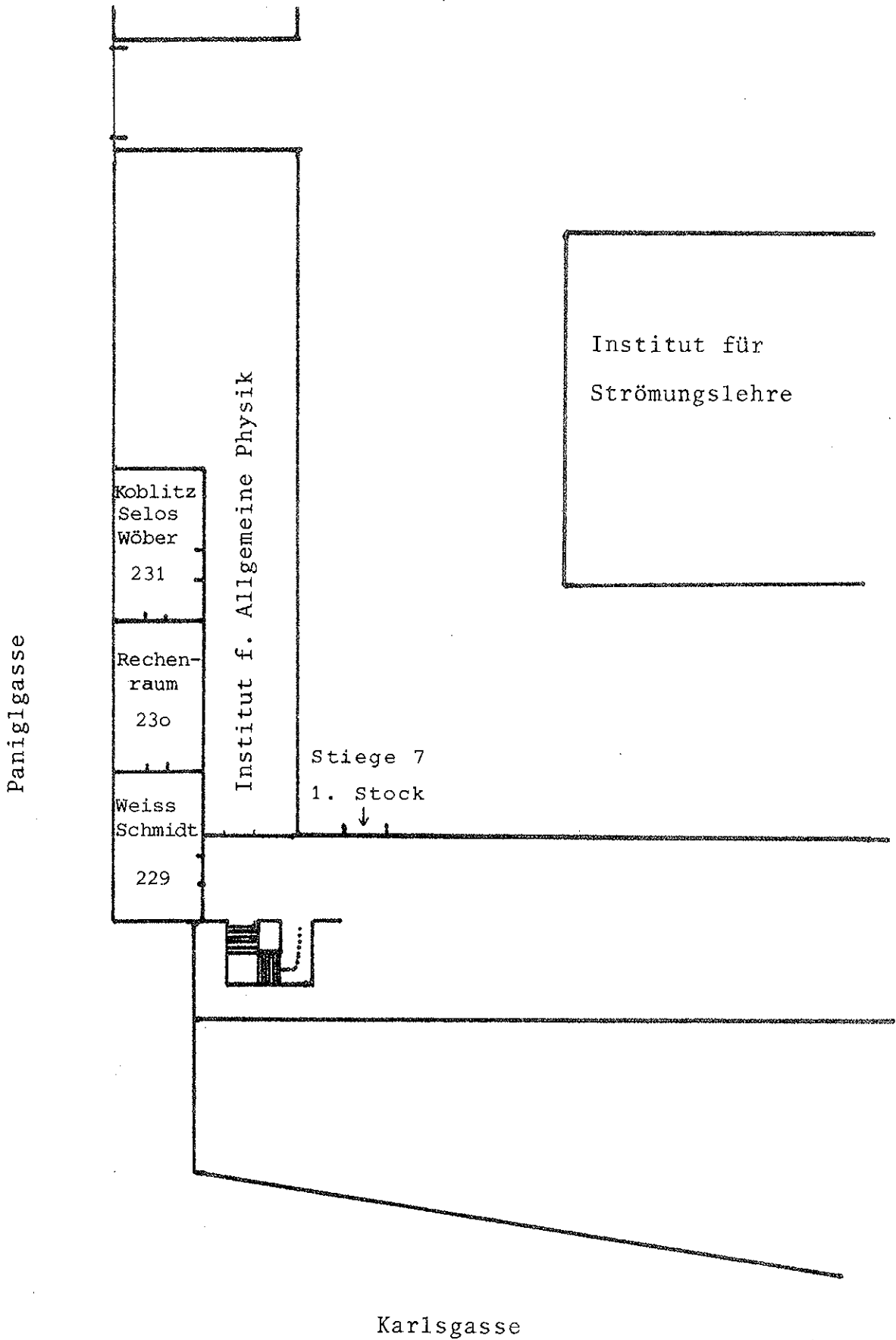
Wer ist zuständig für ... ?

BETRIEB/ORGANISATION	Koblitz
SYSTEM	Wöber
SPRACHPROZESSOREN } UTILITIES }	Wöber Selos
CAMAC	Selos
OPERATING	Schmidt
HARDWARE	Selos/Weiss

Telefonverzeichnis

Rechenraum	507
Dipl.-Ing. Werner KOBLITZ	699, 741
Wilfried WÖBER	699
Walter SELOS	699
Rainer SCHMIDT	109
Walter WEISS	109

Orientierungsplan Hauptgebäude



BEREICH GETREIDEMARKT:

Betriebszeiten:

Operatorbetrieb: Mo. - Fr. 8.30 - 16.30 Uhr
Rechenbetrieb: 0.00 - 24.00:nach Vereinbarung
Programmberatung: nach Vereinbarung

Wer ist zuständig für ... ?

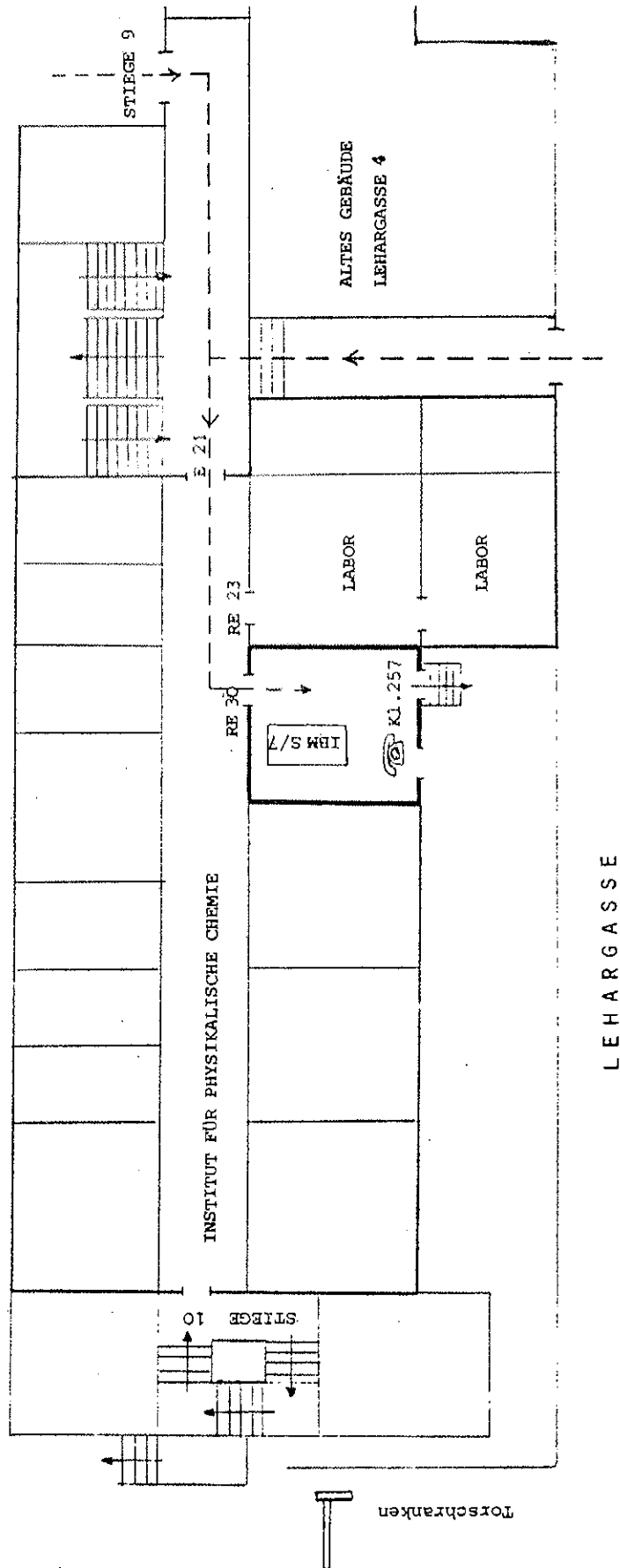
	<u>Wartung</u>	<u>Beratung</u>
SYSTEM	Tauer	Tauer
SPRACHPROZESSOREN	Tauer	Tauer Wehrberger
UNTERSTÜTZUNGSMODULE		
Prozeß-I/O	Tauer	Tauer
Plotter-Software	Tauer	Wehrberger
TP-Software	Tauer	Tauer
OPERATING		
Rechenzeit Reservierung	-	Titzer Hackl
Betriebsmittel	-	Titzer Tauer
Accounting	-	Wehrberger
HARDWARE		
Anschlußtafeln	Hackl	Tauer Hackl
Interfaces	Hackl	Tauer Hackl

Telefonverzeichnis

Rechenraum	GM 257
Dipl.-Ing. Günter WEHRBERGER	GH 745
Ludwig TAUER	GH 745
Friedrich TITZER	GM 257, GH 742
Walter HACKL	GM 257, GH 742

[GH: Gußhausstraße, GM: Getreidemarkt]

Orientierungsplan Getreidemarkt



HRN. D. I. A. BLAUENSTEINER

EDV-ZENTRUM, HYBRIDRECHENANLAGE

IM HAUSE