

edv

183

Herausgegeben
vonder Redaktion
rechentechnrk
datenvera rbeitung
ooR 5.00 NI

aspekte

BASISRECHNER-
SYSTEME



Inhalt

Dr. Werner Schulze: Kleinrechner im Konzept der dezentralen Datenverarbeitung ...	2
Hardware	
Dietrich Neugebauer: Geräte für Grundkonfigurationen der Basisrechner des Kombinars Robotron	5
Michael Hamann: Anwendungskomplex Kommerzielle Basisrechnersysteme	16
Wolfgang Reese: Der Arbeitsplatz für Konstruktion und Technologie A 6454	19
Ronald Nordmeier, Joachim Sobe: Prozessrechnersysteme A 6491/ A 6492	21
Software	
Manfred Synowzik: Das Steuerprogrammsystem MOEX 1600 des Betriebssystems MOOS 1600	24
Michael Gnädig: Die Systemprogramme des Betriebs- systems MOOS 1600	29
Wilfried Klette: System zur Softwareentwicklung PoST 1600	37
Robert Syring: POS für die Basisrechner des Kombinars Robotron	39
Wolfgang Baier, Dr. Manfred Mikut, Wilburga Reller: Grafische Systemunterlagen des AKT A 6454	45
Arndt Lindner: Programmsystem MEPRIM für PRS A 6491 /A 6492	48
Dr. Reinhard Krien: K-I 600Fileprozessor unterstützt Datenbanksysteme	51
Emulation/Simulation	
Robert Syring: Emulationssystem robotron 4200/4201 — K 1600 ...	54
Annerose Albrecht, Volker Becher, Wolfgang Heerklotz: Der 08205-Z-Simulator SIMZ 1600 '	56
Service und Schulung	
Manfred Büge, Dietrich Menz, Werner Hentschel: Technischer Kundendienst für Basisrechnersysteme	59
Erhard Friske, Klaus Schurbaum: Schulung für Basisrechner des Kombinars Robotron	61

Содержание

Д р Вернер Шульце: Мини-ЭВМ в концепции де централизованной обработки данных	2
Аппаратная часть	
Дитрих Нойгебауэр: Аппаратура основной конфигурации базисных вычислительных машин комбината Роботрон	5
Михаэль Хаман: Потребительский комплекс «базисные системы коммерческих расчетов»	16
Вольфганг Резе: Рабочее место конструктора и технолога A 6454	19
Рональд Нордмайер, Иоаким Зобе: Система управляющих машин A 6491 /A 6492	21
Математическое обеспечение	
Манфред Синовики: Система управляющих программ МОЕК 1600 операционной системы MOOS 1600	24
Михаэль Гнздиг: Системные программы операционной системы MOOS 1600	29
Вильфрид Клетте: Система разработки матобеспечения ПОСТ 1600	37
Роберт Зиринг: ПОС для базисных вычислительных машин комбината Роботрон	39
Вольфганг Байер, др Манфред Микут, Вильбурга Реппер: Графическое матобеспечение АКТ A6454	45
Ардт Пинднер: Система программ МЕПРИМ для управляю- щих машин A6491/A6492	48
Д р Райнхард Крин: Процессор файла K 1600 помогает создавать банки данных	51
Эмуляция и моделирование	
Роберт Зиринг: Эмуляционная система роботрон 4200/4201 — K 1600 - - . . .	54
Аннерозе Альбрехт, Фолькер бехер, Вольфганг Керклоц: Симулятор C 8205-Ц симц 1600	56
Сервис и обучение	
Манфред Бюге, Дитрих Менц, Вернер Хенчель: Техническое обслуживание базисных вычислительных систем	59
Эрхард Фриске, Клаус Шурбаум: Подготовка операторов для базисных вычислительных машин комбината Роботрон	61

Contents

Dr. Werner Schulze: Small computers within the framework of decentralised data processing	2
Hardware	
Dietrich Neugebauer: Hardware for basic configurations of the basic computers made by the Robotron Combine	5
Michael Hamann: Application complex 'commercial basic computer systems'	16
Wolfgang Reese: The A 6454 workplace for design and technology	19
Ronald Nordmeier, Joachim Sobe: The A 6491 /A 6492 process computer systems	21
Software	
Manfred Synowzik: The MOEX 1600 control program system of the MOOS 1600 operating system	24
Michael Gnädig: The system programs of the MOOS 1600 operating system	29
Wilfried Klette: The POST 1600 system for software development	37
Robert Syring: POS for the basic computers made by the Robotron Combine	39
Wolfgang Baier, Dr. Manfred Mikut, Wilburga Reller: Graphic software ofAKTA6454	45
Arndt Lindner: The MEPRIM program system for PRS A 6491 /A 6492	48
Dr. Reinhard Krien: The K 1600 file processor assists data bank systems -	51
Emulation/Simulation	
Robert Syring: The robotron 4200/4201 K 1600 emulation system	54
Annerose Albrecht, Volker Becher, Wolfgang Heerldotz: The C 8205-Z simulator SIMZ 1600	56
Service and Training	
Manfred Bilge, Dietrich Menz, Werner Hentschel: Technical back-up service for basic computer systems	59
Erhard Friske, Klaus Schurbaum: Instruction for basic computers made by the Robotron Combine	61

edv-aspekte

Zeitschrift für spezielle Themas
der Informationsverarbeitung,
herausgegeben von der Redaktion
rechen-technik(datenverarbeitung,
1055 Berlin, Am Friedrichshain 22
Chefredakteur: Franz Lall 4 38 73 41
Redakteur: Hans Weit 4 38 73 16
Sekretariat: 4 38 72 33
Fernschreiber: 114 566

Redaktionsschluß: 8. 12. 1982

.Lizenz des Presseamtes beim Vorsitzenden
des Ministerrates der DDR Nr. 1529

edv-aspekte

Erscheinungsweise vierteljährlich zum Bezugs-
preis DDR 5,00 M je Heft
EDV-Artikel-Nr. 1331
Auslandspreise sind dem Zeitschriften-
katalog des Außenhandelsbetriebes
Buchexport zu entnehmen.

Schreibbar: Verlag Die Wirtschaft, Berlin
Druck: (140) „Neues Deutschland“, Berlin

Anzeigenverwaltung:
Berliner Verlag, 1056 Berlin,
Karl-Liebknecht Str. 29, Telefon: 2 70 33 02

Anzeigenannahme:

Berliner Verlag und Annahmestellen
in Berlin und in den Bezirken.
Zur Zeit gültige Anzeigenpreisliste Nr. 12

Im Ausland:

INTERWERBUNG GmbH – Gesellschaft
für Werbung und Auslandsmessen der DDR,
1157 Berlin-Karlshorst,
Rennau n-Druck er-Str. 89

Bestellungen nehmen entgegen:

Für die DDR:
Sämtliche Postämter, der örtliche Buchhandel
und der Verlag Die Wirtschaft Berlin
Inkasso-Zeitraum: vierteljährlich

Im Ausland:

In den sozialistischen Ländern für den zustiin-
digen Postzeitungsvertrieb. In allen anderen
Staaten der örtliche Buch- und Zeitschriften-
handel. Bestellungen des Buch- und Zeit-
schriftenhandels sind mit fichten an

BUCHEXPORT

Volkseigener Außenhandelsbetrieb der DDR,
DDR-7010 Leipzig, Leninstr. 16, Postfach 160
oder an Verlag Die Wirtschaft, Berlin,
DDR-1055 Berlin, Am Friedrichshain 22

Mitglieder des Redaktionsbeirates

Dr. Claus Goedecke - Dr. Rolf Grftler
Prof. Dr. se. Gerhard Keapier • Dr. Rolf Kilian
Hans Kunau - Walter Winch • Axel Rath sack
Prof. Dr. sc. Wolfgang Schoppa (Vorsitzender)
Dr. Wexner Schulze • Horst Stoll
Prof. Dr. Franz Stuchlik • Dr. Dieter Urban

Informationsverarbeitung für den Leistungsprozeß

Immer mehr verschmelzen Leistungsprozesse und die zugehörigen Informationsverarbeitungsprozesse. Auf diesen gesetzmäßigen Vorgang wurde schon mehrfach hingewiesen*. Bei der Anwendungstechnik von Geräten und anderen Hilfsmitteln für die Informationsverarbeitung spiegelt sich diese Entwicklung unter anderem in folgenden Tendenzen wider:

- Leistungsfähige technische Hilfsmittel für die Informationsverarbeitung am Arbeitsplatz der Endnutzer wurden zunehmend eingesetzt, so daß der Bedarf an technischer Unterstützung der jeweils speziellen Informationsverarbeitungsprozesse am Arbeitsplatz befriedigt werden kann.
- Der Charakter der Vermittlungsfunktion zwischen dem Endnutzer von Informationsverarbeitungstechnik und der Technik selbst wird stark reduziert oder beseitigt.
- Neuartige Informationstechnik ist notwendig.
- Die Aufwands- und Nutzelemente des Einsatzes technischer Hilfsmittel für die Informationsverarbeitung verschieben sich.

Diese Erscheinungen sind verbunden mit qualitativ veränderten Anforderungen an die Beteiligten (Endnutzer, Planer von Informationssystemen usw.), insbesondere aber an Entwickler und Produzenten von Informationstechnik. Vor Ihnen steht die Aufgabe, Verarbeitungsleistung (maschinelle Intelligenz), Speichermöglichkeit sowie allgemeine und spezielle periphere Geräte zur Kommunikation mit dem System der technischen Hilfsmittel am einzelnen Arbeitsplatz verfügbar zu machen. Erforderlichenfalls ist dazu eine direkte technische Verbindung zwischen den Gerätesystemen an verschiedenen Arbeitsplätzen und mit übergeordneter, leistungsfähigerer Informationsverarbeitungstechnik zu sichern. (Bezüglich der Verarbeitungsleistung und der Speichermöglichkeit kann dieses Ziel erreicht werden, indem dem entweder physisch am Arbeitsplatz bereitgestellt oder indem fernaufgestellte Verarbeitungskapazität und Speicher vor Arbeitsplatz aus zugreifbar gemacht werden; Kommunikationsperipherie ist auf jeden Fall am Arbeitsplatz zu installieren.) Technische Fortschritte des letzten Jahrzehnts, insbesondere der Elektronik, Mikroelektronik und Kommunikationstechnik haben wesentliche Voraussetzungen zur Lösung dieser Aufgabe geschaffen. Da die Anforderungen der Endnutzer an technische Hilfsmittel zur Informationsverarbeitung stark variieren, besteht die Notwendigkeit, eine sehr große Vielfalt solcher Hilfsmittel bereitzustellen. Die Datenverarbeitungs- und Büromaschinenindustrie hat diese Schwierigkeit bewältigt, indem Geräteklassen geschaffen wurden, die jeweils großen Gruppen von Anforderungen an Informationstechnik entsprechen und innerhalb dieser Modeile mit abgestufter Leistungsfähigkeit und mehr oder minder freien Konfigurationsmöglichkeiten ein weitgehendes Anpassen an die unterschiedlichen Nutzerforderungen gestatten. Eine dieser Geräteklassen ist die der Kleinrechner. Die praktische Verwertung der Möglichkeiten, die der Einsatz moderner Kleinrechner bietet, ist aber in erster Linie eine Frage der entsprechenden Ausgestaltung der langfristigen Datenverarbeitungskonzeptionen in Wirtschafts- und Verwaltungseinheiten. Die jüngsten Entwicklungen auf dem Gebiet der Informationstechnik müssen Anlaß sein, diese Konzeptionen zu überprüfen und zu vervollkommen. Das ist ein folgerichtiger Prozeß.

Die vorliegende edv-aspekte-Ausgabe soll diesen Prozeß unterstützen, indem sie schwerpunktmäßig Grundkonfiguration und Basisbetriebssoftware der Basisrechner erläutert. So werden als Beispiele arbeitsplatzbezogener Konfigurationen der Arbeitsplatz für Konstruktion und Technologie AKT A 6454, die kommerziellen Basisrechnersysteme KBR A 6401 /A 6402 und die Prozeßrechnersysteme PRS A 6491 /A 6492 beschrieben. (Eine weitere arbeitsplatzbezogene Konfiguration, der Bildauswertungsarbeitsplatz BVS A 647x wurde bereits in rd 9/82 vorgestellt.)

Einen Schwerpunkt des Heftes bilden auch die Beiträge zu den höheren Programmiersprachen für die Basisrechner A 64xx, da sie für die Einordnung der Basisrechner als Hilfsmittel zur dezentralen Datenverarbeitung ebenfalls von Bedeutung sind.

Dr. Werner Schulze

■ Siehe r. B. rd 1/81, S. 33

Kleinrechner im Konzept der dezentralen Datenverarbeitung

Dr. Werner Schulze
VEB Robotron-Vertrieb Berlin

Kleine Rechner — große Leistung

Der historische iJrsprung der Kleinrechner ist in den bereits in den 50er Jahren entwickelten Geräten zur Steuerung von materiellen Prozessen (Prozeßrechner für technologische Prozesse, Steuerrechner für die Experimentautomatisierung u. a.) zu sehen. Gegenüber diesen, aus heutiger Sicht dedizierten Geräten, existieren für moderne Kleinrechner nahezu keine Einschränkungen des Einsatzgebietes auf bestimmte Informationsverarbeitungsaufgaben mehr, weil diese Geräteklasse ein außerordentlich breites Leistungsspektrum besitzt. Zur Verbesserung der Übersichtlichkeit werden innerhalb dieses Spektrums international unterschieden:

- Mikro-Kleinrechner („Microminjs“)
- traditionelle Kleinrechner
- Super-Kleinrechner („Superminis“).

Wesentliche technische Charakteristika und Haupteinsatzgebiete sind in Tab. 1 zusammengestellt. Die Tatsache, daß Kleinrechner gegenwärtig einen Anforderungsbereich von einfachen Steuerungsaufgaben bis zu Aufgaben abdecken, deren Lösung in der Vergangenheit die Leistungsfähigkeit cities Großrechners notwendig machte, führt international zu einer starken Zunahme der Anzahl solcher Geräte (Tab. 2).

Die Kleinrechner des VEB Kombinat Robotron

Das Kombinat Robotron hat sich in seinem Erzeugniskonzept langfristig auf die eingangs dargelegten Entwicklungstendenzen auch auf dem Gebiet der Kleinrechentechnllk eingestellt. Dabei konnte auf langjährige Erfahrungen bei Entwicklung und Produktion von Erzeugnissen dieser Klasse zurückgegriffen werden. Bereits 1961 wurde in der DDR der Kleinrechner 5ER 2 serienmäßig produziert. Ihr folgte der leistungsstärkere D4A (C 8205), der auch im Ausland starke Verbreitung fand. Die bisher größte Bedeutung für die Volkswirtschaft der DDR erlangten jedoch die Rechner der Familie robotron 4000 mit dem Prozeßrechnersystem (PRS) 4000 und den Kleinrechnersystemen (KRS) 4200 ипд 4201. (Der Kleinststeuerrechner robotron 4100 sei der Vollständigkeit wegen genannt. Er wurde für spezielle Gerätesteuerungen genutzt.)

Die weitere Entwicklung auf dem Gebiet der Kleinrechentechnik im Kombinat Robotron wird durch die Einordnung der gegenwärtig produzierten Geräte dieser Klasse in das Erzeugnisprogramm Dezentrale Datentechnik bestimmt /1, 2/. Dieses

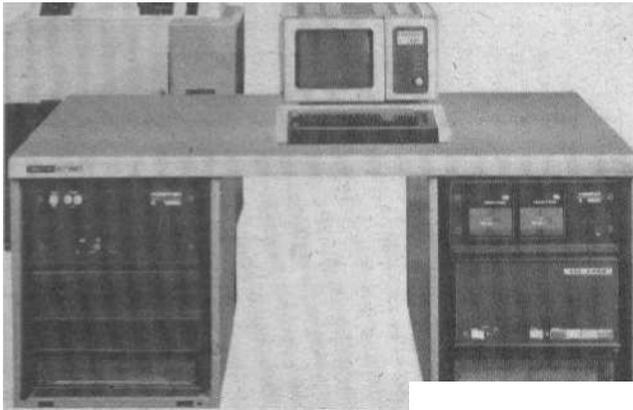
Tab. 1 Technische Charakteristika und Hauptanwendungsgebiete von Kleinrechnern

Leistungsgruppe	wesentliche technische Charakteristika	Haupteinsatzgebiete
Mikro-Kleinrechner	LSI-Technologie Wortbreite 8 (oder 16) Bit Arbeitsspeicher bis 64 K Byte eingeschränktes Spektrum peripherer Geräte	lokale Steuerungsaufgaben für technologische Prozesse, Experimente Oberwiegend OEM-Einsatz
traditionelle Kleinrechner	LSI-Technologie Wortbreite (8 oder) 16 Bit Arbeitsspeicher bis 1 M Byte umfangreiches Spektrum peripherer Geräte (einschließlich aufgabenbezogener Sonderbaugruppen und -geräte) Betriebssysteme mit höheren Programmiersprachen	alle Aufgaben der Informationsverarbeitung, die von der technischen Leistungsfähigkeit her abgedeckt werden können: Steuerungsaufgaben ökonomische Datenverarbeitung, Lösung numerischer Aufgaben, Textverarbeitung, grafische Datenverarbeitung, Bud- verarbeitung Einsatz als Ein- und Mehrplatzsysteme Einsatz in Mehrprozessorsystemen und Verbundnetzen
super-Kleinrechner	VLSI-Technologie wie Wortbreite (16 oder) 32 Bit Adreßraum bis 4,3 G Byte virtuelle Speichertechnik umfangreiches Spektrum peripherer Geräte, komfortable, teilweise hardwaregestützte Betriebssysteme mit umfangreichem Programmiersprachenangebot	traditionelle Kleinrechner mit Konzentration auf verarbeitungssensitive Aufgaben (beispielsweise Mustererkennung, Signalverarbeitung, Bildverarbeitung), damit Erschließung weiterer Anwendungsgebiete

Tab. 2 Entwicklung des Umfangs von Kleinrechnerinstallation im NSW (nach /2/)

	Anzahl Installationen (%)	Gesamtwert Installationen (%)
1980	100,0	100,0
1981	124,8	130,9
1982	153,0	169,6
1983	187,2	216,8
1984	227,4	273,9
1985	273,9	341,3

wurde unter Berücksichtigung der objektiven Entwicklungsgesetzmäßigkeiten auf dem Gebiet der Informationsverarbeitung konzipiert. Das gilt auch für die Stellung der Kleinrechner innerhalb des Erzeugnisprogramms. In ihm sind derzeit sowohl Kleinrechner enthalten, die der Gruppe der Mikro-Kleinrechner zuzuordnen sind, als auch solche, die zur Gruppe der traditionellen Kleinrechner gehören. Letztere sind unter der Chiffre A 64xx zusammengefaßt. Da sie für eine basisnahe Datenverarbeitung bestimmt sind, werden sie in der Terminologie des Kombinats Robotron als Basisrechner (A 640x) und arbeitsplatzbezogene Konfigurationen auf der Grundlage von Basisrechnern (A 64xx) bezeichnet.



Kommerricü technersystem tabotron A 6401 Werkrote

Basisrechner — zentral oder dezentral?

Um die Ste'lung der Basisrechner und der daraus abgeleiteten arbeitsplatzbezogenen Konfigurationen im Rahmen der dezentralen Datenverarbeitung richtig bestimmen zu können, sind einige Bemerkungen zum Inhalt dieser Organisationsform der Informationsverarbeitung am Platze. Zunächst erscheint die dezentrale Datenverarbeitung als ein relativer Begriff. In bezug auf die Kombinatleitung ist beispielsweise die Datenverarbeitung in einem Kombinatbetrieb dezentral, wobei das technische Hilfsmittel für die dezentrale Datenverarbeitung im Kombinatbetrieb wahrscheinlich eine EDVA sein wird. Innerhalb des Kombinatbetriebes kann dann durchaus eine zentrale Datenverarbeitung realisiert sein. Kleinrechner spielen in dieser Auslegung des Begriffs der dezentralen Datenverarhei-

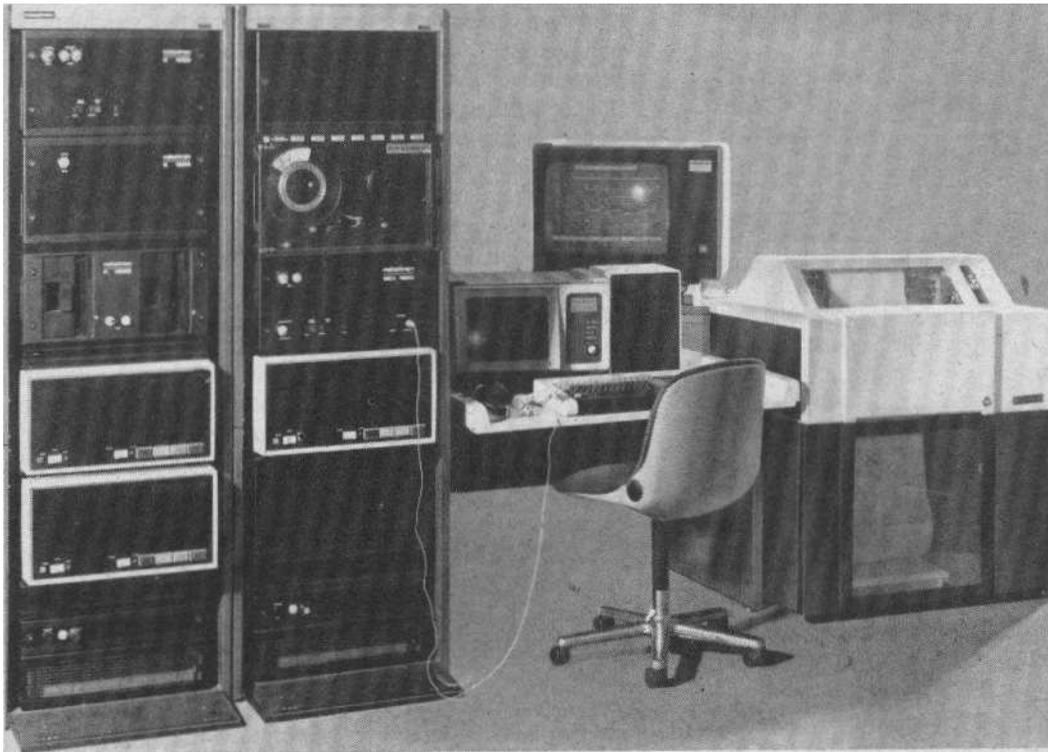
tung dann eine Rolle als technisches Hilfsmittel, wenn die Leistungsfähigkeit zur Bewältigung des Verarbeitungsumfanges am „dezentralen“ Ort ausreicht. Das wird in der Regel auf unteren Hierarchiestufen, beispielsweise in den Relationen Betrieb — Werk oder Betrieb Abteilung der Fall sein. In diesem Sinne wurden Kleinrechner bereits in der Vergangenheit zur „dezentralen“ Datenverarbeitung eingesetzt. Sie werden überwiegend im Regime stapelverarbeitung betrieben und für die Lösung einer Mehrzahl unterschiedlicher Verarbeitungsaufgaben der jeweiligen Wirtschafts- oder Verwaltungseinheit geruutzt („Rechenzentrums“-Betrieb).

Diese Form der Verwendung von Kleinrechnern wird zunächst auch weiterhin eine gewisse Bedeutung behalten. Der Begriff der dezentralen Datenverarbeitung ist nach heutigem Verständnis aber allgemeiner zu sehen: Er bezeichnet die Verarbeitung von Daten an dem Ort innerhalb einer Wirtschafts- oder Verwaltungseinheit, an dem sie entstehen und /oder an dem die Verarbeitungsergebnisse benötigt werden — in der höchsten Form im Zuge der Bearbeitung von Vorgängen (Transaktionen). Technische Hilfsmittel sind für die Unterstützung dieser Form der Datenverarbeitung um so besser geeignet, je enger sie den spezifischen Anforderungen des einzelnen Arbeitsplatzes eines Bearbeiters von Vorgängen (im allgemeinsten Sinn) angepaßt werden können. Moderne Kleinrechner besitzen in dieser Beziehung eine bemerkenswert hohe Flexibilität. Insgesamt sind Kleinrechner im Rahmen der dezentralen Datenverarbeitung derzeit bedeutungsvoll für

- den arbeitsplatzbezogenen Einsatz
- den Einsatz in dezentralisierten Kleinrechnerzentren ,
- die Ausführung von Hilfsfunktionen in Verbundsystemen (Steuerrechner, Vermittlungsrechner u. a.)

Konzeption	arbeitsplatzbezogene Konfiguration			Kurzbezeichnung dzw. Chiffre	
	Grundkonfiguration				arbeitsplatzbezogene Peripherie
	zentrale Verarbeitungseinheit	Arbeitspeicher	Rechnerstandard- peripherie		
Kleinrechner	K 1620	28 K Worte	Bedieneinheit Kassettenspeicherspeicher Festplattenspeicher Malnetbandgerät Folien Speicher Einheit Kassettenspeichernetzband- einheit Seriendrucker Paralleldruckgr	KBR A 6401	
	K 1630	124 K Worte		kBR A 6402	
Beispiele arbeitsplatz- bezogener Konfigurationen auf Grundlage von Basisrechner 1	den Anforderungen entsprechende Grundkonfigurationen			Ungetzeles Bild - Schirmterminal Bildschirmterminal	TBR A 6420
				Datenstation Betriebsdetertterminal	□1\$ A 6422
				Digitalesgerät grafisches Bildschirmgerät Plotter	9V5 A 645x
				Gralkbaugruppe Bildspeicher Lichtsteuerung Lichtdisplayprozessor Display	BV5 A 6472
				Prozelperipherie	PR5 A 49x

Abb. 1 Gerätetechnische Konzeption der Basisrechner A 640x und der darauf aufbauenden Konfigurationen A 64xx



Demonstrationsbeispiel der Kopplung eines robotron K 1630 mit dem Forschungsmuster des Sprach-Ein-/Ausgabegerätes SEA 1600. Derartige Lösungen bieten sich als arbeitsplatzbezogene Konfigurationen künftig dort an, wo B n de und Gesichtssinn des Bediennets bereits mit anderen Funktionen ausgelastet sind, wie zum Beispiel an grafischen Displays, an Sortierarbeitsplätzen usw. Werkfoto

Modulares Konzept bei Hard- und Software

Die Rechner A 64xx des VEB Kombinat Robotron sind für alle oben genannten Einsatzrichtungen geeignet, insbesondere aber für die zuerst genannten vorgesehen. Grundlage dieser Eignung ist einerseits das gerätetechnische Konzept, andererseits das Softwarekonzept der Basisrechner.

Bei der *Gerätetechnik* wird die erforderliche Flexibilität durch das streng modulare Konzept der Basisrechner erreicht. In sei

Rahmen können zunächst Grundkonfigurationen definiert werden, die aus unterschiedlich leistungsfähigen Verarbeitungseinheiten und Speichern sowie einer aus dem angebotenen Gerätespektrum auswählbaren Standardperipherie bestehen. Derartige Grundkonfigurationen können selbständig oder im Fernverbund mit gleich- oder übergeordneten Rechnern in dezentralisierten Kleinrechenzentren eingesetzt werden. Möglichkeiten für den arbeitsplatzbezogenen Einsatz werden geschaf-

fen, indem Grundkonfigurationen um arbeitsplatzbezogene, spezialisierte periphere Geräte ergänzt werden. Im einfachsten Fall kann es sich dabei um an Arbeitsplätzen von Endnutzern installierte allgemeine oder spezialisierte Terminals handeln (terminalorientierte Basisrechner). Abb. 1 verdeutlicht die gerätetechnische Konzeption. Ergänzend ist auf die mit der gewählten Lösung gegebenen Möglichkeiten für Weiterentwicklungen im Rahmen vorhandener und für die Entwicklung neuartiger arbeitsplatzbezogener Konfigurationen hinzuweisen (Foto oben).

Neuartige Konfigurationen entstehen auf gerätetechnischer Seite mittels Einbeziehen entsprechender arbeitsplatzbezogener peripherer Geräte. Weiterentwicklungen werden insbesondere durch Aufwertungen im Bereich der Grundkonfiguration erreicht. Dabei erweist sich das Einbeziehen der Basisrechner in das System der Kleinrechenzentren der sozialistischen Länder (SKR) als vorteilhaft, da über den auch an den Basisrechnern

Schicht	Architektur	Beispiele				
		Software für AKT A 645x		Software für BVS A 647x		
4	spezialisierte Anwendungssoftware	Paket für rotahans - symmetrische Teile	. . . ,	Paket für Pernerkundung der Erde	Paket für Bewegungsanalyse
3	aufgabenspezifische Basissoftware	Grafisches Betriebssystem (GK5)		Image Processing Utility (IPU)		
2	Netzsystemerweiterung		Treiber für arbeitsplatzspezifische Peripherie (siehe Abb. 1)	I		
1	Basisbetriebssystem	MOOS 1600				

Abb. 2 Architektur der Software für arbeitsplatzbezogene Konfigurationen der Basisrechner A 640x

Geräte für Grundkonfigurationen der Basisrechner des Kombinat Robotron

Dietrich Neugebauer
VEB Robotron-Vertrieb Berlin

verfügbaren SKR-Einheitsbus periphere Geräte aus dem Spektrum des SKR genutzt werden können.

Das *Softwarekonzept* entspricht dem gerätetechnischen Konzept. Grundlage ist eine am arbeitsplatzbezogenen Einsatz der Geräte orientierte Schichtenarchitektur der Software [Abb. 2]. Innerhalb der Schichten ist die Realisierung streng modular vorgenommen worden, so daß Erweiterungen und Modifikationen relativ einfach möglich sind.

Besonderes Augenmerk wurde den Programmiersprachen gewidmet, da sie eines der wichtigsten Hilfsmittel für die Kommunikation des Endnutzers mit dem System darstellen. Neben eingeführten höheren Programmiersprachen werden anwendungsbezogene Sprachen bereitgestellt, die in der Schicht der anwenderspezifischen Basissoftware angesiedelt sind. —

Ein weiterer Bestandteil des Softwarekonzepts ist die unter der Bezeichnung PDST 1600 geführte Technologie zur Entwicklung problemorientierter Software.

Nahtloser Übergang

Die Einführung neuer Datenverarbeitungs- und Büromaschinen ist aus der Sicht der Anwendung technischer Hilfsmittel zur Informationsverarbeitung stets verbunden mit dem Problems der Einordnung in den vorhandenen Bestand an derartiger Gerätetechnik, das heißt der Gewährleistung einer anwendungstechnischen Mindestkompatibilität, welche die weitere Nutzung vorhandener Datenbestände und Software ermöglicht.

Das gilt auch für das Verhältnis der Basisrechner A 640x zur entsprechenden Vorläufertechnik des Kombinat Robotron, hier schwerpunktmäßig zu den Kleinrechnersystemen 4200/4201 und zum Kleinrechner C 8205. Prinzipiell bestehen für die Lösung des genannten Problems die Möglichkeiten der Emulation oder Simulation. Nach sorgfältigen Analysen und auf Grund vorliegender Erfahrungen wurde entschieden, die Parallelfähigkeit zum KRS 4200/4201 mit einem Emulator herzustellen, die zum Kleinrechner C 8205 mittels Simulator. Damit wird die Ablösung dieser Rechner durch Basisrechner A 640x für den Anwender wesentlich erleichtert. Zu beachten ist, daß die Anwendung eines Emulators oder eines Simulators in der Regel keinen Anwendungsfortschritt bringt, da kein arbeitsplatzorientierter Einsatz erreicht wird. Fortschritte in dieser Richtung sind erst nach umfangreichen anwendungstechnischen Vorleistungen möglich.

Literatur

- 11j schule, W.: Grundkonzeption des Erzeugnisprogramms „Dezentrale Dateitechnik“. *rechentechnik/datenverarbeitung* 17 (1980) 9, S. 7
- 12j Autorenkollektiv: Das Erzeugnisprogramm der dezentralen Dateitechnik des VEB Kombinat Robotron. *Neue Technik im Büro* 25 (1981) 2, S. 3s
- 13. Entwicklungstendenzen auf dem Mikrorechnermarkt. ILK (1982) 5-1, VEB Robotron ZFT
- 14j Hartge, L.: *Distributed Data Processing. systems* 911981, S. 57
- 15j Schoppa, W.: Möglichkeiten und Grenzen der dezentralen Datenverarbeitung. *rechentechnik/datenverarbeitung* 15 (1979), Beiheft 4, S. 44

Der Beitrag umfaßt Ausführungen zu den Geräten (Komplexgeräten, Geräten und Baugruppen), aus denen Grundkonfigurationen von Basisrechnersystemen zusammengestellt werden können. Er wendet sich an Projektarten von Informationsverarbeitungssystemen mit Basisrechner des Kombinat Robotron und vermittelt ihnen sowohl für die technische als auch für die anwendungstechnische Projektierung wichtige Informationen.

1. Vorbemerkung

Der Einsatz von Mikroprozessoren in elektronischen Datenverarbeitungsanlagen eröffnet dem Anwender eine wesentlich breitere Palette von Anwendungsmöglichkeiten.

Durch die Vorteile und die neuen Eigenschaften von mikroprozessororientierten Rechnersystemen werden weitere Einsatzgebiete erschlossen. Dabei kommt den charakteristischen Merkmalen

- geringes Volumen
- günstiges Preis-Leistungsverhältnis
- minimierter Energiebedarf
- hohe Zuverlässigkeit

besondere Bedeutung zu.

Die Entwicklung der Mikrorechnerfamilie Robotron K 1600 zielt darauf ab, eine dem Weltstand entsprechende Technik bereitzustellen.

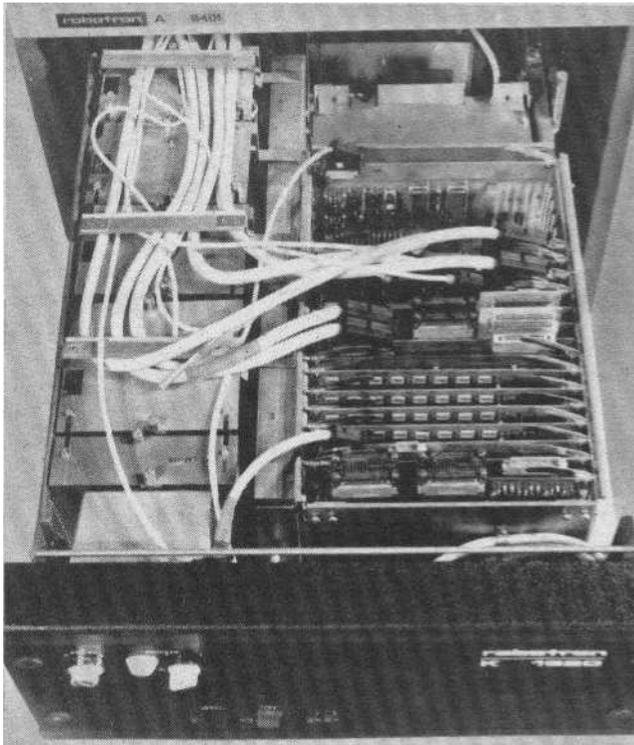
Mittels flexibler Anschlußbedingungen lassen sich die Systeme der Mikrorechnerfamilie Robotron K 1600 optimal an den jeweiligen Anwendungsfall anpassen.

Die Tatsache, daß die Rechnersysteme Zweiadreitrechner sind, bietet auf dem Gebiet der anwendungstechnischen Projektierung bessere Möglichkeiten, als das für herkömmliche Rechner systeme der Fall war.

2. Zentraleinheiten

2.1. Mikrorechner Robotron K 1620

Der Mikrorechner Robotron K 1620 ist ein Modell der mittleren Leistungsklasse, dessen Kernstück die Zentrale Verarbeitungseinheit (ZVIO K 2662) ist.



Mikrorechner K 1620 als Einschub im Basisrechner A 6401 **Werkfoto**

Er kann mit anschlusskompatiblen Speichermoduln bis maximal 28 K Worte (56 K Byte) ausgerüstet werden.

Innerhalb der ZVE K 2662 sind neben zwei LSI-Schaltkreisen mit je 8 Bit Verarbeitungsbreite die Mikroprogrammsteuerung, die Bussteuerung, der Taktgenerator und die Echtzeituhr enthalten. Folgende Funktionen werden von der ZVE K 2662 realisiert:

- Mikroprogrammierte Befehlsliste K 1620
- mikroprogrammgestützte Realisierung von Bedienfunktionen über Bedienperipherie
- Steuerung des Informationsaustausches der ZVE über den Systembus
- der Informationsaustausch erfolgt bezüglich der Adressen synchron und der der Daten asynchron
- Buszuteilungs- und Unterbrechungssteuerung
- zentrale Regenerierungssteuerung der dynamischen Halbleiterspeicher.

Der Mikrorechner robotron K 1620 ist konstruktiv als Einschub für normierte Gefäße nach 5T RGW 834-77 ausgeführt; der Wotul besteht aus drei Steckeinheiten. ZVE, Speicher, Anschlußsteuerungen und Stromversorgung sind gemeinsam im MR-Einschub untergebracht.

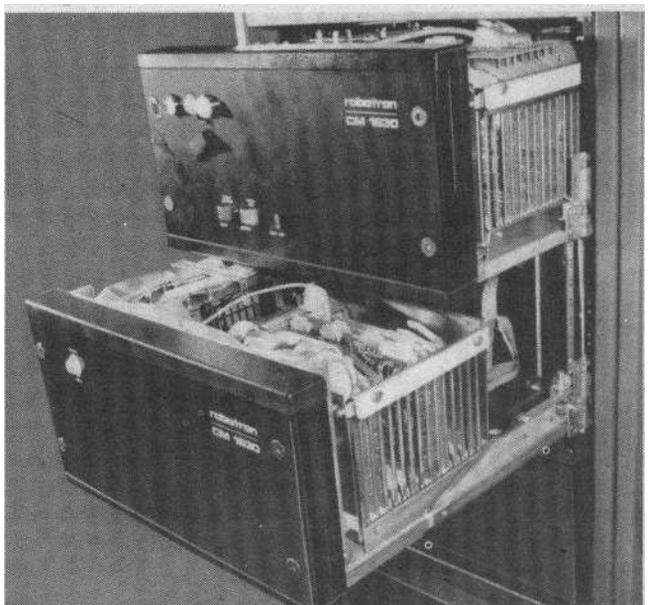
Leistungsparameter des	Mikrorechners K 1620
Verarbeitungsbreite	16 Bit parallel, Wort- und Byteverarbeitung
Zahlendarstellung	Festkomma, Zweierkomplement
Anzahl der Befehle	ca. 400 (einschließlich Modifikationen)
Befehlssystem	SKR-Basisbefehlsliste
Steuerungsprinzip	Mikroprogrammsteuerung, horizontal Mikroprogrammierung, asynchron
Adressierungsarten	12
Anzahl der universellen Register	8
Adressierungsbreite	32 K Worte (davon 4 K Worte für E/A-Geräte-register)

Unterbrechungssystem	vektororganisierte Unterbrechungsbehandlung:
Pegel	5 Unterbrechungsebenen, davon 1 DMA-Eberic
Datenerhalt bei Netzausfall	TTL-Pegel
Automatischer Restart	vorhanden
Echtzeituhr	vorhanden
E/A-Organisation	20 ms Zeitintervall, programmunterstützt
Stackorganisation	programmiert und DMA
Bedienung	vorhanden
	über Bediengerät, Frontpaneel
Befehlsdurchführungszeit	K 1620 [µPLs]
• Zweioperandbefehle	
Register - Register	3,05 - 4,55
Register - Speicher	4,50 - 9,60
Speicher - Register	6,00 - 9,65
• Einoperandbefehle	
Register	3,05 - 5,311
Speicher	5,20 - 10,35
PSW-Transporte	3,05 - 5,25
Branch-Befehle	3,80 - 6,05
Spinngbefehle	3,80 - 8,20
Flaggehandlung	3,80
Trap-Befehle	17,35

2.2. Mikrorechner robotron K 1630

Der Mikrorechner robotron K 1630 ist das leistungsfähigste Modell der Mikrorechnerfamilie robotron K 1600; sein Kernstück ist die Zentrale Verarbeitungseinheit (ZVE) K 2663. Sie wurde als Basis für einen Kleinrechner entworfen, der als leistungsfähiger Kern in Echtzeitsystemen, in Datenverarbeitungssystemen für ökonomische und wissenschaftlich-technische Einsatzfälle sowie für Multinutzer- und Multiprogrammanwendungen eingesetzt werden kann. Der Mikrorechner robotron K 1630 kann mittels der Speicherverwaltungseinheit (SVE) K 2061 mit anschlusskompatiblen Speichermoduln bis höchstens 124 K Worte (248 K Byte) ausgerüstet werden.

Bei Einsatz eines Adhptetikprozessors (ARP) K 2062 als Zusatzrichtung für Festkomma- und Gleitkommaoperationen mit größeren Genauigkeiten kann eine wesentliche Leistungssteigerung des Mikrorechners robotron K 1630 realisiert werden. Darüber hinaus wird durch eine umfangreichere Befehlsliste



Mikrorechner K 1636 mit Erweiterungseinschub als Bestandteil des Basisrechners A 6401 **Foto: Drowski**

und eine erweiterte Paranelarbeit eine höhere Verarbeitungsgeschwindigkeit erreicht.

Der Mikrorechner K 1 630 realisiert folgende Funktionen:

- Zentrale Verarbeitungseinheit
Mikroprogrammierte Befehlsliste K 1 620TK 1630
mikroprogrammgestützte Realisierung von Bedienfunktionen
Adressrechnung für den Arithmetikprozessor
- Speicherwaltungseinheit
 - Umrechnung der virtuellen 16•Bit-Adresse in eine physische 18-Bit-Adresse
 - Gewährleistung eines Speicherschutres
 - Steuerung des Informationsaustausches zwischen Prozessor und Systembus.

Der Mikrorechner robotron K 1630 ist konstruktiv als Einschub für normierte Gefäße nach ST RGW 834-77 ausgeführt; der Modul SVE/ZVE besteht aus 9 Steckeinheiten. ZVE, Speicher, Anschlußsteuerungen und Stromversorgung sind gemeinsam im Mikrorechner-Einschub untergebracht.

2-2.3. Arithmetikprozessor K 2062

Der Arithmetikprozessor K 2062 gehört als Einheit zum Mikrorechner robotron K 1630 und ist ohne die ZVE K 2663 nicht funktionsfähig.

Es ist auf der Basis eines LSI-Arithmetik-Schaltkreises aufgebaut, enthält außerdem eine Reihe Register, ein Steuerwerk auf TTL-Basis und ist mikroprogrammgesteuert. Der ARP ist (über den prozessorinternen Bus und über Direktleitungen mit der ZVE verbunden; er besteht aus 3 Steckeinheiten. Der Informationsaustausch zwischen ARP und ZVE sowie innerhalb des ARP erfolgt asynchron.

Der ARP übernimmt im wesentlichen folgende Funktionen:

- Multiplikation und Division im Festkommaformat (Einfach- und Doppelwort)
- Addition und Subtraktion von Doppelwortoperanden
- Gleitkomma-Arithmetik für Grundrechenarten (ADD, SUB, MUL, DIV)
- Datenkonvertierung von Festkommaformat in Gleitkommaformat und umgekehrt.

Leistungsparameter des Mikrorechners robotron K 1630

Verarbeitungsbreite	16 Bit parallel, Wort- und Byteverarbeitung
Zahlendarstellung	Festkomma, Zweierkomplement
Anzahl der Befehle	ca. 400 (einschließlich Modifikation)
Befehlssystem	SKR-Basisbefehlsliste
Steuerungsprinzip	Mikroprogrammsteuerung, horizontal Mikroprogrammierung, asynchron
Adressierungsarten	12
Anzahl der universellen Register	8+8
Adressierungsbereich	128 K Worte (davon 4 K Worte E(A-Geräteregister)
Unterbrechungssystem	vektororganisierte Unterbrechungsbehandlung, 5 Unterbrechungsebene, davon 1 DMA-Ebene
Pegel	TTL-Pegel
Energieausfallsteuerung	vorhanden
Automatischer Restart	vorhanden
EtA-Organisation	programmiert und DMA
Stackorganisation	vorhanden
Bedienung	über Bediengerät, Frontpanel

Befehlsausführungszeiten K 1630 L]:

- Zweioperandbefehle
 - Register - Register 3,50- 4,25
 - Register - Speicher 4,25- 8,75
 - Speicher - Register 4,00- 9,55
- Einoperandbefehle
 - Register 2,75- 9,75
 - Speicher 3,50-14,75
- PSW-Transporte 3,00- 8,25
- Branch-Befehle 2,70- 3,45
- spmgbefehle 3,80- 9,65
- Flagbehandlung 5,30
- Trap-Befehle 7,50

Rechenzeiten mit ARP (gis]:

	Multiplikation	Division
Festkomma (16) 15 Bit und Vorzeichen	ca. 16	ca. 55
Festkomma (32) 31 Bit und Vorzeichen	ca. 22	ca. 95
Exponent, 23 Bit Mantisse und Vorzeichen	ca. 22	ca. 55
Gleitkomma (64) 8 Bit Exponent, 35 Bit Mantisse und Vorzeichen	ca.55	ca.105

2.3. Anschlusssystem (Interface)

Die allgemeine Systemarchitektur der Mikrorechnersysteme K 1620 und K 1630 sieht vor, daß alle Systemkomponenten wie Zentrale Verarbeitungseinheit, Speichermoduln, Anschlußsteuerungen und Koppeleinheiten über eine elektrisch und konstruktiv standardisierte Verbindung --- den Systembus K 1600 --- zusammengeschlossen werden. Über den Systembus werden Adressen, Daten und Signale zum Teil auch zeitmultiplex über gleiche Leitungen übertragen sowie die Einspeisung für die Stromversorgung der Moduln vorgenommen.

Die Form der Kommunikation über den Systembus ist für alle an den Systembus angeschlossenen Geräte gleich und geschieht nach dem Master-Slave-Prinzip sowohl als programmierter Verkehr der Zentralen Verarbeitungseinheit als auch im direkten Speicherzugriff.

Mittels besonderer Moduln kann der Systembus verlängert werden bzw. ist der systembus abzuschließen.

Die peripheren Geräte werden über verschiedenartige Anschlußsteuerungen, die das jeweilige Interface des peripheren Gerätes bereitstellt, an den Systembus angeschlossen.

Durch einen Busumsetzer kann das Anschlußbild des SKR-Einheitsbus realisiert und somit der Anschluß von Baugruppen und Geräten des Systems der Kleinrechenstechnik (SKR) vorgenommen werden.

Zum Anschluß eines zweiten Blockeinschubes mit oder ohne Verstärkung läßt sich der Systembus erweitern.

Als Standardinterface sind definiert:

- Interface für sternförmigen Anschluß mit Paralleliübertragung (IFSP)
- Interface für sternförmigen Anschluß mit Serienübertragung (IFSS)
- Standardinterface 1000 (SIF 1000) für den Anschluß von SIF-1000-Peripherie
- Interface 52 nach TGL 29077 bzw. entsprechend Empfehlung

lung V.24 und V.28 nach CCITT (für Anschluß von Terminals und Datenfernverarbeitungsperipherie)

— Gerätespezifische Interfaces

- Anschlußsteuerung für Kassettenplattenspeicher
- Anschlußsteuerung für Folienspeichereinheit.

2.4. Interne Speicher

Die internen Speicher der Mikrorechner K 1600 sind modular aufgebaut und ermöglichen eine günstige Konfigurierung entsprechend den Anforderungen. Dabei werden durch Verwendung von RAM- und PROM-Schaltkreisen die Vorteile moderner Halbleiterspeicher genutzt.

Die RAM-Schaltkreise werden für dynamische bzw. statische Operativspeicher-Moduln (Schreib-Lesespeicher) und die PROM-Schaltkreise für programmierbare Festwertspeicher (Lesespeicher) genutzt.

Der Anwender kann sich über den Einsatz verschieden großer Steckeinheiten (Speichermoduln) seine Speicherkapazität wahlweise im Rahmen des definierten Maximum der Mikrorechner selbst konfigurieren. Als Arbeitsspeicher werden in der Regel dynamische Operativspeichermoduln mit und ohne Fehlerkorrektureinrichtung genutzt.

Von der Fehlerkorrektureinrichtung werden Einbitfehler der gelesenen Information korrigiert und Doppelbitfehler erkannt und gemeldet. Ohne Fehlerkorrektureinrichtung wird eine byteweise Paritätskontrolle ausgeführt, und Paritätsfehler werden gemeldet.

Zur Gewährleistung der Stromversorgung der Operativspeicher bei Netzausfall stehen Akkumulatormoduln zur Verfügung.

Die Stützzeit zur Informationserhaltung liegt im Stundenbereich.

2.5. Stromversorgung

Die Mikrorechner der Familie robotron K 1600 verwenden eine modulare Stromversorgung. Diese ist in ihren Leistungsdaten den elektronischen Einheiten angepaßt und wird mit der Elektronik aufgerüstet.

Die Systemstromversorgung umfaßt alle Funktionseinheiten, die das Ein- und Ausschalten des Systems, der Schränke, der Einschübe bzw. Einsätze und der Einzelgeräte am Versorgungsnetz gestatten. Sie gilt für alle Systembestandteile des Systems robotron K 1600.

Sei der Realisierung der Systemstromversorgung wurde die „Analyse des Systems der Stromversorgung und Erdung der Technischen Mittel des SKR“ (Fassung von 1979) berücksichtigt.

Der zentrale Ein- und Ausschalter für das gesamte Rechnersystem befindet sich an der Frontplatte des Rechner-Einsatzes (Schlüsselschalter). Mit der Einschaltung werden alle Schrankseinheiten (max. 16 Stück) mit Netzspannung versorgt. Die Einzelgeräte und Einschübe können unabhängig voneinander ein- und ausgeschaltet werden.

Versorgungsnetz:

Einphasenwechselspannung

$$220\text{V} \begin{matrix} +10\% \\ -15\% \end{matrix}$$

50 Hz oder 60 Hz

je nach Einsatzfall.

Die Systemstromversorgung besteht aus den folgenden Funktionseinheiten.

- Im Rechner-Schrank:
Hauptschalter, Sicherungsblock, Netzfilter (12 A), System-Ein-/Aus, Schrank-Ein-/Aus, (mit Anzeige und Schutz), Schrankverteiler (mit 2 Schuko-Steckdosen), Folgeschaltung
- im Schrank ohne Rechner:
Hauptschalter, Sicherungsblock, Netzfilter (12 A), Schrank-Ein-/Aus (mit Anzeige und Schutz), Schrankverteiler (mit 2 Schuko-Steckdosen), Folgeschaltung
- im Beistellschrank (ohne Systemstromversorgungseinsatz):
Hauptschalter, Sicherungsblock, Netzfilter (12 A), Schutz mit Ein-/Aus-Schalter, Schrankverteiler (mit 2 Schuko-Steckdosen)
- im Einschub/Gerät:
Ein-/Aus-Schalter mit Anzeige, Netzfilter (4 A) oder Entstörglied, Schlüsselschalter (nur im Rechner-Einschub).
- im Systemverteiler:
Systemschutz (Schaltvermögen je nach Systembelastung) Stromversorgungsmoduln und -zusätze
Stromversorgungsmoduln (STM)
sie sind geregelte Sperr- (bei 25 W Ausgangsleistung) oder Flußwandler (bei Ausgangsleistung = 50 W), die als Betriebsspannung die gleichgerichtete Netzspannung (ca. 300 V) benutzen.
Stromversorgungszusatz (STZ)
Ein Stromversorgungszusatz erzeugt bis zu 4 Ausgangsspannungen bei 25 W Einzelleistung und einer Summenleistung von 75 W. Ein Stromversorgungszusatz kann nur gemeinsam mit einem Stromversorgungsmodul betrieben werden.

Technische Daten

Netzspannung 220 V $\begin{matrix} +10 \\ -15\% \end{matrix}$; 47...63 Hz
Einphasenanschluß

Schutzgüte
Schuhklasse I nach TGL 21366 und Standard STP
Z 50.094.100

- Funkentstörung
Funkentstörgrad F1 nach TGL 20885/05, entspricht spannungswerten A nach MM SKR 002-76, Pkt. 2, bei Anwendung eines Netzfilters F3 nach TGL 20885/05 durch das übergeordnete Gefäß realisiert.
Umgebungsbedingungen
Die Baugruppen sind für den Einsatz in Erzeugnissen mit EKL 3, TKL 3 und LKL 3 nach TGL 26465 ausgelegt.

Stromversorgungsmodul

- Ausgangsspannungen
50 W,
100 W,
150 W bei jeweils 5V, 7V, 9V, 12V, 15V, 24V, 36V.
- Stützzeit bei Netzausfall
10 ms bei Minustoleranz der Netzspannung
20 ms bei Nennwert der Netzspannung
- Wirkungsgrad
Für vorgenannte Stromversorgungsmoduln beträgt der Wirkungsgrad in Abhängigkeit von der Ausgangsspannung etwa 70-80 %.

Stromversorgungszusatz

- Wahlweise ein bis vier Ausgangsspannungen
- Ausgangsleistung je Spannung 25 W, die Gesamtbelastung des STZ beträgt max. 75 W
Zum Betrieb eines STZ ist ein 50 W-, 100 W- oder 150 W-Modul erforderlich.
Ausgangsspannungen (positiv oder negativ)
5V, 7V, 9V, 12V, 15V, 24V, 60V.

Notstromversorgung

Für den dynamischen Operativspeicher ist bei Netzausfall eine Notstromversorgung für den Datenerhalt möglich. Dazu wird der Modul für Datenerhalt verwendet, der auch den Akkumulator enthält. Eine Überladung und Tiefentladung des Akkumulators wird verhindert.

Parameter des Akkumoduls:

- Baugruppe, die von einem STM 12 V/100 W gespeist wird
- Leistungsaufnahme etwa 60 W
- Ausgangsspannungen
 - 5 NG = - 5 V/25 mA
 - 12 PG = + 12 V/1000 mA (bei Akkubetrieb)
 - + 12 V/1800 mA (bei Netzbetrieb)
 - 5PG = + 5V/1200 mA.

2.6. Konstruktiver Aufbau, Gefäßsystem und Baugruppen

Die modulare Konzeption der Mikrorechnerfamilie robotron K 1600 findet auch in der konstruktiven Lösung des Rechneraufbaus ihren Ausdruck. Die logischen Funktionseinheiten (Moduln), die als Steckeinheiten ausgeführt sind und über den Systembus verbunden werden, werden in Blockeinschüben konstruktiv zusammengefaßt.

Der Blockeinschub ist für einen schrankeinsbau ausgelegt bzw. kann auch in verkleideter Form als Auftischgerät (z. B. Laborrechner) eingesetzt werden.

Die Schränke sind nach ST RGW 834-77 genormt.

Die Abmessungen betragen:

Breite	600 mm
Höhe	730 mm oder 1800 mm
Tiefe	800 mm.

Die konstruktiven Hauptbaugruppen des Blockeinschubes der Mikrorechnerfamilie robotron K 1600 sind

- Grundrahmen
- Steckeinheiten (STE)
- Steckeinheiteneinsätze
- Lüfterbaustein
- Stromversorgungsmoduln (STV-M)
- Frontplatte (FP).

Sie haben folgende Funktionen;

- Grundrahmen
Der Grundrahmen ist das tragende Element des Blockeinschubes und dient zur Aufnahme der übrigen Hauptbaugruppen.
- Steckeinheiten

Die logischen Funktionseinheiten der Mikrorechner sind auf STE der Größe 215 mm x 170 mm untergebracht. Die STE besitzen Steckverbinder zum Anschluß an die Rückverdrahtungsleiterplatte (RLP) und vorderseitig Steckverbinder zum Anschluß von Interfacekabeln.

Steckeinheiteneinsätze

Die Steckeinheiten werden in einem Steckeinheiteneinsatz vertikal angeordnet. Es werden, entsprechend den Blockeinschüben, Steckeinheiteneinsätze in zwei Längen realisiert. Sie können 28 STE (bei Verwendung eines Akkumoduls 21 STE) bzw. 14 STE aufnehmen.

Die Höhe eines Blockeinschubes beträgt 6U (U = 44.45 mm) Busverbindung

Die Verbindung der STE untereinander ist über eine gedruckte Rückverdrahtungsleiterplatte ausgeführt

- Lüfterbaustein

Die Anordnung der STE in den Steckeinheiteneinsätzen

und der hohe Integrationsgrad auf den STE erfordern eine horizontale Zwangsbelüftung. Zu diesem Zweck werden hochleistungsfähige Lüfter kleiner Abmessungen eingesetzt. Für Steckeinheiteneinsätze mit 28 STE sind vier Lüfter vorgesehen, für Steckeinheiteneinsätze mit 14 STE sind zwei Lüfter notwendig.

Stromversorgungsm od uln

Die Stromversorgungsmoduln sind in Kompaktbauweise ausgeführt und werden über eine Aufnahme auf dem Grundrahmen aufgesetzt. Sie sind steckbar in der Aufnahme angeordnet. Die STV-M werden ebenfalls horizontal belüftet. Frontplatte

Die Frontplatte stellt die vordere Abdeckung des Blockeinschubes dar. Sie ist abnehmbar gestaltet und ermöglicht das Aufschwenken der dahinter befindlichen Montageplatte, welche die benötigten Bedienelemente trägt.

Die wichtigsten Bedienelemente beim Prozessoreinschub sind Schlüsseltaster, Netzschalter mit Anzeige und zwei Taster.

Die Verdrahtung der Bedienelemente erfolgt vorrangig über eine gedruckte Leiterplatte.

3. Periphere Geräte

Die an die Mikrorechner robotron K 1620 bzw. K 1630 über den Systembus K 1600 oder SKR-Bus an interfacespezifische Anschlußsteuerungen anschließbaren peripheren Geräte besitzen eine Reihe von charakteristischen Merkmalen:

- funktionelle und konstruktive Abgeschlossenheit
- autonome Stromversorgung
- Ausführung als Einschub/Einsatz (soweit funktionell möglich) oder als abgesetztes Gerät (Stand-/Auftischgerät).

3.1. Bedieneinheit (BDE) K 8911 Bildschirmterminal (BTL) K 8912

Kurzcharakteristik

- BDE und BTL sind einfache alphanumerische Bildschirmgeräte.
Zum Zweck des Kopierens der Bildschirminformationen kann die BDE mit einem SD robotron 1152 ausgerüstet werden.
Die Tastatur ist eine über kabel anschließbare separate Baugruppe.
- Die BDE 1600 ist ein fest programmiertes Gerät, dessen Steuerprogramm (auf PROM) vom Anwender nicht verändert werden darf.

Technische Daten

Monitor

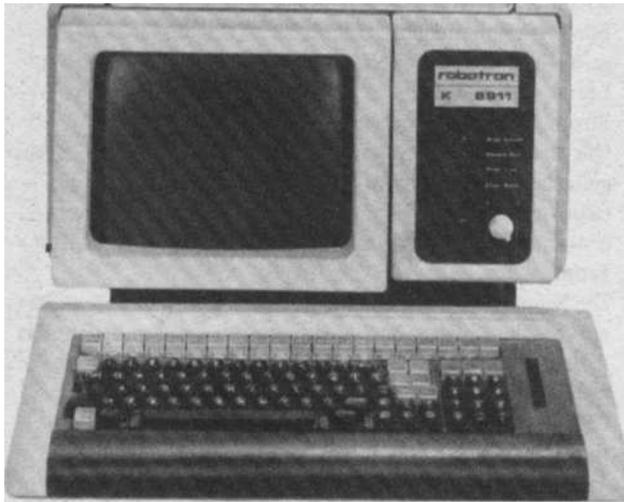
- 31-cm-Bildschirmdiagonale
- 2 Zeichenhelligkeiten
- 24 Zeilen zu 80 Zeichen = 1920 Zeichen
- Zeichenraster (Punkte) 5 x 8

Zeichenvorrat

- 128 Zeichen gesamt, davon 12 Attributzeichen, 96 darstellbare Zeichen, 20 Steuerzeichen

Druckeranschlufi

- SD robotron 1152
- SD robotron 1157



Bedieneinheit K 8911

Werkfoto

Индикация ngsb ed [ngu nyen

– Logikhaugruppen werden für den Einsatz in Geräten mit EKL 3 entwickelt

– Betriebsdauer 3-5schicht-Betrieb

Автоматизация

Abmessungen Bildschirm mit Gehäuse

(H x B x I) 340 x 510 x 410 mm

Abmessungen Tastatur (B x T) 512 x 250 mm, bis 1,5 m frei aufstellbar

Entfernung zur Systemsteuereinheit max. 500 m

Gewicht (ohne Drucker) 25 kg

Строительные данные

Spannung 220V +10%, -15%

Frequenz 47...63 Hz

Leistungsaufnahme 150 VA

Netzanschluss Tiber Schukostecker

Соединения

Typ des interface IFSS

Typ der Anschlusssteuerung (AS) AI

Busanschluss Systembus K 1600

Übertragungsgeschwindigkeit max. 9600 Baud

Übertragungsalgorithmus Echobetrieb

Примечание

Folgende Ergänzungen und Änderungen sind beim BTL K 8912 gegenüber der BDE K 8911 zu berücksichtigen:



Bildschirmterminal K 8912

Werkfoto

1. Monitor

Es sind wahlweise zwei verschiedene Monitore einsetzbar:

- Monitor K 7222.11 (24 Zeilen/64 Zeichen) mit AS 7024.30
- Monitor K 7221.10 (16 Zeilen/64 Zeichen) mit AS K 7023.01

2. Interface

Je nach Einsatzfall des Terminals

- Nahbereich bis 500 m IFSS
- Fernbereich V.24 und Modem

3. Übertragungsalgorithmus

- Blockbetrieb (AP 62/64).

3.2. Lochbandeinheit (LBE) K 6T{}0

Die LBE K 6200 ist ein Gerätekomplex für die Datenträgerein- und -ausgabe auf der Basis des Datenträgers Lochband. Der Lochbandstanzer verfügt über einen passiven Abspuler.

Technische Daten

Лейзенгпараметры

Lochbandstanzer;

- 5-nachgeschaltete und getriggerte Kanäle
- 50 Zeichen/s
- Paritäts-, Bandend- und Bandrißkontrolle

Lochbandleser:

- Lesegeschwindigkeit 300 Zeichen/s im Durchlaufbetrieb
- 100 Zeichen/s im start-stop-Betrieb

Условия эксплуатации

- Temperatur +10°C... 35°C
- Luftfeuchte 80% bei +25°C
- Geräuschpegel max. 75 dB

Автоматизация

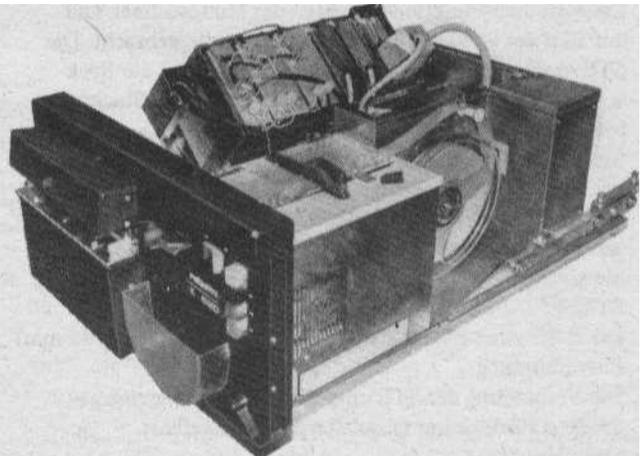
- Einschiebe 19 1011(6 U
- Gewicht je Einschub 35 kg

Строительные данные

- Spannung 220 V +10%, -15%
- Frequenz 50/60 Hz ± 1 Hz
- Leistungsaufnahme 400 VA
- Anschluss max. 20 m

Соединения

- Typ des Interface IFSP
- Typ der AS AI
- Systembus K 1600



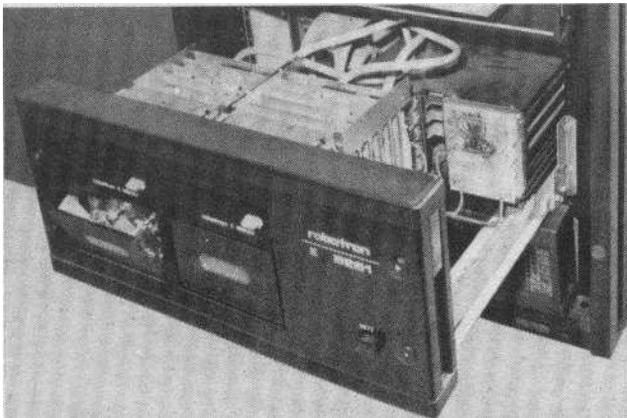
Lochbandeinheit K 6200

Foto: Drowski

3.3. Kassettenmagnetbandeinheit (KMBE) K 5261

Die KMBE robotron K 5261 ist ein Aufzeichnungs- und Wiedergabegerät für 1/8-Zoll-Digitalkassetten und dient vorrangig der Datentragein- und -ausgabe. Die KMBE robotron K 5261 besteht aus zwei KMB-Geräten robotron K 5200.

Die Gerätetechnik ist Dateri- und Aufzeichnungskompatibel zur entsprechenden Datenerfassungstechnik des VEB Kombinat Robotron



Kassettenmagnetbandeinheit K 5261

Foto: Drowski

Technische Daten

Leistungsparameter

Bandgeschwindigkeit	38 cm/s
Start/Stop-Lücke	Nominal 20,3 mm
Spurenanzahl	2 (je 5te Spur)
Prüfmöglichkeiten der information	Read-After-Write
Gerätefunktionen	Herstellen Grundzustand Aufzeichnen/Wiedergabe, Daten- und Steuerblöcke Blockweise Vor-/Rücksetzen Vor-/Rücksetzen über Steuerblock Suchlauf Steuerblock Umspulen Duplizieren

Umgebungsbedingungen

Temperatur	+5 °C bis +40 °C
Luftfeuchte bei 30	95 %
Einsatzklasse	3

Aufstellungsbedingungen

Einschube	19 Zoll/5 U
Gewicht je Einschub	28 kg

Stromversorgung

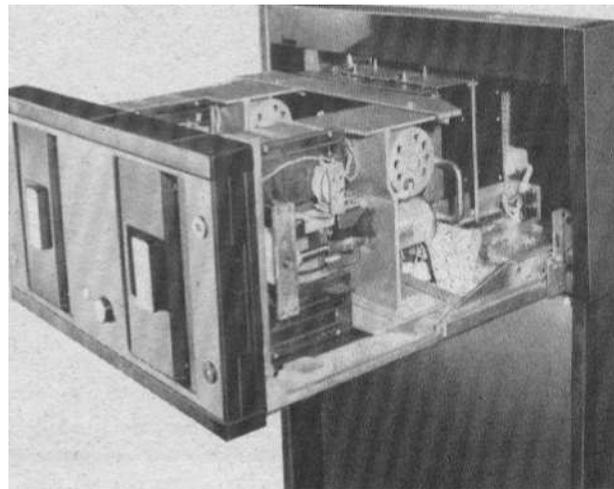
Spannung	220 V + H) %, -15 %
Frequenz	50/60 Hz ± 1 Hz
Enlstörkombination	am Netzzeingangs
Leistungsaufnahme	150 VA

Systemanschlüsse/bedingungen

Typ des Interface	IFS
Typ der AS	MS
Bus-Anschluß	Systembus K 1600
Übertragungsraten	9600 oder 38400 Baud

3.4. Folienspeichereinheit (FSE) K 5665

Die FSE K 5665 ist ein Massendatenspeicher mit wechselbarer, flexibler Magnetplatte (Diskette) und einer Kontaktbefestigung. Sie ist ein Schreib- und Lesespeicher.



Folienspeichereinheit K 5665

Foto: Drowski

Technische Daten

Leistungsparameter

Speicherkapazität je Diskette	256 K Byte zur Informationspeicherung nutzbar
Aufzeichnungsdichte	Äußere Spur 466 Bit/mm Innere Spur 830 Bit/mm
Drehzahl	360 U/min
Übertragungsgeschwindigkeit	250 K Bit/s f 0,1 % beim Schreiben 250 K Bit/s f 7 % beim Lesen
Spurenanzahl je Diskette	77
Sektorenanzahl	max. 32
Mittlere Zugriffszeit	
- je Spur	85 ms
- je Diskette	508 ms

Umgebungsbedingungen

Temperatur	+10 °C bis +40 °C
Luftfeuchte	40% bis 90%
Luftreinheit	max. 1 mg/m ³

Aufstellungsbedingungen

Finshöhe	19 Zoll/6 U
Gewicht je Laufwerk	8 kg

Stromversorgung

Spannung	220V + 10%, -15%
Frequenz	50 Hz ± 0,5 Hz
Leistungsaufnahme	65 VA je Laufwerk

Systemanschlüsse/bedingungen

Typ des Interface	IFS
Typ der AS	AFS
Bus-Anschluß	Systembus K 1600

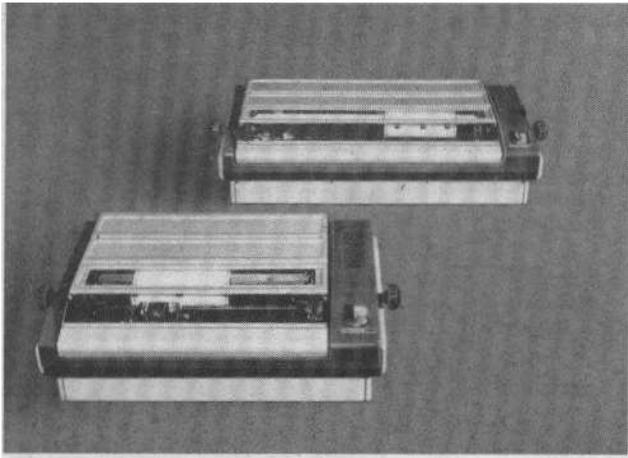
3.5. Seriendrucker (SD) 1152

Der SD robotron 1152 ist ein Drucker mit geschlossenem Schriftzug (Groß-/Kleinbuchstaben) auf Basis eines Typenrades mit einer durchschnittlichen Druckleistung von 30 Zeichen/s. Die Steuerung arbeitet auf Mikroprozessorbasis (U 880). Für unterschiedliche Einsatzfälle steht die entsprechende Formular- und Formularechnik zur Verfügung.

Varianten:

Typ 251 = 132 Zeichen/Zeile
Typ 252 = 210 Zeichen/Zeile, zweibahnige Formularführung möglich.

Die Formulareinrichtung an der Seriendruckergrundvariante ermöglicht die Ausgabe auf Einzelformulare und Journalrolle. Sie ist aufrüstbar mit einer Baugruppe für Leporellotransport



Seriendrucker SD 1152

Werkfoto

oder Kontokarteneinzug oder einer Vorsteckeinrichtung für Einzelbelege.

Papier:

Journalrollen

Rollendurchmesser **80 mm**

Hölsennennendurchmesser **10 mm**

max. Breite

— bei Typ 251 375 mm

— bei Typ 252 540 mm

Leporello (einlagig/mehrlagig)

max. Breite

— bei Typ 251 375 mm

— bei Typ 252 450 mm

Technische Daten

Leistungsparameter

Druckgeschwindigkeit 30 Zeichen/s bei Start-Stop-Betrieb

Lauf-/Tabulationsgeschwindigkeit 180 Zeichen/s

Tabulationsrichtung Vor- und Rückwärts

Zeichen/Zeile 132 bis 158 bei Typ 251

210 bei Typ 252

Zeichenumfang 96 Zeichen (Groß/Klein)

Zeichendarstellung Geschlossener Schriftzug

Schriftarten PICA und OCR in Vorbereitung

Druck I Original/5 Kopien

Zeichensichtbarkeit Mit Funktionstaste: Zeilenschaltung 1 Zeile vorwärts. Dadurch wird zuletzt gedrucktes Zeichen sichtbar. Nach Loslassen der Taste geht Papier in die Ausgangsstellung.

Lebensdauer der Typenscheibe 20×10^6 Anschläge/Type

Umgebungsbedingungen

Temperatur +5 bis +40°C

Temperaturgradient 5°C/h

Luftfeuchte max. 95% bei 30°C

Auflösungsbedingungen

Auftischgerät mit den Abmessungen

— Typ 251 798 x 190 x 660 mm

— Typ 252 996 x 196 x 660 mm

Masse 30 kg

Stromversorgung

Als Auftischgerät besitzt der Drucker eine integrierte Stromversorgung.

Netzspannung 220 V + 10%, -15%

Frequenz 50 Hz \pm 1 Hz

Leistungsaufnahme 250 VA

Schutzart Schutleiter

Netzanschluß Über Kaltgerätenanschlußleitung mit Schukostecker

Systemanschlußbedingungen

Typ des Interface IFSS

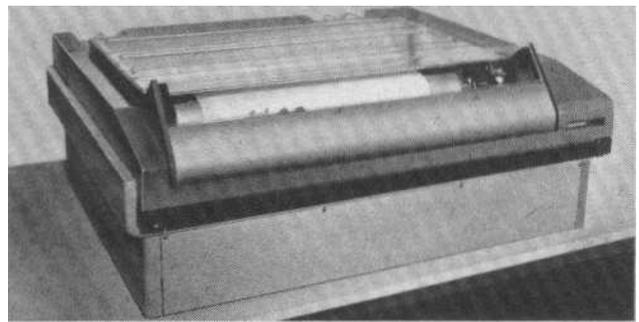
Typ der AS AIS

Übertragungsgeschwindigkeit 9600 Baud

Busanschluß Systembus K 1600

3.6. Seriendrucker (SD) 1157

Der SD robotron 1157 ist ein Auftischdrucker der mittleren Leistungsklasse. Er wird dort genutzt, wo weniger ausgabeintensive Projekte abgearbeitet werden bzw. dort, wo die Basisrechnerkonfiguration schon vom Leistungsvermögen her den Einsatz eines Paralleldruckers nicht rechtfertigt. Der SD robotron 1157 arbeitet nach dem Nadeldruckprinzip. Die Ausgabe von Groß- und Klembuchstaben auf unterschiedliche Formularebreiten ist möglich.



Seriendrucker SD 1157

Foto: Boden

Technische Daten

Leistungsparameter

Ausführung 1157-264 1157-267

Druckgeschwindigkeit 180 Zeichen/s 180 Zeichen/s

„Zeichenumfang“ 96 Zeichen 96 Zeichen

Zeichen je Zeile 132 210

Drucknutzen I Original/5 Kopien

Druckmatrix 9x7 9x7

Zeilenspeicher max. 2 Zeilen

Druckwegoptimierung vorhanden

Formulartechnik Leporello (ein-/zweibahnig),

Endlosrolle, Einzelbeleg

Papieranschub Vorwärts/Rückwärts mit programmierbarer

Positionierung

Formatsteuerung durch Schalter (4"; 6";

8"; 12") oder frei programmierbar

bei zweibahniger Ausstattung unabhängige

Vertikalsteuerung beider Bahnen

Papieranschubleistung

— Einzeiliger Anschub 70 m/s

— Vertikalabtabulation 30 Zeilen/s

Umgebungsbedingungen

Einsatzklasse 3 nach TGL 26 465

Temperatur +5 bis +40°C

Luftfeuchtigkeit max. 80% bei 30°C

Auflösungsbedingungen

Auftischgerät

Abmessungen (B x H x T)

1157-264 724 x 297 x 600 mm

1157-267 922 x 297 x 600 mm

Masse 55 kg—65 kg

<i>Stromversorgung</i>	
Netzspannung	220 V + 10 %, -15 %
Frequenz	47 bis 63 Hz
Leistungsaufnahme	230 VA
<i>Systemanschlußbedingungen</i>	
Typ des Interface	IF55/IFSP
Typ der AS	AISJAIP
Bus-Anschluß	Systembus K 1600

3.7. Paralleldrucker CM 6313 (VT 27065)

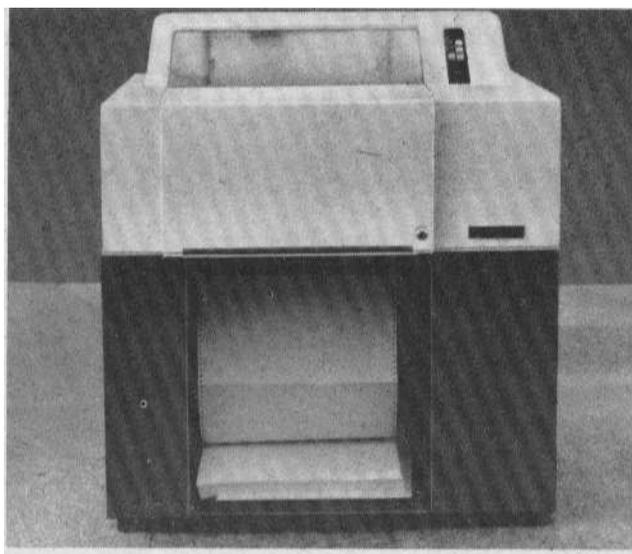
Der Paralleldrucker (PD) VT 27065 ist ein mittelschneller Paralleldrucker zum Anschluß an Klein- und Mikrorechner mit folgenden Merkmalen:

Elektrischer Teil

- Große Leiterplatte mit gesamter Steuerungselektronik auf TTL-Basis
- Puffer für 136 Zeilen
- Auswechselbare Interfacekarte
- Interfacekabel direkt auf Interfacekarte steckbar
- 136 Hammertreiber direkt hinter der Hammerbank angeordnet

Mechanik

- Drucktrommel mit 136 Hämern
- Papiervorschub für Faltpapier mittels Sternräder
- Übersprung der Papierfalte wird durch Nockenscheibe gesteuert
- Automatische Korrektur von Seitenverschiebungen des Drucktuches
- Formularbehälter und Ablagefach abgedeckt
- Vertikal-Formatsteuerung mit 8-Kanal-Lochband als Variante



Paralleldrucker VT 27065

Foto: Boden

Technische Daten

<i>Leistungsparameter</i>	
Druckgeschwindigkeit	660 Zeilen/min
Zeichenvorrat	96 (lateinisch/kyrillisch)
Zeichen/Zeile	136
Zeilenabstand	1/8 Zoll oder 1/8 Zoll
Drucknutzen	1 Original/5 Durchschläge

<i>Papieranforderungen</i>	
— Leporellopapier	100 bis 430 mm Breite 12 Zoll Höhe, durch Vertikalformatsteuerung über LB beliebig max. sechslagig

<i>Umgebungsbedingungen</i>	
Temperatur	+5 bis +40°C
Luftfeuchte	max. 90% bei 30°C
Geräuschpegel	max. 75 dB

<i>Aufstellungsbedingungen</i>	
Standgerät	
Abmessungen (BxTxH)	1000 x 800 x 1050 mm
Masse	200 kg

<i>Stromversorgung</i>	
Spannung	220V +10%, -15%
Frequenz	50Hz ± 1Hz
Leistungsaufnahme	max. 1000 VA
Netzanschluß	Zweipolig über Schukostecker mit fest montiertem Netzkabel am Genfit.

3.8. Anschlußsteuerung (Controller) AMB CM 5001 für Magnetbandgeräte

Bei der Anschlußsteuerung CM 5001 handelt es sich um eine Steuereinheit für Magnetbandgeräte CM 5300 und CM 5303, die den Anschluß von 1 bis 4 MBG des gleichen Types an den SKR-Einheitsbus gestattet. Der SKR-Einheitsbus wird über den Busumsetzer realisiert und ein DMA-Betrieb ermöglicht.

Technische Daten

<i>Leistungsparameter</i>	
Realisiert alle Leistungsparameter der MBG CM 5300 und CM 5303	
Betriebsart	Dauerbetrieb

<i>Umgebungsbedingungen</i>	
Temperatur	1 bis 40 °C
Luftfeuchte	40 % bis 90 % bei 30°C
Luftverschmutzung	max. 0,2 mg/m ³

<i>Aufstellungsbedingungen</i>	
Einschub	19 Zoll/6 U
Gewicht je Einschub	30 kg
<i>Stromversorgung</i>	
Spannung	220V +10%, -15%
Frequenz	50Hz ± 1Hz
Leistungsaufnahme	230 VA

3.9 Magnetbandgerät CM 5300

Das Magnetbandgerät stellt einen externen Speicher mit sequen-
tuellem Zugriff dar.

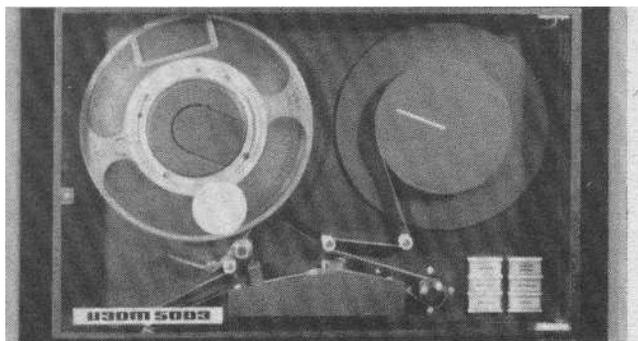
Die Steuerung von jeweils 4 Magnetbandgeräten und die Verbindung zum Obergeordneten Basisrechnersystem erfolgt durch die Anschlußsteuerung AMB (Controller) CM 5001.

Dateneinheiten

1 1/2-Zoll-Magnetbänder	
Spulendurchmesser	max. 216 mm
Bandlänge	360 m/Spule
Typ	DRWO-Computerband

Technische Daten

<i>Leistungsparameter</i>	
Bandgeschwindigkeit	32 cm/s
Omspulzeit	300 s
Aufzeichnungsdichte/ -methode	32 Bit/TT, J11RZ1



Magnetbandgerät CM 5300

Foto: Boden

Spurenanzahl	9
Übertragungsgeschwindigkeit	10 k Byte/s
Betriebsart	Dauerbetrieb
<i>Umgebungsbedingungen</i>	
Temperatur	5 bis 40°C
Luftfeuchte	40 bis 95 % bei 30 °C
Luftverschmutzung	max. 0,2 mg/m ³

<i>Aufstellungsbedingungen</i>	
Einschub	19 Zoll/7 U
Gewicht je Einschub	35 kg
<i>Stromversorgung</i>	
Spannung	220V+10%, -15%
Frequenz	50/60 Hz ± 1 Hz
Netzanschluss	fest installiert
Leistungsaufnahme	250 VA

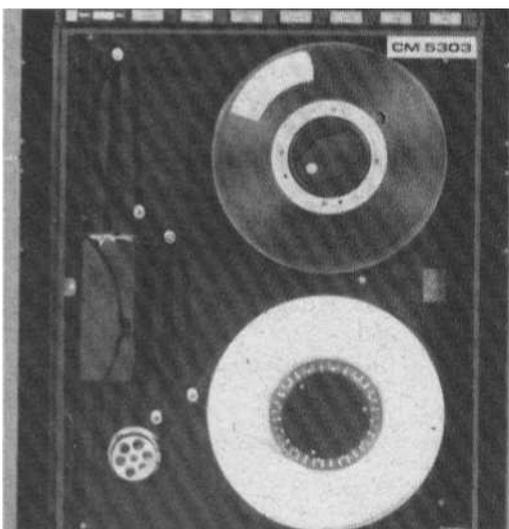
<i>Systemanschließbedingungen</i>	
Typ des Interface	IFMB
Typ der AS	AMB-Controller CM 5001
Busschnittstelle	5KR-Einheitsbus, realisierbar über Busumsetzer (BUM)

TTL-kompatibel

3.10 Magnetbandgerät CM 5303

Das Magnetbandgerät stellt einen externen Speicher mit sequenziellen Zugriff dar.

Die Steuerung von jeweils 4 Magnetbandgeräten und die Verbindung zum übergeordneten Basisrechnersystem erfolgt durch die Anschlußsteuerung AMB (Controller) CM 5001.



Magnetbandgerät CM 5303

Werkfoto

Datenträger

1/2-Zoll-Magnetbänder	
Spulendurchmesser	max. 267 mm
Bandlänge	720 m
Typ	ORWO-Computerband

Technische Daten

<i>Leistungsparameter</i>	
Bandgeschwindigkeit	114 cm/s
Umspizzeit	300 s
Aufzeichnungsmethode	32 Bit/mm, NRZI
Spurenanzahl	9
Übertragungsgeschwindigkeit	32 k Byte/s
Betriebsart	Dauerbetrieb

<i>Umgebungsbedingungen</i>	
Temperatur	5 bis 40 °C
Luftfeuchte	40 bis 95 % bei 30 °C
Luftverschmutzung	max. 0,2 mg/m ³

<i>Aufstellungsbedingungen</i>	
Einschub	19 Zoll/14 U
Gewicht je Einschub	60 kg

<i>Stromversorgung</i>	
Spannung	220V+10%, -15%
Frequenz	50/60 Hz ± 1 Hz
Netzanschluss	fest installiert
Leistungsaufnahme	350 VA

<i>Systemanschließbedingungen</i>	
Typ des Interface	IFMB
Typ der AS	AMB-Controller CM 5001
Bus-Anschluß	SKR-Einheitsbus, realisierbar über Busumsetzer

TTL-kompatibel

3.11. Kassettenplattenspeicher CM 5a00

Der Kassettenplattenspeicher ist ein externer Speicher mit wahlfreiem Zugriff. Er besteht aus einer fest montierten Platte (Festplatte) und einer wechselbaren Platte (Wechselplatte) mit einer Gesamtspeicherkapazität (Wechselplatte + Festplatte) von 5 M Byte.

Datenträger

14-Zoll-Platten	
Festplatte mit 2 speichernden Oberflächen	
Wechselplatte mit 2 speichernden Oberflächen (Kassette)	
Wechselplattentyp	CM EC 5269-01

Technische Daten

<i>Leistungsparameter</i>	
Speicherkapazität	2,5 M Byte je Platte
Aufzeichnungsdichte	87 Bit/mm (2200 bpi)
Drehzahl	2400 U/min
Übertragungsgeschwindigkeit	2,5 M Bit/s
Spurenanzahl je Oberfläche	204
Sektoranzahl	12
Byteanzahl je Sektor	512
Mittlere Zugriffszeit	
— je Sektor	12,5 ins
— je Spur	50 ms

<i>Umgebungsbedingungen</i>	
Temperatur	+ 5 °C bis + 40 °C
Luftfeuchte	95 % bei 30 °C
Luftverschmutzung	max. 1,0 mg/m ³

<i>Aufstellungsbedingungen</i>	
Einschübe	19 Zoll/11 U
Gewicht je Einheit	58 kg



Kassettenspeicher CM 5400

Werkfoto

Stromversorgung

Spannung	220V+10%,-15%
Frequenz	47 bis 63 Hz
Leistungsaufnahme	350 VA je Einschub

Systemanschlüsse

Typ des Interface	IFKP
Typ der AS	AKP
Bus-Anschluß	Systembus K 1600
Kabellänge von der Steuereinheit zum letzten Gerät	max. 15 m
Kompatibilität mit IBM 5440 – Interface	

3.12. Universelles Bildschirmterminal (UBT) K 8931

Das universelle Bildschirmterminal ist ein für höhere Ansprüche einsetzbares, frei programmierbares Gerät. ausgeführt als kompaktes Auftischgerät. Es dient der alphanumerischen **Bildschirmkommunikation** und Datenverarbeitung und ist mit verschiedenartigen Datenträgerbaugruppen und beigestellten Baugruppen ausrüstbar. Das UBT besitzt eine frei aufstellbare, universelle alphanumerische Kompakttastatur.

Aus ergonomischen Gründen wird die Bildröhre um 12° geneigt. Die Steckplätze haben K-1520-Format.

In die kompletten Auftischgeräte können wahlweise die Datenträgerbaugruppen Folienspeicher, Handmagnetgeräte oder Minifolienspeicher integriert werden.

Die **Monitorvariante K 7221** oder **K 7222** kann vom Kunden gewählt werden.

Über IFSS lassen sich SD robotron 1152 bzw. SD robotron 1157 anschließen.

Datenträger:

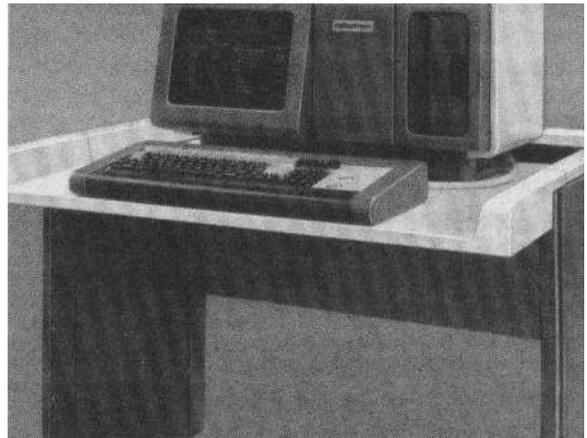
Entsprechend der gewählten Datenträgervariante.

Technische Daten

Leistungsparameter

Steuereinheit K 2526	Doppelprozessor-ZRE 8 Bit Wortbreite
Speicher (RAM)	Kundenspezifisch von 4 bis 64 K Byte aufrüstbar
Speicher (RAM/PROM)	Alternativ zur RAM-Bestückung bis max. 64 K Byte
Tastatur	Alphanumerische Tastatur mit block und Funktionstastenreihe

Bildschirm	31-cm-Diagonale Zeicheninhalte: 16 Zeilen x 64 Zeichen = 1024 Zeichen oder 24 Zeilen x 80 Zeichen = 1920 Zeichen
Datenträger	– Standardfolienspeicher 256 K Byte je Laufwerk (max. 3 Laufwerke) – Kassettenspeicher K 5200 (max. 2 Laufwerke) – Minifolienspeicher 4 M Bit K 5600 (max. 2 Laufwerke)
Drucker	ein SD robotron 1152 oder ein SD robotron 1157
Schrittstellen	V.24;IFSS
Umgebungsbedingungen	
Temperatur	+5 °C bis +40 °C
Luftfeuchte	95 % bei 30°C
Aufstellungsbedingungen	
Abmessungen (B x T)	Bildschirm und ein integriertes Folien- speicherlaufwerk: 672 mm x 408 mm Tastatur: 512 mm x 250 mm
Tastatur bis 1,5 m entfernt aufstellbar. Alle weiteren Datenträgerbaugruppen sowie Drucker sind als Beistell- bzw. Auftischgeräte einsetzbar.	
Gewicht	35 bis 45 kg, je nach Ausstattung
<i>Stromversorgung</i>	
Spannung	220V+ 10%,-15%
Frequenz	50/60 Hz±2Hz
Leistungsaufnahme	200 bis 300 VA, je nach Ausstattung
Netzanschluß	über Schukostecker
<i>Systemanschlüsse</i>	
Typ des Interface	V.24 nach CCITT mit synchroner und asynchroner Übertragung bis 9600 Baud.
	Reibduplex, 1FS
Typ der AS	AIS für MR K 1620/K 1630-Anschluß
Busanschluß	Systembus K 1600



Das universelle Bildschirmterminal robotron K 8931 ist sowohl an die Basissysteme **der Mikrorechnerfamilie K 1600** anschließbar, als auch im autonomen Einsatz zu betreiben – hinsichtlich seiner Leistungsfähigkeit entspricht es dabei etwa dem Buracomputer A 5120.

Foto: Dmowski

Anwendungskomplex Kommerzielle Basisrechnersysteme

Michael Hamann
VEB Robotron-Vertrieb Berlin

1. Einsatzmöglichkeiten der KBR A 6401 und A 6402

Innerhalb der Basisrechnersysteme des Kombinat Robotron gibt es den speziellen und relativ eigenständigen Anwendungskomplex Kommerzielle Basisrechnersysteme (KBR). Er wird charakterisiert von einer Anzahl von Einsatzvarianten, wie

- autonome Rechner in Rechenzentren
- dezentrales arbeitsplatzbezogenes Verarbeitungssystem in Fachabteilungen
- Auskunfts- und Reservierungssystem
- Verarbeitungssystem mit Dialogfunktion zu den im Nah- und Fernbereich angeschlossenen Terminals unterschiedlicher Intelligenz.

Das umfangreiche Spektrum der Einsatzmöglichkeiten der KBR A 6401 und A 6402 ergibt sich aus

- der variablen Speicherkapazität der Mikrorechner K 1620 und K 1630
- der Vielzahl von anschließbaren peripheren Geräten der Datenträgerein- und -ausgabe
- der Anschlußmöglichkeit leistungsfähiger externer Magnetband- und Plattenspeicher
- der Möglichkeit der an-line-Kopplung mit intelligenten Terminals
- der Bereitstellung von zwei nutzerfreundlichen Betriebssystemen für ökonomische und wissenschaftlich-technische Anwendungen problemorientierter Systemunterlagen.

Die kommerziellen Basisrechnersysteme besitzen aufgrund ihrer flexiblen Konfigurierbarkeit ein Leistungsvermögen, das zwischen höher organisierten programmierbaren Tischrechnern und kleineren Anlagen des ESER Hegt.

Die schwerpunktmäßigen Anwendungen umfassen das Gebiet der Bearbeitung ökonomischer und wissenschaftlich-technischer Probleme. Hierzu gehören unter anderem die klassischen Einsatzgebiete

- Lohn- und Arbeitskräfterechnung
- Kostenstellen-, -arten- und -trägerrechnung
- Finanzrechnung
- Produktionsplanung und -lenkung
- Statistik
- Grundmittelwirtschaft
- Materialwirtschaft
- Optimierung
- Matrizenoperationen.

Die kommerziellen Basisrechnersysteme, deren Hauptaufgabe

das Verarbeiten und Speichern umfangreicher Dateien ist, benötigen im Vergleich zu anderen Anwendungskomplexen leistungsfähige externe Speicher. Aufgrund dieser objektiven Notwendigkeit sind an die kommerziellen Basisrechnersysteme außer den Magnetbandgeräten CM 5300 und CM 5303 sowie den Kassettenplattenspeichern CM 5400 zusätzliche Wechselplattenspeicher CM 5405 (Tab. 1) und Kassettenplattenspeicher CM 5410 (Tab. 2) anschließbar.

Tab. 1 Charakteristik des Wechselplattenspeichers CM 5405 (EC 5061 C)

Der Wechselplattenspeicher ist ein Standgerät mit SKR-Kontroller und als externer Daten- und Systemspeicher verwendbar.

Der elektrische Teil besteht aus einem Schrank ISOT 2104 C C001 mit Kontrollor 1066 C, E002, dem Netzteil 1006 C, E 001, der Netzteilsteuerung 2104 C. E 002 und dem Netzfilter.

Es können maximal acht WPS EC 5061 C in Daisy-Chain-Technik (Prioritätsketten) angeschlossen werden.

Als Datenträger finden Plattenstapel mit 11 Platten Verwendung.

- Wortlänge: 16 Bit
- Übertragungsgeschwindigkeit: 7,5 µs/Wort
- Mittlere Zugriffszeit: 62,5 ms
- Fünf Unterbrechungsebenen
- Unterbrechungsebenen: 254
- Oberflächen je Plattenstapel: 20
- Anzahl der Spuren je Oberfläche: 200 (plus 3 Reserve)
- Sektoren je Spur: 10
- Worte je Sektor: 256 (= 512 Byte)
- Nettospeicherkapazität: 20 480 000 Byte
- Aufzeichnungsverfahren: NRZ mit Doppelfrequenz
- Aufzeichnungsdichte: 88 Bit/mm
- SKR-Interface

Tab. 2 Charakteristik des Kassettenplattenspeichers CM 5410

Der Kassettenplattenspeicher CM 5410 ist eine Weiterentwicklung des KPS CM 5400. Das bezieht sich auf eine höhere Speicherdichte in 5 Spuren je Oberfläche.

Nicht aufgeführte Parameter entsprechen denen des KPS CM 5400.

Der zum KPS CM 5410 gehörende Kontroller CM 5109 ist auch für den Anschluß des KPS CM 5400 geeignet.

- 12 Sektoren je Spur
- 256 Worte je Sektor
- 406 Spuren je Oberfläche
- 4 Oberflächen
- Nettokapazität in Byte je Platte: 4 988 928

Die terminalorientierte Konfiguration des kommerziellen Basisrechnersystems erlaubt, über die Ausstattung mit konventioneller Peripherie hinaus, den Anschluß von Geräten der Datenübertragung und -steuerung, nichtprogrammierbarer Dialog- und intelligenter Terminals sowie anwendungsspezifischer Terminals. Für die Datenübertragung stehen Datenfernverarbeitungsprozessoren, Multiplexoren und Übertragungseinrichtungen zur Verfügung.

Hierzu gehören der Konzentrador K 8521 und der Multiplexer K 8523. Sie dienen als festprogrammierte Geräte der multiplexen Steuerung von max. 16 Leitungsadaptern für die Datenübertragung.

An das kommerzielle Basisrechnersystem können für den terminalorientierten Einsatz über einen Multiplexer bzw. Konzentrador im Nahbereich über Interfacekabel und im Fernbereich

über Modem bzw. CDN bis zu 16 Terminals angeschlossen werden.

Die arbeitsplatzbezogene Datenverarbeitung nimmt tendenziell an Bedeutung zu. Schwerpunkt sind hierbei in den Fachbereichen stationierte und den Nutzerforderungen entsprechend konfigurierte Terminals, die mit dem kommerziellen Basisrechnersystem on-line verbunden sind und im Dialogbetrieb arbeiten.

Diese Form der Datenverarbeitung ist besonders dort effektiv, wo

- relativ geringe Datenmengen einzugeben sind
- das Ergebnis sofort benötigt wird
- unmittelbar Auskünfte und Alternativen anzubieten sind.

Repräsentative Anwendungen hierfür liegen

- in Handelsbereichen zur Führung der Verkaufsdisposition
- im Verkehrs-, Transport- und Hotelwesen sowie in Reisebüros für Reservierungsaufgaben
- auf Leitungsebenen zur aktuellen Information über den Erfüllungsstand wichtiger betrieblicher Kennziffern
- im Großhandel zur Nachweisführung der Warenbewegungen
- in der Materialwirtschaft zur Bestandführung und Disposition
- in Geld- und Kreditinstituten zur Verbesserung der Dienstleistungen für den individuellen und kommerziellen Nutzerbereich.

Zu den für derartige Aufgaben verwendbaren anwendungsspezifischen Terminals des kommerziellen Basisrechnersystems gehören neben den Bildschirmterminals BTL K 8912 und UBT K 8931 ein Platzreservierungsterminal und ein Bank- und Sparkassenterminal.

2. Platzreservierungsterminal robotron K 8927

Das Terminal K 8927 ist ein modular aufgebautes Datenendgerät innerhalb eines DV-Systems. Sein Ersteinsatz erfolgt im System zur automatischen Platzreservierung der Eisenbahn für die Personenbeförderung (ARS(E)). Die Anwendungsgebiete dieses Terminals liegen hauptsächlich in Reisebüros, im Hotelwesen und im Dienstleistungsbereich. Das Terminal wird im Bereich der Kundenbedienung eingesetzt und ist durch die Funktionen Anforderung, Auskunft und Belegausgabe gekennzeichnet. Weiterhin ist die Ausgabe von Bedien- und Systeminformationen, Belegungslisten, Berichten und Statistiken sowie die Eingabe von Stammdaten möglich.

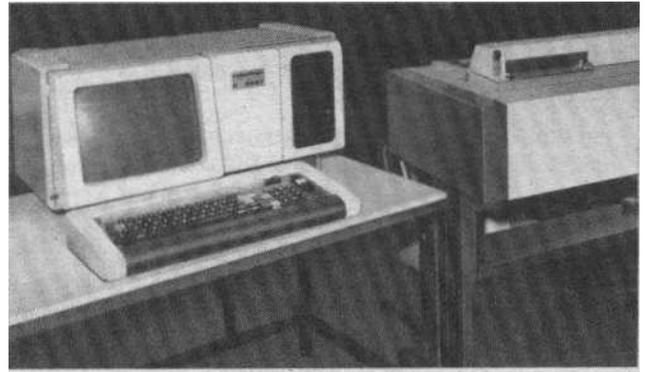
Folgende Merkmale charakterisieren das Terminal: Steuerung durch Mikroprozessor, off-line- und on-line-Betrieb (Dialogverkehr), frei programmierbar.

Die Hauptfunktionen sind Dateneingabe, Anzeige, Bedienungsführung, Kontrolle, Datenaufbereitung, Primärverarbeitung, Beleg- oder Quittungsdruck, Datenspeicherung und Datenübertragung.

Gerätetechnik

Das Terminal besteht aus zwei Hauptkomponenten:

- *Кommunikationseinheit*
Auf Tischgerät mit abgesetzter Tastatur; dazu gehören alle Baugruppen zur Kommunikation, die Steuerung der peripheren Baugruppen, der Datenübertragungsanschlüsse, der Programm- und Datenspeicher und die Bedieneridentifizierungseinrichtung.



Platzreservierungsterminal robotron K 8927

Foto: Boden

• Druckwerk

Standgerät mit Zusatzformulareinrichtung zur maschinellen Vereinzeln von Endlosleporello sowie Kontenvorsteckeinrichtung.

Zentraleinheit:

ZRE K 2526; Speicher K 3520, 3521, 3525 und 3820; Übertragungsraten: maximal 122 k Byte/s; Simultanarbeit wird über Betriebssystem gesteuert. Speicherkapazität: max. 64 K Byte

Tastatur:

85 alphanumerische Zeichenfunktionen; 46 Startfunktionen, davon 36 Stationstasten-Funktionen

Datensichtgerät:

Monitor K 7221 (16 x 64 Zeichen) bzw. Monitor K 7222 (24 x 80 Zeichen/12 x 40 Zeichen auf Bildschirm darstellbar)

Drucker:

Seriendruker 1156; Druckgeschwindigkeit: 100 Zeichen/s
Seriendruker 1152; Druckgeschwindigkeit: 30 Zeichen/s

Zusatzformularausstattung:

zweibahnig Leporello, davon einbahnig maschinelle Vereinzeln von Endlosleporello sowie Konteneinzugsvorrichtung (wahlweise)

Externe Speicher:

Folienspeicher CM 5601; Aufzeichnungsverfahren: IS0/TC-97/SC-11 Nr. 149.

Es besteht die Möglichkeit, das Platzreservierungsterminal mit einer peripheren Ausstattung entsprechend Bildschirmterminal K 8931 bzw. Bürocomputer A 5120 zu konfigurieren.

Software

Zu den maschinenorientierten Systemunterlagen gehören:

- Modulares Steuerprogramm für Simultanarbeit SIEX 1526
- Makrobefehlsinterpreter
- Texteditor
- Makrobefehlsassembler
- Binder
- Makroprogrammtesthilfe
- Hilfs- und Dienstprogramme
- BASIC-Compiler.

Die problemorientierten Systemunterlagen umfassen Anwenderprogramme (ARS/E-typisch) und problemorientierte Programmierhilfen.

3. Bank- und Sparkassenterminal robotron K 8924

Das Terminal wird in zwei Varianten bereitgestellt:

- K 8926
in Form eines Kompaktaufschgerätes mit Steuereinheit und Bildschirm sowie beigestellten Diskettenlaufwerken, Drucker. Schreib-Lese-Einheit für Geldkarten und Komplett-tastatur
- K 8924 — als erweiterter Terminalkomplex;
in Form einzelner geometrisch-funktioneller Huftisch-/ЕиР-
baumoduh^, die nur über Interfacekabel miteinander ,erbun-
den sind /И/.

Diese Varianten bieten Gerätetechnik und Systemunterlagen, mit denen nahezu alle Anforderungen eines spezialisierten Schalterarbeitsplatzes realisierbar sind.

Das Anwendungsfeld dieses Terminals liegt in der Rationalisie-
rung geldwirtschaftlicher Prozesse, bei schwerpunktmäßiger
Einbeziehung schalterspezifischer Aufgaben in Banken und
Sparkassen, bei der Post und im Reisezahlungsverkehr.

Das Terminal kann in den einzelnen Emrichtungen folgende
Funktionen erfüllen:

- *Banken*
— Erfassen aller Ein- und Auszahlungen
— Erfassen von bargeldlosen Zahlungsaufträgen für alle bei
Banken und Sparkassen geführten Konten
— Realisierung von Kontrollfunktionen bei der Erfassung aller
baren und unbaren Zahlungsaufträge
— Übermittlung der erfaßten Daten zum Rechner zwecks Ver-
arbeitung
— Ausgabe von Kontoauszügen an die Kontoinhaber
-- automatische Kassenabrechnung für alle über das Terminal
abgewickelten baren Zahlungsaufträge
- *Sparkassen*
— Verarbeitung aller Daten, die bei der Realisierung des Kassen-
verkehrs und bei der Kontoführung am Schalter anfallen.
Die anderen Funktionen sind denen in Banken analog.
- *Reisezahlungsverkehr*
— automatische Bearbeitung sämtlicher am Schalter anfallen-
der Geschäftsvorfälle des Reisezahlungsverkehrs einschließ-
lich Belegausdruck
— automatische Abrechnung von im Terminal verwalteten
Währungs-ипд Valutaplandateien sowie von Dateien der
Kassenabrechnung.
- *Post*
— Realisierung des Zahlungsverkehrs der Post und
— Realisierung von Zahlungsaufträgen für Kontoinhaber der
Banken und Sparkassen.

Gerätetechnik

Die Hauptkomponenten des Bank- und Sparkassenterminals
rind:

- *Steuereinheit*
ZRE K 2526 (K 1520)
Interner RAM-Speicher für Daten und Anwenderprogramme,
aufriistbar bis 60 K Byte
Interner Betriebssystemspeicher bis 48 K Byte (RAM) mit
Schreibschutz
— Anschlußsteuerungen für periphere Geräte
— GDN als Option

— Stromversorgungseinheit einschließlich verschleißbarer Ein-
ипд Ausschaltleinrichtung

- *Bildschirm*
— Monitor K 7221 oder K 7222 analog UBT K 8931
- *Tastaturen*
— alphanumerische bzw. numerische Tastaturen, den jeweili-
gen Anforderungen des Schalterverkehrs angepaßt.
- *Drucker*
Es werden Drucker mit spezieller Formulartechnik eingesetzt.
Sie sind geeignet für den Belegquittungs- und Entwertungsdruck.
Belegerstellungsdruck, Kassenjournaldruck, Datenerfassungs-
protokollruck und Administrativdruck.
Die Druckbreiten betragen je nach Druckertyp 80, 132 oder
210 Zeichen/Zeile. Druckgeschwindigkeiten: 35/100/180
Zeichen/s.

• *Lese- und Schreib-Lese-Einheit für Geldkarten*
Für den Ersatz der Geldkarten (Plastkarten mit Magnetstreifen
nach ISO 3554 und 4909) gibt es unterschiedliche Schreib-
Lese-Technik:

- einfache Leseeinheit für manuellen Kartendurchzug
- automatische Leseeinheiten und
- automatische Schreib-Lese-Einheiten.

Die Geldkarte dient der

- Realisierung der Bargeldabhebung über Bankschaltertermi-
nals oder Geldautomaten
- bargeldlosen Bezahlung von Waren und Dienstleistungen
Ausführung von Überweisungsaufträgen sowie Kontostands-
abfragen und dem Kontoauszugdruck.
- Das Terminal K 8924 kann bei Bedarf mit *Folienspeicher-*
oder Mini-Folienspeicherlaufwerken oder *Kassettenmagnet-*
Bandgeräten K 5020 konfsguriert werden.

Software

Als Betriebssystem kommt das SIOS 1526, das aus dem Steuer-
programni SIEX 1526 und dem Makrobefehlsinterpreter MINT
1520 besteht, zum Einsatz.

4. Weitere Einsatzgebiete

Neben den traditionellen Einsatzgebieten werden die kommer-
ziellen Basisrechnersysteme robotron A 6401/A 6402 zuneh-
mend als komplexe Anwendungssysteme eingesetzt, zum Bei-
spiel in der Medizin für Aufgaben der patientenbezogenen In-
formationsverarbeitung in Krankenhäusern:

- Anamnese, Patientenaufnahme, Verlegung und Entlassung
- Erstellung der patientenbezogenen Dokumentation
- Auswertung medizinisch-statistischer Daten
- Ablaufoptimierung der medizinischen Diagnostik
- Planung, Leitung und Verwaltung.

Voraussetzung zur Bearbeitung dieser Aufgaben rind Daten-
banken, die auf leistungsfähigen externen Plattenspeichern mit
wahlfreiem Zugriff archiviert rind. Die Dateneingabe und der
Zugriff zu den Dateien erfolgt hierbei vorwiegend über Bild-
schirmterminals-

Literatur.

/IJ Waloszyk, U.: *Bank- und Sparkassenterminal robotron K 8924.*
NTB 26(1982)2

Der Arbeitsplatz für Konstruktion und Technologie A 6454

Wolfgang Reese
VEB Robotron-Vertrieb Berlin

1. Vorbemerkung

Die Rationalisierung der produktionsvorbereitenden Prozesse gewinnt im internationalen Maßstab ständig an Bedeutung. Damit verbunden ist ein überproportionales Wachstum des Einsatzes von CAD- und CAM-Systemen in der Industrie. Solche Systeme ermöglichen für Teilprozesse der technischen Vorbereitung der Produktion Produktivitätssteigerungen bis zu 600 Prozent, im Durchschnitt werden 300 Prozent erreicht.

Die Bereitstellung des Arbeitsplatzes für Konstruktion und Technologie AKT A 6454 ist ein Beitrag des Kombinats Robotron auf dem Gebiet der interaktiven grafischen CAD/CAM-Systeme. Der AKT A 6454 ermöglicht die grafische Ein- und Ausgabe, die grafische Darstellung und grafische Manipulation für verschiedene Anwendungen. Die Bereitstellung von standardisierbaren Grundsystemunterlagen entspricht dem internationalen Stand auf diesem Gebiet (siehe S. 45).

2. Haupteinsatzgebiete

Der Arbeitsplatz für Konstruktion und Technologie AKT A 6454 dient zur rechnergestützten Rationalisierung der technischen Vorbereitung der Produktion.

Haupteinsatzgebiete sind die

- rechnergestützte Erstellung von Zeichnungen
- rechnergestützte Projektierung von Maschinen und Anlagen
- Berechnung von Einzelteilen
- rechnergestützte Programmierung von NC-Maschinen
- rechnergestützte Ausarbeitung von Arbeitsplanstammkarten.

3. Gerätetechnik des AKT A 6454

Der AKT A 6454 ist eine anwendungsbezogene Konfiguration auf der Basis des KBR A 6402 und besteht aus einem Rechner K 1630 mit Standardperipherie, grafischer Peripherie und Systemunterlagen. Als arbeitsplatzbezogene grafische Geräte sind anschließbar:

- Rastersichtgerät RSG K 8917
- hochauflösendes Digitalisiergerät HDG K 6401
- Digitalzeichentisch DZT 90x 120
- Digigraf 1208-3,5 G

Abb. 1 zeigt eine mögliche Konfiguration des Arbeitsplatzes. Art und Anzahl der im konkreten Anwendungsfall anzuschließenden peripheren Geräte hängen von dessen spezifischen Anforderungen ab.



Neben dem im vorigen beschriebenen AKT A 6454 zeigt der abgebildete A 6452 die Vielfalt der arbeitsplatzbezogenen Konfigurationsmöglichkeiten der Basisrechner. Er wurde speziell zur Entwicklung und Konstruktion von Leiterplatten mit rechen technischer Unterstützung geschaffen. Werkfoto

3.1. Mikrorechner K 1630 und nichtgrafische Peripherie (Grundkonfiguration)

Wegen der relativ hohen Anforderungen an die Verarbeitungsleistung, die sich aus der grafischen Arbeit ergeben, wird im AKT A 6454 als Steuerkern ein Mikrorechner K 1630 mit Arithmetikprozessor und einem Arbeitsspeicher von mindestens 128 K Byte verwendet.

Als Standardperipherie KBR bzw. nichtgrafische Peripherie kommen im AKT A 6454 zum Einsatz:

- Bedieneinheit BDE K 8911
- Kassettenplattenspeicher KPS CM 5400
- Magnetbandgerät CM 5300 oder 5303
- Folienspeichereinheit FSE K 5665
- Lochbandeinheit LBEK6200
- Bildschirmterminal BTL K 8912
- Seriendrucker robotron SD 1157

3.2. Grafische Peripherie

3.2.1. Grafisches Display-Rastersichtgerät

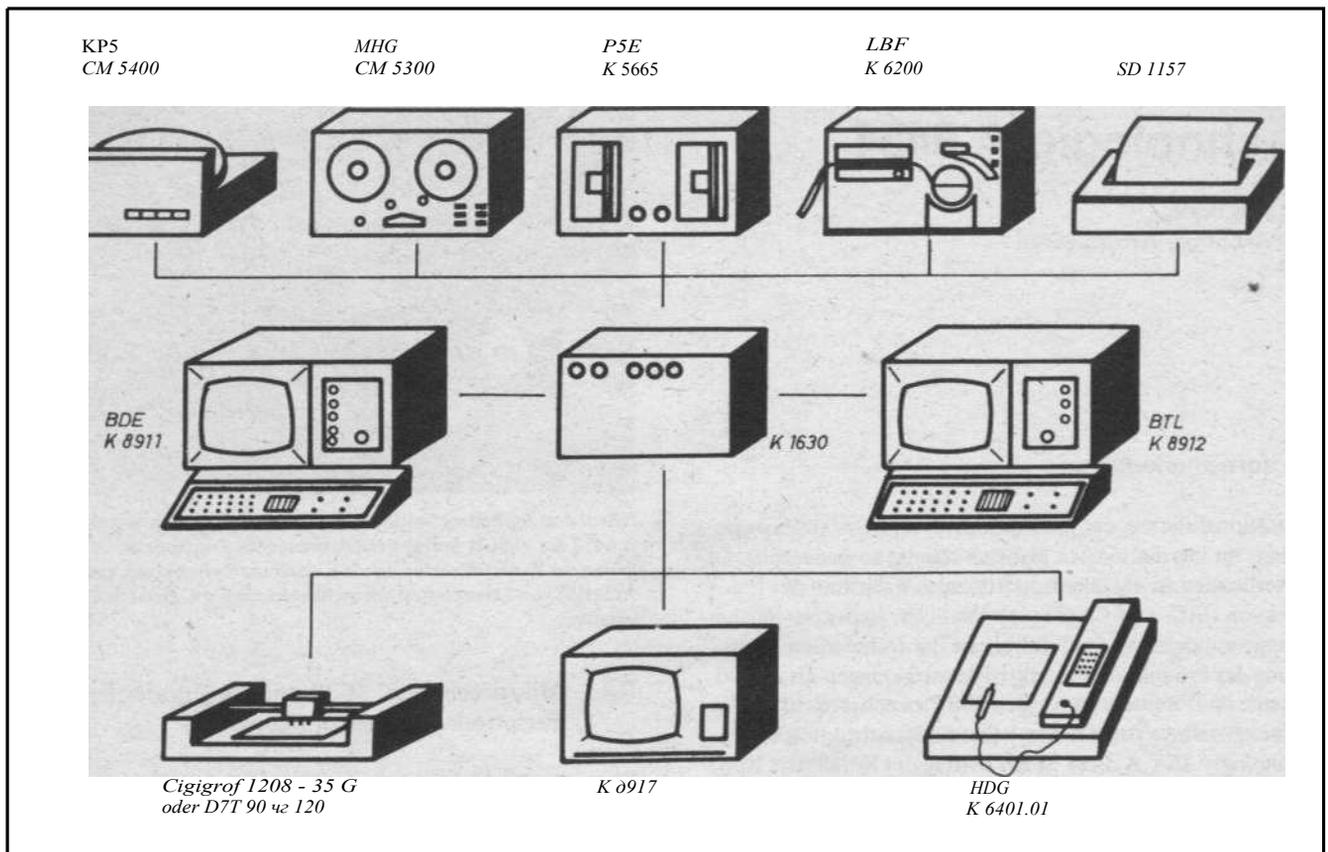
Das Rastersichtgerät mit Monitor (MON K 7222) RSG K 8917 ermöglicht eine passive Ausgabe alphanumerischer und grafischer Informationen und eine interaktive Arbeit über Tastatur und cursor für Anwendungen mit geringen Anforderungen an die darzustellenden und zu bearbeitenden geometrischen Daten. Es verfügt über einen internen Segmentspeicher und Bildwiederholungspeicher. Als Steuereinheit wird ein Mikrorechner K 1520 verwendet.

Technische Parameter

Bildschirmdiagonale 310 mm
Betriebsart alphanumerisch und grafisch

alphanumerischer Modus:

Leistungsfähigkeit wie BDE K 8911
Arbeitsfeld etwa 210 x 145 mm²
Darstellformat 24 Zeilen x 80 Zeichen/Zeile
Intensitätsstufen 2



Абб. 1 Конфигурация вырегл дев АКТ А 6454

grafischer Madus:

Arbeitsfeld etwa 210x100mm²
 Darstellformat 640 x 288 Punkte
 Bll dwiederhol Speicher 640 x 408 Punkte
 Rollmadus auf-/ebwärts
 Linienarten 5
 Intensitätsstufen 1
interface IF55

Technische Parameter

Digitalueichenitirch DZT 90x 120:

Arbeitsfläche 900 x 1200 mm
 Auflösung p,005 mm
 mittlerer **Punktfehler** 0,07 mm
 max. Gcschwindigkeit 170mmfs#5%a
 (achsparallel)
 Beschleunigung 0,12 g -
 Anzahl der Werkzeuge 2
 Linienarten 10
 Interface V.24 /

3.2.2. Hachaufläsendes Digitalisiergerät

Das hochauflösende Digitalisiergerät HDG 1 (K 6401.01) dient der Erfassung grafischer Daten, deren Vorverarbeitung und Verdichtung mittels eines geräteinternen Rechners und der Übergabe als digitale Information an den Basisrechner K 1630 des AKT A 6454 zur weiteren soft- und hardwaremät^igen Verarbeitung. Das HDG ermöglicht die Digitalisierung von Vorlagenformaten bis zur Größe A2 an einem Sitzarbeitsplatz.

Digigraf 1208-3,50:

Arbeitsfläche **1189** x 841 mm
 Auflösung 0,01 mm
 dynamische Zeichengenauigkeit bei 0,2 g Beschleunigung ± 0,1 mm
 Beschleunigung max. 0,3 g
 Anzahl der Werkzeuge 4
 Linienarten 8

Technische Parameter

Tischgerät
 Arbeitsfläche 420 x 594 mm
 Aufösung 0,05
 Genauigkeit fiber die gesamte Arbeitsfläche ± 0,1 mm
 Interface i ss

Betriebssysteme digitaler Rechenanlagen

Von Dr. Christoph Polze
 Herausgeber: VEB Robotron —
 Zentrum für Forschung und Technik Dresden
 Schriftenreihe Informationsverarbeitung, Forts.-Nr.: 109
 224 Seiten, 42 Abbildungen, Broschur, 14,00 M
 Kurzwort: Poize, Rechenanlagen, Bestell-Nr.: 675 454 5

3.2.3. **Plotter**

Fur den AKT A 6454 wird der Digitalzeichentisch DZT des VEB Carl Zeiss JENA bereitgestellt. Der Anschluß des Digigraf (CSSR) ist foX ausgewählte Anwendungen vorgesehen.

Prozeßrechnersysteme A 6491 /A 6492

Ronald Nordmeier, Joachim Sobe
VEB Robotron ZFT Dresden

1. Vorbemerkung

Auf der Basis der Mikrorechner K 1620/K 1630 und der Prozeßein-/ausgabeeinheit (PEAE) ursadat 5000 wurde eine neue Generation von PRS entwickelt, welche die Prozeßrechner auf Basis der Familie robotron 4000/ursadat 4000 ablöst und darüber Hinaus neue Anwendungsgebiete erschließt.

Mittels variabler Ausstattung mit Anschlußsteuerungen und Peripherie wird der als Prozeßleitrechner fungierende K 1600 dem jeweiligen Einsatzfall gerecht, ebenso sind die ursadat-5000-Einrichtungen durch modulare Aufrüstbarkeit an die vorgegebenen prozeßspezifischen Bedingungen anpassbar.

Die PEAE auf der Basis des Mikrorechners K 1520 übernimmt auf Grund ihrer Eigenständigkeit Meßwerterfassung, Primärverarbeitung und Steuerwertausgabe. Der Prozeßleitrechner wird somit entlastet, und die Möglichkeit der Dezentralisierung des Prozeßrechnersystems ist vorhanden.

Der Vertrieb der Prozeßrechnersysteme A 6491/A 6492 einschließlich der PEAE ursadat 5000 des KEAW Berlin erfolgt über den Robotron-Atlagenbau Leipzig.

2. Anwendung und Einsatzgebiete

Wesentliche Einsatzvarianten der PRS sind:

- komplette Prozeßrechenanlagen unterschiedlicher Leistungsfähigkeit und Konfiguration
- Prozeßleitrechner für hierarchisch strukturierte Automatisierungsanlagen
- kombinierter Einsatz von KBR und PRS.

Die Automatisierung von Fließprozessen, wie sie zum Beispiel in Kraftwerken und in der Chemieindustrie auftreten, sowie die Laborautomatisierung sind wichtige Anwendungsgebiete der PRS A 6491 /A 6492.

Die Aufgaben, die der Mikrorechner K 1600 hauptsächlich übernimmt, sind:

- Verarbeitung von Dialogeingaben
- Übernahme aktueller Meßdaten und Alarmmeldungen aus der PEAE
- Aufbereitung und Aktualisierung sowie Speicherung des aktuellen Prozeßzustandes
- Aufbereitung und Ausgabe verschiedener Zustände und Protokolle.

Funktionen, die von der PEAE ausgeführt werden:

- Zeitzyklische Meßwertabfrage
- Primärverarbeitung
- Zusammenstellen von Meßstellendaten zu zugriffsfreundlichen Blöcken

- Behandlung von Störmeldungen und Alarmanalysen
- Behandlung von spontanen Signalen aus dem Prozeß.

3. Systemübersicht Prozeßrechnersysteme

A 6491:

Komplette Prozeßrechenanlage mit ursadat-5000-Einrichtung und Mikrorechner K 1620 einschließlich Systemunterlagen

A 6492:

Wie A 6491, aber mit Mikrorechner K 1630.

Der als übergeordneter Rechner eingesetzte K 1600 kann nach Bedarf mit der im Konfigurationsbestand der Anwendungskomplexe KBR/PRS aufgeführten Gerätetechnik komplettiert werden.

Entsprechend den örtlichen Einsatzbedingungen werden die ursadat-5000-Einrichtungen zentral, nahe dem Prozeßleitrechner oder dezentral in Prozeßnähe und größerer Entfernung zum K 1600 aufgestellt.

Die Verbindung zwischen Systembus K 1600 und PEAE wird über das schnelle serielle Interface (IFLS) der Anschlußsteuerung für technologische Prozesse (ATP) des K 1600 realisiert. Zur Überbrückung größerer Entfernungen wird zusätzlich zur ATP-Steureinheit (SE) die ATP-Übertragungseinheit (OE) eingesetzt. Auch zur Kopplung zwischen zwei K-1600-Systemen kann die DMA-Fähige Anschlußsteuerung AI^P verwendet werden (Abb. 1). Da die Zentraleinheit und Peripherie des K 1620 bzw. K 1630 bereits in /1/ vorgestellt wurden, wird im folgenden nur die prozeßspezifische Gerätetechnik betrachtet.

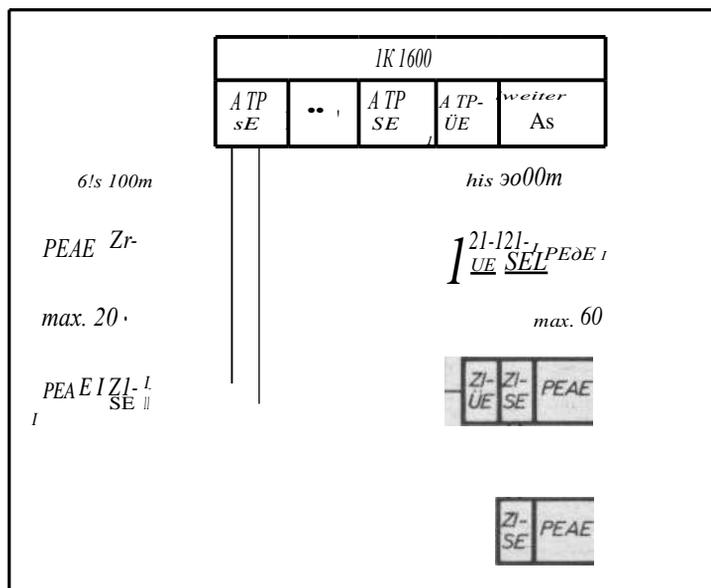


Abb. 1 Systemübersicht des PRS A 6491/A 6492

4. Prozeßein-/ausgabeeinheit ursadat 5000

4.1. Aufbau

Die Prozeßein-/ausgabeeinrichtung als Bestandteil des PRS besteht entsprechend dem Umfang und der Systemarchitektur aus ein oder mehreren Einheiten ursadat 5000. Eine ursadat-Einheit umfaßt dabei 1 bis 3 Grundeinheiten (GE), die in einem

EGS-Gefäß 3. Ordnung (Schrank 800 x 600 x 800/1200/1600/2000 mm³ [B x T x H]) untergebracht sind. Weitere Bestandteile sind:

- Stromversorgungsbaugruppen für die Betriebsspannungen und im begrenzten Umfang für die Geberspeisung
- Netzeingangsteil, einschließlich der Stromversorgungsüberwachungsbaugruppen
- Lüfterbaugruppen und Temperaturüberwachung
- schrankinterne Rangierverteiler, bestehend aus EFS-Steckverbindern zum Anschluß der Prozeßkabel und EFS-Verteilerleisten.

42. ursadat-5000-Grundeinheit

Die GE ist die kleinste funktionelle Einheit des ursadat-5000-Systems. Jede GE enthält einen eigenen Mikroprozessor und Speichermodul. Es werden die Moduln des Mikrorechnersystems K 1520 und die ursadat-Prozeßmoduln eingesetzt. Das Modulsortiment ist dem Konfigurator (Abb. 2) zu entnehmen. Eine GE kann maximal 28 bis 30 adressierbare E/A-Moduln (Proze1 - und DV-Moduln) enthalten. Diese sind in em oder zwei EGS-Baugruppenträgern, mit jeweils 23 Steckplätzen, angeordnet. Alle Plätze verfügen über das Bussystem K 1520. Die Steckeinheiten haben direkte Steckverbinder.

4.3. Modulbeschreibung

Im folgenden werden nur die Prozeßmoduln behandelt. Die K-1520-Moduln sind in /2/ und /3/ beschrieben. Als Prozessor wird ausschließlich die ZRE K 2521 eingesetzt. Mit Ausnahme der Analogeingabe besteht jede GE aus einer Steckeinheit. Der prinzipielle Aufbau der AE ist in Abb. 3 dargestellt.

- Analogeingabe (AE) Karteneinschub (KES) 231x
Sie besteht aus den Komponenten
Grundkarte (AE-G): Multiplexer für 8 Eingänge, Verstärker, ADU
- Expanderkarte (AE-E): Multiplexer für 24 Eingänge

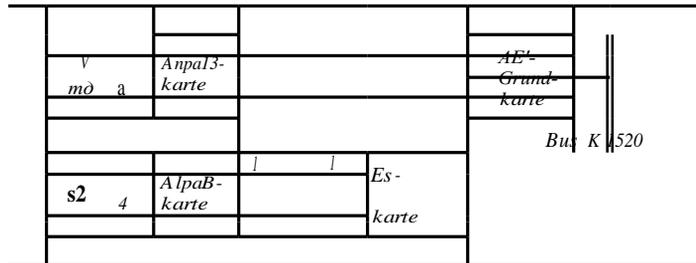


Abb. 3 Analogeingabe ursadat 5000

Signalanpassungskarten (AE-xx): Signalumsetzung auf Einheitssignal 1 V. Filter, 4 oder 8 Eingänge (je nach Typ)
Eingangssignale: 5/10/20 mA; live-zero-Signale (bipolar) 10/20/50/100 mV; **1/10 V** (bipolar)
Widerstand sferngeber
Widerstandsthermometer **PT 100** (-200 °c ... +550 °c)

- Ausgangssignal: **11 Bit** + Vorzeichen
- Eingangskanäle: 56
- Abtastgeschwindigkeit: 10000 Abfragen/s
- 1-Kanal-Analogausgabe (AA-1K) KES 2301
Eingangssignal: 12 Bit (unipolar); 11 **Bit** + Vorzeichen (bipolar)
Ausgang: 1 Kanal; 5/10 mA, live-zero-Signale, 10 V; galvanische Trennung
- 5-Kanal-Analogausgabe (AA-5K) KES 2303
Eingangssignal: 8 Bit (unipolar); 7 Bit + Vorzeichen (bipolar)
Ausgang: 5 Kanäle; 5 mA, live-zero-Signal, 10 V
- Digitaleingabe statisch (DES) KES 2340
Eingang: 16 Bit; Optokoppler; TTL, 5/12/24/24+24/60 V
- Digitaleingabe dynamisch (DED) KES 2342
Eingang: 16 Bit; Optokoppler; TTL, 5/12/24/60 V
- Digitaleingabe statisch mit kurzschlußfestem Treiberschaltkreis (DES-KT) IES 2344
Eingang: 32 Bit; Schaltkreis D 410

X

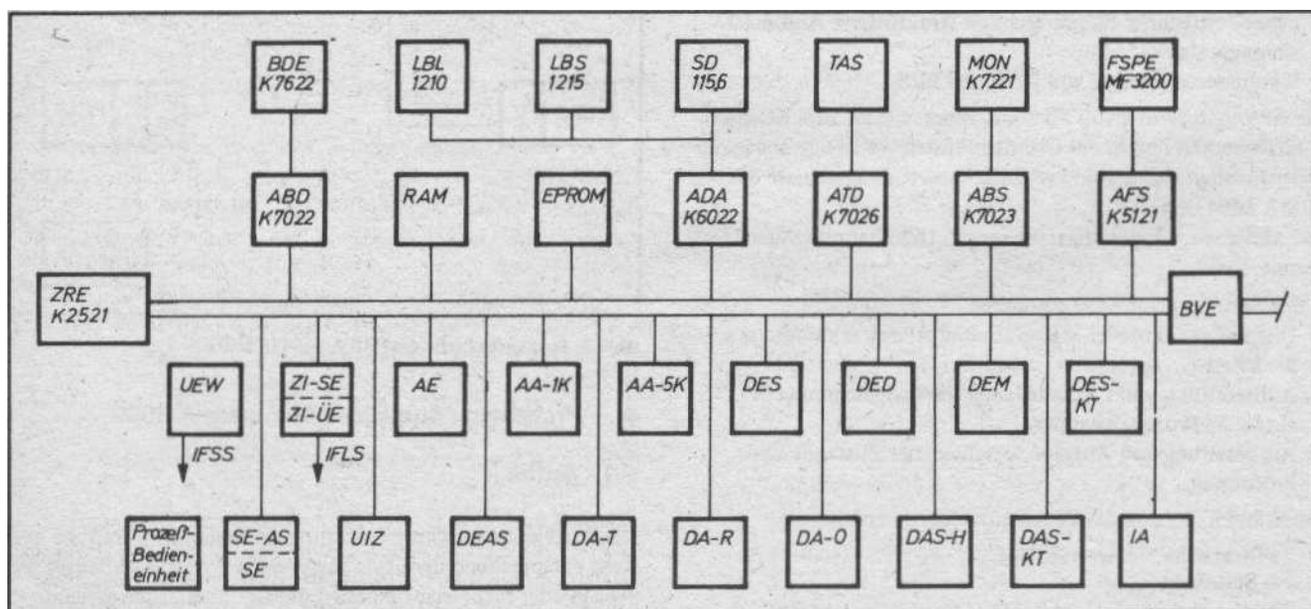


Abb. 2 Konfigurator usadat 5000

- Digitaleingabe multiplex (DEM) KES 2341
Ausgang: 16 Bit; Transistor über Optokoppler;
60 V-/100 mA
Eingang: 16 Gebergruppen zu je 8 Bit; Optokoppler;
12/24/60 V
- Universalimpulszähler (ULZ) KES 2343
Eingang: 4 Kanäle; Optokoppler; TTL, 5/12/24/60 V
20 Hz, 20 kHz, 100 kHz (TTL); Informationsbreite 32 Bit
Betriebsart (software-programmierbar):
Impulszähler 4 x 8 Bit ... 1 x 32 Bit
Zeitgeber + Impulszähler 2 x 8 oder 1 x 16 Bit
Zeitmesser (+ Impulszähler 1 x 8 Bit)
Frequenzmesser 2 x 8 Bit oder 1 x 16 Bit
- Digitalein-/ausgabe (DEAS) KES 2337
Informationsbreite: 3 x 8 Bit, in Stufen zu 4 Bit pro-
grammierbar
Signalpegel: TTL
- Digitalausgabe-Transistor (DA-T) KES 2336
Ausgang: 32 Bit; Transistor, 60 V-/0,12 A/7,2 W;
dynamisch bis 6,8 s
- Digitalausgabe-Relais (DA-R) KES 2331
Ausgang: 24 Bit; Relais RGK 20/1, < 60 V =/0,5 A/10W;
Dynamisch bis 6,8 s
- Digitalausgabe-Optokoppler (BA-O) KES 2335
Ausgang: 16 Bit; Transistor über Optokoppler,
60 V-/0,1 A/3 W; dynamisch bis 6,8 s
- Digitalausgabe statisch mit Haftrelais (DAS-H) KES 2330
Ausgang: 8 Bit; Haftrelais C6, 60 V/0,4 A/6 W; Aus-
gänge rücklesbar
- Digitalausgabe statisch mit kurzschlußfestem Treiberschalt-
kreis (DAS-KT) KES 2334
Ausgang: 32 Bit; Schaltkreis D 410
- Impulsausgabe (IA) KES 2339
Informationsbreite: 32 Bit
Betriebsart (software-programmierbar):
Zeitsignalausgabe, 6,5 l.&s... 6,8 s (interner Takt)
4 Kanäle (vor/rück); Relais RGK 20/1 und Transistor
Impulszahlausgabe, 0,08 Hz ... 77 kHz (interner Takt)
2 Kanäle + Vorzeichen; TTL, Transistor (Vz)
- Überwachungsmodul ((JEW) KES 2328
Überwachungsfunktion: Takt, WAIT, READY, Stromver-
sorgung, 3 freie Eingänge
Ausgang: 2xIFSS (2400 ... 9600 Baud); Störungsmitten-
signal
- Serviceeinheit (SE) und Anschlußsteuerung (SE-AS) KES
2351 für Prüfung, Inbetriebnahme und Wartung sowie Pro-
grammtestung
- Prozedur-Bedienungseinheit
Ein-/Ausgabe von Prozedurkennwerten; Kodierung von Be-
triebsparametern; Eingabe von Kommandos

4.4. Interface und Informationsübertragung

Das mehrpunktfähige Linieninterface IFLS ermöglicht den Echtzeitbetrieb mit verteilten ursadat-Einheiten.

Das Übertragungssystem wird in zwei Entfernungsbereiche unterteilt:

- Nahbus (ohne Übertragungseinheit): 100 m, max. 20 Stationen
- Fernbus (mit Übertragungseinheit): 3 km, max. 60 Stationen, galvanische Trennung, Bi-Phasendifferenz-Modulation.

Die Stationszahl reduziert sich durch die effektive Datenrate und ist stark von den spezifischen Anwendungsbedingungen abhängig. Innerhalb eines Systems sind beide Bussysteme als auch die gemischte Variante Fernbus-Nahbus anwendbar (siehe Abh. 1).

Das IFLS wird auf der Seite des K 1600 mittels ATP und ursadatseitig mittels Zwischenblock-Interface (ZI) realisiert. Beide AS bestehen aus jeweils zwei Steckeinheiten: der Steuereinheit, die den Nahbus realisiert und der Übertragungseinheit. Als Übertragungsmittel werden HF-koaxialkabel verwendet.

Weitere Parameter des Interface sind:

- Übertragungsgeschwindigkeit: 500 k Bit/s
- effektive Datenrate : ca. 120 k Bit/s
- Telegrammlänge 256 Byte.

Die Informationsübertragung erfolgt mit Hilfe begrenzter Bitfolgen (Übertragungsrahmen) byteseriell entsprechend HDLC-Prozedur. Ein Übertragungsrahmen hat folgenden Aufbau: Synchronisationsbyte/Adressbyte/Funktionsbyte/Informationsfeld (entspr. HDLC) 2 Byte Kontrollteil/Synchronisationsbyte.

Es bedeuten:

- Adressbyte: Adresse der Empfangsstation
- Funktionsbyte: Angaben über Format, Stationstyp und Prozedur
- Informationsfeld: Adresse der Sendestation/Anzahl der Datenbyte/Datenfeld, max. 250 Datenbyte.

Das IFLS unterscheidet prinzipiell fünf Stationstypen:

- Bussteuerzentrale
- Steuerstation mit Busanforderung
- Steuerstation
- gesteuerte Station
- Mithörstation.

Für die Systeme A 6491/A 6492 ist festgelegt, daß der K 1600 immer den Rang der Bussteuerzentrale und die PEAE der gesteuerten Station einnimmt, das heißt, die PEAE kann keinen Datentransfer zu anderen Einheiten einleiten. Für den Havariefall ist eine Übergabe der Steuerfunktion an eine andere Station (K 1600, ursadat 5000) möglich.

5. Systemunterlagen

5.1. Systemunterlagen für K 1600

Das für die Mikrorechner K 1620/K 1630 vorhandene modulare Betriebssystem MOOS 1600 sichert schnelle Reaktion auf Prozeßereignisse und eine gut ausgebaute Organisation für die Abarbeitung von Nutzerprogrammen. Für den Anwender werden umfangreiche Mittel zum Herstellen und Testen von Anwenderprogrammen sowie Querschnitts-POS bereitgestellt. Die Organisation des Informationsaustausches mit der PEAE über ATP wird vom UK-Driver des MOOS 1600 Ausgabe 1.2 übernommen.

5.2. Systemunterlagen für PEAE ursadat 5000

Für die ursadat 5000 steht das Echtzeitsteuerprogramm EIEX 1521 mit all seinen Leistungen zur Verfügung. Es unterstützt über die Kanalsoftware alle ursadat-Moduln, den Informationsaustausch über das ZI und auch die DV-Peripherie. Das Steuerprogrammssystem wird entsprechend den Anwendungs-

forderungen generiert. Durch den modularen Aufbau des EIEX besteht die Möglichkeit, eigene Systemkomponenten zu generieren.

Weiterhin sind alle mathematischen Standardprogramme für denk 1520 anwendbar.

Zur Unterstützung der Meßwerterfassung und -verarbeitung auf unterster Ebene wird das generierbare Meßwerterfassungs- und Primärverarbeitungssystem MEPRM entwickelt [4].

Literatur:

- 1) Neugebauer, D.: Geräte für Grundkonfigurationen der Basisrechner des Kombinats Robotron. edv-aspekte 2(1983) 1, S. 5
- 2) Betriebsdokumentation M3krorechner K 1520
- 3) Bader B.: Hinweise zur Konfigurierung von Mikrorechner NTB 24 (1980) 6, S. 177
- 14) Lindner, A.: Programmsystem MEPRM für PRB A 6491 JA 6492. edv-aspekte 2 (1983) 1, S. 48
- 15) Kundendokumentation ursadae 5000
- 16) Branitz, W.: Mikrorechner K 1600 als Prozeßrechnersystem rechen-technik^datenverarbeitung 17 (1980) 9, S. 14



Software-Technologie für Mikrorechner

Von Dr.-ing. Erika Horn, Dr. Hans-Dieter Baumbach und Christian Straach,
104 Seiten, 37 Abbildungen, Broschur, 8'50f.
Ausland 10,40 M
Bestellangaben: 675 496 7/Softwaretechnologie

Die Zahl der Mikrorechner steigt ständig an, ihre Einsatzbreite nimmt rasch zu. Um so dringlicher wird die Notwendigkeit der Rationalisierung der Programmentwicklung für Mikrorechner. Mit dieser Veröffentlichung werden nun erstmalig die Methoden einer systematischen und rationellen Programmentwicklung — der „strukturierten Programmierung“ — für die Entwicklung der Mikrorechner-Software erschlossen. Die Autoren vermitteln sowohl grundlegende Kenntnisse auf dem Gebiet der Software-Technologie als auch konkrete praktische Kenntnisse für die ständig steigende Zahl von Mikrorechner-Programmierern in der DDR.

Die praxisorientierten Darlegungen berücksichtigen das erste in der DDR realisierte Software-Entwicklungssystem P05-Technologie K 1600.

Ihre Bestellung richten Sie bitte an den Verlag Die Wirtschaft, Abt. Vertrieb, 1055 Berlin, Am Friedrichshain 22.

Software

Das Steuerprogramm-System MOEX 1600 des Betriebssystems Moos 1600

Manfred Synowzik
VEB Robotron ZFT Dresden

Für die Basisrechner A 640x werden die Betriebssysteme MOOS 1600 und LAOS 1600 bereitgestellt. Das MOOS 1600 ist ein modular aufgebautes Betriebssystem, das mittels Generierung an die Fordernisse unterschiedlicher Einsatzgebiete angepaßt werden kann. Es realisiert Mehrnutzerbedienung und Multiprogrammierung und ist für die Lösung von Echtzeitaufgaben einsetzbar. Das Betriebssystem LAOS 1600 wird für die speziellen Aufgaben der Laborautomatisierung bereitgestellt. Es gewährleistet Vordergrund-/Hintergrundbetrieb und kann (im Vordergrundbetrieb) für Echtzeitaufgaben mit sehr kurzen Reaktionszeiten verwendet werden. Die beiden folgenden Beiträge geben eine Einführung in das Betriebssystem MOOS 1600.

1. Konzeption

MOEX 1600, Ausgabe 1.2. ist die Exekutive des Betriebssystems MOOS 1600 mit einer breiten Palette von Anwendungsgebieten. Sie eignet sich in gleichem Maße sowohl für den Einsatz zur Datenerfassung, Prozeßsteuerung sowie Labor- und Prüffeldautomatisierung als auch zur Lösung wissenschaftlich-technischer und ökonomischer Probleme. Es handelt sich um ein modular aufgebautes Steuerprogramm-System, das auf Grund seiner hohen Flexibilität optimal an die jeweiligen Einsatzbedingungen angepaßt werden kann. Bei der Konzeption spielten neben der schnellen Reaktionszeit auf Echtzeitanforderungen die Möglichkeiten der Multiprogrammierung und die Probleme der Dateiorganisation und des Dateischutzes eine besondere Rolle. Damit bietet MOEX 1600 alle Voraussetzungen sowohl zur Lösung von Echtzeitproblemen als auch zum Mehrnutzerbetrieb (Teilnehmerbetrieb). Ein leistungsfähiges Verarbeitungsprogramm für Indirektkommando-Dateien einschließlich der zugehörigen Anweisungen ermöglicht eine Stapelverarbeitung entsprechender Nutzerprogramme.

2. Systemleistungen

2.1. Konfiguration

MOEX 1600, Ausgabe 1.2. ist sowohl für die Arbeit mit dem K 1620 als auch für die Arbeit mit dem K 1630 vorgesehen. Es

wird vorausgesetzt, daß der Hauptspeicher nur aus Speicherbausteinen vom Typ RAM aufgebaut ist.

MOEX 1600 unterstützt gegenwärtig die Bedienung folgender Geräte:

- Bildschirm-E/A-Gerät als Bedieneinheit (K 8911)
- Kassettenplattenspeicher (CM 5400)
- Lochbandeinheit (K 6200)
- Kassettenmagnetbandeinheit (K 5261)
- Magnetbandgeräte (CM 5300, CM 5303)
- Seriendruker (SD 1152, SD 1156, SD 1157)
- Bildschirmterminals (K 8931, K 8912)
- Lochkartenleser (EC 6112)
- Folienspeichereinheit (K 5665)
- Paralleldrucker (VT 27065)
- Prozeßein-/ausgabereinrichtung (ursadat 5000)
- BUS-Koppeleinheit (K 4161).

Falls bezüglich des MGS K 1600 nicht andere Einschränkungen gelten, werden maximal 16 Geräte bzw. Controller eines Typs von MOEX 1600 unterstützt.

Als Minimalperipherie sind eine Bedieneinheit, eine Lochband- bzw. Kassettenmagnetbandeinheit sowie ein Plattenspeicher erforderlich. Die Mindestgröße des Hauptspeichers beträgt 16 K Worte bei einem System für den K 1620 und 18 K Worte bei einem System für den K 1630. Zur Minimalausrüstung gehört außerdem eine Echtzeituhr.

4.2. Speicherverwaltung

Eine Task, die abarbeitbare Programmeinheit, wird in einem zusammenhängenden Speicherbereich, der Partition, abgearbeitet. Eine Partition wird durch ihren Namen, ihren Typ, ihre Basisadresse und ihre Größe beschrieben. Zwischen Task und Partition bestehen Beziehungen, die von der Hard- und Softwareleistung des Systems abhängig sind.

Systeme mit Adreßzuweisung

Als Adreßzuweisung wird der Vorgang des Verbindens von Programmadressen (virtuelle Adresse) mit dem physischen Speicher bezeichnet. Die Adreßzuweisung wird hardwareseitig über die Speichervermittlungseinheit (SVE 30) realisiert, die nur beim K 1630 verfügbar ist. Die SVE 30 verbindet Adressen eines Programms (0 bis 32 K Worte) mit physischen Adressen (0 bis 124 K Worte). Die Tatsache, ob ein System mit oder ohne Adreßzuweisung ist, beeinflusst die Herstellung einer Task durch das entsprechende Systemprogramm.

In Systemen mit Adreßzuweisung beginnt jede Task mit der Adresse 0 (virtuelle Adresse). Eine Task kann in jeder Partition abgearbeitet werden, die groß genug ist, die Task aufzunehmen.

Systeme ohne Adreßzuweisung

In Systemen ohne Adreßzuweisung muß dem Task-Builder die Basisadresse der Partition übergeben werden, in der die Task abgearbeitet werden soll. Die Task kann nur in dieser Partition abgearbeitet werden.

Die nutzergesteuerte Partition

Die nutzergesteuerte Partition wird von Systemen mit und ohne Adreßzuweisung unterstützt. Sie kann zu einem Zeitpunkt jeweils nur eine Task enthalten. Um die nutzergesteuerte Parti-

tion effektiver nutzen zu können, besteht die Möglichkeit, sie in maximal sieben Subpartitions zu unterteilen. Jede Subpartition kann eine Task aufnehmen, so daß auf diese Weise sieben Tasks gleichzeitig in einer Partition untergebracht werden können. Eine derart organisierte Partition ermöglicht es, daß derselbe Speicherbereich im Wechsel von einer großen Task und maximal sieben kleinen Tasks belegt werden kann.

Die systemgesteuerte Partition

Die systemgesteuerte Partition wird nur von Systemen mit Adreßzuweisung unterstützt. Die Exekutive organisiert die Speicherbereichszuweisung so, daß so viele Tasks wie möglich gleichzeitig in einer Partition untergebracht werden können.

2.3. Programmorganisation

MOEX 1600 koordiniert die Abarbeitung aller zu bearbeitenden Tasks. Die Abarbeitung wird so organisiert, daß die Systemleistungen effektiv genutzt und kurze Antwortzeiten auf Echtzeitanforderungen erreicht werden. Obwohl jeweils nur eine Task die Steuerung der ZVE (Zentrale Verarbeitungseinheit) besitzen kann, ist Multiprogrammierung möglich, weil bei der Abarbeitung einer Task Wartereiten entstehen (r. B. Warten auf das Ende einer E/A-Operation), in denen andere Tasks bearbeitet werden können. Zur Realisierung dieser Arbeitsweise bedient sich die Exekutive einiger Mittel, die nachfolgend beschrieben werden.

Toxic status

Mittels Eingabe eines Kommandos über ein Bediengerät wird eine Task ins System eingeführt. Dieser Vorgang wird Installieren einer Task genannt. Beim Installieren registriert die **Exekutive** alle notwendigen Daten, um die Task auffinden und abarbeiten zu können. Eine installierte Task ist weder im Hauptspeicher resident, noch bewirbt sie sich um irgendwelche Systemleistungen. Die Task ist lediglich der Exekutive bekannt. MOEX 1600 kennt zwei Task-Zustände:

Eine ruhende Task ist eine installierte Task, für die noch keine Startanforderung vorliegt. Eine **aktive** Task ist eine installierte Task, für die eine Startanforderung vorliegt. Sie bleibt aktiv, bis sie beendet oder abgebrochen wird. Eine aktive Task bewirbt sich auf der Grundlage ihrer Priorität um ZVE-Zeit, sobald die von ihr benötigten Systemleistungen (r. B. genügend Hauptspeicher vorhanden oder E/A-Operation, auf die gewartet wurde, beendet) verfügbar sind — sie ist abarbeitungsfähig. Im anderen Falle wartet sie auf das Freiwerden der von ihr benötigten Systemleistungen — sie ist blockiert.

Für den Fall, daß in der Partition, in der eine Task installiert wurde, kein Platz ist, wird die Task, für die die Startanforderung vorliegt, in eine nach Priorität geordnete Warteschlange eingeordnet. Ein Unterschied zwischen ruhenden und aktiven Tasks besteht darin, daß eine ruhende Task weniger Hauptspeicher belegt als eine aktive. Wird eine Task aber benötigt, kann sie sehr schnell in den aktiven Zustand überführt werden, da die Exekutive alle notwendigen Parameter für diesen Vorgang bereits im Hauptspeicher verfügbar hat.

Priorität

Die Priorität einer Task, auf deren Grundlage sie sich um Systemleistungen bewirbt, wird durch eine ganze Zahl zwischen

1 und 250 verkörpert, wobei die größere Zahl die höhere Priorität bedeutet. Die Priorität kann einer Task während ihres Aufbaus vom Task-Builder, während des Installierens oder während der Abarbeitung zugeordnet werden. Die aktive Task mit der höchsten Priorität, die Zugriff zu allen Systemleistungen hat, die sie benötigt, wird bearbeitet. Wenn diese Task blockiert wird (z. B. Warten auf das Ende einer E/A-Übertragung), prüft die Exekutive, ob eine andere Task die ZVE benötigt. Für den Fall, daß mehrere aktive Tasks warten, kommt wieder diejenige zur Abarbeitung, die die höchste Priorität und den Zugriff zu allen von ihr benötigten Systemleistungen besitzt.

Auslagerung

Die Auslagerung bietet die Möglichkeit, Tasks abzuarbeiten, die sich nicht ständig im Hauptspeicher befinden. Wenn eine Partition eine Task beherbergt, die auslagerbar ist, wird diese auf die Platte übertragen, um Raum für eine Task höherer Priorität zu schaffen, die in derselben Partition installiert wurde und abarbeitungsfähig ist. Wenn die Task beendet ist, wird die vorher unterbrochene Task wieder eingelesen und an der Unterbrechungsstelle fortgesetzt.

Für diesen Vorgang muß vom Nutzer auf der Platte Speicherbereich zur Verfügung gestellt werden. Dieser Auslagerbereich wird der Task während ihres Aufbaus vom Task-Builder (statisch) oder während des Laufs durch die Exekutive (dynamisch) zugeordnet.

Die Realisierung sieht so aus, daß bei der statischen Zuordnung der Auslagerbereich direkt der Task zugeordnet wird. Der Auslagerbereich steht der Task also ständig zur Verfügung.

Bei der dynamischen Zuordnung kann der Nutzer eine oder mehrere Auslagerdateien auf Platten schaffen, um alle auszulagernden Tasks aufzunehmen. Die in diese Dateien auszulagernden Tasks müssen nicht als auslagerbare Tasks aufgebaut werden; die Eigenschaft der Auslagerbarkeit kann der Task beim Installieren zugeordnet werden. Die dynamische Zuordnung hat den Vorteil, daß der Platz auf dem Plattenspeicher ökonomischer genutzt wird. Nachteilig ist, daß der Fall eintreten kann, daß in den Auslagerdateien nicht immer freie Bereiche verfügbar sind.

Zeitscheibensteuerung

In Abhängigkeit vom Charakter der zu lösenden Aufgabe kann mittels der Priorität eine sinnvolle Steuerung der Verteilung der ZVE-Zeit erreicht werden. Zur Lösung verschiedener Aufgabenkomplexe ist die reine prioritätsgesteuerte Abarbeitung der Tasks nicht ausreichend, da auf diese Weise eine annähernd gleichberechtigte Bearbeitung verschiedener Teilprobleme vom Nutzer zuviel organisatorischen Aufwand erforderlich macht. Beim Installieren einer Task registriert die Exekutive diese in einer Systemtabelle, die nach der Priorität geordnet ist. Tasks gleicher Priorität werden in der Reihenfolge eingetragen, in der sie installiert wurden. Die Zuteilung von Rechenzeit an die speicherresidenten Tasks erfolgt entsprechend der Reihenfolge der beim Installieren erstellten Tabelle. Das hat zur Folge, daß bei Tasks gleicher Priorität die Tasks häufiger bearbeitet werden, die zuerst installiert wurden. Um diese Situation zu vermeiden, wird im MOEX 1600 eine Generierungs-Option „Zeitscheibensteuerung“ angeboten. Die Leistung dieser Zeitscheibensteuerung besteht darin, daß Tasks gleicher Priorität periodisch in der Systemtabelle gewechselt werden. Auf diese Weise gelangt jede Task gleicher Priorität zyklisch in die erste Position ihrer

Prioritätsgruppe in der Systemtabelle und damit zur Abarbeitung.

Zeitgesteuerte Auslagerung

Eine ähnliche Situation wie bei speicherresidenten Tasks gleicher Priorität kann sich ergeben, wenn sich aktive abarbeitungsfähige Tasks um gleiche Partitionsbereiche bewerben. Da eine Task die Exekutive nicht veranlassen kann, eine Task gleicher Priorität auszulagern, kann es dazu führen, daß einige Tasks keinen Hauptspeicherezugriff erhalten. Um diese Situation zu vermeiden, wird im MOEX 1600 eine Generierungsoption „zeitgesteuerte Auslagerung“ angeboten. Die Leistung dieser zeitgesteuerten Auslagerung besteht darin, daß Tasks zyklisch ausgelagert werden, um für sich mitbewerbende Tasks gleicher Priorität freie Partitionsbereiche zu schaffen. Die ausgelagerte Task wird am Ende der Schlange, in der die aktiven Tasks stehen, die sich um freien Partitionsbereich bewerben, eingereiht. Auf diese Weise gelangt jede aktive Task zyklisch an die erste Position ihrer Prioritätsgruppe in der Schlange der sich um gleiche Partitionsbereiche bewerbenden Tasks. Die auf diese Weise auszulagernden Tasks müssen als auslagerbare Tasks aufgebaut sein.

Signifikante Ereignisse

Signifikante Ereignisse werden Ereignisse genannt, die dazu führen, daß die Abarbeitung der laufenden Task unterbrochen und die Priorität aller aktiven Tasks erneut überprüft wird. Signifikante Ereignisse sind

- das Ende einer Taskbearbeitung.
- das Beenden einer Uhraktivität (Ablaufen eines Zählers)
- das Ausführen einiger Exekutive-Anweisungen
- die Ausführung eines Taktes der Zeitscheibensteuerung
- das Ende einer E/A-Operation.

Wenn ein signifikantes Ereignis eintritt, prüft die Exekutive die Schlange der aktiven Tasks und startet die Task mit der höchsten Priorität, die abarbeitungsfähig ist.

2.4. Exekutive-Anweisungen

7

Eine Exekutive-Anweisung ist die Anforderung einer Task an die Exekutive, eine bestimmte Operation auszuführen. Die Ausführung einiger Anweisungen führt zum Auftreten signifikanter Ereignisse. Über die Exekutive-Anweisungen haben Tasks Zugriff zu folgenden Leistungen:

- Unterbrechen und Fortsetzen der Taskbearbeitung
- Anfordern der Abarbeitung einer anderen Task
- Beenden einer Task
- Festlegen von Zeitbedingungen
- Beeinflussen der Speicherverwaltung
- Abfragen von Systeminformationen
- Austausch von Informationen zwischen den Tasks
- Ausführen von E/A-Operationen.

Darüber hinaus werden noch weitere Funktionen des Systems den Tasks über Exekutive-Anweisungen verfügbar gemacht. Die wesentlichsten sind die Nutzung von Ereignis-Flags, System-Traps sowie die Segmentierung.

Ereignis-Flagg

Neben dem Einfluß signifikanter Ereignisse auf die Bearbeitung der Tasks können diese genutzt werden, um Aktivitäten der

Tasks untereinander zu koordinieren. Mittels einer Exekutive-Anweisung lassen sich zum Beispiel signifikante Ereignisse mit Ereignis-Flags verknüpfen. Es stehen 64 Ereignis-Flags zur Verfügung, wobei 32 Flags von allen Tasks gemeinsam genutzt werden können, und 32 Flags stehen lokal jeder Task zur Verfügung. Die lokalen Flags können zum Beispiel so genutzt werden, daß sie mit signifikanten Ereignissen verknüpft werden. Die Exekutive setzt dann beim Eintreten des signifikanten Ereignisses das zugehörige Flag, das wiederum in der Task ausgetestet werden kann. Die gemeinsamen Flags werden zur Kommunikation zwischen den Tasks genutzt, um zum Beispiel gewisse Aktivitäten mehrerer Tasks untereinander zu koordinieren.

System-Traps

System-Traps sind ein weiteres Mittel zur Wechsel der Steuerung in der Taskbearbeitung. Eingeleitet werden System-Traps durch die Exekutive in Abhängigkeit vom Eintreten verschiedener Ereignisse. Auf diese Weise hat der Nutzer die Möglichkeit, unmittelbar auf Ereignisse zu reagieren. Zu diesem Zweck verknüpft er eine Task mit einem Ereignis. Die so verknüpfte Task enthält eine Routine, deren Eintrittspunkt der Exekutive über eine Anweisung mitgeteilt werden muß. Der System-Trap überträgt dann die Steuerung an die vom Nutzer geschriebene Routine. Bei den System-Traps unterscheiden wir zwei Arten:

Der *synchrone System-Trap* (SST) zeigt Ereignisse an, welche direkt mit der Ausführung von Befehlen verbunden sind, also immer an gleicher Stelle des Programms auftreten, sooft dieses Programm durchlaufen wird. Ein nicht zugelassener Adreßmodus in einem Befehl erzeugt zum Beispiel solch einen SST.

Der *asynchrone System-Trap* (AST) zeigt signifikante Ereignisse an, wobei der Zeitpunkt ihres Auftretens nicht direkt vor Programm beeinflussbar ist. Zum Beispiel kann das Ende einer E/A-Operation solch einen AST erzeugen.

Segmentierung

Auf Grund der 16-Bit-Adressierungstechnik des K 1600 kann eine Task maximal 32 K Worte direkt adressieren. Falls eine Task mehr als 32 K Worte benötigt, kann mittels Auflösen der Tasks in einzelne Segmente eine Überlagerungsstruktur realisiert werden, die bereits beim Taskaufbau definiert wird. Solch eine Überlagerungsstruktur besteht aus einem Wurzelsegment, das ständig im Hauptspeicher stehen muß und Überlagerungssegmenten, die bei Bedarf in den Hauptspeicher eingelesen werden.

Eine andere Möglichkeit zur Überschreitung der 32-K-Wortgrenze für eine Task bieten die Exekutive-Anweisungen zur Speicherverwaltung. Diese Anweisungen setzen eine Hardware-Unterstützung voraus, die nur beim K 1630 vorhanden ist.

In diesem Falle können alle Segmente einer Task gleichzeitig im Hauptspeicher stehen — es erfolgt lediglich ein Wechsel in der Zuordnung der aktuellen physischen Adressen.

Es werden drei Arten von Adreßräumen unterschieden:

Physischer Adreßraum

Der physische Adreßraum umfaßt den gesamten physischen Speicher, in dem Tasks abgearbeitet werden können. Der physische Adreßraum kann maximal 124 K Worte umfassen.

Logischer Adreßraum

Der logische Adreßraum ist der gesamte physische Adreßraum, zu dem eine Task Zugriff hat.

Virtueller Adreßraum

Der virtuelle Adreßraum entspricht dem mit einer 16-Bit-Adresse direkt adressierbaren Bereich von 32 K Worten. Durch Benutzung der Speicherverwaltungsanweisungen kann eine Task ihre virtuellen Adressen dynamisch zu verschiedenen Teilen ihres logischen Adreßraumes zuordnen und damit ihren logischen Adreßraum über 32 K Worte hinaus ausdehnen.

Die Exekutive-Anweisungen zur Speicherverwaltung ermöglichen es also einer Task, ihren logischen Adreßraum dynamisch zu verändern. Auf diese Weise kann eine Task ihre virtuellen Adressen auch zu logischen Bereichen einer anderen Task zuordnen. Damit besteht die Möglichkeit, daß verschiedene Tasks gemeinsame Daten- und Programmbereiche (Unterprogramme) nutzen können. Die organisatorischen Größen, mit denen die Exekutive arbeitet, sind das Adreßfenster für den virtuellen Adreßraum und die Region für den logischen Adreßraum einer Task.

2.5. Gerätebedienung

Das Bedienen der Peripherie kann unter MOEX 1600 mittels E/A-Driver oder Nutzertask realisiert werden. Für die Standardperipherie werden E/A-Driver als generierbare Bestandteile des MOEX 1600 zur Verfügung gestellt. Der Nutzer hat auch die Möglichkeit, für spezielle periphere Geräte eigene EWA-Driver anzuschließen. Diese Möglichkeit ist besonders für Echtzeitanwendungen von Bedeutung. Um den Speicherplatz für verfügbare, aber selten gebrauchte E/A-Driver nicht ständig zu blockieren, können bei der Generierung einige E/A-Driver so ins System integriert werden, daß sie nur bei Bedarf (mittels Kommando) in den Hauptspeicher geladen werden.

Eine andere Möglichkeit besteht darin, daß die Nutzertask selbst die Bedienung des peripheren Gerätes übernimmt. Zu diesem Zweck unterstützt MOEX 1600 den Anschluß von Nutzertasks an die hardwareseitig vorhandenen Interrupt-Vektoren der Geräte.

2.6. Dateiorganisation

Zur Verwaltung von Dateien auf Magnetplatten und Magnetbändern stellt MOEX 1600 einen Satz von Routinen zur Verfügung. Datenträger, zu denen zugegriffen werden soll, müssen besonders formatisiert sein, das heißt mittels Kommando initialisiert werden. Nutzertasks können jedoch auch zu fremden (nicht wie gefordert formatisierten) Datenträgern zugreifen. Jeder Nutzer kann sich mittels eines Kommandos eine Datei aufbauen, die vom System als Verzeichnis für die Dateien des Nutzers genutzt wird. Beim Erstellen einer Datei wird der Dateiname in dieses Nutzerdateiverzeichnis (UFD) eingetragen und ein Nutzeridentifikationscode (UIC) im Kennsatz der Datei registriert. Der UIC hat das Format [g, m], wobei g und m Oktalzahlen sind, die die Nutzergruppe (g) bzw. die Mitgliedsnummer (m) verkörpern. Da das UFD selbst eine Datei ist, besitzt es ebenfalls einen UIC. In den meisten Fällen stimmen der UIC des UFD und der UIC der Dateien, die in ihm geführt werden, überein. Eine Datei kann aber auch in einem UFD geführt werden, dessen UIC vom UIC der Datei abweicht, so daß eine Datei gleichzeitig in mehreren UFD geführt werden kann. Alle UFD eines Datenträgers sind in einem Hauptdateiverzeichnis zusammengefaßt. Sowohl das Nutzerdateiverzeichnis als auch das

Hauptdateiverzeichnis enthalten jeweils die Namen der Dateien und die Zeiger zu den Dateikennsätzen. In den Dateikennsätzen befinden sich unter anderem die Informationen über die physische Lage der Dateien auf dem Datenträger.

Ein Nutzer, der zu einer Datei per Kommando bzw. Makro angreifen will, muß in einer Dateispezifikation auch das Gerät, auf dem sich der Datenträger befindet und das UFD, in welchem die Datei geführt wird, angeben. Diese Angaben sind Voraussetzung für das Auffinden der Datei. Jeder, der zu einer Datei zugreifen will, muß demzufolge das UFD der Datei kennen (der Name des UFD wird vom UIC des UFD abgeleitet). Die Kenntnis des UFD ist eine notwendige, aber nicht hinreichende Voraussetzung für den Zugriff zu einer Datei. Um den Schutz zu gewährleisten, wird den Dateien eines Datenträgers eine Schutzmaske zugeordnet. Der Nutzer, der zu einer Datei zugreifen will, muß den Bedingungen dieser Schutzmaske genügen. Die Schutzmaske legt für die Nutzergruppen System, Eigentümer, Gruppe und Rest vier Typen des Zugriffs fest. Diese sind Lesen, Schreiben, Erweitern und Löschen.

2.7. Bedienung

Ein Operator kann mittels Kommandos, die über ein Bediengerät eingegeben werden, mit MOEX 1600 verkehren. Einige Kommandos werden als Überlagerungssegmente einer Task, andere als selbständige Tasks realisiert.

Der Operator hat Zugriff zu folgenden Leistungen:

- Steuerung der Taskabarbeitung
- Abfordern von System- und Taskinformationen
- Verwaltung der peripheren Geräte.

Die Kommandos zur Steuerung der Taskabarbeitung sind besonders wichtig, denn mit ihrer Hilfe kann der Operator aktiv in die Verteilung der ZVE-Zeit an die einzelnen Tasks eingreifen. Er hat die Möglichkeit, die Zahl der aktiven Tasks zu beeinflussen, indem er neben der direkten Aktivierung von Tasks die Möglichkeiten der zeitgesteuerten Aktivierung (Absolutzeit und nach Zeitdifferenz, einmalig oder zyklisch) nutzt.

Zur direkten Beeinflussung einiger Systemleistungen stehen dem Operator privilegierte Kommandos zur Verfügung. Privilegierte Kommandos können für von privilegierten Bediengeräten eingegeben werden. Der Privilegiestatus (privilegiert, nicht privilegiert) eines Bediengerätes wird im allgemeinen bei der Systemgenerierung festgelegt. Nach abgeschlossener Generierung kann der Operator per Kommando über ein privilegiertes Bediengerät den Privilegiestatus anderer Bediengeräte verändern.

MOEX 1600 enthält außerdem eine Generierungsoption für Mehrnutzerunterstützung, mit deren Hilfe die einzelnen Nutzer überwacht und gesteuert werden. Ein Systemverwalter weist jedem Nutzer einen Identifikationscode zu, welcher den Privilegiestatus des Nutzers festlegt. Bevor ein Nutzer bei Systemen mit Mehrnutzerunterstützung Kommandos eingeben darf, muß er sich auf einem Bediengerät anmelden. Bei der Anmeldung gibt der Nutzer direkt oder indirekt seinen UIC sowie ein Kennwort an. Das System überprüft, ob UIC und Kennwort zusammengehören und versetzt bei positivem Ausgang der Prüfung das Bediengerät in den Privilegiestatus des Nutzers (privilegiert oder nicht privilegiert). Durch diese Option kommen verschiedene Kommandos hinzu, mit deren Hilfe im wesentlichen die Leistungen der Geräteverwaltung erweitert werden. Es können zum Beispiel Geräte zu öffentlichen und privaten Geräten erklärt werden.

Bei öffentlichen Geräten haben alle Nutzer gleiche Zugriffsrechte, bei privaten Geräten sind die Zugriffsrechte stark differenziert.

Um den Nutzer von der Eingabe sich häufig wiederholender Kommandofolgen an entlasten, unterstützt MOEX 1600 die Verarbeitung von Indirektkommando-Dateien.

Eine Indirektkommando-Datei ist eine Datei, die eine Folge von Kommandos enthält, die an eine spezielle Task gerichtet sind. Für Kommandos an das Kommandoprogramm des MOEX 1600 existiert ein leistungsfähiges Verarbeitungsprogramm für Indirektkommando-Dateien. Spezielle Sprachelemente (Anweisungen an das Verarbeitungsprogramm für Indirektkommando-Dateien), zusammen mit den Kommandos der Exekutive, ermöglichen die Organisation einer flexiblen Stapelverarbeitung.

2.8. Wartungsunterstützung

MOEX 1600 beinhaltet als generierbare Option eine Fehlerregistrierung. Diese Komponente überwacht die Zuverlässigkeit der Hardware. Es werden ständig Informationen über die erkannten Fehler aufgezeichnet – unabhängig davon, ob es sich um einen behebbaren oder nicht behebbaren Fehler handelt.

Der Nutzer kann nun in wählbaren Intervallen diese Task aktivieren, die ihm die gesammelten Informationen ausgibt.

Eine weitere generierbare Option des MOEX 1600 ist die Unterstützung von Diagnose-Tasks. Eine Diagnose-Task testet ein angegebenes Gerät, um die Ursache einer Störung festzustellen und protokolliert die Ergebnisse dieser Untersuchung. Diagnose-Tasks in Verbindung mit der Fehlerregistrierung eingesetzt, können die Systemausfallzeiten wesentlich reduzieren, da auf diese Weise Geräteausfälle frühzeitig erkannt und rechtzeitig behoben werden können. Die Diagnose-Tasks werden bei Bedarf vom Bediener gestartet.

2.9. Speicherschutz

Aufbauend auf die Leistung der Speicherverwaltungseinheit des K 1630 realisiert MOEX 1600 einen umfassenden Speicherschutz. So wird exekutivseitig jeder Task nur der Speicherzugriff erlaubt, zu dem sie irgendwelche Zugriffsrechte besitzt. Einer Task wird nur der Zugriff zu Bereichen, die Bestandteil der Task selbst sind oder die als Common-Bereiche gekennzeichnet sind, gewährt, MOEX bietet weiterhin einen Satz von Exekutive-Anweisungen, mit deren Hilfe eine Task den Speicherbereich, zu dem sie Zugriff hat, dynamisch verändern kann. Unter den Bedingungen des Mehrnutzerbetriebes bzw. der Multiprogrammierung können die Zugriffsrechte sinnvoll differenziert werden (Lesen, Schreiben, Erweitern, Löschen).

2.11. Fiestartunterstützung

MOEX 1600 schließt eine generierbare Restartunterstützung nach Netzausfall ein. Bei Netzausfall werden sämtliche flüchtigen Registerinhalte in den Hauptspeicher gerettet, um eine sinnvolle Weiterarbeit nach Netzwiederkehr zu sichern.

Bei Netzwiederkehr erfolgt gerätetechnisch eine vom eingesetzten Hauptspeicher abhängige Anfangseinstellung. Bei Speichern ohne Datenerhalt wird die Exekutive automatisch neu eingelesen.

Bei Speichern mit Datenerhalt stellt die Exekutive den gerette-

ten Zustand wieder her und bereitet die Abarbeitung aller EJA-Driver, die bei Netzausfall aktiv waren, mit ihrem Netzausfall-Eintrittspunkt vor. Anschließend prüft die Exekutive, ob Nutzer-Tasks über den eingetretenen Netzausfall informiert werden müssen. Nutzertasks haben die Möglichkeit, mittels einer Exekutive-Anweisung einen asynchronen System-Trag bei Netzwiederkehr anzufordern. Ist das der Fall, leitet die Exekutive die AST für alle Tasks ein, die einen AST angefordert hatten. Ist die Restartunterstützung nicht generiert worden, geht MOEX 1600 bei Netzausfall in den Halt-Zustand. Bei Netzwiederkehr wird in diesem Fall in Abhängigkeit vom eingesetzten Speicher das System neu eingelesen oder die ZVE in den Halt-Zustand versetzt.

2.11. Speicherbedarf

Die Minimalvariante der Exekutive benötigt 6 K Worte Hauptspeicher für den hauptspeicherresidenten Teil sowie eine 2-K-Partition zur Aufnahme der Komponenten, die als Tasks organisiert sind. Der maximale Bedarf ist praktisch nur für den hauptspeicherresidenten Teil (Exekutivkern) angebar — er liegt bei 20 K Worten. Der Hauptspeicherbedarf der Komponenten, die als Tasks organisiert sind, ist davon abhängig, ob die Tasks in einer Partition untergebracht oder auf mehrere Partitions verteilt werden; die speziellen Einsatzbedingungen (Zeitverhalten) bzw. Möglichkeiten (verfügbarer Hauptspeicher) spielen also eine große Rolle.

2.12. Generierung

Die Generierung der Exekutive wird entsprechend den speziellen Anforderungen des Anwenders vorgenommen. Sie erfolgt in drei Abschnitten und wird über die Generierungsangaben gesteuert.

Im 1. Abschnitt wird anhand der Forderungen des Anwenders hinsichtlich der gewünschten Leistung der Exekutive und der gerätetechnischen Voraussetzungen der hauptspeicherresidente Teil der Exekutive aus Quellmodul zusammengesetzt; bedingt assembliert und in eine Objektbibliothek überführt.

Im 2. Abschnitt werden die Objektbibliothek der Exekutive in eine Maschinencode-Datei überführt, die vom Anwender benötigten und als Task organisierten Teile der Exekutive (u. a. das Kommandoprogramm) als Maschinencode-Datei aufgebaut und mit dem hauptspeicherresidenten Teil verknüpft sowie die Partitionsfestlegungen getroffen.

Im 3. Abschnitt werden die Systemprogramme als Maschinencode-Dateien aufgebaut.

Der Ablauf der Generierung wird durch Indirektkommando-Dateien gesteuert. Ein Hilfsmittel zur Generierung ist außerdem das virtuelle Kommandoprogramm VMR 1600 (MOEX). Es dient dazu, in einem MOEX-Abbild auf einer Platte benötigte Anfangszustände und Anfangswerte festzulegen. Diese sind dann nach Neueinlesen des MOEX 1600 von der Platte der aktuelle Systemzustand.

Die Abschnitte 2 und 3 werden vom Anwender selbst ausgeführt, während den Abschnitt 1 die Vertriebsbetriebe des VEB Kombinat Robotron übernehmen.

Die Systemprogramme des Betriebssystems MOOS 1600

Michael Gnädig
VEB Robotron-ZFT Dresden

1. Dienstprogramme

1.1. Allgemeines

Dienstprogramme dienen gemeinsam mit den Übersetzern und Interpretern der Programmentwicklung, der Ausführung spezieller Systemverwaltungsfunktionen, der Arbeit mit Dateien sowie der Unterstützung der Ein-/Ausgabe. Sie arbeiten unter Steuerung der Exekutive.

Das Zusammenwirken verschiedener Dienstprogramme zur Herstellung eines lauffähigen Programms zeigt Abb. 1 am Beispiel eines in der Assemblersprache MACRO 1600 geschriebenen Programms.

1.2. Lochbandeingabe PT' 1600 (MOEX)

Aufgaben

Das Lochbandeingabeprogramm PTI (Paper Tape Input) dient zum Übertragen vom Lochbandleser auf Platten-dateien.

Arbeitsweise

PTI verfügt über eine geblockte Lochbandeingabe mit Wechselpufferarbeit. Dadurch hängt die Eingabegeschwindigkeit im wesentlichen von der Geschwindigkeit des Lochbandlesers ab. Die erzeugten Plattendateien sind sequentielle Dateien mit Sätzen variabler Länge.

Die maximale Satzlänge beträgt 132 Byte; längere Sätze werden geteilt.

Eingelesen werden Lochbänder gerader Parität nach Kodetabelle KO1-7H0, ST RGW 356.76.

1.3. Editor EDI 1600 (MOEX)

Aufgaben

Mit EDI können Dateien (Quellprogramme oder anderes Textmaterial) aufbereitet oder über ein Terminal operativ eingegeben werden. Dazu verfügt EDI über Kommandos, mit denen einzelne Zeilen oder Gruppen von Zeilen von einer Eingabedatei in einen Hauptspeicherpuffer gelesen, dort bearbeitet und anschließend zu einer neuen Ausgabedatei geschrieben werden. Der Editor enthält Kommandos zur zeilen- und zur zeichenweisen Aufbereitung, zur Listenausgabe und zum Öffnen und Schließen von Dateien.

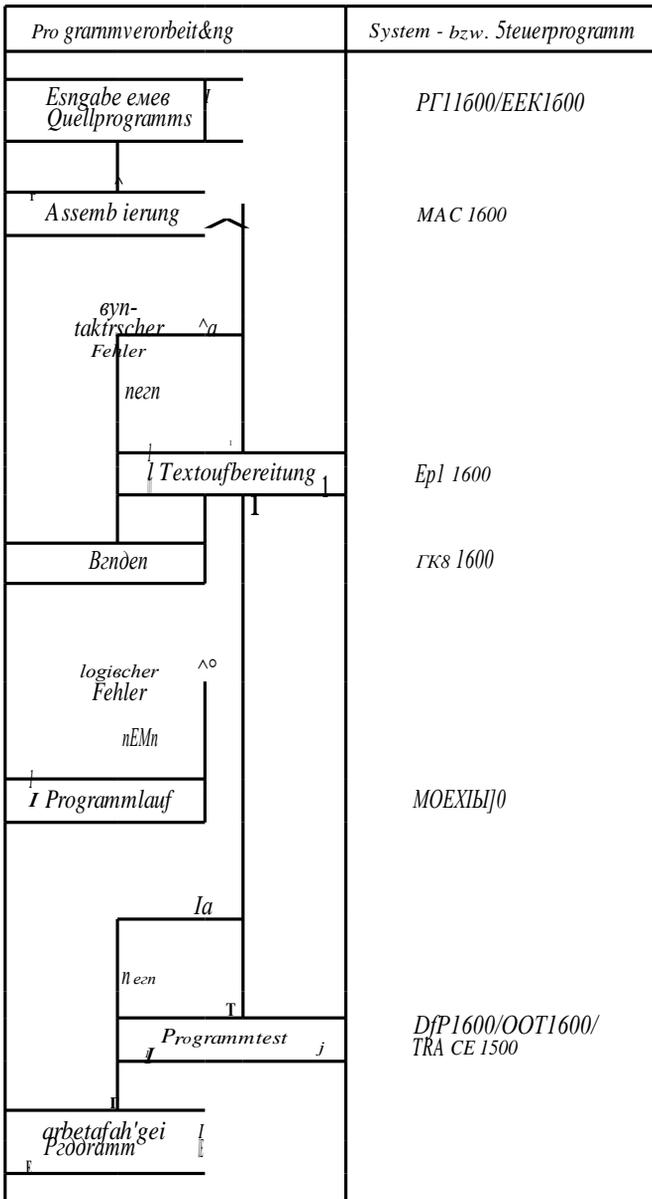


Abb. 1 Beispiel des Zusammenwirkens von Dienstprogrammen des MOOS 1600

Arbeitsweise

Der Editor kann in zwei Modi arbeiten, dem Blockmodus oder dem Zeile-für-Zeile-Modus. Während der Laufzeit des Editors können die Modi beliebig gewechselt werden.

Im *Blockmodus* wird von der Eingabedatei eine bestimmte Anzahl von Zeilen (Block) in einen Hauptspeicherpuffer gelesen. Die Anzahl der Zeilen kann über ein Editorkommando beliebig festgelegt werden. Die standardzeilenzahl je Block ist generierbar. Enthält die Eingabedatei Form-Feed-Zeichen, wird der Puffer nur bis zum Erreichen des Zeichens Form-Feed gefüllt. Damit ist eine seitenweise Aufbereitung möglich, Innerhalb eines Blockes können die zeilen- und zeichenorientierten Kommandos beliebig benutzt werden. Es ist Vor- und Rückpositionieren möglich.

Im *Zeile-für-Zeile-Modus* wird nur jeweils eine Zeile der Eingabedatei in den Hauptspeicherpuffer gelesen, wo sie mit zeilen- oder zeichenorientierten Kommandos aufbereitet werden kann. Beim Vorpositionieren auf die nächste zu bearbeitende

Zeile werden alle vorhergehenden Zeilen auf die Ausgabedatei geschrieben. Eine Rückpositionierung ist demzufolge nicht möglich.

Arthur der helm Editorstart spezifizierter' Eingabedatei (Primäre Eingabedatei) kann während der Laufzeit des Editors über besondere Kommandos eine weitere Eingabedatei (Sekundäre Eingabedatei) eröffnet und zur Datenübernahme ausgewählt werden. Mittels Schließen der ausgewählten sekundären Eingabedatei und Öffnen einer anderen Datei als sekundäre Eingabedatei besteht die Möglichkeit des Zusammensetzens einer Datei aus beliebig vielen Eingabedateien. Der Editor ist besonders für die interaktive Arbeit im Dialogbetrieb geeignet. Darüber hinaus bietet er auch Möglichkeiten der vorprogrammierten Arbeitsweise. Lange Kommandos oder Kommandoketten, die in einem speziellen Fall häufig benutzt werden, können in Makros definiert werden.

Im Rahmen der Möglichkeiten des Betriebssystems arbeitet der Editor Kommandoketten ab, die vor dem Editorstart von einem peripheren Datenträger eingelesen wurden.

1.4. Task Builder TKB 1600 (MOEX)

Aufgaben

Der Task-BUILDER TKB verarbeitet einen oder mehrere Objektmoduln zu einem abarbeitbaren Programm, die Task, und lagert ein speichergetreues Abbild, die Task-Abbilddatei, zusammen mit Informationen über die Task auf die Platte aus.

Arbeitsweise

Ausgangsdaten für die Task sind die vom Assembler oder von Compilern erzeugten Objektmoduln. Zusätzlich wird die Taskbildung durch entsprechende TKB-Kommandos und die Überlagerungsbeschreibung bestimmt.

Neben der Bildung der Task kann sich der Nutzer über den TKB eine Task-Zuordnungsdatei mit der Cross-Referenz aller im Task benutzten globalen Symbole und eine Symboltabelle erstellen lassen. Die Task-Zuordnungsdatei gibt über die in der Task verwendeten Objektmoduln, die Speicherbelegung und über die Task-Eigenschaften Auskunft. Der TKB greift auf die Objektmoduln unter dem Namen zu, unter dem sie vom Compiler bzw. Assembler erzeugt und auf die Platte ausgelagert wurden. Ein Zugriff ist auch auf vorher erstellte Objektbibliotheken möglich. Das erzeugte Task-Abbild wird als Plattendatei unter einem Namen ausgelagert. Die Speicherzuordnungsdatei legt sich entweder sofort ausdrucken oder als Datei auf die Platte bringen. Die Symboldefinitionsdatei wird in einem internen Format erstellt und kann ebenfalls unter einem Namen auf der Platte abgespeichert werden.

Es besteht die Möglichkeit, Task-Eigenschaften anzugeben, die zur Ausführungszeit wirksam werden, sonst gelten Standardannahmen. Mit der direkten Angabe solcher Eigenschaften sind folgende Modifizierungen möglich:

- Beeinflussen der Stellung der Task im Betriebssystem
- besondere Zugriffsrechte zu geschätzten Datenbereichen
- Auslagerbarkeit bei Unterbrechungen
- Verhalten im Fehler- und Programmabbruchfall
- Verwendung bestimmter Hardware-Eigenschaften des Systems
- Anfügen einer Testhilfe

- Deklarieren der Task als Kommandobereich bzw. UP-Bibliothek
- Bilden der Task für Systeme mit oder ohne Speichervermittlungseinheit (SVE)
- Größe des Kellerbereichs
- Taskpriorität
- Festlegen der Taskpartition
- Gerätezuordnungen für die Task.

Die benötigten Kommandos können sowohl direkt als auch über eine Datei (indirekte Kommandodatei) bereitgestellt werden. Mit dem Verbinden selbständig übersetzter Programmmoduln über globale Symbole und durch die Segmentierung ermöglicht der TKB eine sehr übersichtliche und hauptspeichersparende Strukturierung der Task.

Mit Hilfe der Oberlagerungsbeschreibungssprache wird die modulare Baumstruktur der Task definiert. Ein Oberlagerungsbaum besteht im allgemeinen aus einem Wurzelsegment, das sich während des Programmlaufes ständig im Hauptspeicher befindet, und weiteren, sich entweder überlagernden oder verketteten, also parallel im Speicher befindlichen Segmenten, die nur bei Bedarf in dafür reservierte Bereiche geladen werden. Ein Segment besteht wiederum aus einem oder mehreren Programmmoduln. Die Oberlagerungsbeschreibung kann sowohl direkt als auch über Datei bereitgestellt werden.

1.5. Testprogramme DEP 1600 (MOEX), DDT 1600 (INDEX) und TRACE 1600 (MOEX)

Aufgaben

Die Testprogramme DEP (Debugging Program), ODT (On-line Debugging Tool) und TRACE dienen zum Testen von Anwenderprogrammen. Das Programm muß assembliert sein und wird mit Hilfe des TKB zusammen mit einem der Testprogramme zu einer Task verbunden.

Arbeitsweise

• DEP 1600 (MOEX)

Als Startadresse der Task wird vom TKB die Startadresse des DEP vereinbart, so daß sich nach dem Start der Task das DEP mit der Kommandoroutine meldet.

Ober Bedienerkommunikation können folgende Leistungen des DEP angefordert werden:

- Ausgabe von Speicherplatz- und Registerinhalten in mehreren Formaten
- Eingabe in Speicherplätze und Register im Format Oktal (Wort- und Bytemodus)
- Suchen nach Bitmustern (Worten) im Speicher und nach Befehlen, deren Adressen auf eine speziell definierte Adresse Bezug nehmen
- Direkte und interpretative Abarbeitung des Anwenderprogramms mit Protokollierung.

DEP besteht aus vier Hauptmoduln:

- Kommunikation und Kommandoentschlüsselung
- Kommandoprozessoren
- Interpretation und Protokollierung
- Dienstrountinen.

• ODT 1600 (MOEX)

Das Testprogramm ODT ist von Leistungsumfang her eine Untermenge von DEP.

ODT ist mit DEP im Modul „Direkttest“ identisch. ODT ermöglicht keine Befehlsinterpretation und Protokodierung der Befehlsausführung und belegt nur etwa 30 Prozent des Speicherplatzes von DEP. Damit ist es besonders in solchen Fällen von Bedeutung, wo DEP nicht mehr im Internspeicher Platz findet.

• TRACE 1600 (MOEX)

TRACE ist das kleinste Testhilfsmittel. Es benötigt etwa 325 Worte im Speicher.

TRACE besitzt keine eigene Kommunikation mit dem Bediener. Es erhält die von ihm benötigten Informationen vom TKB. Mittels einer TKB-Option können eine Programmanfangsadresse und ein bis vier Protokollbereiche definiert werden. TRACE gibt nach jeder Befehlsausführung innerhalb eines Protokollbereiches Informationen über eine Reihe von Registern aus.

1.6. Programme zur Dateiverwaltung

1.6.1. Dateizugriffsroutinen 1600 des MOOS 1600

Die Dateizugriffsroutinen FCS (file control service) realisieren das Erstellen von Dateien und ihre Verarbeitung im sequentiellen und direkten Zugriff. Programmtechnisch ist der Dateizugriff realisiert mittels Objektmoduln, die in der Systemobjektbibliothek enthalten sind und als Unterprogramme im jeweiligen Anwenderprogramm aufgerufen werden. Die Notation der Aufrufe ist durch Makros nutzerfreundlich gestaltet.

Nutzerseitig besteht die Datei aus einem Dateikennsatz, der die Attribute der Datei und andere Informationen über die Datei enthält, und den logischen Sätzen als eigentliche Informationseinheiten. Diese sind für das System nur durch ihre Länge charakterisiert, ihre innere Struktur ist Sache der Anwenderprogramme. MOOS 1600 läßt Dateien mit Sätzen variabler Länge und Dateien mit Sätzen fester Länge zu. Im allgemeinen ist sequentieller Zugriff möglich. Zu Plattendateien mit Sätzen fester Länge ist über die Satznummer auch direkter Zugriff möglich. Spezifischen Bedürfnissen entsprechend kann über im Anwenderprogramm geretteten Satzpositionen auch zu ausgewählten Sätzen in Plattendateien mit Sätzen variabler Länge direkt zugegriffen werden. Die Dateien werden über die sogenannte Dateispezifikation problemorientiert angesprochen. Sie beinhaltet Dateispezifikation, die Angaben des Gerätes, des Nutzeridentifikationskodes, des Dateinamens, des Dateityps und der Versionsnummer.

Ober den Nutzeridentifikationskode und die Schutzmaske von Datenträgern und Dateien werden differenzierte Zugriffsrechte festgelegt, und zwar in jeweils 4 Nutzerklassen (System, Eigentümer, Gruppe des Eigentümers, alle übrigen) mit jeweils 4 Personen des Zugriffs (Lesen, Schreiben, Erweitern, Löschen). Der Zugriff zu den logischen Sätzen als nutzerseitigen Informationseinheiten wird von FCS organisiert. MOEX 1600 greift direkt zu logischen Blöcken zu, die zum Beispiel für Platte eine feste Länge haben. Für Nutzer, welche die Einordnung der Sätze in Blöcke (Blocken/Entblocken) durch FCS nicht nutzen wollen, bietet FCS auch die Möglichkeit, unmittelbar zu den logischen Blöcken der Datei zuzugreifen.

Als Träger von Dateien im eigentlichen Sinne bietet MOOS 1600 Platten (logisch äquivalent sind Disketten) und Magnetbänder (logisch nicht äquivalent sind die Kassettenmagnetbänder!). Darüber hinaus kann analog einer Datei sequentiell auf Drucker geschrieben oder am Terminal gelesen und geschöpft werden.

Die Arbeit mit Dateien auf Lochband oder Kassettenmagnetband ist möglich, wird jedoch vom System logisch nur über die entsprechenden Systemprogramme P11 bzw. FLX unterstützt. Über das Systemprogramm FEX wird die Umsetzung von MOO5-Dateien in ESER-Dateien (Magnetband) realisiert.

1.6.2. Datei-Transferprogramm PIP 1600 (MOEX)

Aufgaben

Die Hauptaufgabe des Datei-Transferprogramms PIP (Peripheral interchange Program) besteht darin, Dateien zu transportieren. Der Datentransport kann zwischen verschiedenen Geräten (z. B. Platte — Magnetband) oder auf dem gleichen Gerät (z. B. von einem Plattenbereich auf einen anderen) erfolgen. Funktionen von PIP, die Datentransport ausführen, sind beispielsweise:

- Kopieren von Dateien
- Fügen von Dateien (Bildung einer neuen Datei aus mehreren existierenden Dateien)
- Überschreiben einer existierenden Datei mit einer neuen
- Erweitern einer existierenden Datei.

Des Weiteren erfüllt PIP auch Aufgaben ohne Datentransport, unter anderem:

- Löschen von Dateien
- Ausgabe von Dateiverzeichnissen
- Änderung von Elementen der Dateispezifikation () Änderung des Dateinamens, der Zugriffsrechte, des Standardgerätes oder des Standard-UFO)
- Löschen bzw. Eintragen von Dateispezifikationen in ein Dateiverzeichnis.

Arbeitsweise

Die Funktionen von PIP werden über Kommandos aufgerufen. Diese enthalten auch die notwendigen Dateispezifikationen für den Zugriff zu den Dateien. Die Verwendung von Standardparametern ermöglicht eine Verkürzung der Kommandozeile.

1.7. Bibliothekar Lail 1600 (MOEX)

Aufgaben

Der Bibliothekar LBR (Librarian Utility) führt folgende Funktionen aus:

- Erzeugen von Objekt- bzw. Makrobibliotheksdateien
- Löschen von einzelnen Modulen aus den Bibliotheken
- Verdichten einer durch vorangegangene Löschoptionen nur noch teilweise ausgenutzten Bibliotheksdatei
- Einstellen von Standards (Makro- oder Objektbibliothek)
- Löschen bzw. Einfügen von Eintrittspunkten aus Objektbibliotheken
- Herauslösen einzelner Module aus beliebigen Bibliotheken
- Einfügen von Modulen in Bibliotheken
- Austauschen von Modulen gleichen Namens aus einer Bibliothek
- Kürzen von Makrobibliotheken, das heißt Streichen von Kommentartexten.

Arbeitsweise

Das Ausführen der Funktionen wird über Kommandos realisiert. Diese lassen auch Kombinationen der Funktionen zu.

Der Anwender stellt die Bibliotheken nach seiner Wahl zusammen und ermöglicht so einen schnellen Zugriff durch den Task-Builder und den Assembler.

Bibliotheken sind Direktzugriffsdateien, die einen oder mehrere Module gleichen Typs (Objekt- oder Assemblerprogramme) enthalten.

1.8. Programme zur Wartung und Pflege von Datenträgern

Im Rahmen des Betriebssystems MOOS 1600 gibt es noch weitere Dienstprogramme für Datenrettung, Datenübertragung, Prüfung von Dateien, Korrektur von Dateien, Vergleich von Dateien sowie Wartung und Pflege der Datenträger.

Mit Hilfe des *Plattenkopierprogramms* DSC können Platten sowohl von Platte zu Platte als auch von Platte zu Magnetband kopiert werden. Außerdem kann mit diesem Programm eine Platte verdichtet werden, so daß anschließend zusammenhängender Platz für die Aufnahme weiterer Programme zur Verfügung steht.

Mit den Programmen FLX und FEX ist eine *Fannatumwandlung* möglich. Sie bieten die Möglichkeit der Übertragung zwischen Datenträgern in den Dateiformaten des MOOS 1600, des LAOS 1600, für Datensicherung und des ESER. Dabei werden von ihnen eine Reihe peripherer Geräte, zum Beispiel Magnetbandgerät oder Folienspeicher bedient.

Das Programm CMP dient dem *Vergleich* zweier Eingabedateien im KOI-7-Kode. Dabei auftretende Differenzen werden in einer Ausgabedatei bereitgestellt.

5011 eine Platte dahingehend überprüft werden, ob sie lesbare bzw. richtig zugeordnete Dateien enthält, ist das mit VFY möglich. Das Suchen fehlerhafter Blöcke der Platte dagegen geschieht mit BAD.

2. Übersetzer und Interpreter

2.1. Makroassembler MAC 1600 (MOEX)

Aufgaben

Der Makroassembler MAC 1600 (MOEX) übersetzt Quellprogramme, die in der Assemblersprache MACRO 1600 geschrieben sind und erzeugt Objektcode (der vom Task-Builder verarbeitet wird) sowie Übersetzungslisten.

Arbeitsweise

Der Makroassembler arbeitet in zwei Durchläufen. Neben den zu übersetzenden Quelldateien können auch Makrobibliotheksdateien angewiesen werden. Die Systemmakrobibliothek wird von MAC 1600 automatisch berücksichtigt.

Im Übersetzungskommando können spezielle Funktionen angewiesen werden, zum Beispiel zur Übersetzungssteuerung oder zum Umfang der Übersetzungsliste.

Es besteht die Möglichkeit der Ausgabe einer Cross-Referenz-Hate.

Sprachmerkmale

Die Assemblersprache MACRO 1600 kennzeichnen außer dem üblichen Sprachumfang, der aus sämtlichen Maschinenbefehlen, der Benutzung symbolischer Adressen und Darstellungen für die Adressierungsarten besteht, folgende Eigenschaften:

- MACRO 1600 gestattet eine übersichtliche Programmierweise mittels lokaler Symbole mit begrenztem Gültigkeitsbereich.
- Mit Hilfe globaler Symbole können Verbindungen zwischen verschiedenen, unabhängig voneinander übersetzten Quellprogrammen hergestellt werden, die beim Bilden der Task verbunden werden sollen. In jedem Quellprogramm können globale Symbole definiert und/oder benutzt werden.
- MACRO 1600 ermöglicht das Benutzen von Unterprogrammen. Es können interne Unterprogramme als Bestandteil des zu übersetzenden Quellprogramms sowie externe Unterprogramme, die in einer Unterprogramm-bibliothek enthalten sind, abgearbeitet werden.
- Es stehen Anweisungen zur Verfügung, die die Definition von Daten in allen vorhandenen Datenarten innerhalb des Programms gestatten.
- Es gibt Anweisungen zur Definition von Programmsektionen; dadurch werden die logischen Programmeinheiten bereitgestellt, die beim Bilden der Task verarbeitet werden, zum Beispiel zur Herstellung von Überlagerungsstrukturen.
- Es besteht die Möglichkeit der bedingten Übersetzung. Bestimmte Teile des Quellprogramms können in Abhängigkeit von zu testenden Bedingungen wahlweise übersetzt werden. Damit können aus einem Quellprogramm bzw. aus einem einheitlichen Paket von Quellprogrammen über Steuergrößen unterschiedliche (Objekt-) Programmversionen erzeugt werden.
- In der Sprache MACRO 1600 ist eine umfangreiche Makrotechnik enthalten, die die Definition und Benutzung von Makros (Makrodefinitionen und Makroanweisungen) gestattet. Befehlsfolgen, die in gleicher oder ähnlicher Form in einem oder mehreren Quellprogrammen oft wiederkehren, können durch Makros ersetzt werden. Es können interne Makros als Bestandteil des zu übersetzenden Quellprogramms sowie externe Makros (die in einer Makrobibliothek enthalten sind) benutzt werden.

2.2. CDL-1600-Programmiersystem

CDL 1600 ist eine problemorientierte Programmiersprache, die speziell für Belange der Systemprogramm-entwicklung geschaffen wurde.

Gegenüber der Assemblerprogrammierung ergeben sich folgende Vorteile:

- Durch die Anhebung des Programmier-niveaus wird Entwicklungszeit eingespart (bis zu 50 Prozent).
- CDL-1600-Programme sind zuverlässiger, wartungsfreundlicher und verfügen über ein hohes Maß an Selbstdokumentation.
- Die Fehlersuche wird einfacher und erfordert wesentlich weniger Rechenzeit-Aufwand.
- Die Portabilität ganzer Programmsysteme wird möglich; bis zu 80 Prozent des Leistungsumfanges können auf Vorgänger- oder Wirtsrechnern getestet werden.

Einsatzgebiet von CDL 1600 ist vorrangig die Systemprogrammierung, jedoch auch die Prozedursteuerung und die Anwenderprogrammierung. Das CDL-1600-Programmiersystem ist eine leistungsfähige Implementierung mit folgenden Merkmalen:

- Optimierung der Zielprogramme
- umfassende Fehlerdiagnose und andere Serviceleistungen

- Laufzeit-Testhilfe
- einfache Bedienung.

Sprachmerkmale

- strukturierte Ausdrucksmittel
- CDL 1600 orientiert auf die Prinzipien der strukturierten Programmierung und unterstützt den Top-down-Entwurf von Programmen. Daraus resultiert:

- eine Erhöhung der Lesbarkeit der Programme
 - eine Einschränkung der Fehlerhäufigkeit
 - eine Verbesserung der Zuverlässigkeit und Wartungsfreundlichkeit der Programme
 - Modulkonzept
- CDL-1600-Programme sind modular aufgebaut. Die Module werden getrennt übersetzt. Sie kommunizieren über exakt definierte Schnittstellen.
- Prozedurkonzept
- Die Formulierung von Algorithmen erfolgt in CDL 1600 in Form von Prozeduren. Dazu werden problemorientierte Ausdrucksmittel zur Organisation der Ablaufsteuerung von Algorithmen bereitgestellt. mit deren Hilfe Rufe von Prozeduren und Makros verknüpft werden können.

- Makrokonzept
- Makros sind Anweisungsfolgen in der Assemblersprache MACRO 1600. Beim Aufruf von Makros werden diese Anweisungsfolgen in das Zielprogramm expandiert. Damit bietet CDL 1600 die Möglichkeit, alle Ausdrucksmittel der Assemblersprache zu nutzen. Die Vorteile sind:

- Die Ausnutzung aller Hardware- und Betriebssystemeigenschaften wird möglich.
- Der Programmierer hat einen wesentlichen Einfluß auf die Effektivität seiner Zielprogramme.

- Bibliotheken
- Es ist eine variable Arbeit mit verschiedenen Makrobibliotheken möglich:

- Der Inhalt der Makrobibliotheken kann an verschiedene Anwendungsfälle angepaßt werden. Damit können für verschiedene Nutzerklassen problemangepaßte Standarddefinitionen und -operationen geschaffen werden.
- Die Komplexität der in der Makrobibliothek enthaltenen Standarddefinitionen beeinflußt maßgeblich das Programmier-niveau.

Componentmerkmale

Das CDL-1600-Programmiersystem besteht aus einem

- CDL-Compiler
- Bibliothekscompiler und
- sehr kleinen Laufzeitsystem.

Weiterhin wird eine Standard-Bibliothek zur Verfügung gestellt. Die Zielsprache von CDL 1600 ist die Assemblersprache MACRO 1600.

Optimierung

Die Zielprogramme können bezüglich Laufzeit bzw. Speicherplatz wahlweise optimiert werden. Damit werden insbesondere Un-effektivitäten, die in den strukturierten Ausdrucksmitteln ihre Ursache haben, beseitigt. Optimierte Zielprogramme sind in der Effektivität mit Assemblerprogrammen vergleichbar: die

Laufzeit ist etwa gleich, die Speicherplatzverlängerung liegt im Bereich von 1,2-1,4.

Fehlerdiagnose

Die Compiler erkennen eine Vielzahl von Fehlern und weisen diese in der Liste positionsgenau mit ausführlicher Charakterisierung der Fehlerursache aus. Fehler führen zum Abbruch der Zielcodegenerierung. Weiterhin existieren Warnungen, die auf Situationen aufmerksam machen, die mit hoher Wahrscheinlichkeit auf algorithmische Fehler hindeuten. Es erfolgt kein Abbruch der Zielcodegenerierung.

Serviceleistungen

CDL-1600-Programme können in eine übersichtliche und gut lesbare Form transformiert werden (Strukturanalyse). Dabei wird eine Liste mit genauer Kommentierung der Prozedurabarbeitung erzeugt. Weiterhin ist die Anfertigung von Cross-Referenzlisten möglich.

Laufzeit-Testhilfe

Mit Hilfe einer Laufzeit-Testhilfe (TRACE-Routine) Wird die Protokollierung auswahlbarer Programmabschnitte möglich.

Bedienung

Die Bedienung des CDL-1600-Programmiersystems ist vollständig an die Bedienung anderer Systemkomponenten angepaßt. Es wird eine variable Steuerung der Listenausgabe, der Serviceleistungen und der Bibliotheksarbeit ermöglicht.

2.3. COBOL-1600-Programmiersysteme

Die Programmiersprache COBOL ist eine höhere Programmiersprache, die vorwiegend für ökonomische Anwendungen im betriebswirtschaftlichen und kaufmännischen Bereich vorgesehen ist. Mit der Implementierung der Programmiersprache COBOL, die nicht ohne Grund die international am weitesten verbreitete und angewendete höhere Programmiersprache ist, wird die Anwendung des Mikrorechnergerätesystems robotron K 1600 im ökonomischen Bereich unterstützt und ein höheres programmiersprachliches Niveau zur rationellen Erarbeitung entsprechender EDV-Projekte ermöglicht.

Die COBOL-Compiler übersetzen COBOL-Programme in die Assemblersprache MACRO 1600. Nach der Assemblierung der erzeugten Assembl erprogramme und deren Verbindung mit einem COBOL-Laufzeitsystem durch den TKB entsteht ciini abarbeitbares Programm.

Arbeitsweise

Der COBOL-1620-Compiler (COB 1620) übersetzt COBOL-1620-Programme in einem Durchlauf und erzeugt ein aus zwei Programmsegmenten bestehendes Assemblerprogramm. Der COBOL-1630-Compiler (COB 1630) übersetzt COBOL-1630-Programme in drei Durchläufen in ein Assemblerprogramm.

Die vom Compiler geführten Tabellen werden in einem virtuellen Speicher verwaltet. Während des gesamten Übersetzungslaufes erfolgt eine ausführliche Fehlerdiagnose, wobei detaillierte Fehlermeldungen erzeugt werden, die als Bestandteil der Übersetzungsliste oder

als getrennte Fehlerliste dem Nutzer ausgegeben werden. Zusätzlich kann der Nutzer die Ausgabe der Crossliste anweisen. Während von COB 1 620 erzeugte Zielprogramme durchgehend interpretativ abgearbeitet werden, werden die von COB 1630 erzeugten Zielprogramme interpretativ und direkt abgearbeitet. Für COBOL-1630-Sprachkonstruktionen, die relativ häufig auftreten und relativ einfach (über die Maschinenbefehle des K 1630 realisierbar sind, wird ein Zielcode erzeugt, der direkt abgearbeitet wird. Das Abarbeiten der COBOL-Zielprogramme erfordert die Verfügbarkeit der Satzzugriffsroutinen RMS, die Bestandteile der POS für K 1600 sind//.

Sprachmerkmale

w COBOL 1620

Die dem COBOL-Compiler COB 1620 zugrunde liegende Sprachversion realisiert im wesentlichen die Stufe 1 des ANSI-COBOL-Standards von 1974 einschließlich der Verarbeitung von Direktzugriffsdateien. Sie ermöglicht den Aufruf von FORTRAN- und Assembl erunterprogrammen. Bezüglich der Quellsprache besteht Kompatibilität zu COBOL 1520.

- COBOL 1630

Die durch den COBOL-1630-Compiler für den K 1630 realisierte Sprachversion COBOL 1630 ist entsprechend den Festlegungen des international anerkannten ANSI-COBOL-Standards von 1974 definiert worden.

Bezüglich der Quellsprache besteht Aufwärtskompatibilität von COBOL 1620 zu COBOL 1630.

Die Programmiersprache COBOL 1630 besitzt im Vergleich zum ANSI-COBOL-Standard von 1974 folgenden Leistungsumfang:

- Sprachkern (NUCLEUS) auf Stufe 2
- sequentielle Dateiverarbeitung auf Stufe 2
- relative Dateiverarbeitung auf Stufe 2
- indexsequentielle Dateiverarbeitung auf Stufe 2
- Tabellenverarbeitung auf Stufe 1
- Segmentierung auf Stufe 2
- Testuntersetzung auf Stufe 1
- Quellprogrammbibliothek auf Stufe 1
- Unterprogrammtechnik auf Stufe 1.

(Die Stufe 2 entspricht der höchsten Stufe des jeweiligen Moduls.)

2.4. BASIC-Programmiersystem BAS 1600 (MOEX)

BASIC 1600 ist eine problemorientierte Programmiersprache zur Lösung wissenschaftlich-technischer Aufgaben.

BASIC 1600 zeichnet sich vor allem durch folgende *Eigenschaften* aus:

- Einfache Erlernbarkeit
- Unterstützung der interaktiven Arbeit
- Aufruf von Codeunterprogrammen
- Interpretative Abarbeitung von BASIC-Programmen
- Erweiterung des ANSI-Standards für BASIC
- Kompatibilität zwischen BASIC 1520 und BASIC 1600.

Sprachmerkmale

- Einfache Erlernbarkeit

BASIC (Beginners All-purpose Symbolic Instruction Code) ist eine der am einfachsten zu erlernenden Programmiersprachen. BASIC-Anweisungen setzen sich aus englischsprachigen Schlüs-

selworten und Ausdrucken in ub ihher Formelschreibweise zusammen.

- Unterstützung der interaktiven Arbeit

BASIC-Programme können unsortiert aber das Terminal eingegeben werden. Die Eingabezeilen werden syntaktisch überprüft und entsprechend ihrer Zeilennummer einsortiert. Syntaktisch falsche Zeilen werden abgewiesen. Das eingegebene Programm kann ohne weitere Zwischenschritte gestartet werden. Verläuft der Programmablauf nicht wie gewünscht, kann das eingegebene Programm editiert und sofort wieder gestartet werden.

- Aufruf von Codeunterprogrammen

Es besteht die Möglichkeit des Aufrufes von Assembler-Unterprogrammen. Die Unterprogramme müssen vorher in den BASIC-Interpreter eingebunden werden.

- Interpretative Abarbeitung von BASIC-Programmen

Die eingegebenen BASIC-Programme werden interpretativ abgearbeitet. Die sonst üblichen Routinearbeitsgänge Editieren (Übersetzen-Taskbildern-Testen vereinfachen sich. Sofort nach der Eingabe oder Änderung eines Programms kann dieses gestartet werden. Es ist außerdem möglich, kleine Berechnungen auszuführen, ohne ein Programm zu schreiben. Die nötigen Anweisungen werden ohne Zeilennummer eingegeben und von BASIC sofort ausgeführt (Direktmodus oder Taschenrechnermodus)- Die Arbeit im Direktmodus bietet zugleich eine

geeigneten Stellen unterbrochen werden. Variablen im Direktmodus ausgegeben oder verändert werden und der Programmablauf durch eine GOTO-Direktanweisung fortgesetzt wird.

- Erweiterung des ANSI-Standards für BASIC

BASIC 1600 enthält neben den im Minimalstandard für BASIC (ANSI x 3.60-1978) definierten Sprachelementen Erweiterungen zur Verarbeitung von Zeichenketten, zur Verarbeitung sequentieller Dateien, zur formatierten Ausgabe von Daten und zur Überlagerung von BASIC-Programmen. Diese Erweiterungen realisiert BASIC 1600 mit folgenden sprachlichen Mitteln:

Zeichenkettenverarbeitung :

- Zeichenkettenvariable für Zeichenketten von 0 ... 255 Zeichen Länge
- Zeichenkettenfelder
- Verkettung von Zeichenketten
- Standardfunktionen zum Suchen von Teilzeichenreihen; zum Ausschnitt von Teilzeichenreihen usw.

Verarbeitung sequentieller Dateien:

- Eröffnung von Ein- und Ausgabedateien mit OPEN-Anweisung
- Schließen von Dateien mit CLOSE-Anweisung
- Ein- oder Ausgabe mit INPUT #- bzw. PRINT #-Anweisung
- Löschen von Dateien mit KILL-Anweisung
- Umbenennen von Dateien mit NAME-Anweisung

Formatierte Ausgabe von Daten:

- Ausgabe von Daten mit der PRINT-USING-Anweisung

Überlagerung von BASIC-Programmen:

- Mit der CHAIN- bzw. OVERLAY-Anweisung können Programme in Abschnitten abgearbeitet werden. Die COMMON-Anweisung gestattet die gemeinsame Benutzung von Variablen durch mehrere Abschnitte.

- Kompatibilität zwischen BASIC 1520 und BASIC 1600
BASIC-Programme, die vom Dialog-BASIC-Interpreter DBIN 1520 (MEOS) verarbeitet werden, können auch mittels BASIC-

Interpreter des K 1620 abgearbeitet werden, wenn sie keine K-i 520 spezifischen Anweisungen zur Prozeßverarbeitung enthalten.

2.5. FORTRAN-Programmiersystem FOR 1600 (MOEX)

FORTRAN 1600 basiert auf dem FORTRAN-IV-Standard-Sprachumfang, hat aber viele effektive Erweiterungen und kann fast alle Systemressourcen nutzen. Dadurch ist diese Sprache nicht nur zur Lösung wissenschaftlich-technischer Aufgaben nutzbar, sondern sie eignet sich auch hervorragend für Echtzeitanwendungen. Das FORTRAN-System besteht aus dem Compiler und dem Laufzeitsystem oTS. Besondere Kennzeichen sind:

- Einfache Kommunikation mit dem Compiler
- effektive Codegenerierung
- Optimierung des generierten Codes
- nutzerfreundliche Ein-/Ausgabeanweisungen
- umfangreiche Fehlerdiagnose bei der Compilierung und Programmausführung
- leistungsfähiges Paket von Systemunterprogrammen
- Segmentierung von FORTRAN-Programmen beim Taskbildern
- Benutzung speicherresidenter Bibliotheken von FORTRAN Programmen.

Sprachumfang

FORTRAN 1600 entspricht den Festlegungen von ANSI FORTRAN X3.9-1966. Folgende Erweiterungen zum Standard wurden aufgenommen:

- In Ausdrücken können alle Datentypen einschließlich COMPLEX vermischt auftreten.
- Als Feldindex kann jeder arithmetische Ausdruck verwendet werden.
- Bis zu 7-dimensionale Felder sind zulässig.
- Zeichenketten in Apostroph können an Stelle von Hollerith-Konstanten verwendet werden.
- Bei DO-Anweisungen sind für Anfangswert, Endwert und Schrittweite Ausdrücke angebar. Der Wert der Schrittweite kann negativ sein.
Der Steuerparameter in berechneten GOTO-Anweisungen kann ein Ausdruck sein.
- Die Markenliste in der zugewiesenen GOTO-Anweisung ist wahlweise.
- Zusätzliche Ein-/Ausgabeanweisungen
- Standardformat bei Ein-/Ausgaben
- END- und ERR-Konstruktionen zur Spezifikation von Dateiende und Fehlerbedingung.
- In der Liste der Ausgabeanweisungen WRITE, TYPE und PRINT sind Ausdrücke angebar.
- Externe Feldbegrenzungen bei der Dateneingabe zum (7berlesen der Formatspezifikation.
- Listengesteuerte Datenübertragung zur Ausgabe von Daten im Standardformat oder zur Eingabe feldbegrenzter Daten.
Geräteorientierte Anweisungen ACCEPT, TYPE, PRINT.
Unformatierte E/A-Anweisungen mit direktem Speicherzugriff zum Lesen oder Schreiben von Dateien in beliebigem Format.
Konvertierungen von Speicher zu Speicher ENCODE.
DECODE.

Dateisteuerung OPEN, CLOSE.

Die logischen Operationen .AND., OR., .NOT., .XOR. und .EQV. können auf Integer-Daten angewendet werden, um Bitmanipulationen auszuführen.

IMPLICIT-Vereinbarung zur impliziten Typvereinbarung von Namen.

Zusätzliche Datentypen BYTE und LOGICAL 1 zur Speicherung von Zeichen oder kleiner Bitwerte.

Testanweisungen, gekennzeichnet durch ein D in Spalte 1, können in Abhängigkeit vom Compilerschalter bei der Übersetzung als Anweisungen oder Kommentar gewertet werden.

Jede Anweisung kann auf der gleichen Zeile nach einem Ausrufezeichen (!) kommentiert werden.

FOR TRAN-Compiler

Der FORTRAN-Compiler liest eine FORTRAN-1600-Quelldatei und erzeugt auf der Ausgabeseite eine Objektdatei und eine Listendatei. Die zu einem vollständigen FORTRAN-Programm gehörenden Objektmoduln, benötigte Bibliotheksunterprogramme und die Moduln des Laufzeitsystems OTS müssen mit dem Taskbuilder zu einem abarbeitbaren Task verbunden werden.

Der Compiler ist durch extrem kurze Übersetzungszeiten charakterisiert. Er arbeitet bereits in einer Partition von 8 K Byte und kann dann nach Programmeinheiten von etwa 500 Quellprogrammzeilen verarbeiten. Trotz seiner hohen Übersetzungsgeschwindigkeit und des geringen Speicherbedarfs führt der Compiler verschiedene Optimierungen des generierten Objektcodes durch. So werden redundante Ausdrücke eliminiert, Konstanten-Ausdrücke zusammengefasst, Verzweigungsstrukturen optimiert und verschiedene Index-Optimierungen ausgeführt. Die Kommunikation mit dem Compiler entspricht der allgemeinen Form der Systemprogramme im Betriebssystem. Durch Angabe von Schaltern können bestimmte Übersetzungsfunktionen gesteuert werden. Das sind beispielsweise die Auswahl der gewünschten Listenformaten, die Übersetzung von Testzeilen, verschiedene Optimierungen hinsichtlich Speicherplatz und Rechenzeit, die Vektorisierung von Feldern und die Erläuterung von Warnungen.

Laufzeitsystem DT5

Das Laufzeitsystem enthält die zur Ausführung von FORTRAN-Programmen benötigten Routinen. Es kann Bestandteil der Systembibliothek sein. Der Compiler generiert in der Objektdatei Informationen, welche Moduln der OTS den Ablauf des FORTRAN-Programms erzeugen. Der Taskbuilder sucht dann die benötigten Routinen aus der das OTS enthaltenden Bibliothek und verbindet sie zu einem abarbeitbaren Programm.

Fehlerdiagnose

Ein umfassendes System der Fehlerdiagnose unterstützt den FORTRAN-Programmierer sehr wirkungsvoll bei der Entwicklung fehlerfreier Programme.

Bereits während der Übersetzung signalisiert der Compiler ausführlich Programmierfehler. Etwa 100 Fehlerarten können unterschieden werden. Zusätzlich warnt der Compiler vor solchen Programmelementen, die bei der Abarbeitung zu Fehlern führen können. Während der Laufzeit kontrolliert ein Fehlerprogramm die Programmabarbeitung. Dieses System führt eine Fehlerverfolgung durch und lokalisiert die aktuelle Pro-

grammeinheit und die Anweisungsnummer des verursachenden Fehlers. Bei Unterprogrammen wird der Fehler bis in die verursachende Programmeinheit rückverfolgt und eine Liste der Modulnamen und der entsprechenden Zeilennummern ausgegeben.

Zusätzlich kann der Programmierer bereits während der Programmierung Anweisungen zur Fehlersuche und Protokollierung in das Programm einfügen. Diese Anweisungen, mit einem D in Spalte J, übersetzt der Compiler wahlweise als Kommentar oder Anweisungen in Abhängigkeit von der Kommandoeingabe. Mittels Ausgabe ausgewählter Variablen oder PAUSE-Anweisungen kann die Programmabarbeitung verfolgt werden oder diese Anweisungen werden im fehlerfreien Programm als Kommentar gewertet.

Benutzung von Bibliotheken

Die Systembibliothek enthält das Laufzeitsystem OT5 und einen Komplex von Unterprogrammen, der den Zugriff zu fast allen Systemleistungen ermöglicht. Der Nutzer kann sich auch eigene Bibliotheksdateien zu Unterprogrammen in FORTRAN- oder Assemblercode bilden; so kann sich das OTS auch in einer extra Bibliothek befinden. Diese Bibliotheken lassen sich beim Taskbilden mit einbeziehen. FORTRAN-Programme können auch mit Hauptspeicherresidenten Bibliotheksmoduln verbunden werden.

2.6. PASCAL-Programmiersystem PAS 1600 (MOEX)

Die Programmiersprache PASCAL ist kompakt, leicht erlernbar, fördert das strukturierte Programmieren und gestattet die Entwicklung betriebssicherer, effektiver sowie maschinenunabhängiger Programme. PASCAL eignet sich auf Grund der Einfachheit seiner Konzepte und des geringen Umfangs besonders als Lehrsprache.

Der Leistungsbereich von PASCAL überdeckt aber auch die Belange von Problemen der Systemprogrammierung; ebenso kann sie für wissenschaftlich-technische sowie ökonomische Aufgaben verwendet werden.

PASCAL ist eine Weiterentwicklung von ALGOL und enthält moderne Sprachkonzepte der 70er Jahre.

Diese Eigenschaften werden vom Programmiersystem weitgehend unterstützt.

Sprachumfang

Der PASCAL-Compiler für MOOS 1600 verarbeitet PASCAL-Programme entsprechend des B51jISO Working Draft/3 für Standard-PASCAL. Die realisierte Sprache ist bis auf Maschinenabhängigkeiten (Wertebereich, Genauigkeit von REAL-Daten, Zeichensatz) kompatibel zu PASCAL für OS/ES.

PASCAL für MOOS 1600 besitzt des weiteren folgende Merkmale:

- Der Standardtype CHAR umfaßt den vollständigen ASCII-Code.
- Es sind die im Standardvorschlag angegebenen Standardfunktionen und -prozeduren realisiert.
- Es bestehen Anschlußmöglichkeiten für FORTRAN- und Assembler-Routinen sowie für getrennt übersetzte PASCAL-funktionen oder -prozeduren.

Der Compiler

Der PASCAL-Compiler arbeitet in mehreren Schritten. Im ersten Schritt wird ein Zwischencode (P-Code) erzeugt. Dieser Zwischencode kann in einem folgenden Schritt durch Interpretation abgearbeitet werden oder mittels Assern in Objektcode überführt werden. Der Objektcode wird vom Task-Builder zu einem ausführbaren Programm weiterverarbeitet. Dabei können Moduln aus einer FORTRAN- oder PASCAL

Eine Optimierung erfolgt- indem überflüssige Lade- und Speicherbefehle beseitigt werden (Peephole-Optimierung). Verschiedene Compilerfunktionen (Ausgaben und Dienstleistungen) können aber Kommando gesteuert werden. Für Dokumentationszwecke bzw. zur Fehlersuche gibt der Compiler umfangreiche Informationen aus:

- Quelltext mit P-Code-Instruktionszähler
- Fehlerkennzeichnung im Quelltext
- Markierung des Beginns und Endes von RECORD-Definitionen, CASE-Anweisungen und Verbundanweisungen
- Auflistung der generierten P-Code-Instruktionen
- Ein Symbolnachweis kann von einem Zusatzprogramm erzeugt werden.

Der Compiler arbeitet unter der Steuerung des Betriebssystems MOOS 1600. Er belegt eine Partition von mindestens 20 K Worten.

Laufzeitsystem

Das Laufzeitsystem steuert die Ausführung der übersetzten PASCAL-Programme und führt eine Diagnose von Laufzeitfehlern aus.

Das ESA-System benutzt die sequentielle Zugriffsroutinen und Routinen für die E/A über Terminal (Datenendplatz).

Fehlerdiagnose

Während der Compilierung und Ausführung wird eine umfangreiche Fehlerdiagnose ausgeführt. Der PASCAL-Compiler für MOOS 1600 nutzt die durch die Sprache gegebenen Möglichkeiten für die statische Überprüfung von Kontextabhängigkeiten (Typverträglichkeit) weitgehend.

Über den DEBUG-Schalter können zusätzliche Tests auf Einhaltung des Wertebereiches bei Aufzählungs- und Teilbereichstypen und der Indexgrenzen eines ARRAY-Typen bei Indexausdrücken während der Ausführung eines Programms gefordert werden. Bei Programmabbruch wird wahlweise ein Post-Mortem-Dump ausgegeben.

Literatur

1/ syrmg, Rn Pos für die Basisrechner des Kombinats Robotron. edv-aspekte 2 (1983) t, S. 39

system zur softwareentwicklung POST 1600

Wilfried Klette
VEB Robotron-Vertrieb Berlin

1. Einsatzgebiet der POST 1600

Mit der POS-Technologie POST 1600 werden Methoden und Mittel zur systematischen und rationellen Programmerstellung für die Entwicklung von Mikrorechner-Software bereitgestellt.

POST 1600 ist ein technologisches System zur Softwareentwicklung für die Basisrechnersysteme robotron A 640x, das aus methodischen und programmtechnischen Mitteln besteht (Abb. 1).

Die methodischen Mittel der POST 1600 sind anwendungstechnisch aufbereitete Methoden und Verfahren, die vor allem der technologischen Unterstützung des Entwurfsprozesses und dem Testen der Systemunterlagen dienen. Die programmtechnischen Mittel sind zu einem *Technologischen Programmsystem zur software-Entwicklung* (TESO 1600) zusammengefaßt. TESO 1600 unterstützt die technologischen Prozesse Implementieren und Testen sowie die dazugehörigen Nebenprozesse Dokumentieren, Verwalten und Kontrollieren.

Das Zusammenspiel der einzelnen Prozesse zeigt Abb. 2. Der SOS-Entwicklungsprozeß wird nach Tätigkeiten in die Hauptprozesse

- Entwerfen
- Implementieren
- Testen

und die Nebenprozesse

- Dokumentieren
- Verwalten
- Kontrollieren

unterteilt.

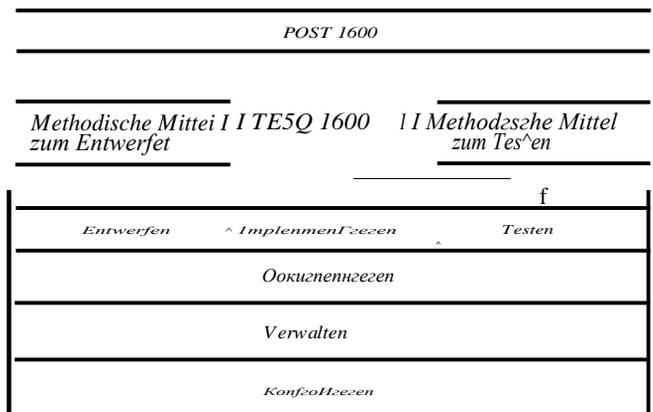
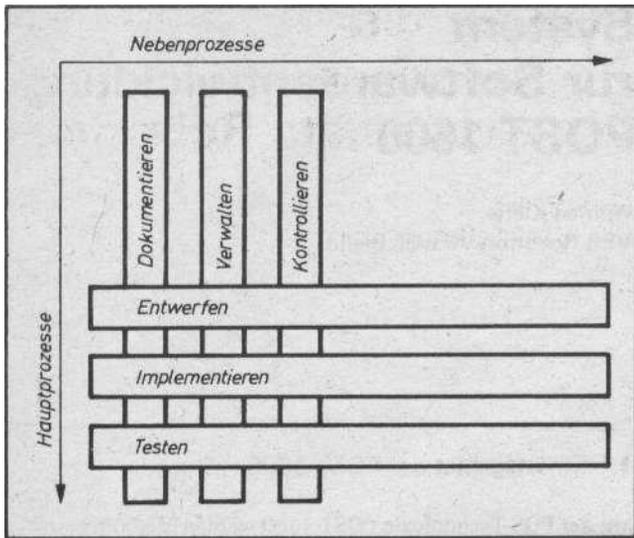


Abb. 1 Übersicht POST 1600



АЪ. 2 Наир[- und НеЪепртгпевве дер ПП5-Entwicklung

Die Hauptprozesse laufen zeitlich nacheinander ab, wobei sie von den Nebenprozessen durchdrungen und überlappt werden.

2. Technologische Prozesse der PQ-Entwicklung

Entwerfen

Ausgehend von einer gegebenen Ziel- und Aufgabenstellung ist in zwei Stufen eine Problemlösung zu erarbeiten.

1. Stufe:

Problemorientiertes Entwerfen

Als Ergebnis entsteht ein problemorientierter Systementwurf mit Angabe

- der Problemstruktur
- von Teilalgorithmen
- der Datenstruktur
- der Ablaufstruktur.

2. Stufe:

Programmierorientiertes Entwerfen

Als Ergebnis entsteht ein programmorientierter Systementwurf mit Angabe

- der Struktur als Hierarchie programmtechnischer Einheiten einschließlich Ablaufsteuerung
- der Datenbeschreibung.

Implementieren

Ausgehend vom programmorientierten Systementwurf wird im Hauptprozeß Implementieren ablauffähige P05 erzeugt.

Drei Hauptphasen werden durchlaufen:

- Kodieren des Entwurfs
- Übersetzen der Quellprogramme in Objektform
- Verbinden der Objektmoduln.

Testen

Im Hauptprozeß Testen werden die implementierten ablauffähigen P05 mittels Ablauftest auf Erfüllung der Ziel- und Aufgabenstellung in Verbindung mit dem konkreten Betriebssystem MOOS 1600 und dem jeweiligen Basisrechner A 640x geprüft.

Verwalten

Die Aufgabe des Verwaltens ist die Abspeicherung, Wiederauffindung und Pflege von P05 im Entwicklungs-, Wartungs- und Anwendungsprozeß.

Dokumentieren

Die Aufgabe des Dokumentierens ist es, alle relevanten Daten über ein POS-Produkt schritt haltend mit dem Prozeß seiner Entwicklung zu fixieren.

Kontrollieren

Die Aufgabe des Kontrollierens ist die technologische Überprüfung der End- und Zwischenergebnisse des Entwicklungsprozesses von POS, schritt haltend mit dem Entwicklungsablauf.

3. TESO 16p0

Das Kernstück der POST 1600 sind die programmtechnischen Mittel, das heißt TESO 1600.

Mit Vorliegen des programmtechnischen Entwurfs beginnt die Arbeit mit TESO 1600. Unter Nutzung der Kodiermittel TESO 1600 wird der Entwurf in Form von Quellprogrammen notiert. Mit der Eingabe der Quellprogramme wird die im Dialog gesteuerte maschinelle Überführung des Entwurfs in ablauffähige Systemunterlagen eingeleitet.

Die Bedienung des Systems geschieht im Mensch-Maschine-Dialog auf problemorientiertem Niveau. Mittels einfacher Kennzeicheneingabe am Bildschirmterminal werden die von TESO 1600 gewünschten Leistungen angefordert. TESO 1600 spezifiziert dann im Dialog mit dem Systemunterlagenentwickler die benötigten Detailangaben selbst und führt anschließend die gewünschten technologischen Aktionen aus. TESO 1600 bietet programmtechnische Mittel unter anderem für folgende Aufgaben:

Editieren

- Bereitstellen von Quelltexten aus entsprechenden Bibliotheken
- Korrektur interaktiv oder mit bereitgestellten Korrekturdaten
- Rückspeln in Bibliothek

Assemblieren

- Bereitstellen des Quelltextmoduls
- Verketten der benötigten Bibliotheken
- wahlweises Katalogisieren des erzeugten Objektmoduls

Verbinden

- Bereitstellen des Top-Moduls
- Verketten der benötigten Bibliotheken
- Binden zum ablauffähigen Programm
- Speichern des gesetzten Nutzerbereichs

Modulanpassung

- Eintragen der erforderlichen Adressierungsmodi
- Anpassen des Moduls an spezielle Einsatzbedingungen

Katalogisieren

- von Quelltexten in Quelltextbibliotheken
- von Objektmoduln in Objektmodulbibliotheken

Pseudomodul

- Erzeugung von Pseudomodul für noch nicht programmierte Moduln niederer Ebenen zum Testen einer Modulhierarchie nach dem TOP-DOWN-Prinzip

Testrahmen

- Erzeugung von Testrahmen für den Einzeltest von Moduln

Datenmanipulator

- Ändern (Löschen, Ersetzen, Einfügen) von Datensätzen bzw. einzelnen Daten in einer Testdatei
- interaktives Eingehen von Daten zur Ausführung der beabsichtigten Änderung in der Testdatei
- automatisches Anpassen des Datentyps der eingegebenen Daten an den intern erforderlichen Typ (Datenkonvertierung)
- Ausgabe des geänderten Satzes auf das Bediengerät
- Protokofierung der ausgeführten Datenmanipulation.

4. Bereitstellungsform der POST 1600

Die POST 1600 wird, vom Robotron-Vertrieb Berlin vertrieben und in Form von Vertriebseinheiten (VE) bereitgestellt.'

Vertriebs-einheit	Bezeichnung	Bestandteile
1	POS-Technologie Konzeption der POS-Technologie und allgemeine technologische Mittel	Konzeption der POS-Technologie Begriffe der POS-Technologie Funktionsorientierte Entwurfsmethode - Datenorientierte Entwurfsmethode Pseudocode Strukturprogramme - HIPO-Entwurfsmethode - Dokumentationssystematik für problemorientiertes Entwerfen Testmethodik
	POS-Technologie POST 1600 Allgemeine methodische Mittel	Übersicht über POST 1600 Kurzcharakteristik TESO 1600 ABI auf des Entwicklungsprozesses mit POST 1600 Programmierorientiertes Entwerfen Dokumentationssystematik der POST 1600
	Technologisches Programmsystem zur Software-Entwicklung TESO 1600	TESO 1600 (MAC) auf Datenträger (Makroassemblersprache) TESO 1600 (FOR) auf Datenträger Anwendungsbeschreibung für TESO 1600 (MAC) Anwendungsbeschreibung für TESO 1600 (FOR) Programmtechnische Beschreibung

1 Die VE 1 wird auch über den VEB Robotron-Zentrum für Forschung und Technik Dresden vertrieben.

POS für die Basisrechner des Kombinats Robotron

Robert Syring
VEB Robotron-Vertrieb Berlin

1. Einführung

Die Fortschritte der Gerätetechnik zur Informationsverarbeitung und der Erkenntnisfortschritt auf dem Gebiet der Systemunterlagenentwicklung haben in den letzten einundzwanzig Jahren unter anderem auch zu beträchtlichen Veränderungen in der Struktur der zu verwendenden Systemunterlagen geführt: Auf seiten der problemorientierten Systemunterlagen (POS) hat der von konkreten Anwendungsgebieten unabhängige Teil eine zunehmende Bedeutung erlangt. Für die Basisrechner A 640x des Kombinats Robotron und die darauf aufbauenden arbeitsplatzbezogenen Konfigurationen A 64xx gibt es diese anwendungsneutrale PoS in zwei Stufen:

- Als von jedem Anwendungsgebiet unabhängige P05, die zur Lösung von allgemeinen, in allen (oder einer Mehrzahl von) Anwendungsgebieten auftretenden Informationsverarbeitungsaufgaben genutzt werden können (in der Terminologie des Kombinats Robotron als Querschnitts-POS (QPOS) bezeichnet) und speziell für Basisrechner von Bedeutung sind

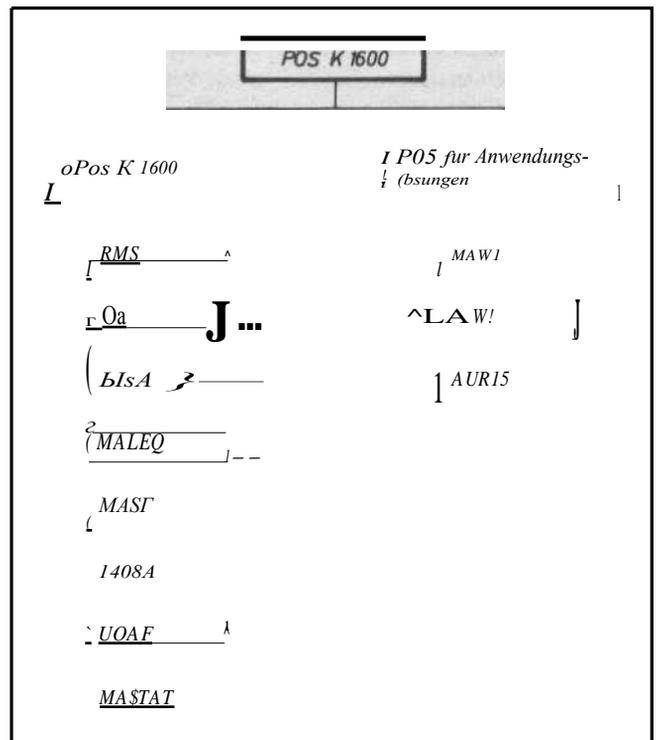


Abb. 1 Struktur der POS K 1600

- als für eine spezielle arbeitsplatzbezogene Konfiguration ersetzbar, aber noch vom konkreten Anwendungsgebiet unabhängige POS 1/1.

Die generelle Struktur der POS K 1600 ist in Abb. 1 dargestellt. Dieser Beitrag gibt dem potentiellen Anwender von Basisrechnern A 640x und von darauf aufbauenden arbeitsplatzbezogenen Konfigurationen A 64xx einen Überblick über die verfügbare QPOS und befähigt ihn, seine Einsatzvorbereitung entsprechend auszurichten.

2. Querschnitts-POS des K 1600

Die in der allgemeinen POS K 1600 enthaltene POS mit querschnittsorientiertem Charakter setzt sich aus folgenden Komponenten zusammen:

RMS 1600	Record Management Services — Satzzugriffssystem
DAVE 1600	Dateiverarbeitung
DABA 1600	Datenbankbetriebssystem
MALEQ 1600	Mathematische Verfahren — Matrizenoperationen und lineare Gleichungssysteme
MAST 1600	Mathematische Standardprogramme
MOBA 1600	Modulbasis
UDAF 1600	Universelles Dateneingabe- und Prüfsystem
MASTAT 1630	Verfahren der mathematischen Statistik.

Für einige dieser Komponenten wird in einer weiteren Ausbaustufe das Leistungsspektrum erweitert.

Die zusätzlich zu entwickelnden Komplexe wurden aus den Erfordernissen bereits praktizierter Einsatzfälle abgeleitet (siehe Abschnitt 2.9.).

2.1. RMS 1600

RMS 1600 ist ein universelles Satzzugriffssystem, das für alle die Aufgabengebiete besonders geeignet ist, bei denen umfangreiche Datenmengen erfaßt, archiviert, aktualisiert und wieder aufgefunden werden müssen. Die vorliegende Version ist nur auf dem MRS K 1630 abarbeitbar.

Aufgrund seines modularen Konzepts kann es leicht an konkrete Einsatzfälle angepaßt werden.

Mit Hilfe von RMS können die Zugriffsarten sequentiell und wahlfrei für Dateien mit den Organisationsformen sequentiell, relativ und index-sequentiell realisiert werden (Tab. 1).

RMS gestattet den simultanen geteilten Zugriff mehrerer Nutzer zu einer Datei. Ein Wechsel der Zugriffsart ist bei jedem neuen Dateioffnen möglich (dynamischer Zugriff). Der im Betriebssystem MOOS 1600 gewährte Dateischutz ist auch für RMS-1600-Dateien vorgesehen.

Aufgrund der Blockungsmethode (Zusammenfassen größerer

Tab. 1 Übersicht des Satzzugriffssystems RMS 1600

Zugriffsmethode	Dateiorganisation			
	sequentiell	relativ	index sequentiell	wahlfrei (Zugriff über:)
				Satz-Nr. Schlüssel Datei-Satz-adresse
sequentiell	K			K
relativ	X,	X		K
index sequentiell	K		X	X

Datenbereiche) wird eine Verringerung der erforderlichen Zugriffe zu den Dateien erreicht.

Programmkonzeption des RMS 1600

Das Nutermakro-Interface stellt für alle Servicefunktionen Macros mit problemorientierten Bezeichnungen bereit. Über die Steuerblöcke erfolgt der Informationsaustausch zwischen Nutzerprogramm und RMS 1600. Alle datei- bzw. satzbezogenen Informationen, die für die Ausführung von bestimmten Aktionen erforderlich sind, werden hierüber dem RMS zur Verfügung gestellt, während nach Ausführung der Operationen Statusinformationen über deren Verlauf von RMS in den Steuerblöcken übergeben werden. Die Nutzer-Macros nutzen weitere Zwischenmacros und die in MAC 1600 enthaltene Möglichkeit der bedingten Assemblierung.

Die einzelnen RMS-Servicefunktionen finden in Objektgruppen, die entweder als Überlagerungsstrukturen oder über Systembibliotheken von den Anwendertasks genutzt werden können.

Schnittstellen der Datenstruktur mit RMS 1600

Zwischen den Leistungen des Operationssystems und der satzorientierten Programmumgebung des Anwenders fungiert RMS 1600 als Vermittlungssystem. Dafür benötigt RMS 1600 Informationen über das Satzformat, die Dateiorganisation, die Zugriffsarten und über die Satzoperationen.

Umsetzen der programmtechnischen in die gerätetechnische Datenstruktur

Im Anwenderprogramm werden Datenverarbeitungsoperationen aktiviert, die von RMS 1600 auf die internen Datenstrukturen umgesetzt und laut Anforderung an das Operationssystem MOOS 1600 zur Gerätesteuerung modifiziert werden (Abb. 2).

Satzoperationen

Folgende Satzoperationen werden von RMS 1600 unterstützt:

FIND	Positionierung eines Satzes
GET	Lesen eines Satzes

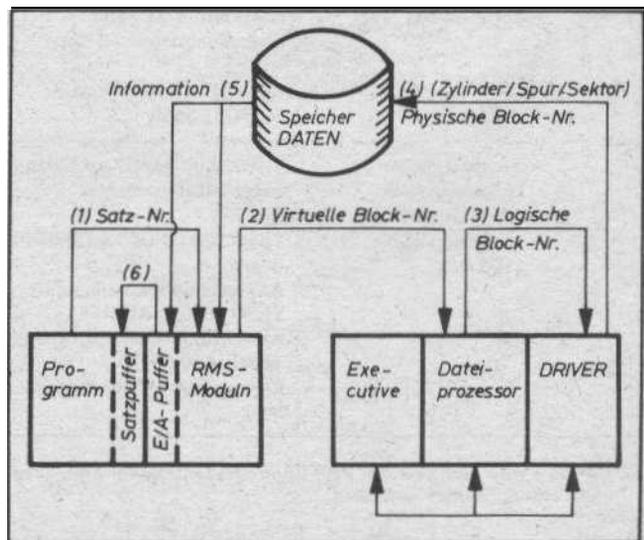


Abb. 2 Umsetzen der programmtechnischen in die gerätetechnische Datenstruktur

PUTT	Schreiben eines Satzes
DELETE	Loschen eines Satzes
UPDATE	Aktualisieren eines Satzes
TRUNCATE	Kürzen einer sequentiellen Datei
CONNECT	Einrichten eines Satzzugriffsstromes
DISCONNECT	Aufheben eines Satzzugriffsstromes
REWIND	Positionieren auf Dateibeginn
FLUSH	Plattentransfer modifizierter E/A-Puffer.

Dateioperationen

Auf der Dateiebene unterstützt RMS 1600 die folgenden Dateioperationen:

CREATE	Einrichten einer RMS-1600-Datei
OPEN	Öffnen einer Datei
DISPLAY	Bereitstellen der Attribute einer Datei
EXTEND	Erweitern einer Datei
CLOSE	Schließen einer Datei
ERASE	Löschen einer Datei.

2.2. DAVE 1600

DAVE00 stellt programmtechnische Mittel für die rationelle Erarbeitung und Verwaltung von datei-strukturierten Datenbanken zur Verfügung. Diese Dateien müssen den Bedingungen des Dateizugriffsystems FCS 1600 entsprechen.

Die programmtechnischen Mittel werden in Abhängigkeit von den zu realisierenden Aufgaben in Definitions- und Manipulationselemente unterteilt (Abb. 3).

Sie sind sowohl über Macro- als auch Objekteinbindung im Anwenderprogramm nutzbar. Damit besteht die Möglichkeit, DAVE 1600 in höheren Programmiersprachen zu verwenden. Die DAVE-Komponenten sind abgestimmt mit den Lösungen der POS-Technologie und setzen deren Anwendung im Applikationsprogramm voraus. (Es sind mindestens die Interface- und Datendefinitionsmacros der POSr 1600 zu notieren)

Kurzbeschreibung der Komponenten

Die Definitionselemente ermöglichen in anwenderfreundlicher Notation Vereinbarungen zu den Dateien.

Mit den Manipulationselementen werden die Grundfunktionen der Datenein- und -ausgabe realisiert. Mit Hilfe dieser E/A-

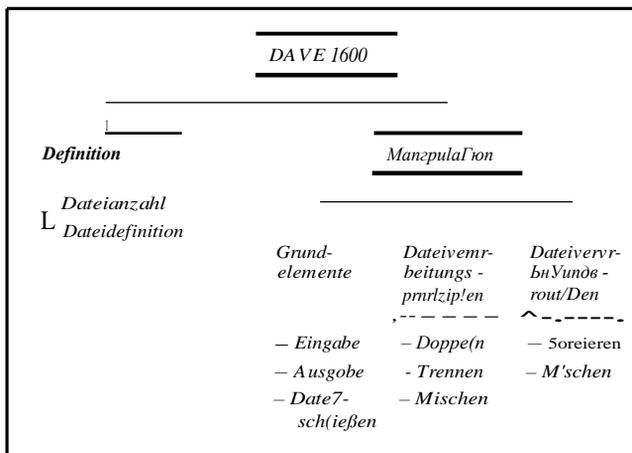


Abb. 3 Übersicht der programmtechnischen Mittel von DAVE 1600

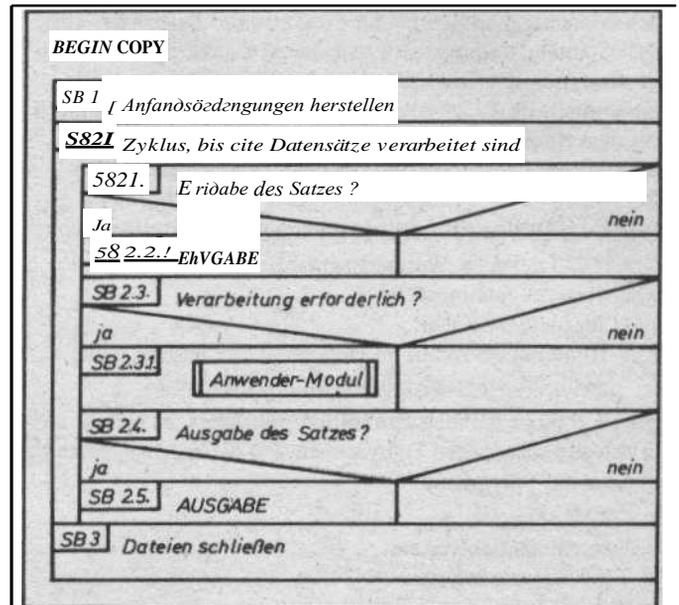


Abb. 4 Struktogramm DOPPELN

Grundelemente lassen sich komplette Dateiverarbeitungsprinzipien und -routinen verwirklichen.

Für typische Dateiverarbeitungsprinzipien werden Komplexmoduln bereitgestellt:

- **Doppeln:**
Die logischen Sätze einer Eingabedatei werden zu einer Ausgabedatei verarbeitet.
- **Trennen:**
Die logischen Sätze einer Eingabedatei werden auf zwei Ausgabedateien verteilt.
- **Mischen:**
Die logischen Sätze zweier sortierter Eingabedateien werden zu einer Ausgabedatei vereinigt.

Mit dem Anwenden dieser Komponenten wird der Nutzer **weitgehend** von der EWA-Organisation und der Steuerung der Verarbeitung befreit. Es sind Austrittspunkte vorgesehen, um anwenderspezifische Moduln abarbeiten bzw. Steuerkriterien setzen zu können (Abb. 4).

■ Sortieren:

Das Sortierprogramm (DVSODI) ermöglicht das Sortieren von Kassettenplattendateien mit fester Satzlänge. Es ist Sortierung am Ort (Sorting in Place) oder Sortierung durch Duplizieren (Sorting by Duplicating) möglich.

Die Anzahl der Sortiermerkmale ist beliebig. DVSODI ist nur als Hauptprogramm abarbeitbar. Die Parameterversorgung wird entweder im Dialog oder mittels Indirekt-Kommando-Datei ausgeführt.

DVSODI benötigt etwa 5000 Worte Speicherplatz und einen Mindestarbeitspeicher von 4 K Byte plus 5 mal Satzlänge.

2.3. DABA 1600

Innerhalb des Aufgabenkomplexes Datenorganisation DATO 1600 wird mit dem Datenbankbetriebssystem DABA 1600 ein abgeschlossenes Programmsystem zur Verfügung gestellt. DABA 1600 unterstützt den Aufbau, die Verwaltung sowie die Nutzung einer Datenbank. DABA 1600 arbeitet als eigenständiges

Anwendertasksystem unter *der* Steuerung des Betriebssystems MOOS 1600. Das Betriebssystem steuert DABA über die Prioritätssteuerung als Multitasksystem. DABA 1600 bedient sich der Komponenten des Betriebssystems bei der Kommunikation mit dem Nutzer, zur Speicherung und zum Dateizugriff, der auf der Ebene von FC5 1600 (Blockein- und -ausgabe) realisiert ist.

DABA 1600 erfordert folgende minimale Gerätekonfiguration:

- ⊠ 1630 mit 64 K Worten Hauptspeicher
- 4 Kassettenplattenspeicher
- | Magnetbandeinheit
- | Terminal bei Arbeit im Programmmodus, ansonsten bei Mehrnutzerbetrieb ein System von Terminals.

DABA 1600 ist modular strukturiert konzipiert. Es besteht aus in sich abgeschlossenen Teilsystemen. Für die ständige Nutzung sind das die Teilsysteme

- DB-Monitor
- Kommunikationssystem
- Transformationssystem
- Zugriffssystem.

Der DB-Monitor übernimmt die DABA-interne Steuerung. Der Aufbau sowie die Verwaltung der DB erfolgt in den Teilsystemen

- DB-Betriebssystemgenerierung
- Datenbankservice.

Die Gewährleistung der Ordnungsmäßigkeit erfolgt *im* Teilsystem Datenschutz und Datensicherheit.

Funktionen dieses Teilsystems werden aber auch in den anderen Teilsystemen realisiert.

Dem Datenbanksystem DABA liegt ein relationales Datenkonzept zugrunde, das heißt, die zu verwaltenden Daten werden in Relationen geführt.

Dieses Konzept läßt dem Nutzer der Datenbank die gespeicherten Informationsbestände als Anzahl von Tabellen erscheinen. Dadurch wird heur Nutzer einerseits die Vorstellung von der konventionell gebräuchlichsten Form der Speicherung von Informationen erzeugt und andererseits völlig von den Datenstrukturen sowie der Gerätetechnik abstrahiert. Damit wird dem gegenwärtigen Trend Rechnung getragen, in zunehmendem Maße auch Anwendungsspezialisten ohne rechentechnische Spezialkenntnisse die Arbeit mit der Datenbank schnell zu ermöglichen. Als Datenbanksprache wird SEQUEL2 eingesetzt. Diese Sprache besteht aus Anweisungen, die in Englisch zu codieren sind. Der Anwendungsbereich von DABA 1600 ist überall dort, wo die wesentlichen Informationsbestände in Tabellenform vorliegen bzw. in diese überführt werden können. Der Einsatz von DABA 1600 bei der Verarbeitung von hierarchischen Netzwerk- oder Baumstrukturen erfordert deren Normalisierung. DABA 1600 wurde in drei Ebenen entworfen. Nur die äußere oder externe Ebene ist für die einzelnen Nutzer interessant. In der *externen Ebene* wird die logische Anordnung der Daten einer Datenbasis aus Sicht der Nutzer dargestellt. Die *konzeptuelle Ebene* stellt die Gesamtheit der in der Datenbank verwalteten Informationen des Anwendungsbereiches dar. Die Struktur von externer und konzeptueller Ebene muß nicht übereinstimmen. Die *interne Ebene* stellt die physische Realisierung der Speicherung der Informationen dar.

DABA 1600 wird im Dialogmodus mit interaktiver oder vorprogrammierter Dialogführung oder im Programmmodus genutzt. Jedes Terminal kann nur von einem Nutzer nach seiner Anmel-

dung bedient werden. Es bleibt bis zu seiner Abmeldung blockiert. Bei der interaktiven Nutzung arbeiten die Nutzer nur mit den Sprachelementen von SEQUEL2. Auszugebende Daten erscheinen auf dem Terminal und/oder für spätere Bearbeitung auf einer *zusätzlichen* Datei, wobei Fehlernachrichten immer auf dem Terminal erscheinen. Bei der vorprogrammierten Dialogführung erfolgt der Dialog indirekt über eine Kommandodatei. Die Kommandodateien werden mittels des Betriebssystems MOOS 1600 auf Magnetplatte oder Magnetband aufgebaut. Beim Programmmodus werden SEQUEL-Anweisungen in die Wirtssprachen COBOL oder MACRO 1600 eingekettet. Durch eine Vortibersetzung werden die SEQUEL-Sprachelemente zu Parametersätzen aufbereitet. Die anschließende Übersetzung mittels Wirtssprachencompiler schafft die Verbindung zu den aufbereiteten Parameterblöcken. Während der Laufzeit der Anwendungsprogramme werden diese Parameterblöcke aufgerufen und überlernen dann die Steuerung der Arbeit mit der Datenbank. Durch diese Arbeitsweise wird das mehrmalige wiederholte Abarbeiten der Datenbankmanipulationen ermöglicht. Ein Programm, das nur aus SEQUEL-Anweisungen besteht, *ist* für die wiederholte Abarbeitung nicht nutzbar. Dem *Datenschutz* im DABA 1600 liegt das Konzept einer Nutzerdatei zugrunde. Diese Datei wird mit Mitteln des Betriebssystems geschlitz. Jeder Nutzer wird darin mit seinem Schutzcode und den Zugriffsrechten auf die einzelnen Datenbestände eingetragen.

Die rechentechnische Unterstützung des Datenschutzes basiert auf den Basisrelationen. Die Beschränkungen können sich sowohl auf Tabellen als auch auf Zeilen und Spalten und Tabellen beziehen. Als unberechtigt erkannte und dadurch abgewiesene Zugriffsversuche werden protokolliert.

Die Datensicherheit wird von einer Reihe von Hilfsmitteln ebenfalls auf der Ebene der Basisrelationen unterstützt. Um einen Schutz vor fallweise auftretenden Datenverlusten zu gewährleisten, erfolgt bei DABA 1600 die zusätzliche Protokollierung aller Datenbewegungen zwischen den Prüfpunkten auf dem Logband.

2.4. MALEQ 1600

7

MALEQ 1600 stellt Programme für mathematische Verfahren zur Verfügung, die bei Aufgabenstellungen sowohl in wissenschaftlich-technischen, als auch in ökonomischen Bereichen anwendbar sind.

Dem Anwender wird eine Sammlung von in FORTRAN geschriebenen Unterprogrammen zur Lösung linearer Gleichungssysteme und für Operationen mit reellen Matrizen angeboten. Die Lösung von *linearen Gleichungssystemen* ist ein sehr häufig auftretendes mathematisches Problem. Im ökonomischen Bereich werden zum Beispiel Verflechtungsbilanzen und im wissenschaftlich-technischen Bereich vielfältige Aufgabenstellungen, wie Interpolation von Funktionswerten und Approximation von Differentialgleichungen durch Differenzgleichungen mit Hilfe linearer Gleichungssysteme gelöst.

Allgemeine *Matrixoperationen* finden eine eigenständige Verwendung bei der Behandlung von Matrizenmodellen, die nicht nur in ökonomischen, sondern auch in technologischen Einsatzbereichen vorkommen.

In Abhängigkeit von der jeweiligen Problemstellung besitzen die Matrizen eine spezielle Struktur, die sowohl für eine speicherplatzgünstige Anordnung, als auch für einen effektiven

Verarbeitungsalgorithmus genutzt werden kann_ Es werden Algorithmen für
 — allgemeine Matrizen (einschließlich quadratischer Matrizen)
 · symmetrische Matrizen und
 — Bandmatrizen
 zur Verfügung gestellt.

Programmkonzeption

Die Unterprogramme realisieren klar abgegrenzte Aufgabenstellungen.
 Zur Erleichterung der Arbeit mit den verschiedenen Matrizenarten stehen dem Anwender spezielle Ein- und Ausgabeprogramme zur Verfügung.
 Es wurde auf eine klare Trennung von Verarbeitungs- sowie Ein- und Ausgabe-Moduln orientiert.

Programmübersicht

- Arithmetische Vektoroperationen
- Operationen mit allgemeinen Matrizen
 - Eingabe
 - Ausgabe
 - Dateiarbeit
 - Kopieren und Transponieren
 - Arithmetische Operationen
- Operationen mit symmetrischen Matrizen
 - Eingabe
 - Ausgabe\
 - Kopieren
 - Arithmetische Operationen
- Operationen mit allgemeinen Bandmatrizen
 - Eingabe
 - Ausgabe
 - Kopieren und Transponieren
 - Arithmetische Operationen
- Lineare Gleichungssysteme mit quadratischer Koeffizientenmatrix
- Überbestimmte lineare Gleichungssysteme
- Lineare Gleichungssysteme mit positiv definierter Koeffizientenmatrix
- Lineare Gleichungssysteme mit einer Koeffizientenmatrix von allgemeiner Bandstruktur.

2.5. MAST 1600

Mit den Moduln der MAST 1600 wird dem Nutzer eine Möglichkeit geboten, den gegenüber den Festkommazahlen erweiterten Zahlenbereich der Gleitkommadarstellung zu nutzen. Voraussetzung für die Anwendung ist, daß das die Moduln aufrufende Programm in MACRO 1600 geschrieben ist.

Programmkonzeption

Die Komponenten der MAST 1600 werden in drei Bibliotheken als Objektmoduln bereitgestellt:

- Moduln für Gleitkommaarithmetik einfacher Genauigkeit
- Moduln für Gleitkommaarithmetik doppelter Genauigkeit
- Moduln für Konvertierungsaufgaben.

Jede der Bibliotheken enthält alle Moduln, die zur Lösung der entsprechenden Aufgaben notwendig sind.

Funktionsübersicht

- Folgende Aufgaben können mit der Modulsammlung der MAST 1600 gelöst werden:
- Grundrechenoperationen
 - Funktionswertbestimmung einiger elementarer Funktionen
 - Negation von Primärzahlen
 - Runden von Gleitkommazahlen (doppelte Genauigkeit in einfache Genauigkeit)
 - Multiplikation und Division binärer Festkommazahlen
 - K01-7-Konvertierung
 - Binärkonvertierung.

2.6. MOBA 1600

Die Modulbasis MOBA 1600 enthält Verarbeitungsmoduln in übersetzter Form. Jeder Modul realisiert eine abgegrenzte Aufgabenstellung mit eindeutiger Zielsetzung.
 Im Rahmen der MOBA 1600 werden keine komplexen Anwendungslosungen angeboten, sondern Hilfsmittel für die Projektierung von Datenverarbeitungsprozessen bereitgestellt.
 Die Moduln wurden vorwiegend entsprechend den Reglementierungen der POS-Technologie entwickelt. Für die programmtechnische Verknüpfung wird ein Interface vorausgesetzt, das vom jeweiligen Verarbeitungsmodul bestimmt wird. Die Anwendung dieser Moduln fordert deshalb die Schaffung der gleichen Bedingungen im aufrufenden Programm, die dem vorgefertigten Modul zugrunde liegen (Verwendung der Makros der POS-Technologie POST 1600 /2/).

Kurzbeschreibung der Komponenten

In der MOBA 1600 sind zwei Gruppen von Moduln enthalten — Moduln mit Querschnittscharakter (MOBA/G) und Moduln für die Projektierung betriebswirtschaftlicher Prozesse (MOBA/B). (Die zuletzt genannten sind nicht querschnittsorientiert-anwendungsneutral, aber aus praktischen Gründen hier eingeordnet.) Die Moduln der MOBA/G lassen sich wie folgt klassifizieren:

- Organisations-Moduln
 Bearbeitung logischer Einheiten (Sätze, Felder) ohne Änderung ihres Informationsinhalts
- Prüf-Moduln
 Prüfung logischer Einheiten
- Arithmetik-Moduln
 Verarbeitung logischer Einheiten nach vorgegebener Rechenvorschrift
- Konvertierungs-Moduln.

Die in der MOBA/B enthaltenen Moduln gliedern sich in fünf Gruppen:

- Nettolohn
 - Lohnsteuerberechnung nach Tages- oder Monatstabellen
 - Berechnung des Nettodurchschnittsverdienstes
 - Berechnung der Geldleistungen der Sozialversicherung
 - Berechnung des Beitrages für die Sozialpflichtversicherung
 - Berechnung von Abzügen
 - Stückelung eines vorgegebenen Auszahlungsbetrages in einer kleinstmöglichen Anzahl von Geldsorten
 - weitere Berechnungen
- Grundmittel
 Mit diesen Moduln kann im Rahmen der Grundmittelrech=

nung das Teilgebiet Abschreibungsrechnung bearbeitet werden. Es werden für jedes Inventarobjekt der vorher aktualisierten Grundmittelstammdatei monatlich die Abschreibungen und der sich daraus ergebende Verschleiß berechnet. Dabei werden wertmäßige Änderungen und erforderliche Rückrechnungen durch verspätete Buchungen berücksichtigt.

- **Kostenrechnung**
Diese Moduln unterstützen im Rahmen der Kostenrechnung die Kostenarten- und Kostenstellenrechnung.
- **Inventuren**
Mit diesen Moduln wird die Programmierung von Anwendungslösungen für automatisierte Inventuren, insbesondere der Grundmittelinventuren unterstützt.
— Vergleich Inventurbestand mit vorgegebenem Soll-Bestand
— Ausgabe der echten Inventurdifferenzen
— Ermittlung und Ausgabe der wiederaufgefundenen Inventurgegenstände
- **Fakturierung**
— Falligkeitsherechnung
— Mengenumrechnung
— Berechnung der Handelsspanne
— Rechnungsbetragsrutschlagsverrechnung
— Rechnungsabgrenzung
— Rechnungsbetragsatzerzeugung.

2.7. UDAF 1600

Beim Einsatz von Basisrechnern A 640x im ökonomischen Bereich, bei der Arbeit mit Datenbanksystemen und -archiven ist eine ausführliche Datenprüfung vor dem Dateiaufbau von großer Bedeutung. Diesem Zweck dient das Programmsystem UDAF. Es umfaßt Programme und Routinen für folgende Aufgaben:

- Umsetzung der eingegebenen Daten auf Einheitscode, Ausblenden von Irrungen, Herstellung einer UDAF-Eingabedatei auf Magnetplatte
- Belegprüfung
- Fehlerprotokoll
- Korrektur der Einzelbelege
- Konvertierung der Dateien ins interne Format und aus dem internen Format
- Umwandlung von Dateien anderer Struktur in UDAF-Dateien und umgekehrt.

Zusätzlich können die Dateien mittels spezieller Programme im Hauptspeicher sortiert und gemischt werden.

Die UDAF-Programme arbeiten satzorientiert. Das heißt, der Satz (Beleg) bleibt auch in der externen Form in einer Datei erhalten.

Nach der Eingabe werden die Belege einer Datei anhand ihrer Satzartbeschreibung überprüft. Diese vom Anwender vorzuziehende Beschreibung muß alle Struktur- und Feldinformationen enthalten — wie Feldunterteilungen im Satz, Datentypen, Feldinhaltsmöglichkeiten oder Wertevorrat, gewünschte Prüfungen usw. Sie bildet damit die Grundlage für alle Verarbeitungsaufgaben der UDAF-Programme. Als Datentypen sind alle numerischen Datenformate (außer Gleitkomma) bis zum Vierfachen Wort, KOI-? und Radix-50-Texte und UDAF-spezifische Vereinbarungen wie Datum, Personenkennzahl und Tabellen dieser Daten zugelassen.

Die Belegdateien in interner Form werden erst nach erfolgreicher

Prüfung und eventueller Korrektur fehlerhafter Belege auf Platte gespeichert. Bei späteren Verarbeitungen kann man davon ausgehen, daß sie nur geprüfte Sätze enthalten. Die Verwendung der Satzartbeschreibungen macht es möglich, daß alle Belegarten von demselben Programm verarbeitet werden. Zur Erweiterung oder Änderung des Anwenderdateisystems genügt es, neue Satzartbeschreibungen einzufügen, um neue Belegarten prüfen zu können. Programmtechnische Änderungen sind dazu nicht erforderlich. Allerdings kann der Anwender zusätzliche Prüfprozeduren selbst an das Belegprüfungsprogramm anschließen.

Das Programmsystem UDAF ist modular aufgebaut, wobei eigene UDAF-Makro- und Objekttextbibliotheken benutzt werden. Es ist für den Einsatz mit den Betriebssystemen LAOS 1600 und MOOS 1600 vorgesehen.

2.13. MASTAT 163a

Mit dem verfahrensorientierten Programmpaket MASTAT können mathematisch-statistische Aufgabenstellungen gelöst werden. Es werden Programme für folgende Komplexe zur Verfügung gestellt:

- Dateneingabe und -manipulation
- Elementare Datenaufbereitung
— Berechnung von Maßzahlen zur Beurteilung von Verteilungen
- Varianzanalyse
— Bestimmung des Einflusses von einem oder mehreren qualitativ erfaßten Merkmalen auf ein quantitatives Merkmal
- Regressionsanalyse linear oder nichtlinear
— Bestimmung der Abhängigkeit eines quantitativen Merkmals von einem oder mehreren quantitativen Merkmalen
- Zeitreihenanalyse
— Untersuchung der zeitlichen Abhängigkeit von Merkmalen.

2.9. oPos-Kompletierung

- **RMS 1600:**
Das Leistungsspektrum wird dahingehend erweitert, daß eine Anwendung der Satzzugriffsroutinen auf dem MRS K 1620 möglich ist.
Weiterhin wird ein System von Dialogroutinen zur Verfügung gestellt, die den Nutzer befähigen, ohne eigenen Programmierungsaufwand interaktive Zugriffe zu Daten und deren Verarbeitung auszuführen.
- **DAVE 1600:**
Zur Manipulation von Datensätzen wird ein Zugriff über Primärschlüssel realisiert, mit dem das Einfügen, Ändern und Löschen von Sätzen durchgeführt werden kann.
- **MOBA 1600:**
Die Modulbasis wird um Moduln zur Primärdatenverarbeitung von Meßwerten komplettiert.

3. Vertrieb der QPOS-Komponenten

Die Querschnitts-POS K 1600 wird entsprechend der territorialen Verantwortlichkeit von den zuständigen Robotron-Vertriebsbetrieben verkauft.

Der Vertrieb der Komponenten erfolgt in zwei Formen:

1. Als vorgefertigte POS für Projektanten
2. Im Rahmen kompletter Projekte für nichtprojektierende Nutzer als Inhalt der Projektbibliotheken.

Bei der Lieferung im Anwenderprojekt ist die Verwendung der QPOS K 1600 normalerweise nicht sichtbar und damit die Bereitstellung der Dokumentation nicht erforderlich.

Als Dokumentationsschriften werden je Komponente die „Anwendungsbeschreibung“ und die „Anleitung für die Programmierer“ bereitgestellt.

Literatur

- /1/ Schulze, W.: Kleinrechner im Konzept der dezentralen Datenverarbeitung. edv-aspekte 2 (1983) 1, s. 2
- /2/ Klette, W.: system zur softwareentwicklung YOST 1600. edv-aspekte 2 (1983) 1 s. 37
- /3/ Anwendungsbeschreibungen der QPOS- Komponenten
- /4/ Informationsschriften des VEB ROBOTRON- Zentrum für Forschung und Technik



Seit der Vorstellung des Gerätekonzeptes „dezentrale Datentechnik“ hat auch der Tenninus Bürocomputer in der Volkswirtschaft der DDR eine weite Verbreitung gefunden. Lassen sich doch diese Geräte aufgrund ihrer Merkmale *arbeitsplatzorientiert, autonome Betriebsweise, leicht konfigurierbar, vom Endnutzer programmierbar* in einer Vielzahl von Anwendungsgebieten als Rationalisierungsmittel verwenden. Das gilt sowohl für klassische Aufgaben aus Rechnungsführung und Statistik sowie für echtzeitnahe Informationsverarbeitungsaufgaben zur Steuerung von Leistungsprozessen, als auch für wissenschaftliche und ökonomisch-mathematische Berechnungen. Mit der Vorstellung von Hardware und Software in edv-aspekte 1/82 wurde das Einsatzspektrum der Bürocomputer bereits aufgezeigt. Zum Umsetzen dieser gerätetechnischen Möglichkeiten in die Realität der Informationsverarbeitung — bei der Einsatzvorbereitung — bedarf es jedoch besonderer Leistungen seitens des Anwenders.

Diesen Prozess soll die Ausgabe 2/83 der edv-aspekte unterstützen, indem unter dem Titel

Erfahrungen mit Bürocomputern

künftigen Nutzern die Erkenntnisse erfahrener Anwender vermittelt werden. Es sind Beiträge sowohl aus den verschiedensten Bereichen der Volkswirtschaft als auch für vielfältige Aufgabenstellungen vorgesehen.

Für künftige Anwender und für den Hersteller von Bürocomputern dürften die Ergebnisse einer Umfrage zur Einsatzvorbereitung und zu Einsatzerfahrungen von Interesse sein.

Bestellungen für edv-aspekte 2/83 richten Sie bitte rechtzeitig an den Verlag Die Wirtschaft, Abt. Vertrieb, 1055 Berlin, Am Friedrichshain 22.

edv-aspekte 2/83 erscheint im 1. Quartal 1983.

Grafische Systemunterlagen des AKT A 6454

Wolfgang Baier, Dr. Manfred Mikut, Wilburga Reller
VEB Robotron-ZFT Dresden

1. Konzept der Systemunterlagen

Die Gerätetechnik des AKT A 6454 bildet in Verbindung mit den Systemunterlagen ein wirkungsvolles CAD/CAM-System zur Erhöhung der Produktivität in der technischen Vorbereitung der Produktion.

Das Systemunterlagenkonzept des AKT A 6454 (Abb. 1) unterscheidet folgende Komponenten:

- Modulares Betriebssystem
- grafische Grundsystemunterlagen einschließlich gerätespezifischer grafischer Schicht
- geometrische Grundsystemunterlagen
- problemorientierte Systemunterlagen mit Querschnittscharakter
- Anwendersystemunterlagen.

Der AKT A 6454 wird ausgerüstet mit dem Betriebssystem MOOS 1600. Alle für den KBR A 6402 verfügbaren Systemunterlagen einschließlich der POS-Technologie und der Querschnitts-POS können genutzt werden.

Für die grafische Arbeit werden folgende Grundsystemunterlagenkomponenten bereitgestellt:

- GKS 1600 Grafisches Kernsystem für Display und Plotter
- DIG 1600 für Digitalisiergerät HOG K 6401
- CBS i 600 Geometriebaustein.

Die Anwendersystemunterlagen werden auf der Basis der Schnittstellen des AKT unter Verantwortung der Hauptanwender im Rahmen von Pilotprojekten des Staatsauftrages „Rationalisierung der technischen Vorbereitung der Produktion“ entwickelt /1/.

Nachfolgend werden schwerpunktmäßig die grafischen Grundsystemunterlagen beschrieben.

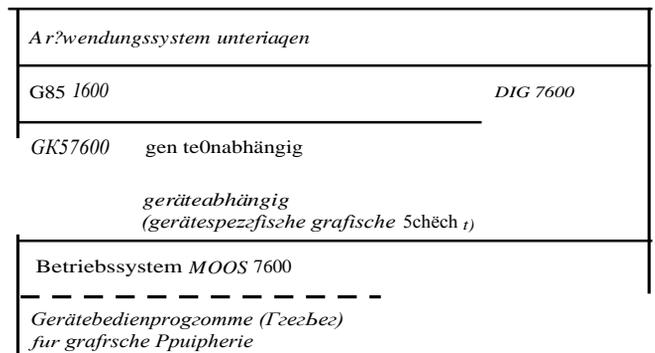


Abb. 1 schichtenmodell der Systemunterlagen des AKT A 6454

2. GKs 1600

2.1. Charakteristik

GKS 1600 ist ein Programmsystem zur Behandlung der grafischen Ausgabe auf Plotter und Display und zur grafischen Eingabe vom Display. Es ermöglicht dem Anwendungsprogrammierer eine weitgehend geräteunabhängige Programmierung. Mittels Einhalten der Nutzerschnittstelle GK5 1600 bei Einbeziehung weiterer grafischer Geräte wird es möglich, neue grafische Geräte ohne wesentliche Änderung der Anwenderprogramme zu betreiben. Die daraus resultierenden Effekte durch Einsparung von Programmierarbeit beim Anwender sind beachtlich und Hauptanliegen des GKS 1600.

Das GKS 1600 wird in Anlehnung an den Standardentwurf der ISO — Graphical Kernel System (GK5), Version 7.0 — entwickelt [2]. Dem Nutzer wird eine Menge von FORTRAN-Unterprogrammen [3] bereitgestellt, mit denen folgende Aufgaben gelöst werden können:

- zweidimensionale Ausgabe (Strecke, Marker, Text, Kreis) von gleichzeitig mehreren Arbeitsstationen mit Auswahl der verfügbaren Darstellungsmöglichkeiten (Linienarten, Farben, Schriftgrößen u. a.)
- grafische Eingabe
- Bildsegmentierung
- Bildfensterbildung und Transformation des Bildes (windowing, clipping)
- Transformation von Bildelementen (Translation, Rotation, Maßstabsänderung)
- Speicherung grafischer Informationen in geräteunabhängiger Form (Metafile, Segmentspeicher).

GKS 1600 wird realisiert in FORTRAN 1600 und teilweise in MACRO 1600 unter Nutzung der POS-Technologie (TESO 1600). Für den AKT A 6454 wird die Version 1 des GK5 1600 bereitgestellt. Die Beschreibung der GKS-Schnittstelle steht den Anwendern seit Mitte 1982 zur Verfügung [3]. Eine erweiterte Version 2 mit Einbeziehung flächenhafter Darstellungen und weiterer grafischer Geräte (wie z. B. hochauflösendes Digitalisiergerät HDG K 6402) ist in Vorbereitung.

2.2. Arbeitsstationen

Eine Arbeitsstation im Sinne von GKS ist die Zusammenfassung von einem grafischen Ausgabegerät (Display, Plotter) und/oder einem oder mehreren grafischen Eingabegeräten (Tastatur, Tablett, Lichtstift, Rollkugel usw.). Eine grafische Arbeitsstation wird vom GKS als ein logisches Gerät behandelt.

Ein Anwendungsprogramm kann gleichzeitig mit mehreren Arbeitsstationen arbeiten (Abb. 2). Mittels Angabe eines Zustandes (geschlossen, geöffnet, aktiviert) bzw. bestimmter arbeitsstationsunabhängiger Attribute (Farbe, Linienart und -breite) können die Bilddarstellung und, falls möglich, der Dialog für jede Arbeitsstation gesteuert werden.

Spezielle GKS-Arbeitsstationen sind das Metafile und der geräteunabhängige Segmentspeicher. Informationen über einzelne Arbeitsstationen sind dem GKS durch Arbeitsstationsbeschreibungstabellen zugänglich.

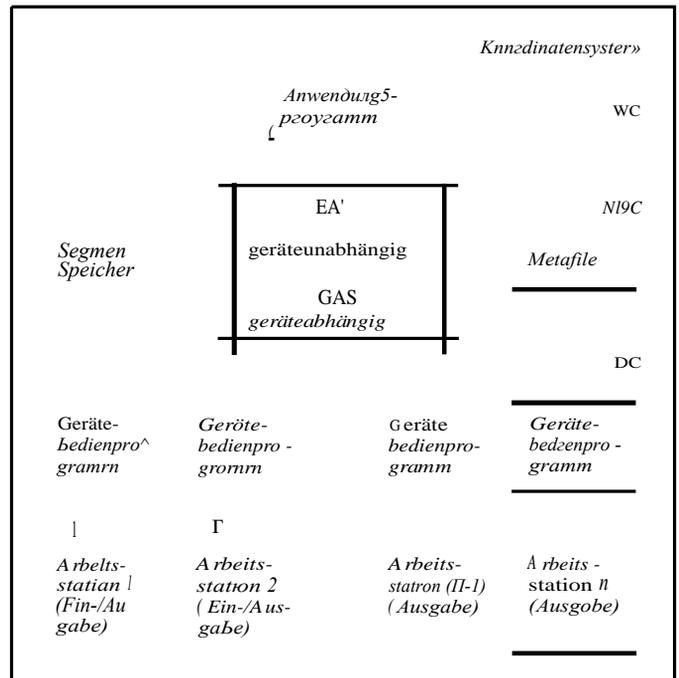


Abb. 2 Arbeitsstationen und Koordinatensysteme des GKS

2.3. Grafische Ausgabe

Die kleinsten grafischen Einheiten, die mit GKS bearbeitet werden können, heißen Elemente. Als Ausgabeelemente stehen dabei zur Verfügung:

POLYLINK	Ausgabe eines Polygons
POLYMARKER	Ausgabe einer Markerfolge
TEXT	Ausgabe eines Texts
GENERALIZED DRAWING	
PRIMITIVE	Ausgabe eines verallgemeinerten Darstellungselementes (Kreis, Kreisbogen).

Zu jedem Ausgabeelement können noch bestimmte Attribute angegeben werden, die die Darstellung der einzelnen Elemente näher spezifizieren (Farbe, Strichart, Schriftrichtung und -größe usw.).

2.4. Grafische Eingabe

Fünf logische Eingabeklassen sind verfügbar, die verschiedene physische Geräte behandeln können:

LOCATOR	Punkteingabe (Rollkugel, Steuerknüppel, Tablett, Spezialtasten in Verbindung mit Cursor)
VALUATOR	Eingabe reelle Zahl (Potentiometer)
CHOICE	Eingabe alternative Auswahl (Funktionstastatur)
PICK	Eingabe Identifikatoren eines Bildelementes (Lichtstift, Spezialtaste in Verbindung mit Cursor)
STRING	Texteingabe (alphanumerische Tastatur).

Jedes logische Eingabegerät kann in drei verschiedenen Modi arbeiten:

REQUEST	Warten, bis eine Eingabe von dem angegebenen Gerät erfolgt ist.
SAMPLE	Der aktuelle Zustand des angegebenen Gerätes wird abgefragt (das heißt, es erfolgt kein Warten auf eine neue Eingabe).

EVENT Die Eingabe wird in eine Eingabewarteschlange eingetragen und kann von dort gelesen werden.

2.5. Segmentierung

Im GKS ist der Aufbau strukturierter Bilder möglich. Jede Darstellung kann aus mehreren Teilbildern (Segmenten) zusammengesetzt sein. Ein Segment wird durch die Funktionen CREATE SEGMENT und CLOSE SEGMENT gebildet. Alle zwischen diesen beiden Funktionen generierten Ausgabeelemente bilden ein Segment, das mit einem eindeutigen Namen gekennzeichnet wird. Der Inhalt des Segments kann nach dem Schließen nicht mehr verändert werden, das Segment kann aber als Ganzes manipuliert (z. B. transformiert, hervorgehoben, identifiziert, sichtbar oder unsichtbar gemacht, gelöscht, in ein anderes em

Mittels Angabe eines PICK IDENTIFIER können einzelne Ausgabeelemente innerhalb eines Segmentes gekennzeichnet werden. Es sind jedoch keine weiteren Manipulationen mit diesen einzelnen Elementen möglich.

Auf den geräteunabhängigen Segmentspeicher lassen sich einzelne Segmente ausgeben. Diese können in andere Segmente eingefügt und in andere Arbeitsstationen übergeben werden. Um Segmente auch bei einer Obertapping in der Darstellung eindeutig identifizieren zu können, erhält jedes Segment eine Priorität.

2.6. Metafile

Das Metafile dient der permanenten Speicherung grafischer Informationen. Gleichzeitig kann damit ein Austausch grafischer Informationen zwischen verschiedenen Anwendungsprogrammen und verschiedenen Gerätesystemen (die natürlich alle die GKS-Schnittstelle benutzen müssen) erfolgen. Das Metafile dient bei GKS der Übernahme der digitalisierten Daten vom DIG 1600. Es wird von GKS wie eine spezielle grafische Arbeitsstation behandelt. Bei der Erstellung wird es als einfaches passives Ausgabegerät behandelt, wobei aber die jeweiligen Funktionen nicht ausgeführt, sondern in codierter Form sequentiell abgespeichert werden. Das Metafile bietet außerdem die Möglichkeit, nichtgrafische Informationen, wie zum Beispiel Kommentare, zu speichern und später wieder zu verwenden. Während eines Programmablaufs können gleichzeitig mehrere Metafiles erstellt werden, die nicht identisch sein müssen.

Bei der Eingabe ist das Metafile eine einfache Eingabearbeitsstation, wobei das Eingeben über die GKS-Funktion INTERPRET ITEM erfolgt. Diese Funktion generiert aus den Metafileinformationen die entsprechenden internen GKS-Funktionsaufrufe. Dabei ist die Abarbeitung wie bei der Ausgabe nur sequentiell möglich. Es können gleichzeitig mehrere Metafiles von einem Anwendungsprogramm als Eingabearbeitsstation bearbeitet werden.

Ein Metafile kann von einem Nutzer entweder nur als Ausgabe- oder nur als Eingabearbeitsstation behandelt werden.

2.7. Koordinatensysteme und Transformationen

Natürgemäß unterscheidet sich das vom Anwender benutzte Koordinatensystem von den jeweiligen Gerätekoordinatensystemen.

GKS unterscheidet deshalb drei verschiedene Koordinatensysteme:

- Weltkoordinatensystem (WC — world coordinate): das vom Anwender benutzte Koordinatensystem
- Normalisiertes Gerätekoordinatensystem (NDC — normalized device coordinate): das vom GKS benutzte anwendungs- und geräteunabhängige Koordinatensystem im Bereich $[0,1] \times [0,1]$
- Gerätekoordinatensystem (DC — device coordinate): das vom jeweiligen Gerät benutzte gerätespezifische Koordinatensystem.

Der Anwender kann nun sowohl achsenparallele, lineare Abbildungen von WC nach NDC als auch pro Arbeitsstation solche von NDC in DC definieren. Dazu dienen die Funktionen SET WINDOW und SET VIEWPORT bzw. SET WORKSTATION WINDOW und SET WORKSTATION VIEWPORT, wobei im ersten Fall wahlweise und im zweiten immer ein „Clipping“ ausgeführt wird. Das „Clipping“ bei der Abbildung von WC in NDC kann vom Nutzer durch SET CLIPPING INDICATOR gesteuert werden.

Die Abbildungen lassen sich während eines Programmlaufs mehrfach verändern.

2.8. Struktur

GKS 1600 enthält einen geräteunabhängigen und einen geräteabhängigen Teil (gerätespezifische grafische Schicht), deren Zusammenspiel die Geräteunabhängigkeit der Anwendungssystemunterlagen sichert.

Die Schnittstelle zwischen beiden Teilen ist eindeutig definiert. An ihr wird eine bestimmte minimale Intelligenz der grafischen Geräte vorausgesetzt. Ist diese nicht gewährleistet, wird im geräteabhängigen Teil ein Intelligenzausgleich vorgenommen. Der geräteabhängige Teil erzeugt aus den vom geräteunabhängigen Teil übergebenen internen GKS-Befehlen die Befehle des jeweiligen Gerätes.

Diese Struktur des GKS erleichtert den Anschluß neuer grafischer Geräte.

3. DIG 1600

Dem Nutzer werden Programme bereitgestellt, die ihm ohne weiteren Programmieraufwand das Digitalisieren am hochauflösenden Digitalisiergerät ermöglichen.

Folgende Funktionen werden unterstützt:

- punktweises und kontinuierliches Digitalisieren
- automatische Punktauswahl beim kontinuierlichen Digitalisieren
- freie Positionierung des Menüfeldes
- Rundung der erfaßten Koordinaten in wählbares Nutzerraster
- Referenzpunktaufnahme und Korrekturrechnung
- Koordinatentransformation
- Eingabe der Z-Koordinate
- Definition und Eingabe von Makros.

Die erfaßten Daten werden in der Struktur des GKS-1600-Metafiles auf eine sequentielle Datei ausgegeben. Das ermöglicht eine Bearbeitung des digitalisierten Daten mit GKS 1600, zum Beispiel Ausgabe auf einen Plotter.

4. G BS 1600

Mit dem Geometriebaustein wird eine Menge von Programmen und Unterprogrammen bereitgestellt, die den Nutzer bei der Entwicklung von Anwendungsprogrammen für die technische Vorbereitung der Produktion, vorrangig in der metallverarbeitenden Industrie, unterstützen. GBS 1600 realisiert als Baustein die Behandlung geometrischer 2-dimensionaler Darstellungen. Folgende Funktionen werden realisiert:

- rechnerinterne Darstellung geometrischer Objekte
- interaktive Arbeit zur Eingabe und Modifikation geometrischer Objekte
- Zusammenfügen komplexer geometrischer Objekte aus vorgegebenen katalogisierten Basiselementen (Punkte, Linien, Kreisbögen, Flächen), wobei sich diese Elemente berühren oder durchdringen können
- Berechnung der Sichtbarkeit
- Erzeugung von normgerechten Zeichnungen.

Als Schnittstelle gegenüber den Anwenderprogrammen wird die Geometrieschnittstelle definiert. Die grafische Ein- und Ausgabe erfolgt mit GKS 1600.

Mit GKS 1600 werden konstruktionsbezogene und technologiebezogene Sprachelemente für die Beschreibung von Objekten bereitgestellt.

Literatur:

- /1/ Tagungsmaterial IKM 82, Leipzig (1982)
- /2/ Draft International Standard ISO/DIS. Graphical Kernel System. Funktional Description. ISO TC97/SC5/WG2 Nr. 117, Januar 82
- 13, Unterprogramme zum GKS 1600 für FORTRAN 1600. Arbeitsunterlage des Entwicklers, VEB Robotron ZFT
- 14 interne Arbeitsunterlagen des VEB Robotron AFT zum hochauflösenden Digitalisiergerät und zu GBS 1600
- 151 Bernd, E.; Herbst, H.; Kotzauer, A.: Auswertung einer Implementierung einer Testversion von GKS-Routinen auf der Basis GIPS/ESER. Bericht der Wilhelm-Pieck-Universität Rostock. WPU DIGRA Nr. 105

Programmsystem MEPRIM für PRS A 6491 /A 6492

Arndt Lindner
Akademie der Wissenschaften der DDR,
Zentralinstitut für Kernforschung

1. Ziel der problemorientierten Systemunterlagen

In Prozeßrechneranlagen A 6491/A 6492 werden die Aufgaben der Prozeßkopplung von Mikrorechnern robotron K 1520 (ursatad 5000) wahrgenommen. Für diesen Einsatz wurde als Steuerprogramm das Echtzeitsteuerprogrammssystem EIEX 1521 entwickelt /1/.

Das hier vorgestellte Programmsystem MEPRIM dient zur Erfassung und Primärdatenverarbeitung von Meßwerten. Es arbeitet unter Steuerung des EIEX 1521. Der Entwurf des Systems berücksichtigt besonders den Einsatz für Prozeßzustandsanzeige und Prozeßüberwachungsaufgaben /2, 3/, das heißt Einsatzfälle mit kontinuierlichen zeitzyklischen Bearbeitungsanforderungen.

Die problemorientierten Systemunterlagen können mittels Generierung an unterschiedlichste Einsatzfälle angepaßt werden. Die gesamte Konzeption läßt nahezu beliebige Erweiterungen durch anwendereigene Teile zu.

2. Programmkonzeption

Der Programmentwurf trennt Meßwerterfassung und Primärdatenverarbeitung vollständig.

Ausgehend von der Definition einer Datenbasis wurden Programmmoduln erarbeitet, die Manipulationen innerhalb dieser Datenbasis ausführen. Die Datenbasis gliedert sich in Primärdatenbasis und Datenbasiserweiterungen und besteht aus Datentypen. Datenbasiserweiterungen können auch vom Anwender angefügt werden. Innerhalb der Meßwerterfassung kommt dem Datentyp „Meßwert“ und innerhalb der Primärdatenverarbeitung dem Datentyp „Kenngröße“ eine zentrale Bedeutung zu.

Mit der Trennung von Meßwerterfassung und Primärdatenverarbeitung werden eine Reihe von Vorteilen erreicht. So entfallen für zusammengesetzte Werte (Kenngrößen), zum Beispiel die thermische Leistung eines Kernreaktors, die Definitionen von Pseudomeßstellen. Andererseits können Korrekturmeßstellen, zum Beispiel Vergleichsmeßstellen bei Temperaturmessungen mittels Thermoelement, als solche behandelt werden und brauchen nicht extra in die Kette der Primärverarbeitung aufgenommen zu werden. Sie erfordern also nicht die Definition einer speziellen Kenngröße.

Alle Hauptprogramme des Systems MEPRIM werden von EIER als Tasks behandelt und müssen nach bestimmten Gesichtspunkten in das Vorrangssystem des EIEX eingeordnet werden. Bis auf wenige Ausnahmen sind alle Unterprogramme im System



beiheft 4/1981

Rechnergestützte McRdatenerfassung und -verarbeitung

Das Heft verdeutlicht den Problemkreis der Verbindung von Geräten der Meßdatenerfassung und -verarbeitung mit den notwendigen Technologien und Programmen. Es vermittelt insbesondere Erfahrungen aus der Praxis für die Praxis.

Ihre Bestellung richten Sie bitte an den
Verlag Die Wirtschaft, Abt. Vertrieb,
1055 Berlin, Am Friedrichshain 22.

MEPRIM als wiedereintrittsfähige Unterprogramme geschrieben. Dadurch können sie ohne spezielle Unterprogrammorganisation von beliebigen Tasks benutzt werden.

Die problemorientierten Systemunterlagen arbeiten ohne problemorientierte Meßstellennummern bzw. Kenngrößennummern und ohne problemorientierte Datentypbezeichnungen. Der Zugriff zu Datentypen erfolgt im allgemeinen über Datentypnummer und Nummer des Datentypenelements. Während der Programmgenerierung können auf der Ebene von Assemblerprogrammen Namen für Datentypenelemente benutzt werden. Durch den Einsatz von Makros läßt sich dabei eine leichtverständliche Programmstruktur erreichen.

3. Datenbasisdefinition

Die Datenbasis setzt sich aus einer abgeschlossenen Primärdatenbasis und Datenbasiserweiterungen zusammen.

Tab. 1 enthält die Angaben zur Primärdatenbasis.

Tab. 1 Angaben zur Primärdatenbasis

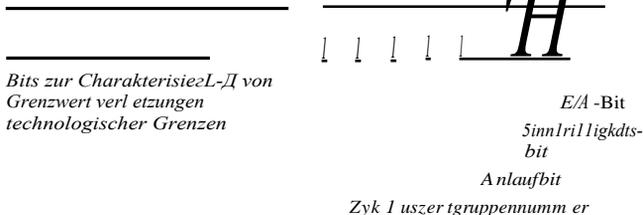
Datentyp	Länge eines Datentypenelements (Byte)	Format	Inhalt
ADC-Wert	2	Festkomma	ADC-Werte
Meßwert	3	Gleitkomma	Meßwerte
Meßwertstatus 1	1	S 1	Statusinformationen des Meßwertes
Kenngröße	3	Gleitkomma	Kenngrößen
Kenngrößenstatus	2	S 2	Statusinformationen der Kenngrößen
Verarbeitungsblockliste	2	Adresse	Adressen der Verarbeitungsblöcke der Kenngrößen
Parameterblockliste	2	Adresse	Adressen der Parameterblöcke der Kenngrößen

Die Formate S 1 und S 2 sind folgendermaßen aufgebaut:

Format S 1



Format S 2



Tab. 2 Datentypen zur Standarddatenbasiserweiterung

Datentyp	Länge eines Datentypenelements (Byte)	Format	Inhalt
Minimum	6	Gleitkomma	Minimalwert und Zeit seines Auftretens
Maximum	6	Gleitkomma Zeit	Maximalwert und Zeit seines Auftretens
Trend	variabel	Strukturdaten, Gleitkomma	Trendlisten von Kenngrößen
Variable 1	1	Integer	Einbyte-Variable (z. B. Zähler)
Variable 2	2	Integer oder Sedezimal	Zweibyte-Variable (z. B. Zähler, Adressen)
Variable 3	3	Gleitkomma	Dreibyte-Variable (z. B. Dimensionierungskonstanten, Zwischenergebnisse)

Die Standarddatenbasiserweiterung besteht aus sechs Datentypen (Tab. 2). Vom Anwender können, wenn notwendig, weitere Datentypen definiert werden.

4. Leistungsumfang der Meßwerterfassung

Das Meßwerterfassungsprogramm fragt die Meßstellen in fortlaufender Reihenfolge ab, führt einen Sinnfälligkeitstest durch und stellt die Meßwerte in der Datentypenliste Meßwert für die weitere Bearbeitung zur Verfügung. Der Abfragezyklus ist für alle Meßstellen gleich. Er muß ein ganzzahliges Vielfaches des Grundtaktes der EIEX-Uhr sein.

Meßstellen können programmtechnisch ein- und ausgeschaltet werden. Ausgeschaltete Meßstellen werden vom Meßwerterfassungsprogramm nicht bearbeitet.

Für den Sinnfälligkeitstest sind alle Meßstellen in maximal acht verschiedene Ausfallgruppen einzuordnen. Jede Ausfallgruppe ist durch ein Grenzenpaar (untere und obere Ausfallgrenze) und eine Anzahl zulässiger Grenzwertverletzungen charakterisiert. Wird die Anzahl zulässiger Grenzwertverletzungen in aufeinanderfolgenden Abfragezyklen überschritten, liegt ein Meßstellenausfall vor. Meßstellenausfälle werden als Alarm angezeigt. Die Rückkehr zu sinnfälligen Daten wird analog behandelt. Während zulässiger Grenzwertverletzungen bleibt der letzte sinnfällige Meßwert als Ersatzwert erhalten.

Bei Meßstellenausfall wird ein Zyklus nach der Alarmlösung der nicht sinnfällige Wert übernommen.

Da der Alarm bereits einen Zyklus vor der Übernahme des nicht sinnfälligen Meßwertes ausgelöst wird, sind Reaktionen zur Alarmbehandlung möglich, die die Übernahme des rocht sinnfälligen Meßwertes verhindern (z. B. Ausschalten der Meßstelle).

Bei Einsatz des Karteneinschubs AE-DV (Analogwerteingabe-Datenvorverdichterkarte) kann das Meßwerterfassungsprogramm entfallen, da dessen Aufgaben von dem Karteneinschub wahrgenommen werden.

5. Leistungsumfang der Primärdatenverarbeitung

Die gesamte Primärdatenverarbeitung wird mit Hilfe einer Drei-Byte-Gleitkommaarithmetik ausgeführt. Diese Arithmetik wurde speziell unter dem Gesichtspunkt schneller Rechenzeiten entwickelt [4]. Charakteristische mittlere Rechenzeiten sind in Tab. 3 zusammengestellt.

Operationen mit Null werden *speziell* getestet und wesentlich schneller ausgeführt.

Zum Zweck der Primärdatenverarbeitung wird ein umfangreicher Satz von Standardroutinen bereitgestellt, der jederzeit um anwendereigene Routinen erweitert werden kann. Dabei empfiehlt es sich, Standardroutinen so weit wie möglich zum Aufbau von anwendereigenen Routinen einzusetzen.

Beim Aufbau von anwendereigenen Routinen müssen bestimmte Regeln bei der Registerbelegung beachtet werden.

Alle Standardroutinen werden mit verschiedenen Möglichkeiten der Parameterübergabe angeboten.

Es ist deshalb vorteilhaft, sich während der Einsatzvorbereitung für eine Variante der Parameterübergabe zu entscheiden und diese Variante durchgängig anzuwenden. Die Verarbeitung erfolgt in linearen Blöcken, die mittels Aneinanderreihung von Verarbeitungsroutinen entstehen. Mit Hilfe der Datentypen „Verarbeitungsblockliste“ und „Parameterblockliste“ wird jeder Kenngröße ein Verarbeitungsblock und ein Parameterblock zugeordnet. Da sämtliche Verarbeitungsroutinen wiedereintrittsfähig sein müssen und auf Grund der Trennung von Verarbeitungssteuerung (Verarbeitungsblock) und Parametern der Verarbeitung (Parameterblock) können Kenngrößen mit gleichen Verarbeitungsalgorithmen einen Verarbeitungsblock mehrfach nutzen. Während der Verarbeitung werden die Register B, H, L als Gleitkommaakku genutzt. Ein Verarbeitungsblock beginnt also immer mit einer Routine zum Laden *des* Gleitkommaakkus oder mit einer Routine, die diesen Vorgang einschließt (z. B. Dimensionierungsroutinen).

Das Abspeichern der resultierenden Kenngröße wird von MEPRIM automatisch realisiert. Während der Verarbeitung auftretende Fehler werden von MEPRIM erkannt und als Alarm angezeigt. Durch den Einsatz von Fehlermasken kann bei auftretenden Fehlern das Abspeichern bzw. das Setzen des Sinnfälligkeitsbit im Datentyp „Kenngrößenstatus“ (als Kennzeichen, daß die Kenngröße fehlerbehaftet ist) unterdrückt werden.

Alle Kenngrößen werden in Zykluszeitgruppen (maximal 32) eingeordnet. Mittels Vorrangorganisation des Echtzeitsteuerungssystems EIER 1521 wird gewährleistet, daß die Bearbeitung von Kenngrößen mit großen Zykluszeiten durch die Bearbeitung von Kenngrößen mit kleinen Zykluszeiten unterbrechbar ist. Für folgende Aufgaben werden Standardroutine angeboten:

— Laden von Meßwerten, Kenngrößen und Variablen in den Gleitkommaakku (Register B, H, L) und in die Register C, D, E

Tab. 3 Rechenzeiten der Drei-Byte-Gleitkommaarithmetik

Operation	mittlere Rechenzeit [µs]
Addition	140
Subtraktion	140
Multiplikation	440
Division	690
Quadratwurzel	870

- Abspeichern von Variablen aus dem Gleitkommaakku
- Sinnfälligkeitsprüfung von Meßwerten und Kenngrößen (Auswertung des Sinnfälligkeitsbit in den Statuslisten)
- Arithmetische Operationen (Addition, Subtraktion, Multiplikation, Division, Quadratwurzel)
- Glättung
- Summenbildung, Mittelwertbildung und Varianz
- Grenzwertüberwachung
- Bestimmung von Extremalwerten mit Abspeicherung von Zeitmarken
- Trendverfolgung.

6. Serviceroutinen

Zur Bedienung des Systems MEPRIM sowie für den Zugriff zu Datentypen werden Serviceroutinen angeboten. Sie können als Unterprogramme von alien Tasks aufgerufen werden. Es handelt sich um wiedereintrittsfähige Unterprogramme. Die Übergabe von Parametern erfolgt in Registern. Als Parameter treten zum Beispiel Meßstellennummern, Kenngrößennummern, Datentypnummern usw. auf.

Es werden alle Register außer AF gerettet.

Falsche Parameter werden erkannt. In diesem Fall wird die Serviceroutine nicht abgearbeitet, und bei Verlassen der Routine wird das Carry-Flag gesetzt.

Serviceroutinen stehen für folgende Zwecke zur Verfügung:

- Bin- und Ausschalten von Meßstellen und Kenngrößen
- Setzen und Rücksetzen des Sinnfälligkeitsbit bei Meßstellen und Kenngrößen
- Setzen und Rücksetzen des Anlaufbit bei Kenngrößen
- Ändern der Ausfallgruppennummer bei Meßstellen
- Ändern der Zykluszeitgruppennummer bei Kenngrößen
- Ändern von unterer und oberer Ausfallgrenze und Zahl zulässiger Grenzwertverletzungen bei Ausfallgruppen
- Zugriff zu Datentypen (lesen und schreiben).

7. Behandlung von Fehler- und Systemmitteilungen

Während der Arbeit des Programmsystems MEPRIM können eine Reihe von Fehler- und Systemmitteilungen (Alarme) erzeugt werden. Alle derartigen Meldungen werden in einen Alarmpuffer eingetragen. Dieser ist als Ringpuffer organisiert. Seine Größe ist genummerungsabhängig. Das Kennzeichenbit einer Meldung enthält Informationen über deren weitere Behandlung.



Kennzeichen für Art d. Meldung
 interne Reaktion (1-ja, 0-nein)
 Weitergabe der Meldung (7-ja, 0-nein)

Die Zuordnung von Art der Meldung und Kennzeichen ist generierbar. Bin Maßnahmeprogramm wertet die Eintragung des Alarmpuffers aus. Für die interne Reaktion des Maßnahmeprogramms sind vom Anwender Behandlungsunterprogramme zu schreiben. Interne Reaktionen **auf Alarme** können zum Beispiel das Ausschalten von Meßstellen oder Eintragen von Ersatzwerten sein. Dem Behandlungsunterprogramm werden alle Angaben des Alarms als Parameter übergeben.



Mikrorechner-Entwicklungssystem robotron A 5601, das sich auch für die Generierung der problemorientierten Systemunterlagen für die Prozeßrechnerysteme A 6491/A 6492 eignet. Werkfoto

Bei der Weitergabe der Meldung wird diese um die Uhrzeit ergänzt und in einen zweiten Puffer eingetragen. Dieser Puffer wird vom Maßnahmenprogramm mit Hilfe eines EIEX-Rufes SLRE (Slaverequest) zur Übertragung an den Masterrechner bereitgestellt /5/.

Der Anwender hat die Möglichkeit, die Fehler- und Systemunterteilungen von Anwenderprogrammen in die vorhandene Behandlung einzuordnen.

B. Generierung

Die problemorientierten Systemunterlagen sind generierungsfähig und damit gezielt an spezielle Einsatzfälle anpassbar. Sie stellen darüber hinaus bestimmte Anforderungen an die Generierung des Echtzeitsteuerprogrammsystems EIEX 1521. Diese betreffen die Generierung des Steuerkerns, der Zeitorganisation, der Prozeß- und Unterprogrammorganisation.

Da die Programme des Systems MEPRIM mit dem Haupt- und dem Tauschregistersatz des Mikroprozessors arbeiten, ist der parallele Betrieb von MEPRIM und Floppy-Disk-Laufwerken nicht möglich.

Die Generierung der problemorientierten Systemunterlagen erfolgt in zwei Teilen. Teil 1 umfaßt die Generierung der Programme, die Auswahl der Verarbeitung- und Service-routinen und die Generierung der Größe der Datentypenlisten. Im zweiten Teil der Generierung werden die Belegung der Datentypenlisten, das heißt die Art der Verarbeitung und die Parameter der Verarbeitung festgelegt.

Die Festlegungen des ersten Teils der Generierung sind im allgemeinen während des Programmablaufs nicht änderbar.

Für die Generierung werden als Hilfsmittel die bedingte Assemblierung und die Makrotechnik eingesetzt. Sie kann auf einem Mikrorechnerentwicklungssystem robotron A 5601 (MRES 20) durchgeführt werden.

K-1600-Fileprozessor unterstützt Datenbanksysteme

Dr. Reinhard Krien
VEB Robotron-ZFT Dresden

1. Zielstellung und Definition

Kleinrechner haken neben ihren traditionellen Anwendungsgebieten in der Produktions- und Fertigungssteuerung; bei Überwachungsaufgaben sowie wissenschaftlich-technischen Rechnungen eine zunehmende Verbreitung auf dem Gebiet der nicht-numerischen und ökonomischen Datenverarbeitung gefunden. Besonders bei den letztgenannten Aufgaben besitzt die Ab-speicherung, Verwaltung und das Wiederauffinden umfangreicher Datenbestände M Form von Dateien bzw. Files auf Platten-speichern eine große Bedeutung.

Dieses Vordringen der Kleinrechner in Aufgabengebiete, die bisher weitestgehend durch EDVA abgedeckt wurden, wird auch in Qualität und Quantität der für Kleinrechner bereitgestellten Systemunterlagen deutlich. Die modernen Betriebssysteme der Kleinrechner (z. B. OS/RW der CM 3/4 bzw. MOOS 1600 des Mikrorechnersystems robotron K 1600) enthalten unter anderem Programmsysteme, die es den verschiedenen Nutzern ermöglichen, auf Makroassembler-Niveau bzw. dem Niveau einer höheren Programmiersprache Dateien virtuell einzurichten und auf die gewünschten Sätze bzw. Blöcke zuzugreifen, ohne den physischen Ort der Sätze bzw. Blöcke der Datei zu kennen.

Dieser Nutzerkomfort bei der Datenverwaltung stößt aber besonders bei IOeinrechnern an Effektivitätsgrenzen, die im Absinken des Systemdurchsatzes, in Laufzeitineffektivitäten und Aufgabenbegrenzungen aufgrund zu geringen Hauptspeicherplatzes spürbar werden.

Mit dem Fileprozessor wird deshalb das Ziel verfolgt, die Datei- bzw. Filearbeit im System K 1600 zumindest auf dem Niveau der „Basiszugriffsmethoden“ zu effektivieren. Es wird angestrebt, den Leistungsbereich des K 1600 bei vorausgesetztem hohen Anteil an Dateiarbeit insbesondere bezüglich Speicherplatz und Durchsatz auszuweiten. Weiterhin sollen Unterstützungen bei der Implementierung leistungsfähiger, nutzerfreundlicher Kleinrechner-Datenbanksysteme (DBS) gegeben und mit der Mehrprozessorstruktur der Schritt in Richtung Datenbankmaschine (DBM) unterstützt werden.

Diese Zielsetzung wird auch durch die internationalen Entwicklungstendenzen auf diesem Gebiet motiviert. Diese lassen sich so zusammenfassen:

- Zunehmende Hardwareunterstützung von DBS mit dem Ziel der Einführung von Datenbankmaschinen
- gerätetechnische Basis der DBM sind Multimikroprozessorsysteme, Mikroprozessor-Speicherkopplungen und Kleinrechner

- Funktionsspektrum der DBM: auf der einen Seite Effektivierung des externen Speicherzugriffes, auf der anderen Seite Übernahme der Funktionen von „Spitzen-DBS“
- Forderung nach höherer Nutzerfreundlichkeit der DBS sowie ihres Einsatzes auf kleinen Rechenanlagen führen auf das Problem der gerätetechnischen Unterstützung von DB-Funktionen.

Unter Berücksichtigung der genannten Zielsetzung und der Entwicklungstendenzen soll unter dem Fileprozessor (FP) ein universeller Prozessor des Systems K 1600 mit zugeordnetem Arbeitsspeicher verstanden werden, der mit einem K-1620/k-1630-Verarbeitungsprozessor (VP) über eine Koppereinheit BKE 1600 in Verbindung steht. Mittels Aufteilung von Systemroutinen des MOOS 1600 auf beide Prozessoren und durch Systemprogrammerweiterungen für den Austausch von Steuerdaten führt der FP die im Nutzerprogramm auf dem VP angeforderten Dateizugriffe des FCS 1600 auf dem Niveau des virtuellen Blockzugriffs nach Übermittlung von Steuerdaten selbständig aus.

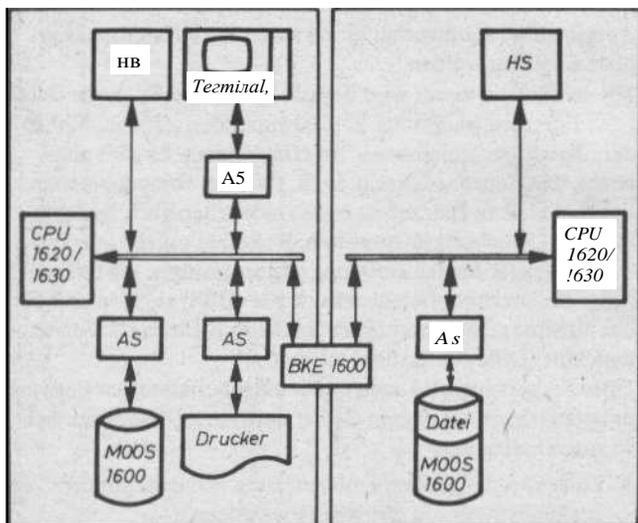
2. Gerätetechnische Basis des Fileprozessors

Die Gerätekonfiguration für den Fileprozessor zeigt Abb. 1. Die K-1620- bzw. k-i 630-CPU sind über eine Buskoppereinheit verbunden. Die Koppereinheit ermöglicht es, im DMA-Verkehr Steuerworte bzw. Datenblöcke zwischen den beiden Hauptprozessoren auszutauschen. Die Koppereinheit wird als peripheres Gerät über ein Gerätetreiberprogramm, das in beiden Prozessoren vorhanden ist, angesprochen. Das Ansprechen der Koppereinheit und die logische Einordnung der über die Koppereinheit ausgetauschten Daten in die jeweiligen Programmabläufe muß im allgemeinen auf Nutzerniveau erfolgen bzw. beim FP durch die entsprechenden Steuerroutinen.

An den FP sind die Platteneinheiten für die Nutzerdateien und auch für die Systemprogramme (die Platteneinheit am VP enthält die Systemprogramme) und die für die Nutzung nötige Peripherie angeschlossen (Ein-Ausgabe-Geräte, Terminal, Lochbandstationen usw).

Verarbeitungsprozessor

Fileprozessor



Nutzerprogramm mit Dateizugriff (ferner FP-Steuerroutinen, FCS-Read

FCS; ACP-Task; FP-Steuerroutinen

Abb. 1 Hardwarekonfiguration des Fileprozessors

3. Fileprozessor-Simulationslösung

3.7. Organisation des Plattenzugriffes im MOOS 1600

Im MOOS werden Plattendateien physisch in Blockeinheiten der Länge 512 Byte abgespeichert, wobei für den Zugriff (z. B. Lesen oder Schreiben) die physische Plattenadresse des Blockes im entsprechenden Plattengerätregister bereitzustellen ist. Quellmoduln, Objektmoduln bzw. Tasks werden ebenfalls als Dateien auf der Platte gespeichert. Die Plattenorganisation erfolgt aber für den Nutzer virtuell.

Dieser spezifiziert die entsprechende Datei durch [UIC]: filename.filetyp; versionsnummer und das Betriebssystem führt vermittels des ACP-Task (auch Dateiprozessor genannt) die physischen Dateizugriffe, z. B. das Laden eines Task in den FIS aus. Der Nutzer wird eine Datendatei im allgemeinen auch virtuell einrichten, indem er die Dienste der zum MOOS 1600 gehörigen Dateizugriffsroutinen FCS 1600, bzw. bei höheren Anforderungen bezüglich des Satzzugriffes die POS RMS 1600 oder perspektivisch das K-1600-Datenbanksystem DABA 1600 nutzt.

Die Transformation des virtuellen Blockzugriffes in den physischen Blockzugriff gemäß physischer Block-Nummer wird vom FCS im Zusammenspiel mit dem ACP-Task und den Parametern des Dateibeschreibungsblockes (FDB), mit dem der Nutzer seine Datei definiert und beschreibt, ausgeführt.

Beim RMS 1600 bzw. auch beim DABA 1600 wird nach analogen Prinzipien verfahren.

Ein erster Zugriff auf eine Nutzerdatei erfordert meist mehrere Plattenzugriffe, da zur Transformation des virtuellen in den physischen Zugriff eine Hierarchie von Datei-Kennsätzen bzw. Verzeichniseinträgen mit entsprechenden Zwischentransformationen zu durchlaufen ist.

Anhand von Abb. 2 wird das Zusammenspiel der verschiedenen Routinen und Tasks bei einem virtuellen Blockzugriff betrachtet. Es wird gleichzeitig gezeigt, ab welcher Schnittstelle der Fileprozessor die Organisation des Plattenzugriffes übernimmt. Im Bild ist eine typische Speicheraufteilung eines K 1620 für einen Nutzertask, der mit dem FCS 1600 arbeitet, dargestellt. Die FCS-Routinen gehören zum Nutzertask und werden während der Taskbildung mit dem Nutzerobjektmodul verbunden. Ein Beispiel soll ein Block einer Plattendatei durch ein READ gelesen werden. Durch das READ S-Makro wird die entsprechende Routine des FCS angesprochen (1), die weitere Routinen initialisiert, bis am Ende ein Anweisungsparametersock entsteht (2). Der Anweisungsparametersock enthält alle vom Betriebssystem benötigten Informationen, um den Plattenzugriff auszuführen. Er wird über den EMT-377-Befehl an die Exekutive MOEX 1600 übergeben (3).

Nach Durchlaufen verschiedener Exekutive-Routinen enthält der Plattentreiber die Steuerung (4). In Zusammenarbeit mit dem Dateiprozessor wird die Transformation virtuelle in logische Block-Nr. ausgeführt (5) und der Plattentreiber setzt die logische Block-Nr. mittels direkter Zuordnung in die physische Plattenadresse um (6).

Nach dem Laden der Gerätregister (7) wird der gewünschte Plattenzugriff ausgeführt (8).

Bei Einsatz des FP müssen der beschriebene Ablauf des Plattenzugriffes geeignet aufgetrennt und die jeweiligen Restroutinen in Verbindung mit weiteren, die FP-Arbeit steuernden Programmoduln auf beide Prozessoren verteilt werden.

Im Falle eines READ- bzw. WRITE-Makros, das hext beim Le-

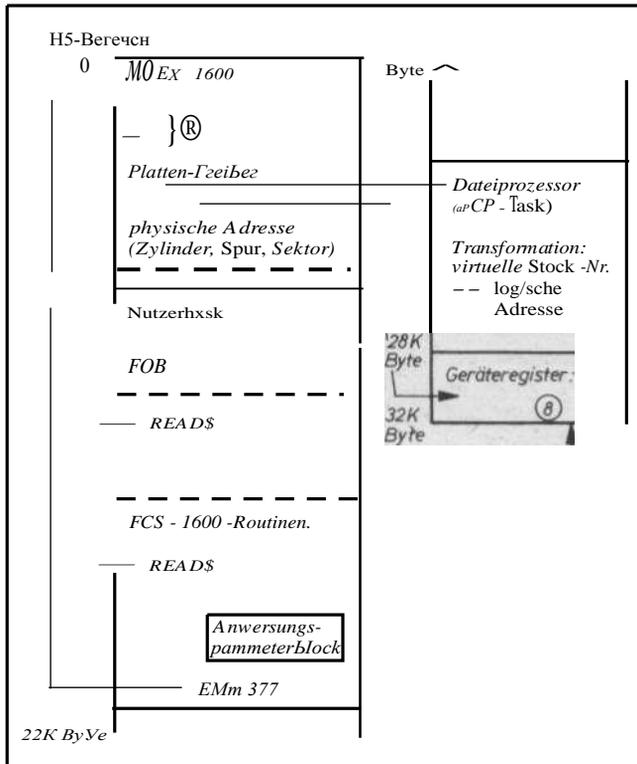


Abb. 2 Ablauf ernet Dateia'dotderung

sen oder Schreiben eines Blockes, wird als Schnittstelle für die Arbeit zum FP der den Blockzugriff einleitende EMT 377 gewählt. Bei der Eröffnung einer Datei löst die entsprechende FCS-1600-

Routine mehrere EMT 377 mit nachfolgendem Plattenzugriff aus. Für die Steuerungsübergabe an den FP ist es deshalb vorteilhafter, nicht den Zeitpunkt des EMT 377 zu wählen, sondern durch „logisches Anheben“ der Schnittstelle, nach einmaliger Steuerungsübergabe, die EMT 377 des OPEN im FP fortlaufend, ohne zwischenzeitliche Rockgabe der Steuerung an den VP, abzuarbeiten.

3.2. Prinzip und Lösungsvariante des Fileprozessors

Der Funktdnsnachweis und erste Untersuchungen zum F ileprozessor wurden vermittels einer Simulationslösung für ein K-1600-Einprozessorsystem vorgenommen. Die weiteren Untersuchungen fanden bzw. finden auf einem K-1600-Zweiprozessorsystem gemäß Abb. 1 statt.

Die Simulationslösung ist so aufgebaut, daü der Übergang vom Einprozessorsystem zum Zweiprozessorsystem durch Austausch von Kommunikationsroutinen möglich wird.

Das Prinzip der Simulationslösung, das Zusammenspiel der beiden Prozessoren und die dafür nötigen Programmkomponenten werden anhand der Abb. 3 diskutiert.

Die 14-K-Partition ist in zwei Sub-Partitionen eingeteilt, so daß das Programm VPROZ in die eine sub-Partition und die Steuer und Zugriffsroutinen des FP als Programm FPROZ in die zweite Sub-Partition als eigenständige Tasks geladen werden. Weiterhin wurde eine von beiden Tasks nutzbare Datenpartition eingerichtet, in welcher der zwischen VPROZ und FPROZ in beiden Richtungen auszutauschende Nachrichtenblock COMBL gespeichert vorliegt. Das Programm VPROZ besteht aus dem eigentlichen Nutzerprogramm mit einer Schreib- und einer Lesedatei, den notwen-

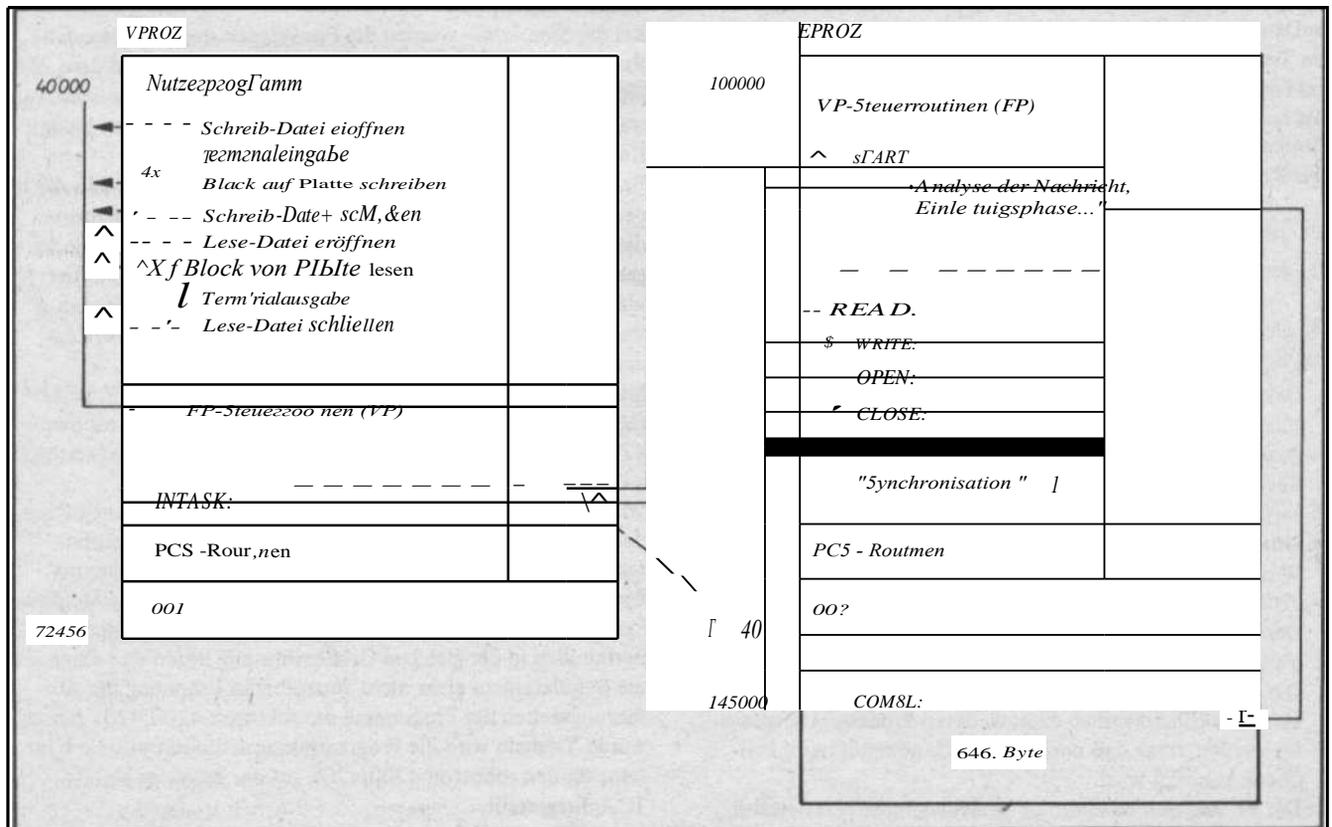


Abb. 3 Zusammenarbeit der Programme VPROZ mid FPROZ (Simulationslösung)

Emulationssystem robotron 4200/4201 – K 1600

Robert Syring
VEB Robotron-Vertrieb Berlin

1. Vorbemerkung

Beim Übergang von einem Rechnersystem auf ein anderes spielen Kompatibilitätsfragen eine bedeutende Rolle, insbesondere dann, wenn das abzulösende Rechnersystem weit verbreitet und über einen relativ langen Zeitraum im Einsatz war. Oft wurden für das abzulösende Rechnersystem umfangreiche Systemunterlagen und Applikationsprogramme geschaffen, die es zu erhalten gilt, da eine Neuentwicklung ökonomisch nicht vertretbar ist.

Für das Erreichen der Programmkompatibilität gibt es zwei Verfahren:

- Simulation
- Emulation.

Bei der *Simulation* werden die Funktionen des abzulösenden Rechners mittels rein programmtechnischer Mittel auf dem Zielrechner nachgebildet. Dabei steigt der programmtechnische Aufwand um so mehr, je größer die Unterschiede zwischen beiden Rechnersystemen sind.

Bei der *Emulation* werden die Unterschiede zwischen den Rechnersystemen durch hardwaremäßige Simulation ausgeglichen, die Kompatibilität wird also mit gerätetechnischen Mitteln hergestellt. Bei modernen Rechnern wird die Emulation auf der Mikroprogrammebene mit Mikrobefehlen, die hardwarenah sind, erreicht. Das Abspeichern der Mikroprogramme erfolgt auf schnellen ROM-Speichern.

Mit der Entwicklung des Mikrorechnergerätesystems K 1600 (MGS K 1600) gilt es, die POS und MO5 der Familie robotron 4200/4201 mit Hilfe einer Kompatibilitätseinrichtung weiter zu nutzen.

Würden bei früheren Übergängen zwischen verschiedenen Rechnerfamilien (z. B. R 300 — robotron 4000) rein programmtechnische Mittel (interpretative Simulation) zur Erzeugung der Programmkompatibilität eingesetzt, scheidet diese Möglichkeit hier aus, da die Operationsgeschwindigkeiten beider Rechnerfamilien in der gleichen Größenordnung liegen und damit die Simulation zu einer nicht vertretbaren Erhöhung der Ausführungszeiten der Programme des robotron 4200/4201 führen würde. Deshalb wird die Programmkompatibilität mittels Emulator für den robotron 4200/4201 auf der Basis des MGS K 1600 hergestellt.

digen FP-Steuerrouinen auf der Verarbeitungsseite, den Routinen des FCS 1600 sowie Hilfsrouinen.

Im Nutzerprogrammteil von VPROZ wird eine Platten-Schreibdatei eröffnet und mit vier Blöcken der Länge 512 Byte über Terminal beschrieben. Nach Schließen der Schreibdatei wird diese als Lesedatei eröffnet, und innerhalb einer Schleife werden die auf Platte geschriebenen Blöcke gelesen und im HS bereitgestellt.

Zur Verwirklichung der FP-Arbeit erfolgt bei OPEN \$ und CLOSE \$ aus dem Makro der Sprung in die FP-Steuerrouinen von VPROZ.

Bei READ \$ bzw. WRITE \$ wird der Absprung in die FP-Steuerrouinen vor dem Emulator-Trag-Befehl EMT 377 ausgeführt.

Die Wirksamkeit der Makros OPEN \$, CLOSE \$, WRRITE \$, READ \$ wird somit in VPROZ unterdrückt. Ihnen sind FP-Steuerrouinen zugeordnet, die zusammen mit weiteren Steuerrouinen den Nachrichtenblock COMBL mit den aktuellen Daten bezüglich der angeforderten FCS-Operation füllen.

Mit dem Rillen von COMBL ist die Arbeitsaufgabe der modifizierten FCS-Routinen in VPROZ abgeschlossen. Durch die Routine INTASK wird die Steuerung an FPROZ abgegeben.

Die wechselseitige Steuerungübergabe zwischen VPROZ und FPROZ erfolgt über gemeinsame Steuerflags.

Das initialisierte Programm FPROZ analysiert die in COMBL übergebene Nachrichtennummer, die die angeforderte FCS-Operation charakterisiert und übergibt die weitere Steuerung an diejenige Routine, die diese Dateioperationen ausführt. Im Programm FPROZ werden dazu der Dateibeschreibungsblock und der Anweisungsparameterblock aus COMBL aktualisiert.

Nach Beendigung der Dateioperation werden von FPROZ die von der Platte gelesenen Nutzerdaten in COMBL zurückgeschrieben und der aktualisierte Nachrichtenblock zusammen mit der Steuerung an VPROZ zurückgegeben.

Nach abschließenden Arbeiten — Bereitstellen der Nutzerdaten im Nutzerpufferbereich und Behandlung der Endmitteilungen — ist die Dateioperation beendet und die nächste kann ausgeführt werden. Bei der Arbeit mit dem Zweiprozessorsystem laufen VPROZ und FPROZ auf jeweils einem eigenen Prozessor. Der Austausch des Nachrichtenblockes erfolgt über die BKE 1600, indem die Kommunikationsroutine INTASK bzw. die entsprechenden Abläufe von FPROZ durch QIO-Rufe bezüglich der Koppereinheit ersetzt werden.

4. Ergebnisse und Schlußfolgerungen

Die Untersuchungen zum FP führten zu folgenden Ergebnissen und Schlußfolgerungen:

- Das Funktionsprinzip des FP und damit die Aufteilung der Filearbeit auf zwei Prozessoren wurde bestätigt.
- Aus Zeitmessungen ergab sich, daß bis zu 90 Prozent der Filearbeit auf den zweiten Prozessor ausgelagert werden kann.
- Mittels Auslagerung von FCS-1600-Routinen können im Verarbeitungsprozessor pro Task 4 bis 10 K Byte Speicherplatz zusätzlich bereitgestellt werden.
- Durch residente Speicherung des ACP-Task im FP ist eine Verkürzung der Programmlaufzeit zu erwarten.
- Die Leistungsfähigkeit eines K-1600-Ein-Prozessorsystems kann mit Hinzunehmen eines weiteren Prozessors ausgeweitet werden, ohne daß noch einmal die gesamte teure Peripherie benötigt wird.
- Der FP läßt sich in Verbindung mit weiteren gerätetechnischen und programmtechnischen Maßnahmen für die Effektivierung der Arbeit von DBS nutzen.



Um die problem- und maschinennormierten Systemunterlagen der zahlreichen Robotron 4200/4201 beim Übergang auf ein K 4600-System weiter nutzen zu können, wurde ein Emulator Robotron 4200/4201 – K 1600 entwickelt, den der Beitrag beschreibt. Werkfoto

2. Der Emulator Robotron 4200/4201 — K 1600

2.1. Einsatzgebiete

Der weitestgehend teils der vorhandenen KRS-Programme ist für den wissenschaftlich-technischen und technisch-ökonomischen Einsatzbereich (WTÜR) entwickelt worden. Aus diesem Grund dient der Emulator in erster Linie zur Weiterverwendung vorhandener WTÜR-Projekte des Robotron 4201.

Außerdem ist eine Nachnutzung neuerer und erprobter KRS-Projekte durch K 1600-Anwender, die einen Rechner des MGS K 1600 mit Emulator besitzen, denkbar.

2.2. Bestandteile des Emulationssystems

Das Emulationssystem besteht aus folgenden Komponenten:

- Emulatorprozessor EMP K 2063
- Hauptspeicher mit 32 K Worten für die auszuführenden KRS-Programme
- Operativspeicher mit 32 K Worten für die Routinen des Emulationssystems
- Unterbrechungssystem
- Nachbildung des programmierten Kanals
- Nachbildung des ESK
- Geräteroutinen
- Simulationsroutinen.

Zum Nachbilden der speicherbezogenen Befehle des Robotron 4200/4201 besitzt der Emulator einen speziellen Prozessor mit dem Befehlsvorrat des Robotron 4200/4201.

Die Nachbildung der peripheren Geräte des Robotron 4200/4201 erfolgt auf den Geräten des MGS K 1600 mit Hilfe von Geräteroutinen, die die E/A-Befehle des Robotron 4200/4201 simulieren. Das Bedienpult des Robotron 4201 wird mittels Simulationsroutine (Monitor) nachgebildet.

Zum Ausführen der Routinen des Emulationssystems wird ein spezieller Operativspeicher verwendet, so dass dem Nutzer der volle Adressraum des KRS-Hauptspeichers mit 32 K Worten zur Verfügung steht. Sowohl der KR5-Hauptspeicher als auch der Emulatorspeicher sind moderne RAM-Halbleiterspeicher.

2.3. Gerätekonfiguration

Das Emulationssystem Robotron 4200/4201 – K 1600 ist für die Arbeit mit dem K 1620 und K 1630 vorgesehen. Voraussetzung für das Abarbeiten von Robotron 4200/4201-Programmen ist dabei ein Emulatorzusatz.

Die Bedienung der unterstützten Robotron 4201-Geräte erfolgt auf der Basis der entsprechenden Geräte der K-1600-Peripherie. Unterstützte Robotron 4201-Geräte, für die keine entsprechenden Geräte in der K-1600-Peripherie vorhanden sind, werden auf artverwandten Geräten der K-1600-Peripherie nachgebildet. Vom Emulationssystem werden folgende Geräte des Robotron 4200/4201 bedient:

- Bedienschreibmaschine
- Lochbandeinheit
- Seriendrucker
- Magnetbandspeicher
- Trommelspeicher
- Kassettenplattenspeicher

Die Bedienschreibmaschine des Robotron 4200/4201 wird auf der Bedieneinheit des K 1600 und der Trommelspeicher des Robotron 4200/4201 auf dem Kassettenplattenspeicher des K 1600 nachgebildet.

2.4. Interruptorganisation

Das Interruptsystem des Robotron 4200/4201 wird mit Hilfe eines Mehrebeneninterruptsystems nachgebildet. Mit Hilfe dieses Interruptsystems verwaltet der Emulator außerdem die Geräteinterrupts der verwendeten Peripherie des MGS K 1600.

Die erforderliche Vergabe der Prioritäten entspricht dabei dem Mehrebeneninterruptsystem des MGS K 1600.

2.5. Monitor

Der Monitor ist ein Kommandoprogramm, das eine zentrale Stellung im Emulationssystem einnimmt. Bestimmte Einstellungen, die am Robotron 4200/4201 gerätetechnisch realisiert werden, erfolgen beim Emulator mit Hilfe des Monitors.

Der Monitor hat folgende Aufgaben:

- Nachbildung des Robotron 4200/4201-Bedienspults
- Nachbildung von Bedienelementen von Robotron 4200/4201-Geräten, die bei K-1600-Geräten nicht existieren
- Eingabe von logischen Anfangswerten für die simulierten Gerätefunktionen
- Anlegen von Registern und Hauptspeicherzellen
- Eingaben in Register und Hauptspeicherzellen
- Übergang in den Laufzustand.

Zur Lösung dieser Aufgaben besitzt der Monitor eine eigene Kommandosprache. Der Syntax dieser Kommandosprache ist an die Syntax der Kommandosprache der Bedieneinheit 1600 angelehnt.

Dabei werden verschiedene Arten von Kommandos unterschieden:

- Schalterkommandos, mit deren Hilfe die Schalter des Bedienspults des Robotron 4201 nachgebildet werden
- Kommandos, mit denen auf Speicherstellen des nachgebildeten

Der 08205-Z-Simulator SIMZ 1600

Annerose Albrecht, Volker Becher, Wolfgang Fleerklotz
ingenieurhochschule Zwickau

- deten KRS-Hauptspeichers und auf die Nachbildung des K- und des A-Registers zugegriffen werden kann
- Startkommandos, mit denen der Prozessor und die Programmausführung ab einer bestimmten Adresse gestartet werden können
- Kommandos, die das Nutzen der Testhilfefunktionen des Emulatormonitors gestatten.

2.6. Bedienung

Ein K 1620 bzw. K 1630 mit Emulatorzusatz kann alternativ in zwei Betriebszuständen arbeiten:

- Abarbeitung von Tasks des MGS **K 1600**
- Abarbeitung von Programmen des Kleinrechnersystems robotron 4200/4201.

Der Übergang zwischen den zwei Betriebszuständen erfolgt mit Hilfe von Bedienkommandos. Der Übergang zwischen den beiden Betriebszuständen bzw. das Laden der Emulatorroutinen und Starten des Emulators wird mit Hilfe der Konsol emulatorroutine des K 1600 bzw. der Executive des Betriebssystems MOOS 1600 ausgeführt.

Der Operator kann mit Kommandos, die über das Bediengerät eingegeben und vom Monitor realisiert werden, mit dem Emulationssystem korrespondieren (siehe Abschnitt 2.5.).

Zur Ausführung von KRS-Programmen unter Steuerung eines KRS-Betriebssystems wird das Betriebssystem nach den Konventionen des robotron 4201 geladen und gestartet. Das Betriebssystem übernimmt dann die Steuerung der weiteren Programmausführung.

3. Generierung

Der Anwender des robotron 4200/4201 hat eine große Anzahl von Anschlussmöglichkeiten für die Anschlußsteuereinheiten des programmierten Kanals zur Verfügung.

Bedingt durch die variablen Anschlussmöglichkeiten der Anschlußsteuerungen und E/A-Geräte beim robotron 4200/4201 muß für den Emulator eine Zuordnung der E/A-Befehlsroutinen und eine Zuweisung der entsprechenden Geräte der K-1600-Peripherie vorgenommen werden. Für den Anwender ist die Zuordnung der Geräte des robotron 4200/4201 zu den Kanaladressen und die Zuordnung der Kanaladressen zu den E/A-Registeradressen der K-1600-Geräte variabel. Der Nutzer führt diese Generierung einmalig vor Beginn der Arbeit mit dem Emulator aus. Er muß dabei folgende Angaben machen:

- welche Geräte des robotron 4200/4201 sollen nachgebildet werden?
- an welche Kanaladressen sollen die Geräte angeschlossen werden?
- welche Maskenbits sind für welche Kanäle gültig?
- Geräteregisteradressen der K-1600-Geräte, die für die Nachbildung der KRS-Geräte verwendet werden.

Diese Informationen werden in die entsprechenden E/A-Tabellen eingetragen. Von dem so generierten Emulationssystem wird ein Speicherabzug hergestellt, der so lange benutzt werden kann, bis eine Änderung der Gerätekonfiguration zu einer neuen E/A-Generierung zwingt.

Bis Mitte der siebziger Jahre wurden ^y am VEB Rechelelektronik Meiüingen/Zella Mehlis Kleinrechenanlagen ^y am Typ C8205 und C8205-Z prod'r'dert. Sie wurden und werden vorwiegend in kleineren Rechenzentren zur Rationalisierung von Leitungs- und Planungsaufgaben eingesetzt, und es entstand dort eine Vielzahl von Programmen.

Viele Anwender des C8205 sind nicht in der Lage, diese Programme in kurzer Zeit rind in vollem Umfang auf die neue Rechentechnik umzustellen. Deshalb wird für sie ein Simulator C8205-Z für KBR A 6401 bzw. A 6402 (SIMZ 1600) geschaffen. Für die Entwicklung des Simulators wurden die Systemunterlagen des 08205-Z 11/, des Simulators 51MC 4200/2/ und SIMZ 4200/4000/3/ verwendet.

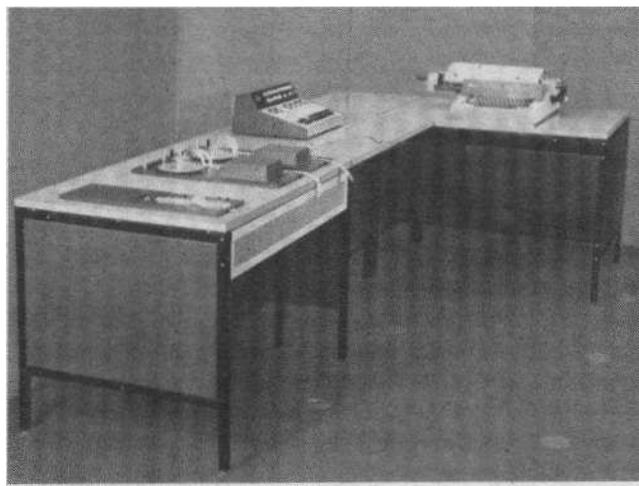
1. Konzept des Simulators 81mZ 1600

Da bereits für die Kleinrechnersysteme robotron 4200/4201 ein Simulator (SIMZ 4201) entwickelt und vom VEB Kombinat Robotron vertrieben wurde, soll für Nutzer dieses Simulators ein problemloser Übergang geschaffen werden. Aus diesem Grund wird die Kommandosprache beider Simulatoren, soweit es gerätetechnisch möglich ist, kompatibel gestaltet. Die (Jbet' einstimmungen zwischen SIMZ 4201 und SIMZ 1600 werden im Punkt 3. dargestellt.

Der zu entwickelnde Simulator nutzt das Betriebssystem MOOS 1600 und belegt den Speicherbereich von Adresse '40000 bis '150000. Diese Speicherbelegung ist insofern wichtig, da sie bei der Generierung die Partitionsfestlegung beeinflusst.

Viele Rechner vom Typ C 8205 sind mit mehr als einem Lochbandleser bzw. -stanzer ausgerüstet. Da im Normalfall die Rechner A 6401 mit nur einer Lochbandeinheit ausgestattet sind, ist eine Simulation der Lochbandeingabe und -ausgabe auf Magnetplatte vorgesehen. Dadurch ist es möglich, Daten bzw. Dateien, die beim C 8205 vom Lochband verarbeitet wurden, auf Magnetplatte zu speichern und anschließend zu verarbeiten. Die verhältnismäßig langsame Lochbandarbeit kann somit teilweise umgangen und die Leistungsfähigkeit des A 6401 besser ausgenutzt werden. Magnetband und Magnetbandkassette werden vom Simulator SIMZ] 6[IO nicht unterstützt.

Zur besseren Überwachung des Programmablaufes beinhaltet der Simulator ein System zur Protokollierung der simulierten C-8205-Befehle und eine wahlweise Ausgabe der erhaltenen Befehlszähler, Operationsregister, Adressregister und Akkumulator. Mittels Geräte- und Kanalzuweisungskommando besteht die Möglichkeit, während der Projektbearbeitung Ausgaben, die für das Terminal (bei C 8205 Schreibmaschinenausgabe) programmiert wurden, dem Drucker zuzuordnen.



Kleinrechner C 8205

Werkfoto

2. Arbeitsweise von S1 MZ 1600

Der Simulator liegt in der Form einer abarbeitungsfähigen Task vor. Diese Task wird mit dem RUN-Konmiando aktiviert und für die Abarbeitung gestartet.

Der Simulator meldet sich auf dem Terminal mit der Ausschrift
SIMZ 1600:

Für den normalen Ablauf wird an dieser Stelle ein N für Neustart erwartet. Ein die Eingabe beendendes Zeichen (CR/LF) braucht hier, wie auch bei einigen anderen Kommandos, nicht gegeben zu werden. Erfolgt an dieser Stelle die Eingabe eines anderen Zeichens, so wird eine Fehlermitteilung ausgegeben und die Kommandoausschrift wiederholt. Liegt eine richtige Eingabe an, so meldet sich der Simulator mit der Ausschrift

#5

und ist zur Eingabe eines Monitorkommandos bereit.

Eine Zusammenstellung aller Monitorkommandos wird in der Tabelle des Abschnittes 3. aufgeführt. Diese Kommandos lassen sich in drei Gruppen einteilen:

1. Kommandos zur Vorbereitung der Abarbeitung
Zum Beispiel #EK, #AR, #AW, #AK
2. Kommandos zur Steuerung der Abarbeitung
Zum Beispiel #PB, #PF, #55
3. Kommandos zum Start und zur Fortsetzung der Abarbeitung
Zum Beispiel #ST, #WE, #GR.

Das vor dem Kommando stehende # kann weggelassen werden. Die Arbeit kann jetzt mit der Eingabe des minimalen Eingabeprogrammes (MEP) fortgesetzt werden, da das kleine Eingabeprogramm (KEP) Bestandteil des simulierten Hauptspeichers ist.

Es ist zu beachten, daß alle Lochbänder vor der Verarbeitung mit einer Lochung Wagenrücklauf/Zeilenschaltung (NL) als Endekennzeichen versehen werden.

Die Speichergröße des C 8205 (Intertrommel Adr. '0000 bis '7777) wird vollständig nachgebildet. Da die Darstellung eines C-8205-Wortes 33 Bit benötigt, erfolgt die Simulation in 3-Worten. Das Umsetzen wird so ausgefüllt:

```
X XXO 000 000 000 000    1. Wort ->Z0, Z1, Z2
0 XXX XXX XXX XXX xxx    2. Wort-->Z3'...,Z17
o XXX XXX XXX XXX XXX    3. Wort->Z18,...,Z32
```

Bei dieser Darstellungsweise werden im Speicher des KBR A 6401 allein für die Simulation des HS 12 K Worte benötigt. Von SIMZ 1600 werden folgende Geräte bedient:

- 2 Lochbandeinheiten
- 1 Terminal
- 2 Drucker
- 1 Wechselpaltenspeicher.

Auf der Magnetplatte werden die simulierte Lochbandein- und -ausgabe sowie die Zusatztrommeln des C 8205-Z nachgebildet.

Die gerätespezifischen Besonderheiten des KBR A 6401 bedingen eine geänderte Zuweisung der Geräte zu den Kanälen des 08205-Z, sowie zum SIMZ4200-Simulator:

Kanalnr.	C 8205		A 6401	
	Eingabe	Ausgabe	Eingabe	Ausgabe
0	SE!	LS2	TT	TT
1	LL1	LS1	LLO	LSO
2	LL2	SA1	DK	TT
3	LL3	5A2	LLI	LS1
4			LLO	SD 0
5			DK	DK
6			TT	SD 1

Die Belegung dieser Kanaltabelle ist mittels Monitorkommando #GE an jeder Stelle generierbar. Sollen die Lochbanddateien auf Magnetplatte gespeichert werden, so muß vor Beginn der Abarbeitung das Kommando #DN G, N, U verwendet werden. Dadurch wird dem System die Gerätenummer des Plattenlaufwerkes, der Name der Datei und der UIC der Datei bekannt gemacht.

3. Kommandoübersicht

Reaktion Bediener	Reaktion A 6401	Bedeutung für SIMZ 1600	Bedeutung für C 8205-Z
RUN SIMZ		SIMZ 1600: Startmeldung	
N	#5	Neustart der Simulation Taste G	
(CR)	#S	Fortsetzung der Simulation	
EK X		Eingabe des Kanals	Artwahl Kanal 0...6
AK	X	Ausgabe des Kanals	Anzeige Kanal
GE K,G1,C		Gerätezuweisung G=TE Terminalausgabe TA Terminalausgabe LL LB-Eingabe LS LH-Ausgabe SD SD 1156 PE LB-Eingabeplatte PA LB-Ausgabeplatte AU Keine Ausgabe	Kanal 0...6
		I=0...6 K=Kanal 0...7 C=IS ISO-Coda R3 R-300-Code	
EW X		Eingabe Wahlschalter X=0 X=1 X=2	SP-Aus B8-Ein SP-Ein BS-Aus SP-Aus BS-Aus SP-Ein BS-Ein
AW	X	Ausgabe Wahlschalter	
E5 XXXX		Eingabe Sperradresse	Sperrschalter
AS	XXXX	Ausgabe Sperradresse	
BT XXXX		Programmstart	HGR IO XXXX' 000.

Reaktion Bediener	Reaktion A 6401	Bedeutung 4[r EIM2 1600	Bedeutung flir C 8205-2
GR WE EJT A1,A2, a1		Generallöschen Rechner Wiedereintritt Eingabe in den AC A1=20...211 A2=Z12...223 01 224 ... Z32	Taste GR K R Eingabe AC
B A 1, 01		Eingabe Befehlsregister A1 = AR 01 =GR	Eingabe Befehlsregister
EZ 01, A1		Eingabe der Zusatztrommel-Eperradresse 01 = Trommelnummer (4 A 1 = Zusatztrammel-Sperr-adresse	s
DFV 1,...,05		Druckformat Seriendrucker UP flir SD 1156• 01 = Linke Randbegrenzung 02 =Rechte Randbegrenzung 03 = 1-E inz eiliger V orsc hu b Ll 2-Einzeiliger Vorschub RE 3-Einzeiliger Vorschub L-I-R 4-Zweizeiliger Vorschub L 5-Zweizeiliger Vorschub R 6-Zweizeiliger VorichubL+R 04 = 0 ohne Leporello 1 linker Leporello 1 rechter Leporello 3 beide Leporellos 05 = Zeilen pro Blatt Rücksetzen der mit OF gegebenen Parameter N(=30 0((02.01.4) von OF Protokollierung der Abarbeitung 01= 0 keine Protokollierung 1 Protokollierung von B2, AR u. OR bei jedem Befehl] 2 Protokollierung von BZ, AC bei jedem Befehl 3 Protokollierung von B2,AR, OR,AC bei jedem Befehl 4 Protokollierung von BZ fei erfülltem Sprung 5 Pratoilierung von HZ, AR,OR bei erfëHtem Sprung 6 Protokollierung ran BZ,AC bei erfülltem Sprung 7 Protokollierung von BZ, AR, OR, AC bei erfülltem Sprung	
RD ET 01,...,ON PF 01			
PB AI, A2		Fesnegurg des Protokoll-bereiches A1 = Anfangsadresse A2 = Endadresse Abbruch an definierter Adresse A1	
SS AF	A5 #5	Meldung bei Erreichen der Abbruchadreise A 1 Rücksetzen von SS Neust.r	
Rs NS	** κE #5	Falsches Kommando Dateiangabe für LB-Plattendatei G = Nummer der Magnetplatte N = Name der Datei (4 Zeichen) U= UIC ixxx,rnr]	
ON G,N,U	** PF GGI #S ** UF#S ** AF #S	Paritätsfehler bet LB-Eingabe ÜberlaufAC Keine Trommelwahl vor Trommeltransport	

Reaktion Bediener	Reaktion A 6401	Bedeutung tiit 51M2 1600	Bedeutung für C 8205-2
		• BS #5 Es liegt ein bedingter Stopp vor	
		** US #S Es liegt ein unbedingter Stopp vur	

▪ Kompatibilitit siMz r600151mr 4201

4. Schluf3bemerkung

Der Simulator befindet sich zur Zeit in der Entwicklungsstufe E4. Geplanter Fertigstellungstermin ist das IV. Quartal 1983. Es ist möglich, daß im Laufe der Testarbeiten am KBR A 6401 noch Änderungen am Funktionsprinzip des Simulators voige-nommen werden. Deshalb ist zum III. Quartal 1983 an der Ingenieurhochschule Zwickau ciii Kolloquium über die Anwen-dung des Simulators geplant. Dabei wird es auch eine praktische Vorführung des Simulators geben.

Literatur

- III Programmierung und Bedienurig der elektronischen Rechenanlage Darr 8205 Z, Heft 1 bis 10. VEB Rechenelektronik Meiningen Zella-Mehlis
- j2/ Simulationsprogramm SIMC 4200. VEB Robotron-Vertrieb Dres-den.
- 13/ MOS SIMtF 4200/4000. VEB Rabotron 2FT



Leistungsfaktor Elektronische Rechentechnik

Erfahrungen bei der A nwendung

Von Dr. Karl Hauptmann und Prof. Dr. Karl-Heinz Steuer 224 Seiten, 26 Abbildungen, 7 Tabellen, Pappband, 14,00 M Bestellungen: 675 494 0/Hauptmann, Leistungsf. RT

Die Wirtschaftsstrategie der achtziger Jahre flir die gezielte Nutzung der elektronischen Rechentechnik und die bisherigen Erfahrungen der EDV-Anwendung in den Kombinatn werden in dieser Arbeit ausführlich behandelt. ökonomische Fragen im Zusammenhang mit der EDV-Anwendung werden ebenfalls Verständlich und für den vorgesehenen Zweck ausreichend dar-gestellt. Die Arbeit gibt den Leitern auf unterschiedlichen Ebenen und in verschiedenen Bereichen der Gesellschaft in geschlossener Form eine Darstellung des Standes und der Per-spektiven der Anwendung der elektronischen Rechentechnik in der DDR.

Ihre Bestellurig richten Sie bitte an den Verlag Die Wirtschaft, Abt. Vertrieb, 1055 Berlin, Am Friedrichshain 22,

Technischer Kundendienst für Basisrechnersysteme

Manfred Bilge, Dietrich Menz, Werner Hentschel
VEB Robotron-Vertrieb Berlin

Mit dem Einzug von Mikroelektronik und Mikrorechentechnik in das Einsatzgebiet bisheriger Kleinrechner wurden vorteilhafte Entwicklungen wirksam, die

- im modularen Aufbau von Baugruppen
- in der Miniaturisierung von Bauelementen und ganzen Funktionseinheiten
- in der Integration der verwendeten Schaltkreise und
- bei der Bauweise der Einschubtechnik zur Unterbringung verschiedener Geräte in universellen Schränken

sichtbar werden.

Diese Entwicklungen der modernen Rechentechnik erfordern auch teilweise neue Organisationsformen des Kundendienstes und veränderte Reparaturmethoden. Nur so kann bei der technischen Betreuung der Anlagen das höhere konstruktiv-technische Niveau der Geräte genutzt werden. Ein Beispiel dafür ist das hier beschriebene System des technischen Kundendienstes des VEB Kombinat Robotron für seine Basisrechnersysteme.

1. Kundendienststrategie

Die neue Kundendienststrategie des VEB Kombinat Robotron basiert auf einem Konzept, welches bei sparsamem Einsatz von personellen, finanziellen und materiellen Fonds den Nutzern eine optimale Verfügbarkeit der Basisrechnersysteme sichert.

Diese Konzeption umfaßt den gesamten Komplex der personellen und finanziellen Vorbereitung und Durchführung des technischen Kundendienstes bis hin zu einem effektiven Einsatz von Meß- und Prüfmitteln und der notwendigen Ersatzteile und Baugruppen.

Kern der Kundendienststrategie für die Basisrechner und die darauf aufbauenden arbeitsplatzbezogenen Konfigurationen des Kombinats Robotron ist die Leistungsabgrenzung zwischen Anwender, den Kundendienstorganen des Kombinats Robotron und ggf. ausländischen Serviceorganen.

Für die Organisation des technischen Kundendienstes im Inland und Ausland bestehen für einen optimalen Arbeitseinsatz Möglichkeiten der Arbeitsteilung entsprechend Tab. 1. Die im Einzelfall zu wählende Form der Leistungsabgrenzung wird gemeinsam mit dem Anwender festgelegt.

2. Methoden des Technischen Kundendienstes

Die im Abschnitt 1. skizzierte Kundendienststrategie geht unter anderem davon aus, daß Mikrorechner gegenüber bisherigen Rechnern auf Grund ihrer konstruktiven Gestaltung vielfältige Vorteile bei der technischen Betreuung bieten. Hierzu gehört zum Beispiel der Einsatz der Prüfsystemunterlagen, die eine Fehlerortung in weitaus größerem Umfang als bisher zulassen. Es wird ein umfangreiches Programmpaket bereitgestellt, welches in Verbindung mit Servicebaugruppen (Steckeinheiten, Prüf-Anschlußsteuereinheiten, Kurzschluß-Steckeinheiten und zugehörige Kabel) das Lokalisieren der Fehler im allgemeinen bis auf die defekte Steckeinheit sowie eine Funktionsprüfung der Rechereinheit ermöglichen.

Die lokalisierte Baugruppe wird dann ausgetauscht; die defekte Baugruppe wird in einer Werkstatt des Kombinats Robotron repariert.

Bei Exportanlagen kann von einem nationalen Kundendienstorgan die Baugruppenreparatur ausgeführt werden. Die dazu nötige Ausrüstung und Technologie wird vom Kombinat Robotron bereitgestellt. Fehler an mechanischen und elektrischen Baugruppen werden direkt an der Anlage beseitigt.

Des Weiteren kommen neben den bekannten Meßgeräten auch in begrenztem Umfang Logik- und Signaturanalysatoren zum Einsatz. Für den Prozessor wurde ein spezielles Prüfgerät entwickelt, welches im mobilen Kundendienst verwendbar ist. Eine wesentliche Neuerung stellt das im Aufbau befindliche Service-Zentrum Basisrechnersysteme des VEB Robotron-Vertrieb Berlin dar. Hier wird die Möglichkeit geschaffen, mit Hilfe der rechnergestützten Fehlersuche an einem zentralen Ort den an den Rechner tätigen Technikern Unterstützung bei der Fehlerbeseitigung zu gewähren. An festgelegten Meldepunkten wer-

Tab. 1 Verteilung der Aufgaben der technischen Betreuung der Basisrechnerysteme des VEB Kombinat Robotron

Aufgabe	Robo- Iron	aus- ländisches Service- organ	Anwender
Technikerschulung	x	1	1
Montage/Inbetriebnahme	x	1	—
Wartung (täglich)	—	—	x
Wartung (lt. Arbeitsplan)	—	1	x
Fehlerdiagnose mit PSU	x	x	K
Reparatur an K 1620/1630 und BDE	K	x	x in Sonderfällen
Reparatur an Anlage, PE; STE-, BG-, Gerätetausch	x	x	H
Werkstattreparaturen	x	x	—
Spezialistendienst/Ilavarietion tzu ng	x	1	—
Hardware-Änderungen	x	x	x gemäß Auftrag
Diagnose- und erungsdien st	x	x	—
Ersatzteil-/Arbeitsmittelversorgung	X	x	2
Baugruppentausch	x	x.	x
Unterstützung bei schwierigen hardware-Fehlern durch Service-Zentrum Berlin	x		

1 — Es können individuelle Regelungen entsprechend landespezifischen Gegebenheiten und Ausbildungsstand der Techniker getroffen werden

2 — bei Eigenbetreuung im Ausland (ohne nationales Kundendienstorgan)



erfältige Diagnoseprogramme erlauben die Fehleran-
im Dialog mit dem Rechner. ski-ortting
Foto: Rentschel

den die Signale mit den im Service-Zentrum abgespeicherten Werten verglichen (Bildung eines Referenzcodes), um so bei schwierigen Fehlern auf defekte Baugruppen schließen zu können.

Die Übermittlung der Daten wird in der Anlaufphase auf herkömmlichen Übertragungswegen (Telefon, Fernschreiber) realisiert. An dieser Dienstleistungseinrichtung können alle interessierten Anwender der Basisrechnersysteme teilhaben.

In Ergänzung der genannten Reparaturmethoden wurde in den Vertriebsbetrieben ein Baugruppenaustausch vorbereitet, der dem Nutzer der Basisrechnersysteme einen sofortigen Austausch seiner defekten gegen eine entweder industriell oder in Spezialwerkstätten instandgesetzte Baugruppe sichert.

3. Vorbereitung des Technischen Kundendienstes

Zur Realisierung der Kundendienststrategie für die Basisrechnersysteme des Kombinats Robotron wurde eine Reihe von Voraussetzungen geschaffen.

3.1. Technikerschulung

Vom Schulungszentrum des VEB Robotron-Vertrieb werden zu den einzelnen Geräten Grund- und Aufbaulehrgänge angeboten. Im Export kann die Schulung von ausländischen Kundendienstorganen übernommen werden.

Voraussetzung für den Besuch dieser Lehrgänge ist die einmalige erfolgreiche Absolvierung des Grundlehrganges Mikrorechner-Geräte-Systeme K 1600.

Die Auswahl der Lehrgänge erfolgt entsprechend der Arbeitsteilung bei Wartung und Reparatur zwischen den Anwendern und den Robotron-Vertriebsbetrieben bzw. aus ausländischen Kundendienstorganen.

3.2. Organisatorische Regelungen

Zu den wesentlichsten organisatorischen Festlegungen zur Durchsetzung der Kundendienststrategie gehören:

- sinnvolle Arbeitsteilung bei Reparatur und Wartung der Rechner zwischen den Anwendern und den Kundendienstorganen
- volle Nutzung der modernen Fehlersuchrhythmen wie Prüfsystemunterlagen (PSU), Service-Steckeinheiten sowie der Mci geräte durch Anwender und das Kundendienstorgan
- Beteiligung der Anwender an dem im VEB Robotron-Vertrieb Berlin installierten Servicezentrum bei der Ferndiagnose schwieriger Fehler
- Bereitstellung der Meltechnik der Anwender am Anlagenstandort
- Abgabe und Austausch von defekten Baugruppen durch den Anwender gegen regenerierte bei den Robotron-Vertriebsbetrieben (Inland) bzw. in Stützpunkten oder Technischen Zentren (Ausland)
- Aufbau von Reparaturwerkstätten in den Robotron-Vertriebsbetrieben für definierte Baugruppen und Geräte
- Aufbau von service-stützpunkten bzw. Technischen Zentren bei Kundendienstorganen bzw. Großanwendern im Ausland.

4. Schlussbemerkungen

Die von den Vertriebsbetrieben des Kombinats Robotron vorbereiteten Maßnahmen für den technischen Kundendienst ergeben für die Verfügbarkeit der Basisrechnersysteme ein volkswirtschaftliches Optimum. Die veränderten ökonomischen Bedingungen insbesondere auf dem Energiesektor, bei Investitionen und Kosten erfordern in größerem Umfang als bisher die Dezentralisierung der Leistungen. Das vorstehend umrissene Konzept für den technischen Kundendienst für Basisrechnersysteme des VEB Kombinat Robotron trägt diesem Anliegen Rechnung.



Grundlagen der elektronischen Datenverarbeitung für Ökonomen

Hochschul-lehrbuch

Herausgeberkollektiv, Ltg. Prof. Dr. habil. Siegfried Apelt
288 Seiten, 113 Abbildungen, 20 Tabellen, Pappband, 16,00 M
Bestellangaben: 675 493 2/Grundlagen DV

Das Lehrbuch ist für die Ausbildung von Studenten der Grundstudienrichtung Wirtschaftswissenschaften auf dem Gebiet der Informationsverarbeitung bestimmt. Die Aufgabe des Buches besteht darin, den Studenten Kenntnisse über wesentliche Begriffe und Zusammenhänge auf dem Gebiet der Informationsverarbeitung zu vermitteln, ihnen einen Überblick über die grundlegenden Voraussetzungen für die Anwendung der Informationsverarbeitung in der Ökonomie, über den Aufbau und die Wirkungsweise von elektronischen Datenverarbeitungsanlagen, die Datenorganisation sowie moderne Nutzungsformen der Datenverarbeitung zu geben und sie zur aktiven Mitwirkung bei der Erarbeitung und Nutzung von Anwendungsprojekten, die der Intensivierung des sozialistischen Reproduktionsprozesses dienen, zu erziehen.

Schulung für Basisrechner des Kombinats Robotron

Erhard Friske, Klaus Schurbeum
VEB Robotron-Vertrieb Berlin

1. Zielstellung der Schulung

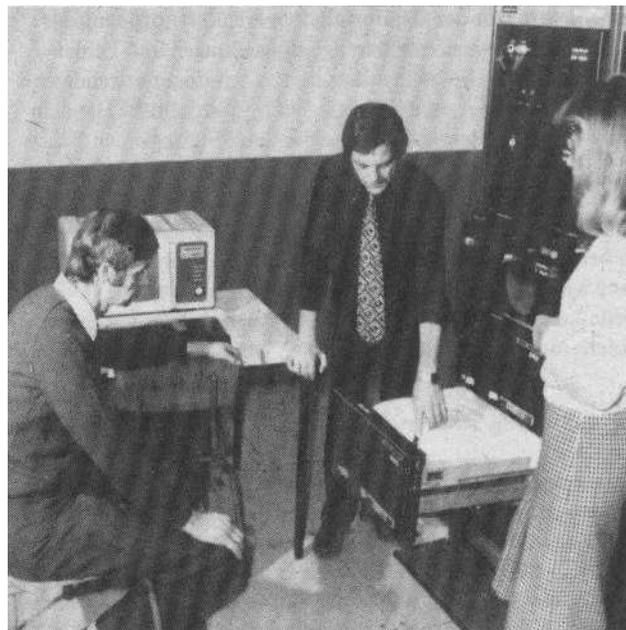
In den letzten 12 Jahren hat sich im VEB Robotron-Vertrieb Berlin ein Schulungszentrum für EDV-Kleinrechnersysteme herausgebildet. Mit der Entwicklung der Basisrechner A 640x und der darauf aufbauenden arbeitsplatzbezogenen Konfigurationen A 64xx ergaben sich auch für das Schulungszentrum neue und interessante Aufgaben. Für uns war es selbstverständlich, daß mit der Einführung der neuen Rechnersysteme auch höhere Anforderungen an das Ausbildungsniveau gestellt werden müssen. Ziel für die Bereiche Anwendungs- und Instandhaltungstechnik war es gleichermaßen, mit einer noch höheren Effektivität die Schulung vorzubereiten und durchzuführen. Grundgedanke unserer Überlegungen ist es, in einer hohen Qualität bei kürzeren Lehrgangszeiten die gestellten Bildungsziele zu erreichen.

2. Schulung Instandhaltung

Von großer gesamtgesellschaftlicher Tragweite ist die Forderung nach Erhöhung der Effektivität der Volkswirtschaft. Die Verringerung von Ausfallzeiten an Rechnersystemen ist daher von besonderer Bedeutung. Um diese Ziele zu verwirklichen, müssen neue Wege gegangen werden, um besonders die Fehlersuche und Fehlerbeseitigung rationeller zu gestalten. Ausgangspunkte unserer Vorstellungen zu einem Schulungskonzept für die Basisrechnersysteme waren Entwicklungshinweise sowie die Kundendienststrategie des VEB Kombinat Robotron. Die Hauptorientierung zur Fehlerbeseitigung in Havariefällen ist durch den Baugruppentausch von Kundendiensttechnikern gegeben. Dazu ist es erforderlich, daß vom Anwender gezielte Angaben gemacht werden können, die auf die mögliche Fehlerursache hinweisen können. Da nicht in jedem Fall ein Fehler mittels Baugruppentausch behoben werden kann, ist es erforderlich, einen bestimmten Kreis von Technikern des Anwenders mit dem notwendigen Wissen auszubilden. Damit ergeben sich von vornherein zwei Ausbildungsrichtungen mit den Bildungszielen

- Befähigung, Fehler mittels Baugruppentausch zu beseitigen bzw. gezielte Hinweise und Angaben der Fehlererscheinung für den Einsatz von Spezialisten zu geben
- Befähigung, Fehler mit Hilfe von mitteltechnischen Mitteln und Kenntnissen der logischen Struktur zu beseitigen.

Im Lehrgangssystem Basisrechner, Teil Instandhaltung (Abb. 1), sind die beiden Ausbildungsrichtungen deutlich erkennbar. Grundsätzlich muß jeder Techniker zunächst einmal den Grund-



Praxisbezogene Techniker Ausbildung im Schulungszentrum des VEB Robotron-Vertrieb Berlin. Foto: Hentachei

schulung besuchen. Dort werden die notwendigen Kenntnisse über die prinzipielle Arbeitsweise, die Programmierung, die Wirkungsweise der einzelnen Funktionseinheiten, die Bedienung und die Diagnosemöglichkeiten vermittelt. Mit dem erfolgreichen Abschluß dieses Lehrganges sind die Teilnehmer befähigt, weiterführende Lehrgänge zu besuchen. Für die erste Ausbildungsrichtung ist dafür der Lehrgang Systemdiagnose vorgesehen. Schwerpunkt ist hier die Diagnose des gesamten Systems mit Wartungshilfen des Moos 1600 sowie die Erstellung eigener Gerätebedienprogramme mit und ohne Betriebssystemleistungen. Wir sind dabei von der Tatsache ausgegangen, daß als echtes Kriterium für die ordnungsgemäße Arbeitsweise des Basisrechners nur eine fehlerfreie Programmabarbeitung unter Steuerung des Betriebssystems gelten kann.

In der anderen Ausbildungsrichtung werden in erster Linie Kundendiensttechniker sowie Techniker größerer Anwender an den Geräten ausgebildet, die zum Systembestand der Basisrechner gehören. In diesen Lehrgängen werden ihnen die logisch-funk-

Anwender- Techniker	Kundendienst- Techniker
• Grundlehrgang MG5 K 1600	
<ul style="list-style-type: none"> • 5y5[етдгдпове • Weitere Lehrgänge siehe Technikerkundendienst denoII territorialer Setruun9sIorm 'Я zustmmdigen Kurzend+izsf- ngodane5 	<ul style="list-style-type: none"> • Ka55e[[ennxl9ne[band • Frolierzspncher • LochöandezheiI • M4rlrplexer, Konz rOtor • Tezmezal • Kassettenptattenspercher • Magzretöandspekher • Paralleldrucker • Serverdrucker • Prozessor K 7620/K 1630 - nur für Spezialisten I • 5y5sterngererrerrun9 (nur für Spezzel/stenl

Abb. 1 Lehrgangssystem für Basisrechner A 640x

tioneile Struktur der Geräteeinheiten erläutert sowie mittels praktischer Übungen notwendige Einstellungen und Fehler-suche trainiert. Die Teilnehmer sind so mit dem notwendigen Wissen ausgerüstet, in der Praxis durch folgerichtiges Handeln sowie mit Hilfe der erlernten Fehler-Such-Strategie den Fehler einzugrenzen und zu beseitigen. Oberstes Ziel unserer Techniker- ausbildung ist es, solche logischen Denkstrukturen herauszu- bilden, die dazu befähigen, in möglichst kurzer Zeit eine Havarie zu beseitigen. Für Spezialisten der Robotron-Kundendienst- organe werden im Rahmen der **Instandhaltungsschulung** nach spe- zielle Lehrgänge für den K 1620/K 1 630 sowie für die System- generierung durchgeführt.

3. Schulung Anwendungstechnik

Ein wesentlicher Bestandteil der Einsatzvorbereitung für Basis- rechner und darauf aufbauende arbeitsplatzbezogene Konfigu- rationen ist die Schulung von Organisatoren, Programmierern und Bedienern. Mit dem erhöhten Leistungsangebot des Opera- tionssystems werden zwangsläufig größere Anforderungen an Lehrkräfte und Lehrgangsteilnehmer gestellt. Um diesem Ziel gerecht zu werden, wurde ein Lehrgangssystem entwickelt, das mit seiner Modularität alien Zielgruppen gerecht wird. Im Gegensatz zur Schulung Instandhaltung ist die Schulung Anwendungstechnik in hohem Maße von der jeweiligen kon- kreten Art der Basisrechner bzw. arbeitsplatzbezogenen Kon- figurationen abhängig. Im folgenden wird deshalb beispielhaft das Lehrgangssystem Anwendungstechnik für die kommerziellen Basisrechnersysteme KBR A 6401 /A 6402 erläutert (Abb. 2). Hier ist als vorbereitender Lehrgang ein Übersichtslehrgang vor- gesehen. Es wird ein Überblick über das Mikrarechnergerätesy- stem, das Betriebssystem und die unterschiedlichen Program- miersprachen gegeben. Damit soll deni Nutzer ein Leitfaden für die optimale Strategie zur Realisierung seiner Vorstellungen in die Hand gegeben werden. Darauf aufbauend werden speziell für Programmierer die Lehrgänge Assemblerprogrammierung und Betriebssystem für WTÜR-Einsatz durchgeführt. Verbun- den mit praktischen Übungen am KBR A 6401/4 6402 werden



Ausbildung in der Lehrgangsrichtung Anwendungstechnik im Schulungs- zentrum des VEB Robotron-Vertrieb Berlin. Foto: Hentschel

die Teilnehmer befähigt, in der Sprache MACRO 1600 Pro- gramme zu schreiben sowie das Leistungsangebot des Betriebs- systems MOOS 1600 voll zu nutzen. Zu beachten ist, daß diese beiden Lehrgänge eine fachliche Ei- genschaft bilden. Wahlweise können anschließend Lehrgänge zum POS-Angebot sowie Systemprogrammierung belegt werden. Für die problemorientierten Sprachen COBOL, FORTRAN und PASCAL werden, aufbauend auf den Übersichtslehrgang, ebenfalls Lehrgänge angeboten. Speziell für Bediener des KBR A 6401/A 6402 wird eine 14tägige Schulung abgehalten. Neben der eigentlichen Bedienung des Rechnersystems und des Be- triebssystems werden den Teilnehmern die Wertungshilfen des MOOS 1600 vorgestellt, damit auch sie in der Lage sind, bei Havarien gezielte Angaben an das ruständige Kundendienst- organ zu übermitteln. Mit dem Aufbau eines Servicezentrums beim VEB Robotron-Vertrieb Berlin werden diese Kenntnisse eine unabdingbare Notwendigkeit sein.

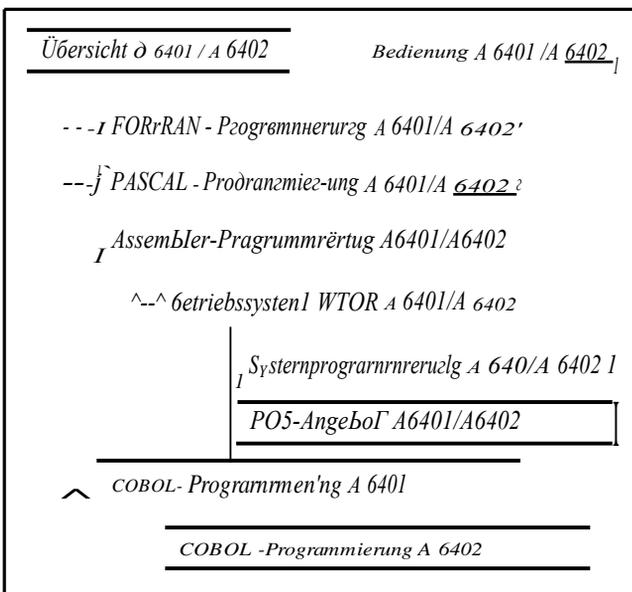


Abb. 2 Lehrgangssystem KBR A 6401/A 6402, Teil Anwendungstechnik

Literaturverzeichnis von Seite 51

- 11/ Tomand', E. P.: Des Echtzeitsteuerprogrammingsystem EIEEX 1521. rechen technikdatenverarbeitung, Beiheft 4/1981
- 12/ Stanke, K.: Meßwertfassung und -verarbeitung in der Prozeß- steuerung. rechen technik/datenverarbeitung, Beiheft 4/1981
- 13/ Steingroewer, G., Lindner, A.: Meßwertfassung und Primärauf- bereitung mit dem robotron K 1520 unter Nutzung des EIEEX 1521. rechen technik/datenverarbeitung, Beiheft 4/1981
- 14/ Müller: Nutzerdokumentation des Byte-Gleitkomma-Arithmetik- registerorientiert. TU Dresden, 1981
- 15/ Enkelmann, W.: Eine Softwarelösung zur Kopplung der ursatd 5000 mit einem Rechner K 1620/K 1630 im Prozentrechnersystem A 6491/A 6492. msr 25 (1982) 9

aspekte abkürzungen

AA	Analogausgabe
ASU	Anschlußsteuerung für Bedieneinheit
ASS	Anschlußsteuerung für Bildschirm
ADA	Anschlußsteuerung für SIF 1000
AE	Analogeingabe; auch Ausgabeeinheit
AFS	Anschlußsteuerung für Folienspeicher
AIS	Anschlußsteuereinheit für serielle Datenadapter
AKP	Anschlußsteuerung für Kassettenplattenspeicher
AKT	Arbeitsplatz für Konstruktion und Technologie
AMB	Anschlußsteuerung für Magnetbandgerät
ARP	Arithmetikprozessor
AS	Anschlußsteuerung
Ass	Anschlußsteuerung für Geräte mit IFSS-Interface (Nahübertragung)
AST	Asynchroner System-Trap
ASV	Anschlußsteuerung für Geräte mit V.24-Interface (Fernübertragung)
ATP	Anschlußsteuerung für serielles Linieninterface zum Anschluß der Prozelein-/ausgabe
BAD	Busadapter
BDE	Bedieneinheit
BER	Buserweiterung
BG	Baugruppe
BHE	Bus-Koppeleinheit
BTL	Bildschirm-Terminal
BUA	Bus-Abschlug
BUM	Busumsetzer
EVE	Busverstärker
BV5	Bildverarbeitungssystem
DA-0	Digitalausgabe-Optokoppler
DA-R	Digitalausgabe-Relais
DAS	Digitalausgabe statisch
DA-T	Digitalausgabe-Transistor
DBM	Datenbankmaschine
DES	Datenbanksystem
DC	Gerätekoordinatensystem (device coordinate)
DEAS	Digital-Ein-/j-Ausgabe
DED	Digitaleingabe dynamisch
DEM	Digitaleingabe multiplex
DEP	Testprogramm (Debugging Program)
DES	Digitaleingabe statisch
DIs	Datenerfassungs- und Informationssystem
EDI	Editor
EGS	Einheitliches Gefäßsystem
EMP	Emulatorprozessor
FCS	Dateizugriffsroutine (File control service)
FP	Frontplatte; auch Fileprozessor
FS	Folienspeicher
FSPE	Folienspeichereinheit
GDN	Gleichstrom-Datenübertragung mit Niederpegel
GE	Grundeinheit
IA	Impulsausgabe
IFLS	Schnelles serielles Interface
IF5P	Interface für sternförmigen Anschluß mit Parallelübertragung
IFSS	Interface für sternförmigen Anschluß mit Seriellübertragung
IPU	Funktionelles Betriebssystem zur Bildverarbeitung (Image Processing Utility)
KBR	Kommerzielles Basisrechnersystem
KD	Kundendienst
KEP	Kleines Eingabeprogramm
KEs	Karteneinschub

KMB	Kassetteneinheit
ICMBE	Kassettenmagriet bandeinheit
LEE	Lochbandeinheit
LB	Lochbandleser
LB R	Bibliotheksprogramm (Librarian Utility)
LBS	Lochbandstanter
MEP	Minimales Eingabeprogramm
µC5	Mikrorechner-Gerätesystem
MON	Monitor
MO5	Maschinenorientierte Systemunterlagen
MR	Mikrorechner
M RKB	Mehrrechner-Koppelbus
NDC	Normalisiertes Gerätekoordinatensystem (Normalized device coordinate)
ODT	Testprogramm (On-line Debugging Tool)
DPS	Operativspeicher
PE	Peripherie
PEAE	Prozellein-/ausgabeeinheit
PFS	Programmierbarer Festwertspeicher
PIP	Datei-Transferprogramm (Peripheral Interchange Program)
P05	Problemorientierte Systemunterlagen
pRS	Prozessorrechnersystem
Psu	Prüfsystemunterlagen
PSW	Programmstatuswort
PT	Lochband-Eingabeprogramm (Paper Tape Input)
QPt5	Querschnitts-PD5
RAM	Löschbarer, programmierbarer Schreib-/Lese-Speicher mit wahlfreiem Zugriff (Random Access Memory)
RLP	Rückverdrahtungsleiterplatte
ROM	Festwertspeicher mit wahlfreiem Zugriff (Read Only Memory)
SD	Seriendrucker
SDA	Serieller Datenadapter
SE	Steuereinheit; auch Serviceeinheit
SIF	Standard-Interface
SKR	System Kleinrechner
SST	Synchroner System-Trap
ST	Standard
STE	Steckeinheit
STV-M	Stromversorgungsmodule
STZ	Stromversorgungszusatz
SVE	Speicherverwaltungseinheit; auch Speichermitteilungseinheit
TAK	Telegrafie-Anschlußkassetten
TBR	Terminalorientiertes Basisrechnersystem
TM	Technische Mittel
tIE	Übertragungseinheit
UE'''	(7berwachungsmodul
UFD	Nutzerdatenverzeichnis
UIC	Nutzeridentifikationscode
UIZ	Univiersalimpu lsrah l er
VMR	Virtuelles Kommandoprogramm
VP	Verarbeitungsprozessor
WC	Weltkoordinatensystem (World coordinate)
WT	Wechselstromelektronikübertragungseinrichtung
ITFJR	Wissenschaftlich-technische und ökonomische Berechnungen
Z RE	Zentrale Recheneinheit
ZVE	Zentrale Verarbeitungseinheit
ZZE	Zähler-Zeitgebereinheit

ekte Jahresinhaltsverzeichnis 1982

1/82

Bürocomputer

Dr. Werner Schulze:
Dezentrale Datenverarbeitung und Bürocomputer, S. 2

Gerätetechnik

Reinhard Donath:
K 1520 als rechen-technische Basis für Bürocomputer, S. 7
Gerhard Anfig:
Bürocomputer **rabatron A 5110**, S. 12
Ulrich Waloszczyk,
Albrecht Hinkel:
Bürocomputer A 5120 tied
A 5130 - Gerätetechnik mit Struktur, S. 14

Systemverlagerung

Viktor Bernt, Peter Kraft:
Software für Bürocomputer des VEB Kombinat Robotron, S. 23
Autorenkollektiv: Einheitliches Betriebssystem für **A 5120** und **A 5130**, S. 26
Ulrich Arnold, Dieter Schulze:
Betriebssystem GROS für Bürocomputer, S. 34
Dr. Wolfgang Nyderie,
Christo Mintschew:
COBOL 1520 für **Bürocomputer**, S. 40
Prof. Dr. Helmut Adler,
Dr. Peter Hoffmann,
Dr. Hans-Ulrich Karl:
PASCAL 1520 für Bürocomputer, S. 42

Service und Schulung

Jürgen Losczynski:
Technischer Kundendienst für **Bürocomputer**, S. 46
Siegfried Hämmerling, Ingo Rau,
Dietrich Sternberg:
Schulung für Bürocomputer, S. 45

Anwendungen

Helmut Benedix:
Verwaltung von Ersatzteilbeständen mittels **A 5110**, S. 52
Dr. Wolfgang Abt:
Abrechnung und Analyse von Warenbewegungen mittels Bürocomputer, S. 55
Eberhard Romfeld:
Anwendungslösungen für die Bürocomputer A 5120 und A 5130, S. 59

2/82

Rechenzentrumsbetrieb

Leistungsbewertung

Dr. Gerhard Bergholz:
Leistungsanalyse und Leistungserhöhung von Rechenanlagen, S. 2
Jürgen Maikath, Dr. Gerd Wähler:
Leistungsanalyse und Leistungsvergleich an EDVA, S. 7
Gottfried Pfuher:
Kontinuierliche Leistungsüberwachung virtueller Maschinen, S. 12
Bernhard Stiefel:
Leistungsmessung an Datenbanksystemen, S. 19
Dr. Eberhard Lipski:
Untersuchung des Systemverhaltens mittels Hardware-Monitor, S. 23

Technologie und Verfahren

Sabine Klaus, Volker Hillert:
Der Produktionsprozeß im Rechenzentrum bei Dialogbetrieb, S. 27
Fritz Mudra:
Testbetrieb mit CRJE, S. 30
Joachim Hoke, Hans-Dieter Linke:
Das Konzept virtueller Maschinen im praktischen Rechenbetrieb, S. 34
Bernd Brehme:
Arbeitsorganisation in Rechenzentren bei Anwendung von Datenbanksystemen, S. 38
Reinhard Karge,
Dr. Helmut Praschek:
Arbeit mit einem Quellbibliothekssystem, S. 42

Automatisierung von Hilfs- und Nebenprozessen

Joachim Müller:
Rechnergestützte Produktionsplanung und EDVA-Belegung mit dem Programm system MKP, S. 48
Günter Zäpernick, Manfred Franke, Jürgen Iliasia:
Dialogorientierte operative Produktionsdisposition von maschinentechnischen Leistungen, S. 54
Dr. Helmut Praschek,
Reinhard Karge:
Maschinelles Datenträgerverwaltungsprojekt, S. 57

3/82

Softwareentwicklung

Erwurfprinzipien

Dr. Jürgen Vogel:
Datenverarbeitungsprojektierung - Stand und Entwicklung, S. 2
Christian Horn:
Rechnergestützte Methoden in der Softwareproduktion, S. 9
Dr. Günter Rothhardt:
Erfahrungen mit der Strukturierten Programmierung, S. 12
Dr. Peter Zschockel:
Datenstrukturumabhängige Programmierung, S. 17
Dr. Reinhard Lief:
Lösungsansatz der datenorientierten Entwicklungstechnologie, S. 20
Dr. Kuno Schmidt:
Architekturtypen von Programmsystemen, S. 23
Dr. Günter Rothhardt:
Programmentwicklung am Entwurfsarbeitsplatz ARM-M, S. 27

Methoden

Dr. Wolfgang Zschau:
CDL 2 als Mittel zur Strukturierung und Implementierung, S. 29
Dietmar Horn:
Ein Steuersystem für Programmsysteme, S. 32
Dr. Peter Pfeiffer:
Das Makrosystem SKODA, S. 34
Ulrich Böhm, Bernhard Lehnig:
Dynamische Moduinitialisierung, S. 36
Dr. Manfred Gerisch:
Verwalten von Bibliotheken und Komponenten mit UNI-3, S. 39
Dr. Mieczeslaw Bazewicz:
Einfluß der Nutzeranforderungen auf die Architektur verteilter Computersysteme, S. 41

Sprachanwendungen

Siegfried Hübet:
Generator für normierte PL/I-Programme im OSTES, S. 46
Dr. Reiner Dumke:
Aufbau einer Programmbank mit MALIS, S. 49
Martin Hartwig, Eckhard Stein:
Ada -- Programmiersprache mit neuen Konzepten, S. 51
Dietrich Holz, Roland Riedel:
Erstellen von Programmamaufplänen mittels PAF, S. 55

Ökonomie und Qualität

Klaus Riedel:
Berücksichtigung von Anwenderforderungen beim Programmwurf, S. 58
Manfred Florian:
Qualitätssicherung in der DV-Projektierung, S. 61
Rainer Trautloft:
Selektion von Implementierungsstrukturen, S. 63

4/82

Entwicklungstendenzen

Prof. Dr. Franz Stuchlik:
Informationsverarbeitung unterstützt neue Technologien geistiger Arbeit; S. 2

Informations- und Datenbanken

Prof. Dr. Juliusz L. Kutikowski:
Wissensdarstellung in Rechnersystemen, S. 8
Dr. C5c Miroslav Ferdrych:
Entwurf eines Datenmodells mittels Typentheorie, S. 14

Modellierung und Verarbeitung

Prof. Dr. Gunter Schwarze:
Simulations-, Datenbank-, Modellbanksysteme, S. 18
Prof. Dr. Dr. Igor N. Moltschanow:
Probleme der Parallelverarbeitung, S. 21

Software

Prof. Dr. Hans Loeper,
Prof. Dr. Gerd Stiller,
Dr. Hans-Jörg Jackett:
Softwaretechnologie tied Programmiersprachen, S. 26
Prof. Dr. Dr. Igor N. Moitschanow,
Prof. Dr. Franz Stuchlik:
Externe Software, S. 31

Krapbinenikation

Prof. Dr. Helmut Löffler:
Rechnemunterstützte Telekommunikation, S. 36
Igor N. Moitschanow:
Datenfernverarbeitung - die Kostenpolitik, S. 43

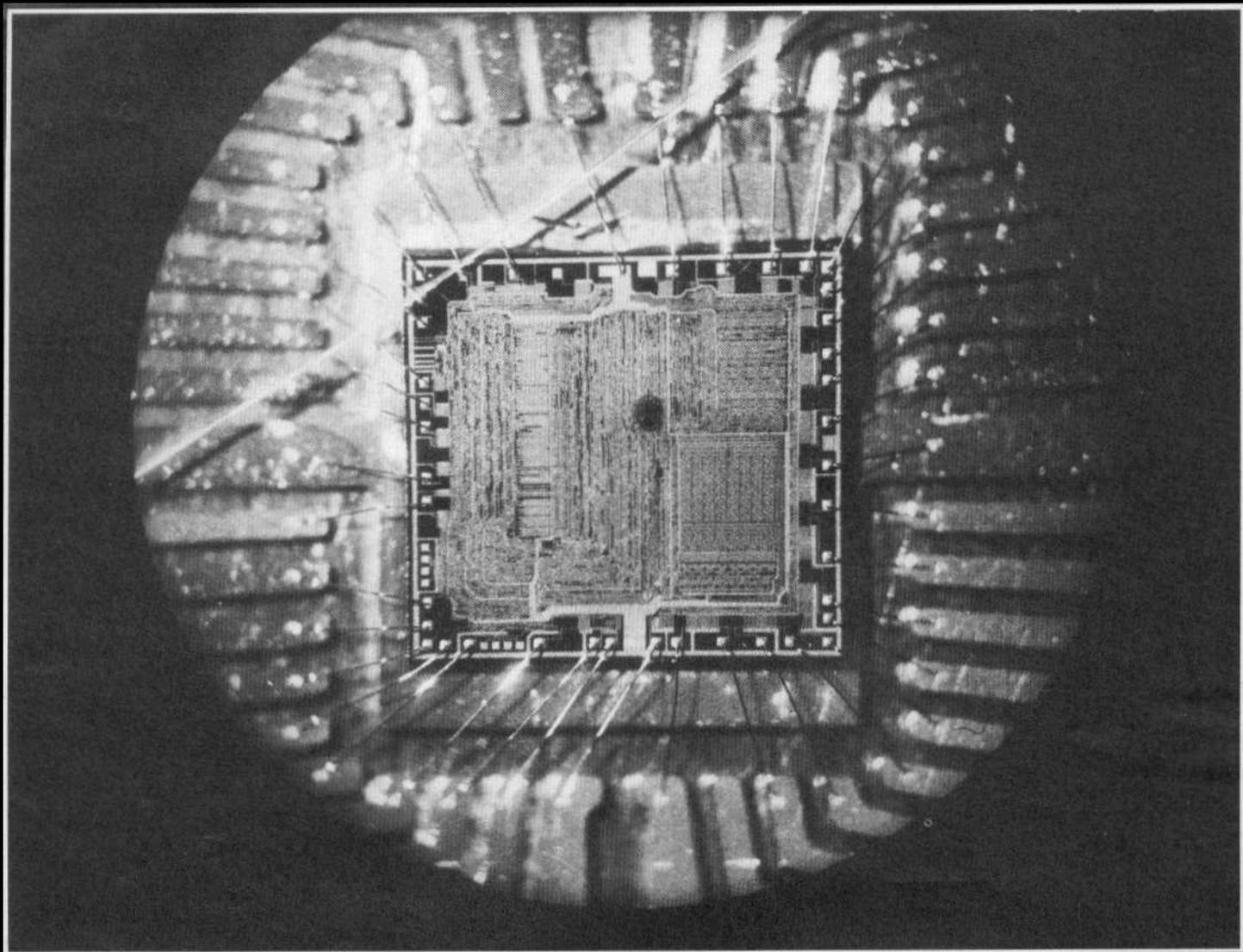
Architektur und Standards

Prof. Dr. Karl-Heinz Werler:
Architektur rechnerunterstützter Systeme der TEVO, S. 47
Dr. Mieczeslaw Bazewicz,
Prof. Dr. Franz Stuchlik:
Nutzerschnittstelle verteilter Informationssysteme, S. 51
Dr. Adolf Kotzauer:
Grafische Standardisierung und ihre Folgen, S. 54

Anwendung und Anwender

Dr. Gerd Koselowski,
Prof. Dr. Gerhard-Eberhard Kühne,
Karl Mawrin,
Dr. Hans-Dietrich Hempel:
Patientenbezogene Informationsverarbeitung, S. 57
Prof. Dr. Wolfgang Schoppen:
Weiterbildung von Leitern der automatisierten Informationsverarbeitung, S. 60
Dr. Erhard Berndt:
Implementierung einer Testversion von GK5-Routinen, S. 63

Herausforderung Mikroelektronik



Von einem Autorenkollektiv unter Leitung
von Günther Fraas
etwa 288 Seiten, Glanzfolie,
etwa 19,80 M
Bestellangaben: 675 595 1 /Herausf.
Mikroel.
Erscheint voraussichtlich
im I V. Quartal 1983

Die Mikroelektronik und ihre verschiedenartigen Anwendungen stellen Anforderungen an unsere und vielleicht auch künftige Generationen, wie sie selbst von Fachleuten in ihrer ganzen Tragweite noch nicht überschaut werden können. Anliegen dieser Publikation ist es, auch schwierige technische und naturwissenschaftliche Zusammenhänge einfach und verständlich darzulegen und eine neue Denkweise anzuregen, in der die Mikroelektronik als das gesehen werden kann, was sie ist: eine Herausforderung an uns alle.

Verlag Die Wirtschaft, 1055 Berlin, Am Friedrichshain 22





Basisrechner sind sowohl in autonomer Betriebsweise als auch in Direktkopplung als Terminal in Großrechner-systemen vielfältig einsetzbar.

Grundlage für das Erreichen der dazu notwendigen Flexibilität ist die modulare Konzeption der Mikrorechnerfamilie K 1600. So werden die logischen Funktionseinheiten als Steckeinheiten ausgeführt und in Blockeinschüben konstruktiv zusammengefaßt. Mit der Austauschbarkeit dieser Einschübe ergibt sich ein breites Spektrum

peripherer Geräte, Multiplexoren und universeller problemorientierter Terminals. Als vorteilhaft erweist sich das Finbeziehen der Basisrechner in das System der Kleinrechner der sozialistischen Länder, da über den Einheitsbus auch periphere Geräte aus dem Lieferumfang des SKR genutzt werden können. Die Beiträge in diesem Heft und die Konfigurationsbeispiele auf dieser Seite verdeutlichen dieses sehr anschaulich.