

GENERAL AUTOMATION

1042

informiert . . .



PROGRAMM DEVELOPMENT SYSTEM

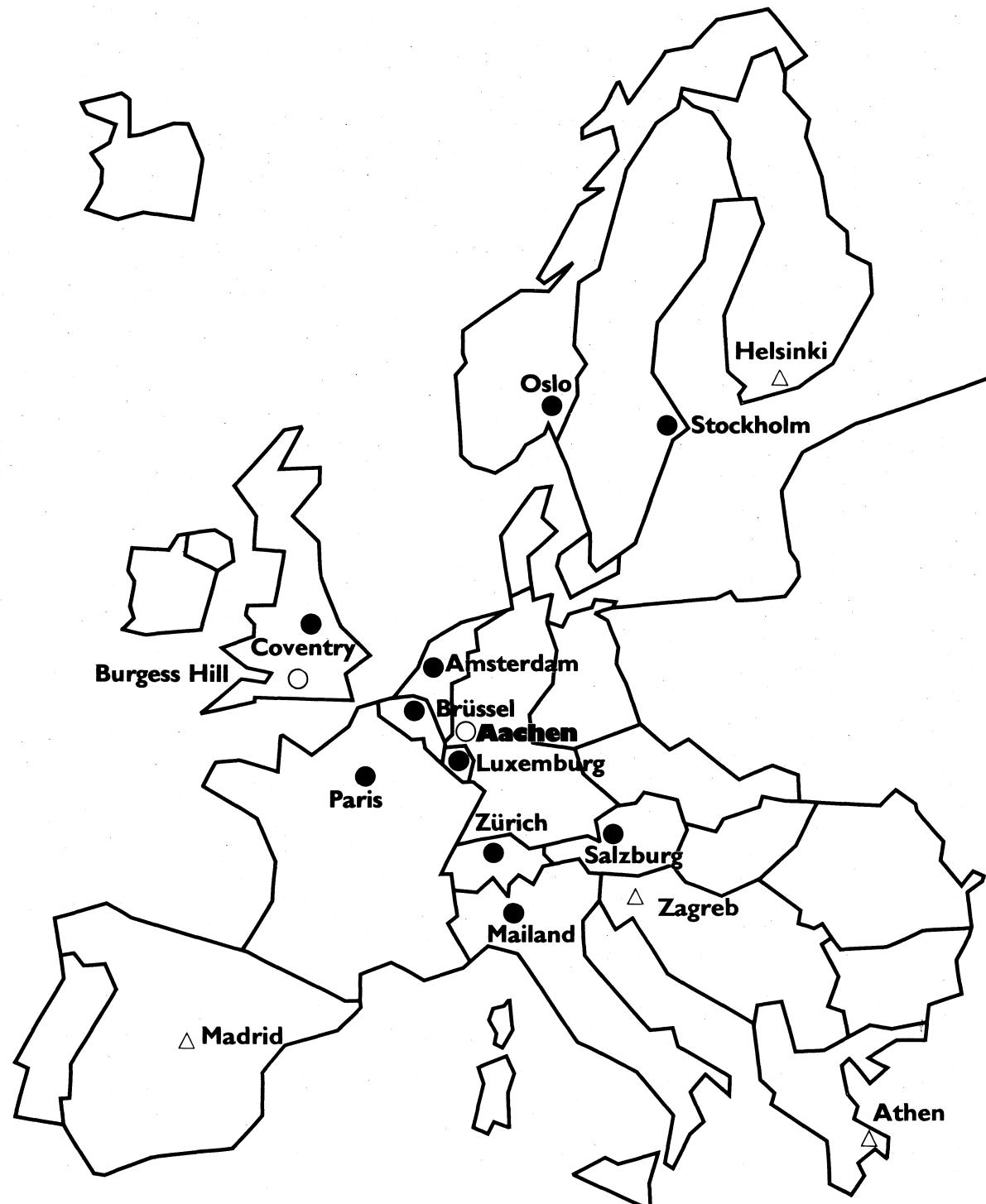
PDS 216/10

von

GENERAL AUTOMATION



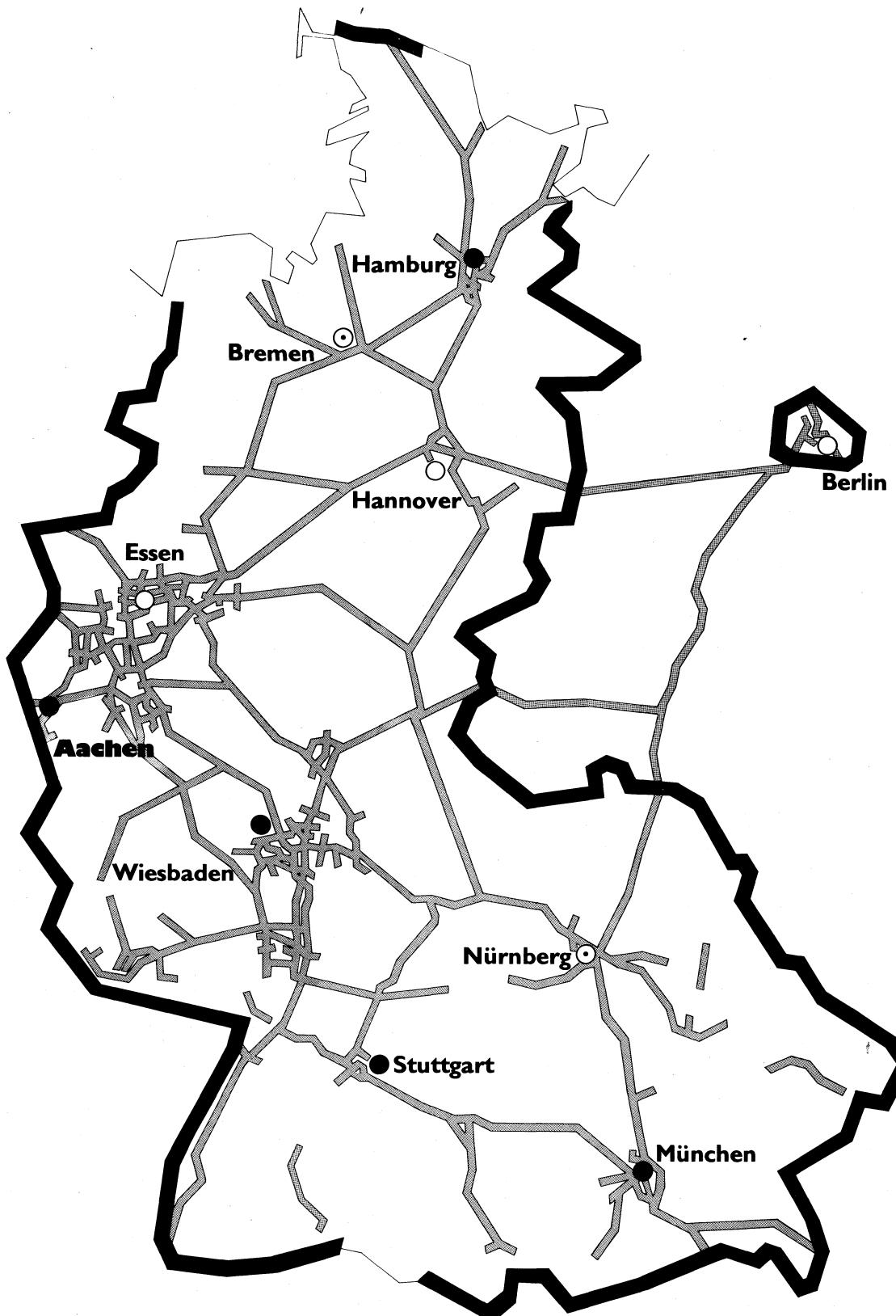
GENERAL AUTOMATION in Europa



- National Zentrale
- Produktionsstätten
- △ Vertriebspartner



Vertriebs-/ Service-Netz BRD



- Vertriebsbüro mit Service
- Servicebüro
- ◎ geplantes Servicebüro



Firmen-Porträt

Gründung der ERA-Elektronische Rechenanlagen, Aachen 1962

Gründung der GENERAL AUTOMATION in Anaheim 1967

Übernahme von ERA in den GENERAL AUTOMATION-Verband
als 100%ige Tochter 1970

PERSONALSTAND 1978

GA weltweit 1.800

GA Europa 350

G A S 153

(62 Hard- und Software-Ingenieure)



Umsatzerfolge

JAHRESUMSATZ 1970

(01.08.1977 bis 31.07.1978)

	DM
GA weltweit	220 Mio.
GA Europa	42 Mio.
G A S	20 Mio.

ANLAGENVOLUMEN SEIT GRÜNDUNG

	DM	Einheiten
GA weltweit	1 MRD.	15.000
GA Europa	220 Mio.	3.000
G A S	110 Mio.	800



Unsere Leistungen

- Anwenderberatung
- Lieferung von Standard Hardware
- Lieferung von Standard Software
- Lieferung von Spezial Hardware
- Lieferung von Anwender Software
- Schlüsselfertige Installationen
- Zusammenarbeit mit führenden Software-Häusern
- Hardware Service
- Software Service
- Projektverantwortung



Unsere Einsatzbereiche

INDUSTRIEAUTOMATISIERUNG

- Werkzeugmaschinen Steuerung
- Prozeßdatenerfassung
- Prozeßsteuerung
- Betriebsdatenerfassung
- Hochgeschwindigkeits-Datenerfassung
- Hierarchische Netzwerke

KOMMUNIKATIONSSYSTEME

- Banking (S. W. I. F. T)
- Rechnerverbundsysteme
- Telexvermittlungssysteme
- Großrechner Anschlüsse

KOMMERZIELLE SYSTEME

- Banking (S. A. F. E.)
- Finanzbuchhaltung
- Lohn und Gehalt
- Lager-/Bestandskontrolle



Unsere Einsatzbereiche

VIDEO-DATENERFASSUNG UND AUSWERTUNG

- Qualitätskontrolle
- Vollständigkeitskontrolle
- Scheinwerfereinstellung

ANWENDUNG IM OEM-BEREICH

(Original Equipment Manufacturer)

- MESSER GRIESHEIM (Brennschneidemaschinensteuerung)
- SCHENCK (Hydropulser, Rollenprüfstände)
- ÖRLIKON (Werkzeugmaschinensteuerung, Bearbeitungszentren)
- POSALUX (Werkzeugmaschinensteuerung)
- VÖEST (Programmier Plätze)
- SEL (Communication, Verkehrssteuerung, Zugüberwachung)
- BABCOCK (Hochregalsteuerung)
- TÜV (automatische Fahrzeugprüfung)
- DATRON (kommerzielle Anwendungen)



TISCH-COMPUTERSYSTEME als wirtschaftliches
Instrument für die DV-GRUNDAUSBILDUNG

von

Wolfgang K. Giloi

Technische Universität Berlin, Fachbereich Informatik
und University of Minnesota



ZUSAMMENFASSUNG

Nachdem der Bedarf an Rechenkapazität an den deutschen Universitäten in den vergangenen Jahren abgedeckt zu sein schien, melden in neuerer Zeit etliche Universitätsrechenzentren den Anspruch auf größere und leistungsfähigere Nachfolgesysteme für die vorhandenen Computer an. Der Grund dafür liegt darin, daß in den vergangenen Jahren immer mehr wissenschaftliche Disziplinen eine Grundausbildung in Datenverarbeitung in ihre Studienplätze aufgenommen haben und daß diese Grundausbildung meistens im Timesharing-Betrieb von dem im Universitätsrechenzentrum vorhandenen Großrechner mitgetragen werden muß. Eine solche Lösung führt zu erheblichen Investitionen bzw. bindet Rechnerkapazität, die im Stapelbetrieb wirkungsvoller genutzt werden könnte. Sie war daher nur solange zu vertreten, wie keine bessere Alternative existierte. Mit dem Aufkommen der Mikroprozessoren ist es inzwischen aber möglich geworden, Tisch-Computer anzubieten, die bezüglich des Software-Komforts und der Leistung alle für die DV-Grundausbildung zu stellenden Anforderungen erfüllen (und dabei manche größeren Systeme übertreffen), zu einem Preis, der nur unwesentlich über dem einer Timesharing-Datenstation liegt (bei normaler Auslastung liegen die Kosten für Amortisation und Verzinsung bei 1 - 2 DM pro Stunde!). Insbesondere ist hier richtungsweisend das an der University of California, San Diego, entwickelte UCSD PASCAL-System, das in dem Aufsatz näher besprochen wird. Tisch-Computer, die auf diesem System basieren, haben sich in den U. S. A. im vergangenen Jahr sehr stark verbreitet, und mit der Einführung solcher Systeme in Europa ist in nächster Zukunft zu rechnen.



1. EINLEITUNG

In den naturwissenschaftlichen und ingenieurwissenschaftlichen Disziplinen, den wirtschaftswissenschaftlichen und sozialwissenschaftlichen Fächern und natürlich auch in der Informatik ist heute eine gewisse Grundausbildung in Datenverarbeitung zum festen Bestandteil der jeweiligen Studiengänge geworden. Eine solche "DV-Grundausbildung" umfaßt in der Regel eine Einführung in eine höhere Programmiersprache, in die anwendungsspezifischen Methoden und numerischen Verfahren zur Problemlösung mit dem Computer, sowie in die Umsetzung dieser Methoden in entsprechende Computerprogramme, wobei im letzteren Falle der Schwerpunkt auf dem strukturierten Programmieren und der Programm-Verifizierung liegt. Zum Erstellen, Editieren und Testen seiner Programme stehen dem Studenten in der Regel abgesetzte Datenstationen - entweder in der Form von Bildschirmgeräten oder von Fernschreibern - zur Verfügung, die im Timesharing von einer entsprechend leistungsfähigen Rechenanlage bedient werden.

Es ist von Kritikern eines Ausbildungsbetriebs an interaktiven Datenstationen im Timesharing oft nicht ganz zu unrecht eingewandt worden, daß dieser ein blindes Probieren und "Herumspielen" begünstigt, im Gegensatz zum Stapelverarbeitungsbetrieb, bei dem ein sehr viel stärkerer Zwang besteht, von Anfang an mit einem wohlkonzipierten Programm an den Rechner zu gehen. Auf der anderen Seite sind die Vorteile der interaktiven Programmentwicklung, unterstützt durch die sehr komfortablen und leistungsfähigen Editoren und Dateienverwalter moderner Betriebs-



systeme, so entscheidend - abgesehen von dem Vorteil, abgesetzte Datenstationen beliebig über ein Campus oder gar eine ganze Region verteilen zu können - daß jeder Gedanke an eine Rückkehr zum Stapelverarbeitungsbetrieb in der Programmierungsausbildung unrealistisch wäre. Dies wird in der (nahen) Zukunft, die ganz dem "personalized computing" gehören wird, in noch stärkerem Maße als bisher gelten.

Der Preis, der von Investitionsseite für einen Timesharingbetrieb mit einer großen Zahl von Datenstationen zu zahlen ist, ist sehr hoch. Abgesehen von den Kosten für die Datenstationen (gegenwärtig mindestens etwa DM 10.000,--) und den Kosten für die Miete der Leitungen und Modems muß die Rechenanlage selbst stärker mit Speicherkapazität und Kapazität für "peripheral processing" ausgerüstet sein, als es der reine Stapelverarbeitungsbetrieb erfordern würde. Auf der anderen Seite führt der reine Timesharingbetrieb zu einer sehr schlechten Auslastung des teuersten Teils der Rechner-Hardware, der CPU. Der Ausnutzungsgrad der Anlage wird zwar in der Regel dadurch verbessert, daß man zusätzlich zum Timesharingbetrieb noch eine Hintergrund-Stapelverarbeitung durchführt, aber es bleibt die grundsätzliche Tatsache bestehen, daß für den Timesharingbetrieb erhebliche Mehraufwendungen (verglichen mit einer reinen Stapelverarbeitungs-Konfiguration) erforderlich sind, während der Wirkungsgrad der Anlage durch den zusätzlichen "overhead" deutlich absinkt. Der Komfort und die sehr viel bessere Verfügbarkeit des Timesharing kostet damit einen nicht unbeträchtlichen Preis.



Die Vorteile des Terminalbetriebs in der DV-Grundausbildung sind aber so groß, daß man bisher fast immer bereit ist, diesen hohen Preis zu zahlen. Um ein Beispiel unter vielen zu nennen, sei die Situation aufgezeigt, wie sie vor etwa zwei Jahren an der University of Minnesota, einer der führenden Staats-Universitäten der U.S.A. mit ca. 70.000 Studenten, bestand. Hier findet man neben vielen kleineren Anlagen (viele davon ebenfalls im Timesharing betrieben) im zentralen Rechenzentrum der Universität eine CDC CYBER 72, die nur für die Ausbildung im Timesharing betrieben wird, sowie eine CDC CYBER 74, die sowohl den Stapelverarbeitungsbetrieb wie auch ebenfalls ein gewisses Maß an Timesharingbetrieb durchzuführen hat. Daneben betreibt der Staat Minnesota ein zusätzliches Timesharingnetz für alle staatlichen Universitäten (darunter die University of Minnesota), Fachhochschulen und Oberschulen, das auf einer Zweiprozessoranlage UNIVAC 11 10 basiert. Bekanntlich ist eine gewisse DV-Grundausbildung - überwiegend unter Verwendung von BASIC (horribile dictu!) - in den U.S.A. fester Bestandteil des Mathematik-Curriculums an jeder besseren high school. Für diesen Timesharing-Service für "educational purposes" hat der Staat Minnesota einen besonderen, sehr erheblichen Haushaltsposten eingerichtet, so daß dieser Service den Benutzern gebührenfrei angeboten wird (dies ist ungewöhnlich für die U.S.A., wo auch an den Staats-Universitäten die Studenten in der Regel für jede Leistung eine gewisse Gebühr entrichten müssen).

Aufgrund der immer stärker werdenden Inanspruchnahme des Timesharing-Service, verbunden mit dem schlechten Wirkungsgrad die-



ser Betriebsform, sah sich die Rechenzentrumsleitung bereits Mitte der siebziger Jahre gezwungen, von einer "Tarifpolitik", die ursprünglich die Benutzung des Timesharing-Service stimulieren sollte, zu einer restriktiven Politik umzuschalten. Nur die CYBER 72 wurde noch für Ausbildungszwecke kostenfrei zur Verfügung gestellt. Die Zahl der verfügbaren Programmiersprachen und der Umfang des dem einzelnen Studenten eingeräumten Speicherbereichs ("workspace") wurde stark beschränkt. Damit wurde ein wesentlicher Vorteil des Vielfachzugriffs zu einer großen Anlage aufgegeben, nämlich die Möglichkeit, auch große Programme bearbeiten zu können und eine Anzahl von Benutzern Programme und Daten gemeinsam benutzen zu lassen. Allerdings fielen diese Restriktionen deshalb nicht so sehr ins Gewicht, als diese Möglichkeiten in der DV-Grundausbildung ohnehin kaum genutzt werden. Im Timesharing-Service, der auf der CYBER 74 bereitgestellt wird, wurde die bestehende große Vielfalt von Sprachen und Programmsystemen sowie die Möglichkeit, sehr große Programme bearbeiten zu können, beibehalten. Dieser Dienst steht allerdings nicht für die Grundausbildung zur Verfügung, sondern vorwiegend für die Forschung und spezielle Kurse des Hauptstudiums. Die Benutzung ist gebührenpflichtig, d. h. bei der Bemühung um Forschungsmittel, sei es von einer Institution der Forschungsförderung oder von der eigenen Universität, muß man sich auch um die benötigten Mittel zur Rechnerbenutzung bemühen.

Wir haben dieses Beispiel so ausführlich betrachtet, weil wir glauben, daß es charakteristisch für jede größere Universität ist, nicht



nur in den U.S.A. Es ist sicher falsch zu behaupten, daß die Forschung an den deutschen Universitäten ihre Heimstatt verloren hat. Es ist aber sicher nicht falsch anzunehmen, daß durch die Notwendigkeit der Bewältigung steigender Studentenzahlen und die dabei unvermeidbare stärkere "Verschulung" der Universität das Verhältnis von Lehre und Forschung sich zu Ungunsten der Forschung entwickelt hat. Da von der Möglichkeit des Rechnereinsatzes in der Forschung praktisch in allen wissenschaftlichen Disziplinen seit Jahren voll Gebrauch gemacht wird, dürfte der Bedarf an Rechnerkapazität an den deutschen Universitäten in den letzten Jahren von der Forschungsseite her kaum gestiegen sein. Wenn trotzdem allenthalben die Universitäts-Rechenzentren ihre vorhandenen Anlagen als "zu klein" melden und Wünsche nach größeren und leistungsfähigeren Nachfolgesystemen anmelden, dann kann das nur daran liegen, daß immer mehr Disziplinen in immer stärkerem Umfang DV-Grundausbildung betreiben.

In den U.S.A. war wegen der restriktiven Wissenschaftsförderungspolitik der siebziger Jahre der einfache Ausweg einer Erweiterung der Rechnerkapazitäten für viele Universitäten verschlossen. An der oben als Beispiel angeführten University of Minnesota wurde daher Anfang 1977 erwogen, durch Einführung einer generellen Kostenpflicht, auch für die DV-Grundausbildung, den Anteil der Time-sharing-Nutzung zu reduzieren, um genügend Rechnerkapazität für die Forschung (für deren Nutzung echt bezahlt werden muß), übrig zu behalten. Allerdings brauchten in der Folge solche einschneidenden Maßnahmen nicht verwirklicht zu werden, dank der Entwicklung, von der in diesem Beitrag die Rede sein soll.



2. ANFORDERUNGEN AN AUTONOME MIKROPROZESSOR-SYSTEME FÜR DIE DV-GRUNDAUSBILDUNG

Die Anforderungen, die an ein Rechnersystem für die DV-Grundausbildung zu stellen sind, lassen sich wie folgt zusammenfassen.

- (1) Unbegrenzte Verfügbarkeit. Das System sollte dem jeweiligen Benutzer praktisch unbegrenzt zur Verfügung stehen. Die bei den Timesharingzeiten bestehende Situation, wonach es zu den Hauptzeiten oft unmöglich ist, in das System hineinzukommen oder aber, falls einem dies doch gelingt, die Antwortzeiten unerträglich groß werden, sollte möglichst vermieden werden.
- (2) Geeigneter Mensch-Maschine-Dialog. Das System sollte dem Benutzer die Führung eines Dialogs auf einer hinreichend hohen Ebene gestatten, unabhängig von irgendwelchen Ideosynkrasien der Hardware (Kommandosprache!). Dies bezieht sich auf alle Anweisungen zur Dateienverwaltung, Compilierung und Programmausführung.
- (3) Geeignete höhere Programmiersprachen. Es muß diejenige höhere Programmiersprache implementiert sein, die man aufgrund ihrer syntaktischen Klarheit und Sauberkeit und ihres didaktischen Nutzens als das geeignete Vehikel für die Einführung in die Konzepte des strukturierten Programmierens ansieht.
- (4) Rechenleistung. Für die relativ einfachen Programmieraufgaben, die gewöhnlich in der DV-Grundausbildung den Studenten gestellt werden, genügt eine relativ bescheidene Rechenleistung.



Wichtiger als eine hohe Rechenleistung ist eine gute Übersetzungseffizienz des Compilers.

- (5) Speicherkapazität. Für die in der DV-Grundausbildung typischen relativ einfachen Programmieraufgaben genügt eine für heutige Begriffe verhältnismäßig geringe Kapazität des dem Benutzer verfügbaren Arbeitsspeichers. Ein Speicherraum von z. B. 64 K-bytes ist in der Regel mehr als ausreichend. Selbstverständlich muß zusätzlich dazu aber noch ein Sekundärspeicher für die Archivierung von Programmen und Daten vorhanden sein.

Bis vor etwa zwei Jahren war es unmöglich, ein solches System als autonome Anlage zu erstellen. Die dramatische Entwicklung der LSI-Technologie, die zu sehr billigen Speichern und Mikroprozessoren geführt hat, macht dies jedoch neuerdings möglich, zu einem Preis, der nur unwesentlich über dem Preis eines entsprechenden "unintelligenten" Terminals liegt. Dies hat die folgenden Gründe:

Auch ein unintelligentes Datensichtgerät benötigt eine Stromversorgung, ein Fernsehraster-Sichtgerät, eine Tastatur (der teuerste Einzelbaustein), einen Prozessor und einen Speicher. Was hinzukommt, um aus einem solchen Datensichtgerät ein autonomes Computersystem zu machen, ist im wesentlichen: 1. eine Erhöhung der Speicherkapazität und 2. der Einbau einer Floppy Disk. Dadurch erhöhen sich die Hardware-Kosten um etwa 50 %.

In den U.S.A. befinden sich solche Systeme seit etwa zwei Jahren auf dem Markt. Ihr Preis liegt gegenwärtig zwischen 5500 und 8000 Dollar.



Die Kosten für Amortisierung und Verzinsung eines solchen Gerätes belaufen sich bei einem Ausnutzungsgrad, der dem der bisher üblichen Timesharing-Terminals entspricht, auf etwa \$ 0,50 pro Stunde! An Sprachen stehen vor allem PASCAL, daneben auch BASIC, FORTRAN IV und APL zur Verfügung. An der oben als Beispiel angeführten University of Minnesota wurde im Frühjahr 1977 begonnen, solche Systeme versuchsweise einzusetzen. Nachdem sich zeigte, daß ihr Nutzen alle Erwartungen weit übertraf, befinden sich inzwischen eine große Zahl solcher Tisch-Computer im Einsatz. Es ist zu erwarten, daß solche Systeme spätestens 1979 auch in Europa eingeführt sein werden, zu einem Preis, der etwa bei DM 15.000 liegen mag. Es ist ferner zu erwarten, daß die genannten Preise sich (nach heutigem Kurs und Geldwert) spätestens 1982 auf ca. 3.000 Dollar bzw. DM 8.000,-- erniedrigt haben werden.

3. DAS UCSD PASCAL - SYSTEM

Die ungewöhnlich schnelle Verbreitung, die die genannten Tisch-Computer in den vergangenen 1 - 2 Jahren in den U.S.A. fanden, ist vor allem das Verdienst von Professor K. L. Bowles und seinen Mitarbeitern am Institute for Information Systems der University of California, San Diego (UCSD). Diese Gruppe entwickelte ein ungewöhnlich gutes und leistungsfähiges PASCAL-System für den Einsatz auf Mikro- oder Minicomputern, das im folgenden hier etwas näher besprochen werden soll.



Das UCSD PASCAL - System [1] ist ein vollständiges, interaktives Software - System für kleine und kleinste Rechner, das aber alle wesentlichen Eigenenschaften besitzt, die man sonst bei mittleren oder großen Systemen findet. Das System befindet sich noch in ständiger, aufwärtskompatibler Erweiterung und wird für eine Reihe von Mikroprozessoren und Minicomputern voll von der UCSD von einer eigens dafür eingerichteten Gruppe hauptberuflicher Mitarbeiter unterstützt. Gegenwärtig umfaßt das UCSD-System im wesentlichen die folgenden Komponenten:

- ein "single user" - Betriebssystem
- ein PASCAL - Compiler für standard PASCAL, erweitert durch spezielle Datentypen für die Zeichenketten-Manipulation (string operations) sowie Anweisungen für die Bildschirmprogrammierung (graphics), die Handhabung von Platten-Dateien und die Systemprogrammierung. Damit ist das USCD-PASCAL eine recht mächtige Obermenge des Wirth'schen PASCAL [2].
- ein äußerst komfortabler Editor für Textverarbeitung und Programm-Editieren auf dem Bildschirm. Daneben gibt es einen zeilenorientierten Editor für das Arbeiten mit Fernschreib-Terminals.
- eine Dateienverwaltung für die Unterhaltung einer beliebigen Bibliothek von Dateien auf der Platte (floppy disk)
- interaktive Programm-Testhilfen (debugger), die es erlauben, Programm-Haltepunkte (break points) zu setzen, einzelne Programmzeilen auszuführen und die momentanen Werte von Variablen abzufragen.
- Dienstprogramme zum Drucken, für die Datenfernübertragung, für die Diagnose von Fehlern bei der Datenübertragung von der Platte,



für den Tischrechnerbetrieb, und anderes mehr.

- ein BASIC-Compiler, der es erlaubt, Unterprogramme, die in PASCAL geschrieben sind, in BASIC einzufügen (es gibt in den U.S.A. immer noch Leute, die glauben, BASIC in der DV-Grundausbildung oder sogar in der Anwendungsprogrammierung (MUMPS!) verwenden zu müssen).

Weitere Systemkomponenten, die in naher Zukunft verfügbar sein werden, sind:

- eine Adaption des Programmpakets der University of California, Irvine, für den rechnerunterstützten Unterricht (CAI). Dieses Paket wird auch ein Lehrprogramm für PASCAL enthalten.
- Assemblierer für verschiedene Mikroprozessoren wie z. B. INTEL 8080, ZYLOC Z 80, LSI/11.

Ferner werden im Augenblick Konzepte diskutiert, um PASCAL durch weitere Datentypen¹⁾ zu erweitern, die in besonderem Maße die Spezifizierung und Implementierung beliebiger Datentypen und damit die Programmierung von Datenbanksystemen und Programmen für die betriebliche Datenverarbeitung (business EDP) unterstützen.

Das UCSD-System ist weitgehend maschinenunabhängig implementiert. Dies ist dadurch erreicht worden, daß nach einem auf N. Wirth zurückgehenden Konzept zunächst eine abstrakte Maschine definiert wurde, die sog. P-Maschine.

1) Unter einem Datentyp verstehen wir eine bestimmte Datenstruktur und eine Menge von Standard-Operationen auf dieser Struktur.



Gegenwärtig existieren Implementierungen des UCSD-Systems für die Mikroprozessoren INTEL 8080, ZYLOC Z 80, DEC LSI-11 sowie für die Minicomputer der Firmen DIGITAL EQUIPMENT (PDP 11), GENERAL AUTOMATION (GA 16), UNIVAC (V 75) und NANODATA(QM-1). Das UCSD-System wird von der University of California für die nominelle Gebühr von 200 \$²⁾ abgegeben. Prinzipiell kann sich jeder ein System unter Verwendung eines der genannten Minicomputer, eines Fernsehraster-Display mit Tastatur und einer Floppy Disk selbst aufbauen und dazu von UCSD die Software für 200 \$ beziehen. Es gibt aber komplette Systeme (GENERAL AUTOMATION, TERAC) zu dem bereits genannten Preis von ca. 6.000 \$ in den U.S.A. oder ca. DM 15.000, -- in Deutschland. Dieser Preis ist niedriger als die Summe der OEM-Preise für die Komponenten des Systems. Außerdem haben solche Systeme den Vorteil, daß die Systemverantwortung und die Wartung vom Hersteller übernommen werden.

Für die DV-Grundausbildung ist als Leistungszahl die Übersetzungsleistung entscheidend. Hierfür wird das TERAC-System, das mit einem LSI-11 - Prozessor und 56 K-bytes Speicher arbeitet, eine Leistung von ca. 600 Programmzeilen pro Minute angegeben [4]. Das GA 216 - System von GENERAL AUTOMATION [5] enthält einen schnelleren Prozessor und 64 K-byte Speicher und erreicht damit eine Übersetzungsleistung von ca. 2000 Programmzeilen pro Minute.

2) Die University of California kann ihrer Satzung nach kein eigenes Einkommen haben. Deshalb ist der genannte Betrag ein reiner Unkostenbeitrag, der auch die Programm-Wartung einschließt.



4. DAS ARBEITEN MIT DEM UCSD-SYSTEM

Das Arbeiten mit dem UCSD-System ist durch die große Benutzer-freundlichkeit des Betriebssystems so einfach, daß ein Student kaum einer Anleitung dazu bedarf. Die hauptsächlichen Module des Systems werden durch einfache Kommandos aktiviert, wobei ein solches Kommando im Drücken einer bestimmten Taste der ASCII-Tastatur besteht. Soll z. B. der Editor aktiviert werden, so drückt man die Taste "E", usw. Zur Erinnerung schreibt das System ständig in der obersten Zeile des Bildschirms die Namen der hauptsächlichen Kommandos und ihre Bedeutung aus; d. h. man findet die Anzeige:

Command: E(edit, R(un, F(ile, C(ompile, X(ecute, D(ebug,
I(nit, H(alt

Von diesen Hauptkommandos an verzweigt sich das Kommandosystem baumförmig.

Die Eingabe des Kommandos X (execute) führt dazu, daß das System zunächst nach dem Namen einer Datei fragt, die ein ausführbares Programm (Objektcode) enthält. Nachdem der Benutzer diesen Namen eingegeben und die Wagenrücklauf-Taste gedrückt hat, beginnt das System mit der Programmausführung. I (nit dient dazu, das System zu reinitialisieren. H(alt beendet eine Programmausführung ordnungsge-mäß und stoppt den Prozessor.

Während der Programmentwicklung legt sich der Benutzer vorüber-gehend eine Anzahl von Platten-Dateien an, die gemeinsam als "Ar-beitsdatei" (workfiles) bezeichnet werden.



Am Ende eines Editierungsprozesses legt der Editor automatisch (by default) den editierten Text - gewöhnlich ein Quellprogramm - in einer zu den Arbeitsdateien gehörenden "Textdatei" ab. Der Compiler (durch das Kommando C aktiviert) ruft den Inhalt der Textdatei als Eingabe ab und versucht ihn zu übersetzen. Ein auf diese Weise übersetztes Programm wird in eine "Codedatei" eingetragen, und das nachfolgende Kommando R (run) startet dann die Ausführung dieses Programms. Das in der Textdatei gespeicherte Programm kann jederzeit wieder mit Hilfe des Editors verändert, werden, und anschließend kann unmittelbar das R(un - Kommando gegeben werden. In diesem Falle ruft das System automatisch zunächst den Compiler auf, und eine Programmausführung wird nur nach erfolgreich durchgeföhrter Compilation gestartet. Mit Hilfe einer Dateienverwaltung kann der Benutzer die einzelnen Komponenten der Arbeitsdatei in eine festangelegte Platten-Datei retten, oder aber Dateien von dort in die Arbeitsdatei laden, oder Dateien umbenennen, usw. Der Dateienverwalter wird durch das Kommando D (datei) aufgerufen.

Bei Aufruf des Editors wird in der obersten Zeile des Bildschirms wieder ein "Kommando-Menü" ausgeschrieben, das jetzt lautet:

)Edit: A(djst C(py D(lete F(ind I(nsrt J(mp R(place Q(uit
X(chng Z(ap

Die Textstelle, an der eine Veränderung vorgenommen werden soll, wird wie üblich durch eine Schreibmarke (cursor) markiert. Der Editor führt alle Funktionen des Formatierens (Wortverarbeitung!), Indentierens, Einfügens, Auslöschen, Ersetzens, Kopieren, usw. aus. Diese Funktionen können durch die oben angegebenen Kommandos aufgerufen werden. Auf nähere Einzelheiten brauchen wir hier nicht



einzugehen, da der Leser sicher mit ähnlichen Bildschirm-Editoren (z. B. dem des UNIX-Systems) vertraut ist.

Ein besonders nützliches Programm ist auch der "debugger", der es erlaubt, einzelne Programmzeilen auszuführen bzw. die aktuellen Werte von Variablen abzufragen. Der Debugger erlaubt es ferner, an beliebigen Programmzeilen Haltepunkte (break points) zu setzen. Während einer solchen Programm-Suspension können die an dieser Stelle geltenden Werte von Variablen examiniert werden, und die Programmausführung kann anschließend fortgesetzt werden.

Besonders eindrucksvoll sind auch die graphischen PASCAL-Erweiterungen im UCSD PASCAL. Diese Anweisungen erlauben eine Bildschirmprogrammierung im Rahmen der für Fernsehraster-Displays gegebenen Grenzen [6] (die typische Bildschirmauflösung beträgt bei den auf dem Markt befindlichen Systemen 192 x 280 Punkte). Dadurch können Rechenergebnisse in vielfältiger Weise in der Form von Kurven und Diagrammen veranschaulicht werden.

Zusammenarbeit der Benutzergruppen

Die University of California, San Diego, übernimmt nicht nur die Wartung des UCSD PASCAL-Systems, sondern vermittelt auch den Austausch von Anwendungs- und Lehrprogrammen zwischen den Benutzern des Systems (Software/Courseware Exchange). Hierzu gehört auch die zur Verfügungstellung eigener Lehrprogramme. Es kann erwartet werden, daß dieser Austausch für alle Anwender des Systems von Nutzen sein wird, besonders aber bei seinem Einsatz in der DV-Grundausbildung.



Die Zusammenarbeit der wachsenden Zahl von Benutzern kommt auch in einer kürzlich in San Diego abgehaltenen Benutzerkonferenz zum Ausdruck (21.-25.8.1978), auf der Fragen einer wünschenswerten PASCAL-Erweiterung besprochen wurden. Im Prinzip hält sich das UCSD PASCAL zwar möglichst eng an die von Jensen und Wirth [2] veröffentlichte Sprachdefinition. Gewisse Erweiterungen sind aber mit der immer stärkeren Verbreitung von PASCAL als Anwendungs- und Systemprogrammierungssprache nicht zu umgehen. Auf der genannten Konferenz wurde z. B. die Forderung nach einer getrennten Compilierbarkeit von Programmteilen, einer verfeinerten Kontrolle des Gültigkeitsbereichs von Namen bei der Parameterübergabe zwischen getrennt compilierten Programm-Modulen und anderem mehr erhoben. Das UCSD PASCAL stellt in seiner jetzigen Form bereits eine Erweiterung der von Jensen und Wirth definierten Sprache dar (Einbau des Standardtyps STRING, nichtsequentielle Dateiorganisation, read/write-Operationen). Eine neue Version, die in Kürze freigegeben werden wird, wird einige der auf der Benutzerkonferenz erarbeiteten Erweiterungsempfehlungen berücksichtigen. Für die Anwendung von PASCAL als Systemprogrammierungssprache ist die Einfügung von Synchronisationskonstrukten für konkurrente Prozesse in der Diskussion sowie die Hinzufügung geeigneter Bausteine für die Konstruktion von Datenstrukturen bzw. Datentypen. Es zeichnet sich bereits jetzt ab, daß die Aktivitäten von Professor Ken Bowles und seinen Mitarbeitern an der UCSD wesentlich dazu beigetragen, PASCAL zum verdienten Durchbruch zu verhelfen.



LITERATUR

- [1] UCSD-IIS UCSD PASCAL - A (Nearly) Machine Independent Software System for Micro and Mini Computers, University of California, San Diego, Institute for Information Systems (Feb. 1978)
- [2] Jensen, K. and Wirth, N., PASCAL User Manual and Report, Springer-Verlag, 1975
- [3] UCSD-IIS, A Brief Description of the USCD PASCAL Software System, University of California, Institute for Information Systems (June 1978)
- [4] TERAC CORPORATION, terac 8510/a Graphics Computer System, Publ. No. 52-0001-00 terac corporation, 14405 North Road, Scottsdale, Arizona 85260
- [5] GENERAL AUTOMATION, Systembeschreibung 216/10 Program Development System
- [6] Giloi, W. K., Interactive Computer Graphics, Prentice-Hall 1978
- [7] Bowles, K.L., Mikrocomputer Based Mass Education (to be published)
- [8] Bowles, K.L., PASCAL versus COBOL - Where PASCAL Gets Down to Business, BYTE 3, 8 (Aug. 1978), 122-142



EIN NEUER TISCHRECHNER

GA 216/10 PDS

FÜR STRUKTURIERTE PROGRAMMENTWICKLUNG

GENERAL AUTOMATION

AACHEN - VERLAUTENHEIDE



H I N T E R G R U N D

Immer mehr wissenschaftliche Disziplinen nehmen eine Grundausbildung in Datenverarbeitung in ihre Studienpläne auf.

U M F A N G D E R G R U N D A U S B I L D U N G

- Einführung in eine höhere Programmiersprache
- Einführung in Methoden zur Problemlösung mit dem Computer
- Einführung in Methoden für die Umsetzung von Problemlösungen in Computerprogrammen (strukturiertes Programmieren, Programmverifikation)

O R G A N I S A T I O N D E R G R U N D A U S B I L D U N G

Datenstationen, die im Time-Sharing-Betrieb von dem im Universitätsrechenzentrum vorhandenen Großrechner bedient werden.

NACHTEILE: Die Rechnerkapazität könnte im Stapelbetrieb wirkungsvoller genutzt werden.

Hohe Investitionen für

- Anschaffung von Datenstationen
- Miete für Leitungen und Modems
- Erweiterung der Speicherkapazität
- Erweiterung der Kapazität für "Peripheral Processing"

VORTEILE: Interaktive Programmentwicklung
Unterstützung durch Editoren
Unterstützung durch Dateienverwalter
Geographische Verteilung von Datenstationen



FORDERUNG

Universitätsrechenzentren erheben Anspruch auf größere und leistungsfähigere Nachfolgesysteme.

BEISPIEL U N I V E R S I T Y O F M I N N E S O T A

<u>Rechenzentrum</u>	CDC CYBER 72 (Time-Sharing - Grundausbildung)
	CDC CYBER 74 (Stapelbetrieb und Time-Sharing)
	UNIVAC 1100 (Time-Sharing - Ausbildung im Staat Minnesota)

Einschränkungen Mitte der 70er Jahre

	Gebühren
	Weniger Sprachen
	Weniger Workspace (keine großen Programme)
	Kein Program/Data Sharing
ZIEL:	Einschränkung der Time-Sharing-Nutzung für die Grundausbildung

<u>Lösung</u>	Einsatz von autonomen Mikroprozessor-Systemen für die DV-Grundausbildung.
---------------	--

Seit Frühjahr 1977 sind mehrere Systeme im Einsatz, ihr Nutzen übertraf alle Erwartungen.

ZIEL:	Gesamte DV-Grundausbildung mit Mikroprozessoren
-------	---

<u>Nebenprodukt</u>	Interessante Forschungsprojekte
---------------------	---------------------------------



ANFORDERUNGEN AN AUTONOME MIKROPROZESSOR -
SYSTEME FÜR DIE DV - GRUNDAUSBILDUNG

- Unbegrenzte Verfügbarkeit

 Antwortzeiten

 Log-In

- Geeigneter Mensch-Maschine-Dialog

 Kommandosprache für Dateienverwaltung

 Compilierung

 Programmausführung

- Geeignete höhere Programmiersprache

 Strukturiertes Programmieren

- Rechenleistung

 Relativ bescheiden (Grundausbildung)

 Hohe Übersetzungseffizienz

- Speicherkapazität

 Relativ gering (z. B. 64 K-Bytes)

 Sekundärspeicher für Programm- und

 Datenarchivierung

Ein derartiges System kann mit LSI-Technologien zu einem Preis hergestellt werden, der nur unwesentlich über dem Preis eines entsprechenden "unintelligenten" Terminals liegt (erhöhte Speicherkapazität, Floppy Disk)



IN DEN U.S.A. EXISTIERENDE SYSTEME

PREISE	\$ 5.000 - 8.000
AMORTISIERUNG/VERZINSUNG	\$ 0,5 pro Stunde
SPRACHEN	PASCAL BASIC FORTRAN IV APL

EUROPA

Einführung solcher Systeme - 1979

PREIS	DM 15.000
PREISERWARTUNG 1982	DM 8.000



PASCAL PROJEKT DER UNIVERSITY OF CALIFORNIA,

SAN DIEGO - UCSD PASCAL SYSTEM -

Leistungsfähiges PASCAL System für den Einsatz auf autonomen Mikro- und Minicomputern.

Aufwärtskompatible Erweiterung für eine Reihe von Mikro- und Minicomputern.

Weitgehend maschinenunabhängige Implementierung, basierend auf einer abstrakten Kellermaschine (P-Maschine) und BIOS (Basic Input Output System)

KOMPONENTEN

- Single User Betriebssystem
- PASCAL Compiler mit Standard PASCAL Erweiterungen
 - Datentyp STRING
 - Bildschirmprogrammierung
 - Handhabung von Plattendateien
 - Systemprogrammierung
- Editoren (Bildschirm, Fernschreib-Terminal)
- Dateienverwaltung (Floppy Disk)
- Interaktive Programm-Testhilfen (Debugger)
- Dienstprogramme für Drucken
 - Datenfernübertragung
 - Diagnose bei Datenübertragung v. d. Platte
 - Tischrechnerbetrieb



- BASIC Compiler
 - binden von PASCAL Unterprogrammen mit
BASIC Programmen
- Programmpaket für rechnergestützten Unterricht
 - Lehrprogramm für PASCAL, University of
California, Irvine⁺
- Assembler für verschiedene Mikroprozessoren
 - INTEL 8080, ZILOG Z80, LSI-11, GA-16/110⁺
- Erweiterung von PASCAL mit Datentypen für die
Implementierung von Datenbanksystemen⁺

Preis \$ 200 (Unkostenbeitrag, einschließlich Wartung)

+ Geplant für 1979



DAS GA 216/10 PDS SYSTEM

Komplettes Program Development System mit UCSD PASCAL

Preis ca. 17.000 DM

(niedriger als die OEM-Preise für die Komponenten des Systems)

Übersetzungsleistung: 600 Programmzeilen pro Minute

Mensch-Maschine Interface

Kommandosprache: Schlüsselwörter, z. B. E(DIT,
C(OMPILE

Kommandodictionär: Baumstruktur
ständig auf dem Bildschirm gezeigt

Dateienverwaltung: Arbeitsdateien
Textdateien
Plattendateien (permanent)

Editoren: vergleichbar mit UNIX Editor

Debugger: Ausführung einzelner Programmzeilen
Ausgabe von Variablen-Werten
Break Points

Graph. System: Fernsehraster-Display +
PASCAL - Erweiterungen +
Kurven- und Diagrammdarstellung +

+ Geplant für 1979



Systemkomponenten

LSI-Prozessor GA-16/110

64 bis 128 K-Bytes RAM

Bootstrap Loader (ROM),

Power Fail/Restart

Doppellaufwerk für Floppy Disks

31 cm Bildschirm

programmierbare Schreibmarke (CURSOR)

Vorrat von 128 - 256 Zeichen (ASCII)

Graphik mit hoher Auflösung

(ab 2. Quartal 1979)

Tastatur

Geplante Erweiterungen

Schnelldruckeranschluß

5 - 30 Mega Byte Platte

Communication Controller

(Anschluß an einen Großrechner)

Prozeß - Ein-/Ausgabe



SOFTWARE/COURSEWARE EXCHANGE FÜR DAS
UCSD PASCAL SYSTEM

Austausch von Anwendungs- und Lehrprogrammen zwischen den Benutzern des Systems

Austauschmedia: Telefon, Platten, Magnetbänder, Lochstreifen

Austauschgegenstand: Programme, Dokumentation, Nachrichten der Benutzergruppe (Wartung)

Durch die große Anzahl der Mitglieder in der Benutzergruppe wird erwartet, daß die Kosten für Software-Entwicklung unter den Kosten der jeweiligen Hardware bleiben.

Informationen über Erweiterungen des Systems werden durch einen Newsletter Service an die Mitglieder der Benutzergruppe geleitet.

UCSD führt einen Dialog über standardisierte Erweiterungen für PASCAL.



216/10 Programm Development System

Tischcomputer für strukturierte Programmentwicklung und Softwareausbildung mit dem PASCAL-System der UCSD*

* UNIVERSITY OF CALIFORNIA, SAN DIEGO



- Autonomes Computersystem
- Bildschirmorientierter Mensch - Maschine - Dialog
- Speicherkapazität 128 K-byte
- Integriertes Floppy-Disk Doppellaufwerk

- Interaktive Compiler für höhere Programmiersprachen
- Hohe Übersetzungsgeschwindigkeit
- Graphische Darstellung

GENERAL AUTOMATION



Warum autonome Tischcomputer

Die Grundausbildung in der Datenverarbeitung wird durch den Einsatz von leistungsfähigen MikroComputern immer mehr vom bisherigen Time-Sharing-Terminal-Betrieb auf „intelligente“ Terminals bzw. autonome Tischcomputer übergehen.

Dafür gibt es eine Reihe von Gründen:

- höhere Verfügbarkeit eines Tischcomputers
- Vermeidung der Wirkungsgradprobleme des Großrechners im Time-Sharing-Betrieb
- vergleichbarer Software-Komfort, insbesondere für die Ausbildung
- bessere Wirtschaftlichkeit im Vergleich zum Time-Sharing-Betrieb

Weitere Vorteile

Daneben bieten sich Systeme mit leistungsfähigen Sprachen als modernes, unabhängiges Hilfsmittel bei der computergestützten Lösung von Aufgaben an Hochschulen und wissenschaftlichen Instituten an.

Die stürmische Entwicklung der LSI-Technologie macht den Übergang vom „intelligenten“ Terminal zum Tischcomputer jetzt zu einem nur leicht erhöhten Preis möglich. Was hinzukommt, ist im wesentlichen mehr „Intelligenz“, dazu Hauptspeicherkapazität und der Einbau eines Massenspeichers in Form von Floppy Disks.

... und warum PASCAL?

PASCAL hat neben den Sprachen BASIC und APL wesentlich zu der ungewöhnlich schnellen Verbreitung von Tischcomputern in den USA beigetragen. Dies gilt insbesondere für das hier implementierte, ungewöhnlich gute und leistungsfähige PASCAL-System, das an der University of California, San Diego (UCSD) entwickelt wurde.

Es zeichnet sich aus durch

- ein „single-user“ Betriebssystem
- einen PASCAL-Compiler, erweitert durch STRING-Operationen und graphische Anweisung* zu einer mächtigen Obermenge des Wirth'schen PASCAL
- einen äußerst komfortablen Bildschirm-Editor
- eine Datenverwaltung für beliebige Bibliotheken
- interaktive Programm-Texthilfen (u. a. Break-Points)
- Druck- und Datenfernübertragungsprogramme

* ab 1979 im GA 216/10 verfügbar

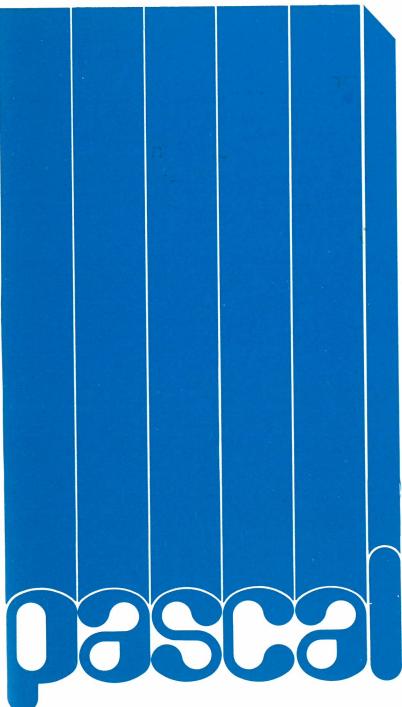
Systemkomponenten

- LSI-Processor GA-16 /110,
500 ns Mikrozyklus
- 64 bis 128 K-byte RAM (16 bit
und byte parity),
400 ns Zugriffszeit
- Bootstrap Loader (ROM),
Power Fail/Restart
- Doppellaufwerk für Floppy Disks
- 31 cm Bildschirm; 1920 Zeichen
zu 24 Zeilen mit je 80 Zeichen
mit 50 Hz Bildwiederholfrequenz
- programmierbare Schreibmarke
(cursor), Zeichendarstellung
normal, invertiert oder blinkend
- Vorrat von 128 Zeichen (ASCII
Groß-/Kleinschreibung),
erweiterbar auf 256 Zeichen,
Zeichendarstellung und Graphik
mit hoher Auflösung
- Tastatur: 52 Schreibmaschinen-
tasten, 29 Funktionstasten,
6 LED Funktionstasten

Das 216/10 Programm Development System interessiert
mich. Ich bitte um

- weitere Informationen
- einen Besuch Ihres Mitarbeiters
- eine Vorführung
- ein Angebot über _____ Einheiten

Gewünschtes bitte ankreuzen.



Technische Spezifikation

Zentraleinheit:

- nMOS LSI-Technologie
- 500 ns Mikrozykluszeit und 2,0 μ s durchschnittliche Befehlausführungszeit
- Statusanzeige
- Asynchroner Speicher Daten-Bus
- 128 K-byte direkt adressierbar (Befehle und Daten)
- Halbleiterspeicher RAM/PROM/ROM
- Elf Adressierungsarten
- 16 allgemeine Arbeitsregister, (2 pre-index, 6 post-index)
- Hardware Multiplikation/Division
- vollständiger Satz arithmetischer und logischer Befehle

- Spannungsausfallerkennung / automatischer Wiederstart
- Systemfehlererkennung
- Paralleler asynchroner Daten-Bus
- Programmierte Ein-/Ausgabe

Speicher: nMOS 400 ns Zugriffszeit, Dualport

Laufwerke: Kapazität formattiert 204.800 byte pro Platte
Transfergeschwindigkeit: 31 K-byte/s, mittlere Zugriffszeit: 370 ms

Umgebungsbedingungen: 0 - 40° Temperaturbereich
5 - 80% Luftfeuchte

Anschlußwert: 230 V \pm 10%,
47 - 63 Hz
1000 Watt
Leistungsaufnahme



GENERAL AUTOMATION

Heider-Hof-Weg 23
5100 Aachen
Tel. 02405/64241

Absender:

Antwortkarte



Reminder:

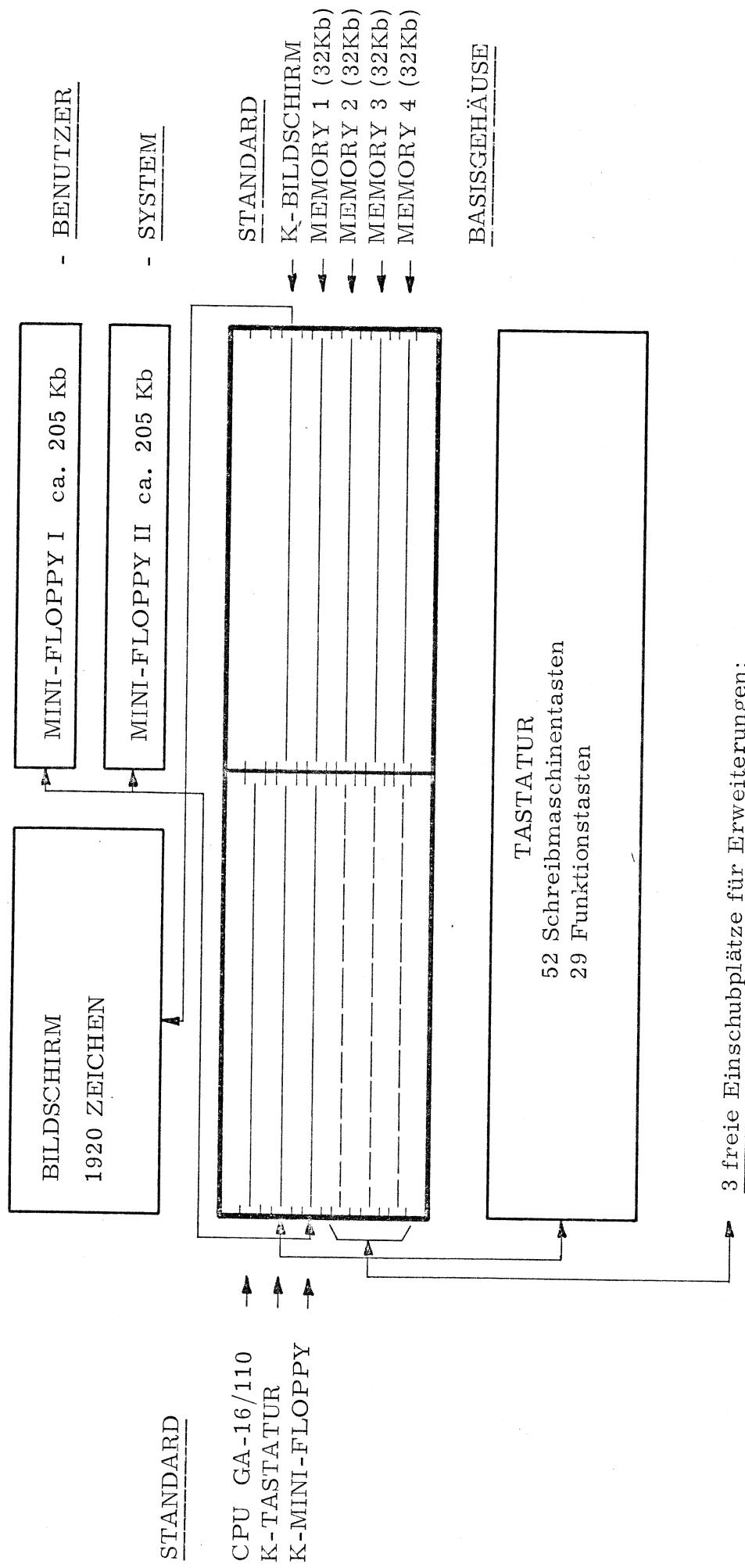
Am _____ habe ich Unterlagen bei General Automation über den Tischcomputer 216/10 angefordert.

GENERAL AUTOMATION

Heider-Hof-Weg 23
5100 Aachen-Verlautenheide

pascal

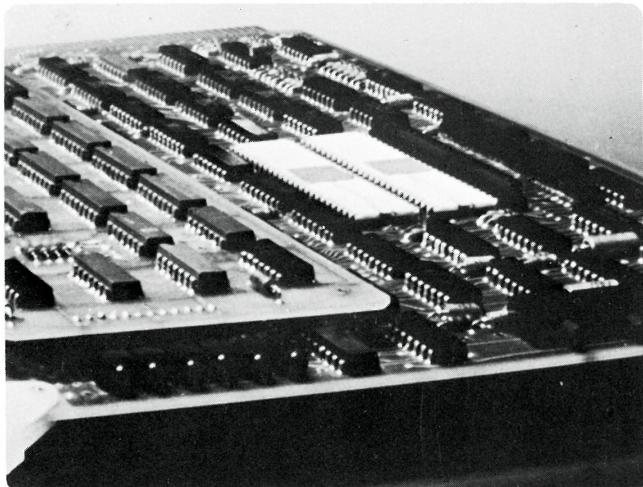
Physikalische Zusammenhänge im
GENERAL AUTOMATION PDS 216/10⁺



GENERAL AUTOMATION

SOLUTION SERIES

GA-16/110 GA-16/220 Microcomputer



Neue Maßstäbe

Die neuen Microcomputer von General Automation - GA-16/110 und GA-16/220 - setzen neue Maßstäbe hinsichtlich Leistung und Wirtschaftlichkeit auf dem LSI-Microcomputer- und Microprozessormarkt. Die Rechner sind sowohl für Spezialsysteme als auch für den OEM-Markt entwickelt worden. Diese vollständigen 16-Bit Rechner mit ihrer für Microprozessoren außergewöhnlich kurzen Befehlausführungszeit und ihrem großen Befehlsvorrat halten jeden Vergleich stand.

Der GA-16/110 ist auf einer einzigen kleinen Steckkarte aufgebaut und in dieser Form schon voll funktionsfähig. Aufgrund seiner Abmessungen und der geringen Leistungsaufnahme ist er vor allem zur Integration in komplexe Systeme geeignet.

Das Modell GA-16/220 ist durch Erweiterungen aus dem GA-16/110 entstanden. Unter anderem unterscheidet er sich durch ein Mikrokonsolprogramm, Teletype Controller, serieller Ein-/Ausgabe und einen Kanal mit direktem Speicherzugriff.

EIGENSCHAFTEN

Die Rechner GA-16/110 und GA-16/220 sind mit 16 allgemeinen 16-Bit Registern ausgestattet. Diese Register sind in zwei Gruppen aufgeteilt, die das schnelle Umschalten von einem Registersatz auf den anderen ermöglichen. 6 Register werden als Index-Register benutzt, davon 2 zur basis-relativen Adressierung und 2 als Subroutine-Linking.

GA-16/110 und GA-16/220 Standard Eigenschaften

- nMOS LSI-Technologie
- 500 nsec Micro-Zykluszeit und 2,5 microsec. durchschnittliche Befehlausführungszeit
- Microprogrammierbarer Prozessor
- Statusanzeige
- Asynchroner Speicher Daten Bus
- 64 K Worte direkt adressierbar (Befehle und Daten)
- Halbleiterspeicher RAM/PROM/ROM/EPROM
- Speichererweiterung bis zu 65.536 (16-Bit) Worten
- Elf Adressierungsarten
- 16 allgemeine Arbeitsregister, (2 pre-index, 6 post-index)
- Hardware Multiplikation/Division
- Vollständiger Satz arithmetischer und logischer Befehle
- Zwei Interrupt-Klassen (GA-16/220): NI- externe Ein-/Ausgabe-Interrupts
- Unbegrenzte Anzahl von externen Interrupt-Ebenen (64 Standard)
- Spannungsausfallerkennung/automatischer Wiederstart
- 1 msec Echtzeituhr (nur 16/220)
- Zeitüberwachung (OMA)
- Systemfehlererkennung
- Paralleler asynchroner Daten Bus
- Programmierte Ein-/Ausgabe
- DMA Kanal (nur 16/220)
- Mikrokonsolprogramm (nur 16/220)
- Teletype/Sichtgerät Steuerwerk (nur 16/220)
- Anschluß für Fernbedienung
- Register Inhalte können als Befehl ausgeführt werden
- Register-Register Transfer für besonders schnelle Operationen
- Wort-, Byte- und Bit-Adressierung



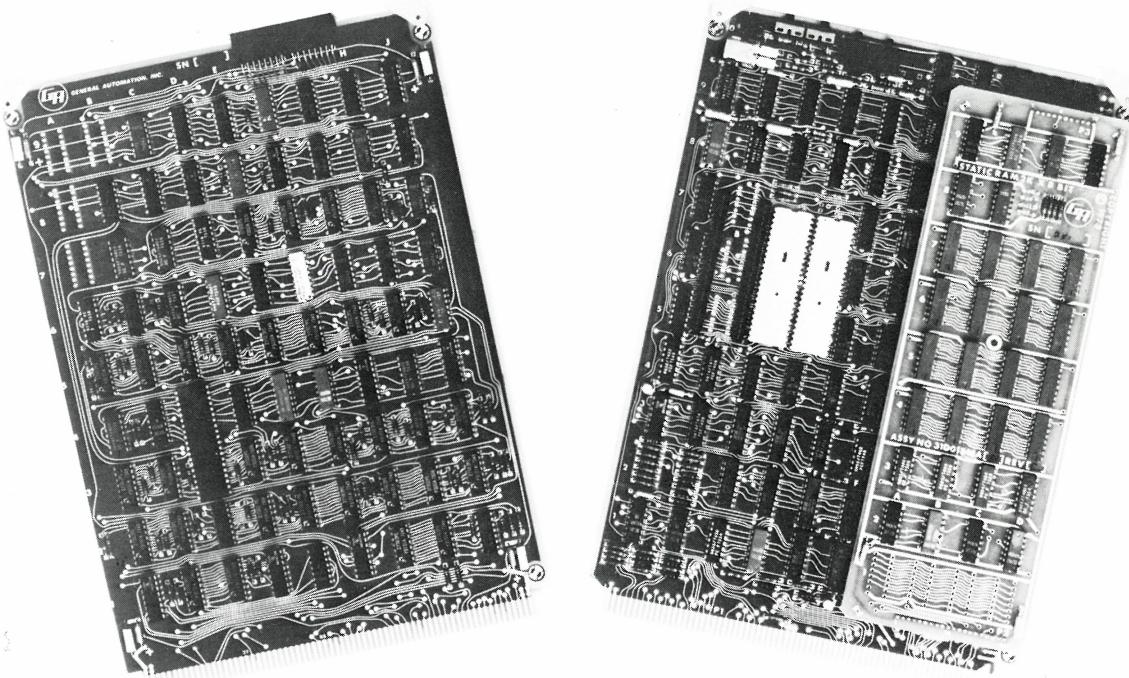


Abbildung 1 zeigt den Microcomputer GA-16/220, der aus zwei Platinen besteht.

GA-16/110 und GA-16/220 Optionen

- Interrupt Zeitüberwachung
- Interrupt für Speicherpartitätsfehler und Analyse
- Interrupt für Schreibversuch im geschützten Speicher einschl. Analyse
- Schnelle vorzeichengerechte Multiplikation/Division
- Gleitkomma-Arithmetik
- Automatische Ladeprogramme
- Mehr als 100 verschiedene DMA- und programmierte E/A-Steuerwerke

JSR	Unterprogramm Sprung	2,0 µ sec
RTR	Rücksprung aus Unterprogramm	2,5 µ sec
SKP	Bedingter Sprung	2,0 µ sec
SBIT	Bit Setzen	3,2 µ sec
STBY	Byte Speichern	2,5 µ sec
DTOM	Daten aus Speicher (Ein-/Ausgabe)	3,0 µ sec

LEISTUNGSMERKMALE UND TECHNOLOGIE

Das Herz der GA-16/110 und GA-16/220 Rechner sind zwei microprogrammierte LSI- (Large Scale Integration) Mikroprozessoren in n-Kanal Silicon MOS Technologie. Während ein Prozessor die Register, Arithmetik und Logik Einheit (RALU) darstellt, arbeitet der zweite Prozessor als Control Read Only Memory (CROM). In diesen Chips sind die Organisation und der Befehlsvorrat microprogrammiert. Die Microbefehle sind 34 Bit lang, und nur so sind die Schnelligkeit und die Leistung der Rechner zu erklären.

Hier ein paar Beispiele für Befehlausführungszeiten, die sich aus dem Microzyklus von 500 nsec zusammensetzen:

Befehl	Ausführungszeit
LDR/STR Laden oer Speichern beliebiger Register	2,5 µ sec
ADDC Registeraddition mit Vergleich	2,5 µ sec

Kompatibilität zur gesamten Rechnerpalette von General Automation.

Die GA-16/110 und GA-16/220 Rechner sind Hardware- und Software kompatibel. Alle Programme laufen auf allen Rechnern. Alle Programme laufen auf allen Rechnern.

Der GA-16/110 ist ein besonders preiswerter, jedoch leistungsstarker Rechner, der in erster Linie für Systeme entwickelt wurde, d. h. er arbeitet als Teil einer Anlage, indem er Steuerungs- und Überwachungsaufgaben übernimmt. Die Programme für dieses System werden entweder auf einem SPC-16 oder auf einem GA-16/440/330 oder 16/220 erstellt. Im Vergleich zum GA-16/110 kann der GA-16/220 hardware- und softwaremäßig zu einer vollständigen Rechenanlage erweitert werden, auf der alle GA Betriebssysteme laufen mit Ausnahme des Memory Management Systems des GA-16/440. Die Ein-/Ausgabeschnittstellen der GA-16/110/220-Systeme sind identisch zum SPC-16 und zum GA-16/440/330; deshalb können alle vorhandenen Steuerwerke des SPC-16 auch in diesen Systemen benutzt werden. DMA-(Direkter Speicher Zugriff) Steuerwerke sind jedoch nur an den GA-16/220 anschließbar.

64 K Adressierung

Die Speicher der GA-16/110/220 Modelle können bis zu 65.536 Wörtern direkt adressiert werden. Das vereinfacht das Programmieren für größere Systeme, die mehr als 32 K Worte Speicher benötigen. Zusätzlich stehen elf Adressierungsarbeiten für den gesamten Speicherbereich zur Verfügung.

Adressierungsarten:

- Direkt
- Direkt, indiziert
- Indirekt
- Indirekt, indiziert
- Programm relativ
- Programm relativ indirekt
- Basis relativ
- Basis relativ, indiziert
- Basis relativ, indirekt
- Basis relativ, indirekt, indiziert
- Literal

Die vielseitigen Adressierungsarten sind einmalig auf dem Microcomputermarkt.

Speicher

Die GA-16/110 und GA-16/220 Systeme können mit Halbleiter-Speichern wie folgt ausgebaut werden:

Speicherplatten, die auf das Rechnerboard gesteckt werden:

- 2 K x 16 Bit RAM (Lesen und Schreiben)

Auf jedem Aufsteckspeicher können 64 Wörter mit Schreibschutz für Urlader oder Anwenderprogramme reserviert werden.

Speicherplatten, die in das Ein-/Ausgabe-System gesteckt werden:

- 16 Bit - 8 K RAM (Lesen und Schreiben)
- 18 Bit - (Byte Parity - 8 K oder 16 K RAM
(Paritygenerierung auf jedem Speichermodul)
- 22 Bit (Bit Fehler Korrektur) 8 K RAM (in der Entwicklung)

Speicher Parity und Schreibschutz (MPP) – Option –

Die Speicher Parity Einrichtung generiert einen nicht verhindbaren (NI)-Interrupt, wenn ein Parityfehler im Speicher auftritt. Voraussetzung: Das System muß mit 18 Bit Speichermodulen bestückt sein.

Die Schreibschutz-Einrichtung ermöglicht das Schützen von Speicherbereichen gegen Schreibversuche. Der Schutz bezieht sich sowohl auf Programme als auch auf den direkten Kernspeicherzugriff (DMA). Es können per Programm 1 K Worte Segmente – insgesamt 64 Bereiche – als Schutzbereich definiert werden; ein dynamisches Ein- und Ausschalten der Bereiche ist möglich. Der Schreibversuch in einem geschützten Bereich erzeugt einen nicht verhindbaren (NI)-Interrupt.

KONSOLE

Das Prozessor Board ist mit einer Microkonsole bestückt. Sie besteht aus Schaltern und LED Anzeigen. Damit können Daten und Adressen eingegeben, Programme gestartet und der Status abgelesen werden. Der GA-16/110 ist nur mit einer „Reset“ Taste ausgerüstet.

Zur komfortablen Bedienung steht ein ROM Programm zur Verfügung, welches ermöglicht, mit einer Teletype oder einem Sichtgerät direkt mit dem Rechner im Dialogverkehr zu arbeiten. So sind folgende Funktionen möglich:

- Anzeigen und Ändern von Register-Inhalten
- Anzeigen und Ändern aller Speicherzellen-Inhalte
- Laden und Stanzen von Binärstreifen
- Programm-Einzelschritt
- Programmierter Programm-Halt (Trap)
- Programmstart

Neben dem obenerwähnten ROM Bedienprogramm werden 256 Wörter RAM Speicher zum Speichern von Informationen wie Status, Register, Flags usw. bereitgestellt. Ein Teil dieses Speichers kann mit Programmtesthilfen belegt werden. Zusätzliche 256 Wörter PROM Speicher können Urlader für sämtliche GA-Eingabegeräte wie Teletype, Lochstreifenleser, Kartenleser, Floppy Disk, Plattspeicher etc. aufnehmen. Diese 256 Wörter können jedoch auch vom Benutzer für eigene Prozeduren belegt werden.

INTERRUPTS

Die GA-16/110 und GA-16/220 unterscheiden zwei Gruppen von Interrupts. Die auftretenden Interruptbedingungen werden nach dem Prioritätsprinzip abgearbeitet. Jeder Interrupt-Ebene ist ein Interruptvektor zugeordnet.

NI – nicht verhindbare Interrupts

- Netzspannungsaufall
- Automatischer Wiederstart
- Cold Start
- Speicher Parity Fehler
- Speicher Schutz
- Zeitüberwachung des Interrupt Systems
- Nicht vorhandener Befehlscode

Zum Schutze des Betriebssystems und der Anwendersoftware sind die oben aufgeführten Interruptbedingungen nicht maskierbar, d.h. nicht verhindbar. Da diese Bedingungen nur bei nicht einwandfrei laufendem System auftreten, werden diese Fehler durch Hardware Einrichtungen erkannt und analysiert. Während eines NI-Interrupts schützt die Hardware das aufgerufene Programm und sichert den Maschinen-Status.

Ein-/Ausgabe Interrupts

Diese Gruppe dient zur Überwachung externer Geräte. Da jede Speicherzelle einen Interrupt-Vektor darstellen kann, verfügen die GA-16/110 und GA-16/220 über eine nahezu unbegrenzte Anzahl von Interruptebenen. Die Ebenen unterliegen einer Prioritätssteuerung und sind einzeln maskierbar. Außerdem ist es möglich, die Prioritäten bei Bedarf per Software dynamisch zu ändern.

BEFEHLSVORRAT

Der Befehlsvorrat des GA-16/110 und GA-16/220 ist mit dem des SPC-16 kompatibel; es sind jedoch neue Befehle hinzugekommen. Bei der Auslegung des Befehlsvorrates hat man besonders auf die Anforderungen in der Industrieautomatisierung Rücksicht genommen.

- Bit Befehle, wie Testen, Setzen oder Zurücksetzen einzelner Bits im Speicher. Besonders zur Überwachung digitaler Statussignale geeignet.

MECHANISCHER AUFBAU

- Byte Befehle, Adressierung von rechten und linken Bytes über den gesamten Bereich. Besonders zur Textverarbeitung geeignet, da ein Zeichen durch ein Byte dargestellt wird.
- Register-Register und Speicher-Register Befehle einschl. arithmetischer und logischer Befehle.
- Speicher-Register-Vergleiche zum schnellen Suchen bestimmter Werte im Speicher
- Execute-Register – Register Inhalte können als Befehl ausgeführt werden. Gerade in Verbindung mit Read Only Memories (ROM) können mit dieser Eigenschaft Programmabläufe ständig geändert werden.
- Sprünge in Unterprogramme und Rücksprünge über ein Register, das die Rücksprungadresse enthält. Das Retten der Rücksprungadresse per Hardware verringert den Programmaufwand.
- Ein-Wort-Befehle zum Laden oder Sichern aller Register vereinfachen die Programmierung für re-entrant Programme und Interrupt-Verarbeitung.

SOFTWARE

Die Betriebssysteme des GA-16/220 sind auf die Hardwareeigenschaften abgestimmt und gelten als besonders leistungsfähig. So stehen auch höhere Programmiersprachen und komfortable Programmierhilfen zur Verfügung: z.B. Macro Assembler, erweitertes FORTRAN IV, ANSI COBOL, BASIC und viele Hilfsprogramme.

Folgende Betriebssysteme sind auf dem GA-16/220 Rechner lauffähig:

- FSOS-16 – Free Standing Operating System, ein speicherresidentes Betriebssystem zur Programmerstellung
- CONTROL I – Plattenorientiertes Stapelverarbeitungssystem
- CONTROL II – Speicherresidentes Echtzeitbetriebssystem
- CONTROL III – Plattenorientiertes Echtzeitbetriebssystem mit Foreground/Backgroundverarbeitung, einschl. Multiprogramming und Stapelverarbeitung.

Der GA-16/110 wird mit dem Betriebssystem FSOS-16 und einer speziellen Version von CONTROL II unterstützt; Außerdem stehen Ladeprogramme zur Verfügung (Echtzeitbetriebssysteme auf Anfrage).

EIN-/AUSGABE

Der GA-16/110 besitzt zwei Ein-/Ausgabe Datenkanäle und zwar einen programmierten Ein-/Ausgabe Kanal zum Anschluß von SPC-16 Steuerwerken und einen DMT Kanal (Direkter Speicher Transfer). An diesen Kanal sind zwei DMA Geräte anschließbar.

Der GA-16/220 besitzt zwei zusätzliche Ein-/Ausgabe Kanäle und zwar einen seriellen (110 Baud oder 9600 Baud) für Linienstrom oder V 24 Schnittstelle und ein SPC-16 kompatibles DMA Steuerwerk zum Anschluß von schnellen Peripheriegeräten.

Dem OEM-Kunden werden sowohl einzelne Karten, die vollständige Rechner darstellen, als auch komplette Systeme mit Netzversorgung, Ein-/Ausgabe-Einschüben und Verkabelung geliefert. Neben der Kartenversion sind folgende Systeme lieferbar:

- GA-16/220 mit Kompaktgehäuse für 6 Ein-/Ausgabe-Karten und integriertem Netzteil.
- GA-16/220 mit großem Gehäuse, bis zu 64 K Speicher, 9 Ein-/Ausgabe-Steckerplätze, zusätzliches Ein-/Ausgabe-Gehäuse für 18 Karten und abgesetztes Netzgerät.

Alle Gehäuse sind von vorne zugänglich.

Abmessungen: 22,2 cm hoch, 48,3 cm breit. Eine Batteriepufferung für beide Systeme ist lieferbar.

SPEZIFIKATIONEN

Aufbau

- Microprogrammiert
- 16 Bit-Register und Daten-Bus
- Zweier-Komplement-Arithmetik
- Software- und E/A-kompatibel zur GA-16 Serie und zum SPC-16

Technologie

- | | |
|-------------|-------------------------------|
| - Prozessor | - LSI n-Kanal MOS |
| | 500 nsec Microzyklus |
| - Speicher | |
| RAM | - nMOS 500 nsec Zykluszeit |
| PROM | - Bipolar 100 nsec Zykluszeit |
| - | - Multilayer Technik, 4 Lagen |

Halbleiterspeicher

- Dynamische Speicher, RAM Steckkarte 8 K und 16 K x 16 Bit/18 Bit/22 Bit Automatisches „Refreshing“ auf dem Modul
- Aufsteckkarte RAM 2 K statisch RAM + 64 Cold Start PROM
- 64 K Worte max. Speichergröße (PROM auf Anfrage)

Interrupts

- Unbegrenzte Anzahl von prioritätsgesteuerten Interrupt-Ebenen



NI – (nicht verhindernbare) Interrupts

- Spannungsausfall
- Automatischer Wiederstart
- Speicher Parity und Schreibschutz Interrupts
- Zeitüberwachung des Interrupts
- Cold-Start
- Nicht vorhandener Befehlscode

Ein-/Ausgabe-Interrupts

- Echtzeituhr
- Bedienkonsole „Bereit“
- Console Interrupt
- Externe Interrupts

EIN-/AUSGABE

Programmierte Ein-/Ausgabe (110 & 220)

- 53 parallele Leitungen für Adressierung, Daten und Steuerfunktionen
- Maximale Länge 16 m
- Prioritäten abhängig vom Kartensteckplatz
- 64 Gerätadressen von der Software unterstützt
- 120 K Worte/sec. max. Transferrate
- Transfer: Speicher und Register in beiden Richtungen (Ein-/Aus)
- Kompatibel mit allen SPC-16 Controllern (nicht DMA)

DMA-Kanäle (nur 220) – direkter Speicherzugriff

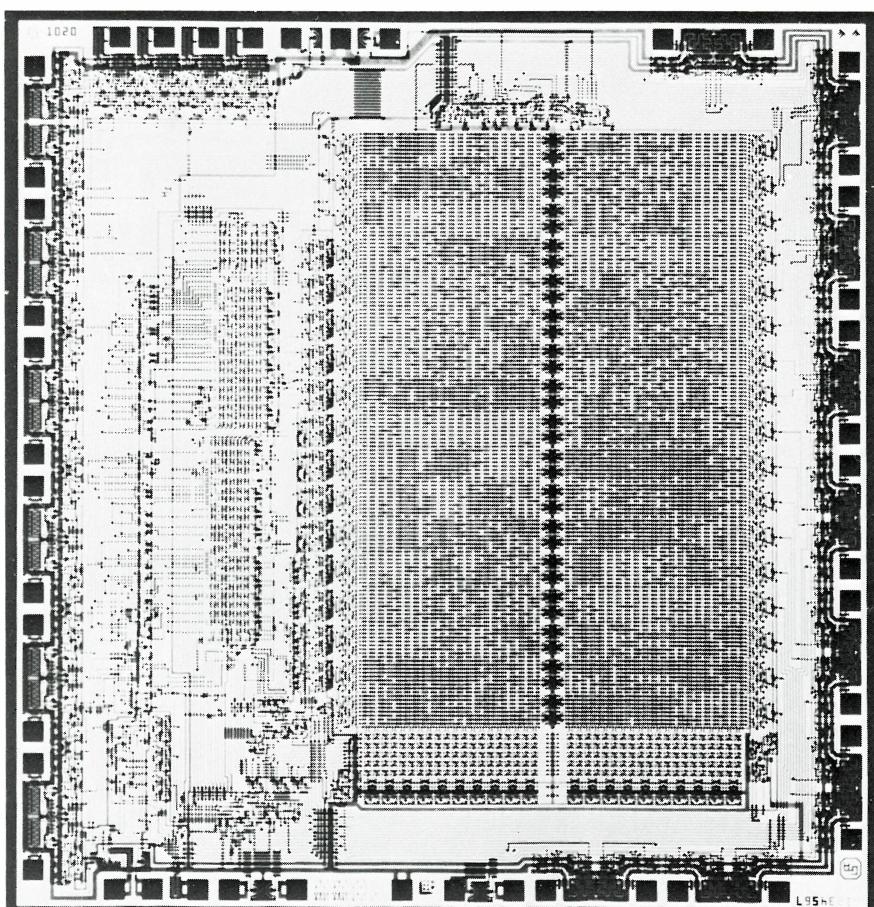
- 74 parallele Leitungen für Adressierung, Daten und Steuerfunktionen
- Maximale Länge 8,30 m
- Prioritätsverwaltung
- Cycle Stealing
- 1,0 M Worte max. Transferrate
- Alle SPC-16 DMA-Geräte anschließbar

DMT-Kanal (110)

- Anschlußmöglichkeit für zwei DMA-Geräte an Memory-Bus
- Controller müssen in unmittelbarer Nähe vom Speicher und Prozessor installiert werden
- Cycle Stealing
- 2,0 M Worte/sec. Transferrate

Serielle Schnittstelle (nur 220)

- Drei-Drahtleitung zum Anschluß von Geräten mit serieller Schnittstelle
- 110 Baud für Teletype
- 9600 Baud für Sichtgerät
- Linienstrom- oder V 24 Anschlußkarten



Vergrößerung des GA-16/110-220 Microprozessor-Chips CROM, das den Microcode einschl. erweitertem Befehlsvorrat des SPC-16 enthält.



Abmessungen

OEM-Boards

- Alle Boards 19,7 x 27,9 cm
- 110 Rechner 1 Board
- 220 Rechner 2 Boards
- 8 K + 16 K Speicher 1 Board
- Speicher Parity und Schreibschutzeinrichtung 1 Board
- Netzgeräteeinschub 1 Board mit 10,9 cm Breite
- Batteriepuffer 1 Board mit 8,1 cm Breite

OEM-Gehäuse

- Hauptgehäuse und zusätzliches E-/A-Gehäuse: 21,9 x 48,3 x 38,8 cm
- Hauptnetzgerät: 27,3 x 48,3 x 14,6 cm

Leistungsaufnahme

- Netzgeräteeinschub + 5 V 12 A
+ 15 V 2,5 A
- 15 V 1,5 A
- Hauptnetzgerät 220 Volt Wechselspannung,
47 bis 63 Hz
- Batteriepuffer 220 Volt Wechselspannung,
47 bis 63 Hz
- Batteriepuffer + 5 V und ± 15 V

Bei Spannungsausfall ungefähr 3 Stunden Reserve bei einem System mit 8 K Speicher und einer Batterie mit 1,5 Ah Kapazität. Anschluß von 12 V Autobatterien ist möglich.

Umgebungsbedingungen

- Betriebstemperatur 0° C bis 50° C
- Relative Luftfeuchtigkeit 0% bis 90% ohne Kondensation

GENERAL AUTOMATION

Heider-Hof-Weg 23 · 5100 Aachen · Tel.: 0 24 05 / 641

BRD

GENERAL AUTOMATION
Spaldingstraße 74
2000 Hamburg
Tel.: 0 40 / 2 45 141-3

GENERAL AUTOMATION
Ulmer Straße 16
7250 Leonberg b. Stuttgart
Tel.: 0 7152 / 410 41-2

GENERAL AUTOMATION
Erasmusstraße 8
6200 Wiesbaden
Tel.: 0 6121 / 8 80 81-4

GENERAL AUTOMATION
Auweg 10
8046 Garching
Tel.: 0 89 / 3 20 33 49

Österreich

GENERAL AUTOMATION
Schranngasse 14
5027 Salzburg
Tel.: 0 62 22 / 7 24 71-2

Schweiz

GENERAL AUTOMATION
Postfach 588
CH-8301 Glattzentrum/Wallisellen
Tel.: 01 / 830 65 50



Unterschiede zwischen

STANDARD PASCAL und U.C.S.D. PASCAL

Das Standard Pascal wird im PASCAL USER MANUAL AND REPORT (2. Ausgabe), herausgegeben von Kathleen Jensen und Nikolaus Wirth (Springer Verlag, 1975), beschrieben.

Die wesentlichen Unterschiede liegen im Bereich von FILES und der EIN-/AUSGABE generell. Als besonderen Vorteil möchten wir hier den RANDOM-Zugriff auf Plattendateien erwähnen.

Einzelheiten können dem UCSD PASCAL Handbuch entnommen werden (siehe Bezugsquellen).

Ein GENERAL AUTOMATION Handbuch, das vom Sprachumfang gesehen mit dem der UCSD identisch ist, befindet sich in Vorbereitung (Dezember 1978).



BEZUGSQUELLENNACHWEIS

TITEL	AUTOR	HERAUSGEBER
User Manual and Report	Kathleen Jensen Nikolaus Wirth	Axel Springer Verlag, Hamburg
Problem Solving Using PASCAL	Kenneth L. Bowles	Axel Springer Verlag, Hamburg
A Version of an Industry Pascal versus BASIC An Exercise	Carl Helmers	Byte Magazine Byte Publications Inc. 70 Main Street Petersborough N.H. 03458 /U.S.A.
Designing structured Programs		Ausgabe August '79
Pascal versus COBOL Where Pascal Gets Down to Business		
In Praise of PASCAL		
PASCAL A Structurally Strong Language		
Compilation and Pascal on the New Microprocessors		
UCSD (Mini-Micro Computer) Institute for Infor- PASCAL	mation-Systems University of San Diego	Institute for Informa- tion Systems UCSD Mailcode C-021 La Jolla, Cal. 92093 U.S.A. Tel. 001/714/452- 4723



GENERAL AUTOMATION

Heider-Hof-Weg 23 · 5100 Aachen · Tel.: 02405/641

BRD

GENERAL AUTOMATION
Spaldinger Straße 74
2000 Hamburg
Tel.: 040/245 141-3

GENERAL AUTOMATION
Ulmer Straße 16
7250 Leonberg b. Stuttgart
Tel.: 07152/41041-2

GENERAL AUTOMATION
Erasmusstraße 8
6200 Wiesbaden
Tel.: 06121/88081-4

GENERAL AUTOMATION
Auweg 10
8046 Garching
Tel.: 089/3203349

Österreich

GENERAL AUTOMATION
Schranngasse 14
5027 Salzburg
Tel.: 06222/72471-2

Schweiz

GENERAL AUTOMATION
Postfach 588
Ch-8301 Glattzentrum/Wallisellen
Tel.: 01/830 6550