DATAMATION 66

September

edp
in
europe

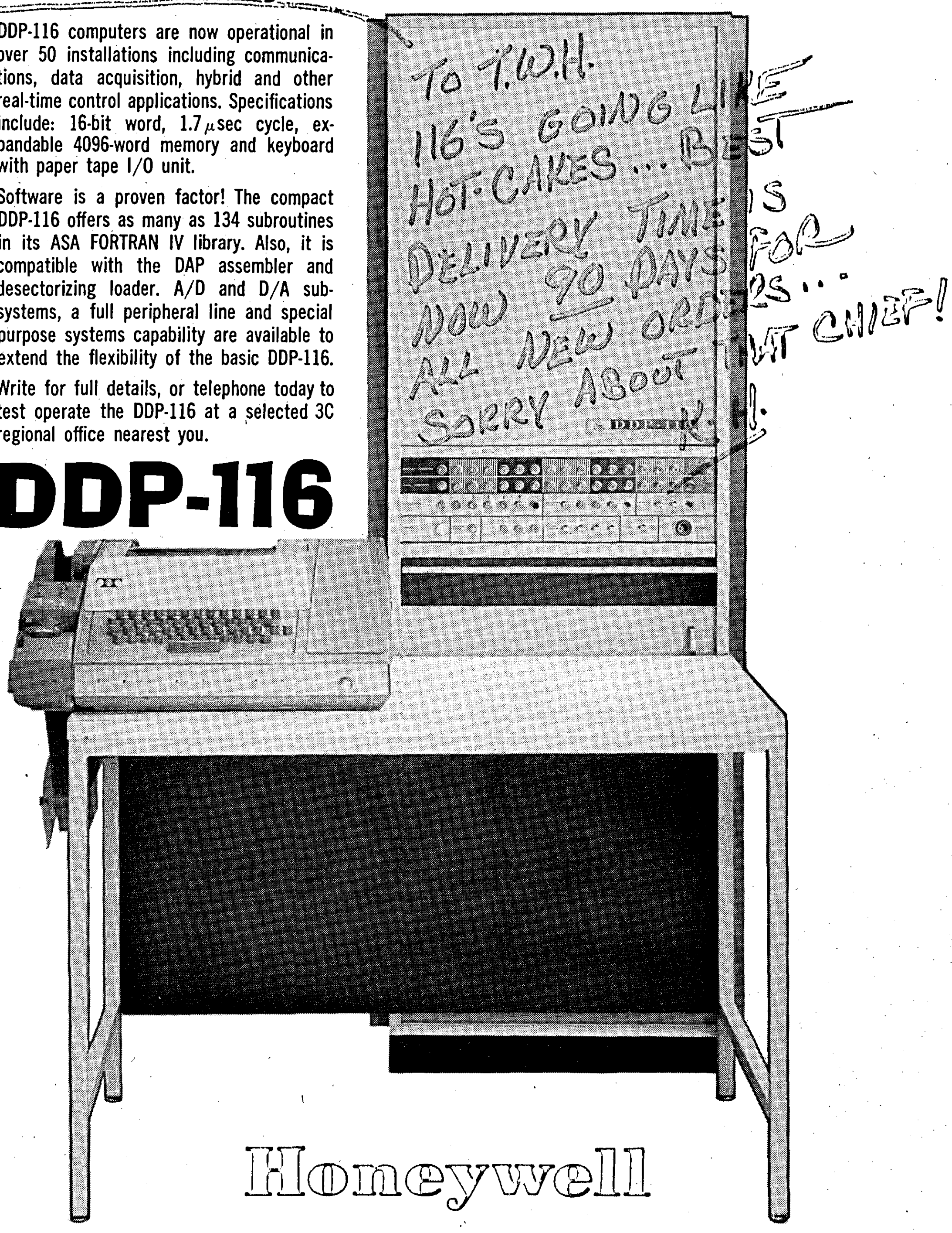
ONLY 3C OFFERS 30 DAY DELIVERY ON THE HOTTEST COMPUTER IN THE $28,500 PRICE RANGE

DDP-116 computers are now operational in over 50 installations including communications, data acquisition, hybrid and other real-time control applications. Specifications include: 16-bit word, 1.7 μsec cycle, expandable 4096-word memory and keyboard with paper tape I/O unit.

Software is a proven factor! The compact DDP-116 offers as many as 134 subroutines in its ASA FORTRAN IV library. Also, it is compatible with the DAP assembler and desectorizing loader. A/D and D/A subsystems, a full peripheral line and special purpose systems capability are available to extend the flexibility of the basic DDP-116.

Write for full details, or telephone today to test operate the DDP-116 at a selected 3C regional office nearest you.

DDP-116



SDS announces Sigma 2, a fat-free computer designed for systems.

Sigma 2 is a small, very fast, extremely reliable real-time computer with highly sophisticated software.

It costs $26,000 with Model 35 Teletypewriter, paper tape reader and punch, 4 fully buffered automatic I/O channels, and 4,096 words of core memory.

Memory is expandable to 65,536 words, all of which can be directly addressed. Cycle time is 900 nanoseconds.

Sigma 2 does multiprogramming and multiprocessing. It can control a real-time situation in the foreground while simultaneously performing a general-purpose job in the background—all with full memory protection. Re-entrant software greatly multiplies speed and efficiency. Sigma 2 can change its environment from one program to another in 4 microseconds.

With 20 input/output channels available, Sigma 2 can carry on many I/O operations simultaneously and very rapidly—up to 6,000,000 bits per second. A full word can be read in or out directly without the use of an I/O channel.

Memory protection is extremely flexible. Under program control, Sigma 2 can dynamically alter areas of protection while the machine is running. It takes only 2 microseconds to change protection for 4,096 words. Yet it is impossible for a background program to gain access to areas of memory under foreground protection.

Sigma 2 contains about ⅓ as many components as comparable machines. Integrated circuits, modular design and a unique logical organization make this possible. As a result, Sigma 2's standard of reliability is far beyond anything previously known in the industry. Even its typewriter is the most rugged machine on the market.

Sigma 2 is designed to handle such critical real-time applications as aerospace and industrial control, nuclear experimentation, and communications switching and control, and at the same time do general-purpose computation.

Also, Sigma 2 can serve as a local or remote satellite to its big brother, Sigma 7. It can use Sigma 7's memory in addition to its own, and it can operate all the Sigma 7 peripherals.

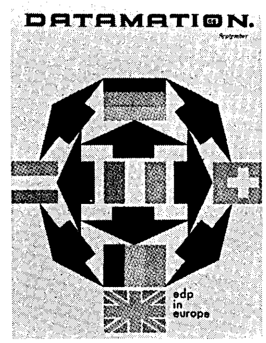
Software for Sigma 2 includes Basic Control Monitor, Basic FORTRAN, SDS FORTRAN IV, Real-Time Batch Monitor, basic and extended assemblers, and a library of mathematical and utility programs.

The first Sigma 2's will be delivered (with software) in 1966.

SDS

Scientific Data Systems, Santa Monica, California

CIRCLE 5 ON READER CARD



september
1966
volume 12 number 9

Publisher	GARDNER F. LANDON
Editor	ROBERT B. FOREST
Managing Editor & Art Director	CLEVE MARIE BOUTELL
Associate Editors	EDWARD K. YASAKI WILLIAM J. ROLPH
Assistant Editor	HEIDE H. THORSTENSON
Eastern Editor	ANGELINE PANTAGES
European Editor	CHARLES WHITE
Contributing Editors	HOWARD BROMBERG ROBERT V. HEAD
Washington Reporter	PHIL HIRSCH
Editorial Adviser	ROBERT L. PATRICK
Technical Consultant	LOWELL AMDAHL
Production Manager	MARILEE PITMAN
Director of Circulation	FRANK DeCARLO
Circulation Fulfillment	ALGENE TRAINA

Eastern District Managers	JAMES M. MORRIS WARREN A. TIBBETTS 141 E. 44th St., New York N.Y. 10017 [212] MUrray Hill 7-5180
New England District Manager & Vice President	WARREN A. TIBBETTS 112 West Haven Rd., Manchester, N.H., [603] NAtional 5-9498
Midwest District Manager	JOHN BRENNAN 205 West Wacker Drive, Chicago, Ill. 60606 [312] Financial 6-1026
Western District Manager & Vice President	HAMILTON S. STYRON 1830 W. Olympic Blvd., Los Angeles, Cal., 90006 [213] 385-0474

EDITORIAL OFFICES
1830 WEST OLYMPIC BLVD.
LOS ANGELES, CALIF. 90006

BPA
Circulation audited by Business Publications Audit

ABP
Member, American Business Press, Inc.

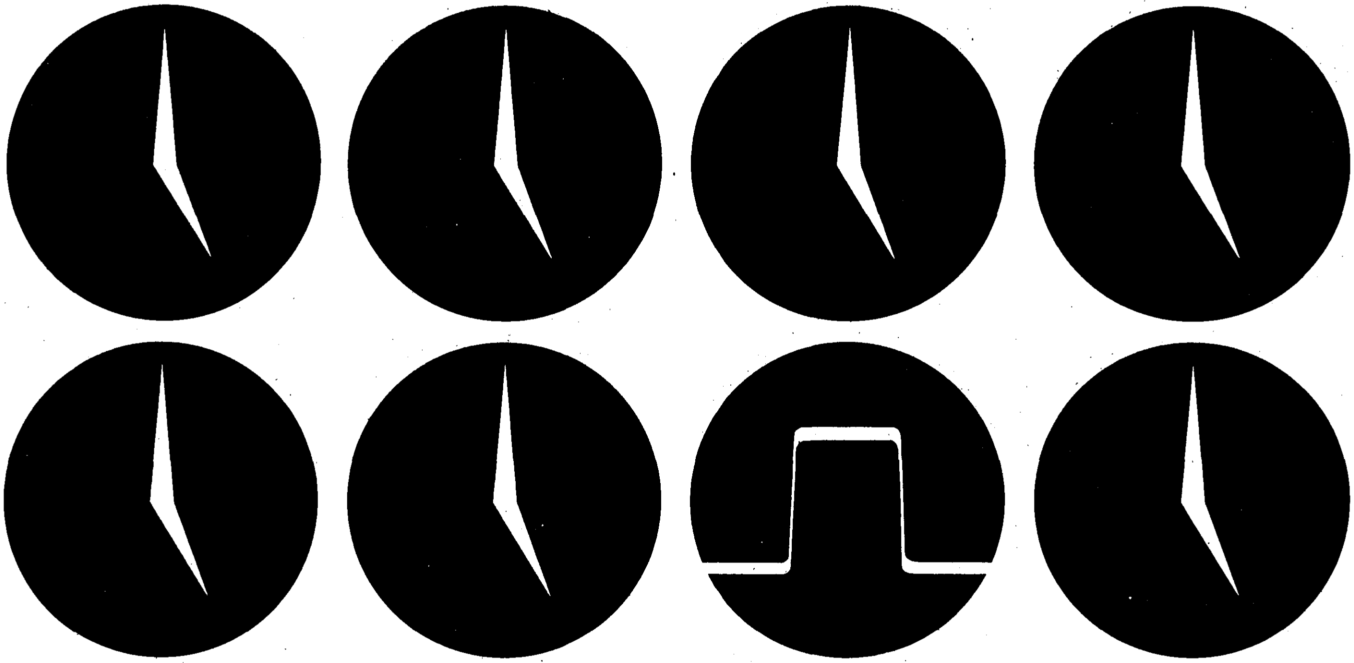
DATAMATION is published monthly on or about the tenth day of every month by F. D. Thompson Publications, Inc., Frank D. Thompson, Chairman; Gardner F. Landon, President; Gilbert Thayer, Senior Vice President: Executive, Circulation and Advertising offices, 141 East 44th Street, New York, New York 10017 (Murray Hill 7-5180). Editorial offices, 1830 W. Olympic Blvd., Los Angeles, California 90006. Published at Chicago, Ill.
DATAMATION is circulated without charge by name and title to individuals who qualify by our standards and are employed by manufacturers or users of automatic information-handling equipment in all branches of business, industry, government and the military. Available to others by subscription at the rate of $15.00 annually; single issues (when available) $1.50. Reduced rate for qualified students. Foreign subscriptions are on a paid basis only at a rate of $25.00 annually. No subscription agency is authorized by us to solicit or take orders for subscriptions. Controlled circulation paid at Columbus, O. and Chicago, Ill. Form 3579 to be sent to F. D. Thompson Publications, Inc., 205 W. Wacker Dr., Chicago, Ill. 60606.

Microfilm copies of DATAMATION may be obtained from University Microfilms, Inc.; 313 No. First St., Ann Arbor, Michigan.

Printed by Beslow Associates, Inc.

This issue 65,761 copies

A real-time, time-sharing systems computer



and a way to measure volts on 5 nanosecond pulses.

If you're in data acquisition, you can get either or both from Raytheon Computer.

Using a new disc-oriented, real-time monitor, the Raytheon 520 can simultaneously acquire and process real-time data, control a data system from up to 20 remote display-control stations and compile, assemble and execute FORTRAN IV and FLEXTRAN programs on a job or batch basis.

And the new NANOVERTER™ input device for high speed data systems includes a remarkable 5-nanosecond sampling device, and a 12-bit analog-to-digital converter for ± 2% accuracy at 45KC throughput.

With the monitor the 520 can respond to real-time interrupts, transfer data to core or disc via a direct memory access channel, transfer programs from disc to main memory and then shunt processed data to disc, magnetic tape, printer or other storage or output device.

The 520 monitor makes use of two unique features—direct memory access and dynamic memory protect including a memory map.

Direct memory access switches main memory in four microseconds between external devices—either peripheral equipment or another computer—without interrupting the 520 central processor.

Memory protect prevents inadvertent loss or output of stored real-time or batch processing data during interrupt program runs or job program compiling, assembly or execution. A special memory map keeps track of occupied and available memory locations in 2000-word segments and automatically assigns available memory to new programs.

Besides FORTRAN IV and FLEXTRAN, 520 software includes Real-Time FORTRAN IV. This separate and distinct processor is based on Raytheon's exclusive one-pass FORTRAN IV which is language compatible with the widely-used FORTRAN IV (version 13). Real-time FORTRAN IV simplifies the programmer's handling of real-time problems with features like RECURSIVE, PROTECT, CONNECT AND COUNT TIME Statements, and useful debugging aids like TRACE mode, Memory Map and DUMP.

The 520's one microsecond main memory boosts data acquisition word transfer rates to IMC. Besides the NANOVERTER, other real-time systems hardware includes the Multidevice Controller for interfacing as many as 512 external data systems devices and establishing up to 1024 levels of priority interrupt, and the unique Multiverter,™ which combines up to 96 channels of IC multiplexing, a 50-nanosecond sample and hold amplifier and analog-to-digital conversion in a single 5¼ inch chassis.

The Raytheon 520 is currently being specified and delivered for real-time and hybrid systems in the $100,000 class and up. Find out why by writing today for Data File C-132. Raytheon Computer, 2700 S. Fairview Street, Santa Ana, California, 92704.

RAYTHEON

Trademark of the Raytheon Company for its data system.

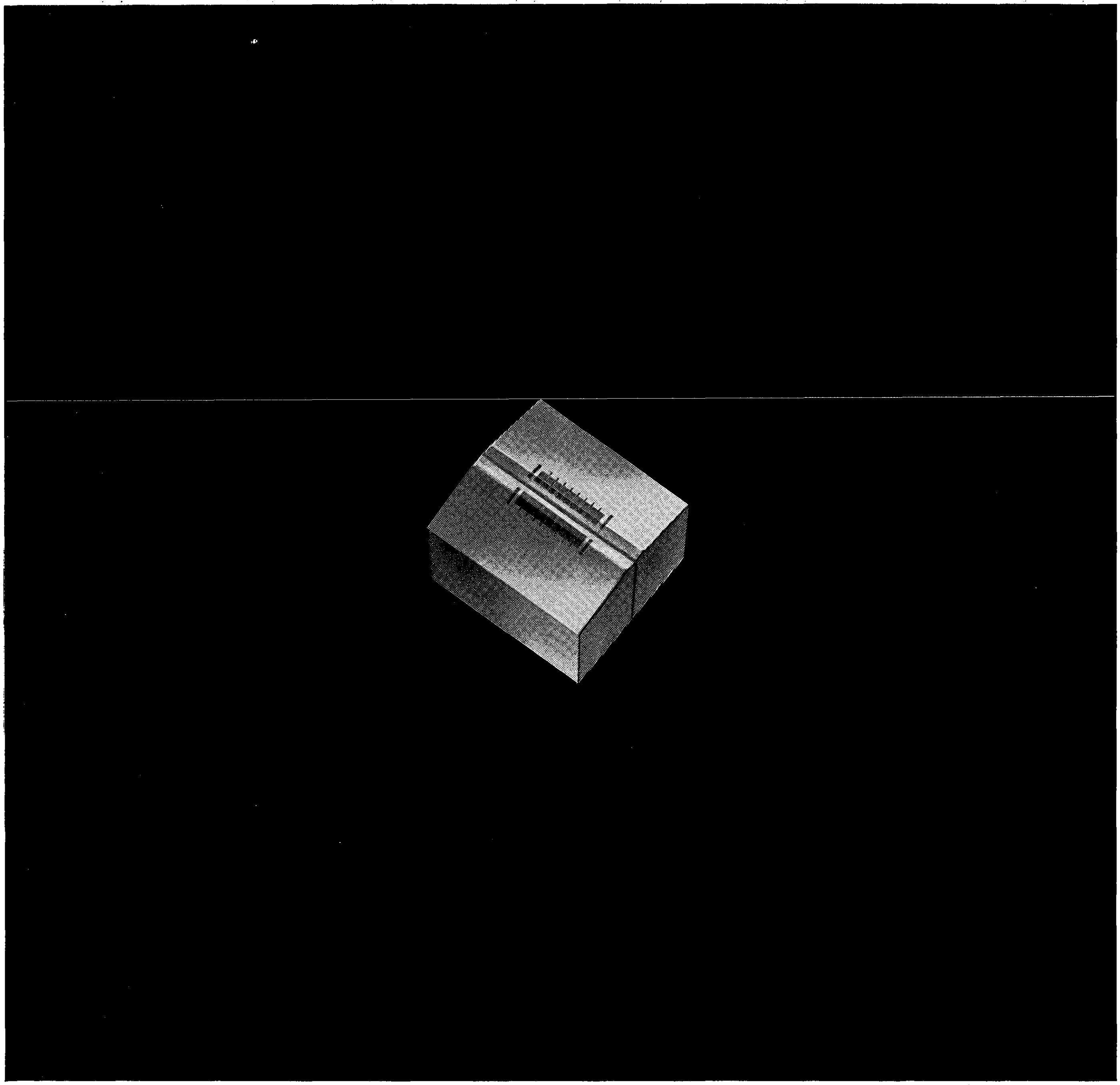
CIRCLE 7 ON READER CARD

DATAMATION®

september
1966
volume 12 number 9

automatic
information
processing
for business
industry & science

datamation departments



Here's all that touches the tape oxide surface

Potter Instrument Company's new single-capstan tape transports have a tape path in which tape oxide surfaces touch nothing but the read/write head . . . and touch not even that during high speed rewind.

Besides this, the new Potter transports have a guidance system precision matched to industry standards which guarantees 100% tape interchangeability with the most widely used transports now in operation. Here are some details:

Potter now has two all-new single-capstan transports, the SC-1080 for speeds to 150 ips; and the SC-1060 for speeds to 120 ips. Both units have tape paths that eliminate the usual rollers, air bearings and other hardware that endanger data reliability and reduce tape life. On a model-by-model comparison with transports of other manufacture, Potter's have

POTTER
® INSTRUMENT COMPANY, INC.

in Potter's new single-capstan transports

the simplest tape path and consequently provide minimum tape wear. In the high-speed model, the head retracts except for actual read/write operation.

Because of design simplicity, the transports are the industry's lowest in cost while providing the highest performance. Their packing density is 800 bpi NRZ 1-1600 bpi phase modulated recording. They rewind in less than 80 seconds for a full 2400 ft. reel. Both units feature solid state drive electronics and full test-points for easy maintenance.

For full details on the revolutionary design of these new units, and their fully compatible performance characteristics, write, phone, wire, or fill in the coupon.

PI-108

Send me full details on your single-capstan transports

Name________________

Company________________

Address________________

City__________ State__________

151 Sunnyside Blvd. • Plainview, N.Y. 11803 • (516) 681-3200 • TWX (510) 221-1852 • Cable-PICO
In Europe: McGraw-Hill House, Shoppenhangers Road, Maidenhead,
Berkshire, England • Maidenhead 20361

CIRCLE 9 ON READER CARD

Burroughs sets the pace for the computer industry

The new B 2500 and B 3500 are the latest Burroughs 500 Systems to be developed for business, scientific, and data communication tasks. In every measure of hardware and software performance, they far outpace other computer systems in the low- to medium-price range.

Two major factors are responsible for the exceptionally high performance-to-price ratio of the B 2500 and B 3500. The first is a design principle common to all Burroughs 500 Systems. In 1960, Burroughs Corporation determined that, in the future, computer performance would depend as much on software as on hardware. Events have proved this to be true. For this reason, *every* Burroughs 500 System has been *designed from the beginning* by *teams* of engineers and software experts.

Many economies result for the user. For example, the Master Control Program for the B 2500 and B 3500 not only performs many more useful functions than other automatic operating systems, but also reduces by a factor of 10 the amount of main memory that must be set aside for its exclusive use. Other gains are made in compiling times, programing ease, and speed and efficiency of operation. In short, the teamwork approach to computer design has allowed Burroughs to build a better bridge of communication between the B 2500 and B 3500 and their human users.

A second major factor is the use of monolithic integrated circuitry in construction of virtually all logic and the two control memories. The Burroughs B 2500 and B 3500 make use of complementary transistor logic, plus some use of array monolithics—two proven design concepts at the forefront of this newest logic technology. The results are smaller, faster, more reliable circuits at lower costs—and operating speeds measured in billionths of a second.

Like the other Burroughs 500 Systems—the larger B 5500 and the very large B 8500—the two newest systems can handle a variety of input/output activities simultaneously—as many as 20 at a time with the B 3500—while the processor continues its work. They multiprocess many unrelated jobs at one time, keeping the whole system fully utilized and greatly speeding the turn-around time for jobs. *And, since no human being could manage and schedule their multiple split-second operations, they are self-managing through their control programs.* All this, in the low- to medium-price range.

No wonder Burroughs is regarded as the pacesetter for the computer industry.

Burroughs Corporation

Detroit, Michigan 48232

CIRCLE 10 ON READER CARD

calendar

DATE	TITLE	LOCATION	SPONSORS
Sept. 27-29	Time-Sharing $195.	Hotel Royal Monceau Paris, France	Computer Usage Education, Inc.
Sept. 29-30	Users' Conference	Netherland Hilton Hotel Cincinnati, Ohio	Honeywell 200 Users' Group
Oct. 3	Convention: Information Handling in Behavorial Sciences	Miramar Hotel Santa Monica, Calif.	American Documentation Institute
Oct. 3-5	National Electronics Conference	McCormick Place Chicago, Ill.	
Oct. 3-6	On-Line Computing Methodology. $225.	New York City Essex House	Informatics Institutes
Oct. 10-11	Workshop on Multiprogramming	Host Farm Lancaster, Pa.	I.E.E.E.
Oct. 13-14	Users' Conference	Sheraton Hotel Dallas, Texas	VIM (CDC 6000) Users' Group
Oct. 17-19	National Meeting on Operations Research	Jack Tar Hotel Durham, N.C.	Operations Research Society of America
Oct. 17-21	Exposition and Conference	McCormick Place Chicago, Ill.	Business Equipment Manufacurers Assn.
Oct. 18-20	Symposium on Information Display	Boston, Mass.	Society for Information Display
Oct. 19-21	Users' Conference.	Prom Town House Omaha, Nebraska	CUBE Burroughs
Oct. 22	Intercollegiate Student Computer Conference	Pace College Westchester Pleasantville, N.Y.	
Oct. 23-28	EDP Audit and Controls. $250.	Governor's House Hotel, Baltimore/ Washington, D.C.	Automation Training Ctr.
Oct. 24-26	Computer Workshop	Lafayette, Ind.	Purdue University
Oct. 24-28	Project Management with CPM and PERT $150.	Georgia Tech Atlanta, Ga.	School of Industrial Engineering
Oct. 25-28	Fall Conference & Exposition	Biltmore Hotel Los Angeles, Calif.	DPMA
Oct. 27-28	ECHO Meeting	American Hospital Assn., Chicago, Ill.	Electronic Computing Hospital Oriented
Oct. 31-Nov.	Users' Meeting	Vacation Village Hotel West Mission Bay San Diego, Calif.	UAIDE Automatic Information Display Equipment
Nov. 8-10	Fall Joint Computer Conference	Civic Center San Francisco, Calif.	AFIPS

3M TASK FORCE Reports: Microfilm developments in Electron Beam Recording technology.

Can a printer doodle a bar chart while waiting for another character to be transmitted from "slow" internal computer memory?

Such a prospect seems unreasonable unless we put aside the mechanical monster concept we associate with EDP output. Bulky paper feed mechanisms, whirling drums and chains, and inflexible forged metal type are increasingly awkward companions of flexible and speedy electronic data processors.

A more appropriate partner will soon be available through EBR technology. EBR, an acronym for direct Electron Beam Recording, makes it possible to print with electron beams which can be electronically deflected at data rates compatible with magnetic tape recording. Printing can be in a variety of character sizes and styles, with computer-drawn pictures, or with graphics inserted from an alternate image source.

Special electron sensitive film materials accept information at megacycle rates with the instantaneous image processing necessary to this system concept.

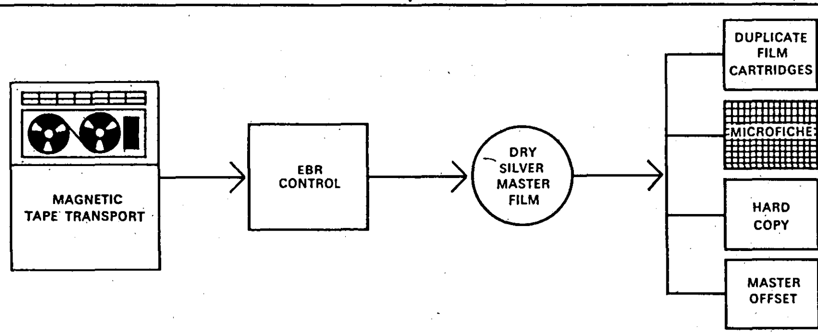
Appropriate logic provides for the organization and formating of the alphanumeric and pictorial data presented to the EBR printer from the computer.

It is already possible to anticipate this system because of advances in electron beam recording technology. The electron beam has long intrigued data processing designers as an input-output method because of its inherent speed and the ease with which it can be modulated, deflected and focused. Improvements in vacuum seal design at the 3M Company have enabled construction of an electron beam recorder in which the material is moved through a vacuum without affecting the recording properties of the material. Because of the high energy efficiency in direct recording, the recording material can be highly sensitive to electrons and yet instantly processed. This recording material can be used to create duplicate films, reproducible masters, or immediate hard copy on microfilm printers. Printing speeds of 60,000 characters per second and material cost of less than 1/10¢ per page open wide areas of possible application.

Progressive companies in such industries as banking, transportation, utilities, retailing, data processing services, publishing and manufacturing are planning significant savings in EDP output operations.

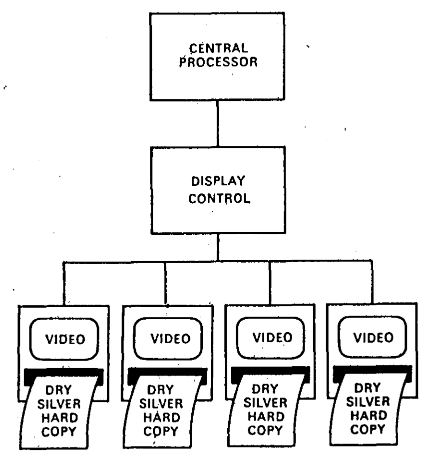
For more information and an EBR System Manual, write on your letterhead to:

3M Company
Attention: Rolf Westgard
(612-733-4995)
2501 Hudson Road
Dept. FDJ-96
St. Paul, Minn.



CRT displays with hard copy

On-line video terminals for the display of computer stored data are finding increasing acceptance. A limitation on this acceptance has been the difficulty of recording the display in permanent form. The mechanical alternative is between the slow character-at-a-time typewriter and costly faster printers. The relatively low-light level on the tube face made direct image hard copy difficult to achieve. Dry Silver emulsions can provide a dry print of the displayed image in a few seconds. Cost per print is less than on most office copiers. For information on how this could be adapted to your on-line displays, drop us a line.



Looking for a job with a challenge?

Expanding programs in electronic data processing technology are creating openings for engineers, systems analysts and marketing personnel. The kind of a job you may be looking for.

Please submit inquiries to:
Employment Department
3M Company —
Information Systems
2501 Hudson Road
St. Paul, Minnesota 55119

3M COMPANY

CIRCLE 12 ON READER CARD

letters

drum storage specs

Sir:

An additional magnetic drum storage characteristic that could have been described in John S. Craver's fine article ("A Review of Electromechanical Mass Storage," July, p. 22) involved the selection of an interlace in recording addresses around a drum's surface.

A multiple I/O channel system usually has some priority scheme built in for the servicing of I/O requests. If a word is presented by a drum subsystem and is not acknowledged (not "swallowed" by the computer because its time is being utilized for a higher priority task), then an additional revolution will occur between transfer of successive words of a block transfer to or from drum.

Interlacing successively addressable words (e.g., on a 100-word/track drum, 0, 50, 1, 51, etc.) increases the word availability time and may tend to alleviate serious systems problems.

EUGENE F. KLAUSMAN
Univac Division
Sperry Rand Corporation
New York, N.Y.

end of rumor

Sir:

In the June World Report (p. 109), you say: "Strong rumors are of a deal between IBM and the Polish government for 100 IBM 1401's, 10's and 40's at half original list price."

This information is totally incorrect, and based on rumors that are at least three months old. We do not have any agreement with the Polish government for 100 IBM 1401's, nor do we have any plans for reducing prices of our 1400 series.

R. J. CURRIE
IBM World Trade Corp.
New York, N.Y.

computers in education

Sir:

A friend of mine has a son, age 14, who is both bright and rambunctious. The boy has access to the terminal of a time-sharing system. One day he was talking in class and his teacher became annoyed. He was given the assignment of writing 1,000 times: "I will not talk in class." Needless to say, the time-sharing terminal proved to be quite useful in fulfilling this assignment. This seems to me to be a fine example of how computers could aid in reducing some of the drudgery of school work.

S. BOILEN
Watertown, Massachusetts

tribute

Sir:

The thanks of the data processing profession go to Daniel J. McCracken for the contributions he has made through his effective leadership. We can only wish that he is able to accomplish as much for others, serving in the ministry, as he did for us.

GEORGE W. WARNER
North Bellmore, New York

computers & credit information

Sir:

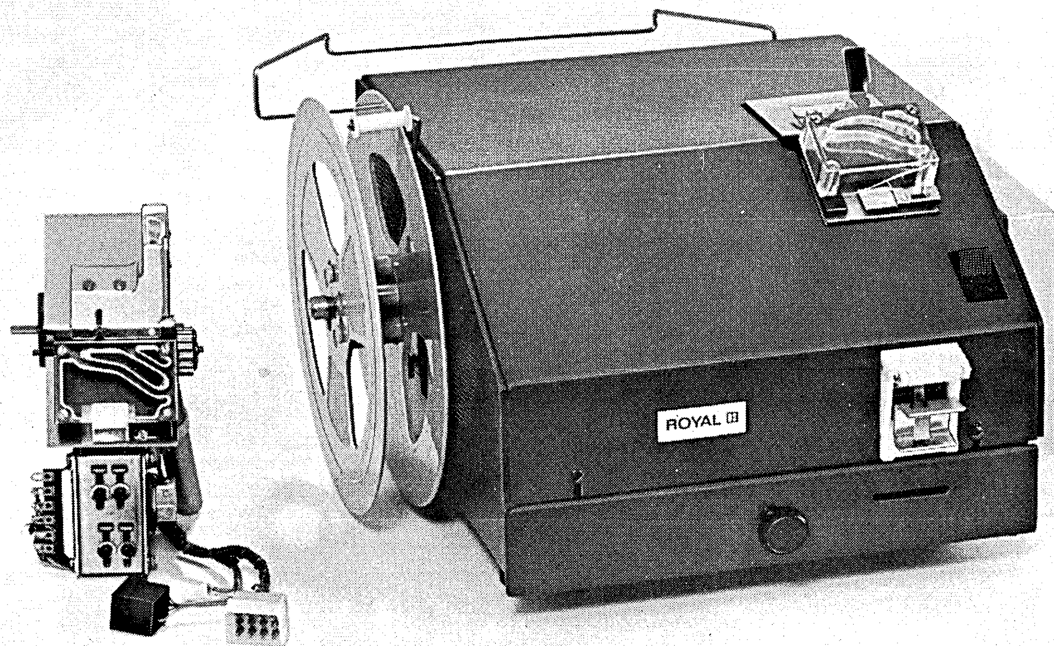
I was interested in reading the statements about "the war for control of credit information" (July, p. 19). Although it is perhaps a small point, it disturbs me a little that an article of this type gives more coverage to CIC, CDC, etc., than to credit bureaus.

Associated with credit bureaus for only one year, I am impressed by one fact in this "war for control" that we in data processing are sometimes prone to forget, or at least ignore. There is going to be a great deal more involved in automating credit information into a nationwide system than merely installing real-time computers. Credit reporting carries many ramifications and many more services for credit granters than might meet the eye of those not experienced in credit reporting but who see it as a good application for computers.

I feel this experience and relationship with credit granters will be the determining factor in favor of credit bureaus in this "war for control."

CLARKE NEWLIN, JR.
Asscoiated Credit Bureaus of America, Inc.
Houston, Texas

How do you like your Roytron™ paper tape punches,



plain or fancy?

Desk or rack mounted? With or without electronic logic and circuitry? Reader incorporated? In any of 24 models? 20, 50 or 75 CPS? Roytron gives you the kind of punch you need because it customizes for every conceivable OEM requirement.

Roytron also gives you something extra in operating features. Like efficient electromagnets and off-center springs for smooth, trouble-free punching. And Roytron tape punches operate on unusually low power (24 volts, 75 ma).

Now the Roytron™ line of tape punching equipment is even more flexible. The new, low cost Series 200 punch is ideal where an asynchronous speed of 20 CPS is best-suited for the job. For other applications, the Series 500 (50 CPS) and Series 700 (75 CPS) are available.

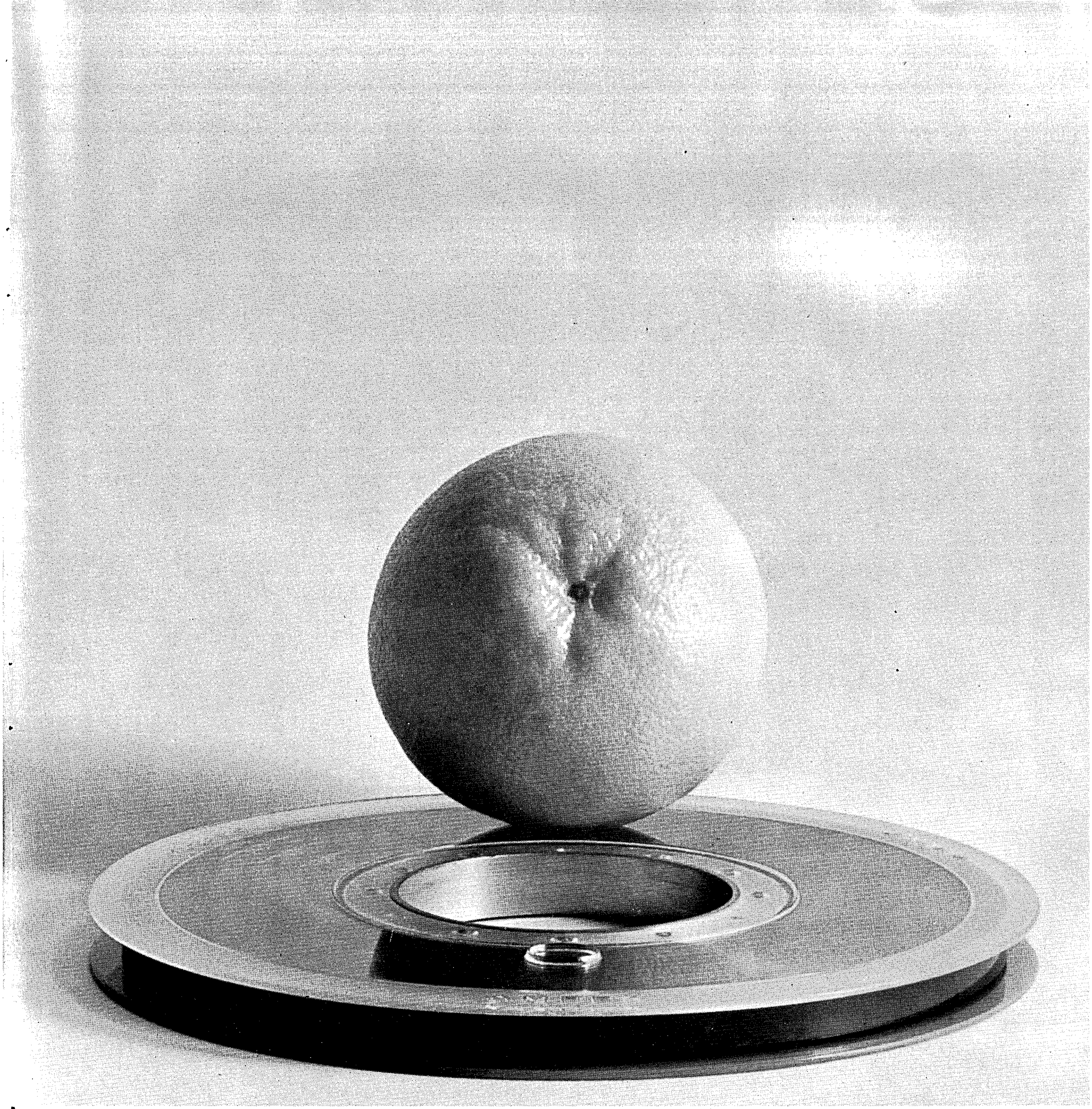
Roytron punched tape equipment comes with a meaningful warranty backed up by a nationwide system of service centers.

Consider the Roytron™ paper tape punch and other Roytron paper tape equipment before you go a step further on any punched tape system. For detailed specifications on the Roytron tape punch that fits your exact requirements, write:

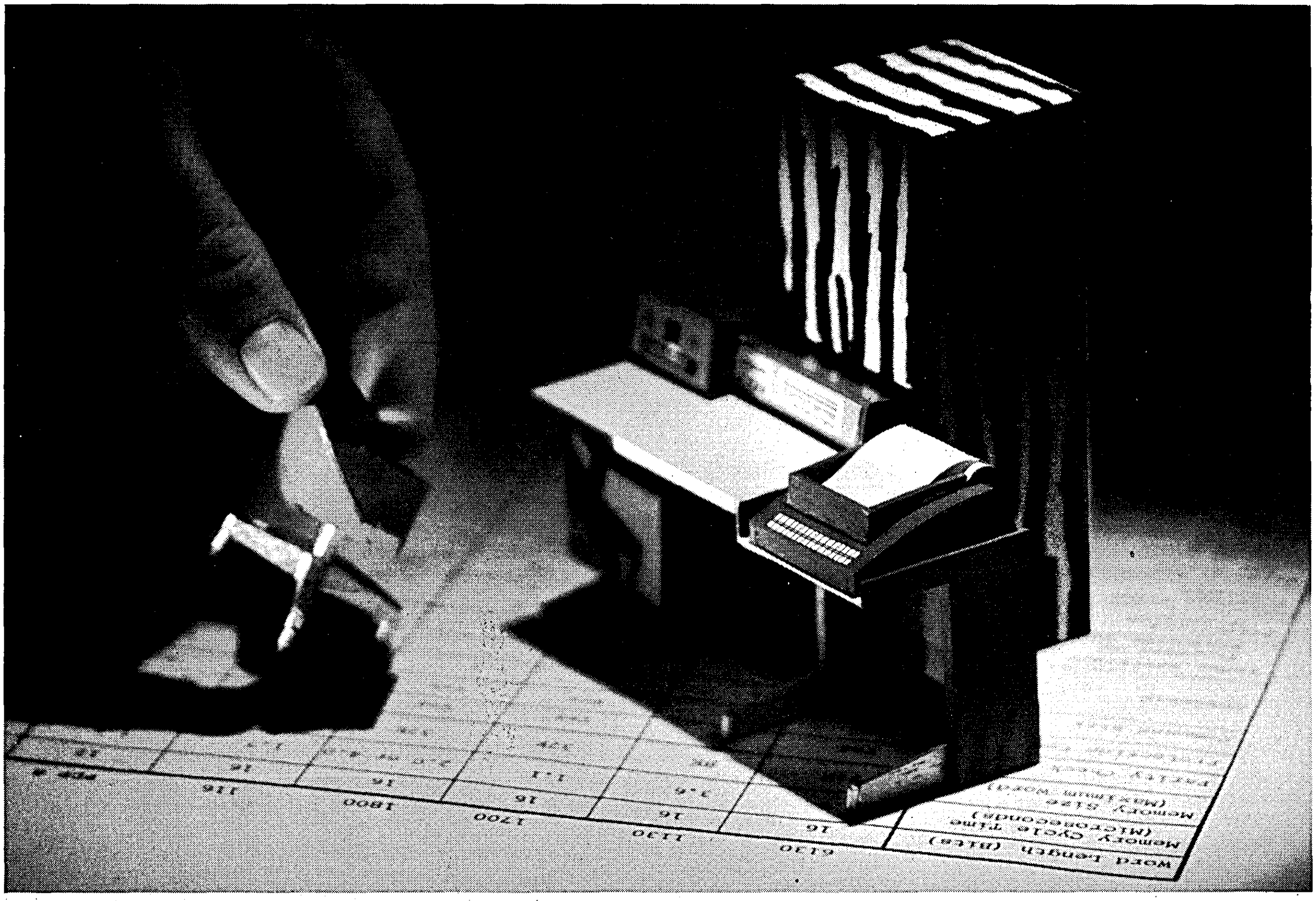
Roytron, Royal Typewriter Company, Inc., Dept. 33-IV, 150 New Park Avenue, Hartford, Connecticut.

A DIVISION OF LITTON INDUSTRIES **ROYTRON**

CIRCLE 14 ON READER CARD



ASI invites you to compare its new ADVANCE 6130



...with the other "compacts"—1700, 1800, 1130, 116.

Not really a "paper tiger" any longer, the 6130 is less paper and more hardware. So if these other machine numbers are familiar to you and you aren't expecting delivery of any one of them in the next few months, hold everything and investigate the 6130. The 6130 has been designed to offer more capability at the same cost than any of the other machines listed. In fact, we've got a comparison chart proving just that.

To receive this comparison plus some additional 6130 data in a plain brown and orange striped wrapper, contact any of the following:

BOSTON, One Garfield Circle, Burlington, 272-5400; WASHINGTON, D.C., 5012 College Ave., College Park, Maryland, 864-6340; DALLAS, Suite 1329, Frito-Lay Tower, 357-4086; HOUSTON, 1900 Southwest Tower, 225-1811; HUNTSVILLE, ALA., Suite 40, 3322 S. Memorial Parkway, 881-4822; NEW YORK, 475 Fifth Avenue, 679-5954; SUNNYVALE, CALIF., Suite 530, 505 West Olive St., 245-3694; LOS ANGELES 15551 Cabrito Rd., Van Nuys, 782-7030; CHICAGO, Suite 208, 119 E. Ogden Avenue, Hinsdale, Ill., 323-0450. ASI Computer Division, Electro-Mechanical Research, Inc.

***ASI COMPUTER DIVISION* EMR**

8001 Bloomington Freeway, Minneapolis, Minn. 55420

CIRCLE 16 ON READER CARD

look ahead

TURNAROUND TIME
AT THE SHARE MEETING

Although buttons reading "PL/I R.I.P." were passed out at the SHARE conference in Toronto, the feeling by the end of the meetings was that IBM intended to come through with full backing.

Initially, the IBM formal reports sounded to members like evidence of fading support and reaction from the audience was noisy -- if not threatening. Later, IBM representatives made clear their firm intentions to follow through on compilers for the full language. One attendee was asked what the time span was between these two different views. "Long enough," he said, "to make a phone call to Frank Cary."

Another interesting problem set discussed: which 360 owners can belong and what about those users who are time-sharers? Answer: users of Model 50 or larger automatically qualify for SHARE membership and those who use "a significant amount of time" on shared machines can petition for membership--will be considered as individual cases.

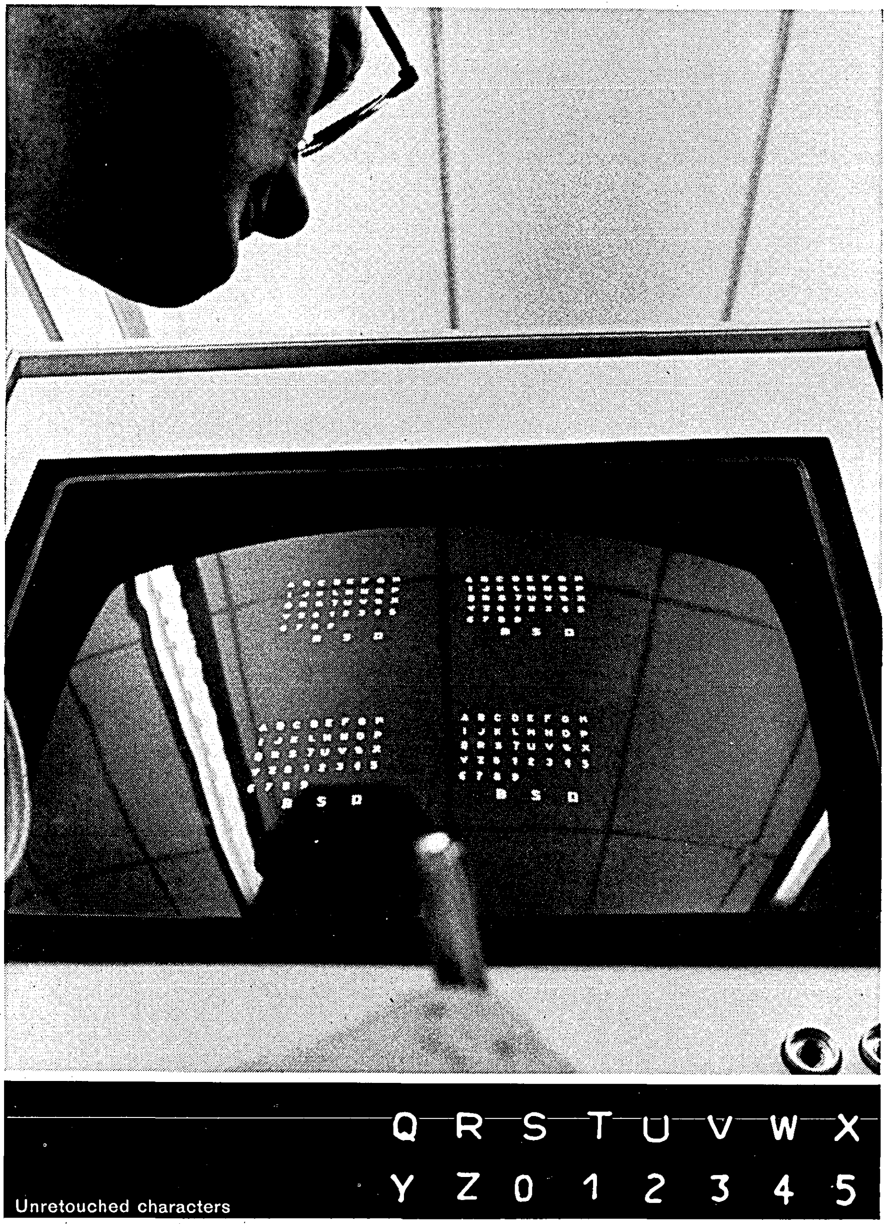
A session was held on the "data base problem" (subject of the February and March Datamation editorials) and a group set up to work on the project.

EAI MOVES FURTHER
INTO DIGITAL FIELD

EAI continues invasion of the digital field with another machine. Moving after the hybrid/simulation and special systems markets, the analog leader next month announces a second, smaller digital entry -- the 16-bit 640. "Several" orders are already in for the system, which starts at under $30K and is most competitive with the PDP-9 and Sigma 2. It has core capacity of 4-32K words, a 1.65 usec cycle time, and can handle up to 64 1/O devices. Software includes ASA Fortran, hybrid programs, math library, assembler, loader, and other programs. Delivery is 10 months, the first installation to be made in Feb. 1967. Further strength in the digital field is shown by the company completing installation of about 100 PSD-1020's, a product added when EAI acquired Pacific Data Systems.

BIT'S LITTLE BIT

Business Information Technology of Natick, Mass., joins the pellmell push on the OEM market with a new 10-bit-word machine, the 480. The processor will offer an 8-usec memory, handle both binary and decimal. The basic byte-oriented machine will include one I/O channel, indirect addressing to any level, TTY console and 30 instructions. Decimal arithmetic, more I/O channels and peripherals available for more $. Basic system cpu's will sell for $9K. Software includes assembler, RPG and arithmetic and utility routines. Deliveries, now quoted as 75 days for a



Unretouched characters

If you want unlimited symbol style written at up to 4 μsec speed...

A Tasker 401 Symbol Writer is for you. Its unique, patented stroke-writing technique can produce virtually any kind of symbol or character at speeds as fast as 4 μsec per character! It can also be remotely located (up to 50 feet) and can drive as many as 50 nonparallel displays.

Dynamic, automatic control allows precision shapes and sizes to be freely intermixed, as your programs or control circuits command. You can mingle capitals, lower case letters, subscripts, italics, symbols, foreign alphabets . . . even specify them tall and narrow or short and wide. And constant writing rate gives uniform line intensity and brightness throughout all symbols.

The basic 401 series writes up to 64 symbols (modularly expandable and alterable), including special symbols and gothic alphanumerics conforming to MIL-C-18012A. Applicable to any CRT display system, models are available for 4 μsec or slower speeds. A 2-μsec model is under development.

For symbol writing far in advance of the conventional, get in touch with Tasker.

look to Tasker

Tasker Instruments Corp. / 7838 Orion Ave. / Van Nuys, Calif. 91409 / (213) 781-3150

CIRCLE 17 ON READER CARD

Moving?

If you're going to have a new business address, please fill out the form at the right now and return it to us. We'll make sure that DATAMATION keeps up with you. To save time, we need your old address, just as it's printed on the mailing label, (or the whole label is better yet) as well as your new address.

define your terms, please

Whatever your work in information processing, you'll communicate better with the help of this authoritative glossary.

Name
Title
Comp
Addre
City
NEW
Name
Title
Comp
Addre
City
What i
Your S

an in
DA
This 6
of tech
essing.
tronics
This gl
as a r
familic
as a c
respon
Please

D
141 I
Pleas
Nam
Addr
City
Pleas
State

look ahead

(Continued from page 17)

basic system, will hopefully come down to 60. And a faster, bigger BIT machine is on the way.

A STEP TOWARDS THE QUICKSAND

Univac, which has been doing well with about the only working large-scale software, joins the manana crowd with a new operating system for the 1108. Called Exec 8, it will feature real-time, conversational and batch subsets, is scheduled for 2nd quarter '67 delivery. Basic language is something called Fortran V, which is IV plus capabilities for handling program files and "bit fiddling" -- addressing by bits. The F-V batch compiler takes up 20K lines of print.

TYMSHARE DEVELOPS BARGAIN ACOUSTIC COUPLER

Tymshare, Inc., west coast time-sharing service about to go on the air commercially, has also developed a low-cost acoustic coupler; it should sell for about $300. They also figure they can package a KSR-33 teletype (minus the legs) for $452 and an adapter at $64. Put them together and you have a portable terminal for less than $900, including tax, requiring only an electrical outlet and a telephone. Lift the phone, dial Tymshare (or any other t-s system), put the phone receiver in the coupler, and you're on. A prototype has been tested, with the user dialing into the Dartmouth system from California successfully.

RUMORS AND RAW RANDOM DATA

A gun-shaped camera will be the first Aeroflex Labs product to demonstrate their new Multiplex Recording Technique (multiple images on a negative, same size as negative). To be in prototype by December, the small camera system will take 50 pictures/negative, no flashbulbs needed, and has facilities for developing, projecting, and printing. The company has received more than 2000 inquiries on MRT from computer, document retrieval, camera, and other industries... Data Disc, Inc., of Palo Alto, has developed a single silicon assembly with 16 read/write heads, complete with diodes. The hybrid i.c. unit reportedly could lower head cost to about $1/track versus the current $25/track...Rumors are that IBM recently increased its monthly sales quota 40%...Sign of the times: hardware-oriented IEEE has hired a software consultant to help improve its computer installation...IBM has bought time on the NYU CDC 6600, sent men with stop watches to use it in the wee hours...Adams Associates runs radio and newspaper ads offering a $200 reward for information leading to "hiring of computer analysts and programmers." And a NY software firm, evidently seeking bearded help, advertises in The Village Voice ...Jerome Weinberg, formerly head of MIS at CAI, is setting up a NYC consulting firm, MANDATE...The American Standard Vocabulary for Information Processing has been published. One recipient reports that he wrote down ten words and phrases -- before opening the glossary -- that he considered of current importance: virtual memory, information utility, time-sharing, byte, management information system, operational system, command-and-control system, decision table, integrated circuit, control-and-forward. Only three of them were listed, suggesting that the jargon changes too fast for glossary makers to keep up...A government official notes that "time-sharing is like wife-sharing -- you can't really make it work unless you have abnormal tastes."

OVER THERE

To an American, Europe is a fantastic maze of international, intranational, linguistic and cultural barriers, composed of more or less equal parts of mistrusts, fears and hatreds, some of them centuries old. Rigid conventions, bureaucratic red tape, complex procedures and poor communications compound the problems of doing business in Europe. And the implications of all this for information processing are rather profound.

At work, for instance, are such institutions as a professional caste system, in which Ph.D.'s from certain universities receive almost automatic acclaim. A company may hire a university professor as a consultant, refusing to listen to the man who runs the computer organization. A general conservatism and resistance to new ideas slows progress. Full employment and the lack of competent personnel across the board is not seen as an opportunity to take advantage of data processing. Employers can't lay people off, and they won't listen to the possibilities of attrition and reassignment. A penchant for family and personal ties makes it difficult — for Americans especially — to peddle new ideas and approaches.

Europeans in general take a much more casual, leisurely approach to business than do Americans. The long lunch, an American institution, is equally popular in England. "But," says one U. S. systems man who worked there, "Americans don't spend all morning setting the luncheon up." Appointments are missed without notification. One American was asked to wait while the man he had an appointment with ate his dinner.

Communication within firms is far less formal and rigorous than in the U. S. In one case, the head of a medium-sized company ordered a small-scale computer without consulting his tab shop supervisor. At one large company, different departments have farmed out highly overlapping edp jobs.

There is a chronic and widespread shortage of experienced information processing people, but the universities — bound to upper class classical education traditions — have failed to produce enough of the right kind of people to meet demands. As a result, there is a good supply of high-level, theoretical expertise . . . a shortage of "doers." The emphasis on ALGOL in a FORTRAN world is only one example of the excellence-at-the-expense-of-reality attitude in European universities.

Firms seeking to expand find it takes four months or more to bring a new man in after he's accepted the job . . . he must often give three months' notice. Hiring Americans can be an expensive and not always successful attempt to solve the problem. That's partly because of financial reasons: Americans expect a 20% salary increase to work overseas; then there's the expense of relocating their families. Too, Americans — used to refrigerators, supermarkets and similar luxuries — don't always see the charm of working in a strange land where housing is short and prices high.

There are, of course, technical barriers to progress as well. The poor quality, low transfer rate and high cost of communications is one. Lack of government-supported research has handicapped technical development.

Then there's plain old provincial nationalism. A Frenchman approached an American after a technical briefing of defense people and apologized in perfect English for having asked his questions in French. "We are required to do so," he said. Some countries fail to support the International Computation Centre because its director is a Frenchman. National pride, economic barriers, and indifferent marketing have prevented any one European nation from establishing a healthy computer industry.

Nearly every American seer has his own neat quantitative estimate of how far behind the American scene European information processing lags. Some say five years. One French expert says that his country's software is five to 10 years behind. A temporarily transplanted American points to the absence of the wall-mounted pencil sharpener, the fact that it takes $1000 and a week to get a phone installed in downtown Paris and snaps, "They're 20 years behind us."

Whatever the quantitative measurement, the gap is there. And the social, political and economic barriers which partition Europe — while they handicap American endeavors to develop the edp market—are much more destructive to Europe's attempts to establish its own information processing industry and to take maximum advantage of one of today's most powerful tools.

EUROPEAN EDP

the view from england

by CHARLES WHITE

The total European market for computers and peripherals (covering data processing, scientific, industrial processing, education and analogue work) is expected to top $1 billion for the first time by the end of 1966. Of the three major users in terms of number of installations, West Germany tops the league and has done so consistently throughout this decade. A few months ago it hit the 2000 mark. The U.K. and France follow with about 1600 and 1200 machine installations, respectively. At the present rate of expansion, the combined Western countries should comfortably exceed a target of 12,000 machines by 1970.

An examination of the utilisation of computers indicates that some of the less populated countries, such as Switzerland, have both a far better installation rate per head of population and a proportionately higher number of large-scale systems in operation. These last facts give some indication that Europe offers neither a unified market nor an easy place for an American company to find the domestic front. A mistaken excursion across the water for a quick profit may be a good way to lose a shirt. In fact, only three corporations, IBM, Univac and NCR, have (to my knowledge) made significant contributions to profitability in the past from their international computer operations. Several later entrants to the market appear to be in a healthy position, and RCA had an extremely good run with a policy of non-direct intervention by licensing its 301 to a French and British manufacturer.

There are many factors influencing, and often hindering, developments. The most obvious is the activity of the local manufacturers, who in the U.K., France and Germany are receiving a stimulus from the relevant government departments in the form of cash grants and a certain amount of "buy home-produced products" pressure on government purchasing agencies.

User groups do not progress at the same rate in every country. Introduction of computers into banking and insurance or the steel and chemical sectors of the economy may be rapid in one nation and slow in the next. This type of situation tends to aggravate software progress, to produce a largely applications-oriented environment and relatively expensive way of life for the user. In general, the rate of progress made by an industry depends on the encouragement it gets from government as a vital contributor to the economy.

This accounts for progress in steel and machine tools in the essentially manufacturing British way of life; in automation in mining, heavy industry processing and military fields in Sweden, which is very dependent upon natural iron resources for income and is defence independent; and in heavy computer adoption in banks and finance houses of the monetary-minded Swiss. The approach in the Netherlands has been different again. There, commercial users have been great pioneers of computer cooperatives both on an application basis, such as for payroll and invoicing, and industrial basis, such as banking.

the leading users

The two user groups that have developed in a consistent fashion are the airlines and nuclear research. The reason for the airline trend is almost certainly due to the similarity of business of all international carriers, which has produced a fair set of international standards for procedures and documentation of major activities. Most of the frills of each installation are based on local demands. An example is recording a passenger's name and address throughout all transactions instead of adopting a reference number for most purposes. Novel developments have been in tasks such as computer calculation for weight and balance—where Scandinavian Airlines (SAS) scored a first.

Some of the reported reservation systems are as follows:

British European Airways—twin Univac 490's started operation earlier this year.

British Overseas Airways Corp.—a Standard Telephone & Cables simple enquiry system being replaced by twin 360/65's for a complete automatic reservation, weight and balance, and management information system.

Scandinavian Airlines Systems—two IBM 1410's plus special communications interface installed in 1964 and to be upgraded with twin Univac 492's. Scandinavian also handles message switching of communications traffic via a Univac 418.

Air France—an early Teleregister seat availability scheme to be replaced by Univac 2108's for a total company operation.

A journalist and well-traveled observer of the development of computing in England and on the Continent, Mr. White is Datamation's European Editor.

Deutsche Lufthansa—twin Siemens 3003 real-time processors installed last year.

The nuclear physics and atomic energy establishments are prime users of large-scale machinery. Stretch (IBM 7030), as well as 7090's, are used in both Britain and France; the international centre, CERN, at Geneva has a CDC 6600 as the outstanding big system in a mixed bag of installations; the Univ. of Paris has reduction of data from bubble chamber experiments as a major application for its Univac 1107; and an Atlas 2 and large English Electric KDF 9's have been installed by the United Kingdom Atomic Energy Authority. There are a host of smaller machines in operation for the nuclear physics groups. Britain alone has six Digital Equipment PDP-8's for local laboratory experiments, and has virtually put DEC in business in Europe, despite the fact that the company is the new boy in the U.K. The airlines and nuclear physics fraternities are more the exception than the rule as examples of big information processing users in every country.

There is no doubt about the heightening ambitions of Europe. Unfortunately, each nation has allocated different priorities, hence producing a fragmentation of applications investment. A look at two of the smaller countries, Norway and Denmark, shows both great aspirations and the obvious difficulties of working on limited budgets. These problems tend to be multiplied for the bigger industrial nations since their economic growth is proportionately little different. Even in Germany, industrial growth is levelling off, and the country is moving out of an abnormally easy economic environment (for Europe) into a more average one.

developments in scandinavia

In Norway there is no significant production or development of instruments, controllers or computers. The exception has been special development in numerical control for shipbuilding equipment. The country has added a Univac 1107 and a CDC 3600 to its scientific computing community, but there are less than 100 systems installed. They are divided about 50/50 between science and commerce.

Major contributions to computer applications come from Oslo and Bergen Universities, the Technical Univ. at Trondheim and the Defence Research and Atomic Energy Institutes, but they are mainly in areas such as ship classification and research, meteorology, control theory and power network operations, and surveying and engineering construction calculation. Applications of dp to the mechanical engineering industries have been slower to develop and there are few experienced people to plan and implement stock control, production planning and documentation systems.

Two original items of software research coming from the Norwegian Computing Centre (NCC) at Oslo, equipped with an 1107, are a simulation language, Simula, and the NCC linear programming system. Norway's problem is clearly one of what she should do and what she should leave alone.

Neighbouring Denmark has a few more installations, about 120, and also supports a unique little industry. It is based on Regnecentralen, Copenhagen, a non-profit, independent institute which was formed 11 years ago by the Academy of Technical Sciences for developing the computer sciences—either on purchased or self-built machines. Opting for the latter choice, a machine called Dask was built. It was followed six years ago by the first Gier system. Computers and peripherals carrying the Gier label are now made by a production division of Regnecentralen; 32 main frames have been sold and 28 installed. The machine has a 42-bit word length, 6.6-microsecond core cycle time, 2nd 1,024 words backed up by 12K drums with a 14KC transfer speed.

Regnecentralen had their first Gier in operation during 1960 and pursued software development, particularly on ALGOL, and had an influence on European thinking rivalled only by the Germans and Dutch. Some of the first ALGOL compilers, 1961 and 1962, were running at the Copenhagen centre and an ALGOL 64 is near completion. Compilers have been written for other Continental European users and machine makers. Gradually, the centre has forced its way into the commercial dp field. A COBOL compiler has been written for a German manufacturer, Siemens, and a standard payroll program is run weekly on a Gier magnetic tape system for 70 Danish firms.

In practice Regnecentralen operates more like a university research and development centre with a very sensitive commercial antenna than like a manufacturing organisation. Many of its staff of 200 teach, conduct almost independent research studies, and participate directly in other business enterprises. The unit has added two CDC 1604's to its inventory.

One of the other biggest Danish users is a local government cooperative operating an IBM 7040 and 360/30. The majority of other installations are dp users of the 1401 and 1004 calibre. Copenhagen is also the home of the Scandinavian Airlines computer system, and Siemens has real-time customers for a telephone company, a car ferry reservation scheme and some banks. In fact, the computing scene is very different in emphasis from its neighbour, and, in this respect, typical of variations right across Europe.

The biggest computer manufacturing centres lie in the U.K., France, Germany, and to some extent, the Netherlands. The last country is the home of the giant Philips electrical group which has been regarded as the slumbering giant for three or four years. Periodically there are rumbles from this direction—it has produced message switching units, billing and invoicing machines, and some small stored program machines—but has yet to unleash itself onto the market proper. Nevertheless, Philips and its subsidiaries supply a big part of core stores (including faster than 0.5-microsecond designs) and components to other makers. A family of integrated circuit processors for process control have been designed by a special computer R&D centre at Appeldoorn, and their release is expected at almost any time now.

Software expertise could be Philips' problem in exploiting the market, and their 40% stake in the tiny firm Electrologica, based in The Hague, is believed to be a recognition of such a prospect. Electrologica is slightly reminiscent of Denmark's Regnecentralen. It carved out a slice of the scientific community in the Benelux countries, plus France and Germany, and is respected among the very clannish ALGOL groups of Europe.

the pace in sweden

There is a second sleeper of the electronics industry in the giant Saab of Sweden. It has a technical licensing agreement on computers with Honeywell, dominates its country's defence purchasing, and makes numerical controllers for machine tools. The oft quoted, "What's good for General Motors is good for the U.S." could be equally applied to Saab in Sweden, but corporate policy is to take things at their own pace.

Firms farther south, on the other hand, are at hammer and tongs vying for their place in the sun. None of the mainframe makers could be called 100% secure, but prospects look more favourable than two years ago.

In Britain ICT has notched over 500 customers in 18 months for its 1900 series, and returned into the black—

by a hair's breadth. Present efforts are directed to mass production of software on an unprecedented scale for a European manufacturer. Development ranges from Nicol (which is a conversion language for tab and 1004-type users to rewrite applications directly for the cheap 1901) to Simbol, a large machine simulation language. ALGOL, FORTRAN and COBOL are included among the repertoire—the only item missing is a definite implementation policy for PL/I. ICT's biggest decision for the future may be how or when to move from six-bit character manipulation on its 24-bit word machines to the eight-bit byte.

ICT's main home competitor, English Electric-Leo-Marconi, is nursing its integrated circuit System 4 series through for first deliveries next year. Bought-in technical know-how from RCA relieves some hardware backache and hopefully some software heartache. With a big share of the university market on KDF9's and a near $8-million worth of Leo 326's going into the Post Office, the problem of overlapping two generations of software is going to take skilled management.

The third of the entrenched British designers is Elliott Automation, who played their ace earlier in the year with the 900 series of microcircuit machines priced well under $40,000. Their aim is to drive a hole in the industrial control, education fields, and military systems business and to get production up 500 machines a year. Elliott's computer division services up to 100 management companies within the group, several of which are much larger than the computer side.

As a bonus, Elliott manufactures for NCR the 315 systems destined for outside North American destinations; and NCR has been on a winning streak over the past 12 months—particularly in their strong areas of banks and insurance. A block order of six has come from the Netherlands for 315's followed by similar size business from Germany, Spain, France, Belgium and Scandinavia. In return, National markets some Elliott equipment in the data processing field and shares appropriate software development across machines. NCR's compiler for small commercial machines, Language H, is available on both companies' data processors. An ALGOL for NCR 315's has come from the British office and has almost certainly been assisted by Elliott expertise in preparation, for the latter company is quite adept in this area.

Ferranti pipped Elliott Automation to the post by bringing out a micro-circuit range for military and process control a few months earlier. Much of their work is classified, but a real-time compiler has been written for fire and control defence schemes and radar tracking, and is expected to find its way into industrial applications.

anglo-american ties

Another Anglo-American tie is between Scientific Data Systems and British General Electric Co. The latter is making the Series 90 and Sigma 7 under license, and indicates that substantial orders are near completion after a quiet 12 months.

Still penetrating Europe, Honeywell is supplying its international needs for H200's from a factory in Scotland. After a flying start in Britain, the growth on the continent has somewhat slowed, although since bureaus were established in many capitals last year, the number of 1401's to have fallen to the H200 has steadily risen.

British interest is concentrated on the government schemes for the National Computer Centre now being built at Manchester, and the Ministry of Technology's endeavors to stimulate software development. The National centre will have a KDF9 big enough to take the Egdon operating system, a package devised by the FORTRAN-based Atomic Energy Authority for 32K store and magnetic-disc-backed installations. University machines are being upgraded to take Egdon, and the new centre should act as a clearing house for the bulk of the scientific and education computing community.

Frenchmen are casting similar glances to see if DeGaulle's *"force de frappe"* is to provide salvation for them. The Bull-General Electric combine is hanging on to its place with about 15% of the business, but prospects throughout Europe have been slow in placing expected orders for the 600 series.

GE appears to be concentrating on its old love of process control more than pushing the data processing potential gained from Company Machines Bull. The government-backed Citec, with powerful parents such as CSF and Compagnie Generale d'Electricite, is also more committed to the industrial user. Negotiations with the U.K. for an Anglo-French range of processors have broken down, and this may cause minister M. Alain Peyrefitte to push for a renaissance of the remnants of Company Machines Bull. He is reported as the architect of plans under inspection by the Elysee Palace, and to be eager for a completely independent industry. France is one of IBM's two biggest manufacturing units outside America, the other being Germany. At Essone, about 30 miles from Paris, manufacturing is concentrated on SLT modules, a reversal from 1400 series days, when it was the centre of 1410 and 7010 mainframe fabrication.

In West Germany, Siemens & Halske is in a similar position to Britain's English Electric. Banking on the success of RCA's Spectra 70, it is making it under license and quietly logging up orders for first deliveries soon. In the meantime, some real-time experience is being gained with installations of their own 3003 design for reservation-type duties. Telefunken is in about the same state in the scientific and engineering business. It has fewer commercial customers, although the firm has recently attacked the operations research-type applications with management planning aids such as Pert and LP packages.

At Siemens, they hope to extend their share of the near IBM-monopolised market from about 5% to about 15-20%. Components for the machines, carrying a 4004 series label to replace Spectra 70, will come mainly from the U.S. This is expected to be whittled down to about 25% imported when new plants are ready. Manufacturing investment in new facilities for the computers alone amounts to $10 million. Another $5 million will cover software and peripheral innovations for local needs.

common problem: manpower

There is one problem shared by all European countries in the throes of rapid computer expansion. It is the difficulty of obtaining and training manpower. By 1970 more than 200,000 extra people will probably be needed. Exact figures are hard to come by, but estimates made in Germany, France and Britain state their expected labour forces by this time as more than 20,000.

Some of the new people will come from re-training programmes, as was indicated in the British Ministry of Labour's Manpower Survey, published last December, which forecasts the growth of installations to 5,400 by 1970 accompanied by an elimination of 300,000 clerical jobs. The personnel difficulties are further confused by wide variations in salary policies. In some organisations, particularly American subsidiaries operations, project managers are paid in line with the better U.S. scales of between $15,000 to $30,000. Local European outfits paying more than $12,000 even for a top data processing manager consider themselves among the big spenders. ■

AUTOMATION IN EUROPE

kaleidoscopic

by W. K. de BRUIJN

Although the development of the use of computers in Europe has been fast during the last five years, it has not been as fast as in the United States. (Fig. 1). The data for this graph were taken partly from the report "Development of the Computer Market in Europe," published in 1963 by the Netherlands Information Processing Research Centre in Amsterdam and partly from the computer censuses from the ADP Newsletter. The broken line shows the growth of the European computer market that was expected in 1963. The real growth of the use of computers in Europe has been about 65% more than these 1963 estimates. For this there are several reasons.

The estimate published in 1963 was based on data up to the middle of 1962. After that time a new series of computers entered the market—new in the sense that they were smaller and cheaper. These computers opened a completely new market. Lots of enterprises that had considered themselves too small to be able to afford a computer of their own suddenly realized that they had to reconsider. Many of these firms now have a computer installed or on order.

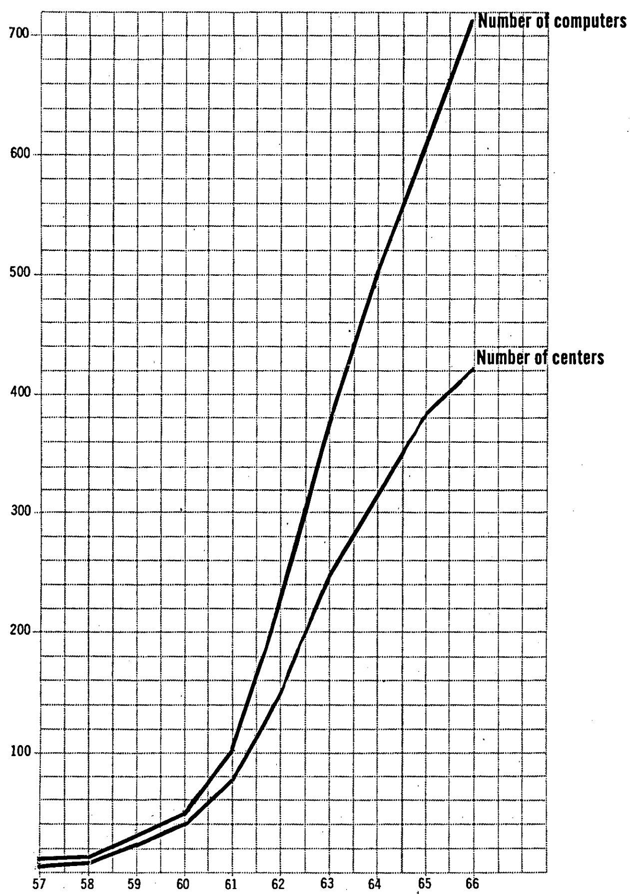
In some countries (France, for instance) the used computer market came into existence. This also led to computer use by small organisations. Notably, agricultural organisations entered the computer market in this way.

Some countries making a late start, such as Norway and Denmark, have progressed faster than was expected. New user groups have appeared. For instance, the introduction of new applications has resulted in newspapers and publishing firms entering the field; contractors and builders have found PERT techniques of extreme interest; and other groups have been stimulated by examples and have, as a result, computerized faster than was expected. An example of the latter is local authorities.

Senior member of the scientific staff of the Netherlands Information Processing Research Center, Mr. de Bruijn has written articles in both Dutch and English on international aspects of edp. Recent credits include "Fifteen Facets of Administrative Automation" (1964), "Use of Computers with Local Government in Western Europe and North America" (1965), and a report on the further development of the European computer market to be published this autumn.

The number of consultants in the automation field is growing fast. In many countries, auditors and business consultants realized that there was a dire shortage of specialists and have hastened to do something about it. Many consultants, who some years before were only interested in conventional techniques, have now set up special departments for automation problems. Also, American consulting firms have realized that there is a new market and have established branches in one or more European countries. (Incidentally, in many European countries associations of consultants prohibit members from advertising.)

Fig. 1 Computers installed in U.S. and Europe



The average computer centre is also growing. Fig. 2 shows the number of computers and the number of computer centres in banks and GIRO (a credit transfer organization using post office facilities) institutions in Europe projected to the end of 1966.

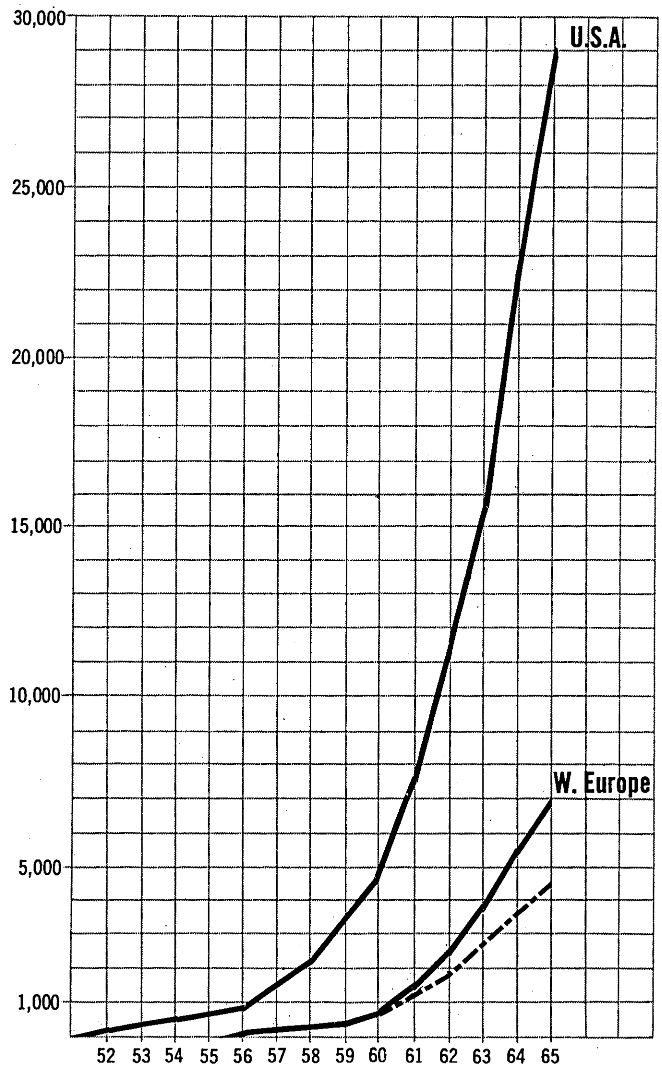
The data for this graph and several of those mentioned later in this article are taken from an updated report on the development of the European computer market, to be published by the Netherlands Information Processing Research Centre, Amsterdam.

Increase in the number of computers per centre is, in summary form:

1959	1.22	1963	1.57
1960	1.26	1964	1.58
1961	1.34	1965	1.59
1962	1.53	1966	1.70

It is, however, possible that as a result of the many small computers now entering the market this average will stop growing within the next few years as smaller banks will make a start with automation thereby increasing the number of centres with only one computer. However, these figures show only one side of this growth. Another and perhaps more important side cannot be shown because data is not available: the expansion of computer centres that do not add a second or third computer to their equipment. Many replace their old machines with newer, bigger ones. Others enlarge their capacity by adding extra peripheral equipment, memory modules, etc. A representative of a European manufacturer once stated that a considerable part of their turnover consisted of adding new equipment to their systems already installed. This hidden growth of computer use is not spectacular but is certainly very important.

Fig. 2 Computers and computer centers in banking



The 1963 report showed the following breakdown of the 2100 computers installed in Europe: 38.5% in industrial and commercial firms, 14.8% in universities and scientific institutions, 10% in service bureaus, 9.1% in insurance companies, 8.6% in banks and GIRO offices, 8.3% in local authorities, half of which were in public utilities, and 7.7% in government.

Corresponding data projected to the end of 1965 show that the percentage of some of these groups has fallen whereas others now have a bigger part of the total. This pertains to about 7,000 computers installed in Europe. In this category are universities and scientific institutions (10.7%), insurance companies (7.7%), service bureaus (8.9%) and government centres (4.9%). On the other hand, slight gains have been made by local authorities (9.3%) and banks (9.8%). The big winner is the industrial and commercial group which has installed over 3,000 computers—some 45% of the total. The following figures show a breakdown of this group by industry:

Food and related industries	5.7%
Metals	5.3%
Electrical equipment	4.9%
Cars, ships, and aircraft	4.6%
Chemicals	4.2%
Iron and steel	3.6%
Textiles	2.5%
Oil	1.8%

Because many firms are integrated, it is nearly impossible to differentiate between manufacturing and trading so that industry and commerce have been combined.

The 1963 report also showed the growth of computer use as measured by the ratio of machines installed per one million workers, exclusive of agriculture and fisheries. These figures were calculated to the end of 1965 below.

Country	1961	1965
Belgium	36	103
Germany	36	127
France	37	124
Italy	25	82
Netherlands	34	106
U.K.	26	88
Norway	27	114
Sweden	60	132
Denmark	22	106
Switzerland	68	182
Austria	18	67
Spain	6	27

These figures show that Switzerland is still leading and that Sweden is runner-up, but that the latter's place is being threatened by Germany and France. Remarkable, too, is the way Norway and Denmark have caught up with the other continental countries. These figures have to be interpreted according to "breakdown of the population to age classes" and "age at which children are allowed to start working." Still, this is the best standard of comparison that has been used so far.

On the sales side there have also been some important changes. Most important of all has been in increased effort of American manufacturers to penetrate the European market. Firms like General Electric, Honeywell, RCA, Control Data and Burroughs have in different ways started to enter this market. For example, GE by taking over Bull and Olivetti; RCA by joining forces with Siemens and, in a way, by their relationship with English-Electric; Honeywell and CDC by starting branch offices in various countries and by cooperating with Saab; and Burroughs by adding its computer line to its already extensive line of office machinery sold by its many branch offices throughout Europe.

Amalgamations have further reduced the number of manufacturers. Significant examples are ICT, EMI and Ferranti and that of English-Electric and Leo, both in England. The most recent case is that of Philips—a newcomer in the market—which took over Electrologica and has also acquired 40% of the German firm Siemag.

IBM, Univac, and NCR, the first to invade the European market, are still going strong. Small computers like the Bull Gamma 10, IBM 360/20, Univac 1004 and the Elliott 803 have been especially successful. These small computers account for probably over 70% of the computers sold.

Comparing the development of the European with the U.S. computer market has always been difficult for a number of reasons. Europe consists of a large number of markets, each with its own language, different outlook and habits and also a different set of manufacturers. Another difference is the degree of pressure applied by the various governments on buyers to choose products from the home industry.

Another difficulty in drawing comparisons is the difference in the buying capacity of the dollar in the U.S. and Europe. Roughly speaking, a dollar is worth twice as much in Europe as in the States. Because computer prices are generally based on dollars and are equal all over the world, a European customer has to make good about double the amount of expenses by savings—compared to an American customer. This, of course, is true only for the selling price or the rent of the computer, not for other costs like analysis, programming, site preparation, and so on.

Other differences are the relatively smaller number of large and relatively higher number of small firms. And last, but not least, Europe contrasted with the States invests relatively little in military and space research. This research causes a very important computer market in America whereas in Europe the number of computers in use for this type of work is nearly negligible.

The most important factor, however, that hampers development in Europe is probably the lack of sufficiently trained specialists. Here the fact that America has an advantage of about eight years longer experience and a larger number of installed computers is of the utmost importance. The need to train a growing number of systems analysts, programmers and other computer specialists is not yet fully recognised. Of course, there are people who recognised this need and the implications, but in general the number is small.

The Netherlands is one of the first (if not the first) of the European countries where a series of examinations has been set up, with governmental approval, to enable people to obtain an officially recognised certificate in data processing. This series consists of seven examinations, the study for which takes—in evening courses—three to four years. The seven examinations are:

1. Technical aids and appliances for administrative work; this requires knowledge of all types of office machines.
2. Programming and operating computers.
3. Business administration and economics.
4. Mathematical techniques for decision-making.
5. Administrative organisation.
6. Techniques for systems analysis.
7. Organisation of information processing.

Besides this set of examinations, there is another consisting of two parts, each taking a year of evening courses for preparation. Called practical examination for administrative automation, it is designed for those who are going to work in computer departments on the organisational side, such as junior systems analysts. The training concentrates heavily on the practical side of the work, such as techniques for charting, inquiring, interviewing, reporting, etc.

If the need for these and similar training programs will be recognised on a broader scale than it is now, and the necessary measures are taken to fill this need, a very important obstacle in the development of automation will be eliminated. This signs are there that this will soon happen. ■

PROGRAMMING PERSONALITIES IN EUROPE

it takes all kinds

by LIBELLĀTOR

One might think that computing in Europe could not really be very different from computing in America, since Europeans use American computers anyway. Once there was a viable, indigenous European computer industry but it is rapidly fading away due to the American onslaught—aggressive marketing, price cutting, mergers, licensing agreements, etc. But, leaving aside economic and industrial factors, let us turn to the programmer himself.

As does perhaps no other scientific or semi-scientific endeavor, computer programming allows the practitioner to express his personality and creativity in a unique way. His only real constraints are that the final results be nearly correct (to at least three or four decimal places) and that the program be finished within six to nine months after the scheduled completion date. Technical documentation is sometimes requested of him but the pressure of a new program is usually sufficient to scare it away. (If one really must document, he can always invent a new metalanguage to make it more interesting.) Our occupation is far more like an art than a science and, as long as the vagaries of individual personality and "style" are tolerated by an uncomprehending management, it will continue to be, in spite of programming language standards. Even if we are all one day going to be programming in PL/I, I am sure there will continue to be differences in approach and results due to traditional differences in personality and attitude between American and European programmers, and among European programmers themselves. The author has worked for 10 years as a programmer in America, England, France, Germany, Switzerland, with short stays in Norway and Holland, and has worked with programmers from these countries, as well as Armenian, Chinese, Egyptian, Greek, Indian, Israeli, Italian, Jugoslav, and Spanish programmers, He can report a number of interesting differences but must do so under a pseudo-classical pseudonym to avoid being tarred and feathered by his associates.

lunch is serious business

In America one usually observes programmers during lunch hour playing games: Bridge, Chess and Go seem to be most popular (in about that order). I have never seen European programmers playing games during lunch hour. In Europe, especially in France, the big lunch is far too sacred an institution to sacrifice to any professional or intellectual activity. The visiting American can almost depend on getting a few quick runs in between 12:00 noon and 2:00 pm; however, he must expect sometimes to operate the machine himself.

Because of the wide gulf which exists between the American and the European approach to computing, and the noticeably resultant lag on the Continent, American consultants in Europe often get quite a big head from dealing with and teaching people with respect to whom they are relatively so advanced. I have heard a number of young programmers say that they plan to go back to America and start a software company. If this is their desire then perhaps they should remain in Europe, for when they return to the U.S. they will probably find themselves to be several years behind the state of the art.

what is an assembler?

On arriving in Europe, the American programmer is astonished to discover the contrast between the far-out university computing groups, each of which seems to have designed and written its own ALGOL compiler, and the government, industrial and business computing groups which program everything in assembly language (or worse, basic machine coding in octal or decimal). This contrast is especially noticeable in Germany. One European programmer well trained in American techniques went to a new installation in Germany and taught a two-week course on the assembly language of a forthcoming machine. After two weeks of attentive silence, one of his students ventured to inquire: What is an assembler? The manufacturer than beat a hasty retreat and later sent in an instructor to teach a three-week assembly language course. This man—also a German—was very experienced in teaching programming and brought across the idea of symbolic representation, at least in individual instructions. Following his course, an American systems programmer was placed on site as a consultant; he had to spend weeks to get across the idea of symbolic programming. This might not be surprising if the students were all novices but instead they were all very experienced. Some of them had five years programming experience in octal machine language.

In Germany one seldom hears of ALGOL or FORTRAN outside the university; assembly language programming is very much the thing and the industrious Germans excel at it. In France, one finds little patience with the details of any sort of basic coding. There is still a French ALGOL, there was once a French FORTRAN II, but nowadays FORTRAN IV (English) is quite popular. The French seem to be good FORTRAN programmers with one exception: whoever wrote the first FORTRAN programmer's manual in French must have either left out the chapter on subroutines or else have written a very obscure one. French FORTRAN programmers tend to write incredibly long and complex main programs without a single subroutine. These programs often take longer to compile than to execute with a tape-oriented FORTRAN compiler and always seem to overflow a drum-based system. The project manager of a new installation in Paris was brought FORTRAN main programs consisting of 1000 statements, having DO-loops nested seven or more deep and numerous arithmetic statements running six to eight continuation cards. Their furious owners insisted that these programs had run before (they had not) so must run again without changes. The solution to all three excesses is obvious but *sous-programme* was a new word to them.

French engineering education is more theoretical and abstract than pure mathematics in America. One French "programmer" with an MA degree in Engineering came to us without ever having seen a computer. He had such courses as graph theory, coding theory, logical design, Turing machines, abstract algebra, and even a bit of engineering. Incidentally, after a brutal course in FORTRAN and assembly language programming he made a first-class programmer. Occasionally, however, when confronted with a program he didn't want to write, he would prepare an analytical solution in less time than it would have taken him to program the problem for the computer. It takes all kinds.

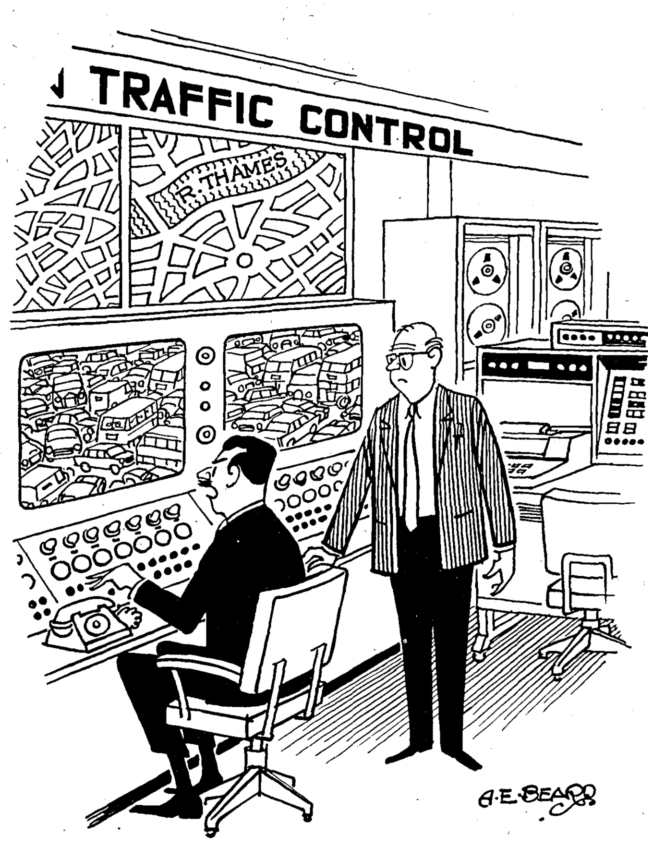
Even very experienced European programmers have very little practice with systems programming or with integrated data processing systems; however, these terms have become very popular over here in recent years. When a programming group manages to get some kind of big complex piece of systems programming to work, they invariably become puffed up with a sense of their own self-importance, overestimate the value of their contribution and quit to find better positions. Overall, programming seems to follow the European pattern in research and development, and even industrial development; i.e., carried out by the lone individual researcher, or the leader who is far more competent and experienced than his subordinates. The project accomplished by "team work" is such a rarity that any team of equals which coalesces to program or develop a system seems to believe it has wrought a miracle of some sort. Team work is so common in America that its advantages are taken for granted. Not so in Europe.

who is a programmer?

One curious facet of programming in Europe is determining just who is a programmer. Capable computer specialists who occupy themselves with getting results from computers seem to be very loath to use the title "programmer;" it is somehow rather demeaning, as if they worked for a living rather than pursued a profession. Part of this may be due to the fact that the "programmer" title does not fit into the title structure in Europe. It isn't nearly as impressive as pre-initials such as Dr., Dr. Ing., Dipl. Ing., etc., or post-initials used to indicate degrees and professional society membership, especially as used in England. This very situation may result from the fact that relatively few European programmers have a professional or academic background. In America, programmers in scientific data processing seem to have been trained as engineers, physicists or mathematicians. Often programmers for non-numerical data processing have been trained in accounting, business, or "came up through the ranks." In Europe the last of these possibilities is most often the case. Thus the programmer is more often considered another kind of clerk rather than another kind of engineer, as he is in America.

English programmers spend more time in talking about programming than doing any. When they do get down to programming, they exhibit the fierce individuality and independence for which the English are known. Once when assigned the task of writing a demonstration program for the announcement of a new computer at a British business equipment exhibition, a senior programmer accepted the assignment but, when asked for it two days before the exhibition, announced that he had not done it after all. His excuse was that the particular machine probably should not be marketed in his country for a number of well-thought-out (and probably valid) reasons. When asked if he analyzed each assignment (according to his own canons) in terms of its possible effects on the company's balance sheet, he replied ingenuously, "Of course, it's a person's responsibility." We were both lucky: the machine was diverted to Switzerland and thus not displayed at the business show in London.

In contrast to his American and English counterparts, the French programmer is often sérieux—i.e., taciturn and reflective. When a new large-scale machine was installed in Paris and demonstrated to the programming and operating staff by the execution of 14 programs in parallel, the French group watched every detail in stoic silence. The American demonstrating the multiprogram Executive was a console expert and typed almost constantly in response to the Executive system's furious bursts of requests and comments. After this *tour de force,* only one comment came from the group of about 25 programmers and operators: "This is certainly an American operating system," said a taciturn fellow from Bretagne. When asked why he said that, he responded: "Because it is all the time talking." ■



"This must be what comes of buying that American computer, Carstairs—don't they drive on the right over there?"

THE ZUSE Z3

german predecessor of the mark I

by WILLIAM H. DESMONDE and KLAUS J. BERKLING

In 1941, three years before the Mark I was placed in operation at Harvard, a program-controlled computer was used by a leading German aeronautical research center. This machine, the Zuse Z3, was built by Konrad Zuse for the Deutsche Versuchsanstalt für Luftfahrt.

Little is popularly known in the United States about the pioneering activities of this German inventor and industrialist. Zuse, born in Berlin on June 22, 1910, started his career as an engineer in the aircraft industry. He soon became aware of the tremendous number of montonous calculations necessary for the analysis and design of static and other aircraft structures.

Pondering on methods for facilitating these time-consuming computations, he conceived of performing the calculations by machine. A computer machine, he reasoned as early as 1934, would have to be able to follow a sequence of simple computational steps, in which all variables and intermediate results would be named. Numbers would be entered from an input unit, and results would be placed in an output device. Internally there would have to be a memory, arithmetic unit, and a control section. Arithmetic would be performed in floating point, with separate units for the exponent and fraction. Numbers and instructions would be represented in the binary number system, and bistable switching units would be used. Machine operations would be synchronized by a central clock. Zuse claims he anticipated Shannon's symbolic analysis of relay circuits by recognizing that all arithmetic and control functions could be reduced to the logical operations AND, OR, and NOT. Zuse quit his job in the aircraft industry in 1936 to have more time to develop his ideas on program-controlled computers. For his first experimental machine, the Z1, he utilized mechanical bistable switching elements for memory and computing circuits. These elements were based on pins arranged in matrices to represent bit positions. Metal sheets with slots were stacked, and the movement of the sheets was transmitted by the pins. The movement of a sheet depended on the positions of the pins in the slots. In those days, components of this type occupied less space and were more reliable than electro-magnetic circuits. By 1937, working in his parents' apartment, he had hand-built a test model for a mechanical memory for 16 - 24 bit numbers. The Zuse Z1, completed in 1938, was completely mechanical, with a binary floating point arithmetic unit.

Because of the difficulty of hand-producing exact mechanical parts, this experimental machine did not work satisfactorily. Zuse decided to make use of relays to attain greater precision. Combining the mechanical memory of the Z1 with an electromechanical arithmetic unit built from scrapped telephone relays, Zuse built a second computer, the Z2. The machine was just about complete when its inventor was called into military service at the outbreak of World War II.

At this juncture, Zuse sought support for his ideas about computers. Like many creative people before him in the history of technology, he met with a brushoff from officialdom, and in some quarters was denounced as a swindler. Succeeding after a year in obtaining a release from German armed forces, Zuse continued his development work.

William H. Desmonde, Ph. D., is the author of Computers and Their Uses and Real-Time Data Processing Systems: Introductory Concepts. He entered the computer field in 1955 with Univac, later was a computer consultant for Price Waterhouse & Co. At present he is a research staff member, IBM Research Center.

He finally received formal encouragement to build a more powerful computer from the Deutsche Versuchsanstalt für Luftfahrt, a German aeronautical research organization.

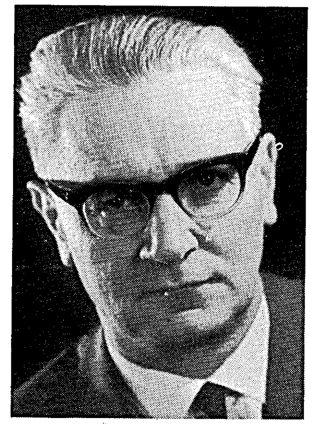
This computer, the Zuse Z3, was completed and placed in practical use in 1941. It was slightly faster than the Mark I, the first big American machine. The Z3 performed about three or four additions per second, and multiplied in from four to five seconds. The machine was a considerable advance over its predecessors. The program was entered by means of a movie film, eight-hole channels being punched in the reel to represent instructions. Commands available were Add, Subtract, Divide, Extract Square Root, Multiply by 2, Multiply by 10, Multiply by 01, Multiply by ½, and Multiply by −1. The relay memory had a capacity of sixty-four 22-bit floating point numbers; the exponent occupied seven bits, mantissa fourteen bits, and sign one bit. The Z3 had built-in translation from decimal to binary and from binary to decimal. Control circuits were composed of step-switches and chains of relays. Altogether there were 2600 relays in the computer, 600 of them used in the floating point arithmetic unit. Data were entered through a keyboard providing for four decimal numbers positioned with decimal point. Output likewise consisted of four decimal digits, displayed by light bulbs on the console.

Later on in World War II, Konrad Zuse founded his own small business, the Zuse Apparatebau. The firm built special computers for use in the production of guided missiles. Mass production of these weapons was causing intolerable deviations from the aerodynamic symmetry in the wings of the missiles. To correct for this about 100 measurements were made of each wing. This data was entered via an analog-to-digital converter into the computer, which calculated the equivalent angle of incidence of each wing. So it was possible to equalize both wings and empennage by adjusting screws. This calculator, a forebear of our modern process control computers, was in operation day and night from 1942 to 1944.

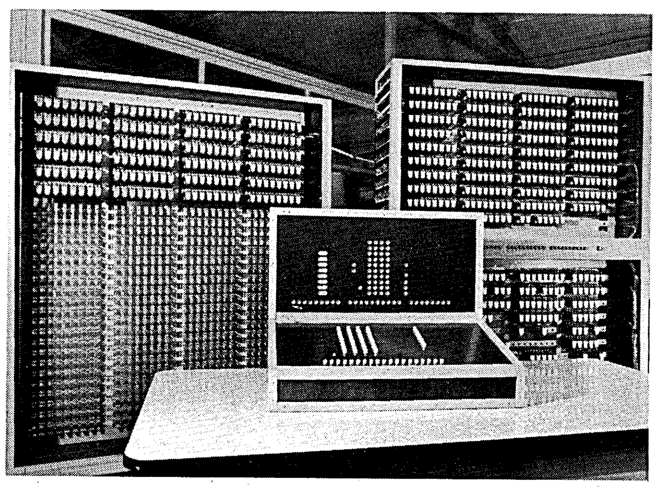
During this period, Zuse built the Z4, an improved version of the Z3, but still with electromagnetic arithmetic unit and mechanical memory. This machine was the only survivor of World War II, by the end of which all of the other Zuse computers were lost. Following the war, Zuse reconditioned the Z4, equipped it with punched tape input for data and with conditional transfer instructions, and leased it in 1950 to the Eidgenossische Technische Hochschule in Zurich. Five years later, the Zuse Z4 was sold to the French Department of Defense, where it was used for another four years. The computer was so reliable that it was customary to let it work through the night unattended.

The success of the Z4 led Zuse to build another relay machine, the Z5, for the Leitz Optical Company in Wetzlar, Germany. This computer, delivered in 1953, was six times faster than the Z4. To handle iterations, instruction tapes were physically looped. Several program tape readers were provided for reading in loops containing different subroutines. Exit from the loops was controlled by conditional transfer instructions.

After producing the Z5, Zuse's firm, the Zuse Kommandit Gesellschaft (now owned by Brown Boveri Company), began manufacturing computers on an assembly line basis. By 1962 the company had delivered 200 computers. Among these were forty-two Z11's with relays, fifty Z22's with electron tubes, and thirty-four transistorized Z23's.



Dr. Konrad Zuse and the Zuse Z3. 1941 machine has floating point and built-in decimal/binary conversion.



The Z22 was the firm's first electronic computer. Development commenced in 1956, and deliveries began in 1958. But as early as 1937 Zuse had envisaged the electronic computer. In that year, he teamed up with Dr. H. Schreyer to develop electron tubes and circuits for a computer. Working together, they completed in 1942 a model circuit for 10-bit numbers, comprised of about 100 tubes. They jointly suggested to the German Government that the electronic computer be built with 1500 tubes and a one millisecond clock-time. The idea was ignored as impossible and unimportant. Schreyer was eventually drafted, and the project was discontinued in 1942. Had they received more encouragement at the time, Zuse and Schreyer might well have developed the world's first electronic computer.

In 1945, Dr. Zuse created a universal formula language that he called Plankalkül. This language was the forerunner of modern program languages. ■

Klaus J. Berkling has a Ph.D. in physics, joined IBM World Trade Corporation in Germany in 1961. He is now a research staff member at the IBM Research Center.

more time and less money

ECONOMICS OF PROGRAMMING PRODUCTION

by ROBERT W. BEMER

A presentation at the Symposium on the Economics of ADP, held last October in Rome, Italy, this article appears in the proceedings of the symposium, which is available from the North-Holland Publishing Co., Amsterdam, The Netherlands.

A software consultant recently advised a large computer manufacturer that each programmer should write eight instructions per hour. I believe that he also stipulated that the sequence should be valid. The astute programmer will immediately write a generator for the computer to produce valid sequences of 10,000 instructions per hour and depart for the Riviera. I should prefer getting a product of maximum utility.

Thinking of what this might have meant in the remote computer past (such as 1956), I recall doing the PRINT I system for the IBM 705 at a completed cost of $17 per instruction. Since it was a compact semi-interpretive system of some 1200 instructions in all, the cost did not seem at all out of line. Yet you may be assured that the programmers were not paid at the rate of (8 x 17) $136 per hour less machine time! The SAGE system of 1957 cost $50 per average instruction, so we must assume that they did not write eight per hour.

Taking this to a more modern absurdity, consider a typical FORTRAN processor of perhaps 35,000 instructions. According to the rule this would require 4,375 hours. At U.S.A. rates of $10 per hour per programmer, it would be delightful to get such a system for a mere $43,750. More likely it will be $437,500 until we are better able to mechanize such production. Actually it is almost possible in the particular case of FORTRAN[1], but I mean to specify the everyday situation.

All this leads to the point that various applications of computers have various complexities, and cannot be accomplished under rigorous and invariant standards. Not that I do not believe in standards, but in the case of programming I must say that I simply do not know what a programmer should produce. That is the business of his manager. What I do say is that when the methods outlined in this paper are followed, any type of programming can be produced at a very great saving over the usual methods of today.

[1] Digitek Fortran (Advertisement), *Datamation* Aug., 1964, pp. 35-38.

documentation

Functional Specifications. Programmers are prone to build without plan. This is the most expensive method for any type of architecture. Even starting with a complete flowchart is nevertheless building without a plan, for they are not equivalent—one may have a perfect flowchart for the wrong process.

The members of each level in the programming hierarchy, from product planner to detail coder, should be obliged to write down in a formal manner the following information according to their responsibility:

1. What is the purpose of the program unit?
2. What are the inputs and their forms?
3. What are the outputs and their forms?
4. What processes are applied to the inputs to yield outputs?
5. What is the inventory of tools (usable store, utility routines, other program units, executive controls) available?
6. What are the constraints of time and interaction with other program units?
7. What are the operational design goals and characteristics?
8. What are the characteristics of interface with other program units?

Mr. Bemer, who has been working with computers for 17 years, is a software consultant to the General Electric Co. He has also been associated with Univac as director of system programming, and with IBM as director of programming standards. He was a member of the ACM Council for six years, represented AFIPS on the IFIP/ICC vocabulary committee, and was a major participant in the development of the ISO/ASCII code.

These functional specifications should be completely settled before any flowcharting, programming or coding is attempted. They should be matched against similar specifications for other units to detect either conflict, duplication, or imbalance in the system. Duplication in particular is a major cost item. Proper functional specifications allow Programmer A to find out that he has a similar need to Programmer B, such that they can share a subroutine.

Operating Characteristics. Some of the greatest losses in computer efficiency occur when unbuffered decisions must be taken by human operators. Therefore the preliminary design should always state the operating characteristics of the program units as embedded in the entire system. Among the items to be specified are:

1. Error conditions and messages.
2. Restart conditions and necessary actions.
3. Complete alternatives to all possible decisions.

Descriptions and Manuals Every program unit must have some descriptive material associated to provide a permanent record of the characteristics which affect the user of the program. Functional specifications do not necessarily have to be made available to the user, as they may belong properly to "technical documentation" which does not have this requirement. This material may consist of but a single sheet of paper, or it may be a complete manual in the case of large and complicated programs. In either case it is advisable to insist that a rough draft be made before any flowcharting or coding is attempted. Obviously certain characteristics cannot be known until the program runs, but it is preferable to indicate this in the original version by a note such as "method unknown—could be . . . . . , or . . . . . , or . . . . ." As the decisions are made, the draft should be updated. This acts as a constant reminder and prevents overlooking of design needs.

When possible, it is preferable for the documentation of minor programs to be integral with the program in the form of annotation. This should lead to prevention of program changes without corresponding changes to documentation. However, formal methods of control should be exercised to ensure that the narrative is still consistent as changed.

standards

Terminology. A major way of lowering programming costs, often ignored, is to better the communication between contributing programmers. Since many programs now have international utility, it is advisable to adopt terminology from the only internationally agreed effort, the IFIP/ICC Vocabulary of Information Processing.[2] This work is structured by concept, and is worthy of careful study prior to usage for looking up individual terms for reinforcement. Missing or newly developed concepts should be brought to the attention of the IFIP/ICC committee.

Other Standards. When costs are a consideration, it is foolish to program without a minimum of standards. There should be an active standards unit in every production programming group, policing compliance with national and international standards as available.[3,4,5] In addition there must be internal, local standards on such items as:

1. Consistency of appearance and documentation.
2. Calling sequences.
3. Description of programming units with respect to algorithm of solution, restrictions, degenerate cases, range, valid classes of data, test cases, etc. It is advisable to adopt a widely tested method to be able to interchange and use the programs of others for economy.[6]

Listings. Listings are the backbone of documentation. However, in early production it is difficult to properly balance use of handwritten correction and notes, on the one hand, with amendment by complete reprocessing on the other. Patches are to be avoided unless they may be accomplished by the most foolproof and mechanical methods.

One must keep good records at any stage of development. In 1963 a large software house did a major system by keeping changes solely as a superimposition of various tape systems, without benefit of updated listing. In the instability of early development there were a few times when the current version was destroyed, necessitating a reconstruction period of up to two weeks to recover the current system. At their rates, each occurrence lost up to $80,000! Doing it all by punch cards and overnight off-line listings would have been cheaper. It is always better to spend extra effort to keep the cleanest possible record, so that each iteration may be taken as a complete restarting point.

There is a simple compromise. Allow a little extra space on each listing page by not filling it completely. Give the top line of each page an artificial identifier if it does not already have one, all of which are kept in a list associated with page number. Since most modern programming systems produce coding relocatable by hardware or software, actual store assignments are somewhat irrelevant. Thus it will often be necessary to make a new listing page selectively only where programming changes occur.

design

Checklists. Because of the nature of the work, a programmer usually desires to invent something. However, given a variety of previous wheel designs it is likely that he will spend this effort on something not so often re-invented. This is the purpose of the checklist.[7] It recognizes that most programming problems are of a highly recurrent nature. It also recognizes that total recall of all contingencies or ways of doing things is unlikely for most programmers, just as the doctor does not always remember the totality of symptoms without aid. For example, it is trivial for the programmer to check off or complete such items as:

1. The source code for this assembly system may come from (punch cards, paper tape, magnetic tape, OCR, ...)
2. If the computer stops with (give here a combination of conditions), the operator should (. . . . .)
3. Data named (. . .) are (always/often/never (numeric/alphabetic/ (other)) and require (. . .) positions on a (punch card/paper tape) in the format (. . . . .) and position defined by (. . . . .).

Obviously the last item could often be in multiple, and could therefore be compacted in a tabular form.

Flowcharting and Logic Equations. Programs should be carefully designed, by whatever means. Flowcharts enjoy a certain popularity for clarity. However, they are usually not so necessary when programming in a language like COBOL. Logic equations have the capability of

---

[2] Wilde, A. R., et al, IFIP/ICC Vocabulary of Information Processing, North-Holland Publishing Co., Amsterdam, 1965.

[3] International Standards Organization, Technical Committee 97, Computers and Information Processing, Scope, Geneva, 1961.

[4] ISO/TC97/WG-G (Secr-29)62, Second Draft Proposal, Flowchart Symbols for Information Processing, ASA, New York, Aug., 1964.

[5] ISO/TC97/SC2 (Secr-37)130/FE, Fourth Draft PrProposal, 6 and 7 bit Coded Character Sets for Information Processing Interchange, AFNOR, Paris 2e, March, 1965.

[6] Grems, M., *Proposal for an ACM-JUG Computer Applications Digest*, Minutes ACM Council, May, 1965.

[7] Bemer, R. W., *A Checklist of Intelligence for Programming Systems*, Communications ACM 2, March, 1959, pp. 8-13.

being formally manipulated for minimization or seeing that all negations are accounted for.

Modularity. Always build a program of any size in discrete modules, with known inputs and outputs, together with the interior process. These should be so independent that they may be linked together in almost any order, just like railroad cars. This might require 3% more instructions overall, but it is worth it in costs of maintenance and diagnosis.

Every program unit should be created in three forms for testing:

1. As a self-contained unit, complete with synthetic input output, created perhaps by a generator.
2. In a form suitable for usage within its own major program.
3. In a form suitable for use within the overall system.

Often the extra instructions required for (1) and (2) may be removed mechanically for the final stage.

production control

Due to the invisibility of programs, normal control methods are ineffective. Mechanized control and feedback is even more important than the precise organization of supervision. The steps are:

Estimation and Budget. Software units of the minimum size feasible for individual control are defined, named and given identifying numbers. Planning provides a working description of function. Supervisors estimate the total elapsed time and cost for man- and machine-hours. This is the primary input to the budget. In the case of large concerns with many programmers at different locations, precise definition of a programming unit to be fabricated allows for competitive bidding among these groups, with corresponding expectancy of cost reduction.

Labor Distribution. Supervisors distribute the total elapsed time by benchmarks (functional specifications, flowcharts, implicit quality test, coding, checkout in vacuo, checkout in processor, checkout in system, documentation, explicit quality test, release). Labor distribution reports are developed by means of time cards. These are correlated to the estimates. The individual programmer periodically estimates the percentage of completion of each unit. If the system is run on a computer, it is possible to flag estimated overruns in hours and delivery times, inconsistencies in reporting precedence dangers on PERT schedules, etc.

Correction and Adjustment. Supervisors add revised benchmark estimates to project charts, which show initial estimates, last revised estimates and actual completions. In danger areas, management may rebalance the staff, redesign, etc. The eventual users of the programs are notified of revised dates so they may modify plans, check contractual commitments. As these are official company records, detected falsifiers may be discharged, as merited discipline is usually effective in reducing costs. The supervisors may be recalibrated as optimists or pessimists, but more often they will automatically adjust their estimating as a byproduct of the system. Data presents itself for practical standards of production, in those areas where it is feasible to have such standards.

diagnostic methods

General. Computer operation has become more complex with each year of usage. Part of this is attributable to the wider use of FORTRAN, ALGOL and COBOL languages. But even assembly languages have become more complex, and all of these now run under executive systems likely to be more intricate than the languages. Under such conditions, the programmer is likely to be at a loss to find out whether a malfunction is due to:

1. A hardware malfunction.
2. A malfunction in the programming system he is using.
3. An operating mistake.
4. Data errors, such as unexpected type, outside of expected range, physical errors in preparation or reading, etc.
5. His mistake, such as a misunderstanding or disregard for the rules of syntax, grammar, construction, file layout, system configuration, flow process for solution, etc.

The hardware field engineer is subject to the same confusion. However, there are certain ways of discovering the class of the malfunction and directing the evidence to the proper authority for correction. The programmer should not be too surprised if, after following the methods outlined here, this turns out to be himself in most cases.

When using a programming system, remember that there is probably no single person that understands the entire system and its individual components well enough to diagnose 100% of the troubles. This means that most diagnosis must be done by *cause* and *effect*, rather than tracing through the operation. The "black box" simile must be appreciated and used. One must put certain inputs into the box, observing the form of the outputs. One then varies the inputs and observes the corresponding changes (if any) to the form of the outputs. By careful design of the inputs and their variation it is possible to deduce which internal element of the black box must be at fault.

This means that the programmer must adopt the scientific method of "design of experiment." The object is to get as much information as possible during each run (or experiment) and to make as few runs as possible. Thus many items of information should be obtained from each run, but the variations must not interfere with each other to the extent of obscuring information, and each bit of information should lead to the next set of modifications by reducing the possibilities.

Before the user can call upon outside help, it is his responsibility to clearly demonstrate the malfunction. Further, he should provide the *minimum* segment of the program which exhibits the malfunction. Thus isolation is the first process to undertake.

It is much cheaper to be prepared for a malfunction than not. A good rule to adopt is that "The program is wrong when first ready for testing." The unusual ("degenerate" to the mathematician) case occurs when the program is correct just prior to production runs. Cases are known where the average number of times to compile or assemble a program for test was in excess of fifty before it operated satisfactorily. This is too expensive and delays production to an intolerable point.[8]

Practical Methods. 1. Multiple Service per Run. There are few things as shameful as seeing a programmer run a program to blowup point, take a full dump of the store and get off the machine. This is expensive in machine time and slows his productivity. Observe the following program structure:

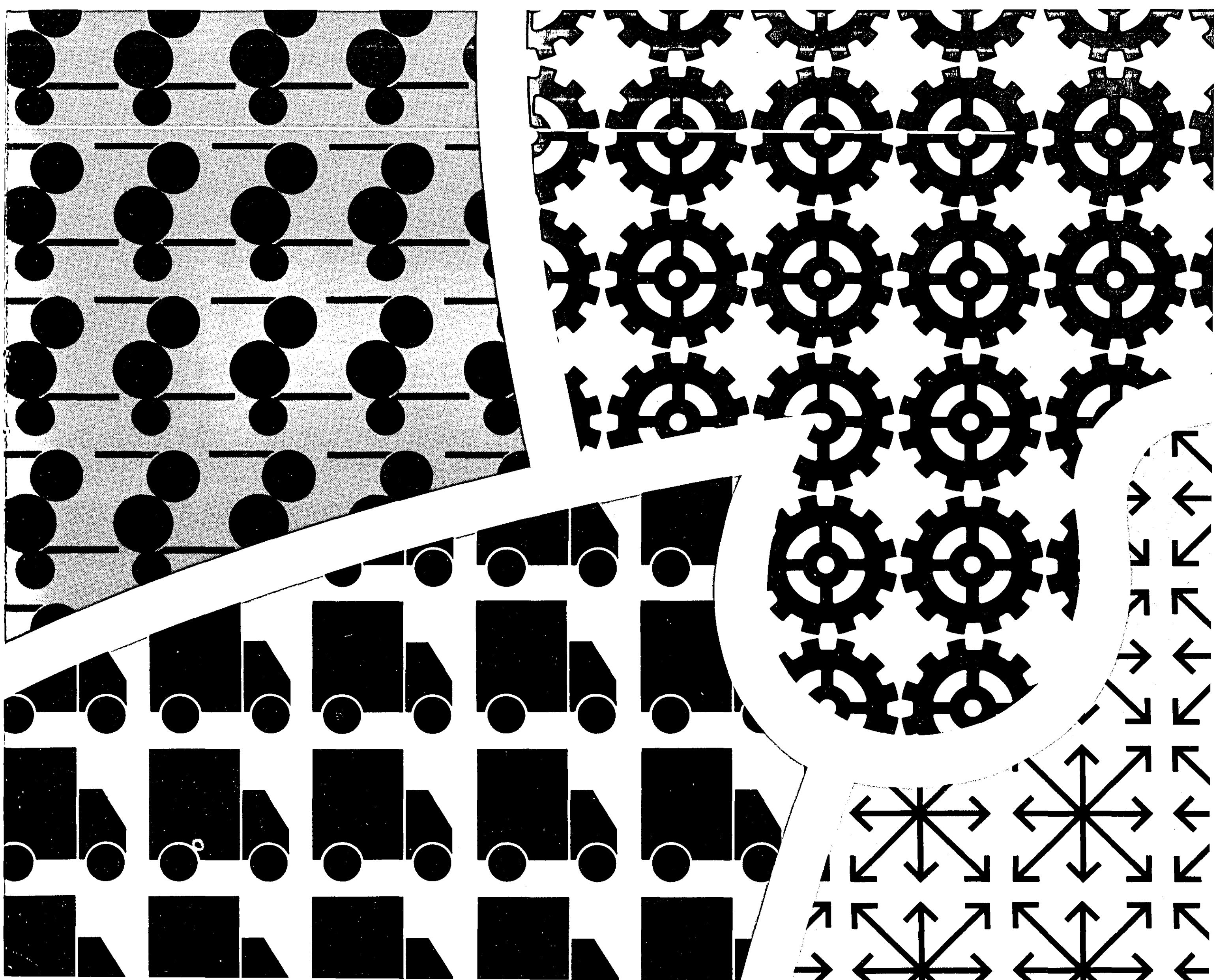
Read initial values of parameters
* List values as read
Compute A, B and C
* Read correct values for A, B and C according to the initial values given. Call them a, b and c.
* Compute A = a, B = b and C = c.
If all zero, print "A, B and C OK" and jump to "Next

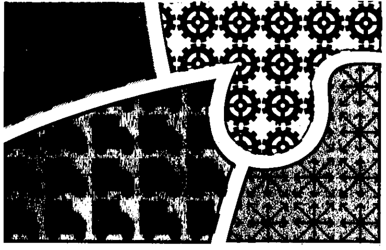
[8] Senko, M. E., *A Control System for Logical Block Diagnosis with Data Loading*, Communications ACM 3 (1960), pp. 236-240.

One of a series on topics of importance to data processing management

Honeywell report on computer Application Systems

Most data processing equipment users are impressed by the dramatic increase in problem solving power offered by third generation computers. To help users exploit this power with new applications and improved versions of existing applications, many computer manufacturers provide a variety of industry-oriented application systems. When properly designed, an application system can be an extremely useful computer tool. Whether a company is taking its first step into electronic data processing or replacing old and outmoded equipment, application systems can significantly reduce the effort required to plan and implement computer solutions to the user's data processing problems. This report describes the important role application systems play in the effective harnessing of new generation computing power, defines the design philosophy behind Honeywell's extensive and constantly growing catalog of application systems, and illustrates how an application system can help the user cash in on his most profitable data processing operations.





A REALISTIC LOOK AT APPLICATION SYSTEMS

Application systems are based on the premise that certain broad data processing functions are common to many companies and are therefore applicable throughout a particular industry. This might imply that these functions can be precoded and offered to users as "canned" or "off-the-shelf" programs which can be put to work immediately on the solution to a particular problem. Except for applications of a fairly limited nature, such approaches have proved relatively unsuccessful because even within the same industry, two companies rarely have identical data processing requirements. In other words, an off-the-shelf system is either so general that it burdens the user with inefficient operation or so specific that it cannot provide a direct fit to a user's particular needs. Therefore the well designed application system is the one that does not take a rigid prepackaged approach to system design. Instead, it anticipates differences among users and provides practical sub-system modules which offer each user the flexibility necessary to tailor the system to his requirements. The following list points out some of the things you should and should not expect from an application system.

APPLICATION SYSTEMS CAN:

1. Equip you with the computer manufacturer's systems know how, gained through years of experience with companies involved in applications similar to your own.
2. Provide useful guide lines in tailoring a system to specific needs.
3. Help reduce the major portion of system set-up cost — application analysis and design.
4. Provide precoded functional subroutines which can be inserted into users' programs.
5. Provide a sound framework upon which the user can build an integrated management information system.

APPLICATION SYSTEMS CANNOT:

1. Determine what your data processing objectives should be.
2. Automatically make data processing operations more profitable if management has not carefully determined the actual costs involved in carrying out an application.
3. Provide the highest level of performance on new generation computers if the application systems are merely updates of older systems embodying the processing concepts of older, less powerful computers.

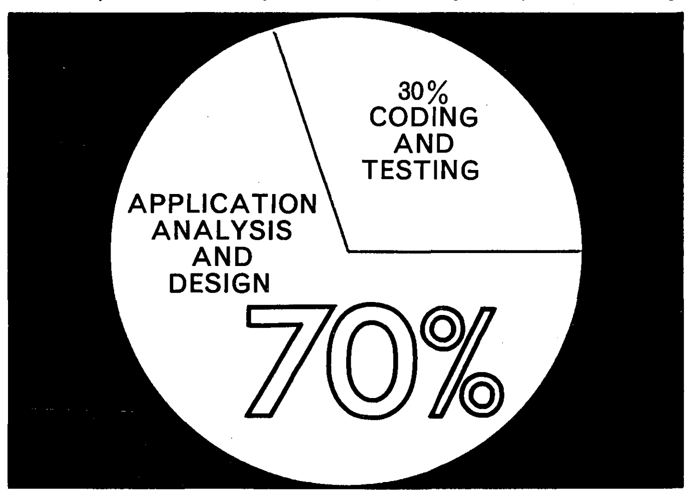
DESIGN PHILOSOPHY OF HONEYWELL APPLICATION SYSTEMS

One of the most important steps in the successful exploitation of a computer is finding the profit producing applications. These are the applications that help improve management decision making and control of operations, and which, once they have been pinpointed, must be carefully analyzed and designed. This will involve such activities as

1. Report specification
2. Determining how the job will be divided into manageable computer runs
3. Specifying transaction processing rules.
4. Designing logic flowcharts
5. Determining the file design
6. Planning appropriate controls
7. Determining manual procedures

Experience has shown that analysis and design of an application system requires close to 70% of the start-up effort. The remaining 30% covers programming and testing.

Honeywell application systems are directed primarily at reducing the effort and cost associated with the major part of the job by providing application analysis and design "building blocks". Every Honeywell application system consists of detailed documentation about the application itself, general and detailed flow charts describing every aspect of the system, item designs, program logic, and input/output formats. In addition, to ease implementation problems, Honeywell provides fully



coded and tested program modules for applications which are general in nature and common to the needs of many companies.

THE BUILDING BLOCK APPROACH

While it is true that the data processing needs of many companies are similar with respect to certain broad data processing functions, they are not identical. Therefore Honeywell designs its application systems to be as modular as possible in order to simplify individual specialization and tailoring of systems to specific needs. A Honeywell application system can be viewed as a logical configuration of system design and programming building blocks, capable of being tailored to the unique requirements and equipment configuration of the user's company. Carrying this building block concept a step further, each Honeywell application system can be viewed as a subsystem of an integrated management information system. Each individual application system is designed to meet the needs of a restricted area of the organization, but its open-ended design anticipates its use with other systems to meet the needs of the organization as a whole. Thus any particular system building block can be used as a separate entity or as a part of an integrated system. Once a user has put a basic application into operation, other system functions can be computerized with a minimum of additional effort, time and money.

USER PARTICIPATION INSURES PRACTICAL DESIGN

Application systems cannot be developed in a laboratory. The most effective systems are developed by a partnership between user and manufacturer. Virtually all of Honeywell's application systems have been developed in conjunction with progressive companies in major industries. For example, Honeywell's STET (Specialized Technique for Efficient Typesetting) system was developed in partnership with the St. Petersburg Times. A system for the distribution industry called PROFIT (Programmed Review, Ordering, and Forecasting Inventory Technique) was developed in conjunction with a group of eight Honeywell customers. And a system for inventory control forecasting was developed with Honeywell's Microswitch Division. Teamwork such as this insures that each system is designed to be not only a technical success, but far more important, to be a commercial success to the company that will eventually use it.

FACTOR — THE BUILDING BLOCK CONCEPT IN ACTION

FACTOR is Honeywell's phased approach to an integrated factory management system. It is summarized here to illustrate how Honeywell's building block concept enables a company to grow in planned, manageable steps toward an integrated management system.

FACTOR is designed as a single, total entity to control an entire organization — rather than a piecemeal development of many more or less independent applications. It includes close to a dozen interrelated system building blocks, each one designed to meet the specialized needs of limited areas in the organization. Because FACTOR is designed around a central data base, each building block fits into the overall system in such a way that redundancy of data storage and the transmission of useless information from one area of the organization to another are eliminated.

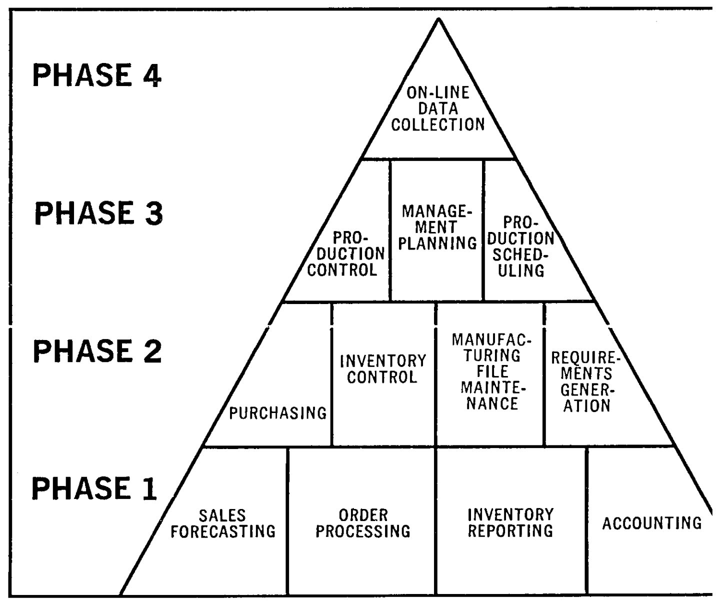
As illustrated in the following column, FACTOR is designed to be implemented in four major phases. Thus the complete system can develop gradually; the period of time depending on the resources, experience and unique requirements of each company.

PHASE 1 As an example of the activities to be computerized during Phase 1 consider the functions performed by the Inventory Reporting building block. This subsystem handles the day-to-day transactions — issues, receipts, etc. — from the stockroom. It issues shipping documents to fill orders as stock is available and delivery is due. It also checks on planned production, issues replenishment requests to production planning and notifies of imminent stock-outs and surpluses. The subsystem covers both finished goods and manufacturing inventories.

Another important Phase 1 building block is Order Processing. This subsystem deals with customer orders from preliminary logging, technical verification and credit checking through to entry into the central data base and reporting of possible delivery dates. When the order becomes due and stock is shipped, the Order Processing subsystem issues an invoice to the customer.

PHASE 2 During Phase 2 certain initial applications are augmented and important new applications are added. For example, once basic inventory data has been established by the Sales Forecasting and Inventory Reporting subsystems, all the functions of inventory control can be implemented using the Inventory Control subsystem.

A major addition to the system during Phase 2 is the Requirements Generation building block. From an input consisting of finished goods and spare parts requirements, this subsystem carries out an explosion and netting process to generate net requirements of subassemblies at all levels, piece parts, raw materials and purchased parts by time period. Output from the subsystem is a production program expressed in quantities of piece parts, subassemblies, and final products to be made in each time period.



FACTOR BUILDING BLOCKS

PHASE 3 In Phase 3 the importance of the central data base is highlighted by the Management Planning building block. Within this subsystem are a number of programs for producing overall management information on the basis of data accumulated in the central data base. Among the valuable reports which may be produced are factory utilization, forecast factory load reports, budgetary status data, cash, and capital assets reports. External economic and market data enters the data base where various techniques (e.g., linear programming and simulation) are available to produce meaningful planning data for management.

PHASE 4 In many situations, an efficient production control system can operate without recourse to an automatic data collection system. However as tighter control and coordination of operating elements becomes more critical, it will be desirable to incorporate on-line data collection and inquiry stations.

Honeywell Series 200 equipment configurations can be easily designed to match the users' needs for each phase of his progression toward the fully integrated system. Generally speaking, Phase 1 building blocks are

A REALISTIC LOOK AT APPLICATION SYSTEMS

oriented to sequential tape processing, Phases 2 and 3 require the addition of random access devices, and Phase 4 includes data communication capabilites. Series 200 offers the broad range of processor, peripheral and data communication facilities necessary for the successful implementation of an integrated management system.

A CONSTANTLY GROWING CATALOG OF APPLICATION SYSTEMS

Honeywell supplies application systems for use with Series 200 computers in a wide range of industries. Brief descriptions of some of the more advanced major systems follow. Use the coupon below to obtain more information on these and other Honeywell application systems.

PRINTING AND PUBLISHING

Honeywell has several systems to offer newspapers, magazines, and general publishing companies. These include Honeywell's STET (Specialized Technique for Efficient Typesetting) systems for handling hyphenation and/or justification for hot-metal or photocomposition typesetters.

STET-1 is a comprehensive typesetting system which provides facilities for the layout, justification and hyphenation of all types of copy for hot-metal line casters. STET-1 is one of a series of linked subroutines written to facilitate modification, extension, and integration into a larger system. A STET monitor is available which allows a single central processor to do both accounting and typesetting on a shared time basis.

Photo-STET includes all of the facilities of STET-1 plus additional features which permit the full exploitation of presently available photocomposition devices. Special care has been taken to insure compatibility between the STET hot-metal and the Photo-STET input routines. This enables the same operator to keyboard either hot-metal work or photocomposing in a shop using both processes.

FINANCE Brokers, investment fund trustees, mutual fund administrators, investment councilors, and insurance companies are some of the potential users of Honeywell's Computerized Portfolio Analysis system. The programs in this system enable a subscriber to Standard and Poor's Compustat Tape Service to evaluate and compare the status and performance of several companies rapidly and simultaneously. The system is capable of calculating several important ratios and indicators of a company's growth stability, and prospects. It also produces a company-versus-industry comparison, allowing an even more informed analysis to be performed. The analysis can also be used to select companies which meet criteria specified by the investment analyst.

SERVICE CENTERS A flexible Estimating and Control system provides the service center with several tools to help maximize profits through accurate job costing. One part of the system can be used in estimating and controlling systems analysis and programming efforts required to implement new computer applications. It matches programming complexity to available programming skills in order to project completion dates. It can be used as the job progresses to match actual performance against scheduled performance.

A second part of the system prepares accurate estimates of computer run times for new jobs. It accepts input in the form of simple narrative statements describing the user's application and the equipment configuration it is to be run on, and produces detailed reports on the computer time used for each run in the application.

DISTRIBUTION Honeywell's integrated systems for the distribution industry start with four fundamental building blocks:

SALE — an order-processing sub-system that enters orders, issues customer invoices and warehouse picking documents, and updates the appropriate accounting records.

CASH — an accounting system that integrates accounts receivable, accounts payable, and general ledger accounting, and also summarizes and analyzes data for management reports.

PROFIT — a dynamic order-strategy system for inventory control and purchasing that optimizes inventory levels and replenishment orders at the most desirable customer service level.

DISPATCH — a system that schedules the transportation for a distribution operation; it groups orders by truck or carload and provides the best delivery route consistent with predetermined travel and unloading times.

Honeywell

ELECTRONIC DATA PROCESSING

): Honeywell EDP
60 Walnut Street
Wellesley Hills, Mass.
Attention: Information Services

E

ease send me additional information on Honeywell application stems for the industries checked below:

BANKING	☐ LIFE INSURANCE
DISTRIBUTION	☐ MANUFACTURING
EDUCATION	☐ RETAILING
FINANCE	☐ SERVICE CENTERS
FIRE AND CASUALTY INSURANCE	☐ TRANSPORTATION
HOSPITALS	☐ PRINTING AND PUBLISHING

ıme____________________

tle____________________

ompany____________________

ddress____________________

ty____________________

ate____________ Zip Code________

step"
If not, print

```
        A = ....      a = ....
        B = ....      b = ....
        C = ....      c = ....      , and
*   Compute A = a, B = b and C = c.
    Next step
```

The steps marked with an asterisk should normally be removed only when testing is complete and correct. This can often be done automatically during final compilation by a switch mechanism. Do not remove in stages, as correct sections may be again incorrect upon changes.

It takes little effort to adopt this plan, particularly if called by a FORTRAN subpositive.[9] It ensures that the next program segment can be checked independently in the same run. Good practice dictates that the programmer divide into at least ten such parts per run!

2. *Controlled Data.* Allowing complete freedom of data characteristics during original testing can introduce too many complexities to see clearly what is going wrong. Select certain values for inputs and run them through the algorithm to determine the expected results for selected combinations. Make the selection according to:

a. For numeric parameters, take values at the end points of expected or allowable range.
b. For non-numeric parameters, take typical or singular cases that display all expected characteristics.
c. In either case, vary for minimum and maximum field length.
d. Select "bad" data with specific characteristics such that they should not work in the program.
e. To check moves, do the inverse and compare to itself, like a matrix reinversion. Build this in and remove when correct.

Test to determine that all valid data yields correct answers and that the bad data always yields error conditions and messages. Subtract check answers from actual and blank zeros before printing.

3. *Live Data.* Live data should be used only after obtaining correct results with controlled data. In case of malfunction, check that the live data:

a. Conforms to data characteristics which the documentation shows to affect program action, and matches format rules.
b. Comes from the proper physical input unit.
c. Does not contain invalid characters, singly or in combination.

Check the answer range. Overflow and underflow truncate can give unrecognizable answers.

4. *Desk Checking.* Machine time is still expensive enough to warrant considerable desk checking. I say this despite any claims in this area for on-line man-machine interaction with time-sharing. The programmer should:

a. Check conformity to rules, such as those for justification.
b. See that enough restart points exist for long programs.
c. Compare actual program logic for match with intended logic for match with intended logic as given by a flow chart or equation.
d. Examine live input for peculiar characteristics which could cause erroneous branching, such as bad data, blank records, etc.
e. Inspect the list of identifiers produced and assigned by the processor, looking for conflicts, insufficient definition, completeness and spelling.
f. Check permissible spellings of reserved words, allowable usage of spacing, hyphens and commas, and juxtaposition of illegal word or operation pairs.

Obviously much of this should be detected by a well-designed processor with complete error message facility, but this is not always so.

5. *Branching.* When the decision structure of a program is at all complex, always plan a path flow in the testing. This may be as simple as printing the value and name of the element tested, printing a suitable indicator for branch or no branch. It is good to print, in the extreme righthand columns if convenient, a code or label of the first instruction in the branch to identify the branch selected. Print this during execution of the branch sequence, not when the decision is made!

When the proper branch is not taken for some reason, invert both the test and the branch destinations. For example, the following program segments are identical in function:

```
    IF A = B, GO TO P              IF A ≠ B, GO TO G
  G ............                     GO TO P
    .                              G ............
    .                                .
  P ............                     .
                                   P ............
```

If they work differently, it is obvious that the mistake lies in obtaining the form of A and B.

6. *Operating.* The goal of operating the program in the test environment should be to develop the simplest and smallest program segment which exhibits the malfunction, regardless of whether the eventual cause is shown to be the responsibility of hardware, the software system or the user. To this end:

a. Reduce the program in size and complexity.
b. Isolate suspected sections of coding and equip them to run individually, but in groups one after the other. Test to see if the malfunction has disappeared. If not, add original elements until it reappears.
c. Simplify the section of coding. Replace arithmetic statements by simple statements like A=B. Simplify variable names. Put complex flow in line.
d. Check all diagnostic messages for clues.
e. Check to see if dual or complement types of instructions also cause the malfunction, or simply an expected wrong answer.
f. Make several physical copies of the malfunctioning section. Vary in several ways, adjoin copies and run together for efficiency.
g. Reprogram for alternate methods of achieving the same result. This is often the simplest way to overcome blindness to the cause of malfunction.
h. In difficult cases, change values of only one variable at a time for controlled experiment.
i. Make full use of manufacturer-supplied tools such as de-flowcharting, dynamic testing routines, utilities, etc.

Additional treatment[10] is available.

7. *Quality Control.* The best way to avoid malfunctions is to build software with quality controls applied during manufacture. All original programming, changes and additions to programs are done preferably in a computer-controlled environment.[11,12]. Such environments should be in general use by computer manufacturers by 1967, and should be available to users. ■

---

[9] Univac, General Manual, Sleuth II for 1107, UP-3670, 1963.

[10] Univac, P.I.E. Bulletin UP-3910.5, May, 1964.

[11] Bemer, R. W., *Software Systems Customized by Computer*, Proceedings IFIP Congress 65, Vol. II.

[12] Crowley, W. R., *A Possible Future System for Automating Control of the Development, Distribution and Maintenance of Programming Systems*, Proceedings IFIP Congress 65, Vol. II.

CHANGES IN COMPUTER PERFORMANCE

a historical view

by KENNETH E. KNIGHT

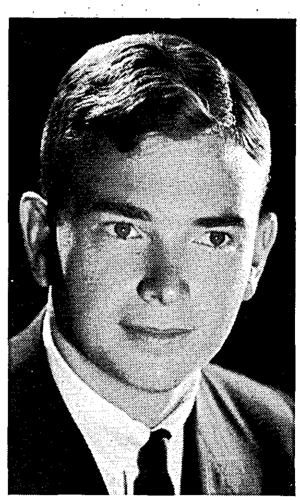
The first 20 years of the computer industry have been hectic ones. Great strides have been taken to provide reliable and inexpensive computation capability. To obtain a clearer picture we will explore our past to see where we have been and how fast we have had to move to get to where we are today. From our analysis of the first 20 years of the computing industry, we have arrived at four fascinating observations that we will discuss in this paper.

1. We generate a performance description for 225 general-purpose computer systems. The performance description estimates the over-all capabilities of each computer system based upon its hardware features and basic elementary operations. We obtain estimates of the performance capabilities for both scientific and commercial computation for 225 different computer systems introduced between 1944 and 1963.
2. Using the performance descriptions for the computers introduced in any one year, we generate a technology curve for that year. The technology curve describes the theoretical performance that can be purchased for different monthly rental expenditures.
3. Grosch's law is upheld. For any one year we find the relation between computing power and system cost to be approximately as follows: Computing power = $(C \approx \text{system cost})^2$; C = constant.
4. Improvement in number of operations per dollar between 1950 and 1962 has been at an average rate of 81% per year for scientific computation and 87% per year for commercial computation.

functional description of gp computers

The capability of each system to perform its computing tasks represents the functional description (or evaluation) of that system. For our purposes we will only look at two aspects of computer performance: 1) Computing power, indicated by the number of standard operations performed per second (P); 2) Cost of the computing equipment, which equals the number of seconds of system operations per dollar of equipment cost (C).

Computing power (P) evaluates the rate at which the system performs information processing, the number of operations performed per second. Two machines solve a specific problem with different internal operations because of their individual equipment features. (P) will, therefore, describe operations of equivalent problem solving value to provide the desired measure of a computer's performance. We will estimate (P) from structure. In order to do this, we first must understand which structural factors influence computing capability. Then we determine the manner in which the structural factors interact to develop the functional model through the use of detailed study of the operation of computing equipment and the problems performed. (P) consists of three main components: 1) the internal calculating speed of the computer's central processor (t_c); 2) the time the central processor is idle and waiting for information input or output ($t_{I/O}$); and 3) the memory capacity of the computer (M). These factors are the important performance measures needed to determine (P). We define t_c as the time (in microseconds) needed to perform 1 million operations, and ($t_{I/O}$) as the



Formerly a consultant with the RAND Corp., Dr. Knight is currently an assistant professor at Stanford Graduate School of Business. He holds a B.S. from Yale, and an M. S. and Ph.D. from the Graduate School of Industrial Administration, Carnegie Tech.

non-overlapped input-output time (microseconds) necessary for these 1 million operations. Therefore, the computer performs $\frac{10^{12}}{t_c + t_{I/O}}$ operations per second. The computer's memory has a strong influence on (P). We found that the memory factor interacts with internal operating time to determine computing power as follows:

$$M \times \frac{10^{12}}{t_c + t_{I/O}} = P$$ [1]

The internal speed of the central processor, t_c, is the time taken by the computer to perform its information processing tasks. The speed equals the internal operation times of each computer, multiplied by the frequency with which each operation is used. To determine the internal speed, therefore, it is necessary to measure the frequency with which the various operations are performed in a typical problem. For scientific computation we considered approximately 15,000,000 operations of an IBM 704 and an IBM 7090, from a mix of over 100 problems. In the analysis of the operations used in this "problem mix" the instructions were grouped into five categories:

1. Fixed add (and subtract) and compare instructions performed.
2. Floating add (and subtract) instructions required.
3. Multiply instructions.
4. Divide instructions.
5. Other manipulation and logic instructions—this category combines a large number of branch, shift, logic, and load-register instructions.

The relative frequency with which each of the five types were used in the scientific programs we traced is presented in Fig. 1. (p. 42).

To determine the frequency with which the different operations were used in commercial computation, nine programs were analyzed in detail (two inventory control, three general accounting, one billing, one payroll, and two production planning). All nine problems were run on an IBM 705, representing over one million operations. We analyzed the nine programs using the same five instruction categories selected for scientific computation. The relative frequency with which each of the five types of instructions were used in commercial computation is presented in Fig. 1. The time the central processor stands idle waiting for information input or output, $t_{I/O}$ is a function of the amount of information that must be taken into the computer, the amount of information that must be sent out of the computer, the rate at which information is transferred in and out of the computer, and the degree to which input and/or output can take place while the central processor is operating.

When we studied the input-output requirements we were unable to count the actual number of pieces (or number of words) read or written. Instead, the time the computer's central processing unit (1) operated alone, (2) operated concurrently with I/O, and (3) idled, waiting for information input-output to take place, was measured. From the actual input-output times, and published input-output rates, it was possible to estimate the number of words read and written. The following computing systems were studied to estimate $t_{I/O}$: IBM 704, 705, 650, 7070, 7090, and 1401; Philco 211; and Bendix G15. The figures for the 7090 were accurately obtained, using the system's clock for single channel I/O, double channel I/O, and double channel I/O with program interrupt. The other figures were obtained by less precise counting methods. The results obtained from the precise 7090 measures, and from the other systems, were very similar and are presented in Fig. 1.

The memory capacity M of a computing system greatly influences its computing ability. Increased memory markedly improves the processing of very large problems which would otherwise be split into subproblems. There are also advantages to larger memories when performing smaller problems because they allow the use of compiling routines, subroutines, etc. Recently, with the advent of multiple input-output capability, and multiple program operation with executive and interrupt routines, larger memories provide additional advantages for all sizes and types of problems.

We were unable to find a feasible means to measure analytically the influence which memory has upon a computer's performance capability. Our best alternative was to obtain the opinions of the individuals who were most familiar with computers. A total of 43 engineers, programmers, and other knowledgeable people were contacted and asked to evaluate the influence of computing memory upon performance. While their opinions varied, their answers were analogous enough to construct the functional model that estimates the effect memory has upon computer performance. The results of our inquiry are presented in Fig. 1.

machines covered

The two characteristics of the functional description for each computer which this study considers are calculated for the general purpose computers (up to the 1963 cut-off date) in the United States known to the author. The list of computers introduced between 1944 and 1963 was obtained through a detailed search of the computing literature. All the systems that did not have structural elements to satisfy the functional model (specifically P), the special purpose computers, were deleted from the list. Computers which are not in the class of functionally similar products defined by the functional model are those that were built and used to perform a set of specialized information processing tasks. As a result these systems contained limited and specialized input-output equipment or limited internal arithmetic capabilities and are not included in our sample.

Most of the recent general purpose computers have been manufactured in quantities from tens to thousands. With quantity production the manufacturers have offered a large number of alternative system configurations. For these computers one functional description does not fully describe the computer. Many of the computers offer over eight memory sizes, three input-output systems, four input-output channel configurations, and four arithmetic and control extras. This represents over (8 x 3 x 4 x 4) 384 different computing systems. Although only a few configurations eventually are produced, the modern systems potentially consist of several hundred alternatives. It would be impossible to calculate (P) for even a few alternatives of each system. We must therefore settle on one configuration for each computer.

There appears to be a good method for selecting the configurations, and that is to consider the most typical configuration of the computer. Where structural changes have been made, we have used the equipment which was available when the system was first introduced. In a few cases where important modifications have been introduced at a later date, these modifications are considered as separate computers, and are treated as such in the study. The calculations of P and C for both scientific

(Text cont'd on p. 45, Fig. 1 on p. 42)

[1] A more detailed description of the development of the functional model is presented in K. E. Knight's *A Study of Technological Innovation — The Evolution of Digital Computers*, an unpublished Ph.D. Dissertation, Carnegie Institute of Technology, 1963.

Fig. 1—Functional Model-Algorithm to Calculate P for any Computer System

$$P = \frac{\dfrac{[(L-7)\ (T)\ (WF)]^{i}}{10^{12}\ [32{,}000\ (36-7)]^{i}}}{t_C + t_{I/O}}$$

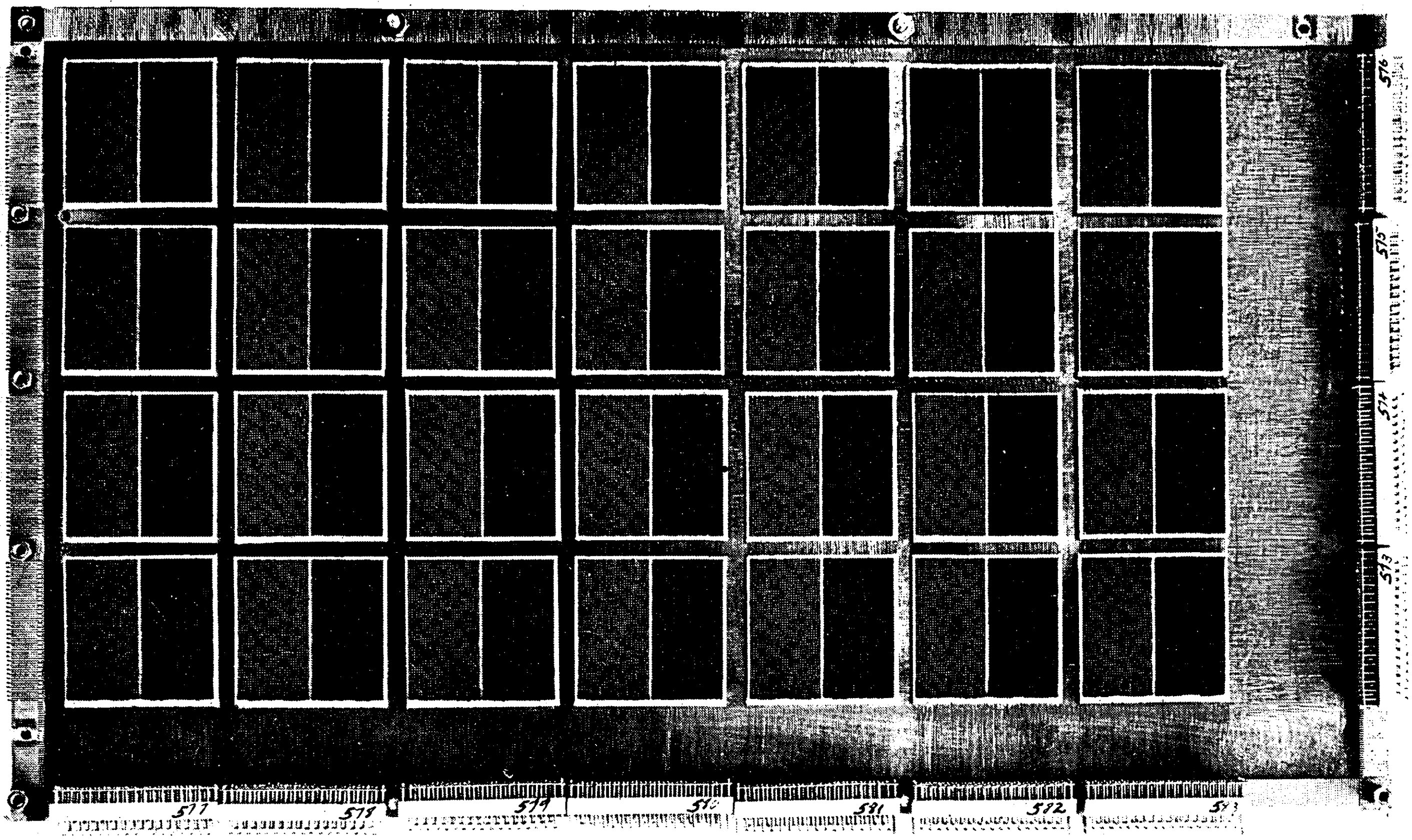
$$t_C = 10^{4}\ [C_1 A_{FI} + C_2 A_{FL} + C_3 M + C_4 D + C_5 L]$$

$$t_{I/O} = P \times OL_1\ [10^{6}\ (W_{I1} \times B \times 1/K_{I1}) + (W_{O1} \times B \times 1/K_{O1}) + N(S_1 + H_1)]\ R_1$$
$$+ (1-P)\ OL_2\ [10^{6}\ (W_{I2} \times B \times 1/K_{I2}) + (W_{O2} \times B \times 1/K_{O2}) + N(S_2 + H_2)]$$

VARIABLES—ATTRIBUTES OF EACH COMPUTING SYSTEM

- P = the computing power of the n^{th} computing system
- L = the word lengths (in bits)
- T = the total number of words in memory
- t_c = the time for the Central Processing Unit to perform 1 million operations
- $t_{I/O}$ = the time the Central Processing Unit stands idle waiting for I/O to take place
- A_{FI} = the time for the Central Processing Unit to perform 1 fixed point addition
- A_{FL} = the time for the Central Processing Unit to perform 1 floating point addition
- M = the time for the Central Processing Unit to perform 1 multiply
- D = the time for the Central Processing Unit to perform 1 divide
- L = the time for the Central Processing Unit to perform 1 logic operation
- B = the number of characters of I/O in each word
- K_{I1} = the Input transfer rate (characters per second) of the primary I/O system
- K_{O1} = the Output transfer rate (characters per second) of the primary I/O system
- K_{I2} = the Input transfer rate (characters per second) of the secondary I/O system
- K_{O2} = the Output transfer rate (characters per second) of the secondary I/O system
- S_1 = the start time of the primary I/O system not overlapped with compute
- H_1 = the stop time of the primary I/O system not overlapped with compute
- S_2 = the start time of the secondary I/O system not overlapped with compute
- H_2 = the stop time of the secondary I/O system not overlapped with compute
- R_1 = 1 + the fraction of the useful primary I/O time that is required for non-overlap rewind time

SYMBOL	SEMI CONSTANT FACTORS DESCRIPTION	VALUES SCIENTIFIC COMPUTATION	 COMMERCIAL COMPUTATION
WF	the word factor		
	a. fixed word length memory	1	1
	b. variable word length memory	2	2
C_1	weighting factor representing the percentage of the fixed add operations		
	a. computers without index registers or indirect addressing	10	25
	b. computers with index registers or indirect addressing	25	45
C_2	weighting factor that indicates the percentage of floating additions	10	0
C_3	weighting factor that indicates the percentage of multiply operations	6	1
C_4	weighting factor that indicates the percentage of divide operations	2	0
C_5	weighting factor that indicates the percentage of logic operations	72	74
P	percentage of the I/O that uses the primary I/O system		
	a. systems with only a primary I/O system	1.0	1.0
	b. systems with a primary and secondary I/O system	variable	variable
W_{I1}	number of input words per million internal operations using the primary I/O system		
	a. magnetic tape I/O system	20,000	100,000
	b. other I/O systems	2,000	10,000
W_{O1}	number of output words per million internal operations using the primary I/O system	the values are the same as those given above for W_{I1}	
W_{I2}/ W_{O2}	number of input/output words per million internal operations using the secondary I/O system	the values are the same as those given above for W_{I1}	
N	number of times separate data is read into or out of the computer per million operations	4	20
OL_1	overlap factor 1—the fraction of the primary I/O system's time not overlapped with compute		
	a. no overlap—no buffer	1	1
	b. read or write with compute—single buffer	.85	.85
	c. read, write and compute—single buffer	.7	.7
	d. multiple read, write and compute—several buffers	.60	.60
	e. multiple read, write and compute with program interrupt—several buffers	.55	.55
OL_2	overlap factor 2— the fraction of the secondary I/O system's time not overlapped with compute	values are the same as those given above for OL_1, a through e	
i	the exponential memory weighting factor	.5	.333



six months ago we developed the first 2½D stack. everybody else is still six months behind.

10 ... at least

why wait?

Everybody's talking about 2½D stacks. We're delivering them.
Write or phone for the complete story.

EM electronic memories

Memory systems, stacks and cores for commercial, military & space applications
12621 Chadron Avenue, Hawthorne, California, (213) 772-5201

CIRCLE 20 ON READER CARD

Adage makes the best computer on the market for signal processing

...it's called Ambilog 200

Ambilog 200 is a computer which combines the best of both analog and digital techniques. Unlike conventional machines which were intended for manipulating numeric data, Ambilog 200 was designed right from the start for signal processing. Signal processing, which encompasses data acquisition, monitoring, editing, analysis, recording, and display, generally requires efficient handling and high-speed processing of analog as well as digital data. Ambilog 200 with its flexible analog/digital interface and powerful hybrid structure is ideal for such applications.

DIGITIZING AND RECORDING

Multiple inputs, from up to several hundred sources, are routed through multiplexer switches under stored program control. At no penalty in sampling rates over conventional systems, Ambilog 200 converts incoming data to engineering units for recording or monitoring. An analog-to-digital converter performs a complete 14-bit conversion in 4 microseconds for digital storage, recording or outputing.

REAL TIME WAVEFORM MEASUREMENT

Peak values, axis crossings, ratios of successive differences, and other characteristics of analog signals are measured in real time. Incoming signals are monitored for events of interest, using complex programmed detection criteria. In a typical biomedical application where "floods of data" are generated, the result is a 100-to-1 reduction in the bulk of magnetic tape output records.

RANDOM SIGNAL ANALYSIS

Parallel hybrid multiplication and summing, 2 microsecond 30-bit digital storage, and a flexible instruction format providing efficient list processing combine to make Ambilog 200 an extremely powerful tool for statistical signal analysis techniques. These include Fourier transformation, auto and cross correlation, power spectrum density analysis, and generation of histograms of amplitude spectra.

ON-LINE DISPLAY

CRT displays of incoming raw data, or of results derived by reduction and analysis, are generated at frame rates of about 30 per second using line-drawing elements. This "quick look" facility helps the user select those processing techniques which best apply to the problem on hand. Display systems include light pen selection of control parameters and processing subroutines to insure close interaction between the analyst and his computing equipment.

• • • •

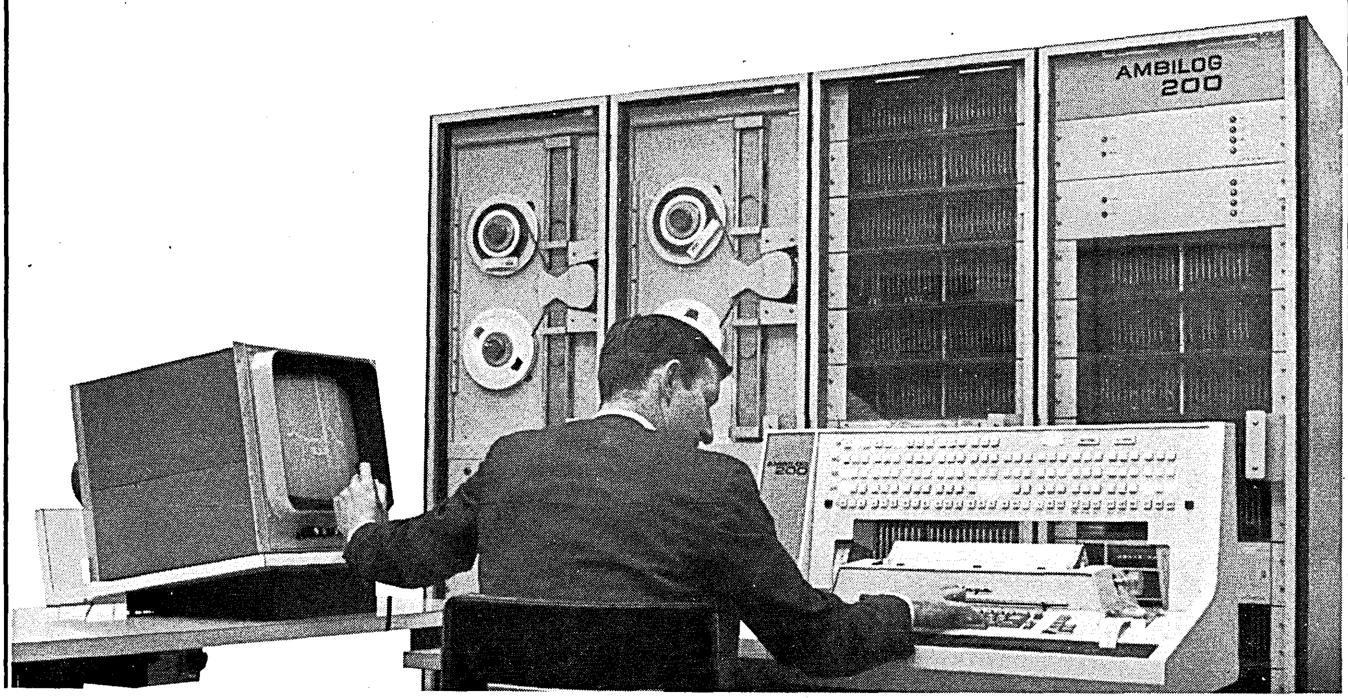
SOFTWARE

Programming aids for use with Ambilog 200 were designed as well to meet the specialized needs of signal processing tasks. An extensive program library includes an Adage assembly system, Fortran, programs for source language editing and on-line debugging and control, and a wide range of applications programs and subroutines. Comprehensive system documentation, and programming and maintenance training and services, are also available.

• • • •

Ambilog 200 signal processing systems are currently being used for seismic research, dynamic structural testing, sonar signal analysis, wind tunnel testing, speech research, and biomedical monitoring. For technical reports describing in detail these installations and other signal processing applications, contact M. I. Stein, Product Manager, Adage, Inc., 1079 Commonwealth Avenue, Boston, Mass. 02215, (617) 783-1100.

Adage INC



CIRCLE 21 ON READER CARD

and commercial computation for the 225 computers we consider are presented in Table 1 (below).

Table I also contains date of introduction for each of the 225 computers we consider. For our study we define the date of introduction as the delivery date of an operating system to the first user. Where the computer is manufactured and used by the same organization, the date of introduction is defined as that when the completed computer passes a minimal acceptance test.

technology curves

Since the functional descriptions consist of two attributes, we can display them on a two-dimensional graph. Fig. 2 (p. 47) contains points obtained when operations/second (P) is plotted against seconds/dollar (C) for computers performing scientific computation. Because of the tremendous range of P and C, Fig. 2 is drawn on log-log graph paper. The number next to each point identifies the corresponding computer as listed in Table I.

From an initial glance at Fig. 2, it is apparent that the early systems generally fall on the lower left portion of the graph, and the newer ones on the upper right. The graph shows how much computing power is obtained at different costs; there are high cost systems (few seconds per dollar) and low cost ones (many seconds per dollar). In any year, an expensive computer has greater computing power (higher number of operations per second) than a less expensive one. It is also apparent from Fig. 2 that for a constant C we obtain greater P over time.

A curve that connects the functional descriptions of the computers in a single year describes the computing tech-

(Cont. p. 49)

Table 1

COMPUTING SYSTEMS

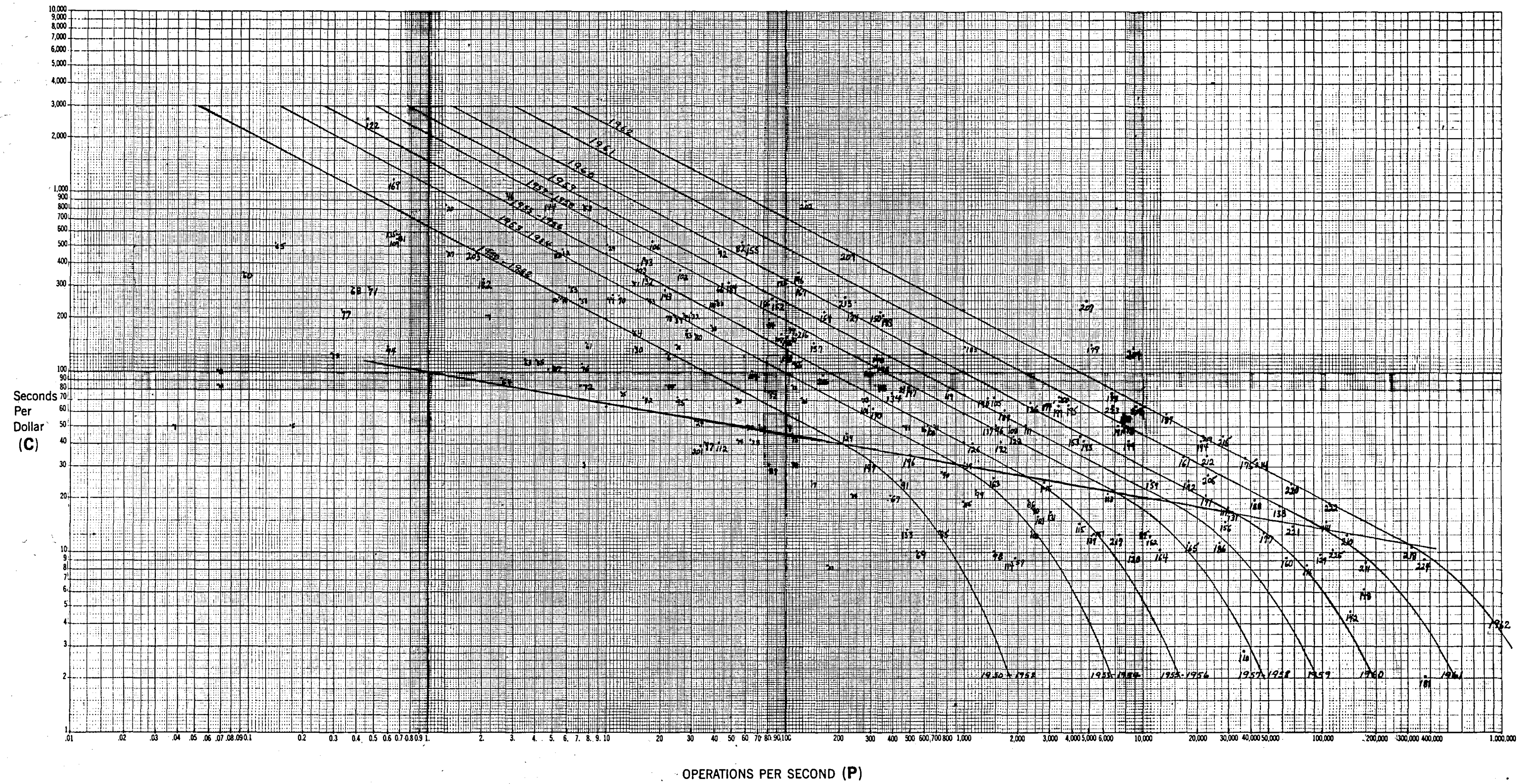
Computer		Date	Scientific Computation		Commercial Computation	
No.	Name	Introduced	P Ops/Sec	C Secs/$	P Ops/Sec	C Secs/$
1	Harvard Mark I	1944	.0379	50.94	0.406	50.94
2	Bell Lab Computer Model IV	March 1945	.0068	509.4	0.035	509.4
3	Eniac	1946	7.448	31.81	44.65	31.81
4	Bell Computer Model V	Late 1947	.0674	84.83	0.296	84.83
5	Harvard Mark II	Sept. 1948	.1712	50.94	0.774	50.94
6	Binac	Aug. 1949	21.75	127.2	11.70	127.2
7	IBM CPC	1949	2.126	207.8	14.37	207.8
8	Bell Computer Model III	1949	.0674	102.2	0.296	102.2
9	SEAC	May 1950	102.8	50.94	253.8	50.94
10	Whirlwind I	Dec. 1950	110.7	31.81	45.57	31.18
11	Univac 1101 Era 1101	Dec. 1950	682.5	50.94	301.8	50.94
12	IBM 607	1950	5.666	479.6	34.06	479.6
13	Avdiac	1950	108.5	84.83	51.20	84.83
14	Adec	Jan. 1951	54.26	42.42	57.16	42.42
15	Burroughs Lab Calculator	Jan. 1951	5.605	254.5	7.718	254.5
16	SWAC	March 1951	632.2	50.94	324.7	50.94
17	Univac I	March 1951	140.1	24.94	271.4	24.94
18	ONR Relay Computer	May 1951	.2937	127.2	1.050	127.2
19	Fairchild Computer	June 1951	2.000	127.2	4.539	127.2
20	National 102	Jan. 1952	1.260	848.3	2.998	848.3
21	IAS	March 1952	467.0	84.83	305.0	84.83
22	Maniac I	March 1952	302.7	101.9	163.4	101.9
23	Ordvac	March 1952	268.8	72.76	127.8	72.76
24	Edvac	April 1952	31.56	54.22	14.86	54.22
25	Teleregister Special Purpose Digital Data Handling	June 1952	12.16	78.93	26.43	78.93
26	Illiac	Sept. 1952	123.1	72.76	50.43	72.76
27	Elcom 100	Dec. 1952	1.278	424.2	3.241	424.2
28	Harvard Mark IV	1952	63.99	42.42	64.95	42.42
29	Alwac II	Feb. 1953	10.17	509.4	12.08	509.4
30	Logistics Era	March 1953	52.85	72.00	39.01	72.0
31	Oarac	April 1953	24.38	141.4	35.71	141.4
32	ABC	May 1953	29.88	212.1	11.66	212.1
33	Raydac	July 1953	171.3	8.483	244.6	8.483
34	Whirlwind II	July 1953	233.4	21.21	95.96	21.21
35	National 102A	Summer 1953	4.089	116.5	8.400	116.5
36	Consolidated Eng. Corp. Model 36-101	Summer 1953	38.31	181.8	21.07	181.8
37	Jaincomp C	Aug. 1953	4.745	103.9	3.375	103.9
38	Flac	Sept. 1953	61.55	50.94	107.9	50.94
39	Oracle	Sept. 1953	1002.	31.81	563.4	31.81
40	Univac 1103	Sept. 1953	749.0	28.34	666.2	28.34
41	Univac 1102	Dec. 1953	460.3	50.94	240.0	50.94
42	Udec I	Dec. 1953	16.38	72.67	21.93	72.67
43	NCR 107	1953	16.99	254.5	34.44	254.5
44	Miniac	Dec. 1953	10.91	267.6	9.545	267.6
45	IBM 701	1953	992.7	18.34	615.7	18.34
46	IBM 604	1953	2.766	974.2	20.19	974.3
47	AN/UJQ-2(YA-1)	1953	21.48	84.83	56.16	84.83
48	Johnniac	March 1954	319.2	84.83	284.9	84.83
49	Dyseac	April 1954	72.18	50.90	172.4	50.90
50	Elcom 120	May 1954	5.471	261.9	6.456	262.0
51	Circle	June 1954	14.04	318.1	10.59	318.1
52	Burroughs 204 & 205	July 1954	80.84	77.94	187.3	77.94
53	Modac 5014	July 1954	6.238	299.8	10.09	299.8
54	Ordfiac	July 1954	2.607	92.51	6.011	92.51
55	Datatron	Aug. 1954	113.7	113.2	243.1	113.2
56	Modac 404	Sept. 1954	7.116	254.5	15.29	254.5
57	Lincoln Memory Test	Dec. 1954	1925.	9.285	768.7	9.285
58	TIM II	Dec. 1954	7.414	848.3	7.439	848.3
59	Caldic	1954	23.99	203.8	41.34	203.8
60	Univac 60 & 120	Nov. 1954	.0924	356.3	1.473	356.3
61	IBM 650	Nov. 1954	110.8	155.9	291.1	155.9
62	WISC	1954	7.736	145.7	6.413	145.7
63	NCR 303	1954	3.491	117.6	8.281	117.6
64	Mellon Inst. Digital Computer	1954	14.23	169.9	10.55	169.9
65	IBM 610	1954	.1408	519.6	0.437	519.6
66	Alwac III	1954	44.80	302.7	91.42	302.7
67	IBM 702	Feb. 1955	394.4	20.78	1063.	20.78
68	Monrobot III	Feb. 1955	.3743	299.8	1.188	299.8
69	Norc	Feb. 1955	545.8	10.17	268.2	10.17
70	Miniac II	March 1955	11.76	267.6	17.44	267.6
71	Monrobot V	March 1955	.4678	295.5	1.607	295.5
72	Udec II	Oct. 1955	7.244	84.83	10.65	84.83
73	RCA BIZMAC I & II	Nov. 1955	285.6	5.668	967.9	5.668
74	Pennstac	Nov. 1955	26.75	212.1	22.98	212.1
75	Technitral 180	1955	110.0	46.19	190.1	46.19
76	National 102D	1955	7.317	112.3	14.20	112.3
77	Monrobot VI	1955	.3293	222.7	0.966	222.7
78	Modac 410	1955	24.18	203.8	51.84	169.9
79	Midac	1955	101.6	169.9	29.00	169.9
80	Elcom 125	1955	31.24	164.1	29.01	164.1
81	Burroughs E 101	1955	.6898	580.0	2.319	580.0
82	Bendix G15	Aug. 1955	57.34	419.9	30.25	419.9
83	Alwac III E	Nov. 1955	41.50	249.4	90.15	249.4
84	Readix	Feb. 1956	80.63	194.9	87.99	194.9
85	IBM 705, I, II	March 1956	734.0	13.27	2087.	13.27
86	Univac 1103 A	March 1956	2295.	19.49	1460.	19.49
87	AF CRC	April 1956	81.66	31.81	28.97	31.81
88	Guidance Function	April 1956	5.246	461.9	7.744	461.9
89	IBM 704	April 1956	10,670.	13.18	3,785.	13.18
90	IBM 701 (CORE)	1956	2378.	17.81	1807.	17.81
91	Narec	July 1956	444.8	25.45	190.6	25.45

(Table 1 cont. p. 46)

Table 1 (Cont.)

Computer		Date	Scientific Computation		Commercial Computation	
No.	Name	Introduced	P Ops/Sec	C Secs/$	P Ops/Sec	C Secs/$
92	LGP 30	Sept. 1956	41.94	479.6	32.75	479.6
93	Modac 414	Oct. 1956	28.26	169.9	42.94	169.9
94	Elecom 50	1956	.5990	139.2	1.776	1039.
95	Udec III	March 1957	25.11	72.76	20.85	72.76
96	George I	Sept. 1957	1538.	50.94	571.9	50.94
97	Univac File O	Sept. 1957	35.20	41.02	73.17	41.02
98	Lincoln TXO	Fall 1957	1,471.	10.19	359.6	10.19
99	Univac II	Nov. 1957	1,155.	22.27	2,363.	22.27
100	IBM 705 III	Late 1957	2,379.	13.27	7,473.	13.27
101	Teleregister Telefile	Late 1957	286.0	65.98	935.9	65.98
102	Recomp I	Late 1957	25.76	363.8	16.14	363.8
103	IBM 608	1957	15.21	389.7	60.69	389.7
104	Mistic	1957	64.28	101.9	24.50	101.9
105	Maniac II	1957	1,491.	72.84	1,421.	72.84
106	IBM 609	1957	18.19	530.7	75.21	530.7
107	IBM 305	Dec. 1957	94.47	163.0	96.47	163.0
108	Corbin	1957	1,794.	50.90	2407.	50.90
109	Burroughs E 103	1957	.6736	551.8	2.286	551.8
110	AN/FSQ 7 & 8	1957	36,730.	2.834	15,560.	2.834
111	Alwac 880	1957	2,198.	50.90	959.7	50.90
112	Univac File I	Jan. 1958	42.49	41.05	92.04	41.05
113	Lincoln CG24	May 1958	6,394	21.21	5,933.	21.21
114	IBM 709	Aug. 1958	1,869.	8.882	10,230.	8.882
115	Univac 1105	Sept. 1958	4,433.	14.50	5,527.	14.50
116	Lincoln TX2	Fall 1958	82,050.	8.483	34,000.	8.483
117	Philco 2000-210	Nov. 1958	29,970.	17.81	28,740.	17.81
118	Recomp II	Dec. 1958	41.36	249.4	28.03	249.4
119	Burroughs 220	Dec. 1958	810.2	79.94	1,616.	79.94
120	Mobidic	1958-1960	8741.	10.19	12,250.	10.19
121	Philco CXPO	1958	2,622.	15.91	1,576.	15.91
122	Monrobot IX	1958	.4598	2,545.	1.334	2,545.
123	GE 210	June 1959	1,884.	44.54	5,085.	44.54
124	Cyclone	July 1959	234.6	215.0	119.6	215.0
125	IBM 1620	Oct. 1959	94.79	331.7	47.20	331.7
126	NCR 304	Nov. 1959	1,136.	40.23	2,445.	40.23
127	IBM 7090	Nov. 1959	97,350.	9.742	45,470.	9.742
128	RCA 501	Nov. 1959	638.7	38.97	1,877.	38.97
129	RW 300	Nov. 1959	218.6	45.58	534.3	45.78
130	RPC 9000	1959	14.50	138.6	9.521.	138.6
131	Librascope Air Traffic	1959	3043.	16.94	6,130.	16.94
132	Jukebox	1959	16.56	338.9	18.66	338.9
133	Datamatic 1000	1959	480.8	13.44	1,455.	13.44
134	CCC Real Time	1959	393.8	77.17	280.3	77.17
135	Burroughs E 102	1959	.6670	580.0	1.847	580.0
136	Burroughs D 204	1959	2,354.	68.00	1,183.	68.00
137	AN/TYK 6V BASICPAC	1959	1,365.	50.90	493.0	50.90
138	CDC 1604	Jan. 1960	58,290.	18.34	20,390.	18.34
139	Librascope 3000	Jan. 1960	5,177.	12.47	25,320.	12.47
140	Univac Solid State 80/90 I	Jan. 1960	329.1	124.7	489.6	124.7
141	Philco 2000-211	March 1960	105,844.	14.845	55,740.	14.85
142	Univac Larc	May 1960	142,600.	4.619	40,450.	4.619
143	Libratrol 500	May 1960	21.07	286.0	20.38	286.0
144	Monrobot XI	May 1960	4.839	890.7	10.30	890.7
145	IBM 7070	June 1960	2,813.	23.98	5,139.	23.98
146	CDC 160	July 1960	119.3	354.3	49.63	354.2
147	IBM 1401 (Mag. Tape)	Sept. 1960	496.7	83.14	1,626.	83.14
148	AN/FSQ 31 & 32	Sept. 1960	172,200.	6.235.	48,360.	6.285
149	Merlin	Sept. 1960	8,306.	42.42	2,925.	42.42
150	IBM 1401 (Card)	Sept. 1960	340.9	215.0	967.8	215.0
151	Mobidic B	Fall 1960	5,251.	12.72	8,630.	12.72
152	RPC 4000	Nov. 1960	89.91	249.4	54.11	249.4
153	PDP-1 (M.T.)	Nov. 1960	4,455.	41.57	2,173.3	41.6
154	PDP-1 (P.T.)	Nov. 1960	166.6	215.	57.16	215.0
155	Packard Bell 250 (PT)	Dec. 1960	62.23	506.9	22.21	506.9
156	Honeywell 800	Dec. 1960	28,790.	14.85	23,760.	14.85
157	General Mills AD/ECW-57	Dec. 1960	143.9	141.7	44.03	141.7
158	Philco 3000	Late 1960	102.2	155.9	66.13	155.8
159	Maniac III	Late 1960	11,140.	25.45	4723.	25.45
160	Sylvania S9400	Late 1960	62,510.	9.306	49.550.	9.306
161	Target Intercept	Late 1960	16,800.	33.89	16,070.	33.89
162	Westinghouse Airbourne	1960	10,950.	12.47	4806.	12.47
163	RCA 300	1960	1,466.	25.98	687.7	25.98
164	Mobidic CD & 7A AN/MYK	1960	12,410.	10.39	15,430.	10.39
165	Litton C7000	1960	18,200.	11.34	5,323.	11.34
166	Libratrol 1000	1960	84.16	254.5	50.85	254.5
167	GE 312	1960	122.0	299.8	47.12	299.8
168	Diana	1960	102.1	127.2	48.85	127.2
169	DE 60	Feb. 1960	.6384	1,155.	1.855	1155.
170	Burroughs D107	1960	311.8	63.62	73.95	63.62
171	AN/USQ 20	1960	22,390.	20.78	23,670.	20.78
172	AN/TYK 4V Compac	1960	1,610.	41.57	616.1	41.57
173	General Mills Apsac	Jan. 1961	16.22	424.2	7.084	424.2
174	Univac Solid State 80/90 II	Jan. 1961	3,199.	69.28	3,044.	69.28
175	Bendix G20 & 21	Feb. 1961	37,260.	33.17	17,060.	33.17
176	RCA 301	Feb. 1961	323.0	113.4	1,055.	113.4
177	BRLESC	March 1961	47,240.	12.72	28,550.	12.72
178	GE 225	March 1961	6,566	77.94	7,131.	77.94
179	CCC-DDP 19 (Card)	May 1961	5,159.	138.6	3,027.	138.6
180	CCC-DDP 19 (MT)	May 1961	7,908.	59.38	8,073.	59.38
181	IBM Stretch (7030)	May 1961	371,700.	2.078	631,200.	2.078
182	NCR 390	May 1961	2.034	328.2	10.43	328.2
183	Honeywell 290	June 1961	354.3	207.8	182.8	207.8
184	Recomp III	June 1961	48.28	311.8	35.76	311.8
185	CDC 160A	July 1961	1,015.	138.6	1,780.	138.6
186	IBM 7080	Aug. 1961	27,090.	11.34	30,860.	11.34
187	RW 530	Aug. 1961	13,460.	59.38	5086.	59.38
188	IBM 7074	Nov. 1961	41,990.	19.49	31,650.	19.49
189	IBM 1410	Nov. 1961	1,673.	62.35	4,638.	62.35
190	Honeywell 400	Dec. 1961	1,354.	71.67	2,752.	71.67
191	Rice Univ.	Dec. 1961	7,295.	50.90	2378.	50.90
192	Univac 490	Dec. 1961	17,770.	24.94	15,050.	24.94
193	AN/TYK 7V	1961	4,713.	41.57	9,077.	41.57
194	Univac 1206	1961	20,990.	42.42	17,700.	42.42
195	Univac 1000 & 1020	1961	3,861.	66.33	3,292.	66.33
196	ITT Bank Loan Process	1961	492.6	34.64	1,916.	34.64
197	George II	1961	298.	31.81	675.1	31.81
198	Oklahoma Univ.	Early 1962	7,723.	50.90	2,616.	50.90
199	NCR 315	Jan. 1962	3,408.	65.63	11,460.	65.63
200	NCR 315 CRAM	Jan. 1962	3,364.	73.36	9,896.	73.36
201	Univac File II	Jan. 1962	33.46	38.97	94.49	38.97
202	HRB-Singer Sema	Jan. 1962	129.2	890.7	56.94	890.7
203	Univac 1004	Feb. 1962	1.789	479.6	25.29	479.6
204	ASI 210	April 1962	8,868.	135.5	4,114.	135.5
205	Univac III	June 1962	22,720.	27.11	22,790.	27.11
206	Burroughs B200 Series-B270 & 280	July 1962	163.3	95.93	615.3	95.93
207	SDS 910	Aug. 1962	4,841.	249.4	2,355.	249.4
208	SDS 920	Sept. 1962	9,244	65.63	4,964	65.63
209	PDP-4	Sept. 1962	220.2	479.6	75.97	479.6
210	Univac 1107	Oct. 1962	138,700.	12.47	76,050.	12.47
211	IBM 7094	Nov. 1962	175,900.	8.782	95,900.	8.781
212	IBM 7072	Nov. 1963	22,710.	34.64	8,694.	34.64
213	IBM 1620 MOD III	Dec. 1962	214.8	259.8	56.89	259.8
214	Burroughs B5000	Dec. 1962	43,000.	32.82	15,910.	32.82
215	ASI 420	Dec. 1962	27,790.	44.54	11,090.	44.54
216	Burroughs B200 Series—Card Sys	Dec. 1962	114.3	160.1	437.2	164.1
217	RW 400 (AN/FSQ 27)	1962	7,437.	12.47	11,240.	12.47
218	CDC 3600	June 1963	315,900.	11.34	74,900.	11.34
219	IBM 7040	April 1963	21,420.	44.54	90.79	44.54
220	IBM 7044	July 1963	67,660.	23.98	23,420.	23.98
221	RCA 601	Jan. 1963	68,690.	13.86	58,880.	13.86
222	Honeywell 1800	Nov. 1963	110,600.	17.81	57,750.	17.81
223	Philco 1000 Transac S1000	June 1963	6,811.	65.63	10,440.	65.63
224	Philco 2000-212	Feb. 1963	369,800.	9.169	84,230.	9.169
225	Librascope L 3055	Dec. 1963	114,000.	10.39	30,620	10.39

Fig. 2 Functional Descriptions for Scientific Computers (The numbers in this graph to identify each computer correspond to the numbers in Table I.)



nology for that year. Improved performance consists of a continual shift over time, enabling an increased number of operations per second to be performed for a given cost.

We now wish to use our data to develop the technology curves. Unfortunately, the points for a particular year in Fig. 2 do not form smooth parallel curves. For any one year considerable scattering occurs because (1) not all systems are equally technically advanced, and (2) there are errors in the estimates of P and C.

The first reason for the scatter of points needs little explanation. In the computing industry, there have been many systems introduced, and these have resulted in a wide range of performance from improved to poorer. Some systems will make significant improvements and fall far to the right of the other points. Alternatively, many systems will not match the capabilities of existing computers and will lie in the range of the industry's previous know-how.

The second reason for the scatter is the expected variance in the estimates of the functional descriptions. P was obtained by means of the functional model, which estimated each system's actual performance. There are differences both in the pricing policies of the manufacturers and in our ability to determine what equipment constitutes each particular system that creates a variance in C. In the calculations we performed, many small errors could have crept into the estimates of P and C to produce random error, even if all the systems came from an identical level of technological knowledge.

Recognizing that variance exists, it is necessary to use a curve-fitting technique to estimate the desired technology lines. For this study we have used least square regression analysis. From a visual analysis of Fig. 2 it appears that the technology curves for the different years are approximately the same in shape, with a shift to the right over time. Thus, the data were fitted to the following equation:

$$\ln(C) = \alpha_0 + \alpha_1 \ln(P) + \alpha_2[\ln(P)]^2 + \beta_1 S_1 + \beta_2 S_2 + \ldots + \beta_7 S_7 \qquad \text{Eq. 1}$$

The α's and β's represent the regression coefficients to be determined by the least squares analysis. The $S_1, \ldots, S_7$ represent dummy variables (or shift parameters) for the different years considered. To fit the curve, the data were grouped into eight time periods (i.e., 1962, 1961, 1960, 1959, 1957-58, 1955-56, 1953-54, and 1950-51-52). The earlier years were combined because of the small number of systems introduced in each of these years. The dummy variables were used in the following manner: for 1962, $S_1, \ldots, S_2$ were all set equal to 0; for 1961, $S_1 = 1$ and $S_2 = S_3 = \ldots = S_7 = 0$; for 1960, $S_2 = 1$ and $S_1 = S_3 = \ldots = S_7 = 0$; . . . and finally for 1950-51-52, $S_7 = 1$ and $S_1 - S_2 = \ldots = S_6 = 0$. (P) and $[\ln(P)]^2$ were both initially included in the equation since a visual analysis of the lines made them appear curved.

After the initial regression estimate, all points that were more than ½ a standard deviation below and to the left of the curve for their year were omitted. Eliminating points in this manner provides a distinct procedure for determining which points we will include in the final determination of technology curves, and forces the technology curves to the right to provide a more accurate picture of the performance limits.

The regression analysis, using the data for computer performance in scientific computation with Equation 1, showed that $[\ln(P)]^2$ term was not significant. The least squares technique was then used to fit Equation 2 to the data.

$$\ln(C) = \alpha_0 + \alpha_1 \ln(P) + \beta_1 S_1 + \beta_2 S_2 + \ldots + \beta_7 S_7 \qquad \text{Eq. 2}$$

For the linear equation, all the terms were significant and the correlation coefficient was r = +.9569. Since the correlation coefficient equaled only +.9596 with Equation 1, it appeared most reasonable to use the simpler linear equation to plot the technology curves. In the calculation of both the polynomial and the linear equations, over 120 observations were used so that the sample sizes would be adequate. The equation for the scientific computation technology curves is as follows:

$$\ln(C) = 8.9704 - .51934\,[\ln(P)] \qquad \text{Eq. 3}$$

−.3650 (1961)	−1.6639 (1955-56)
−.7874 (1960)	−1.9859 (1953-54)
−1.0724 (1959)	−2.5013 (1950-51-52)
−1.3028 (1957-58)	

The eight curves described by Equation 3 are drawn in Fig. 2.

We now perform a similar analysis for commercial computation. The results of the calculation of the technology curves for systems performing commercial computation are shown in Equation 4.[3]

$$\ln(C) = 8.1672 - .459\,[\ln(P)] \qquad \text{Eq. 4}$$

−.3643 (1961)	−1.187 (1955-56)
−.6294 (1960)	−1.454 (1953-54)
−.8561 (1959)	−2.164 (1950-51-52)
−.9011 (1957-58)	

The eight curves drawn from Equation 4 are shown in Figure 3 (p. 51).

grosch's law upheld

We analyze the meaning of the technology curves by first restating the general equation for the curves:

$$C = (\alpha_0)\,(P)^{\alpha 1}\,(e^{\beta 1})\,(e^{\beta 2}) \ldots (e^{\beta 7}) \qquad \text{Eq. 5}$$

where $\ln \alpha_0 = \alpha_0$

and $\alpha_0, \alpha_1, \beta^1, \beta_2, \ldots, \beta_7$ are the values calculated with the least square regression analysis.

From Equation 5 we obtain the following:

$$\text{seconds/dollar} = k \left(\text{Shift parameter to adust for year}\right) (\text{operations/sec})^{\alpha 1} \qquad \text{Eq. 6}$$

For any particular year we can combine the constant, k, and the shift parameter into a new constant C [(k)] x (shift parameter) = C. If we, therefore, set $a_1 = -\alpha_1$, Equation 6 now becomes:

$$\text{dollars/second} = \frac{1}{C} (\text{operations/sec})^{a 1}. \qquad \text{Eq. 7}$$

For scientific computation the value for $\alpha 1 = -.519$ so that a_1 equals .519. For commercial computation $\alpha_1 = -.459$ so that a_1 equals .459. We can therefore assume that a_1 is *(Cont. p. 54)*

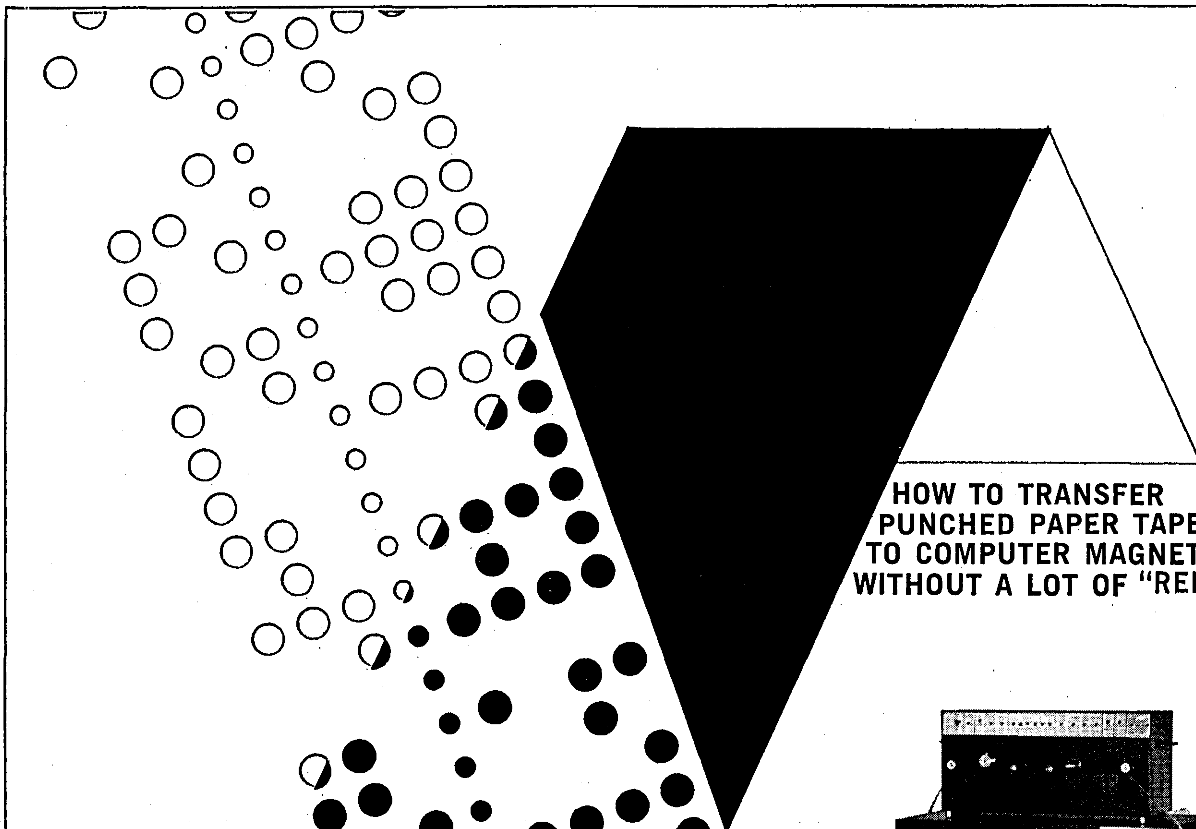
[2] For this equation the following list contains the standard error and the test of significance (student's t test) for each regression coefficient.

Regression Coefficient	Standard Error	t value
$\alpha 1$	.0171	−30.41
		−2.112
β_2	.1608	−4.897
β_3	.1887	−5.682
β_4	.1687	−7.723
β_5	.1992	−8.349
β_6	.1750	−11.34
β_7	.1943	−12.87

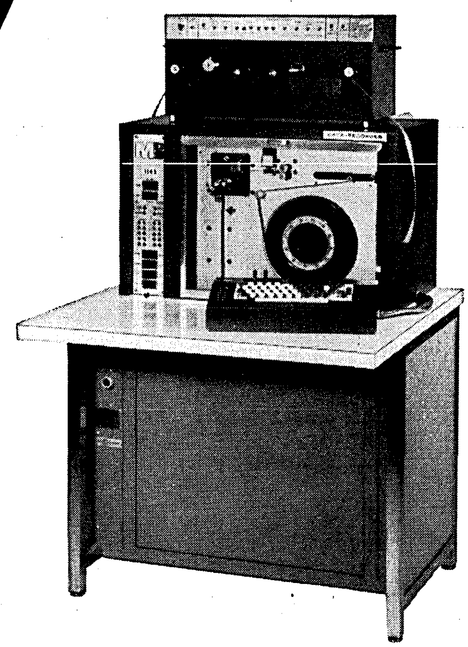
[3] The correlation coefficient, r, for the linear equation (Equation 2) was +.8543. The curves using Equation 1 and Equation 2 were similar and the correlation coefficients almost equal so that the simple linear equation was used to construct the technology curves.

[4] The following list contains the standard error and the test of significance (students' t test) for each regression coefficient in this equation.

Regression Coefficient	Standard Error	t value
α_1	.02983	−15.39
β_1	.2589	−1.407
β_2	.2758	−2.282
β_3	.2895	−2.957
β_4	.2537	−3.551
β_5	3029	−3.917
β_6	.2789	−5.214
β_7	.3109	−6.901



HOW TO TRANSFER PUNCHED PAPER TAPE DATA TO COMPUTER MAGNETIC TAPE... WITHOUT A LOT OF "RED TAPE"



1105 PTR DATA-RECORDER SYSTEM

Soon after the MDS 1101 DATA-RECORDER was introduced, DP executives began to comment: **"Why don't you give us a combination of the 1101 and a Punched Paper Tape Reader? Something that transfers data from punched paper tape direct to computer magnetic tape."**

We can't always promise this "push-button service", but here's what you've asked for... the MDS 1105 PTR DATA-RECORDER SYSTEM!

It's a modified 1101 DATA-RECORDER linked with a 3005 Paper Tape Strip Reader. **When not in use in the system, the DATA-RECORDER can be operated the same as the 1101... for transcribing data from source documents direct to computer magnetic, tape, and verifying.**

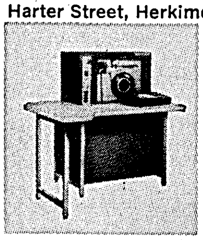
Transferred paper tape data will be untranslated copy... paper tape codes are transferred bit-for-bit, without change, to the magnetic tape. Except for the channel 8 punch, and parity bit punch, each paper tape data hole becomes a bit on the magnetic tape.

The 1105 system accepts punched paper tape recorded in odd or even parity, or without parity. It will read up to 8,000 characters per minute, in Continuous Read/Write mode... from 5, 7 or 8 channel paper tape. During **Interrupt** Read/Write Mode, data can be manually entered and added to the data read from the paper tape.

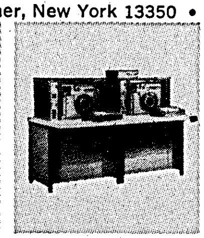
Interested? Tell us your application. We'll send more complete details, tell you how the 1105 PTR DATA-RECORDER SYSTEM can cut your "RED TAPE" problems.

MOHAWK DATA SCIENCES CORPORATION
Harter Street, Herkimer, New York 13350 • Telephone 315/866-6800

SALES-SERVICE OFFICES IN MAJOR MARKET AREAS

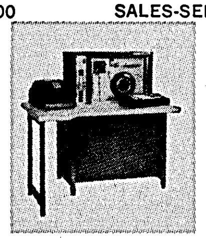


MDS 1101
Keyed
DATA-RECORDER



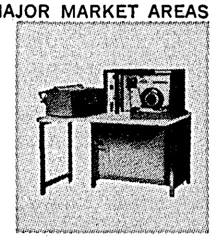
MDS 1102MTP
Multi Tape Pooler
DATA-RECORDER

MDS 1103LDC
DATA-RECORDER
Long Dist. Comm.



MDS 1104AMC
DATA-RECORDER
Adding Mach. Control

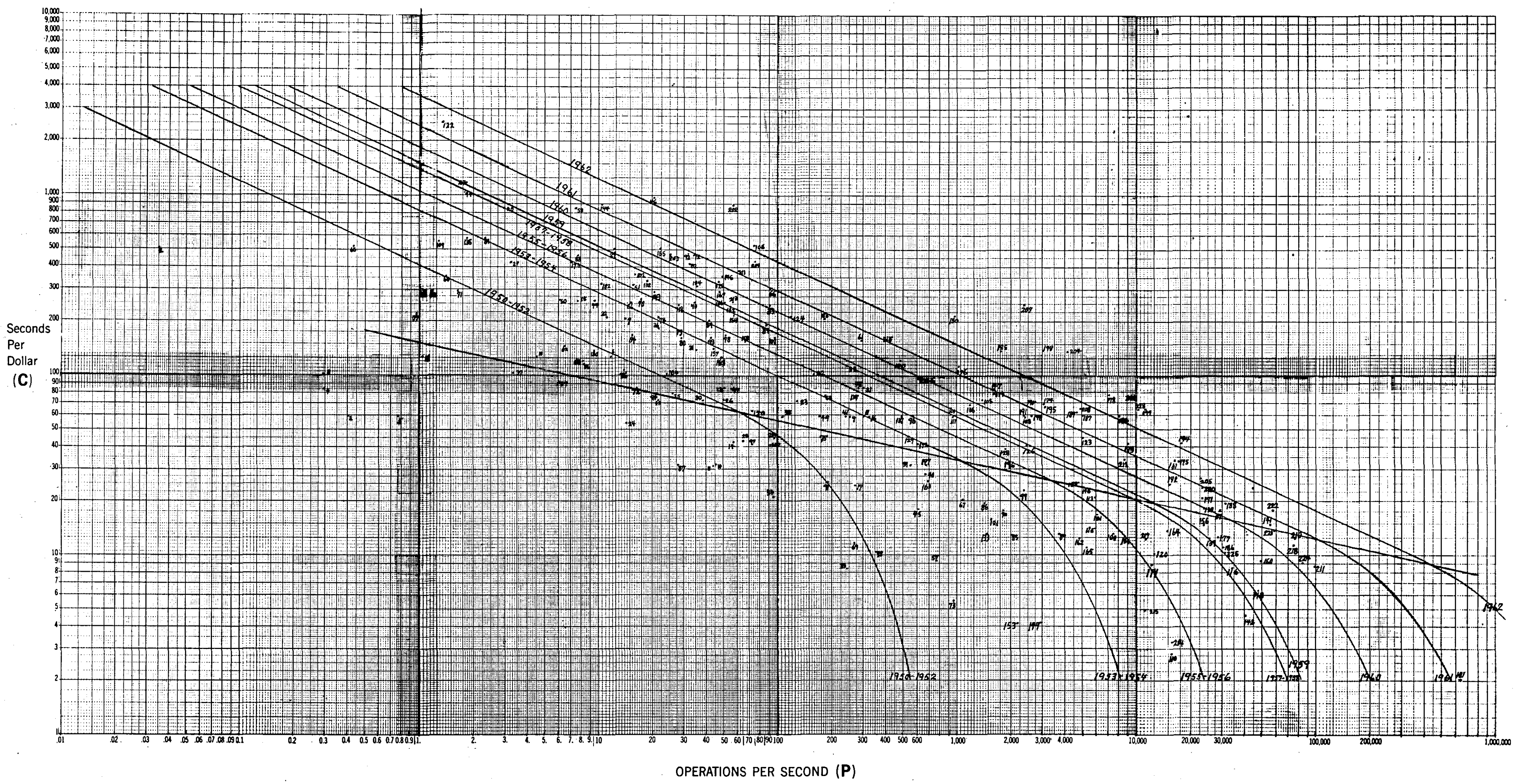
MDS 1106PCR
DATA-RECORDER
Punched Card Reader



MDS 1108PRD
DATA-RECORDER
with Printer

CIRCLE 23 ON READER CARD

Fig. 3 Functional Descriptions for Commercial Computation (The numbers used in this graph to identify each computer correspond to the numbers in Table I.)



approximately equal to .5 and rewrite Equation 7 as follows:

$$\text{System Cost} = \frac{1}{C}\sqrt[2]{\text{Computing Power}} \qquad \text{Eq. 8}$$

This represents a very interesting result because it indicates that within the limits of the computing technology one can construct four times as powerful a computer at only twice the cost.

$$\text{Computing power} = (C \text{ system cost})^2 \qquad \text{Eq. 9}$$

That computing power increases as the square of cost was proposed in the late 1940's by Herb Grosch. Since that time the relationship expressed in Equation 9 has been referred to as Grosch's Law. We have seen the industry develop a sense of humor over its 20-year life with frequent jokes being made in reference to Grosch's Law. In a recent article by Charles W. Adams, "Grosch's Law Repealed," the author proposes to "replace the square (Grosch's Law) by the square root."[5] Grosch's Law has received much attention because of its implications about economies of scale, yet has never been supported with adequate quantitative data. We still need to question whether the Law (Computing power=Constant $(\text{Cost})^2$) is true, or if it is the artifact of the computer companies' pricing policy. The popularity of the Law and the difficulty in setting prices leads us to suspect the possibility of some bias in our data.

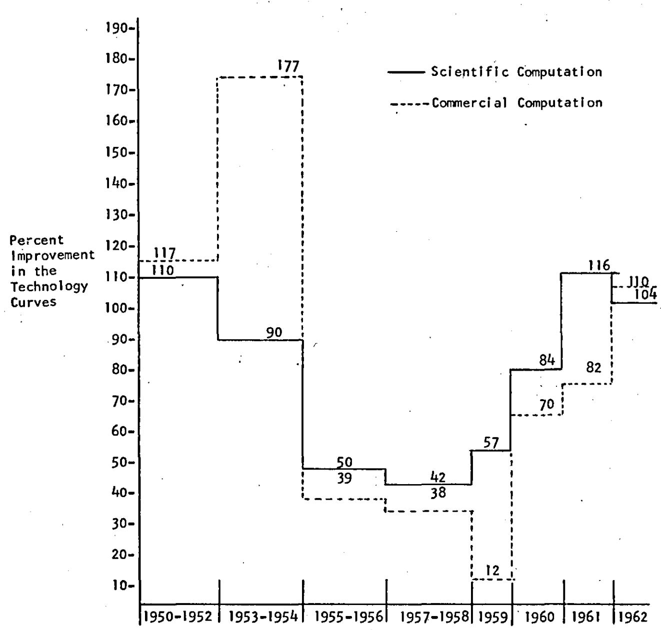
We must express another word of caution before we attach too much significance to Grosch's Law. In calculating the technology curves we were able to use the systems actually built. The equations derived are, therefore, applicable within the limited range of computers studied. Special consideration has to be given to the fact that there are definite limits to the maximum computing power that can be obtained at any one time. As the bounds of technological knowledge are reached, additional computing power is purchased at a very high price. For high value of P the technology curve will not remain a straight line but will curve downward to show an ever increasing negative slope. The reason that this did not show up in the regression analysis is that only a few computers came close to the maximum limits of computing power. Three noticeable ones are the AN/FSQ 7 and 8 (the Sage computers), the Univac Larc and the IBM Stretch. These computers each obtained a new high evaluation for absolute computing power, but at considerably lower number of operations/dollar. Grosch's Law did not hold for these three machines because the increases in power were obtained at less than the squared, or even a 1 to 1, relationship with Cost—the slope of the curve, or a^1, is less than −1. We cannot build larger and larger computers at reasonable costs since at any point in time there are absolute limits to the size and speed obtainable. This fact needs to be kept in mind when talking about Equation 8. The most powerful computing systems we could possibly build today or tomorrow would not be the most economical.

In order to estimate where the turning point occurs we use the computers that have had, at one time, the maximum absolute efficiency. For scientific computation there are eight systems, and for commercial computation ten. We add to Fig. 2 and 3 lines of maximum efficiency through these points. The point where the line crosses the technology curves for each year provides an estimate of where the technology curves start to slope downward to yield diminishing marginal returns for systems with greater computing power. The latter curves are drawn freehand on Fig. 2 and 3 to show their approximate shape.

performance improvements from 1950 to 1962

The continual stream of performance improvements appears to result from the dynamic nature of the industry itself. Most people in the computing field search conscientiously for faster and more economical machines. However, most of these individuals have a limited idea of what has happened over the past 20 years. For instance, they greatly underestimate the number of innovative systems produced and the amount of performance improvement which actually has taken place. The shift in the technology curves illustrates the performance advances. From 1950 through 1962 the technology curves have an average improvement of 81% per year for scientific computation and 87% per year for commercial computation. It is seen from Fig. 4 that there has been some variance in yearly percent improvement. The improvements in both scientific and commercial computation have been fairly similar, with the first five years, 1950-1954, and the last three years, 1960-1962, showing greater improvement than the years, 1955-1959. The large commercial computation improvement in 1953-1954 that we mentioned earlier as being significantly above the mean, can be explained by the great increase in speed and the number of machines using magnetic tape units. Since commercial computation relies more upon input-output capability than does scientific computation, the improvements and increased utilization of magnetic tapes aided this category more than they did the other. No simple explanation has been found for the other variations shown in Fig. 4.

Figure 4.—Average Yearly Shift of the Technology Curves



As a result of tremendous improvement from year to year, a computer has been marketable for from 3 to 6 years. With the great rate of improvement, by the time most users get around to purchasing a system it is greatly inferior to the newer ones being introduced. This illustrates the tremendous obsolescence problem the industry must face if the present rate of improvement continues. The problem will become especially acute if purchasers try to order machines now being designed for delivery one or two years away, rather than take an existing machine. Most computers already in production require from six months to two years for delivery. ■

[5] C. W. Adams, "Grosch's Law Repealed," *Datamation* (July 1962), pp. 38-39.

In-house reprogramming is simple. All it takes is time and money and time and money and time and money and time and money

by Frank Wagner

Converting an installation from one computer to another always looks like the ideal do-it-yourself job. After all, the programs are all written, debugged, and working. The equipment is compatible. And, being a wise manager, you further eschew trouble by decreeing that the new programs shall be identical copies of the old. The conversion process thus appears to be purely mechanical, and you assign your youngest programmer to it (for reasons of morale and efficiency) with a clear conscience. But don't settle back too far in your easy chair. You may encounter some problems which will have you reaching for the aspirin bottle, and yelling "Why didn't someone tell me?" O.K., we'll tell you.

SELF-INFLICTED TORTURES

You've no doubt already guarded against *undocumented programs.* So, the first real problem you'll have to contend with is the *bad write-up.* The written specifications do not agree with the program, and you find out only when the rewritten routine is checked out with live data. By the time the discrepancies have been corrected, your cost and schedule have slipped by 20%. Or, you may not be so lucky and miss by 50-75%. Then there are *machine-dependent routines.* These are written by "clever" programmers who love to use illegal instructions. Many a channel program depends on the subtle timing considerations of a particular machine. The poor neophyte, assigned to reprogram such a routine, will often wind up wondering just what it is that the program is really supposed to do. More time and money lost. Worse yet, however, are *software-dependent* programs. A relatively simple case to find and fix is exemplified by the "standard" arithmetic routine which gives slightly different results than the one it replaces. Much harder to track down are the instances where all the installation programs have been written to negate the effect of some obscure bug in the overall operating system. Most likely no one could find the error four years ago, so everybody stopped trying, and finally its very existence was forgotten.

PROMISES, PROMISES...

The new equipment presents its own problems. True, the manufacturers sing an enticing song: "Our hardware is compatible... our software is compatible... our manuals are compatible... our salesmen are compatible..." ad nauseum. But there are degrees of compatibility ranging from the exact match to the totally alien. The trick is to understand the differences, anticipate and appraise the problems, and assign to each its proper degree of importance. For instance, just where are the syntax differences between OS/360 F-level COBOL, BOS COBOL, and CDC COBOL 63? And how do they all relate to COBOL 61? Are the differences easy to cope with, or is the job three times what you expect? There are hardware differences too. "Our tapes are compatible" cry the compatible salesmen. But on what level? Character codes? Binary? Only if you ignore parity checks? Or only if you don't try to read or write records of less than three characters? And what about those tapes you got from an outside source, written by write amplifiers with different tolerances? You have learned how to run them on your old equipment. Are the new tape units all *that* compatible? Listen to the promises, but remember the gal who said "My husband and I are perfectly compatible. He likes to stay out late with the boys, and so do I."

ENTER THE TINKERERS

Finally, in spite of all your admonitions to keep the new programs identical to the old, there's always a temptation to make a few (just a few) improvements. The programmer's very soul revolts at the idea of duplicating some obviously poor characteristics. ("It's really easier to change than to copy"). The users, too, will ache to get rid of some real or imagined inconvenience, or to add a bell here and a whistle there. At Informatics we have a standing offer of reward for the first evidence of a truly identical reprogramming job ever done in-house. There's only one sure way to get one: pay a software house a fixed and firm price to produce it. The profit motivation will assure the careful examination of every bell and whistle with a jaundiced eye. And if improvements are made, to keep you a happy customer, they cost you nothing.

DON'T TRY IT IF YOU HAVEN'T KNOCKED IT

Which brings us right to the point: converting an installation is not as simple as it looks. If you are planning a reprogramming job of some size, come to Informatics and talk to us about helping you. (Or, if you feel qualified and ambitious, let's talk about your helping us.) We have the management depth to tackle this kind of problem. Remember the first large-scale reprogramming task ever undertaken? It involved simultaneously converting 18 large scientific computing installations from IBM 701 to IBM 704 equipment. Many policy decisions made today still go back to the lessons learned from that experience. Some important influences in the computing field grew out of it: SHARE programming standards, incentives for FORTRAN and COBOL, and monitor systems requiring separated input/output systems. Most of our staff members are veterans of that effort and have, additionally, gone the route as managers of their own installations. You recognize the names: Bauer, Wagner, Frank, Hill, Rector, Postley, Cohen, Bigelow, Bright, Lemons, Corbin, Kaylor, Mersel, Mallet, Stofko. If they can't help you, there's only one other place to turn: go see a good psychiatrist. He won't solve your problem, but he'll make you think you like it.

If you would like a copy of Frank Wagner's new paper "Design of a General Purpose Scientific Computing Facility", just print your

Name ____________________

Company ____________________

Street ____________________

City ____________________

State ______________ Zip ________

and return to:

informatics inc.®
5430 Van Nuys Boulevard
Sherman Oaks, California 91401

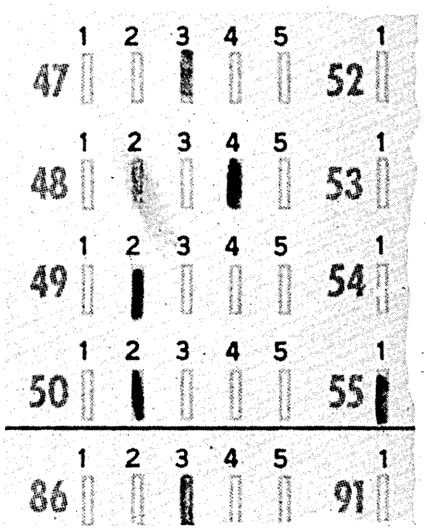
An equal opportunity employer

CIRCLE 26 ON READER CARD


DIGITEK 70 Optical Reader
DIGITEK

Now, any ten-year-old can feed data to your computer.

**Now anyone,
anywhere in your organization,
can talk to your computer.
All they need is
a No. 2 pencil.
All you need is
a Digitek
Optical Reader: $29,750.**



Straight lines: that's all it takes.

Short, vertical lines that can stand for numbers, letters, people, things, opinions—in English, Swahili, BCD.

(Remember the Board Exams? The Preference Tests? Enrollment Forms? Chem 101 Finals? That's where the idea of using a pencil mark to represent data began. Incidentally, if you do have tests to mark, we can do that, too. But that costs $39,780.)

You can put up to 2440 of those straight lines on an 8½ x 11 sheet, and then feed it to our Digitek optical reader. To be precise, you can feed 2400 of those sheets an hour to our Digitek optical reader; it's three times faster than others.

The Digitek system will:

- ☐ Transfer the data to paper tape. Or to mag tape. In a read format that suits your computer: we're compatible with them all. Matter of fact, we'll even paint the Digitek unit to match what you've got.
- ☐ Provide item analyses.
- ☐ Reads only the darkest mark in each response grid. (See the poor erasure on item 48? It doesn't count.)
- ☐ Sort out all improperly marked sheets.
- ☐ And switch from job to job in less than twenty minutes.

And Digitek's price doesn't hurt at all: less than $30K for the reader and your choice of mag tape or punched tape output.

The forms cost as little as $8.50 a thousand. The pencils, you've got in your desk. And you already have a data-acquisition terminal anywhere you've got an employee.

Isn't it time you made your mark in data processing? If you'd like to know how, drop us a line. Optical Scanning Corporation, 47 Spencer Avenue, Fairless Hills, Pa.

Optical Scanning Corporation

ASSEMBLE OR COMPILE?

eeny, meeny, miney . . .

by CHRISTOPHER J. SHAW

Believe it or not, there are still people around to whom the question of assemblers versus compilers is not a dead issue, despite the reams of paper and hours of talk that have so far been expended on it. These people buy a computer for their application and you tell them, "You ought to use a compiler for writing your programs. You'll save money." Then they ask you, "How will I save money, and how much?" And if, as sometimes happens, the first thing they have to do is order a custom-built compiler because the computer manufacturer doesn't supply one that's suitable for their application and the second thing they have to do is order an extra module of core memory to make sure their application will still fit the machine, then they're apt to listen quite closely to your answer. And if all you have are some five-year-old clichés about the advantages of procedure languages and the disadvantages of assembly languages, they may not be too impressed with it.

Besides these skeptical computer buyers, there's another group of people, professional system programmers, who fervently and often eloquently advocate sophisticated macro-assembly programs as being more flexible and generally superior to any compiler yet launched.[1] And nobody has effectively refuted their arguments yet, that I know of.

In any discussion on the utility of programming languages, it seems that the main problem is that there are very few facts, and very many opinions. What I have to say in this article isn't going to redress that balance by much. Nevertheless, I have my own opinions on the relative merits of procedure languages on the one hand, and assembly languages on the other—particularly in the areas of programmer training, program production and maintenance, program communication and transfer, and program execution—and I have a few facts to pass on concerning the experience of one organization (System Development Corporation) in the use of one procedure-oriented programming language (JOVIAL).

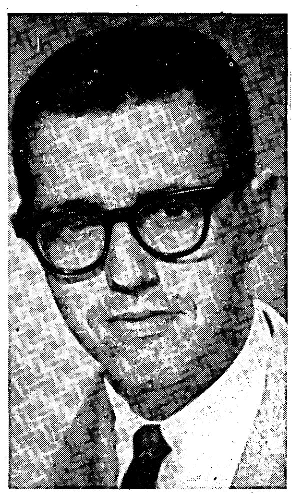
programmer training

One of the claims frequently made for procedure-oriented programming languages is that they are easier to learn than machine-oriented assembly languages. This claim is not universally valid nor even typically true. An assembly language for a simple, straightforward machine can be much easier to learn than a complex procedure language such as ALGOL, FORTRAN, COBOL, or JOVIAL. Certainly, manuals for procedure languages are no shorter or easier to read, nor are procedure-language programming courses less time consuming. The amount of intellectual effort needed to acquire professional capability for programming probably depends less, therefore, on the programming language being learned than it does on the quality of the training manual and of the instruction. Furthermore, much that a professional programmer needs to know—the application (e.g., air defense), system testing strategies, other programming tools—are largely independent of the programming language.

With regard to ease of learning, procedure languages may typically have smaller vocabularies than do assembly languages, but their syntax is almost invariably more complicated,[2] while assembly-language syntax stays pretty much the same from one language to the next; only the vocabulary changes. Moreover, this common syntax is extremely simple. The basic form is the familiar operator prefix notation—an operator followed by a list of operands.

Because of this syntactic simplicity, an experienced programmer may not even have to spend much time learning a new or unfamiliar assembly language before using it. A well-indexed reference manual may be all he needs to start coding.

On the other hand, it is undoubtedly much easier for a part-time programmer or nonprogrammer (for example, a system analyst) to learn just the basic elements of FORTRAN, say, than it is to learn all the details of FAP (the FORTRAN Assembly Program). But of course it is quite unnecessary to learn all the details of an assembly language before using it. A subset of the machine instructions,



Mr. Shaw, a project leader in the Programming Technology Staff at System Development Corp.'s Research and Technology Div., was instrumental in developing and documenting the JOVIAL programming language. He is currently chairman of the ACM National Special Interest Committee on programming languages and chairman-elect of the L. A. Chapter of the ACM.

[1] A leader of this school of thought is Mark Halpern, who has expressed his views in numerous articles, in particular, "Evolution of the Programming System, *Datamation,* July 1964, and "XPOP: A Command and Control Programming System," *Datamation,* December 1964.

[2] The nuances of ALGOL, for example, still provoke controversy among experts.

a subset of the pseudo-instructions, plus a well-chosen set of macro-instructions—these are all the occasional user really needs to know.

program production and maintenance

Perhaps the major and most impressive claim made for procedure languages, as compared with assembly languages, is that they reduce significantly the amount of effort needed for program production and maintenance. The basis for this claim is twofold: (1) a source program written in a procedure language is shorter (contains fewer steps) than an equivalent source program written in an assembly language; (2) a program written in a procedure language is easier to modify than one written in an assembly language.

Program Size. In terms of the number of individual symbols or characters that make up a program, it is not at all clear, in general, which language type has the advantage. More redundant procedure languages, like COBOL, may even require more symbols or characters per program than any assembly language. So far as program production and maintenance is concerned, however, the main thing is not how much the programmer has to write; rather, it is the number of individual program steps he must keep track of. Consequently, for a procedure language, the appropriate unit for measuring program size is the statement; and for an assembly language, the appropriate unit is the instruction.

instruction: statement

The ratio varies from language to language and from compiler to compiler, but typically a compiler for a procedure language will generate an average of four or five machine instructions per statement. A reduction of programming effort by a corresponding factor of four or five cannot be inferred from this, however, because the statements in a procedure language are typically two or three times more complex than the instructions in an assembly language. Furthermore, heavy use of macro-instructions in an assembly language can reduce the number of steps in a program almost as much as can the use of a procedure language.

Program Modification. Perhaps the most important general advantage of procedure languages is that they simplify and facilitate program modification, which is important because it starts minutes after the first line of code is written and continues until the program is scrapped. Moreover, it probably occupies more of a programmer's time than any other job.

Procedure languages ease the program modification job primarily by providing facilities for naming many often-referenced and often-changed aspects of programs, particularly those aspects relating to the type, structure, coding, and storage of data. When an item of data is changed, then, it is usually only necessary to change a single, data-definition statement, rather than search the entire program to change all references to the item.

Program Testing. An important consequence of the reduction in the number of program steps and the ease of program modification afforded by procedure languages, as compared with assembly languages, is a reduction in the amount of effort needed for program testing,* which is a vital and time-consuming part of the program production process.

Now the advantages of program testing in a procedure language rather than in an assembly language can be largely offset by a lack of useful source-language testing aids and compiler diagnostics. If the programmer has to read an octal dump, for example, to analyze the operation of his program, little has been gained.

The use of a slow compiler can, in certain circumstances, increase turnaround time enough to seriously retard testing; and an incompletely checked-out compiler can, by itself, make program testing spectacularly frustrating. In addition, there are many aspects of program testing—such as test design, test data preparation, and the analysis of test results—that are little affected by the language used to write the program. Nevertheless, program testing consists largely of error detection and correction, and by reducing the number of program steps and eliminating machine-oriented coding details, a procedure language reduces the opportunity for making errors and make existing errors easier to detect. Also, detected errors are easier to correct in a procedure language partly because of the ease of program modification that results from the use of such languages, and partly because each error in a procedure-language program will, on the average, involve fewer program steps.

All of the above arguments in favor of procedure languages as tools for program production are dependent on the applicability of the language to the problem being solved. Some languages, like ALGOL and FORTRAN, are best suited to scientific and engineering applications. Others, like COBOL, are best suited to commercial applications. Such languages can be used outside their intended areas of application, but the benefits to be derived from this are doubtful.

program communication and transfer

Program Readability. It is widely held that programs written in a procedure language are more readable than programs written in an assembly language. Certainly, this belief is substantiated by the fact that procedure languages tend to use English words (instead of three-letter mnemonic symbols) and familiar algebraic notation. I feel, however, that program readability is more dependent on the number of steps in a program than it is on either the complexity of the steps on the notation in which they are expressed. Thus, procedure languages promote program readability mainly by requiring fewer program steps to solve a given problem. It is worth noting here my opinion that programs with more than a certain number of steps are absolutely unreadable, no matter what language they are written in. Three lines of JOVIAL, say, might be easier for the nonprogrammer to read than the same routine written in an assembly language, but ten pages of JOVIAL and ten pages of assembly language would probably be equally incomprehensible to him. Besides, there is little advantage in having programs that look deceptively easy to read to someone who is actually unfamiliar with the language in which they are written.

Program Documentation. Closely related to the question of program readability is that of program documentation. Programs written in a procedure language supposedly require less documentation than equivalent programs written in an assembly language. This does not mean, however, that programs written in procedure languages require little documentation. On the contrary, much of a program's documentation—such as operator's handbooks and user's manuals—are needed anyway, no matter how readable the program listing is. Even if you could write programs in English, you would still have to document the real meaning of what it is they do, for an undocumented program, in whatever language, is just a list of individual actions, and while each separate step may be completely clear in its effect, the program itself is

* With typically optimistic schedules, this may appear not as a reduction of effort, but as a better quality program.

meaningless without an understanding of the purposes behind the actions.

While procedure languages do reduce the need for program documentation, this reduction is largely caused by eliminating the need for detailed program flow charts and other step-by-step program descriptions.

Program Transferability. The question of program transferability—the ease with which an old program can be fitted into a new or different computer—is also related to program readability. Unlike assembly languages, most procedure languages are largely (but not completely) machine-independent, and thus promise to facilitate program transfer greatly.

However machine-independent the language in which a program is written, it is seldom possible to transfer it to a different computer without making some changes to it. As I have mentioned, procedure languages facilitate program modification, but the key question is: With a procedure language, can the programmer make the necessary changes without digging into the program to find out in detail how it works? Because that is the hard part; the rest of the job is relatively simple—whether the programmer just modifies the program or rewrites it entirely.

Assembly language programs have essentially no transferability, except among so-called "program compatible" families of computers. So, it is almost always necessary to learn the details of an assembly-language program before it can successfully be transferred to a different machine. But with a procedure language, it sometimes *is* possible to change a program so that it can be compiled to run on a different machine without delving into all the details of its operation—a capability that is more likely to hold for smaller, more independent programs, and programs originally planned for possible transfer.

Sometimes, however, even with a machine-independent procedure language, it is not possible to transfer a program without first learning the program thoroughly. This is likely to be the case with programs that are closely integrated parts of larger programs, and with programs written to take advantage of unique machine characteristics. In such cases, of course, the cost of program transfer is increased. Still, the greater readability of procedure languages makes learning the details of a program's operation easier, so that even here, program transfer usually requires less effort than complete reprogramming.

program execution

Program Efficiency. The efficiency of a program is customarily measured in terms of the amount of storage space it takes and the amount of computer time it uses. It is quite clear that no compiler yet built can match the best efforts of a skilled assembly-language programmer in conserving these resources.[3] After all, any program—even a compiler-generated program—can be improved. But it is also clear that this fact is largely irrelevant, at least in the vast majority of cases. For one thing, the results of procedure-language programming must usually be compared with the mediocre efforts of average assembly-language programmers, and for another, the factors of program storage space and execution time often play little or no part in the cost of problem solving by computer.

Great variations exist among compilers in the efficiency of the object code that is generated. Typical reasons for this are: more emphasis is placed on code optimization in some compilers than in others; compiler writers vary in experience and skill; a compiler that has been used and improved for several years will likely generate much better code than one that has just been released; code optimization algorithms can be much more difficult to construct for some computers than for others; some procedure languages can be more easily translated to efficient object code than can others;[4] and some permit the use of much more efficient data structures than do others.

Despite these variations in object-code efficiency among compilers, some generalizations are possible. With a good compiler, for example, a poor or novice programmer will typically turn out better, more efficient code than he would using an assembler—the compiler places many inefficient coding techniques out of his reach. An average programmer using a good compiler will turn out code that is just about as efficient as the code he would produce using an assembler. An expert programmer, on the other hand, will turn out less efficient code using a compiler than he would using an assembler. But if he is using a good, well-seasoned compiler, one that has seen several years of maintenance, improvement, and use, his programs will typically be only 10-15% larger and slower than they would had he written them in assembly language.

Determining Program Execution Time. For any program of more than trivial size and complexity, it can be extremely difficult to come to any useful, general conclusions about program execution time. Determining program execution time for any given set of inputs is usually simple enough: you merely record the current time immediately before and immediately after executing the program. But it is a rare program whose execution time is independent of its inputs, and for a program with hundreds, thousands, or even tens of thousands of input variables, it can take considerable experimentation with the program, and a great deal of sophisticated statistical analysis, to arrive at any really valid timing estimates. So where program timing estimates are needed, as in the development of a real-time system, the programmer usually makes some *a priori* judgments, based on his intimate knowledge of the program, about which of the input variables most critically affect the execution time of his program. Based on these judgments, he concocts a set of three test cases, hopefully representing a maximum load situation, and a typical load situation. Then he times the program on the test cases, and if the results confirm his expectations, he has estimates for maximum, minimum, and typical execution time.

This process is clearly independent of the programming language in which the program was written. But the argument is occasionally heard that it is easier to make timing estimates for an assembly-language program than for a program in a procedure language. This argument is invalid. It is undoubtedly based on the fact that you can easily time the individual instructions in an assembly language, but not in a procedure language—a fact that has no general bearing on the problem of timing complete programs.

sdc experience with jovial

As part of a System Development Corporation research project[5] aimed at improving cost estimating techniques for computer programming, cost data were collected on 74 different programs—ranging in size from less than 200 to more than 300,000 computer instructions, and covering a wide variety of applications. Included were utility programs, simulation programs, data storage, retrieval, and reduction programs, and real-time programs for satellite monitoring, file management, military planning, air traffic control, and air defense. All were written by SDC.

The cost data collected for manpower and computer

[3] See "Specifying Object-Code Efficiency," *Datamation*, April 1966.

[4] Rosin provides some examples of this in his article, "The Quick and Dirty Compiler," *Datamation*, June 1965.

[5] Sponsored by the U. S. Air Force Electronic Systems Division. The work reported here has been documented in Weinwurm and Zagorski, "Research into the Management of Computer Programming: A Transitional Analysis of Cost Estimation Techniques," SDC publication TM-2712, 12 November 1965.

usage cover only the activities of program designing, coding, testing, and documenting, and do not cover prior system analysis or subsequent system testing—activities not really applicable to some of the programs.

Manpower costs for the 74 programs ranged up to 1,600 man months, and actual production rates ranged from 30 to 2,000 computer instructions per man month. Many of the programs were written for large computers, such as the Control Data 1604 and 3600, the IBM 7090, and the AN/FSQ-32; others were written for such small machines as the Control Data 160-A and the IBM 1401. The amount of computer time used in developing these programs ranged up to 9,000 hours, and the number of pages of documentation delivered with them ranged up to 14,000 pages.

Of the 74 programs, 60 were written in assembly language and 14 were written in JOVIAL. The cost data for the programs were analyzed to compare assembly language programming costs against the costs of programming with a procedure language such as JOVIAL. The results of this comparison are given in the following table.

The number of computer instructions, for both the assembly-language programs and the procedure-language programs, includes all the new computer instructions produced, whether delivered to the customer as part of the final program, or discarded for any reason (such as a change in requirements). The number of pages of program documentation includes both internal (working) documents and external documents (delivered to the customer).

SDC experience, as indicated in the table, has therefore been that JOVIAL programming, when compared with assembly-language programming (without taking into account system testing), doubles the program production rate and halves the computer usage rate and the documentation rate. However, because of the small size of the sample, because of the fact that it includes programs written in only one procedure language (JOVIAL), because many of the assembly-language programs did not employ macro-instructions, because the procedure-language programs were written for the larger, faster machines, and because the data were not adjusted to take into account the inefficiencies of compiler generated code, it is impossible to draw any valid, quantitative generalizations from the table.[6]

ASSEMBLY LANGUAGE AND COMPILER LANGUAGE COST COMPARISON*

	PROGRAM PRODUCTION RATE	COMPUTER USAGE RATE	DOCUMENTATION RATE
ASSEMBLY LANGUAGE	285 (322) 361	21 (24) 27	71 (88) 106
PROCEDURE LANGUAGE (JOVIAL)	398 (555) 712	9 (12) 16	23 (40) 57
	in computer instructions per man month	in hours used per thousand computer instructions	in pages produced per thousand computer instructions
SIGNIFICANCE OF MEAN DIFFERENCE	95%	90%	80%

* Entries have the format low-value (mean-value) high-value. The low-to-high ranges represent a 70% confidence interval, implying a 70% probability that average rates taken from a different sample would fall in the range; significance is expressed as a probability that the mean difference observed in the sample did not occur by chance. Because of the small number of programs in the sample, the results were especially susceptible to distortion by outlying cases. To reduce this distortion, a total of five values, more than three standard deviations from the median, were replaced by values at 3.0 and 2.9 standard deviations from the median—a process known statistically as *Winsorizing*. (The median was used rather than the mean because of the skewed distribution of the sample.) The *Student t* distribution was used to estimate the population means and also the significance of the difference between the means.

summary

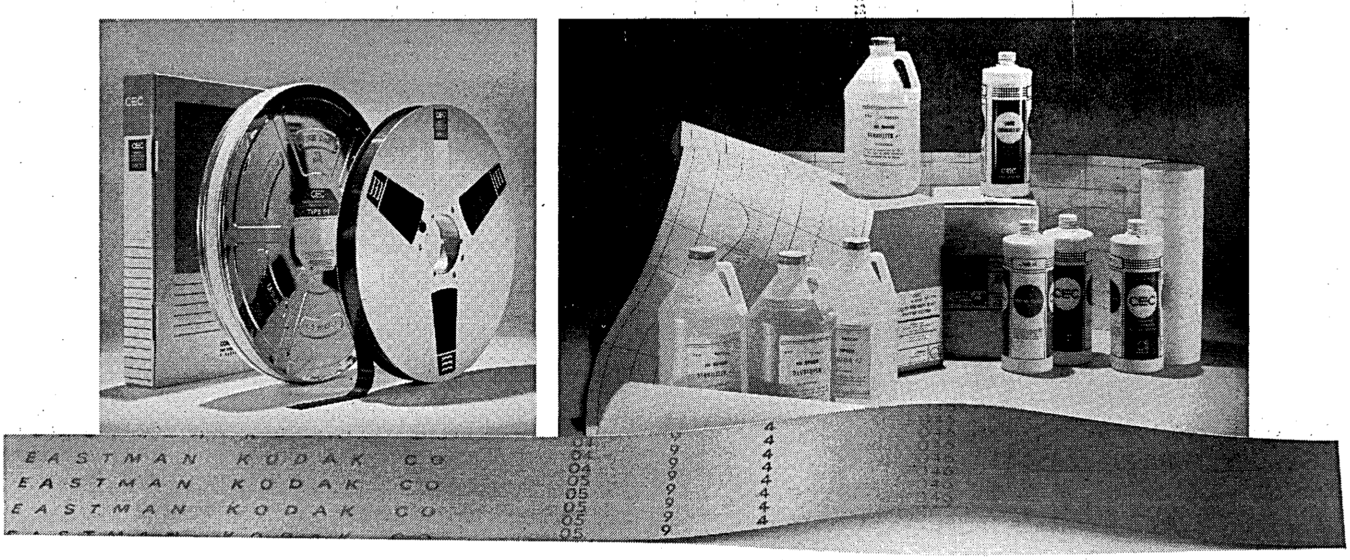
In this article, I have expressed the following opinions on the general question of procedure-language versus assembly-language programming:

- In general, there is no appreciable difference in the amount of training needed to acquire professional competence in the use of either a procedure language or an assembly language.
- The use of an appropriate procedure language, by reducing the number of steps in the source program and by easing the job of program modification, can significantly reduce the amount of effort needed for program production and maintenance.
- The use of a procedure language instead of an assembly language improves the communication of algorithms between programmers—primarily by reducing the number of steps needed for their expression. This improvement results in a reduced need for detailed, step-by-step program documentation.
- Procedure languages, because they are largely machine independent and because they make programs easier for programmers to read and to modify, can greatly facilitate the transfer of programs between different computer types. Of course, the transfer of programs that are system-dependent is not always practical, however machine-independent, readable, and changeable they are.
- Object-code efficiency is often not an important factor; nevertheless, with a good compiler, an average programmer will usually turn out code that is just about as efficient as the code he would produce using an assembler.
- The use of a procedure language instead of an assembly language does not increase the difficulty of obtaining program timing estimates, since these estimates seldom involve the execution times of individual program steps.

I have also discussed the experience of SDC in the use of JOVIAL. Although only a small amount of numerical data are available, this experience indicates that JOVIAL programming, when compared with assembly language programming, doubles the program production rate and halves the computer usage rate and the documentation rate. ■

[6] For a discussion of this, see Weinwurm, George, Cost Estimation for Computer Program Development: A Progress Report and an Evaluation, Presented at the 4th IFORS Meeting at MIT, August 29 to September 2, 1966. Also available as SDC SP-2418/001/00.

Now...a single source for every tape recording and oscillograph need



CEC, a leading manufacturer of magnetic tape recorders and oscillographs, is the *only* company which offers single source responsibility for *all* tape, paper, and chemical needs.

Result: a substantial contract advantage to the user—combined with *assured* quality control, convenience, and delivery.

A complete family to serve you

CEC Magnetic Tapes. Created and produced by Eastman Kodak, these tapes offer significant advances and benefits unmatched by any others.

- Only CEC tapes are color-coded on the box, can, reel; even digitally numbered on the tape itself for instant identification and traceability.
- Only CEC tapes are divided into *four* specific categories to meet the exact requirements of every data recorder.
- Only CEC tapes come shielded in metal containers – packed in cardboard filing boxes covered with protective plastic sleeves.
- Only CEC tapes are protected from shipping and storage damage by means of a plastic waffle hub, thus preventing tape serration and flange deformation.

DATACOLOR 88. A two-color red and cyan emulsion paper which produces traces in distinctly different hues for *rapid* and *positive* identification.

DATAFLASH 55. The extreme sensitivity of this paper now makes it possible to obtain *perfect* oscillograph records at writing speeds in excess of 75,000 ips.

DATAFLASH 56. A high writing speed product which produces superb traces, timing, and grid lines where Zenon lamps are used.

DATAFLASH 57. High writing speed Dataflash 57 features a "develop-out" option for oscillograms of archival quality.

DATATRACE 11 & 31. Heat sensitive recording papers for use with CEC's DG-5511 Portable Two-Channel Thermal Writing Recorder. High contrast or reproduceable types are available.

DATARITE 33. DATARITE refers to CEC's Datarite Magazine in which oscillograms are automatically developed and dried as quickly as data is recorded. DATARITE 33 is ideal for high writing speed applications.

DATARITE 22. Type 22 has less sensitive emulsion than Type 33. At low writing speeds, it will give excellent oscillograms.

Type 49943-3 Datarite Developer Solution is a non-ammonia, general purpose formula which provides exceptionally high trace contrast and high writing speed capability.

Type 49943-4 Datarite Developer Solution is an ammonia-type formula which combines high writing speeds with reduced paper staining in high humidity environments.

Hi-Speed Chemical Kit — No. 217050. Designed for use in CEC's new 23-109B and 23-109A Processors. Easily capable of processing more than 1500 square feet of oscillographic paper without replenishment, it's the logical choice where numerous rolls of recorded material must be processed.

Liquid Concentrate Chemical Kit — No. 156455. The "economy" package for use with the 23-109A. Processes up to 500 square feet of paper, on an intermittent basis, thereby avoiding costly waste of solutions.

For all the facts about CEC tapes, oscillographic papers, and chemicals, call or write for CEC Bulletin Kit 9050-X6.

CEC
Data Instruments Division
CONSOLIDATED ELECTRODYNAMICS
A SUBSIDIARY OF BELL & HOWELL/PASADENA, CALIF. 91109
INTERNATIONAL SUBSIDIARIES: WOKING, SURREY, ENGLAND AND FRIEDBERG (HESSEN), W. GERMANY

CIRCLE 29 ON READER CARD

with elementary operations

MICRO-PROGRAMMING THE SPECTRA 70/35

by C. R. CAMPBELL and D. A. NEILSON

In recent months several computer manufacturers, domestic and foreign, have announced new product lines that offer processors using read-only memories for control. These read-only memories have a particular number of fixed length words which contain control instructions, instructions that are on a lower machine level than the computing instructions. The systems they are controlling have consistent data paths, so common repetitive programming is applicable. The control instructions transfer data from register to register, trigger main memory, perform the arithmetic and logical choices, test flip-flops and conditionally or unconditionally branch to other read-only memory instructions. These instructions, which are bit configurations of various lengths depending upon machine design, are known as Elementary Operations or, commonly, EO's. A group of Elementary Operations make up a microprogram. The microprogram performs the operating instruction.

Most microprogramming is done in some sort of symbolic coding language, an example of which might be "A + (2) B ⟶ B." This could be translated as, "Add the A register to the two's complement of the B register and store the results in the B register." Along with this particular operation, there could also be a test and branch or an unconditional branch. Depending upon the size of the EO, both the conditional and the alternate branch address could be carried in each instruction.

The bit configuration that results from the symbolic coding causes the correct sequence of operations to take place at the correct time. Now if, in addition, testable indicators such as "carry" or "zero result" are automatically set, the microprogrammer can check the result of his manipulations.

In order to give the microprogrammer sufficient flexibility in his control of the internal operations, several basic functions, such as "add binary," "add decimal," "shift," and "cycle memory" must be provided. These functions are planned, along with the tests and indicators, and the control problems are solved before function definition. When definition is finally made, internal operation can be considered from the symbolic level, free from hardware constraints. If the Elementary Operation is straightforward, the microprogramming of an instruction algorithm is similar to the programming of any application, and the job can be done by a programmer, not necessarily an engineer. This, of course, assumes that the programmer working at the symbolic level cannot create any problems for himself other than programming problems.

As an illustrative example, staticizing an instruction exercises most of the basic functions available to the microprogrammer. A program counter retains the address of the next operating instruction. Using the address stored in the P counter, the microprogram cycles the memory and reads the first two bytes of the instruction to be performed. These bytes are stored in registers for testing. The first byte contains the operation code of the instruction to be performed, and by analyzing certain bit positions the micro-

Before becoming supervisor of microprogramming control of the RCA Spectra 70/35 processor, Mr. Campbell was senior product planner on the 70/15 and /25. A former systems analyst at Thiokol Chemical Corp. and General Motors Ternstedt Div., he holds a BS from Wayne State Univ.

program can distinguish between different instruction formats. The operand addresses are staticized according to the format type, and then the eight-bit operation code is transferred to the read-only-memory address register. This operation affects a branch to the correct microprogram that executes that particular instruction. If the operation code is not a valid one, the branch is to the microprogram for op-code trap interrupt.

Writing microprograms at the symbolic level for internal processor control requires more than just precise EO definition. A few of these requirements and a method to satisfy them can be seen by an examination of the RCA 70/35.

The 70/35 is RCA's latest member of the compatible SPECTRA 70 family. A general system description can be divided into three categories:

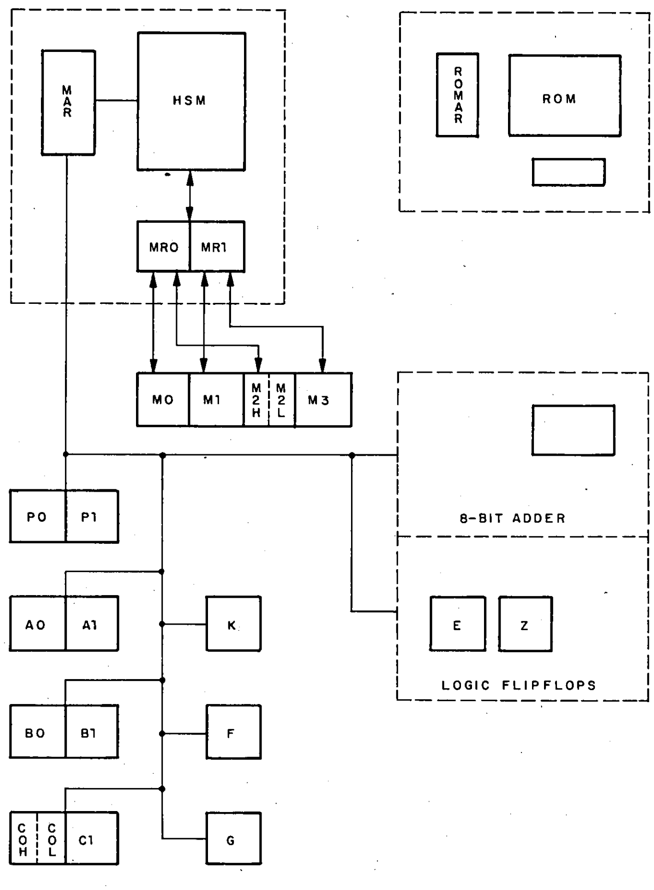
1. Control of the data structure for instruction staticizing and execution is by Elementary Operations (EO's) contained in a nondestructive read-only memory (ROM).
2. All the instructions that are featured in the larger SPECTRA 70 processors are available in the 70/35.
3. All I/O devices in the SPECTRA 70 line communicate with the 70/35 via a multiplexor or selector channel and the RCA standard interface.

In this article we will concern ourselves with the first category. The most significant feature of the 70/35, and the one that guided the general design concept that developed, was the use of a 27-bit EO word.

Several secondary features are unique to the 70/35. However, these are a direct result of the basic EO format decision. Two of the features are: 1) the easy development of a comprehensive EO simulator and assembly system, and 2) the feature that allows performing Elementary Operations obtained from main memory as well as read-only memory.

In December, 1964, RCA announced four new processors, one of which, the 70/45, utilized a read-only-memory type control structure. A read-only memory had been designed for this processor, and it used a 53-bit word. The 70/35 design group decided to add one bit and to use the 54 bits as two 27-bit EO words. Each 54-bit group is accessed in 960 nanoseconds, providing an effective read-only-memory cycle time of 480 nanoseconds. The decoding network for the 70/35 is unique, but the basic read-only memory is the same employed on the 70/45. With the 54-bit word, the read-only memory contains 1024 words. With the 27-bit format this, of course, doubles.

This type of control, therefore, uses 2048 words of lesser complexity as opposed to 1024 larger, more powerful EO's. This, in effect, halves the control memory cost if a one-for-one relationship can be retained. The design group concluded that with a relaxation of performance specifications (in comparison with the 70/45), a simple EO structure could more easily utilize sophisticated programming techniques and make up for the loss in EO power. The relaxed performance requirements allow the technique of modular programming with many common routines, a method that pays a time penalty but saves EO steps. Another factor in the EO word design was that in the final judgment, an EO is still function bound. No matter what the power of any step, whether it be a status level or elementary operation, it still must perform its basic function-branching steps sequentially. The adoption of this EO word enabled all 144 operating instructions to be contained in 2048 EO words. Although 144 instructions seems to be a large amount, the greater number require an extremely simple algorithm. In addition, many instructions are similar in their operation and can be combined into common routines that still provide tolerable timing specifications. When the shared routine method is employed, a sufficient number of the smaller steps are left to handle the more complicated instructions.



70/35 BLOCK DIAGRAM

register communication

Register communication in the 70/35 is implemented through a modified one-byte input bus and a modified one-byte output bus. The microprogrammer can address all the registers in the main data structure, with the exception of the memory address register (MAR) and the memory data register (MR). There are fifteen addressable 8-bit registers in the 70/35 that can be used for data manipulation. Some of these registers are further subdivided into two 4-bit registers which also require separate addresses. The 4-bit registers are used to perform such instructions as Pack, Unpack, Move Numeric, etc. Any

Mr. Neilson is the project manager directing design of the Spectra 70/35 processor. Formerly the product planning manager at RCA's Palm Beach Gardens facility aiding in the development of Spectra 70/15 and /25, he has also been associated with Burroughs and IBM. He received a B.S. in electrical engineering from the Univ. of California.

two 8-bit registers can be used as the operands in any of the register-to-register operations. For the decimal functions the 4-bit registers can be used as a destination with any 4-bit or 8-bit source. This feature aids in the emulation of 6-bit machines. The 70/35 uses a 2-byte memory access with configurations of 16, 32 and 65K. Each of these memory configurations has a section of unaddressable or shaded memory.

In order to apply programming techniques to machine control, functions had to be designed that would provide the programmer with the ability to share routines. One general method, of course, was to make the individual steps very basic, thereby making them common to many algorithms. This was inherent in the extremely basic 27-bit Elementary Operation. Another general method was to provide testing capability in each EO to simplify linkage problems.

The 70/35 Elementary Operations fall into two categories; register-to-register operations, and variable operations (i.e., cycle memory, offset address, etc.). Of the 27 bits, five are always used to describe the function type. Fourteen of the functions are the register-to-register operations. These are described as follows:

EO LAYOUT

W	X	Y	T	J
5	5	5	6	6

W—This is the five-bit function field. This is the EO op code and directs the operation to be performed.

X — This five-bit field describes the source and destination register for the operation as described by W.

Y — This five-bit field describes the other source register for the operation described by W.

T — The six-bit field describes a test to be performed. With six bits, 64 tests are available to the 70/35.

J — This six-bit field has a branch address that is taken as the test is true. Because of the six-bit limitation this branch address is limited to modulo 64.

FUNCTIONS

Transfer —

When the function (W Field) describes a transfer, the bit configuration held in the register described by Y is transferred to the register described by X.

Add —

This function transfers the quantity in the register described by X to a holding register. It then takes the quantity in the register described by Y, adds it to the quantity in the holding register, and gates the result back to the register described by X.

Add One's Complement —

This function is similar to the Add except that the X quantity is complemented before the Add.

The balance of the functions used in register-to-register operations provide for decimal adding and subtracting, adding with a signed binary, and for performing the logical operations (And, Or, Exclusive Or). The register-to-register operations can also be used with an eight-bit literal operand held in the EO word.

Along with the register-to-register operations, several other functions are available to the 70/35 microprogrammer. There is a function to cycle the main memory, a function to quickly create memory addresses from a 4-bit register number, a function to permit internal two-byte transfers register-to-register, and a two-way branching function. All the functions that are not register-to-register allow a choice of either a test and branch or an unconditional branch. The test and branch will still allow only modulo 64 branching. The unconditional branch, however, allows branching to any location in the read-only memory.

Two functions greatly aided routine sharing in the 70/35, the first of which was an "indirect function." This function utilized as its decoding base (function bit configuration), the low order four bits of the op code register. Routines then could be written for a general approach with the arithmetic or logical operation determined by the particular instruction to be performed. By using the four low order bits of the op register, 12 different indirect functions can be performed in the 70/35. Assuming staticizing is concluded, the example is a general illustration of the indirect function:

1. Read operand 2; test
2. Read operand 1
3. Perform indirect functions; test
4. Write operand (result of function)
5. Decrease length and if not zero branch to #1

With initializing and the proper tests assigned, this routine can handle Add Decimal, Subtract Decimal, And, Or, Exclusive Or and with tests that permit skipping one step or another, the routine can also do Zero and Add, Compare Logical or Compare Decimal.

Another function added for programming versatility was the indirect branch, a method by which the branch address is previously prepared in a particular register. Upon conclusion of a routine, the register is transferred to the ROM address register and the proper return address is effected. This is, of course, a common programming technique. However, the important thing is that it was implemented in the 70/35 specifically as a microprogramming aid.

The 70/35 presently has 26 defined functions, with several having multiple variations. This leaves six functions for future unique applications.

The elementary operations available to the 70/35 microprogrammer are as follows:

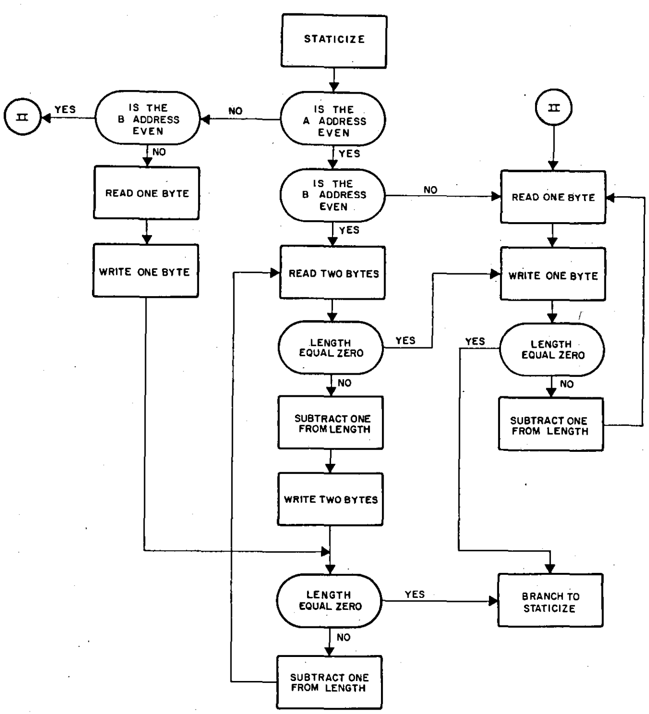
Register-to-Register	Variation
Add	Cycle Memory (3 functions)
Add With Sign	Offset Transfer (address) (2 functions)
Add One's Complement	Two Byte Transfer
Add One's Complement With Sign	Shift
Add With Reset Carry	Literal Increment-Decrement
Add Two's Complement	I/O Control
Add Decimal	Special Branch (2 functions)
Add Nine's Complement	Special Control
And	
Or	
Exclusive Or	
Compare (binary)	
Transfer	
Indirect Function	

Early in the 70/35 design study a simulation system was designed that would be the main tool for testing algorithm logic on the engineering prototype. Microprograms were written in a simple coding language and assembled in an edit phase. When the microprogrammer was ready to test his program, he entered it along with any required initializing data into the simulator phase. This program traced the microprogram and provided a step-by-step internal picture of the machine for the programmer to analyze. After the complete microprogram file was considered logically true, read-only-memory allocation data and logic definition were combined with the file to produce the Elementary Operation bit configurations and the read-only-memory wiring list. In addition to the edit, simulation and read-only-memory assembly, the system provides many statistical runs analyzing test and branch data.

This complete operation was viewed from the symbolic level by the microprogrammer. In the original planning, hardware aids, function requirements and many other significant design decisions were mutually discussed, but subsequently all algorithms were written and tested using the simulator.

In order to demonstrate the operation of a 70/35 microprogram and the symbolic language used to program it, the sequence for Move Numeric and Move Zone is shown with a detailed explanation. The operation code for Move Numeric is $D1_{16}$ (11010001); the operation code for Move Zone is $D3_{16}$ (11010011). During staticizing, the operation code is placed in the F register, the length is placed in the G register, a fully staticized second operand address is placed in the B register and a fully staticized second operand address is placed in the A register. With the operation code in the F register, the microprogram can distinguish between Move Numeric and Move Zone by

MOVE SS FORMAT OP CODE $D2_{16}$



FUNCTIONAL FLOWCHART OF THE MICROPROGRAM FOR THE SS FORMAT MOVE INSTRUCTION

examining the one bit where the two op codes differ. This example assumes the instruction has been staticized.

1. $A \rightarrow S$, RR, +1 (the symbolic expression for a cycle memory read)
2. $M_2 \rightarrow C_0$
3. $B \rightarrow S$, RR, +0 $F_1=0$ Branch to 7
4. $C_{0H} \rightarrow M_{2H}$
5. $B \rightarrow S$, WRS +1 G=(00) Branch to Staticize
6. $G-(1) \rightarrow G$ Branch to 1
7. $C_{0L} \rightarrow M_{2L}$ Branch to 5

1. The second operand is read from main memory and put into M_2. The A address is incremented by one.
2. Register M_2 is transferred to C_0. Both M_2 and C_0 are sub-divided into two four-bit registers.
3. The first operand is read from main memory and put into M_2. In addition the bit F1 is tested to see whether the branch is taken. If not, the next sequential step is performed.
4. The zone from the second operand previously stored in C_0 replaces the zone of the first operand.
5. The modified first operand is written to main memory and the B address is incremented by one. The G register, which holds the length count, is checked for zero. A length of zero indicates the instruction is complete. If the test is true, the branch to staticizing is taken. If the test is not true, the next sequential instruction is performed.
6. One is subtracted from the G register and the microprogram unconditionally branches to Step 1.
7. In Step 3, a test was made for instruction type by checking bit F1. If the instruction was Move Numeric, Step 3 is followed by Step 7 which replaces the numeric portion of the first operand and unconditionally branches to Step 5.

special features

The 70/35 can contain an optional second read-only-memory bank. Assuming one is used to perform the 144 SPECTRA 70 instructions (this is not a systems requirement), this provides another 2048 27-bit elementary operations for some other purpose. With a control structure that is "data oriented" rather than "instruction oriented," it is possible to emulate almost any machine organization. In fact, with the hardware improvements available, the 70/35 emulates older 6-bit machines with a substantial improvement factor.

Another unique feature on the 70/35 is the implementation of the Diagnose instruction. The Diagnose instruction is primarily a test and maintenance feature which aids troubleshooting in the field and initial checkout in manufacturing. However, it can also test future read-only-memory bit configurations for accuracy.

The object of any maintenance aid is to identify the problem area. To do this effectively, it was determined that the test and maintenance programs should be able to control the internal structure of the machine. The adoption of the small EO word structure provided the 70/35 with the ability to easily perform microprogramming steps obtained from main memory. This method allows the test and maintenance programs to make internal checks never before possible. It also permits the programmer to simulate foreign instructions at the microprogram level.

The Diagnose instruction will perform any function that can be performed from the ROM except input/output instructions. For I/O, the test and maintenance routine must jump back to the program level. However, because they are able to control internal machine operation this presents no problem. Termination interrupts can be handled at the program level or at the microprogram level. The preparation of test and maintenance microprograms can use a variation of the same simulation system used by the microprograms in ROM. As in the regular microprograms, coding is done in the simple assembly language and all work is done at the symbolic level. The Stimulation System provides the listings, the testing, and the object bit configuration.

The techniques developed in the current processors using read-only-memory control offer a high degree of flexibility for future system design and marketing. If the Elementary Operations are basic and easy to understand, a user could program his own internal data structure. The specialized customer has selected system configurations that approximate his real need, but at the same time has long wished for a system tailored to his unique application. The design method used with read-only-memory controlled processors make this unique system practical for both user and manufacturer. ■

CONFIGURING MULTI-TASKING SYSTEMS

know your workload

by ROBERT L. PATRICK

Multi-tasking is defined as the use of a single CPU for the simultaneous processing of two or more jobs. No coordination is required prior to job input. After the first job is loaded, the software allocates core storage and I/O devices and then loads additional jobs to the limit of the available facilities. Barring higher priority jobs, the jobs, once loaded, are run to completion.

Multi-tasking as a technique is not new. Honeywell pioneered the concept with the H-800 in 1960. Although it was somewhat limited in scope, their parallel processing was conceptually equivalent. Other manufacturers also have the proper hardware/software combinations and various names, including multiprogramming and multiprocessing, denote their systems. Once the naming hurdle is surmounted, the interested user will find that all methods are aimed at finding something for the CPU to do during I/O activity.

background

Historically, operating an I/O device used only a fraction of a CPU's facilities. However, the entire computation process stalled until the I/O was completed. If a job used a lot of I/O, such as a commercial application, it was frequent to "waste" more CPU cycles than were productively used. Over the years additional circuitry and software have been added in an attempt to productively utilize some of this wasted time. The additional circuits raised the cost of the control units and the CPU slightly (less than a percent). The additional software took more resident core (about 500 instructions) and used some of the "waste" for its execution.

trade-offs

The most important (and expensive) innovation was the provision included to allow relocation of object programs as they were loaded. Some machines depend solely on software to perform this relocation. With this design, enough information is stored with the object program (in load form) to allow it to be relocated into available space as it is loaded. The penalties of this method are rather severe: the additional characters to allow relocation can reduce the effective I/O transfer rate during loading by 50% (or more).

In addition to this load time penalty, debugging and error diagnosis under multi-tasking are harder (since a program seldom occupies the same area on two consecutive runs and the addresses are always different) and accounting routines must be larger and more complex to assure equitable distribution of charges (these routines use more resident core and use some of the "waste" for execution). Furthermore, the object programs are structured to use arbitrarily small chunks of memory (say 2K words) so that the storage allocation algorithm (which must precede the relocation at load time) is offered the largest variety of choices when mapping queued jobs into the available space.

Depending on the relative proportions of interlocked input, output and compute (I/O balance), the net gains from this more sophisticated hardware and software can still be spectacular. A job that is I/O limited can cohabit during execution with another which is compute limited and both of them can be finished in an elapsed time period approximating the execute time of the longest one running solo. This is typical of the staged demonstration. However, what the salesman usually neglects to mention is that the jobs with inherently balanced I/O pay a performance penalty each time they are loaded and executed. On sufficiently short programs, this penalty approximates the execution time of the job!

user alternatives

After the user has chosen a specific computer, he usually has little control over these software matters. The form of the object program is so basic to the design of the software (compilers, loader, library routines) that the user must almost take what he gets. However, there are a few alternatives left open to him.

First, the applications programmers can be coerced to combine short runs into longer ones so the load-time penalties are paid a fewer number of times per answer. Second, the programmers can be thoroughly trained in the new features of the new system. Since you are paying for these features, you should exploit them to gain improvements (sometimes intangible) in the program preparation process, debugging, or file management.

Of course, an installation manager with a real hardship can modify the software to improve it, provided he has the required talent in-house or the budget to acquire it.

The most palatable alternative to the manager is to configure his system properly so multi-tasking will really work well in his shop.

minimum hardware

It's pretty obvious that a computer system must have enough primary I/O devices (card readers and printers) to handle the average load. It is less obvious that extra primary devices are sometimes justified whenever peaking in either input or output causes the CPU to become idle or continue to run one solo printer long after all input and processing have been completed.

Many shops do not know their workload well enough to have established firm job classes. Further, many shops lack a statistical distribution of job classes by running time, core required, primary devices used and secondary file storage devices required. The core requirements determine how many jobs *may* be loaded for simultaneous multi-processing. Given sufficient core, it is possible to achieve multi-tasking operation if *all* these other necessary requirements are also simultaneously met:

1. It is necessary to have several jobs in the ready queue whose total core requirements are less than the amount of memory installed.
2. It is also required to have enough card readers, printers and other primary devices to handle the combined total requirement.
3. It is mandatory to have enough tapes to fill the total requirement for all simultaneous jobs.

If any one of these three conditions is not met, the jobs may be loaded but cannot operate without device substitution. If it is necessary to use an on-line utility program to pre-store or post-print to or from a tape or disc as a substitute for a primary device, you may need to adjust your configuration. If you substitute frequently and need capacity, get I/O devices and not CPU. Conversely, frequent device substitution when you are running less than 20 hours/day means reduced budget and longer turnaround.

The system with the maximum throughput is a configuration with enough *independent* file access mechanisms for the totality of all files all jobs simultaneously require. Your salesman has probably told you that many separate work files may coexist on one disc drive. While he is literally correct, he probably neglected to mention performance or that insidious bugaboo: interference.

disc interference

When we first installed the 7040-7094 Direct Coupled multi-processors, some of us learned about interference the hard way. Now, three years later, some S/360s and 6600s are suffering from the same disease. If you don't get the performance you expect, yet have spent enormous sums for core, tape, and I/O gear, you may be suffering from interference. If the execution time for the same object program with the same data varies significantly on two separate test runs with a loaded system, you've got interference!

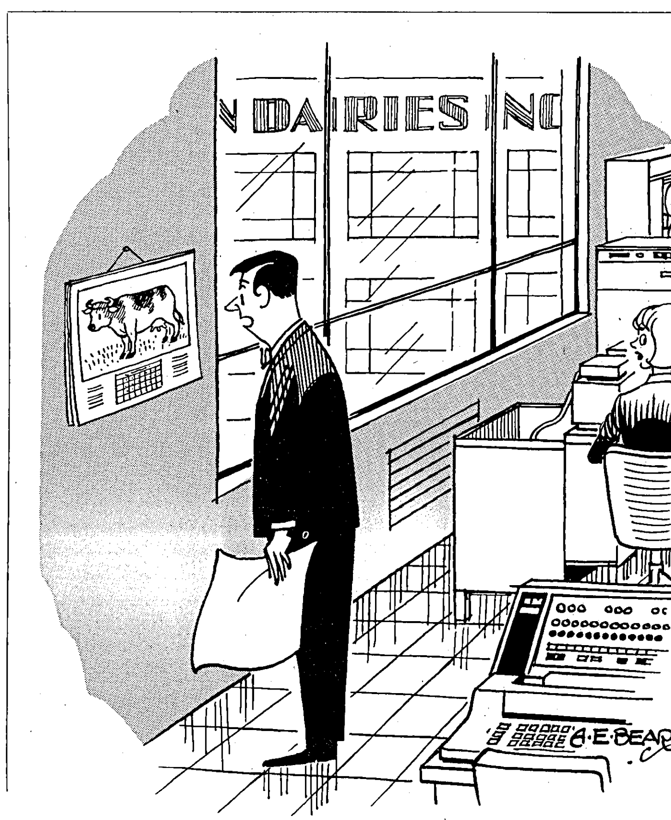
Disc interference is a phenomenon of multi-tasking caused when the duty cycle of the disc is so high that extensive queues build up while the CPU is forced to wait for disc service.

The most pitiful instances of the disease occur when an installation manager has stretched his budget to the limit and put in a profusion of discs, drums, tapes and primary I/O, but has neglected his file assignments. In this event he still suffers from the primary symptom (variable execution time) although he has plenty of access mechanisms. Sometimes discs beyond the minimum complement are installed without reconfiguring the software to exploit them. Thus, the CPU still waits on a disc, but additional disc modules are available and ready. When this occurs, the software assignment tables must be revised to even the load and get the same duty cycle on all devices. It is sometimes necessary to double (or multiple) store a non-reentrant compiler if it is frequently used and your runs are short.

configure to your load

For any one CPU the configuration with the maximum throughput (for that CPU model) is one with sufficient core to allow almost 100% of the CPU cycles to be used. This requires a balanced configuration of devices. Economics dictates all devices be heavily and evenly loaded, and a separate access mechanism (channel, control unit, and read/write head) be provided for each file that is simultaneously open. While this configuration has extremely high performance, it also has an unbelievable cost.

With the current trend in rental agreements, the most economical configuration is the one that will process a day's worth of *your* workload in 24 hours. Clearly this is a personal matter depending on your workload and its seasonal variation. Beyond this minimum, more capacity, reduced turnaround, and more rental can be had by tuning your configuration to your workload. However, you must know the characteristics of the sequence of jobs as they arrive at the ready queue in order to tune it. ■




All right, MR KONG! We've got your number!
IT'S
TRUE
KING!
gh.

BREAKING CAMP!

In the fall of 1964, an article entitled Notes on Camp, by Susan Sontag, appeared in the Partisan Review and since then our lives have been gravely affected by the emergence of Batman, Barbra Streisand, Tiffany lamp shades, non-James Bond spy movies, Pop Art, Edward Albee plays, and the like.

It now appears likely that our professional lives have not escaped this third stream of taste that is neither good nor bad but encompasses the extravagant, silly, awkward, old, stuffy, arty, bizarre, and overpowering. As the *New York Times* noted, "Camp has come along to fill the singular need for a word to describe all those things that, until recently, have loosely been called 'too much' and 'fantastic.' " And as Robin might say, "That calumnious computer camp cult warrants further looking into."

Camp has been divided into the two major categories of Pure Camp and Intentional Camp. The Pure category is stratified into High, Middle, and Low. In addition to these levels, there are two modes: Active and Passive. For a clearer definition of each type, I have selected certain notes from the Partisan Review article. They appear within quotation marks.

PURE CAMP

"One must distinguish between naive and deliberate Camp. Pure Camp is always naive. Camp which knows itself to be Camp is 'camping,' which is usually less satisfying."

"The pure examples of Camp are unintentional; they are dead serious. The art nouveau craftsman who makes a lamp with a snake coiled around it is not kidding, nor is he trying to be charming; he is saying in all earnestness: 'Voila! the Orient!' "

"In naive, or pure, Camp, the essential element is seriousness, a seriousness that fails. Of course, not all seriousness that fails can be redeemed as Camp. Only that which has the proper mixture of the exaggerated, the fantastic, the passionate, and the naive."

Thus we have in this category:

The Burroughs Corporation
Management Information Systems
Candy stores with 360's
Computer movies
Sneaker-clad programmers
The U.S. Federal Government

LEVELS OF CAMP

A level of Camp refers to an intellectual or academic stratum. For example, Orlando, Winnie the Pooh, and Batman run from High to Low.

HIGH

AFIPS Proceedings
The Brooks Bill
Time-sharing
ARPA
ALGOL X and Y

MIDDLE

PL/I and II
Western Union
DPMA
CODASYL
Quiktran
ADAPSO

LOW

DOD COBOL manuals
Starting new consulting firms
GUIDE
Information utilities
CPC's
Sorting needles

INTENTIONAL CAMP

"Probably, intending to be campy is always harmful."

SDC
Bearded programmers
Cyberculture
Project MAC
SCERT
Bauer's Second Law

ACTIVE AND PASSIVE MODES

Active and Passive Camp definitions are rather straightforward. Frugging at Arthur's and watching an eight-hour Andy Warhol underground movie are obvious cases.

ACTIVE

Computing Reviews
Data base systems
Program warranties
BEMA
Console debugging
Light pens
Coding in machine language

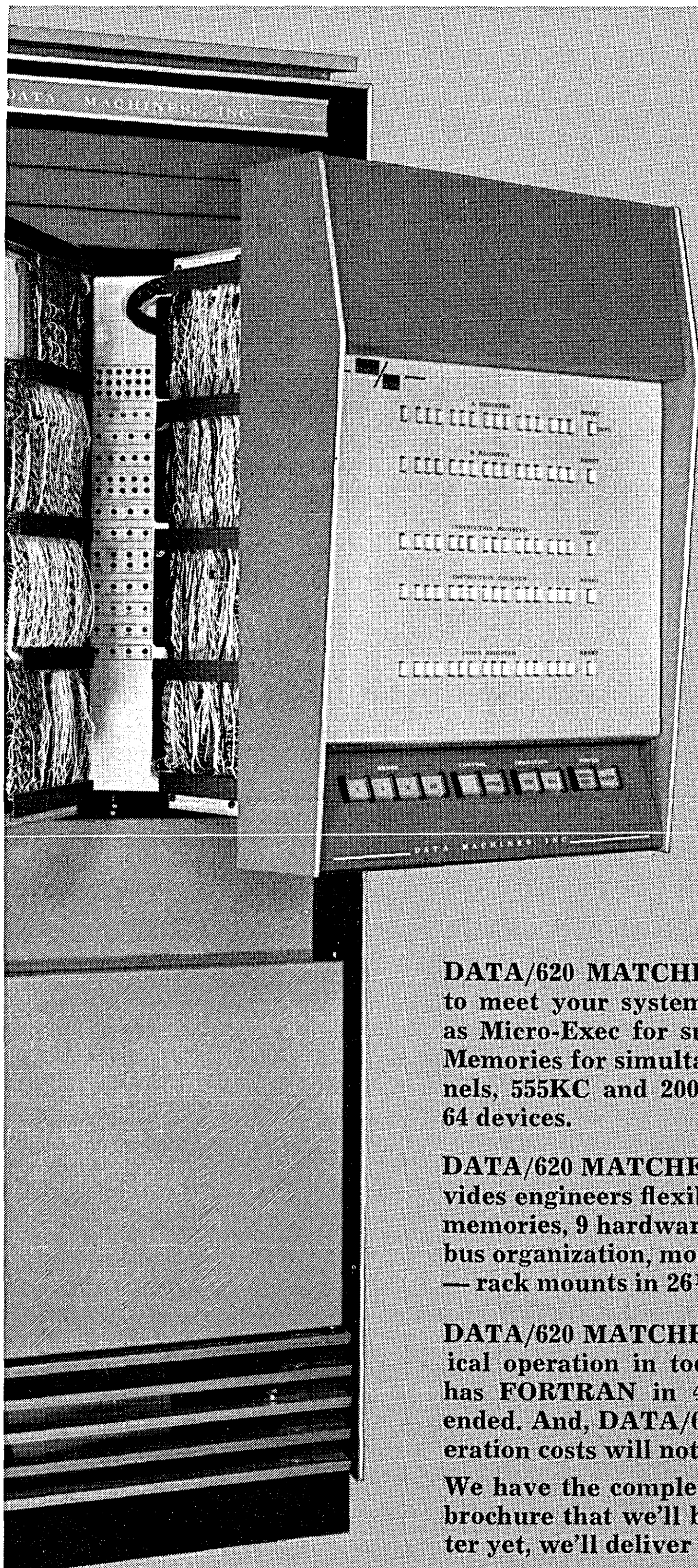
PASSIVE

IBM scientific symposia
ACM Journal
SHARE
ASA standards committees

It should be kept in mind that the application of Camp categories refers solely to style and not content. It doesn't argue about good or bad but offers a different, supplemental set of standards.

"Camp sees everything in quotation marks. It's not a lamp, but a 'lamp'; not a woman, but a 'woman.' To perceive Camp in objects and persons is to understand Being as Playing a Role. It is the farthest extension, in sensibility, of the metaphor of life as theater."

—"Howard Bromberg"



New DATA/620

the one computer built to match!

DATA/620 MATCHES YOUR SYSTEM. Specifically designed to meet your system requirements with such useful features as Micro-Exec for sub μsec processing rates, Independent GP Memories for simultaneous operation, two memory access channels, 555KC and 200KC, six types of I/O facilities and up to 64 devices.

DATA/620 MATCHES YOUR ENGINEERING. DATA/620 provides engineers flexibility with 16 or 18 bit words, 1.8 μsec core memories, 9 hardware registers, 0 to 12 v. logic levels, universal bus organization, modular construction to fit your configuration — rack mounts in 26¼ inches of space.

DATA/620 MATCHES YOUR BUDGET. Designed for economical operation in today's and tomorrow's systems, DATA/620 has FORTRAN in 4K, Assembler, Aid, Maintain — all open ended. And, DATA/620's low initial cost of $27,900 and low operation costs will not break your system budget.

We have the complete DATA/620 story in a fact filled 36 page brochure that we'll be happy to send you immediately. Or better yet, we'll deliver a DATA/620 . . . in 90 days.

DATA MACHINES

1590 Monrovia Avenue, Newport Beach, California Telephone (714) 646-9371 • TWX (714) 642-1364

Division of DECISION Control, Inc.

INSTRUCTION BY COMPUTER

by H. A. SCHWARTZ and H. S. LONG

One of the most pressing problems facing modern industrial society is that of providing the necessary quantity and quality of education to its members. Much of the burgeoning rate of technological advance giving rise to this educational problem was brought about by the development of computers. Consequently, it would seem only poetic justice that we set the computer to the task of alleviating the situation. Of course, if these motives were the only bases for the application of computers to the field of education and particularly for the development and investigation of computerized teaching devices, these practices would long since have been discontinued. The continuously expanding activity in this area shows that computer-assisted instruction appears to offer some distinct advantages over many present techniques of instruction. This paper will explore some of these advantages.

programmed instruction

To appreciate fully the potential of computer-assisted instruction, it will be valuable to look first at the general field of programmed instruction (PI). PI texts, properly written, tested and administered are valuable instructional devices and have, on many occasions, demonstrated the ability to provide standardized instruction, decentralized training, individualized pacing and other significant benefits. However, such educational roses are not without thorns. In general, the problems encountered in the use of PI texts fall into three categories: administration, production, and data collection.

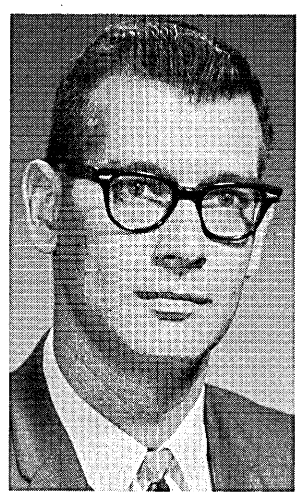
Administratively, PI has some shortcomings—first of all the freedom, in terms of portability, of the PI text permits the student to study at home or almost anywhere. Yet, experience has taught us that this same freedom may be the student's undoing, since he is also free *not to study*. This often leads to a study schedule consisting of long periods of inactivity, punctuated at test times by intense study or cramming—a schedule seldom conducive to retention of the material. Secondly, as PI technology progressed, texts which employed such features as branching, skipping, etc., were produced to provide more individualized instruction. Thus, a student who had answered a particular question correctly might be instructed by the text to skip ahead in the program, while the student who answered incorrectly would, presumably, go on to the immediately succeeding item. In practice, however, it was found that a student, upon missing one of these questions, might often remark, "Oh yes! I knew it all the time. It just slipped my mind," and would then proceed to skip over the needed material.

Finally, from an instructional point of view, the method of determining the accuracy of a student's answer in PI texts leaves much to be desired. In general, when the student has answered a question in a PI text, he verifies his answer by exposing the correct answer. If his answer were correct, all is well and he proceeds to the next question. However, if his answer were incorrect, he has seen the correct answer and cannot pretend he doesn't know it when he goes back and answers the question again. He has thus been robbed of the opportunity to try to think his way through that question. Branching texts circumvent this problem, but in so doing they produce another by employing a multi-choice format, which makes the student's task one of recognition rather than recall.

The problems of data collection and production (including required revisions) are closely interrelated. In operational, high-volume situations, it is extremely difficult, if not impossible, to make changes in a PI text after it has been released. Indeed, one of the criteria used to determine whether a particular course is suitable for programming is its stability. The consequence of this lack of ability to change the program is that an extreme amount of care must be taken in the original production. This requires a considerable amount of testing prior to release. However, in order to collect useful data during the individual testing phase, the author must distort the student's environment. In the actual study situation, the student may be working on the text at home, on the beach, or on a train. But during the testing phase the only way to obtain an adequate record of the student's reactions is to have the student study under the observation of the author. Moreover, once the text has been released, there is simply no means of collecting data beyond that of having the student maintain a diary of time spent, his criticisms, or some other such form of report.

Thus, data collection, production, and administration have been major shortcomings of PI texts and consequently were some of the initial obstacles to be overcome if computer-assisted instruction was to be considered a serious contender in the area of self-instructional techniques.

Now then, let us turn our attention to the topic at hand—computer-assisted instruction. To begin with, let us first describe the system now being employed within the Field Engineering Division of IBM to test the general feasibility of computer-assisted instruction as an indus-

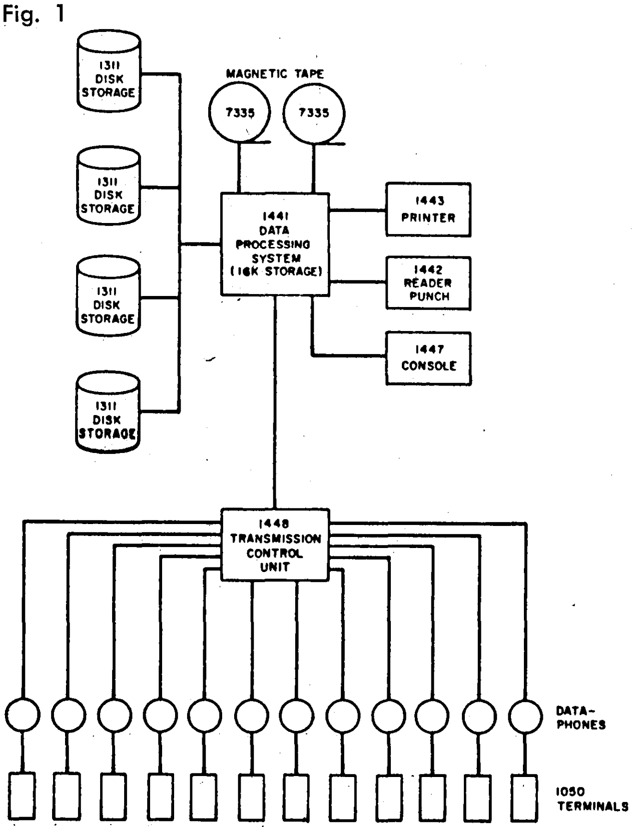


Dr. Schwartz conducts research in educational processes, techniques and media as related to CAI at IBM's Field Engineering Div. He previously was engineering psychologist with the IBM Space Guidance Center investigating skill retention for extended mission space vehicles and behavioral measurement programs for manned orbiting laboratories. He holds a PhD in experimental psychology from Johns Hopkins Univ.

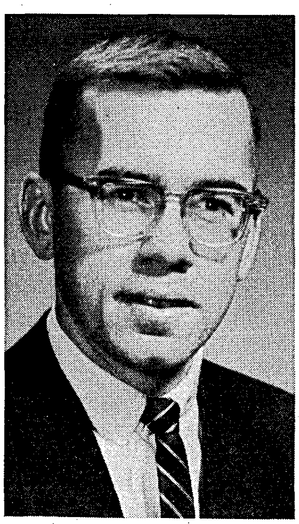
trial training technique, keeping in mind that the optimal or eventual system may bear little physical resemblance to that presently in use.

At the present time, the vehicle for computer-assisted instruction is the IBM 1440 computer system, configured as shown in Fig. 1. Each component is standard data processing equipment and none was designed specifically

Fig. 1



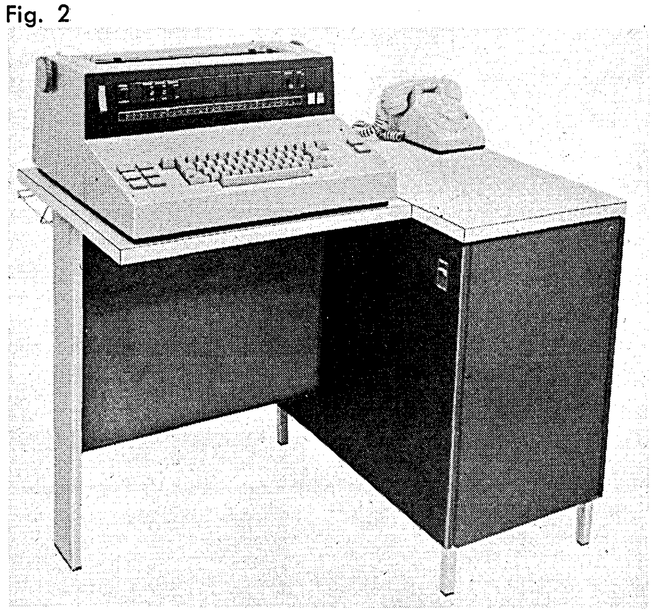
for this application. The IBM 1050 data communications terminal, shown in Fig. 2, serves as the input-output device for use by both authors and students. Presently, up to 24 students and/or authors can operate simultaneously on the same or on different courses. The limiting factor in adding more terminals to the system appears to be the amount of time delay that the student will tolerate between his response and the subsequent reply of the system. Courses are written by the author in an easy to learn language known as Coursewriter. (Coursewriter is an announced IBM program, designed to permit instructors unfamiliar with computers or with computer programming to write courses for computer-assisted instruction.) The course material may be entered into the system via the 1050 terminal and is stored on IBM 1311 disc units. IBM 7335 tape units are used to store records of the student's responses.



After several years as a mathematics instructor, Mr. Long is currently directing experimentation in the use of CAI in customer engineering at IBM's Field Engineering Div. in Poughkeepsie, N.Y. The inventor of the modified 056 Verifier (self-instructional device), he received a B.S. in math and science from Millersville State College and an M.S. in applied math from Carnegie Tech.

Prior to taking a computer-assisted instruction course, each student must be properly enrolled, or registered in that course, at which time he is assigned a student number. This number serves several purposes. It identifies the student, and allows the system to collect and maintain a record of his correct and incorrect responses, free comments, dates and times of study sessions, requests for help, and the exact sequence of material taken. The student

Fig. 2



number also tends to prevent unauthorized personnel from attempting to operate the system. Finally, the student number acts as a "bookmark" so that when the student returns at some later date, the system returns him to the point at which he previously left the program.

In operation, the student reports to his computer-assisted instruction terminal location and, using the telephone data set, dials the centrally located computer. When the communication link has thus been established, the student types "sign on." Fig. 3 shows a sample sequence from a demonstration program as it would appear to a student.

It should be noted, of course, that the sequence illustrated in Fig. 3 is only one of a number of possible sequences; that is to say, the material comprises a branching program. However, in contrast to the branching programs presented in PI texts, this program incorporates both constructed and multiple choice answers and, more important, can indicate to the student that he is incorrect and possibly even give him specific cues without revealing the correct answer.

In Fig. 3, line 47, the student has indicated that he requires a definition before proceeding. He orders the system to perform as a glossary. To mark the student's place so that he can later return, the system replies with the student's present location in the program. When the student has obtained the desired information, he can return to his previous place by ordering the system to go to that location.

In the event that the student feels totally unable to arrive at the correct answer to a question, he can type the word "help" (as shown on line 59). The system will respond by indicating the location of the question on which the assistance was requested and will then type the correct answer to the student. Before being permitted to go on, the student must first reproduce the correct answer, thus ensuring his attention to it.

At line 53 the student has offered a free comment by prefacing his remarks with the symbols. This allows the system to be interrogated later specifically for such free comments.

Two additional things should be mentioned about Fig. 3. First, had the student correctly answered both of the major test questions (the square of 55 and the square of 35), the program would not have ended but would rather have gone on to ask the student to try his hand with a three-digit number. If a student were to answer this advanced question correctly, the system would treat him as a superior student and would go still deeper into the subject by asking him to try to identify and finally to construct the algebraic expression which underlies the use of this shortcut method of squaring numbers ending in 5.

Secondly, note that to the question: The square of 55 is - - - - - , the student incorrectly answered 30625. (See lines 56 and 57, Fig. 3.) The system responded with sympathy and with a request that the student try again, but unfortunately for the student, there was no real cue or remedial information. This is simply due to the fact that the answer 30625 was not anticipated by the author of the course as an answer to the question, and thus no specific reply was included in the course. In a moment we shall see how such a situation might be rectified.

Fig. 4 illustrates a portion of the course we have been discussing, but as it would appear from the standpoint of the author entering course information from his own 1050 terminal. The numbers in the left-hand column are printed by the system during the input operation and merely indicate successive lines of input. The two-letter groups in the second column are mnemonic operation codes used in the Coursewriter language. Thus, the mnemonic *qu* indicates to the system that the author wishes to present the immediately following material to the student and then to await a response from him. The mnemonic *ad* instructs the system to add some value to a specific counter. This enables the system to keep track of the number of times a particular question is asked, the number of correct or incorrect responses given by the student, etc. The mnemonics *ca* or *cb* indicate, respectively, the correct answer and the acceptable alternative correct answers anticipated by the author. The mnemonics *wa*

Fig. 3

```
                    Student Printout
                          for
             Number Squaring Demonstration
*
1.      sign on
2.  type course name
3.      amdnosq
4.  type your number
5.      s0360
6.  your name is _____.
7.      h. s. long
8.  We are now prepared to help you learn a very quick method of squaring two-digit
9.  numbers that end in five (such as 35). However, we don't want to bore you by presenting
10. material you already know. Therefore, please answer the following question. Do you
11. already know a quick method of squaring numbers such as 45, without using pencil and
12. paper?
13.     lots of them
14. Entry error. Please answer with a yes or no.
15.     yes
16. Good for you. I'm glad to hear you already can perform this operation. However, I'm
17. going to ask you to prove it.
18. Question: The square of 35 is _____.
19.     625
20. I'm sorry, but you missed that question. It looks like you could profit from an exposure
21. to this material. So here we go.
22. Do you know what the successor of a number is? Answer yes or no.
23.     no
```

Fig. 3 (Cont'd.)

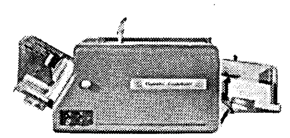
```
*
24. When counting, the number which follows another is called the successor of that number.
25. Example: The successor of 3 is 4.
26. Question: The successor of 25 is _____.
27.     24
28. No. The successor of a number is one greater than that number. Your answer is one
29. less than that number. Now answer once more.
30.     26
31. Now you're getting it!
32. The successor of 87 is _____.
33.     88
34. Now for a bit more involved practice.
35. Eight times the successor of eight is _____.
36.     64
37. No. The successor of 8 is 9. Thus, the product of 8 times its successor is _____.
38.     72
39. That enough about successors. Now let's square.
40. To square a two-digit number ending in 5 we can:
41.     1. Multiply the left-most digit by its successor.
42.     2. Write 25 after the result of Step 1.
43. Example: The square of 25 is
44.     1. 2 times 3 (left-most times successor) = 6.
45.     2. Write 25 after the product of Step 1 = 625.
46. Question: The square of 55 is _____.
47.     go to glossary
48. You are leaving provesucc 133.
49. Enter the word you wish defined. Use lower case.
```

Fig. 3 (Cont'd.)

```
*
50.     successor
51. successor: when counting, the number which follows another is called the successor
52. of that number.
53.     ###I was confused on the definition
54.     go to provesucc-133
55. You are leaving glossary s 2
56. Question: The square of 55 is _____.
57.     30625
58. Sorry about that, but your answer is not correct. Try again.
59.     help
60. provesucc 133
61. 3025
62.     3025
63. Very good. There is one more question. The square of 35 is _____.
64.     1225
65. There you go!
66. Thats the end of our little program. Thank you for your attention.
```

* These numbers do not appear in the actual student printout. They are shown here for the convenience of the reader.

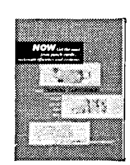
Who Cares?



There's no need to point an accusing finger at your customers, stockholders, or employees when a Cummins Carditioner is part of your punch card data processing or computer operation. The Carditioner lets you ignore the "do not mutilate, staple or fold" warning on punch cards—no matter what condition they're in. It's the one machine that can speed up computer input and balancing. At the rate of 250 cards per minute the Carditioner puts them back in processing shape, at once eliminating three-quarters of your repunching and reverification requirements—and the errors accompanying repunching. And the Carditioner handles 51, 66, 80 and 90 column cards. It's available on rental, lease or purchase. Get the details from your local Cummins representative.

REQUEST THIS BOOKLET . . . it tells all about the Cummins Carditioner, the machine that completely reconditions mutilated punch cards.

CA SINCE 1887 **Cummins** IN BUSINESS AND BANKS ● SINCE 1887
CUMMINS-CHICAGO CORPORATION
A CUMMINS-ALLISON COMPANY
4740 N. RAVENSWOOD AVE. • CHICAGO, ILLINOIS 60640
SALES AND SERVICE IN ALL PRINCIPAL CITIES



CIRCLE 31 ON READER CARD

or *wb* perform a similar function for incorrect answers. (Note: A recent function developed for the Coursewriter permits the author to instruct the system to ignore such variations as upshifts, downshifts, spaces, etc., when establishing alternative answers. Other functions permit the author some latitude in dealing with the possible varieties of misspelled and/or multiword answers.) The mnemonic *ty* instructs the system to type, to the student, the immediately following material whenever one of the preceding correct or incorrect answers is given. The mnemonic *un* indicates a message to be typed to the student whenever he responds with an answer that does not match one of the preceding ca, cb, wa, or wb answers. More than one un message may be included for a given question and these will be presented to the student in order. Thus, the first time a student responds in an unanticipated manner to the question at line 130, the un message at line 153 will be typed to him. The second unanticipated answer by this student will cause the un message at lines 156 and 157 to be presented to him, and if he then gives a third unanticipated answer, the un message at line 159 will be presented to him.

Relating Fig. 4 to Fig. 3, note that the student's answer, 30625, given in Fig. 3, is not listed as a specifically anticipated incorrect answer by the author in Fig. 4. Thus, when that answer was given by the student, the first un message (see line 153, Fig. 4) for that question was typed back to the student.

The ability of the system to collect student data is illustrated by the Student Response Report shown in Fig. 5. Every response made by the student as he proceeds through the course can be stored by the system. These data are available on command to any authorized person according to any desired combination of the categories listed at the top of the report. Thus, it is possible to request data for all courses or for any specific course, for all students or for specific students, and for all operations (ca's, cb's, wa's, wb's, un's, or free comments) or for any specific operation. In addition, this information can be requested for any or all questions or sections of the courses. In the report shown in Fig. 5, we have requested all the operations for all the questions in course amdnosq for student number s0360. Thus, if we look at the fourth line from the bottom of Fig. 5, we find, reading from left to right, that in course amdnosq, student s0360, on April

Fig. 4

```
                    Author Listing
                         for
             Number Squaring Demonstration

130 qu  Question: The square of 55 is _____.

131 ad  +1//c1
132 ad  -c4//c4
133 qu

134 ca  3025
135 cb  three thousand twenty five
136 ad  +1//c4
137 ad  +1//c4
138 wa  2525
139 wb  two thousand five hundred and twenty five
140 wb  twenty five twenty five
141 ty  You squared the left-most digit. This is not correct. You
142     must multiply the left-most digit by its successor, then write
143     25 after your answer. Try again.

144 ad  +1//c3
145 ad  -3//c4
146 wa  2530
147 wb  two thousand five hundred and thirty
148 ty  You multiplied the right-most digit by its successor. You should
149     multiply the left-most digit by its successor then write 25 after
150     your answer.
151 ad  +//c3
152 ad  -3//c4
153 un  Sorry about that, but your answer is not correct. Try again.

154 ad  +1//c3
155 ad  -3//c4
156 un  The left-most digit is 5. This time its successor (6) is 30.
157     Write 25 after that number. Try again.

158 ad  +1//c3
159 un  I do believe you are having difficulty. Consult your instructor!

160 ad  +1//c3
161 qu  Very good. There is one more question. The square of 35 is
162     _______________.
```

Fig. 5

STUDENT RESPONSE REPORT

Course	Student	Date	Response (Label)		Item (Label)		Response Type	Response
amdnosq	s0360	4/19/65	beginsq	035	beginsq	6	un	lots of them
amdnosq	s0360	4/19/65	beginsq	016	beginsq	6	cb	yes
amdnosq	s0360	4/19/65	qusqutsti	017	qusqutsti	3	un	625
amdnosq	s0360	4/19/65	successor	011	successor	3	ca	no
amdnosq	s0360	4/19/65	provesucc	014	provesucc	3	wa	24
amdnosq	s0360	4/19/65	provesucc	004	provesucc	3	ca	26
amdnosq	s0360	4/19/65	provesucc	059	provesucc	58	ca	88
amdnosq	s0360	4/19/65	provesucc	117	provesucc	97	un	64
amdnosq	s0360	4/19/65	provesucc	098	provesucc	97	ca	72
amdnosq	s0360	4/19/65	provesucc	153	provesucc	133	go	go to glossary
amdnosq	s0360	4/19/65	glossary	002	glossary	2	go	go to provesucc 133
amdnosq	s0360	4/19/65	provesucc	153	provesucc	133	un	###I was confused on the definition
amdnosq	s0360	4/19/65	provesucc	153	provesucc	133	un	30625
amdnosq	s0360	4/19/65	provesucc	134	provesucc	133	un*	3025
amdnosq	s0360	4/19/65	provesucc	134	provesucc	133	ca	3025
amdnosq	s0360	4/19/65	provesucc	162	provesucc	161	ca	1225

Student requests for help are coded by the system as un* to distinguish them from ordinary unanticipated responses. Thus the entry shown here is the record of the student's response of help shown in Figure 3.

19, 1965, made a response at the course location indicated by the label, provesucc 153. This response, which was made to a question which occurred at course location provesucc 133, was not an anticipated response in the course. The actual response made by the student was 30625.

Fig. 6

```
                          Student Record Report

enter request
see ss

enter course & student
type as
course//student

amdnosq//s0360

STUDENT RECORD REPORT AS OF 12/02/65
STUDENT          NAME    OFF   COURSE     TIME    DATE    LOCATION    RCD
s0360        h. s. long  h73   amdnosq    06.00   12/02   lastout  1
enter request




sign off
you have been signed off.
```

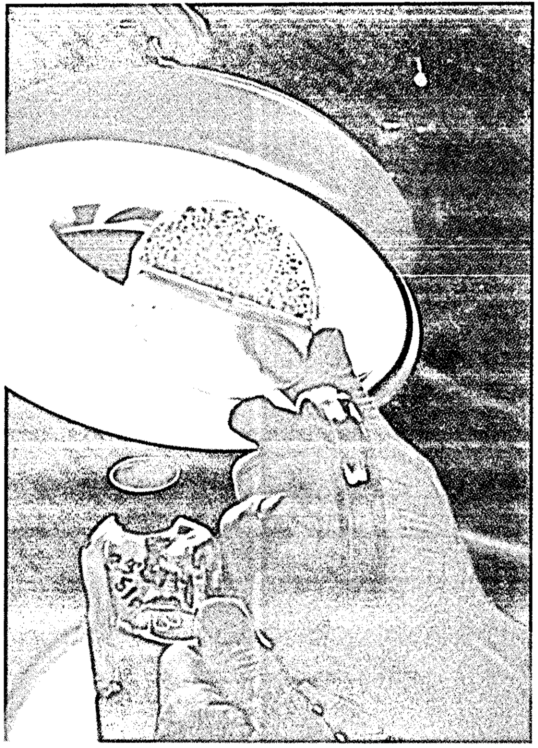
For determining problem areas in courses, the report in Fig. 5 is extremely valuable. However, for administrative and control purposes, the student record report shown in Fig. 6 is desirable. As with other reports, the information can be requested for any combination of courses and students. In Fig. 6 we have asked for the progress of a particular student enrolled in course amdnosq. Looking at the first line and reading from left to right, we find that our student, s0360, whose name is h. s. long and who works in office number h73, is enrolled in course amdnosq and, as of the requesting date (12/02/62) has spent a total of 6.0 hours* on the course. He last studied at the terminal on 12/02/65 and, when he last signed off, he was at location lastout 1. By periodically obtaining and comparing data such as these, it would be possible for a remotely located monitor to maintain a fairly continuous check on the students' rates of progress and to detect any problems in their adherence to their assigned schedules of study.

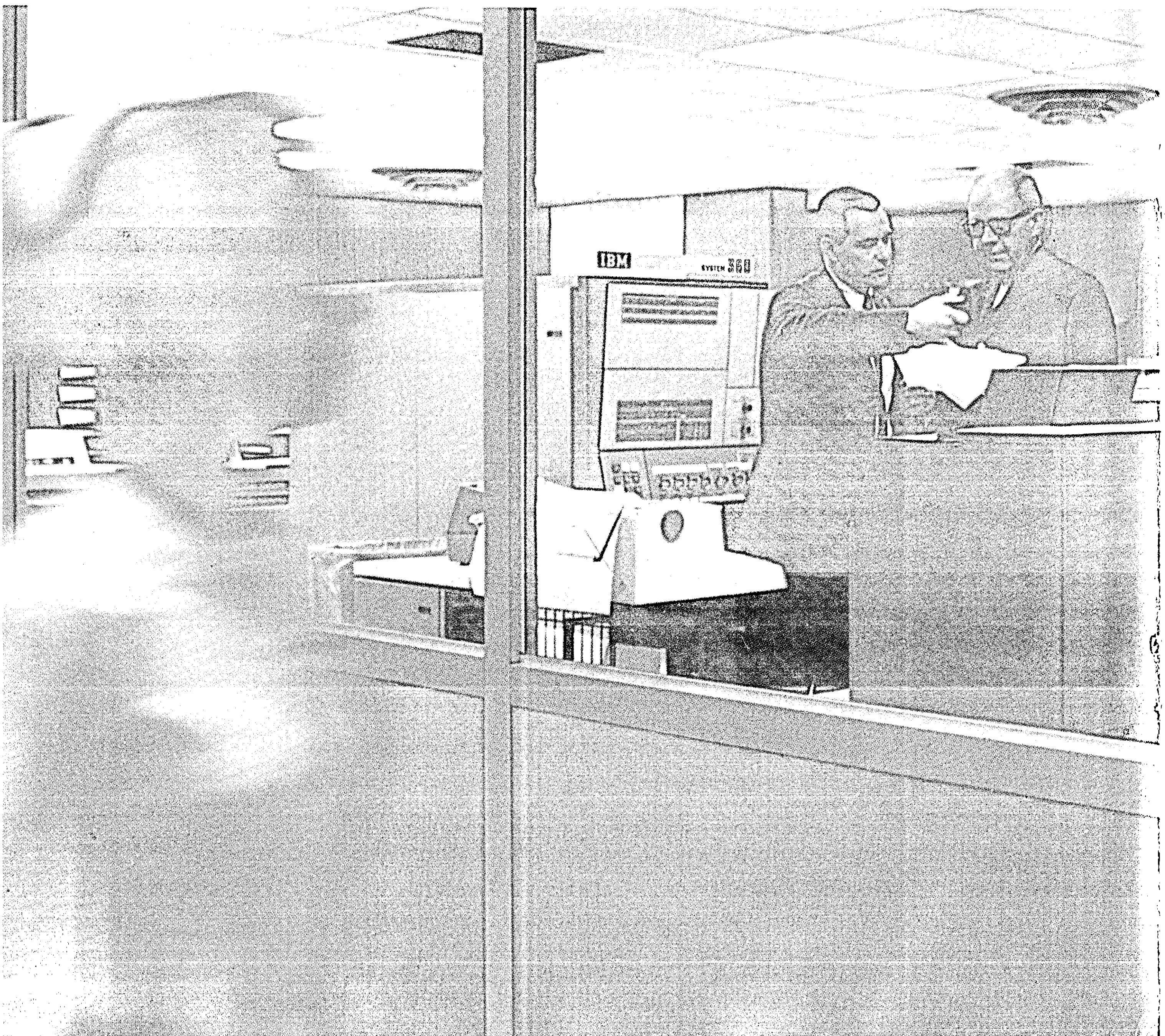
Now, to turn our attention to the problem of making changes in programs, let us assume for the moment that a Student Response Report of the type shown in Fig. 5 has revealed that not just one, but rather a number of students, had given the incorrect response, 30625. To the author of the course, such a situation would indicate that the students are somehow being led astray. As would an instructor in a classroom situation, the course author should be prepared to offer specific remedial material to explain to the students why their answer is incorrect. Fig. 7 illustrates the procedures involved in making a change in the course to accommodate this situation. Authors, like students, must first sign on and identify both the course and themselves before being permitted to continue. Authors, however, are registered with a specific code, which, unlike the codes given to students, allows them to create and/or alter course material. After the author has correctly identified himself, the system asks the author to type the control word. The control word is essentially

* Although the amdnosq course is purely a demonstrational program and takes only a short time to complete, student s0360 has presented the demonstration a great many times and therefore the 6.0 hours does not, in this case, reflect the actual time required for the student to complete the program.

From the original painting by Neil Boyle

FOR IBM CIRCLE 34 ON READER CARD →




IBM
SYSTEM 360

At Monmouth Medical Center, SYSTEM/360 checks in–checks out...

George Bartel, Hospital Administrator, has big plans for Monmouth Medical Center, Long Branch, New Jersey. That's why SYSTEM/360 is there.

Just a few weeks after its arrival, SYSTEM/360 took over patient billing, and went on to automate all administrative paperwork.

Nine more days, and SYSTEM/360 converted the process to direct-access disk files. An easy transition with IBM SYSTEM/360 Basic Operating System (BOS/360 for short).

After information is entered, it can be processed in microseconds.

This opens an entirely new realm of operations—a real-time Hospital Information System.

A complete information hookup is planned. Keyboard communications will connect SYSTEM/360 to vital areas—nurses' stations, X-ray, pathology, pharmacy and others.

New information can be entered immediately. Services ordered and validated. Treatments reviewed by instant printout. Drug control data displayed. Resources scheduled and allocated.

The goal—the most efficient use of facilities and time.

Monmouth Medical Center is growing. And SYSTEM/360 is ready to meet this growth, because the System's modular design makes it easy to expand, to solve more problems, to take on more information-handling chores.

Monmouth Medical Center likes its IBM SYSTEM/360.

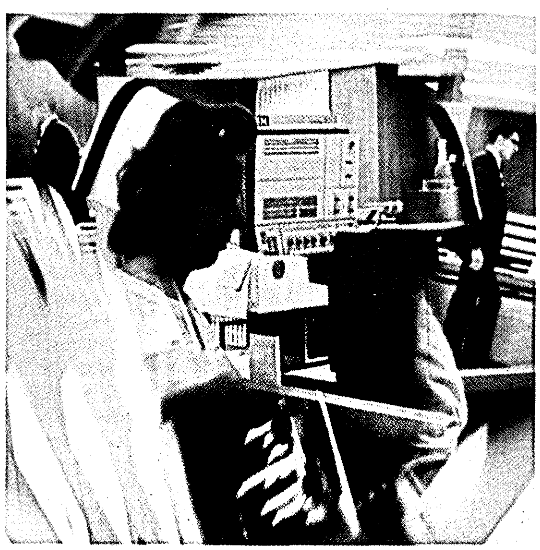
So do hundreds of other companies in all kinds of industries.

They like its performance, speed and versatility.

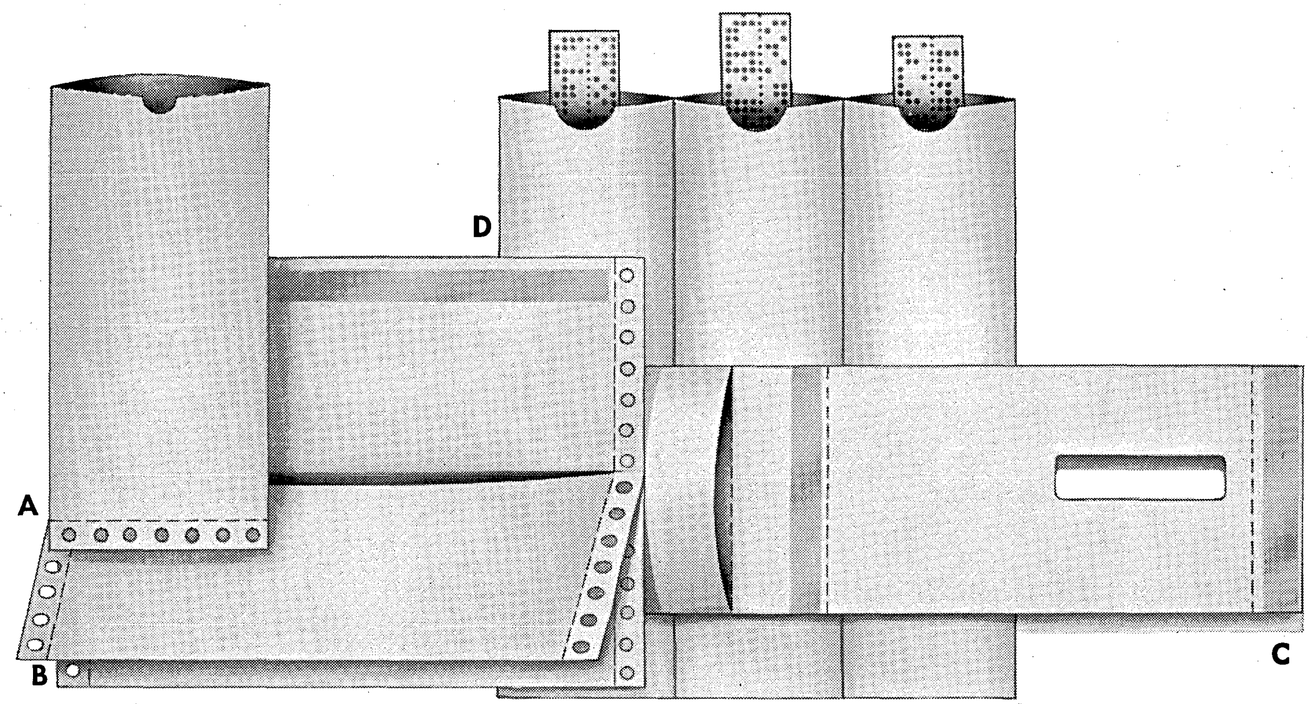
We think you will, too.

IBM®

and starts a new era.



Hold everything.



the command for the operation that the author wants the system to perform. The author then types, type provesucc-152//provesucc-152. This instructs the system to type out the information stored at that course location. The author requested this information merely to make certain that he was at the proper place in the program before making the desired alteration. After satisfying himself as to his course location, the author then instructs the system to insert 30625 as an anticipated incorrect answer (wa), and to follow that answer with a ty statement which will provide a specific cue to the student.

Fig. 8 shows a student's version of that portion of the program as it appears after the course alterations. The blocked-in portion of Fig. 8 should be compared with lines 56-58 of Fig. 3, which shows the program prior to change. In the prior version, the response 30625 evokes only a general statement that the student is wrong. In the updated version, the same response now provides the student with a specific explanation of his error.

The data collection and the quick update capabilities of the system have profound implications for the construction of instructional programs. Computer-assisted instruction would appear to permit an author to write the initial version of his course without the extreme attention to detail and extensive pretesting which characterized the production of P.I texts. Once this version has been written and administered to a small number of students, the author can interrogate the system and obtain a complete record of each student's performance. From this, he can then determine where changes are necessary. He can then, from his terminal, produce the desired course changes

Fig. 7 and 8

```
Example of Student Printout after Revision
shown in Figure 7

    72
That's enough about successors. Now let's square.
To square a two-digit number ending in 5 we can:
    1. Multiply the left-most digit by its successor.
    2. Write 25 after the result of Step 1.
Example: The square of 25 is
    1. 2 times 3 (left-most times successor) = 6.
    2. Write 25 after the product of Step 1 = 625.
Question: The square of 55 is ______.
    30625
Good try, but you appear to have misread Step 2 of the rule. Don't write down the
square of 25, but rather 25 itself. Now give it another try.
    3025
There is one more question. The square of 35 is ______.
    1225
There you go!
That's the end of our little program. Thank you for your attention.
```

and from that moment all students taking the course will receive the updated version. His actual course changes may consist of additional material, deletion of unwanted material, changes in sequence, changes in score-keeping and in branching criteria, etc. It should also be possible to prepare courses in somewhat unstructured areas, such as troubleshooting, by presenting the problem and having numerous students type out, as a free comment, their approach to the problem. These comments could then be

analyzed and categorized, and placed in the course as anticipated correct or incorrect answers.

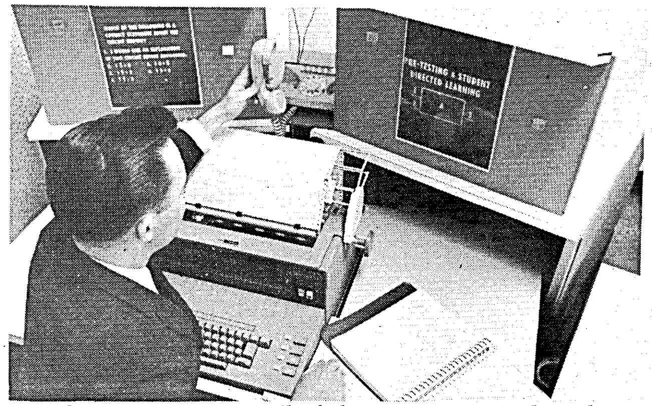
We began this paper by pointing out that programmed texts represented an important advance in educational technology, but that there were some significant shortcomings, primarily in the areas of data collection, course construction, and course administration, which discouraged their use in many situations. We have attempted to show how computer-assisted instruction might be employed to meet these shortcomings.

It is obvious, however, that no firm conclusions can be drawn in the absence of empirical, substantiating data. Moreover, these data, particularly in the realm of industrial training, are not obtainable through the use of short demonstration programs administered under controlled, laboratory-type situations.

As an attempt to provide some of the requisite data concerning the efficacy of teaching rather extensive amounts of technical material in an actual working environment through remote computer-assisted instruction, a pilot study was recently initiated by the Advanced Maintenance Development group of IBM's Field Engineering Division. IBM customer engineers, located in Washington, D.C., Philadelphia, Los Angeles, and San Francisco, report for two-to-four hours per day, five days a week, to an IBM terminal located in their vicinity. During the remainder of their working days, they are available for their normal duties.

At the terminals in Philadelphia and Los Angeles, the student, at the direction of the system, refers to an accompanying notebook containing illustrations, diagrams, explanatory text, etc. For example, the 1050 terminal might then type to the student, "Turn to Fig. 48". The terminal might then type out a question relevant to that reference, or the question might be found in the reference itself. In San Francisco and Washington, the same material is presented; however, the presentation is made via an experimental display device that contains the material stored as film chips, and presents them, under the control of the computer, on either of two 10 x 13 screens, as shown in Fig. 9. The purpose of this situation is to determine

Fig. 9



whether computer control of the course material might improve or impair the student's performance or acceptance of computer-assisted instruction. It prevents the student from accidentally turning to the wrong page in the notebook, and consequently, when the diagrams tend to be similar, from giving answers which the system rejects as incorrect. It also prevents the student from browsing through or

quickly reviewing the material.

The students in the pilot study are customer engineers who are scheduled to attend an extensive course in the servicing of the IBM System/360, presented at a centralized education center.

The computer-assisted instruction courses being employed in the study are:

1. CE 360 Pre-school—this course is designed to provide the customer engineer with certain basic information prior to his arrival at the centralized education center. It covers such topics as numbering systems, solid logic components, programming systems, etc.
2. Fundamentals of Data Processing—this course, as the name implies, is designed to provide a general introduction to data processing systems for those who, according to their performance on a computer-presented pretest, require this information before going on to the CE 360 Pre-school course.

Both courses are available in book form. The Fundamentals of Data Processing course is available as a programmed text. The CE 360 Pre-school course is available as a self-study text within IBM, and is the presently operational form of instruction in this subject matter. The performance of the students using these self-study texts will serve as one of the general guidelines for evaluating the performance of the computer-assisted instruction students in the pilot study. However, it must be borne in mind that studies of this type are to be considered as exploratory—the prime purpose is that of determining the feasibility of a new technique rather than its superiority or inferiority to a specific present technique. During the early days of PI, the countless studies comparing "this machine with that text" or "this text with that classroom" served well to point out the futility of attempting to arrive at generalizable conclusions from the comparison of specific, nonrepresentative methods of instruction.

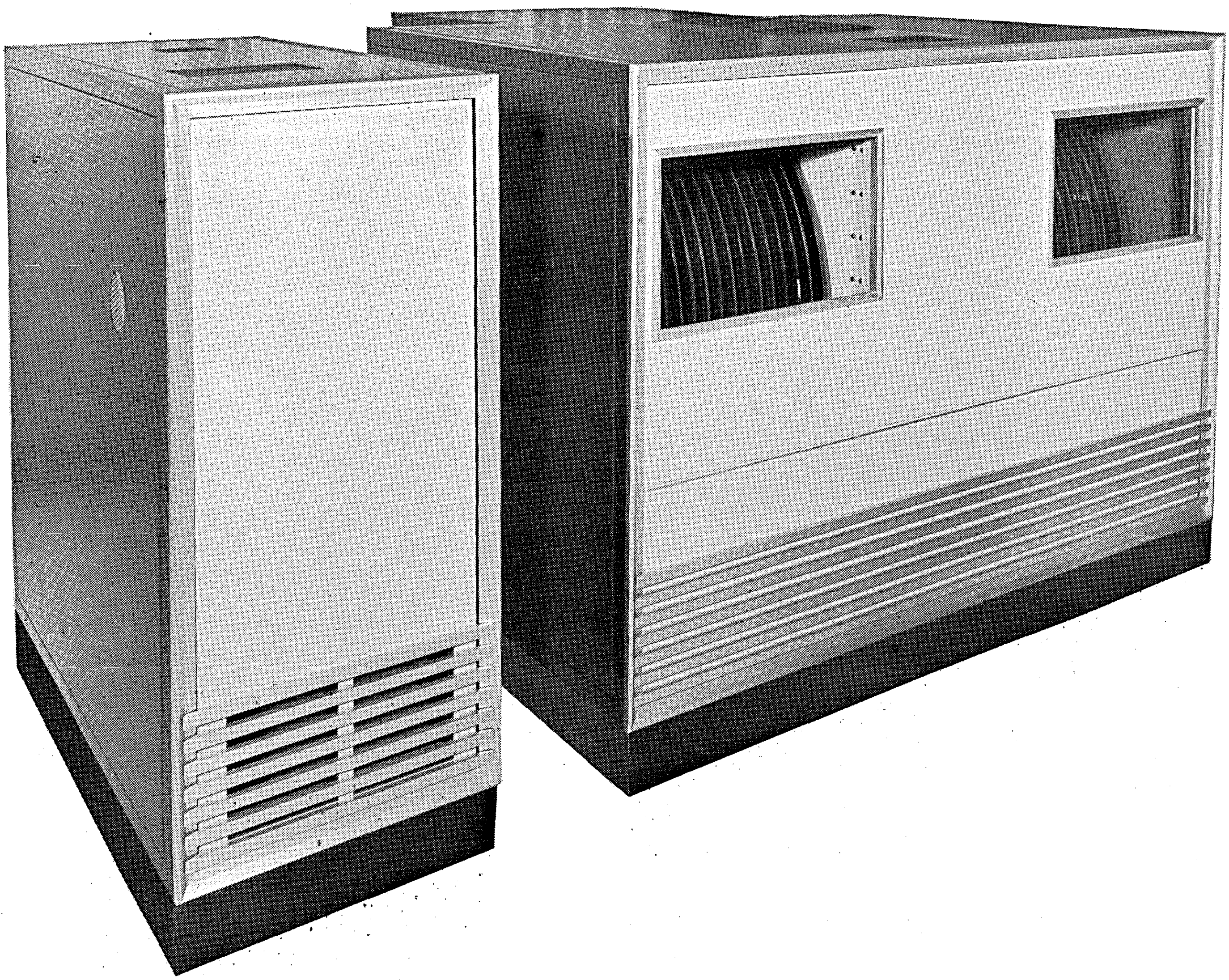
The data being acquired in the study are:

1. Final examination scores
2. Total time to complete the courses
3. Attitudinal data (from post-course questionnaires completed by each student)

While the results to date are encouraging, it would be premature to do more than merely indicate that in comparison with those using the self-study texts, the computer-assisted instruction students: (1) perform equally well on the examination, (2) require less time to complete the material, and (3) rate this method of instruction as preferable to self-study texts, but qualifiedly less preferable than a regular classroom presentation.

In dealing with the topic of computer-assisted instruction, we have endeavored to remain within the scope of present activity. We have not attempted to present the system described herein as an ultimate, or even a highly polished instructional system for industrial use. Nor have we attempted to extend our discussion to those potential, but as yet untried, computer-assisted instruction operations such as testing, retention training, information retrieval, construction of PI texts, simulation, or others.

It is the authors' point of view that progress in the application of computer technology to the instructional process is best brought about through a series of successive approximations. The system we have described in this paper should be thought of as one of these approximations. ■



A PROBLEM THAT TEACHES COMPUTING

by FRED GRUENBERGER

The problem situation described here has proved useful in the early stages of a course in computing. Such a problem should be readily defined, appear simple, and illustrate key points that the instructor wishes to make.

In Fig. 1, there is shown an array of all the proper functions in lowest terms. (It appears in many advanced algebra books as part of a discussion of countable infinities.) By such an arrangement, every possible proper fraction appears just once and in a specific position. Thus, each fraction can be assigned a position number. The fraction 1/2, for example, occupies position number 2; the fraction 1/7 occupies position number 13, and so on. The successive fractions at the left end of the rows have position numbers 1, 2, 3, 5, 7, 11, 13, 19, 23, 29, and so on.

With this much on the blackboard, a great deal of discussion can be stimulated. The aim at this point is problem definition. How is the array developed? Can we recall the definitions for "proper" and "lowest terms"? Do we recall an algorithm for reducing a fraction to lowest terms? (3/12 is easy, but what about 280/693?) Does that series of left-end position numbers lead to any conclusion? (It looks like a series of primes, but the next entry is 33,

Fig. 1. The Array of All Proper Fractions.

```
1
1/2
1/3  2/3
1/4  3/4
1/5  2/5  3/5  4/5
1/6  5/6
1/7  2/7  3/7  4/7  5/7  6/7
1/8  3/8  5/8  7/8
1/9  2/9  4/9  5/9  7/9  8/9
1/10 3/10 7/10 9/10
1/11 2/11 3/11 4/11 5/11 6/11 7/11 8/11 9/11 10/11
1/12 5/12 7/12 11/12
 .    .    .    .
 .    .    .    .
 .    .    .    .
```

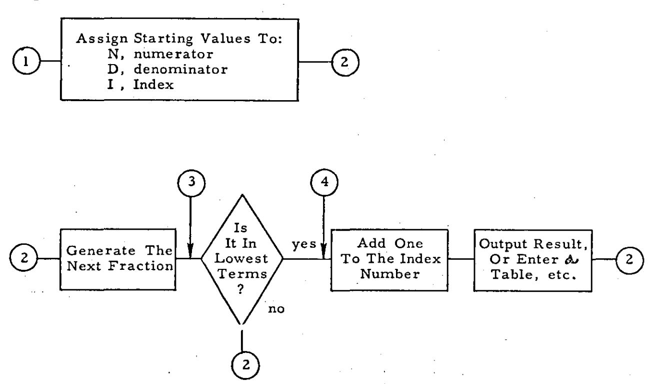
as some student should be led to discover.) Is it obvious how we could develop the next row, and the one after that? As soon as the class can call out the next fractions to be written on the board, the method for generating the array is defined, and everyone should feel fairly comfortable.

Now some problems can be stated:

1) For any given fraction in lowest terms (e.g., 19/97), what is its position number? (answer: 2825).
2) For any given position number (e.g., 10000), what is the fraction at that position? (answer: 120/181).
3) How could we program a computer to generate a table of the position numbers for the left-hand entry of each row, as well as the number of fractions in that row?

At this stage, it should be pointed out that these problems are, indeed, good computer problems. They are well

Fig. 2. The Overall Attack on the Problem



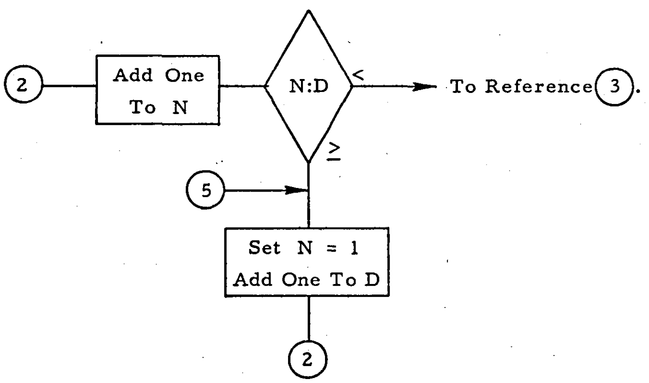
defined; a method of attack is readily developed; the problems fit any computer; there is a large amount of repetition; and there is a tremendous payoff. Indeed, on this last point, it can be pointed out that for any extensive exploration of the problem, computer implementation is the only feasible means of attack. Moreover, the problem

Mr. Gruenberger, a well-known author, lecturer and film producer in the field of computing, is presently working with customer training at Informatics, Inc., specializing in on-line systems. A frequent contributor to Datamation, he is listed in the 1966 edition of American Men of Science.

is serial in nature; two computers won't get there any faster. Serial problems (in the real world—this one is artificial) are one of the factors that contribute to demands for ever-faster computers.

The flowchart of Fig. 2 indicates a rough or broad brush approach to the problem. Several further principles

Fig. 3. Amplifying Reference 2 of Fig. 2



of good computing practice can now be shown. The logical ideas at references 2 and 3 lend themselves to subroutining; that is, the solution can be segmented.

From the broad brush viewpoint of Fig. 2, attention can be focused on the subproblems involved, and from there to possible shortcuts and ways of speeding things up.

Reference 2 of Fig. 2 is easily expanded into the flowchart of Fig. 3. The logic of Fig. 3 could hardly be improved.

But what of reference 3? A nice class discussion can be held on the concepts of "lowest terms." Since Euclid's Algorithm has not been taught in our schools, we have an opportunity to enrich the presentation by showing that elegant method, which leads to a neat flowchart. Let us illustrate Euclid's Algorithm with the fraction $^{154}/_{273}$.

The two numbers are equated in this fashion.

$$273 = 154 \cdot Q + 119.$$

That is, the smaller number times some quotient (whose value is of no interest) plus some remainder (which must be smaller than the original numerator) equals the larger number. We seek the largest number that will divide both 273 and 154. If it divides the two terms containing those numbers, then it must divide the remainder also. Hence, the original problem has been reduced to a new one involving smaller numbers: namely, 154 and 119.

So the process repeats:

$$\begin{aligned} 154 &= 119 \cdot Q_2 + 35 \\ 119 &= 35 \cdot Q_3 + 14 \\ 35 &= 14 \cdot Q_4 + 7 \\ 14 &= 7 \cdot Q_5 + 0, \end{aligned}$$

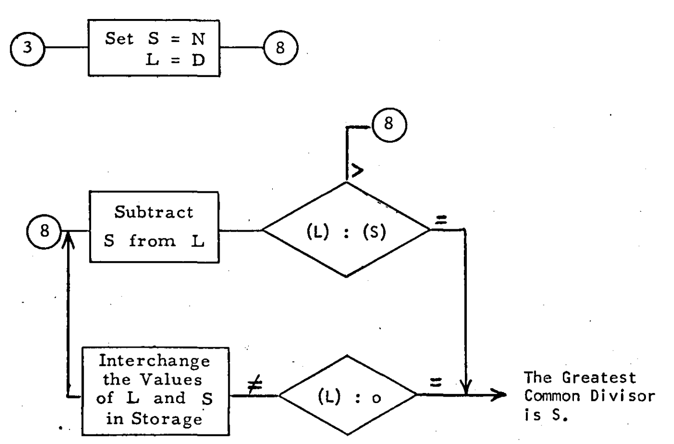
and must end eventually in a remainder of zero. When it does, the last divisor is the greatest common divisor of the original numbers. If the g.c.d. is unity, then the original fraction was in lowest terms.

The algorithm can be implemented entirely with add-type operations, as shown in Fig. 4. Even on a slow computer, the implemented flow chart of Fig. 2 runs fairly fast (say, several hundred fractions per minute, while the numbers are small).

When the numbers involved get large (the index for 1/2000 is 437467), the process slows down. Various speed-up schemes can now be considered. The Euclid Algorithm can obviously be speeded up by using division, for example. It might pay (at reference 5 of Fig. 3) to test each new denominator for primality; for a prime denominator, all values of N are valid, and much testing can be bypassed. There might be still other shortcuts possible. It should be noted in passing that the whole problem is really the calculation of the Euler phi-function, somewhat disguised.

After all this discussion, more can be learned when the flowcharts are committed to code. More shortcuts can be

Fig. 4. Euclid's Algorithm

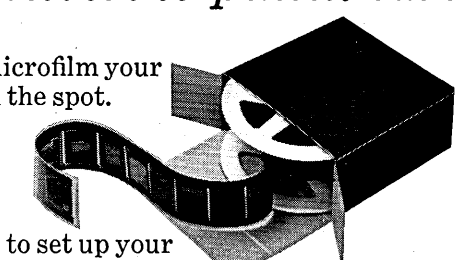


explored (an obvious one occurs when the denominator is an even number). This could well be the best place to discuss what it is we want to optimize when we use a computer to solve a problem. Is it storage space, or machine time, or programming time, or elapsed time, or what?

The entire presentation, from the construction of Fig. 1 through actual running on a computer, fits nicely into three hours. ■

Click.
Click.
Click.
Your records are ready for data processing.

Now, your records don't have to be tied up when you convert to punch cards.



We come in and microfilm your company records on the spot.

Then, we make xerocopies of your records from the microfilm. And the xerocopies are used to set up your data processing unit record system.

We meet government security standards. So confidential material stays confidential. And you get back all existing copies, plus microfilm file, if desired.

No inconvenience. No delays. No records lost. No matter how big or complicated the job is.

Just the click, click, click of the microfilm camera.

XEROX

XEROX CORPORATION, ROCHESTER, NEW YORK 14603. BRANCH OFFICES IN PRINCIPAL U. S. CITIES.
XEROX IS A TRADEMARK OF XEROX CORPORATION.

CIRCLE 45 ON READER CARD

Go ahead—clip it out. We'll send you all the details.

Xerox Record Conversion Service DA 9

Xerox Reproduction Service
P.O. Box 3300, Grand Central Station
New York, New York 10017

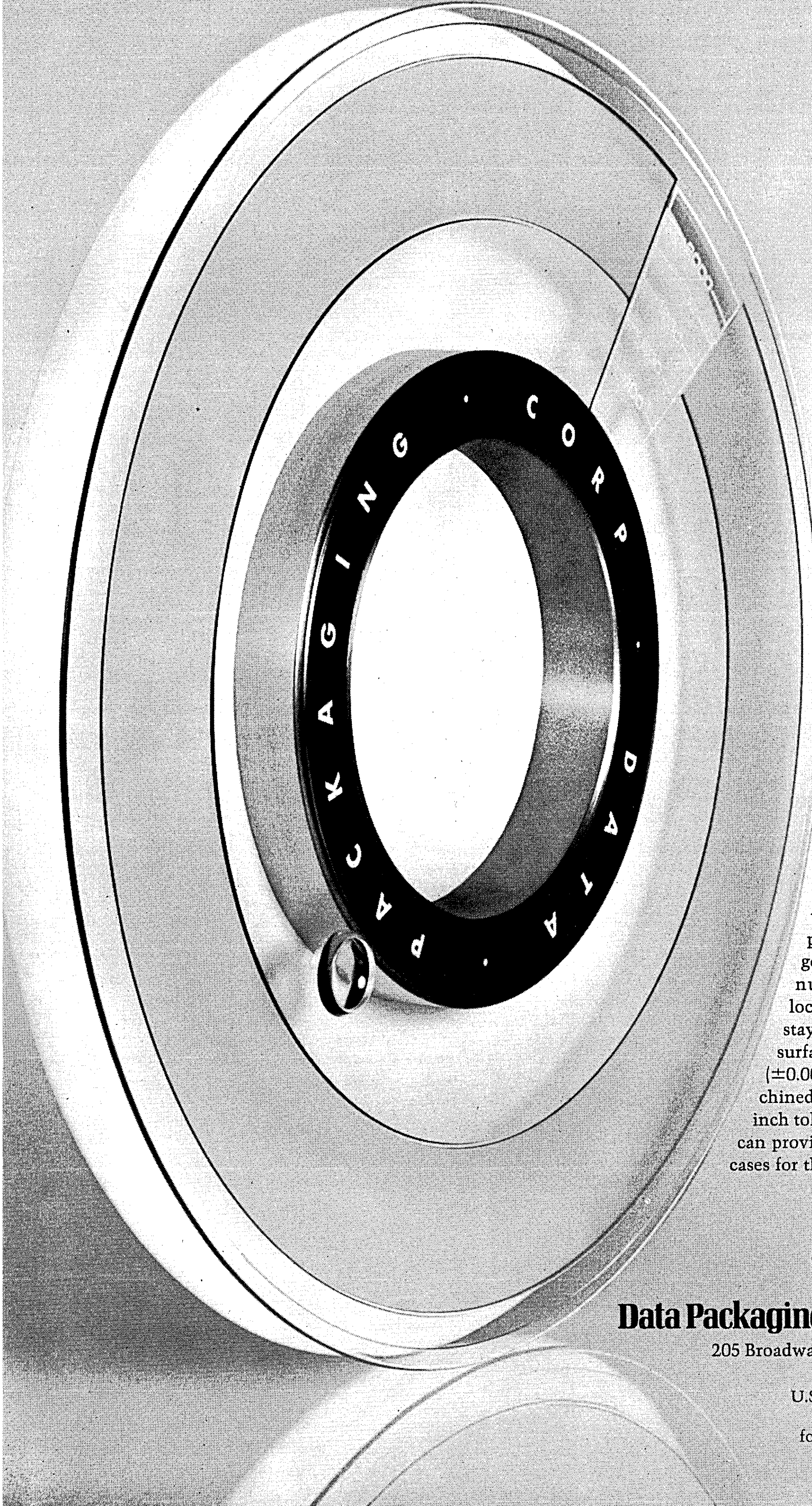
NAME ____________

TITLE ____________

COMPANY ____________

ADDRESS ____________

STATE ____________ ZIP ____________



Be choosey.

The best reels for computer tape have aluminum hubs and winding surfaces. The best way to get them is to ask for them. / This is why you should. / Aluminum winding surfaces don't compress under tape pressure, don't explode when the pressure is gone, don't distort after any number of winds. / Flanges locked to the hub are parallel, stay parallel. / Hub and winding surface are machined concentric (±0.001 inch tolerance), faces machined flat and parallel (±0.001 inch tolerance). / Every tape supplier can provide Data Packaging reels and cases for the choosey ones. Be choosey.

Data Packaging Reels and Cases

205 Broadway, Cambridge, Massachusetts

U.S.A. patent numbers are 3229928 and D202831. Other U.S.A. and foreign patents issued or pending.

news briefs

RCA'S GAMBLE SHOWS SIGNS OF PAYING OFF

When RCA went the 360 route with its Spectra 70, a lot of eyebrows and questions were raised. Could RCA *really* compete with the No. 1 on its own terms? And when IBM hardware and software slipped, many wondered how RCA could avoid the same fate.

The final answers aren't in, but early evidence suggests that RCA may have made a smart move. It appears, for instance, that the 70 can run "many" 360 assembly language programs without intervention. On a 70/45 of the same configuration as a 360/40, assemblies ran from 2-1 to 8-1 faster, according to RCA. Sorting records of various sizes, a 70/25 ran 20% faster on the average than a 360/30. A 70/45 using 60KC tapes competed with a 90KC-tape 360/50 on five assemblies. The 45 was twice as fast on three, equal on one, and 20% slower on one with a big tape library and lots of macros.

On the OS level, RCA has translated a user-produced 360/30 multiprogramming operating system to 70/45 assembly language at the rate of five man-days per 1000 instructions. It seems, therefore, that RCA *could* convert—legal questions aside—IBM software, including OS, sorts, merges, etc. . . . at least on small-to-medium systems where equivalent peripherals are available from both companies. Still to be answered: how large systems working under more complex operating systems will compare. Nevertheless, the early picture is brighter than many—RCA included—had anticipated.

As late as April '65, RCA's software outlook was far from healthy. The systems programming group was only 90 strong, and applicants weren't beating down the doors to get in. On April Fool's Day, nonprogrammer Art Carroll took over systems programming, started to define goals: 90-95% compatibility with the 360 at assembly language level, 100% COBOL and FORTRAN compatibility.

Defined between April and August were 160 different programming projects. To date, 90 have been released on or ahead of schedule, and another 25 were due out in August on schedule. Of the rest, 20-25 will be two months late at the most . . . a respectable record in a tardy-prone industry.

Still not cocky, Carroll says, "We know it's a long war." But it appears that the systems programming group, now 280 strong, has won its first battle. And with first half '66 sales approaching $100 million, RCA is confident its me-too gamble was the right one.

CEIR PRESIDENT HOLLAND RESIGNS

Policy differences over potential investment opportunities and disagreements over the future direction of CEIR are the possible causes of the resignation last month of president Robert D. Holland. Herbert W. Robinson, founder and chairman, is the new president.

Holland joined CEIR in Dec., '62, as chief financial officer, took over as head of the nationwide consulting, programming and service bureau firm in Sept. '65. He played the key role as the company divested itself of heavy machine commitments and branched out into education, leasing, service bureau franchising and time-sharing.

But the most recent financial report for the 9-month period ending June 30 showed an operating income of only $136,300 (not including $412,-900 received through the sale of two 7094's) on gross income of $17,491,-900 . . . as opposed to operating income of $1,022,700 on gross income of $15,204,200 for the previous year's first nine months. (These figures do not include special gains for '65 and '66, respectively, of $544,000 and $1,-234,500.)

Holland will resume edp financial consulting in D.C.

Other changes include the appointment of Robert G. Dee to group vp (he'll be in charge of franchise activities, computer time sales, the pilot time-sharing program and business services); Oscar Lurie, a vp who heads

FIRST IBM INSTRUCTION SYSTEM SHIPPED TO STANFORD

Stanford University's Institute for Mathematical Studies in the Social Sciences is installing the first IBM 1500 instructional system, scheduled for classroom use when school opens in September.

The 1500 will be used for computer-assisted instruction of first-graders in mathematics and reading. Sixteen student stations and two teacher stations will time-share the central computer, with CRT displays, slide projectors, and audio units completing the system. Instruction will be furnished at each student's rate and the computer will accumulate scores for analysis and reports of student progress.

PDP-8/S, the $10,000 computer...

PDP-8/S: A full, general purpose, digital computer for real time analysis. 4K core memory (expandable). usec speeds. 66 plus instructions. Complete, proven software, including FORTRAN. Flexible input/output bus. Teletype included.

Available now, the new PDP-8/S — a direct lineal descendant of the PDP-8, the most flexible, versatile, approachable, on-line, real time, high speed digital computer ever made.

Both use the same basic design concept. Both have the same size memories. Both are expandable. Both use the same instructions, use the same software libraries.

They do not, however, work at the same speeds. And they do not cost the same amount of money.

The PDP-8/S adds in 32 microseconds (compared with 3.0 microseconds for its parent). If you need the speed, the PDP-8 is for you.

The PDP-8/S costs $10,000. Think of it. Full computer. Proven hardware. Proven software.

Any DIGITAL field office can give you the details. Or write.

...and the new, big, PDP-9

PDP-9, compact, powerful data processor for on-line, real time applications. 18 bit word. 2 usec add time. 18,000,000 bits/sec I/O transfer rate. One word direct addressing of full 8K memory. Hardware ready now. Software ready now.

The PDP-9 is a complete, ready-to-use data processor. Basic hardware includes the 8K core memory (expandable to 32K), a 300 cps paper tape reader, a 50 cps paper tape punch, a teletype keyboard, Direct Memory Access channel plus 4 built-in data channels, and a real-time clock. It is constructed with — and interfaces with — standard FLIP CHIP™ modules.

Software includes real-time FORTRAN IV, a versatile macro assembler, a 6 and 9 digit floating point arithmetic package, an on-line editor, an on-line debugging system, a control monitor, and a modular I/O programming system. Basic software is fully compatible with the PDP-7. Extended software package expands to fully utilize all configurations.

What the PDP-9 gives you is simply this: more inputs and more outputs — faster, more simply, more effectively — than any other machine in its class.

$35,000. First deliveries in time for Christmas, 1966.

digital
COMPUTERS · MODULES

DIGITAL EQUIPMENT CORPORATION, Maynard, Massachusetts 01754. Telephone: (617) 897-8821 • Cambridge, Mass. • Washington, D. C. • Parsippany, N. J. • Rochester, N.Y. • Philadelphia • Huntsville • Orlando • Pittsburgh • Chicago • Denver • Ann Arbor • Los Angeles • Palo Alto • Seattle • Carleton Place and Toronto, Ont. • Reading, England • Paris, France • Munich and Cologne, Germany • Sydney and West Perth, Australia • Modules distributed also through Allied Radio

CIRCLE 47 ON READER CARD

the computer leasing division, to membership on the management committee. Louis G. Rothschild, a former undersecretary of Commerce, has joined the company as executive director of the management committee.

AMA SPONSORS EDUCATION SEMINAR

Education must deal differently with information if it is to take full advantage of today's technology, Emmanuel G. Mesthene, director of Harvard's Program on Techonology and Society, told an AMA-sponsored seminar on "Educational Realities" in New York last month.

"In the past," said Mesthene, "about 80% of a teacher's time was spent dumping information into a student's head." The rest was devoted to teaching the student how to use that information. But computers, with their capacity, speed, and patience, indicate that they can take over the mechanical part of teaching.

This fact, he said, suggests that only mechanical teachers should fear the computer, which offers to liberate educators for the human task of guiding students, helping them to use information to achieve wisdom. There is a tendency to view the tools of technology as merely a better way of doing a known job. But such tools actually change the job by making new things possible, said Mesthene. "If we see the future in terms of old values, we sell the future short."

A more or less realistic look at Computer Assisted Instruction was taken by a two-man panel.

Ralph E. Grubb, IBM, showed some slides of results of comparative evaluations of different types of instruction which indicated that the mode of instruction has more effect on student performance as one moves down the intelligence scale. The implication: CAI is of relatively greater benefit to slower students. In passing, Grubb indicated that it was possible to learn Coursewriter—an IBM programming language designed to help educators write CAI programs—in about a half an hour. And he summarized the results of an IBM experimental CAI program for maintenance engineers, which reveals, among other things, that CAI is a "significantly more rapid means of instructing."

Harold E. Mitzel, Pennsylvania State University educational researcher, disagreed with Grubb's evaluation of Coursewriter: he does not find it simple. He also indicated that his studies show CAI is a slower means of instructing than classroom/textbook and programmed text methods.

Mitzel described some of the work going on at Penn State, and spelled out five barriers to CAI: the hardware/software gap; lack of criteria for evaluation; too much author time required to prepare CAI programs; lack of research on the appropriate mix of CAI and other instructional methods; the lack of program compatibility on different systems.

The most realistic remarks were delivered for R. Louis Bright, of the Bureau of Research, Office of Education, Dept. of Health, Education & Welfare by Grubb: although Bright's office has 50 proposals from educational institutions seeking CAI research funds, money is available to fund only three or four. And two—Stanford and Pittsburgh Universities—have already been selected.

CELESTRON OFFERS EDP EVALUATION SERVICE

Celestron Assoc., Inc., Valhalla, N. Y., says it will offer computer-simulated evaluation services for computer selection, particularly time-sharing systems; industrial and economic/business processes; and development of software systems. The services, called Autosim, will have as basic tools GPSS (general purpose simulation system) used in real-time, and SIMSCRIPT. The firm is also developing modular simulation techniques for design of program translators. (Celestron has developed X-ACT, a translator for converting IBM 7090 programs to CDC 1604 programs.)

The firm, in preparing models for computer selection, will rely in each contract on data given the user by the manufacturer. Autosim-software techniques will be particularly applied to on-line design of operating systems and compilers; routines peculiar to simulation will be replaced by functional techniques, such as I/O control programs, to obtain the final product. The two-man staff of the Autosim projects is expected to increase to 12 within six months.

PAPERS INVITED AS SJCC PLANNING GETS UNDER WAY

The call is out for technical papers to be presented at the 1967 Spring Joint Computer Conference, April 18-20 at Atlantic City, N. J. Conference chairman is Brian W. Pollard of RCA.

Deadline for papers is Nov. 1 and the list of suggested subjects covers just about all conceivable aspects of information processing. Papers selected will appear in the Proceedings, to be published by the Thompson Book Co. Maximum length is 2500 words. Five copies should be mailed to the technical program chairman, M. P. Chinitz, at 326 Township Line Road, Norristown, Pa. 19403.

SALARY SURVEY RESULTS WILL BE AVAILABLE

A survey of programmer salaries formerly conducted by System Development Corp. has been taken over by Industrial Relations Counselors Service of New York City.

The current survey, in which 150 organizations participated, includes four levels of programmers and two levels of supervisory personnel, broken down by industry, general application (scientific, business, software), degree level, and geographical location. Results of the study—to be made available to the participating organizations soon—include 14,500 cases.

Deadline for organizations wishing to participate in next year's survey is Jan. 1. The fee for the survey results is $250. For an additional $100, participating firms can take part in an exchange with other companies of salary information.

GUIDELINES PROPOSED FOR PROGRAM PATENTS

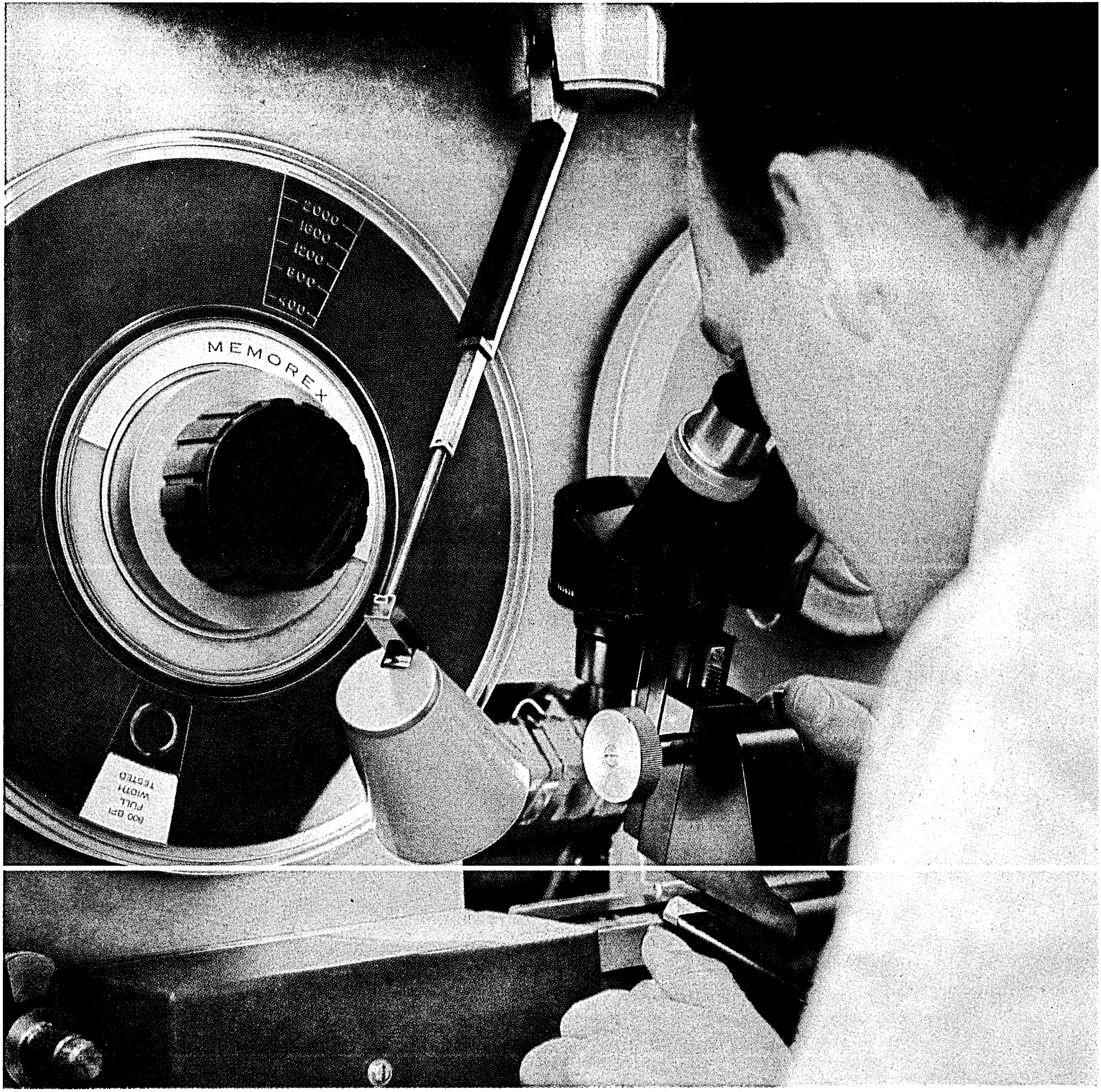
The Patent Office intends to adopt a set of guidelines for "the examination of applications for patents on programming methods and apparatus" and will consider comments on them if received by Oct. 4, when a public hearing will be held at the Department of Commerce Building.

The guidelines, submitted by the Patent, Trade-Mark, and Copyright Division of the American Bar Association were described by one lawyer active in the field as "garbled, but a step in the right direction." They can be found in the Aug. 2 issue of the Official Gazette of the U. S. Patent Office.

Much of the discussion represented by the guidelines centers on the fact that historically "mathematical process discoveries and mathematical formulas used therein may not be patented." Thus an attempt is being made to distinguish between an "algorithmic process," which may not be eligible for a patent, and a "utility process," which deals with "tangible things and substances" and therefore would be patentable.

BURROUGHS, UNIVAC, RCA BIDDING ON ILLIAC IV

Bidders for the parallel-processor ILLIAC IV, specified by the Univ. of Illinois, are Burroughs, RCA, and Uni-



Take it from Jack Hatcher:

"MRX-III computer tape is 3 to 5 times more durable than the leading competitor's"

And Jack wasn't easy to convince. He didn't care how well the tape performed in its more than six months of pilot line and laboratory testing. He wasn't impressed by the fact that 718,000 shuttle passes didn't even begin to scratch the tape. He merely shrugged at the favorable reports from the more than fifty installations where exhaustive field-testing was performed. He just said, "Well, we'll see."

That's Jack's job, as an engineer in our product evaluation lab, to see for himself how each new Memorex tape measures up. He's paid to break it down, if he can, subjecting it to any and every kind of use and abuse to which a computer tape could possibly be exposed. But on MRX-III, Jack broke his pick.

Now *you* test it. For a sample of MRX-III (on its new color-coded reel), write us at Santa Clara, Calif.

MEMOREX

Precision Magnetic Products

CIRCLE 49 ON READER CARD

Memorex Branch Offices in Boston, New York, Philadelphia, Washington, Atlanta, Orlando, Dayton, Chicago, Detroit, St. Louis, Dallas, Denver, Los Angeles, San Francisco, Honolulu. Offices and Affiliates in London, Cologne, and Paris. Distributors in Japan, Canada, India, Australia, and New Zealand.

vac. It is rumored, however, that IBM may put in an uninvited bid.

An 18-week design period will end Nov. 11, with the winner to get $6 million hardware order. Some $2 million worth of programming will be done by students at the university; the manufacturer will only have to supply the assembler.

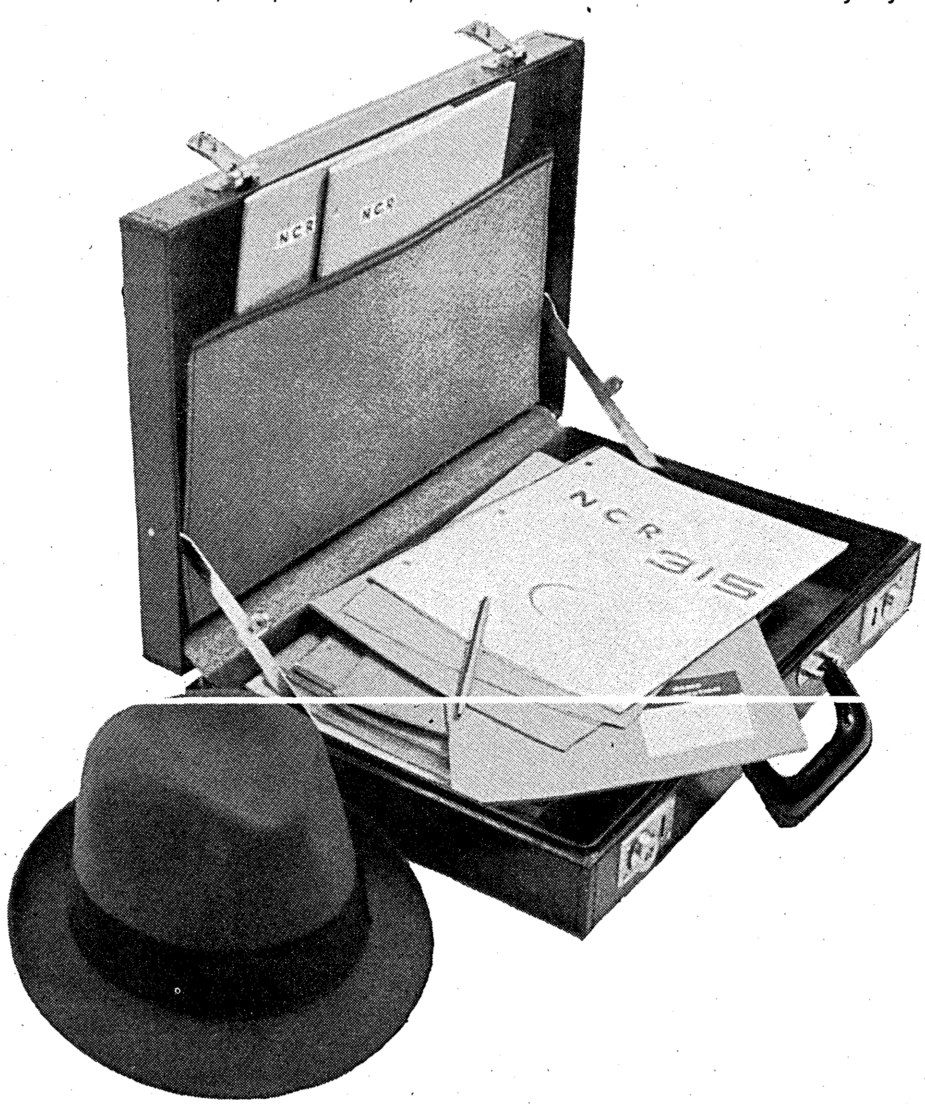
Processors for the big system will use 64-bit words, with the whole complex to include a million words of main memory. A delivery schedule of two years has been established, beginning when the decision is made on which company gets the job.

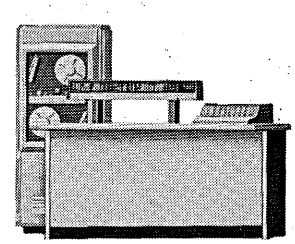
SOCIAL SECURITY OFFICE GETS CUSTOM PAGE READER

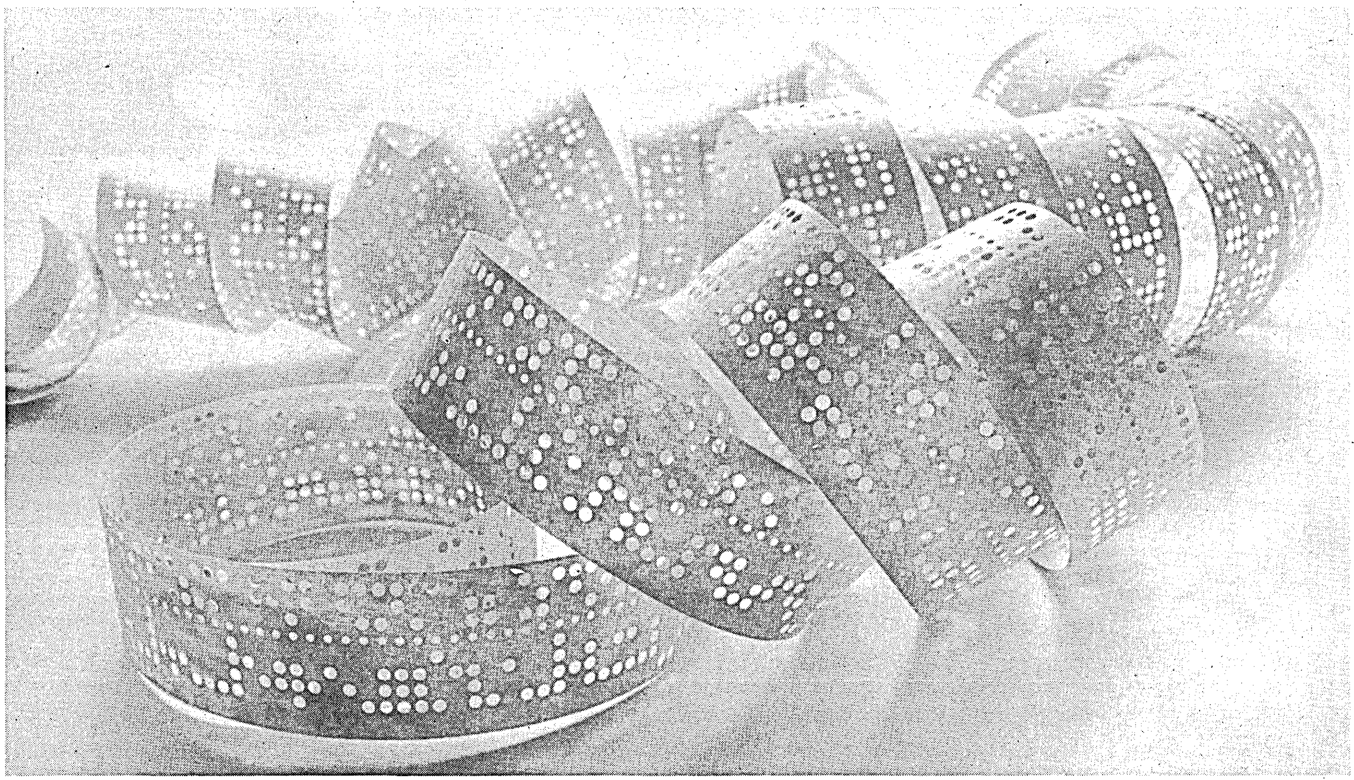
A special optical page reader that can read more than 200 different type faces at speeds up to 650 lines a minute has been developed by IBM for the Social Security Administration in Baltimore. The IBM 1975, installed and on rent for $16,500 a month, will be used to read the quarterly wage reports on 70 million workers submitted by 3.5 million employers. A 360/30 will transfer the data to mag tape. SSA has previously put the reports on punched cards, using up to 140 keypunch operators. The new system promises to save $750K the first year.

The systems scanner consists of a cathode ray tube, high resolution optical system, and photomultiplier detectors. A character recognition section contains a read-only storage of 8K words. The scanner, handling alphanumeric characters, upper and lower case, has a recognition time of 800 usec per character. New techniques in video processing, such as contrast control, dynamic video thresholding, and data reduction circuits, have been used to minimize problems in recognizing degraded forms and prints. IBM has not announced any plans to market the $775K unit.

● Following closely the Air Force release of captive System Development Corp. (see June, p. 17), another AF research house has announced an upcoming change in leadership. Effective Jan. 1, 1967, Henry S. Rowen, currently assistant director of the Bureau of the Budget, becomes president of RAND Corp., Santa Monica, Calif. He replaces Franklin R. Collbohm, head of RAND since its founding in '48, who retires and continues to serve as consultant. The announced change has stirred speculation that RAND, like the SDC it spawned, will become much less dependent on AF contracts.

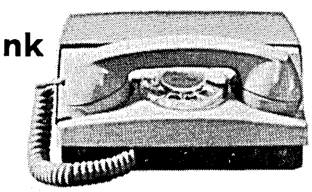

NCR
NCR
NCR 315





Clark Equipment Company gets data from 127 sales offices, 4 manufacturing plants, and a major warehouse as soon as it's recorded

Bell System communications is the vital link



Bell System data communications services link Clark's distant locations to a centralized computer center at Buchanan, Michigan. The result is better management control of all activities—sales, inventory, purchasing, production, payroll and accounting.

With current and accurate information, Clark management can quickly adjust to changing marketing conditions. Important orders get priority scheduling for production and shipment. And yet, purchasing, production and inventories stay at optimum levels.

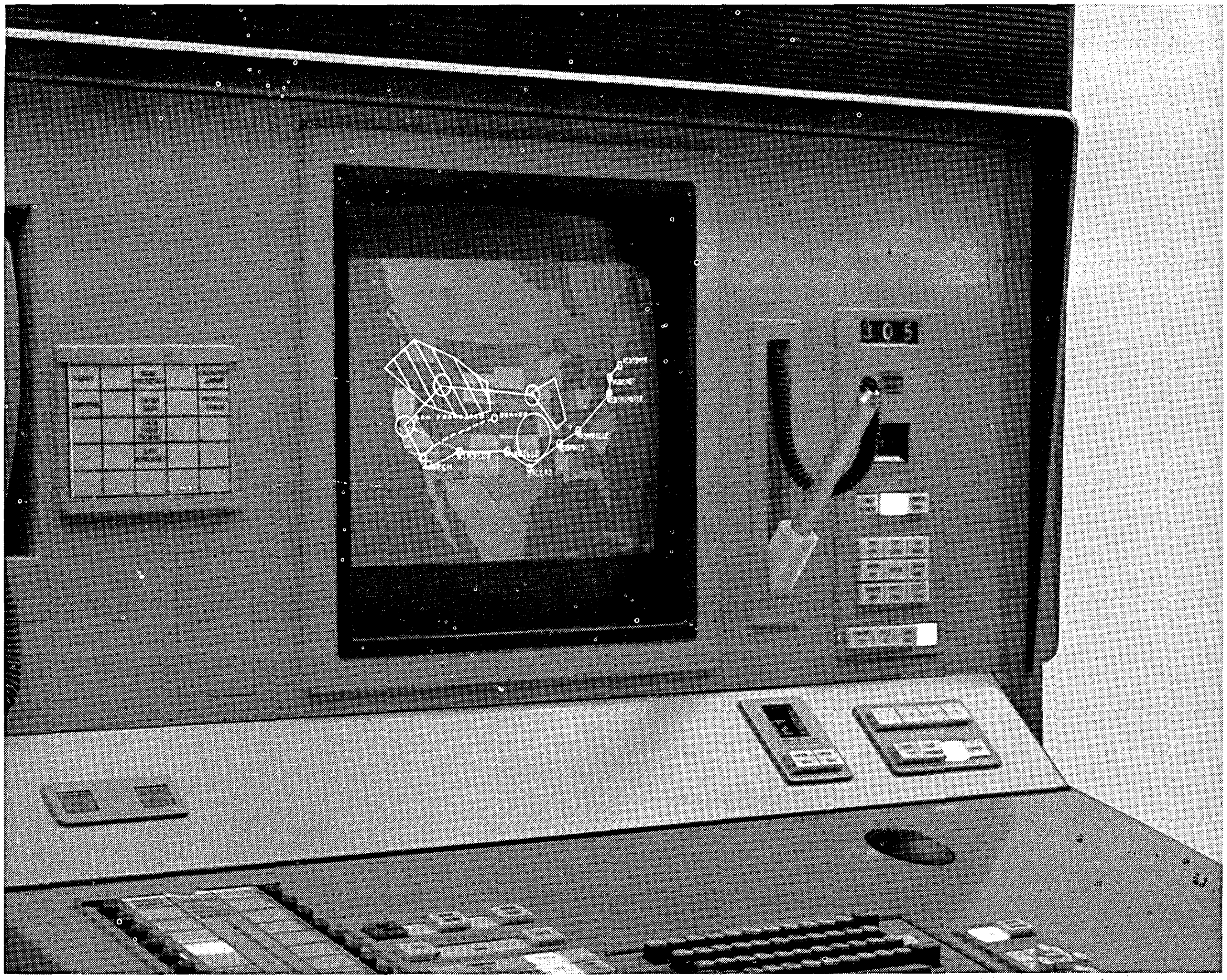
An integrated information system of this size uses computer switching with store and forward capabilities. The fully automatic Clark system polls satellite stations, receives and transmits messages, assigns priorities, and converts different speed and code formats to one standard code. Other features of the switching unit provide the necessary supervisory control of the network.

Consider the economies a real-time, integrated information system can bring to your business with automatic data processing linked with fast, reliable communications.

Today's dynamic competition requires many companies to consider organizing for data processing in some phase of their operations. It's important to start organizing communications at the same time.

So when you think of data communications, think of the Bell System. Our Communications Consultant is ready and able to help you plan an integrated information system.

AT&T Bell System
American Telephone & Telegraph
and Associated Companies



user's choice...with the BR-90

COMBINED ELECTRONIC AND PHOTOGRAPHIC DISPLAYS . . . REAR-PORTED CRT . . . STORED PROGRAM LOGIC . . . These and other exclusive features combine to make the BR-90 the most versatile and flexible console available today.

Whether your computer system is used for on-line man/machine communications, off-line data manipulation, or both—you'll find the BR-90 compatible. And it can be added to your existing system with little or no change in your input/output programs or equipment.

The BR-90 lets you express your problem in your language . . . not the computer's. Simple, changeable keyboard overlays allow you to tailor the BR-90's display functions to suit your system's requirements, growth demands, and changes.

You tell us what your requirements and applications are . . . we'll show you how the BR-90 can solve your problems.

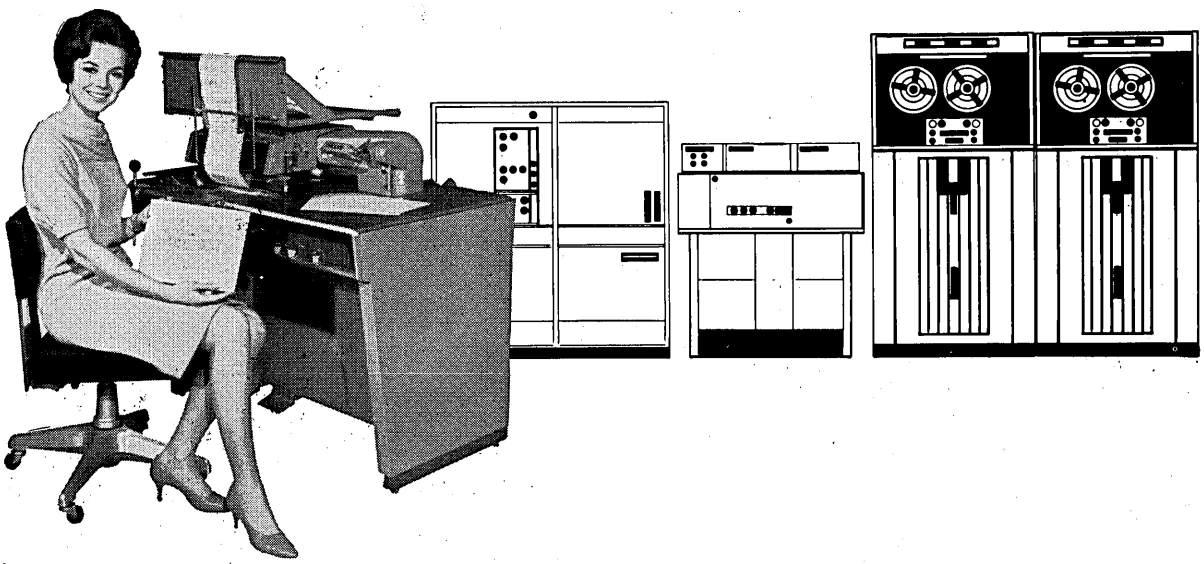
For further information contact the Bunker-Ramo sales office nearest you NOW.

Offices are located in Waltham, Mass.; Dayton, Ohio.; Rome, N.Y.; and Washington, D.C. Or write: H. A. Kirsch, The Bunker-Ramo Corporation, Defense Systems Division, 8433 Fallbrook Ave., Canoga Park, Calif. 91304. Telephone: (213) 346-6000.

THE BUNKER-RAMO CORPORATION

NOW!

an all-new method designed to expand the capabilities of a computer or punch card system



Here's a new way to get more work out of your present system without a lot of changes, revisions or costly additions to your equipment.

It's the Addressograph® 431—an all-new low-cost data printer designed to work *with* your computer or punch card system, and give you permanent black . . . non-fading facsimile copies of original data.

Flexibility of the 431 enables you to print computer output information onto conventional business forms of practically any size, shape or thickness. Multiple transfer copies can be obtained from a single data master.

Records of employee earnings and deductions, customer purchase and payment ledgers, inventory movement records and a host of other useful management control records can be automatically posted at high speed. Historical data is always instantly accessible for management analysis and use—"off-line" from the computer.

If you want a direct and economical method of printing paper documents your present system can't handle—like time and attendance records, reference cards, envelopes and promotional material—the 431 provides this too.

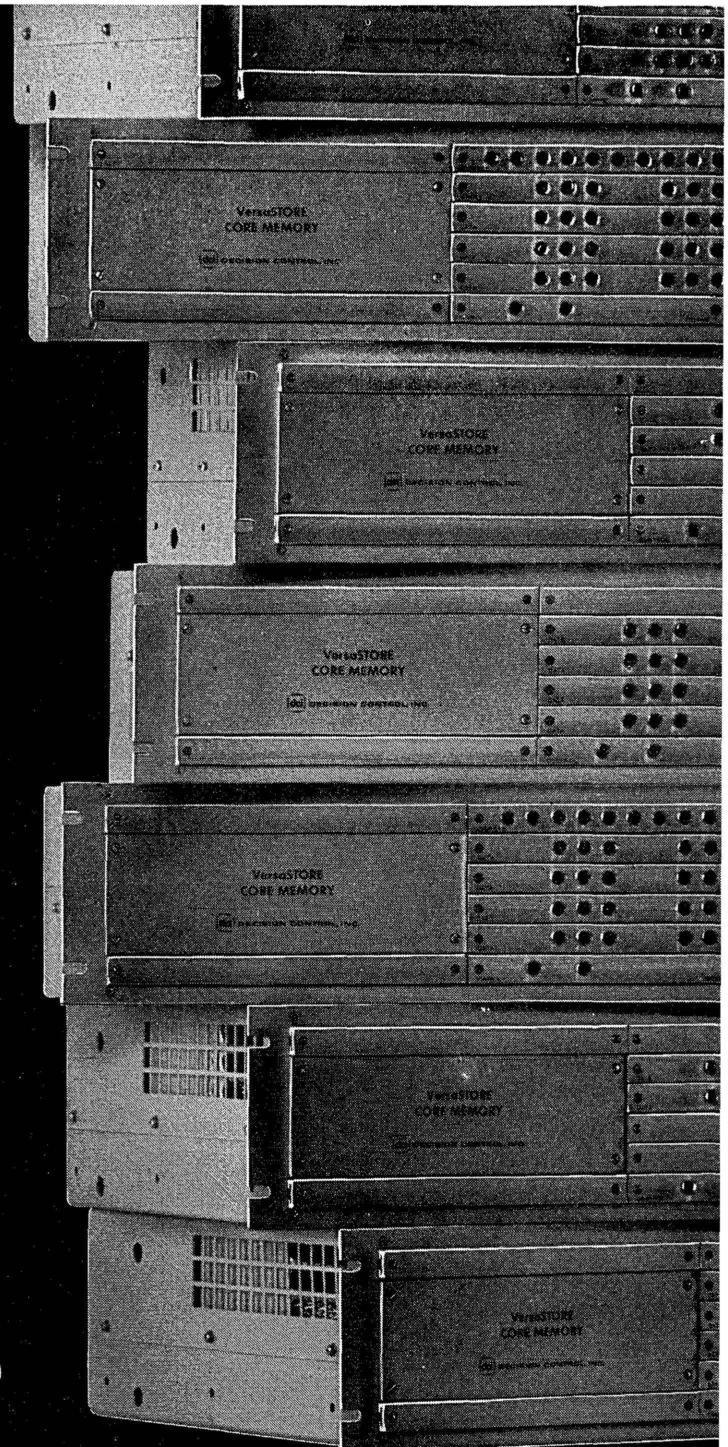
So, if you want to increase the capability of your present system, get more *useable* information and get it speedily, accurately, and at low cost, do this—call your Addressograph man (he's in the Yellow Pages). Or write Addressograph Multigraph Corporation, 1200 Babbitt Road, Cleveland, Ohio 44117.

Addressograph
DIVISION OF
ADDRESSOGRAPH MULTIGRAPH CORPORATION

CIRCLE 53 ON READER CARD

from Decision Control

BIG MEMORIES



SMALL MEMORIES

and anything in between

Decision Control makes big core memories, small core memories, and in between size core memories—to match your system requirements.

The big memories are big enough for most computer mainframe applications, and range in size from 8K to 64K with word lengths of up to 74 bits.

The small memories are small enough for buffers and other small fast access storage. They are available in increments of 256 to 4,096 words of up to 24 bits.

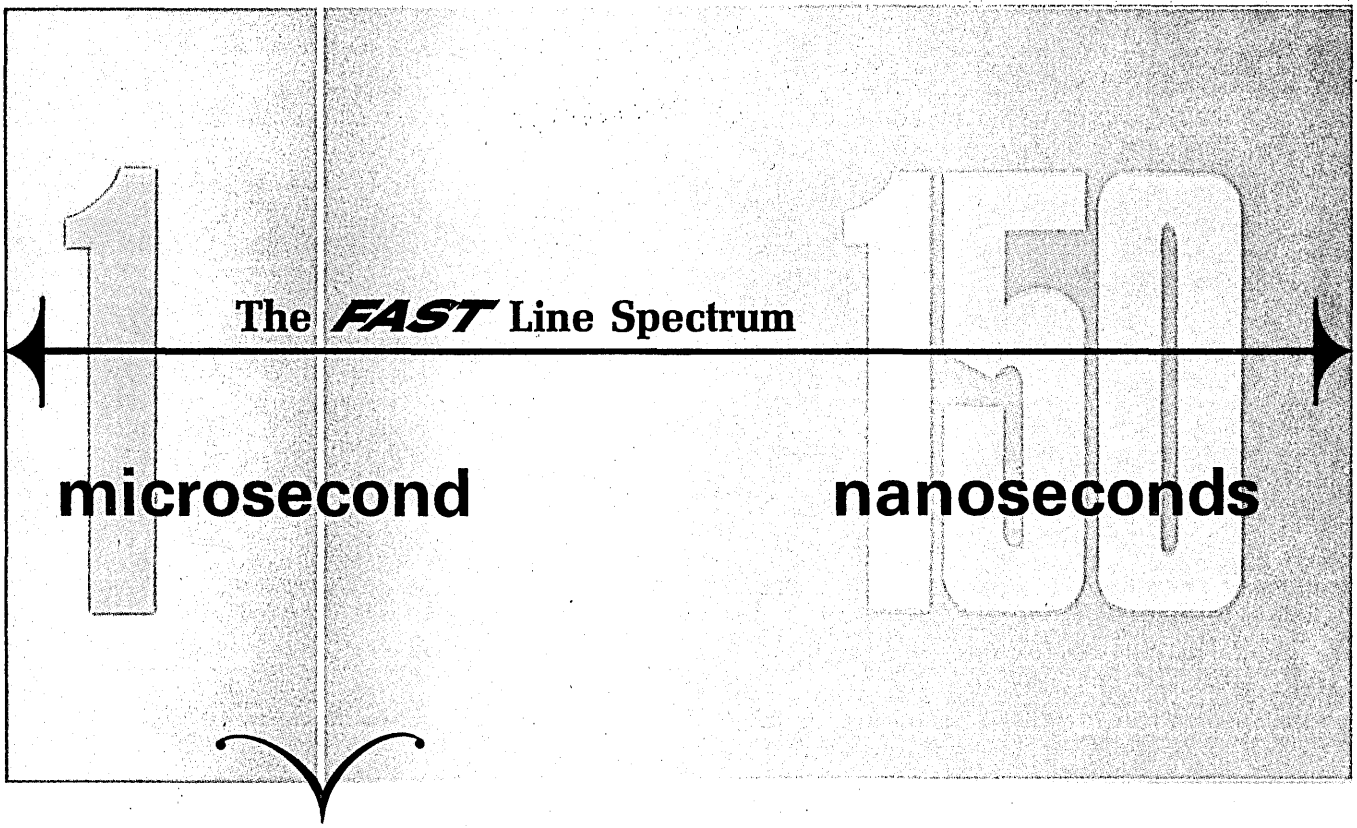
All of Decision Control's VersaSTORE memories are designed with such advanced features as PNP to NPN interface—input range of 3v. to 24v.—continuous lamp display of data registers—2 usec operation—and integrated circuits for maximum reliability.

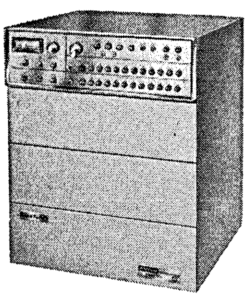
So, if you need a big memory, or a small memory, or anything in between, Decision Control has it— a VersaSTORE core memory to fit your requirements. For a fact-filled applications brochure, write to:

DECISION CONTROL, INC.
1590 Monrovia Avenue, Newport Beach, Calif. Tel. (714) 646-9371 • TWX (714) 642-1364

CIRCLE 54 ON READER CARD

Fabri-Tek's FAST Line* 500 core memories...





FAST Line 500 core memories:
Operation Summary

Cycle Time: Choose any cycle speed between 500 and 1000 nanoseconds.

Access Time: 250 to 400 nanoseconds, depending upon cycle time and control logic.

Access Modes Available: Random, Sequential or Random Sequential.

Capacity: To meet your requirements.

Optional Operating Modes: Read-restore, clear-write, read-modify-write.

Input/Output Levels: Typical —0.5 volts and +2.5 volts.

Control Panel Options: Address and data register indicators, self-testing controls, flag signals, marginal voltage check, manual load and data retention "Data Saver".

Power Required: 115 or 220 volts, ac, single phase, 50—60 CPS. Optional power supply available.

Packaging: Relay-rack modules or free-standing cabinet.

Available soon . . .
FAST Line 500 core memories bulletin
WRITE TODAY

Meeting your core memory requirements in the 500-1000 nanosecond range

If your next computer requires a core memory in the 500-1000 nanosecond speed range, the Fabri-Tek FAST Line 500 is the answer.

This family of core memories gives you a free choice of speed and capacity in this particular speed band of the spectrum.

From initial engineering to final assembly and test, the Fabri-Tek FAST Line concept means "pre-customizing" of design for the best combination of speed, capacity and price.

FAST Line 500 core memories use all-silicon integrated circuits in the logic, decoding and timing sections as well as all-silicon discrete components to insure high MTBF.

FAST Line 500 stacks take maximum advantage of 20-mil cores to provide wide operational margins at sub-microsecond speeds.

Take advantage of Fabri-Tek's technology and "pre-customized" design. Ask about FAST Line!

Call: 612-935-8811—TWX: 910-576-2913 or
Write: Fabri-Tek Incorporated, 5901 So. County Rd. 18, Minneapolis, Minnesota 55436.

*Trademark

The Leader in Memory Technology

CIRCLE 55 ON READER CARD

new products

i/o terminal

The Data Terminal consists of a mag tape recorder with a capacity of 100K alphanumeric characters and an interface for the 202 DataPhone. Optional additions to the recorder are a typewriter, Teletypewriter, 80-column card

reader and paper tape reader. The recorder enables unattended transmission, which can be from a remote terminal to a computer center or to another terminal; the transmission is bidirectional. DARTEX INC., Santa Ana, Calif. For information:

CIRCLE 150 ON READER CARD

sampling oscilloscope

A relay-operated unit featuring fully automatic programming, the model 120AR-P operates from punched card paper and mag tape. It will test all switching parameters, such as rise time, fall time, turn-on time, propagation delay, and pulse width. A remote sampling head provides a band-pass of 3.5 KMC or 100 ps rise time. Other building blocks include a programmable switching matrix, wave form reader, programmable pulse generator, normalizer and a Go-No Go comparator. GENERAL APPLIED SCIENCE LABORATORIES INC., Westbury, N.Y. For information:

CIRCLE 151 ON READER CARD

display console

The MACC (modular alter & compose console) unit can combine electronically-generated symbols and slide-projected images. Applicable both off- and on-line, it can update, edit, and erase displayed information. Dynamic data on the 12 x 12-inch CRT can be combined with up to 128 stored slide overlays in color or B&W. Slides can be changed in two seconds, and 64 characters/line and 48 lines in typewriter mode operation can be displayed. Vectors can be drawn between any combination of the 1024 x 1024 positions. ITT FEDERAL LABORATORIES, Fort Wayne, Ind. For information:

CIRCLE 152 ON READER CARD

computer clock

The Time Monitor maintains a permanent running record on mag tape, paper tape or punched cards. It records the overall time required to prepare, utilize and unload a computer, the actual running time, actual time for preparation and unloading, inactive time and reasons for stoppages. It can identify operators, program, and

PRODUCT OF THE MONTH

The PDP-9 is a parallel, binary 18-bit computer with a cycle time of 1 usec. It features a microprogrammed memory that, among other things, does away with the major state generator in the PDP-7, with which it is compatible. The NDRO memory also decodes instructions, controls register gating, etc. Aimed at the systems and control markets, the PDP-9 joins a field comprised of the CDC 1700, ASI 6130, IBM 1800, IEC 1010, and DDP-116, with which it is competitive in speed and price.

The add time is 2 usec, and I/O transfer rates are up to 18-million bps. Other features include 8-32K words of core, automatic indexing, indirect addressing, multi-level priority interrupts. A high-speed parallel adder, which serves as a path between all active registers, is said to combine two 19-bit words in 200 nsec. Software includes a real-time FORTRAN IV, macro assembler, control monitor, 6- and 9-digit floating-point arithmetic, and on-line editing and debugging.

Optional peripherals include mag tape, discs, drums and displays. A basic system, including 8K words of core, 300-cps tape reader and 50-cps punch, teleprinter console, direct memory access channel, four data channels, and a real-time clock, sells for $35K. Deliveries are scheduled to begin December '66. DIGITAL EQUIPMENT CORP., Maynard, Mass. For information:

CIRCLE 153 ON READER CARD

programmers. Model AL-4 provides a complete picture of usage, and AL-3 records the overall time the computer is in operation for each project. APLIED LOGIC CORP., Princeton, N.J. For information:

CIRCLE 154 ON READER CARD

mag tape recorder

FT-151 digital mag tape system is designed for geophysical, mobile, and shipboard field recording applications. The 95-pound unit operates on a 12-volt battery and will function within 32 to 125-degree temperatures and with up to 5 g's shock. It uses 8½-inch reels and provides three tape speeds in any combination within 15 to 150 ips. Tape formats are 7- and 9-channel, although 21-channel units are also available. Maximum start and stop times are 50 msec and 100 msec, respectively. POTTER INSTRUMENT CO., INC., Plainview, N.Y. For information:

CIRCLE 155 ON READER CARD

tape spooler

The RS-1000 operates at 1,000 cps, and accommodates reels with up to 10½-inch diameter. It also has a bi-directional rewind capability. REMEX ELECTRONICS, Hawthorne, Calif. For information:

CIRCLE 156 ON READER CARD

keypunch trainer

The Challenger I plugs into an IBM verifier, displays practice material (on printed tapes), which the student verifies into pre-punched cards. If an error is made, the unit stops and will not proceed until the error is corrected. The speed of presentation of practice material is alterable. Materials on the practice tapes and in the accompanying instruction manual are reportedly compatible with current keypunch training methods. UNITED DATA PROCESSING INC., Portland, Ore. For information:

CIRCLE 157 ON READER CARD

300 lpm printer

Series 4000 printers have a 300 lpm speed, a capacity of up to 160 columns, a set of 64 alpha-numeric characters, and interfaces for any computer. The units accept standard fanfold paper forms in widths up to 20 inches. A 128-character font is available. ANELEX CORP., Boston, Mass. For information:

CIRCLE 158 ON READER CARD

incremental plotters

Models 665 and 663 are drum-type digital units with switch selection of either 0.01 or 0.005-inch step size. The 665, with a 12-inch drum, operates at 900 steps/second in .005-inch step size and 450 steps/second in the .01-inch step size. The 663 has a 30-inch drum and operates at corresponding speeds of 700 and 350 steps/second. Software for the 500 series plotters can be used without modification. CALIFORNIA COMPUTER PRODUCTS INC., Anaheim, Calif. For information:

CIRCLE 159 ON READER CARD

tape buffers

The series 5030 tape recorder buffers are instruments for translating, formatting, and transferring digital data from a-d converters, counters, registers to standard incremental mag tape recorders. The basic buffers accept six 4-bit or four 6-bit characters, but additional inputs can be provided. The buffers generate IR gap commands and other housekeeping functions. CONTROL EQUIPMENT CORP., Needham Heights, Mass. For information:

CIRCLE 160 ON READER CARD

tape drive

Developed for aerospace, medical and industrial applications, the model 7600 tape system records digitally at 1,000 bps and direct recording to 1.6 mHz. Tape speeds range from 1⅞ to 120 ips, and tapes of ¼, ½ or 1-inch are accommodated on reels up to 15 inches. Features include a dual vacuum column and a capstan that is integral with a printed-circuit motor assembly. HONEYWELL TEST INSTRUMENTS DIV., Denver, Colo. For information:

CIRCLE 161 ON READER CARD

delay line memory

Delcom modules contain a magnetostrictive delay line and microelectric read, write, and logic circuitry. They are supplied with delays of up to 1,000 usec and for use at bit rates

of up to 1 MC in the return-to-zero, and 2 MC in non-return-to-zero modes. COMPUTER DEVICES CORP., Huntington Station, N.Y. For information:

CIRCLE 162 ON READER CARD

edge-punch perforator

The LP-2 is now available as a combination card edge/tape punch. Edge punch cards are handled in fanfold stacks, and 8-level data is punched asynchronously at 150 cps. SOROBAN ENGINEERING INC., Melbourne, Fla. For information:

CIRCLE 163 ON READER CARD

electronic calculator

The model 161 is a 16-digit calculator that can add, subtract, multiply and divide, as well as combining any or all of these operations in a single problem. The keyboard has a standard 10-key arrangement. With a decimal key integrated into the nu-

meric keyboard, decimals are automatically placed. Other features include a memory for storing constants, overflow signals, and add/subtract time of 1/100th second. CANON U.S.A. INC., New York, N.Y. For information:

CIRCLE 164 ON READER CARD

process control computer

The PRODAC 250 process control system fills the gap between the small Prodac 50 and large 500. The integrated-circuit system, which uses an SDS Sigma 2 processor, offers a 1-usec cycle time, 256 I/O channels, 4-64K words of core, and no "practical limit" to the mass memory that can be added (up to 48 3-megacharacter drums, for example). SDS fixed-head discs and drums will be used. The PROGEN software system contains an executive system of program management routines, a FORTRAN IV compiler which can write programs for real-time process control, and standard subroutines. A wide range of process communications peripherals is offered. Delivery begins in fall 1967. WESTINGHOUSE ELECTRIC CORP., Pittsburgh, Pa. For information:

CIRCLE 165 ON READER CARD

plotter control

The model 234 permits the Series 200 computer systems to use four different Calcomp 500 plotters on-line. With an interrupt capacity, the termination or interruption of plotting is facilitated. Maximum speeds are 300 increments per second. HONEYWELL EDP., Wellesley Hills, Mass. For information:

CIRCLE 166 ON READER CARD

process computer

The GE/PAC 4050-II for industrial control has a core cycle time of 3.4 usec, compared with the 5.1 usec of the 4050 and 1.7 usec for the 4060. The processor includes arithmetic and control unit, up to 64K (24-bit) words of core, peripheral buffer, eight levels of automatic priority interrupt, and a stall alarm. Compatible with the GE/PAC family, it has circular list processing, floating-point subroutines, and masked-memory search capability. GENERAL ELECTRIC PROCESS COMPUTER BUSINESS SECTION, Phoenix, Ariz. For information:

CIRCLE 167 ON READER CARD

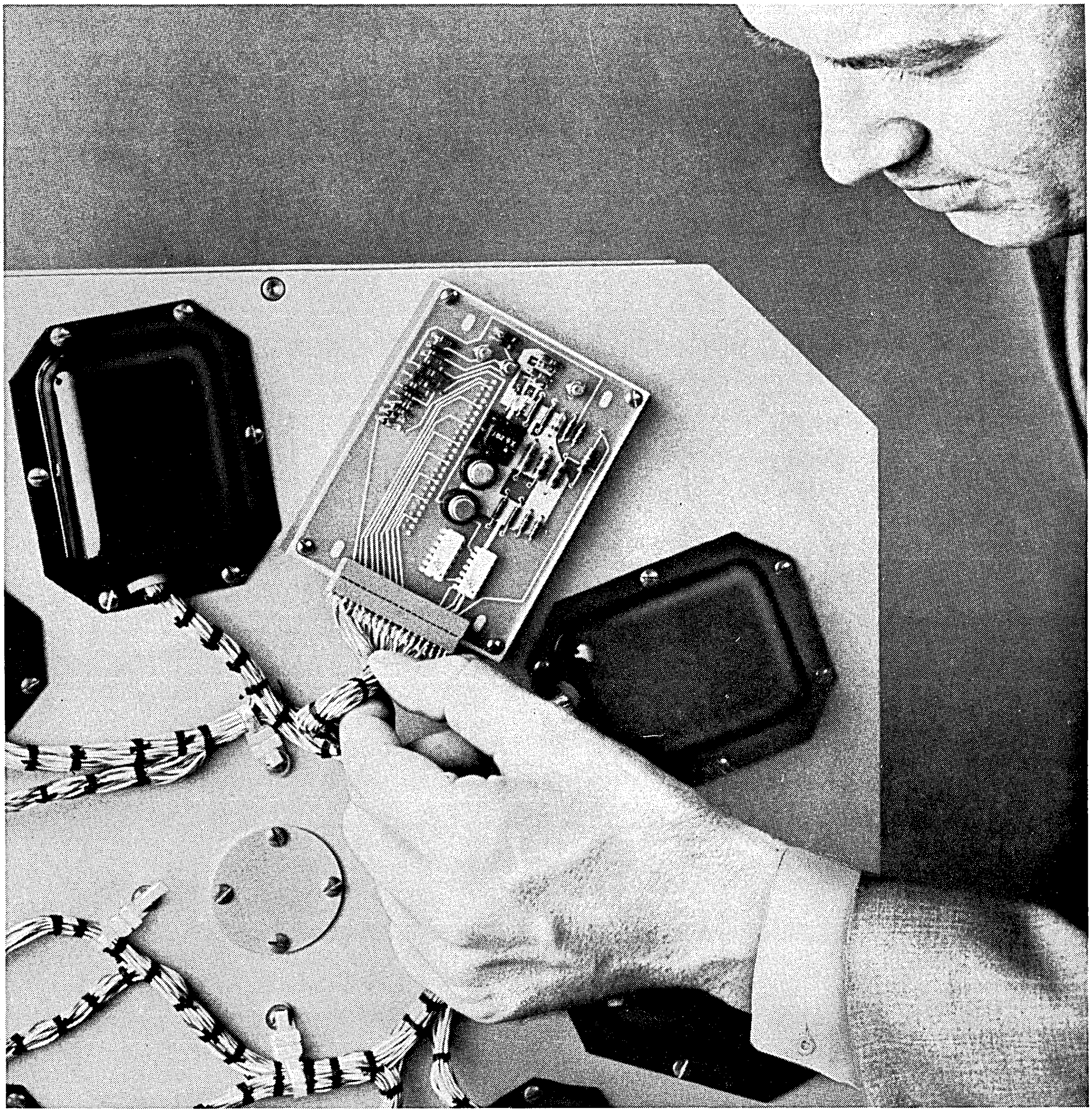
mag tape

A higher-priced tape for the discriminating buyer, the MRX-III is for use at up to 1600 bpi, and is available for both computer and instrumentation applications. It is said to excel in both durability and dropout activity. MEMOREX CORP., Santa Clara, Calif. For information:

CIRCLE 168 ON READER CARD

control instrumentation

The H series of electronic instrumentation includes new controllers, recorders, indicators, and auxiliary stations. Shelf-mounting method permits instruments to be arranged in control room panels by logical process steps. Shelving is available in 1- to 10-unit capacities. The model 62H electronic indicating controller features a remote-local switch for electrically setting the control point from an external source, and adjustable high and low limiting on the controller's output. It is available as a single-station ratio controller, and with a motor-driven set point mechanism that allows its set point



Now Large Memory [at a new low cost per bit]

Now, Librascope's new L416 Integrated-Circuit Disc Memory, specifically designed as a high-capacity, fast-access, low-cost disc file for use with low-cost digital computers in commercial applications.

- 6-and 12-million-bit capacity (expandable to 18 or 24), with cost per bit materially reduced by high packing density.
- fast access, averaging 8 milliseconds.
- outstanding magnetic performance, assured by hard-cobalt disc coating and an automatic-retracting mechanism that prevents flying heads from touching discs even during starting and stopping.
- fully integrated circuitry mounted directly on the head-mounting plates to conserve space, increase operating reliability.
- lightweight: less than 100 pounds for a 12-million-bit memory.
- consider the L416, newest member of the computer industry's most complete line of disc memories.
- for complete details, write for our technical bulletin.

CIRCLE 60 ON READER CARD

COMPONENTS DIVISION

GP GENERAL PRECISION INC.

LIBRASCOPE GROUP

808 Western Avenue

Glendale, California 91201

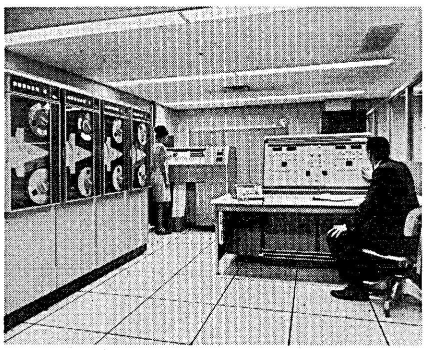
to be adjusted by a pulse source. The model 61H flow controller features a built-in transmitter power supply, local-remote switch, and a 4-position transfer switch. FOXBORO CO., Foxboro, Mass. For information:

CIRCLE 169 ON READER CARD

communications system

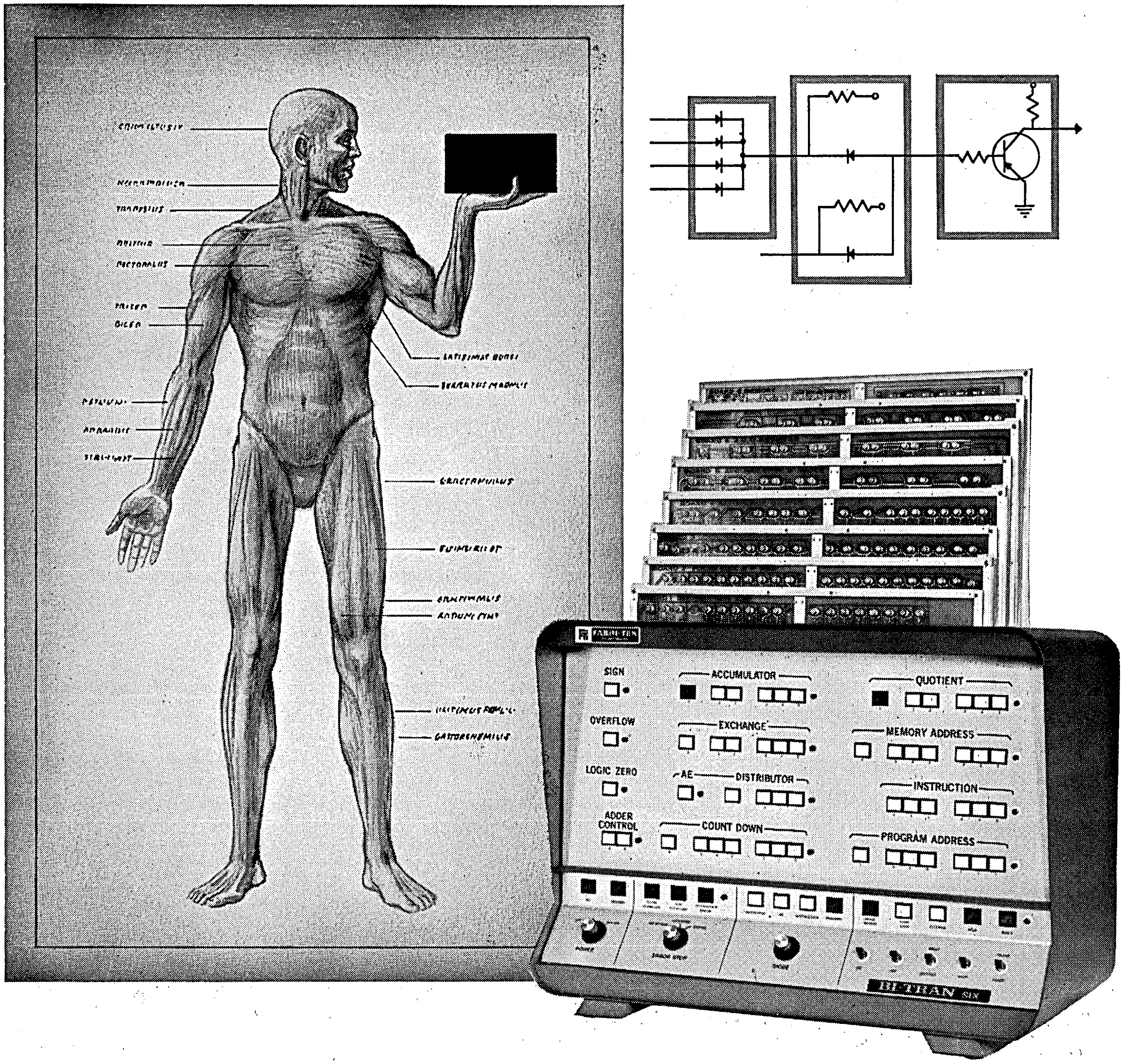
The System 600 Dial-o-verter collects information from remote sources and automatically separates computer-bound data from the message traffic. The system can transmit at 1,000 wpm and receive at 1,500 wpm. Heart of the system, the control station, handles all functions for automatically dialing phone numbers and establishing communications with remote terminals. It contains a memory for storing system logic and remote call sequences, and interfaces with a printer terminal for an on-line record of all communications. Built in are automatic error detection and correction devices. The system can exchange, interpret, format and translate data into computer-compatible language. Automatic polling feature enables a central station to call in information from remote sources on a schedule, with priority messages superseding the regular polling sequence. DIGITRONICS CORP., Albertson, N.Y. For information:

CIRCLE 170 ON READER CARD



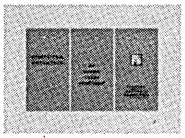
communications adapter

The IBM 1130 computer can now be a communications terminal, as a synchronous adapter has been developed to enable connection of the system to an 360 computer. Connection can be made with common carrier or private lines at 75-300 characters a second. Features of the adapter include: automatic answer of incoming calls on a switched network; program-controlled audible signal for operator attention; line control, data transfer, and code



Computer anatomy can't be taught with black boxes!

You have to expose the "insides" of a computer to teach fundamentals. The BI-TRAN SIX computer education system has no mysterious black boxes which have to be explained on the blackboard. The BI-TRAN SIX brings computer anatomy out into the open for students to see, touch, and feel. The "hands on" BI-TRAN SIX method teaches concepts in minutes . . . black boxes can waste valuable classroom hours.



FREE:

"COMPUTER EDUCATION: An answer to the challenge!"

Educational Products

Fabri-Tek Inc. 5901 County Road Eighteen Minneapolis, Minn. 55436

NAME__________ TITLE__________

ORGANIZATION__________

STREET__________

CITY__________ STATE__________ ZIP__________

TEACH:

- Computer organization • Programming
- Computer maintenance techniques
- Computer circuit analysis and design
- Use of Boolean algebra in computer systems

Teach at any grade level through college and in vocational or military training programs with the BI-TRAN SIX Computer Education System. Write today for details and price information, or call 612/935-8811.

FABRI-TEK INCORPORATED

Leading in memory technology and computer education systems.

CIRCLE 63 ON READER CARD

conversion under program control; communication with a number of core-to-core, punched card, and mag tape transmission units. IBM, White Plains, N.Y. For information:

CIRCLE 171 ON READER CARD

radar computer

The Radar Intercept Calculator is a semi-automatic target intercept system performing target position and course determination automatically. Performing from radar video data and antenna azimuth information, it digitally computes speed and projected target position, and directs interceptors against as many as five targets simultaneously. A microelectronic unit for mobile, tactical environments, it weighs 38 pounds, occupies 0.75 cubic feet. MILITARY ELECTRONICS DIV., MOTOROLA INC., Chicago, Ill. For information:

CIRCLE 172 ON READER CARD

audio response unit

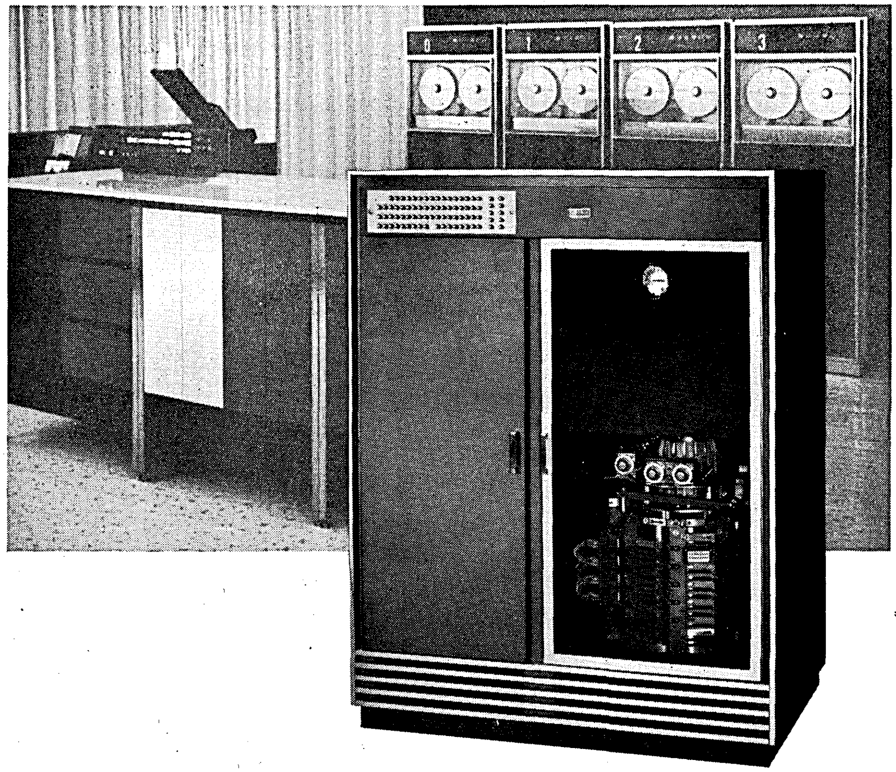
The audio response unit, designed to work with the Spectra 70/35, 45, and 55 computers, can handle inquiries from up to 100 remote points. (The basic unit handles 10 lines, and nine expansion units can be added to serve 100.) Recordings of 31, 63, or 128 words, each repeated three times, can be stored on the magnetic drum. In operation, the caller dials the system's number and, upon signal, transmits the inquiry by pressing key buttons on a telephone device. The computer locates the answer in storage, prepares the message, and converts it to voice response from the drum. RCA EDP, Cherry Hill, N.J. For information:

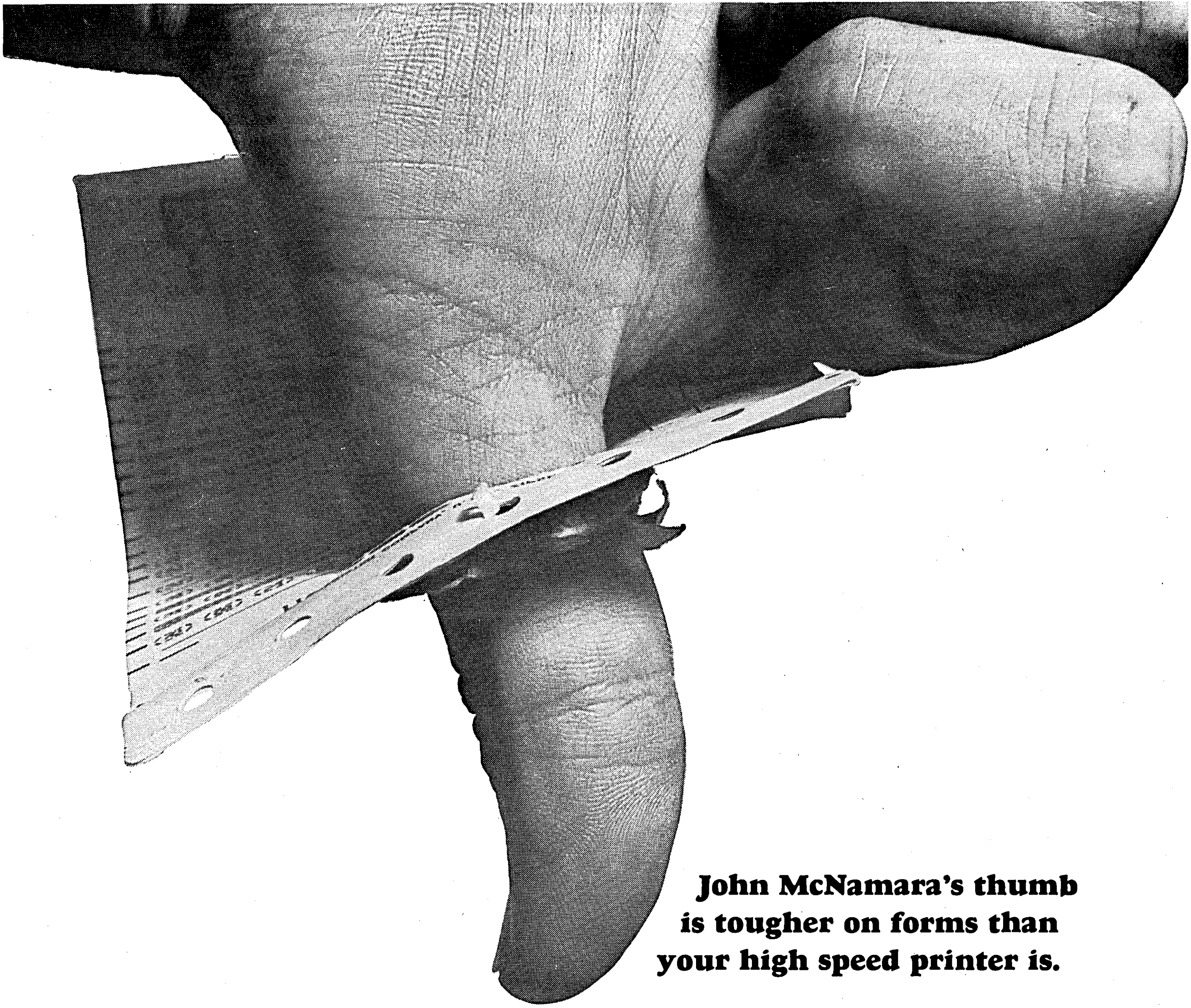
CIRCLE 173 ON READER CARD

accounting configuration

The E4000 consists of an internally-programmed processor with 200 (12-digit) words of core, a control console similar in layout to an alphanumeric accounting machine, card reader/punch, tape reader/punch, and magnetic ledger card reader. The latter reads 48 cpm in the flow mode or on demand from the processor. The processor, whose memory words are individually addressable from the keyboard, has a 5-digit add time of 1.5 msec. Divide time is 52 msec. Applications of the system include payroll, accounts payable and receivable, insurance billing, and inventory control. BURROUGHS CORP., Detroit, Mich. For information:

CIRCLE 174 ON READER CARD





new literature

STANDARDS FOR COMPUTERS & DP: Document produced by the X3.5 Subcommittee (vocabulary) of ASA Sectional Committee X3 (computers and information processing) is available. Sponsor is Business Equipment Manufacturers Assn. BEMA, New York, N.Y. For copy:
CIRCLE 130 ON READER CARD

TAPE READER: Technical bulletin describes the model PTR-90, and specifications and features of the 1,000-cps photoelectric unit. OMNI-DATA DIV., BORG-WARNER CORP., Philadelphia, Pa. For copy:
CIRCLE 131 ON READER CARD

DISC FILE: Eight-page systems brochure describes features of model 2A series 4000 disc, which includes a self-contained environmental control, electronics, and a direct addressable digital head positioning system. Characteristics described include I/O interface lines, address fields, and organization of data. BRYANT COMPUTER PRODUCTS, Walled Lake, Mich. For copy:
CIRCLE 132 ON READER CARD

MAG TAPE SYSTEM: Four-page brochure explains use of BI/SCAN I for transient analysis and display, random data analysis, SNR enhancement of non-periodic signals, real-time enhancement of periodic signals and signal delays for correlation studies. Covered also are headwheel subsystem characteristics, standard system parameters, loop transport characteristics and FM record/reproduce electronics. S. HIMMELSTEIN & CO., Elk Grove Village, Ill. For copy:
CIRCLE 133 ON READER CARD

PRINTER: Four-page pamphlet describes model QF digital printer, a 12-column parallel input printer capable of printing 0 to 9. Code BCD 1-2-4-8 has a maximum rate of 2 lines/second. An available multiplexer option permits two 12-digit instruments to be printed out. ELECTRONIC PRODUCTS DIV., BAIRD-ATOMIC, INC., Cambridge, Mass. For copy:
CIRCLE 134 ON READER CARD

SYSTEMS ANALYSIS IN EDUCATION: 25-page handbook defines terms, considers concepts of systems, synthesis and anasynthesis, simulation and modeling in reference to training and education. Graphs and figures illustrate the processes of systems and supra-systems. EDUCATION AND TRAINING CONSULTANTS, Los Angeles, Calif. For copy:
CIRCLE 135 ON READER CARD

PRINTED CIRCUIT BOARDS: Eight-page brochure describes boards which meet standards set by NASA, DOD and leading manufacturers of avionics, industrial and computer products. Photos show details of multilayer boards, metal substrate boards, plated through holes, integrated circuit mounting and high-density mounting. SCIENTIFIC DATA SYSTEMS, Santa Monica, Calif. For copy:
CIRCLE 136 ON READER CARD

PROGRAM CONTROL: 4-page booklet describes system for control of paperwork production and distribution of copies on a programmed basis; includes three methods for controlling copy quantities. ADDRESSOGRAPH MULTIGRAPH CORP., Cleveland Ohio. For copy:
CIRCLE 137 ON READER CARD

SIMULATION LANGUAGE: Seven-page booklet gives technical summary of the MILITRAN simulation-oriented, general-purpose language. Topics include list processing, general features and MILITRAN statements. SYSTEMS RESEARCH GROUP INC., Mineola, N.Y. For copy:
CIRCLE 138 ON READER CARD

DATA TRANSMISSION: Brochure discusses systems that can be arranged for multi-channel telegraph and intermediate-speed data applications. Includes performance information, equipment description, block diagrams and photographs. LENKURT ELECTRIC CO., San Carlos, Calif. For copy:
CIRCLE 139 ON READER CARD

ELECTRONIC BUYER'S GUIDE: 400 pages list Japanese electronic companies, include address, products, key per-

Pictures spoken here

If ancient man had invented the digital computer, we'd have no problem. His language was pictures.

But the alphabet came along, and we're saddled with computers whose native tongue is one of letters, numbers, and punched cards. Some modern men speak this language, too. But not all.

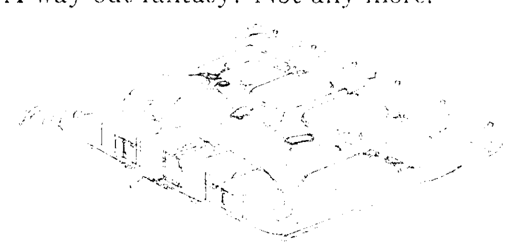
The designer, for instance, represents ideas in drawings. With proper schooling, so can the computer. Two years ago we announced the first such development: the DAC system, design augmented by computers. It used an educated computer speaking some of the designer's language. That was the first big step . . . a move to free the man from routine tasks, to let him spend more time creatively.

What's followed has been step-by-step improvement, bringing us closer to man-machine communication directly in graphics.

Without writing program statements, the designer now can use the computer to generate, manipulate, and evaluate free-form lines and surfaces. Every item in a designer's picture is a variable under his control. As he reviews and selects items on our laboratory console screen he can, for example, gradually develop a complex three-dimensional surface for an automobile.

The goal: Let the designer put a rough sketch on the computer console and make instantaneous changes as he develops his idea into a final exact design—all without translating into computer language.

A way-out fantasy? Not any more.



Typical automobile surface.

General Motors
Research Laboratories
Warren, Michigan 48090

sonnel. Alphabetical listing of all products and individual manufacturers. Cost: $10 f.o.b. STARNETICS, 10639 Riverside Dr., North Hollywood, California 91602.

CYBERNETICS IN EUROPE AND THE U.S.S.R.: 51-page paper opens with short review of the scope and meaning of cybernetics as conceived by Norbert Wiener of the U.S. and A. Berg of the Soviet Union. Outlined are cybernetics activities in the U.K., France, Germany, Italy and the U.S.S.R., followed by changes in various national policies as economies grow. Listed are 127 references, 15 bibliographies and 28 citations of conference proceedings. Order No. AD-632 599. Cost: $2.; microfiche, $.50. CLEARINGHOUSE, U.S. DEPT. OF COMMERCE, Springfield, Va. 22151.

INCREMENTAL OPTICAL ENCODER: Two-page data sheet describes encoder and includes photos, operational data, specifications and a block diagram. Unit described, OPTI-SCAN, features resolution of 2^{16} (65,536) in a 3½-inch diameter housing. Encoder is designed to mount on to a gimbal frame. SEQUENTIAL ELECTRONIC SYSTEMS INC., Elmsford, N.Y. For copy:

CIRCLE 140 ON READER CARD

COMPUTATIONAL LINGUISTICS: Bibliography for 1962 cites 700 domestic and foreign articles, reports, and books relevant to fields of computational linguistics and documentation, with selective coverage in classification theory, computation and programming, non-numerical applications of computers, and psycholinguistics. Order no. AD-630 985. Cost: $3; microfiche, $.75. CLEARINGHOUSE, U.S. DEPT. OF COMERCE, Springfield, Va. 22151.

RATIO DETECTOR: Data sheet describes model 2283 which is an instrumentation module for a core memory test system that employs transient signal analysis to determine whether the peak amplitudes of two serially generated voltage pulses are within a given percentage separation of one another. Included are photographic views of the detector, electrical specifications and descriptions of operation for the various front panel controls. COMPUTER TEST CORP., Cherry Hill, N.J. For copy:

CIRCLE 141 ON READER CARD

Lost at sea: $2,129.68 and 14 days!

TWA compares surface vs. air on a 6,200 lb. computer shipment from Pittsburgh to Rome.

Via Surface:	Time: 15 days	Via Air:	Time: 15 hours
Packaging	$ 700.00	Packaging	$ 20.00
Insurance	1,980.00	Insurance	900.00
Inland freight (Pitt-N.Y.)	428.40	Inland freight	—
Documentation	22.00	Documentation	1.00
Ocean freight	476.08	Air freight	1,832.80
Customs clearance	52.00	Customs clearance	18.00
Inland freight (Naples-Rome)	281.00	Inland freight	—
Interest charges	1,036.00	Interest charges	74.00
	Cost: $4,975.48		**Cost: $2,845.80**

Savings in shipping costs are immediate...and just the beginning. There are other benefits: reduced inventory and warehousing costs, better customer service, increased sales in time-limited situations.

The speed of air also cuts your investment in goods in transit, increases capital turnover and holds obsolescence to a minimum. Get all the details. Call or write your nearest TWA Air Cargo office.

Nationwide
Worldwide
depend on

TWA AIR CARGO

CIRCLE 70 ON READER CARD

world report

PERIPHERALS FOR EUROPE: AN OPEN MARKET?

Announcement of Britain's two most ambitious commercial dp systems, for the BOAC airline and the Post Office, reveals Europe's weakness in peripheral equipment. As reported in Datamation, BOAC is preparing a near-$100 million total management system based on IBM 360's. Up until the end of August, the corporation was judging bids for the 700 CRT displays to be linked to the complex. Contenders were Elliott Automation, Ferranti, and Raytheon. Tenders from Raytheon allow for their U.K. subsidiary, Cossor, to undertake manufacturing.

Raytheon is tipped as favorite for the contract to supply 200 displays to BOAC's North American operations directly and the remainder through Cossor. Weighting the scales in Raytheon's favour is a recent order for several hundred displays to back up the twin Univac 1108's for Air France. Univac is understood to be unable to supply from in-house since its production allocation is at full capacity to meet the 2,000 needed for the United order.

New types of document sorters are now much in demand from the Post Office to add to an $8.5-million order for computers for an automated Giro bank. English Electric-Leo-Marconi has picked up the mainframe business for five Systems 4 model 70's due for delivery from 1968-'70. The GPO has taken an option on another four machines, since there are indications of more activity than originally forecast.

Initial estimates were for a total of 1,250,000 accounts to be handled at a national Giro dp centre under construction at Bootle, near Liverpool. A recent market survey has shown that three times this number are now likely. Still to be solved by the GPO's system team is the problem of data input. Plans are afoot to use a combination of magnetic ink and optical character recognition with cheques and credit transfer slips. Most experience in this type of peripheral lies with Continental machine makers, such as Telefunken which has developed special sorter-reader processors for other Giro users.

PERIPHERALS GET THE CALL

Still on the peripheral theme, Telefunken has developed a CRT display for an experimental air traffic control application. Called the SAP 2000-4, it links into Telefunken's TR4 computer via duplexed communications systems operating at 600, 1200 or 2400 bps.

The SAP 2000-4 uses a 1024-word 24-bit control buffer as a display refresher. A joystick control is used to shift an electronically projected marker around the screen and to feed co-ordinates direct to the computer.

Telefunken has also installed one of its small-town letter sorters at Bochum Post Office. Envelopes are transported past operators who key in code lines for luminous character handling. Incoming mail is coded for sorting into deliverymen's districts and outgoing mail is encoded with an appropriate Zip number. Bochum handles 100,000 letters a day.

(Continued on page 127)

(Continued from page 125)

3.4-MEGABUCK COMPUTERS FOR AIR TRAFFIC CONTROL

Britain's Ministry of Aviation has ordered three Marconi Myriad computers for processing flight plans in the U.K.'s central air traffic control complex planned for London airport. The machines, worth $3.4 million, will be interconnected. Specifications for the systems were for a processing system in which no interruption lasting more than 30 seconds could occur in five years. The Myriads are to feed CRT displays confronting the air traffic controllers.

USERS SET UP CROSS-CHANNEL DATA LINK

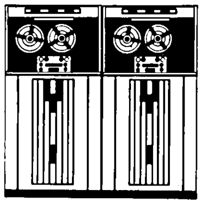
Two firms involved in the development of the Olympus 593 engines, for the Anglo-French supersonic jet airliner Concorde, have established a permanent cross-channel data link for interchange of design information. The companies are Bristol Siddeley and la S.N.E.C.M.A. Paper-tape to paper-tape links were provided by the French company, Thomson-Houston, in a network connecting Bristol Siddeley and the National Gas Turbine Establishment (near London) with each other and with the French Computation Centre at Suresnes, the jet engine research laboratory at Saclay, and S.N.E.C.M.A. at Villaroche (all near Paris).

FOURTH 1107 IN SCANDINAVIA

Denmark's largest industrial group, Burmeister & Wain, has plans for an organisation-wide dp service based on a Univac 1107 and satellite 1004's. It will cover inventory control and production scheduling through steelworks, shipyard and diesel engine manufacturing interests. At the shipyards, the 1107 will be used for design computation and for preparation of punched tape to drive numerical control machinery. This is Univac's fourth 1107 in Scandinavia. Others are at Oslo, Trondheim and Stockholm.

BITS & PIECES

Australia has placed a $1.8-million contract with Univac for computer control of overseas telegraph traffic...Western Geophysical Co. of America is to provide a service to North Sea gas prospectors with a 360/50. It will analyse seismograph recordings and indicate depths for drilling...ICT is promising lower-cost 1900 series, computers at the lower end before the end of 1967 by direct updating with micro-circuits. Compatibility with industry eight-bit-byte systems is expected. The 1900's are 24-bit machines with 6-bit character handling...CEIR Ltd. has introduced U.K.'s first independent mag tape certification service... General Electric has acquired controlling interest in De la Rue Bull Machines, the British marketing outlet formed by banknote printers De la Rue and Company Machines Bull...Elliott Automation last month shipped process control gear to East Germany worth $1 million. Chairman Sir Leon Bagrit claims first place with 50% of Europe's on-line direct digital systems. Sixty-nine are in use, and 83 on order. The installed/on-order position for Elliott is 109 in the U.K.; 13 in U.S.; 14 in West Europe; 15 in E. Europe and China, and one other...A population of 24,000 qualified programmers in Japan is reported by the Japan Electronic Industry Development Assn., whose survey estimates almost 50,000 by 1970.



By Any Computation

DATA PROCESSING PROFESSIONALS

Select

FOX-MORRIS PERSONNEL CONSULTANTS

We have a select group of clients offering unexcelled starting salaries in the $10-30,000 range. (all fees and employment costs paid by client companies).

Assignments in such areas as Software Development, Management Information Systems, Operations Research, Scientific and Commercial Programming Management.

Our National Data Processing Division is professionally recognized as the most complete, effective and confidential service in the placement industry. (Affiliate offices coast to coast).

For full details, reply in strictest confidence by letter or resume, or call collect, W. B. McNichols, Manager — Data Processing Division, 215 LO 8-2535.

FM FOX-MORRIS ASSOCIATES
Personnel Consultants
1500 Chestnut Street Philadelphia, Penna. 19102

CIRCLE 94 ON READER CARD

Griffing, Inc.

CAREER opportunities are available within a broad scope of positions with nationally recognized leaders in Research and Development, Operations Research and Data Processing fields.

FOR EXCEPTIONAL professional and financial advancement possibilities, we invite you to explore the areas most appropriate to your experience and professional objectives.

- BUSINESS PROGRAMMERS
- SYSTEMS APPLICATIONS
- SYSTEMS PROGRAMMERS
- OPERATIONS RESEARCH
- REAL TIME SYSTEMS
- SCIENTIFIC PROGRAMMERS
- DIGITAL/LOGIC DESIGN
- SYSTEMS ANALYSTS
- SOFTWARE SYSTEMS
- MATHEMATICIANS

We have an unexcelled reputation for matching your professional capabilities and goals to the right position, junior, intermediate or senior, and for conducting negotiations on the highest ethical level. All expenses paid by our clients. Courteous, confidential attention to your inquiry. Write:

GRIFFING, INC.

Computer Staffing Consultants

519 Shoreham Bldg. Washington, D.C. (202) RE 7-4754

CIRCLE 95 ON READER CARD

MANAGEMENT INFORMATION SYSTEMS SPECIALISTS

Our Traffic and Distribution Department is staffing a new MIS Group. Proven performers with management potential who are skilled in generating new approaches and ideas are required to develop and execute major systems projects for our production, marketing and distribution functions. Projects involve cost control, profit-planning, data systems studies and programs. Knowledge of O. R. techniques including linear programming, computer systems and systems analysis is essential. Degree in engineering, business, math or management sciences required. Liberal benefit package. Paid relocation to Niagara Falls.

Submit resume stating present salary to:

Mr. Frank F. Mangus
Industrial Chemicals Division
Hooker Chemical Corporation
Niagara Falls, New York 14302

HOOKER CHEMICALS PLASTICS

an equal opportunity employer

CIRCLE 96 ON READER CARD

HARVARD UNIVERSITY

COMPUTING CENTER

Is Seeking Experienced

SYSTEMS PROGRAMMERS

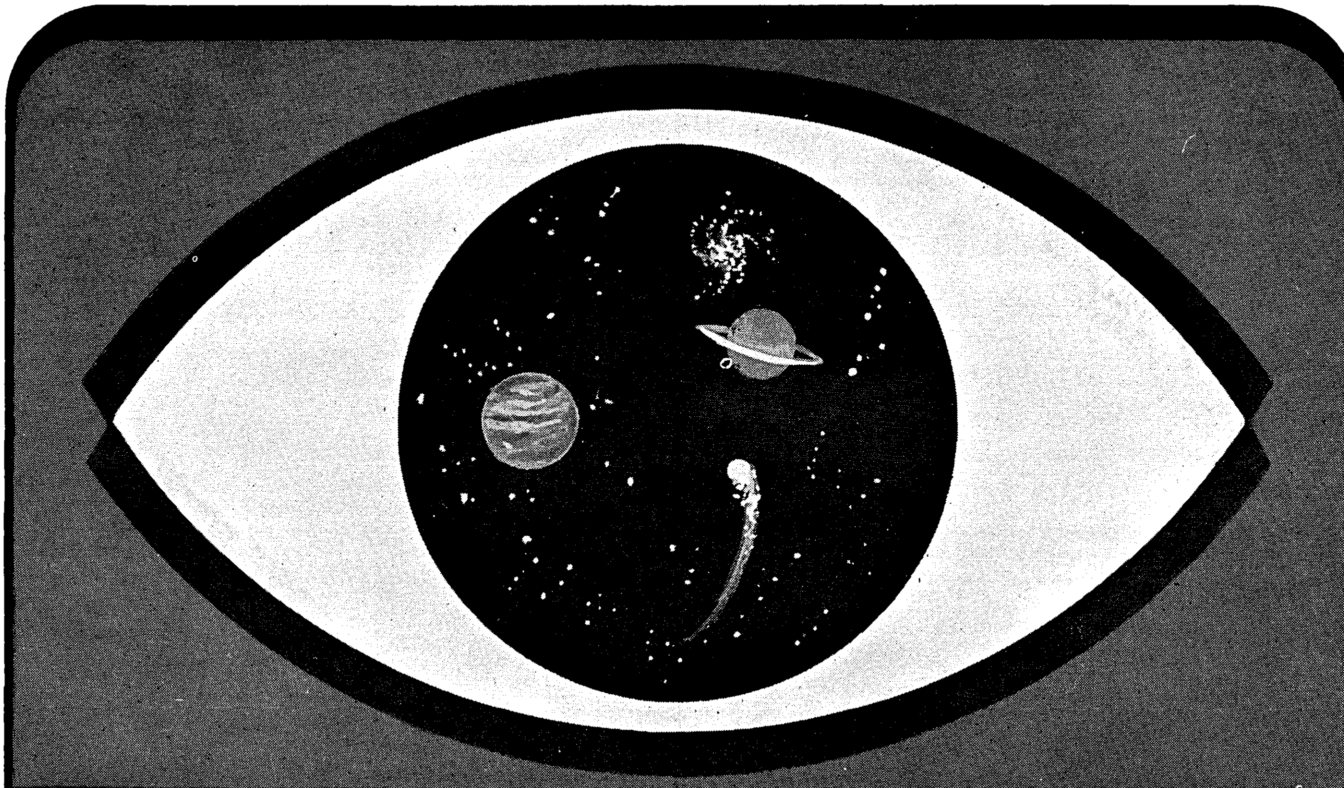
- **For Software Development on the computers we will install in the next two years. These computers range in size from the SDS 92 to the IBM 360/67 to the IBM 360/91.**
- **For Software Development and Maintenance on our operating IBM 360/50.**
- **For Software Maintenance on our two IBM 7094's.**

We offer liberal benefits, an opportunity for continuing education, and attractive salaries.

Resumes may be submitted in complete confidence; each will be acknowledged and should be addressed to W. D. Thompson, Room 635, Personnel Office, 1350 Massachusetts Avenue, Cambridge, Mass. 02138

An Equal Opportunity Employer

CIRCLE 97 ON READER CARD



"I see beyond the range of sight-new earths and skies and seas..."

(written by a noted resident of Concord, Massachusetts over 100 years ago!)

In Sudbury, Massachusetts, RAYTHEON'S Space & Information Systems Division is "seeing beyond" the lunar missions . . . is rapidly programming for future needs *now* with groups of Scientific Programmers and Systems Engineers. Openings are at all levels with breadth of opportunity to match. You will be part of a long-term industry-university project presenting rare opportunities for independent creativity. Work includes guidance and verification of finished products plus preliminary environmental simulation and development of automatic techniques.

Scientific Programmers

Challenging career openings for you in — Apollo Mission Programming · Process Control · Systems Analysis · Data Reduction and Processing · General Scientific Programming.

You'll need at least a BS in mathematics or sciences with a minimum of one year programming or general purpose activity in comparable areas.

Systems Engineers

An equal long-range challenge listed here for the Apollo mission and "beyond" — Systems Analysis · Mathematical Analysis · Numerical Analysis · Mission Design · Guidance Control.

You'll need a BS in physics or mathematics (design and analysis) or BSEE. Minimum of one year of work in these areas.

Help us achieve end results that are universally constructive by programming lunar missions and beyond.

Send resume, including salary history, to John C. Jones, Employment Manager, Raytheon Company, Space & Information Systems Division, Department J, 528 Boston Post Road, Sudbury, Mass. 01776.

RAYTHEON

EXCELLENCE IN ELECTRONICS

An Equal Opportunity Employer

CIRCLE 100 ON READER CARD

AT&T FIGHTS FCC ORDER TO INCREASE TELPAK RATES

Last July, FCC authorized non-regulated microwave users to share channels. Earlier, AT&T assistant vp Bill Ellinghaus had promised that if allowed to share microwave, these users would also be allowed to share Telpak -- a move that could cut rates substantially. Last month, AT&T objected to the microwave decision, but indicated that Ellinghaus's promise would be honored if the objection is overruled.

FCC wants Telpak rates raised and private line tariffs lowered; AT&T is fighting this decision in court. Winning the court fight, apparently, is Ma Bell's main goal. Reportedly, the company is preparing to offer not only shared Telpak channels, but also toll voice/record service; more customers would increase political support for less-than-compensatory Telpak tariffs.

GROUPS STUDY OWNERSHIP OF COMSAT GROUND STATIONS

Comsat's squabble with the carriers over ground station ownership got an extensive airing before House and Senate committees last month. Possibly the most significant statement came from FCC chairman Rosel H. Hyde: "...by not owning ground stations, they (the carriers) have nothing to make a profit on." Two committees, representing all concerned parties, are seeking a settlement of this dispute.

TRANSATLANTIC DATA LINK VIA SATELLITE IS SLATED

An IBM-Comsat experiment involving high-speed (5,000 cps) transatlantic data transmission by satellite is tentatively set to begin in October. If the test works (a virtual certainty), Comsat will propose a similar toll-rate service -- far more extensive, less expensive than what international carriers offer now. WU, RCA, and ITT WorldCom are arguing over who will handle this new business for Comsat.

A satellite can now move record data 30 times faster than existing cables, which suggests rate reduction possibilities. The test will link IBM's U.S. data net with a company lab in France.

CAPITOL BRIEFS

Sen. Javits has introduced a bill (S3740) to finance purchase of computers by local police departments for use in crime prevention...Vico Henriques, slated to join the NBS Center for Computer Sciences and Technology as the second in command and eventual successor to Norm Ream, is joining Arthur O. Young, consultants, instead...BEMA has accepted an NBS offer to prepare a proposed American Cobol standard, which should be ready for review next February...GSA has asked BOB for $45 million to launch the adp revolving fund in fiscal '68; a request, probably for less, should go to Congress next January. Proposed uses include setting up an in-house leaseback program and modifying surplus equipment for other federal agencies ...Congress cut GSA's fiscal '67 dp management appropriation by $172K (to $500K); development of criteria relating purchase specs to suppliers' contract performance may be delayed up to a year...A recent Joint Economic Committee report praises CAI (computer-assisted instruction) equipment, but says software must be improved and costs cut to 25 cents/ student hour. The report advocates coordination of CAI development through a data bank-referral center.

books

Optical and Electro-Optical Information Processing, edited by J. T. Tippett, D. A. Berkowitz, L. C. Clapp, C. J. Koester, and A. Vanderburgh, M.I.T. Press, 1965.

This book is based on a Symposium on Optical and Electro-Optical Information Processing Technology held in Boston in November 1964.

In the area of coherent optical data processing and wavefront reconstruction, several definitive and well written papers are included. Much of the original work which was carried on at the Radar Laboratory at the University of Michigan is described in two excellent papers. Leith, Kozma, and Upatnieks describe the basic concepts of matched complex spatial filtering and wavefront reconstruction. Vander, Lugt, Rotz, and Klooster discuss the application of matched filter theory to coherent optical character reading and present results of some of their early studies in this area. Cutrona carries on these basic discussions and includes additional applications to antenna pattern simulation and optical correlation. Lansraux discusses some independent work in the application of diffraction theory to image quality enhancement by amplitude filtering. Finally, Stroke discusses his work in holography, particularly toward projected applications to x-ray hologram microscopy.

Incoherent optical data processing, although somewhat older and currently less actively studied than coherent optical data processing, has seen many recent advances. There are several papers discussing processing and transmission of two-dimensional incoherent optical information and machine perception and recognition of three-dimensional optical information. There is also a very interesting paper discussing adaptive systems and the role of optical modulation.

Several papers discuss various laser or laser related devices particularly directed toward optical logic functions. The recent developments in lasing fibers provide the background for several discussions of optical-switching in fibers. A number of papers discuss other laser effects potentially useful for logic functions in all-optical or electro-optical computers from the well-known laser quenching effect to

the "dither" effect.

There is an excellent and concise review of injection lasers by Rediker and a more extensive treatment by French of the utilization of light emission from Ga As tunnel diodes for performing digital logic functions. Because of recent advances in enhancing the Kerr magneto-optic effect (rotation of the plane of polarization light incident on a specularly reflecting magnetized surface) it seems possible that magneto-optic light switches may have application in memory and display. With less than complete clarity Smith discusses some theoretical configurations which attempt to optimize the Kerr rotation by impedance matching.

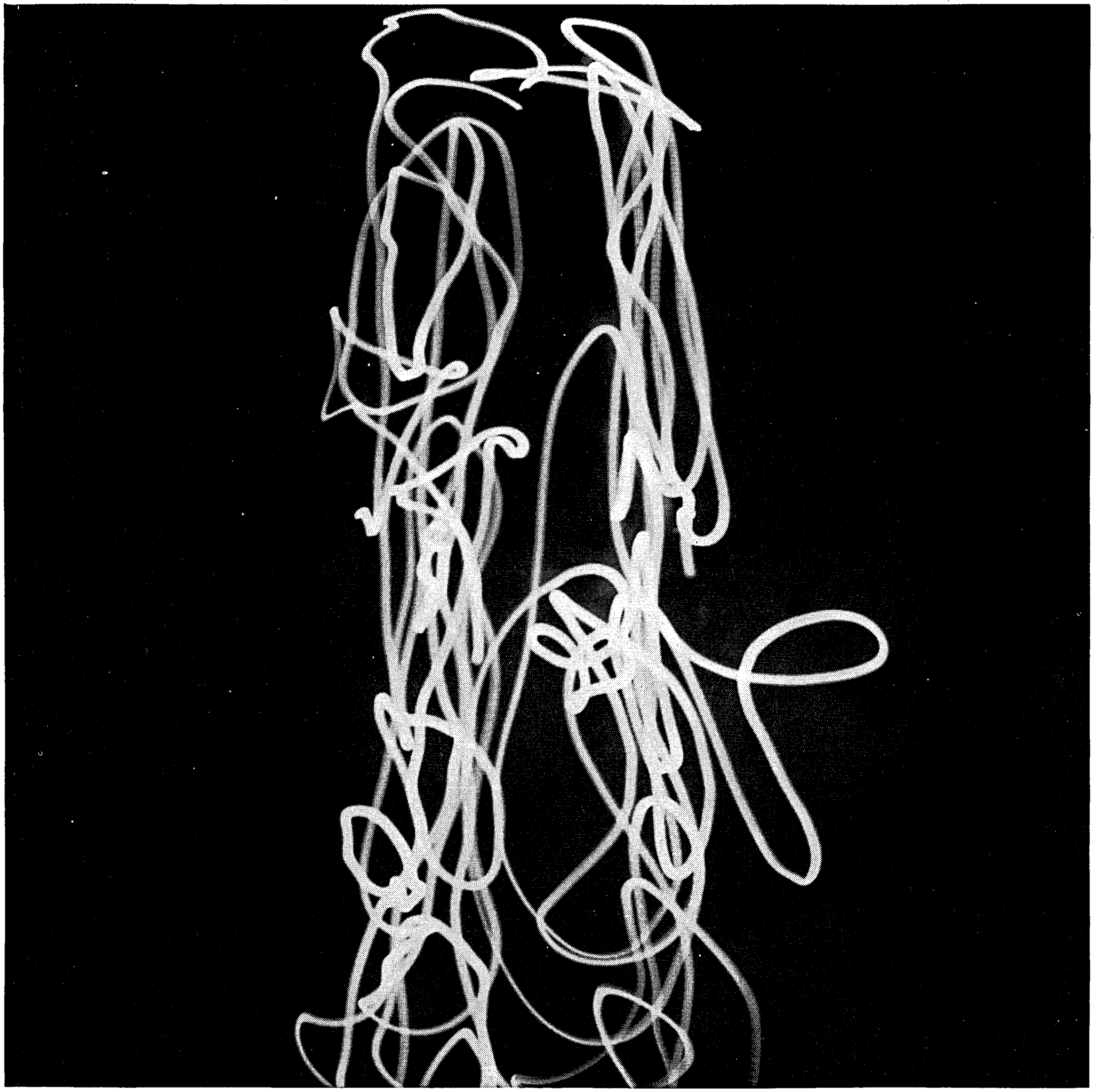
Two light deflecting schemes being investigated at IBM are described in separate papers. In one, a convergent laser beam is digitally deflected by a number of stages each of which makes use of two effects-birefringence of certain crystals (e.g. calcite) and the longitudinal electro-optic Pockels effect. In the second method, the laser is caused to emit in a particular desired direction at a given time.

In the area of storage and display there are papers on wideband readout of thermoplastic recording and a photochromic random access memory. In another paper, an unusual idea of storing conductivity changes with high resolution in thin cadmium sulfide films is presented along with some experimental results.

In addition to many research reviews and more fundamental papers, there are some interesting papers describing various operational systems, including a digital microtape recorder and an image sensor. There are also several good papers of a more fundamental nature on the information limits of optical scanning systems and detectors.

To summarize, many novel and interesting ideas and systems are discussed and several definitive and tutorial papers are included. The aim of the symposium and, consequently, the tone of its proceedings, is directed more toward the experimental physicist and the optical engineer rather than toward software people. However, the physical concepts of electro-optical data processing in many of its various possible forms are treated well enough so that the present day computer user is able to gain some insight into the next possible major generation of computers both with regard to the physical optics involved and to the new forms of logical design that will be necessary.

—ALAN HOFFMAN



Too much lost motion in your present job?

Lost motion comes from spending too much of your time and energy getting nowhere.

At RCA we feel that the only way to keep systems programmers happy is to make sure they get somewhere.

So we offer them a variety of projects with definite goals.

We offer them challenges and opportunities and room to grow.

We give them a say in hardware design.

And we pay well.

Write to us if you've had experience in language processors, operating systems, utility systems or communications systems.

We also have openings in Sales, Field Systems Support, Product Planning and Engineering.

Contact Mr. E. B. Schultz, Dept. SW-11 RCA Electronic Data Processing, Bldg. 202-1, Cherry Hill, Camden, N. J. 08101.

An Equal Opportunity Employer M & F

RCA® **The Most Trusted Name in Electronics**

CIRCLE 110 ON READER CARD

datamart

Address all replies to box number advertisements as follows:

Box..............
DATAMATION
F. D. Thompson Publications, Inc.
141 East 44th St.
New York, N. Y. 10017
Classified Advertising

PROFESSIONAL SERVICES

USED EQUIPMENT

PROFESSIONAL OPPORTUNITIES

3C

Expanding opportunities for Programmers in real time computer applications with 3-8 years experience in software diversification, diagnostics, design automation, compilers, assemblers, or I/O libraries. Your interest is invited. **Send resume in confidence** or call collect: Martin Dorfman (617) 235-6220.

Honeywell

3C COMPUTER CONTROL DIVISION

OLD CONNECTICUT PATH. FRAMINGHAM. MASSACHUSETTS

Opportunities exist in other Honeywell Divisions. Send resumes to Fred C. Laing, Honeywell, Minneapolis, Minnesota 55408. An equal opportunity employer, M&F.

CIRCLE 116 ON READER CARD

"It's devilishly clever"

"What is?"

"The New Cummins Planax continuous form perfect binding system!"

"Prove it!"

"Send for the brochure. It tells the full Planax story!"

Now Cummins makes the original office adhesive binding system—Planax Perfect Binding—available in the United States for the first time to offices of all sizes. The simple, low-in-cost Planax system eliminates the need for binderies using expensive, complex equipment. Developed in Germany a few years ago by a bindery equipment manufacturer, the Planax system swept across Western Europe. The Planax binding system is practical for all binding requirements . . . even those which are now handled less effectively with rings, staples or other gadgets. No other method so efficiently provides the features of *perfect* binding . . . edge-to-edge visibility, flat book opening, and the strength necessary for permanent records. Planax is also more economical, both in original cost as well as in operation. It's available in colors, too.

Devilishly clever? We can prove it with a demonstration. Request the new Planax brochure, or look in your phone directory under Cummins-Chicago and call for an actual demonstration.

CA SINCE 1887 **Cummins** IN BUSINESS AND BANKS ● SINCE 1887

CUMMINS-CHICAGO CORPORATION
A CUMMINS-ALLISON COMPANY
4740 N. RAVENSWOOD AVE. • CHICAGO, ILLINOIS 60640
SALES AND SERVICE IN ALL PRINCIPAL CITIES

CIRCLE 59 ON READER CARD

people

■ Robert Sackman, a former executive vp with Ampex Corp., is now president of a new manufacturer of time-series analysers and computers, TIME/DATA Corp., Palo Alto, Calif. Edwin A. Sloane, also from Ampex, is the vice president; Martin W. Fletcher, previously with Applied Technology Inc., is vp-engineering.

■ Computer Sciences Corp. has announced the appointment of division presidents: William R. Hoover, computer sciences div.; Armig G. Kandoian, communications systems div.; and Alvin E. Nashman, systems sciences div.

■ Robert M. Gordon, former applications programming manager for Scientific Data Systems, has been appointed assistant director of the computer center, Univ. of California, Irvine.

■ Dr. Arthur L. Samuel, a pioneer in machine learning and artificial intelligence, has retired from IBM after 17 years, and is a lecturer in Stanford Univ.'s Computer Science Dept.

■ Paul O. Gaddis has been named director of Westinghouse Electric Corp.'s management systems dept.

■ Richard G. Canning has been appointed to a three-year term on the Advisory Council for the Certificate in Data Processing given by the Data Processing Management Assn.

■ E. V. Scott will head the University Computing Corp.'s Dallas Technical Services (software) Dept. He was formerly southwestern regional sales manager for UCC.

■ Dr. Kurt Eisemann and Lieut. Col. Churchill K. Wilcox, USAF, Ret., have joined Computer Usage Development Corp.'s Boston operation as principal analysts.

■ Robert G. Gillespie is now manager of computer planning, commercial airplane div., Boeing Co., Renton, Washington. He was formerly manager, advanced development dept., Control Data Corp.

■ Lawrence Porter has been promoted to manager—programming, Digital Equipment Corp., Maynard, Mass.

APOLLO / GEMINI:

Business programming and man's expansion into space.

Mr. C. Kenneth De Witt
Manager, Management Data System Department
Houston Aerospace Systems Division
Lockheed Electronics Company

There are few commercial enterprises in the world which can equal, as far as the business programmer or analyst is concerned, the scope and fascination of Lockheed's work for the Manned Spacecraft Center at Houston. Here a select group is contributing to man's expansion into space and supplementing standard processing techniques with challenging new ones, including real time applications.

But the non-standard applications are even more exciting. NASA's PERT "C," already one of the largest PERT systems in existence, is being re-designed, expanded and refined for even greater flexibility and wider application. Preparation of Gemini flight programs is like reading a TV script outlining in advance the actual flight which will be followed with interest by millions of people. Configuration management takes on a new scope and meaning when used to control the numerous contractors in the Apollo program.

Information retrieval involves more than parts lists at the Manned Spacecraft Center. These information retrieval systems will be among the largest in the free world, and they are just beginning now.

Another project in which we are engaged involves a real-time application of a specially-designed computer. It receives transmissions of data from remote transducer tests. Utilizing random-access techniques and real-time processing, it records test results and feeds them back to the remote test sites.

As for hardware, the Manned Spacecraft Center has seven complete computer systems, including large, third-generation digital, analog and hybrid computers available for the support Lockheed is performing for MSC.

If these challenges intrigue you—if you want to be at the career center of man's expansion into space—then we want you on the Lockheed team where personal contribution is the chief responsibility.

Your work in Houston will permit you to undertake evening graduate courses through the doctorate level. And the family living is hard to beat. An attractive community with fine schools, sailing, fishing, golf, and lovely homes has grown up around the Manned Spacecraft Center. Theaters, museums, and all the other cultural and entertainment advantages of a large metropolitan area, are a few minutes drive away in Houston.

If you are a programmer or analyst, drop me a note today outlining your experience and qualifications. Or, if you prefer, telephone collect to my associate, Mr. R. K. Glasgow at (713) HU 8-0080. We will be glad to supply you with whatever further information you may want.

LEC
Lockheed
Electronics Company
A Division of Lockheed Aircraft Corporation
16811 El Camino Real, Houston, Texas
An equal opportunity employer

CIRCLE 126 ON READER CARD

SCIENTIFIC PROGRAMMING ANALYST

To work with a research scientific group in developing monitor systems programs and general software for a large computer facility engaged in a unique research program. A bright resourceful candidate with at least a Bachelors Degree and up to 3 years experience involving a good deal of large computer systems programming will find this a stimulating long range employment opportunity. Supervisory experience in these areas may be partially substituted for direct programming experience.

SEND RESUME
INCLUDING SALARY REQUIREMENTS TO:
ASSISTANT PERSONNEL MANAGER

Brookhaven National Laboratory

ASSOCIATED UNIVERSITIES, INC.
UPTON, L. I., N. Y.

An Equal Opportunity Employer

CIRCLE 128 ON READER CARD

WHAT IS YOUR TRUE WORTH?

FREE Data Processing Opportunities Bulletin

Every month, in the privacy of your own home, you can evaluate the nation's finest openings in the data processing field. Cadillac, the nation's largest executive and professional placement service, represents the majority of the nation's top companies. Their best jobs at salaries from $6,000 to $75,000 appear in our monthly Data Processing Opportunities Bulletin.

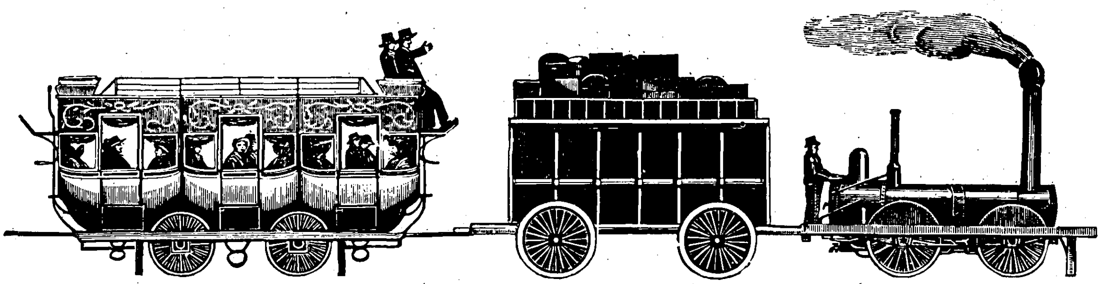
Our placements show that the average data processing man is worth 10% to 20% more than his present income. The Bulletin helps you evaluate yourself in today's market. Both the Bulletin and our *confidential* placement service are free. Client companies pay all costs.

For your free Bulletin, without any obligation circle Subscriber Service Card No. 121. Please use *home address only.*

LON D. BARTON, *President*
Cadillac Associates, Inc.*
29 E. Madison Bldg. Chicago, Ill. 60602
Financial 6-9400

* *"Where More Executives Find Their Positions Than Anywhere Else in the World."*

CIRCLE 121 ON READER CARD



COME TO AUERBACH AND WORK ON PIONEER PROJECTS

If you're a programmer, engineer, mathematician, or scientist with experience and ability in information sciences, there's a demanding (and rewarding!) assignment for you at AUERBACH Corporation.

Typical AUERBACH projects involve advanced work in data management systems, associative memory utilization, system simulation and evaluation, multi-processing systems, operating systems and digital communications.

This is the kind of work you'll find yourself thinking about at 11 o'clock at night or early Sunday morning, not because you *have* to, but because you *want* to. It's that exciting.

Positions available in Philadelphia, Washington, D. C. and New York City. Interested? Send a résumé including salary history and requirements to:

AUERBACH ®

Norman B. Barth
AUERBACH CORPORATION
121 N. Broad Street, Philadelphia, Pa. 19107
An Equal Opportunity Employer

CIRCLE 127 ON READER CARD

advertisers' index

advertisers' index

the forum

The Forum is offered for readers who want to express their opinion on any aspect of information processing. Your contributions are invited.

THIRTEEN PROGRAMMING PARADOXES

What with syntax-directed, re-entrant procedures and the like, computer software is getting harder every year to comprehend. For boss programmers, working programmers, and meta programmers alike, the situation is fraught with peril. Clearly, nobody can usefully grasp a software system when he can't even keep up with ordering (much less reading) the change documents for it. And some of the systems that are being evolved to cover the private parts of the new machines seem to be made of the same whole cloth used to manufacture the Emperor's new clothes. But did you ever stop to think that on a hot day, the Emperor might be the only one around to keep his cool. It is in precisely this spirit that the following budget of programming paradoxes is offered.

1. Software quality—its reliability, efficiency, power flexibility, elegance, simplicity, modularity, etc.—is inversely proportional to the total size of the system, the amount of paperwork required by management, the distance (in miles) between the system designers and the programmers, the number of people on the programming team, and the time spent planning the system by persons other than those responsible for programming it. (In another form, the law simply states that the quality of a program varies inversely as the cube of its cost.)
2. The description of a system bears no relation to the system itself.
3. The more sophisticated the software the less likely it is to satisfy the customer's needs.
4. The halfway mark in developing a software system is when the system programmers declare it works.
5. The larger the machine, the easier it is to write the software . . . and the worse it will be.
6. The ideal (non-existent) computer manufacturer delivers the software it says it will deliver on the day it says it will deliver it.
7. The more human-like a computer becomes, the less work it does.
8. The importance of an advance in software technology is independent of the number of papers published in favor of it, but is directly proportional to the number of papers published against it.
9. The value of a person's opinion on software is inversely proportional to the number of months since he last wrote code.
10. The world's best programmer is also the world's top computer engineer.
11. The value of a software committee is equivalent to the value of the least intelligent member divided by the number on the committee.
12. The only way of consistently producing high quality software is to carefully study top management plans and then do the opposite.
13. The more programmers who work on a program, the longer it will take to produce.

PETER D. JONES

000000001	000000010	000000011	000000100	000000101	000000110	000000111	000001000
000001001	000001010	000001011	000001100	000001101	000001110	000001111	000010000
000010001	000010010	000010011	000010100	000010101	000010110	000010111	000011000
000011001	000011010	000011011	000011100	000011101	000011110	000011111	000100000
000100001	000100010	000100011	000100100	000100101	000100110	000100111	000101000
000101001	000101010	000101011	000101100	000101101	000101110	000101111	000110000
000110001	000110010	000110011	000110100	000110101	000110110	000110111	000111000
000111001	000111010	000111011	000111100	000111101	000111110	000111111	001000000
001000001	001000010	001000011	001000100	001000101	001000110	001000111	001001000
001001001	001001010	001001011	001001100	001001101	001001110	001001111	001010000
001010001	001010010	001010011	001010100	001010101	001010110	001010111	001011000
001011001	001011010	001011011	001011100	001011101	001011110	001011111	001100000
001100001	001100010	001100011	001100100	001100101	001100110	001100111	001101000
001101001	001101010	001101011	001101100	001101101	001101110	001101111	001110000
001110001	001110010	001110011	001110100	001110101	001110110	001110111	001111000
001111001	001111010	001111011	001111100	001111101	001111110	001111111	010000000
010000001	010000010	010000011	010000100	010000101	010000110	010000111	010001000
010001001	010001010	010001011	010001100	010001101	010001110	010001111	010010000
010010001	010010010	010010011	010010100	010010101	010010110	010010111	010011000
010011001	010011010	010011011	010011100	010011101	010011110	010011111	010100000
010100001	010100010	010100011	010100100	010100101	010100110	010100111	010101000
010101001	010101010	010101011	010101100	010101101	010101110	010101111	010110000
010110001	010110010	010110011	010110100	010110101	010110110	010110111	010111000
010111001	010111010	010111011	010111100	010111101	010111110	010111111	011000000
011000001	011000010	011000011	011000100	011000101	011000110	011000111	011001000
011001001	011001010	011001011	011001100	011001101	011001110	011001111	011010000
011010001	011010010	011010011	011010100	011010101	011010110	011010111	011011000
011011001	011011010	011011011	011011100	011011101	011011110	011011111	011100000
011100001	011100010	011100011	011100100	011100101	011100110	011100111	011101000
011101001	011101010	011101011	011101100	011101101	011101110	011101111	011110000
011110001	011110010	011110011	011110100	011110101	011110110	011110111	011111000
011111001	011111010	011111011	011111100	011111101	011111110	011111111	100000000
100000001	100000010	100000011	100000100	100000101	100000110	100000111	100001000
100001001	100001010	100001011	100001100	100001101	100001110	100001111	100010000
100010001	100010010	100010011	100010100	100010101	100010110	100010111	100011000
100011001	100011010	100011011	100011100	100011101	100011110	100011111	100100000
100100001	100100010	100100011	100100100	100100101	100100110	100100111	100101000
100101001	100101010	100101011	100101100	100101101	100101110	100101111	100110000
100110001	100110010	100110011	100110100	100110101	100110110	100110111	100111000

Boeing Needs 100111000 Computer Technology Specialists

That's right—there are presently some 312 positions in various levels for both analog and digital engineers and programmers. If you are a qualified graduate in engineering, a physical science, mathematics, production management or business administration, and interested in advancing your career, check these Boeing assignments:

Manufacturing & Business Data Processing Systems—Assignments involve programming in such areas as production control, manpower forecasting, finance, facilities, quality control, material inventory control and management information systems.

Engineering Computing Systems—Work involves programming engineering applications (structural analysis, digital simulation, fluid dynamics, propulsion systems analysis, etc.) with emphasis on the integrated system approach. Use of geometric mathematical models is also involved.

Computer Methods & Standards—Responsibilities include: development of computer software, establishing computer standards, evaluation and selection of digital computer equipment.

Analog Computation & Flight Simulation—Assignments include developing and applying simulation techniques to the solution of complex aircraft problems, using such techniques as adaptive design, mathematical models and hybrid methods.

Systems & Operations Research—Systems Research positions involve studying and developing analytical models in support of technical management for evaluation of alternate airplane or system design concepts. Operations Research assignments include research in management sciences involving decision-making and operational problems, and assisting in the formulation and solution of these problems.

These positions offer an unprecedented opportunity to advance your career with Boeing in Seattle, made possible in part by Boeing's $3 billion company backlog. Another growth and stability factor is the recent announcement of the 747 jetliner, which will be the largest commercial jet in the history of aviation, seating up to 490 passengers.

To investigate these career assignments, please send your resume today to: Mr. Edwin A. Johnson, Commercial Airplane Division, The Boeing Company, P.O. Box 707-CEQ, Renton, Wash. 98055.

BOEING

COMMERCIAL AIRPLANE DIVISION
SUPERSONIC TRANSPORT DIVISION

An equal opportunity employer

At MAC, our design efforts are directed toward our customers' needs. As soon as we realized there was an urgent need for a more practical approach to handling short run computer applications, MAC "TransiTape" was born . . . "MAC 600" was a natural follow-up. Compact, easy to store, inexpensive to mail and ideal for short jobs, these two reels let your needs decide which is best for you. If you have a short job of 200 feet or less, "TransiTape" is the answer—200 feet of 800 BPI ½" heavy-duty computer tape on a 6-inch reel with all the quality features of larger MAC reels. The "MAC 600" offers 600 feet of 800 BPI heavy-duty ½" computer tape on a 7-inch reel for jobs of small or medium length. We're grateful for the tremendous response. If you're already using "TransiTape" and "MAC 600" reels to good advantage, we appreciate it. If you're not, see your MAC representative. MAC

Or write MAC Panel Company. Mac Panel Co./High Point, N. C.

CIRCLE 2 ON READER CARD

The astonishing performance of Secret Agent, Glass Bond.

See how Ferroxcube's Secret Agent, Glass Bond, gets up to five times greater packing density out of a recording head than arch enemy Epoxy Bond. It's done with your existing plating or coating and no increase in electronics by shrinking gap lengths to 50 microinches or less.

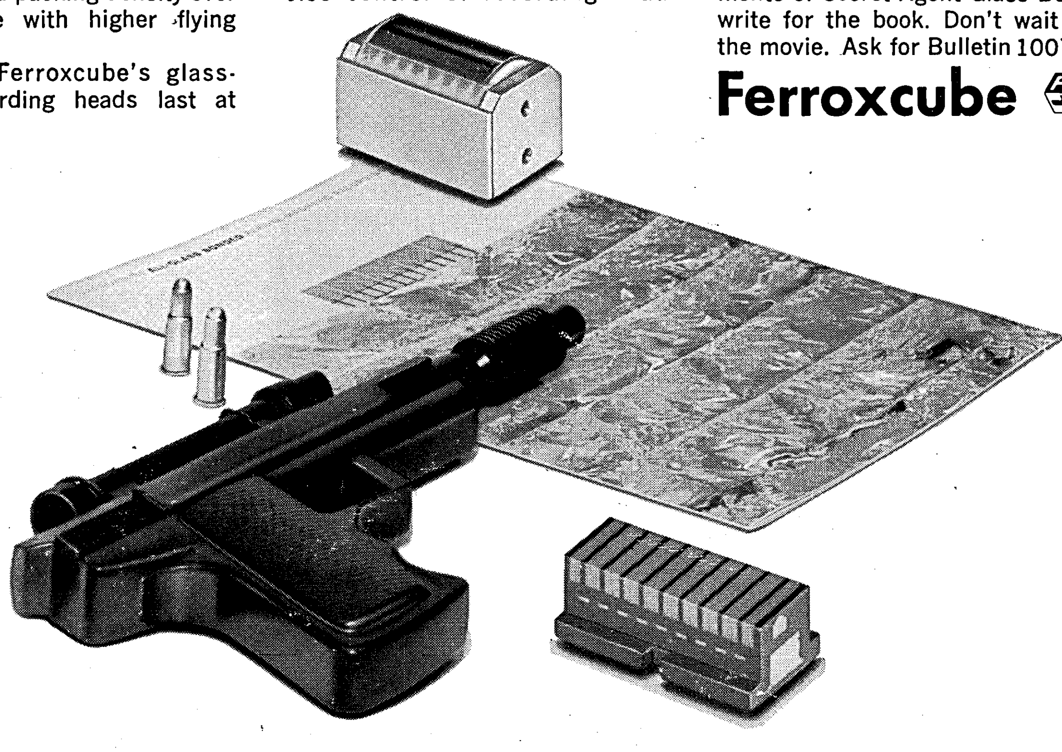
Thrill to flying heads that dare to fly lower (less than 100 microinches) with Glass Bond than with conventional agents. The resulting reduction in space losses means significant improvements in output amplitude and packing density over that possible with higher flying assemblies.

Learn why Ferroxcube's glass-bonded recording heads last at least ten times longer in contact recording applications than laminated metal types and at least three to four times longer than epoxy-bonded ferrite heads. In comparative life tests, wear on a glass-bonded head was almost imperceptible at the time an epoxy-bonded head reached end-of-life.

The secret of our Secret Agent is the unique combination of Ferroxcube's new ferrite material, called 4R5, a special glass-bonding agent, and an exclusive fabricating process which permits more precise control of recording head geometry than any other manufacturing technique. 4R5 and our special glass agent are fully compatible with one another providing a mechanically homogeneous structure which is almost impervious to external environmental stress. And there is no sacrifice in electromagnetic properties; 4R5 has excellent permeability, very low residual retentivity and material and gap losses are negligible in most applications over a wide frequency range.

For the latest performance achievements of Secret Agent Glass Bond, write for the book. Don't wait for the movie. Ask for Bulletin 1007B.

Ferroxcube



FERROXCUBE CORPORATION OF AMERICA
Saugerties, New York 914-246-2811

Boston	Chicago	Dallas	Los Angeles	Minneapolis	New York Area	Phoenix	Philadelphia	Washington, D.C.
617-889-3110	312-261-7880	214-255-0441	213-837-1484	612-888-4681	201-267-5888	602-265-1792	215-927-6262	703-893-4900

CIRCLE 3 ON READER CARD