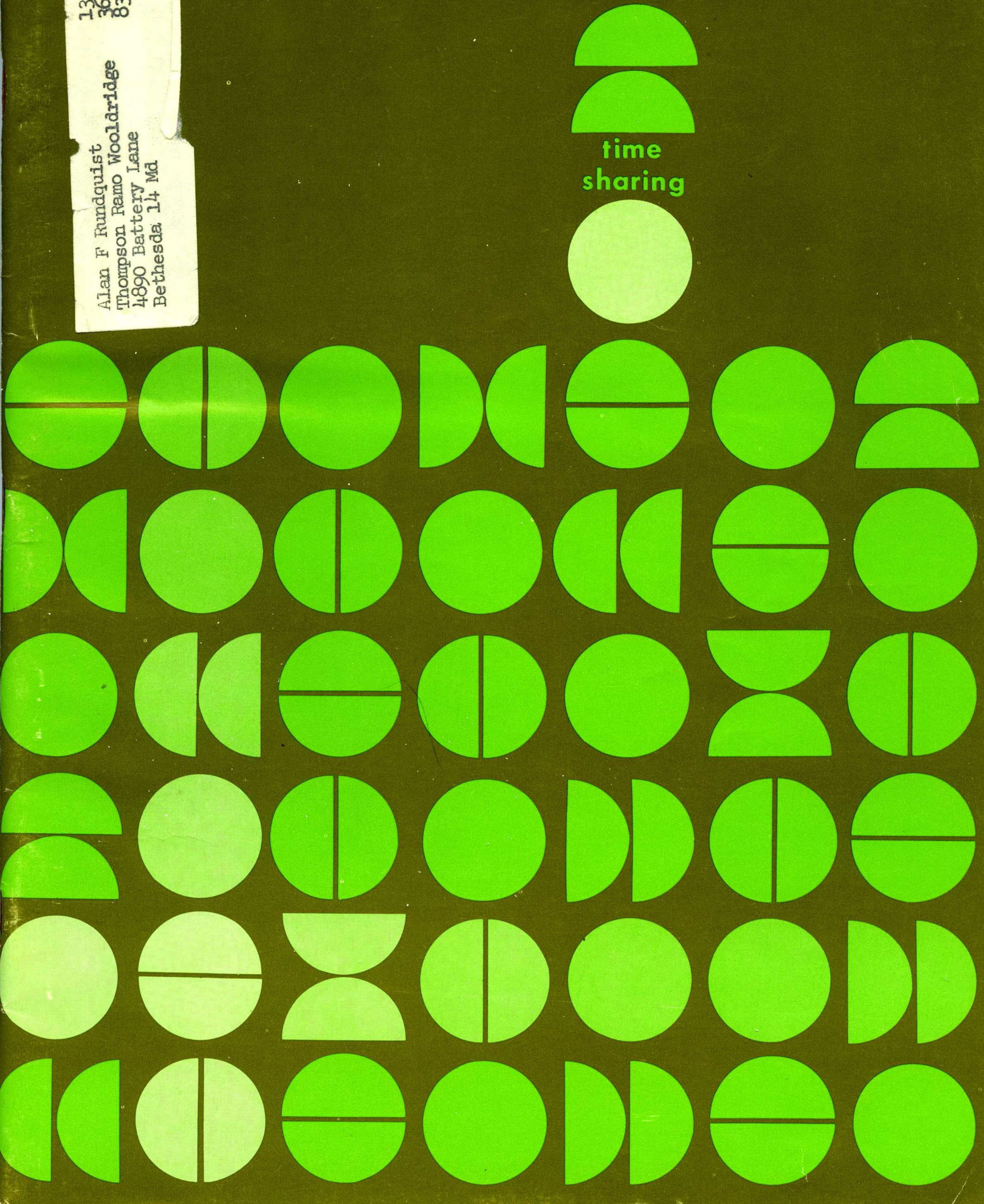
DATAMATION

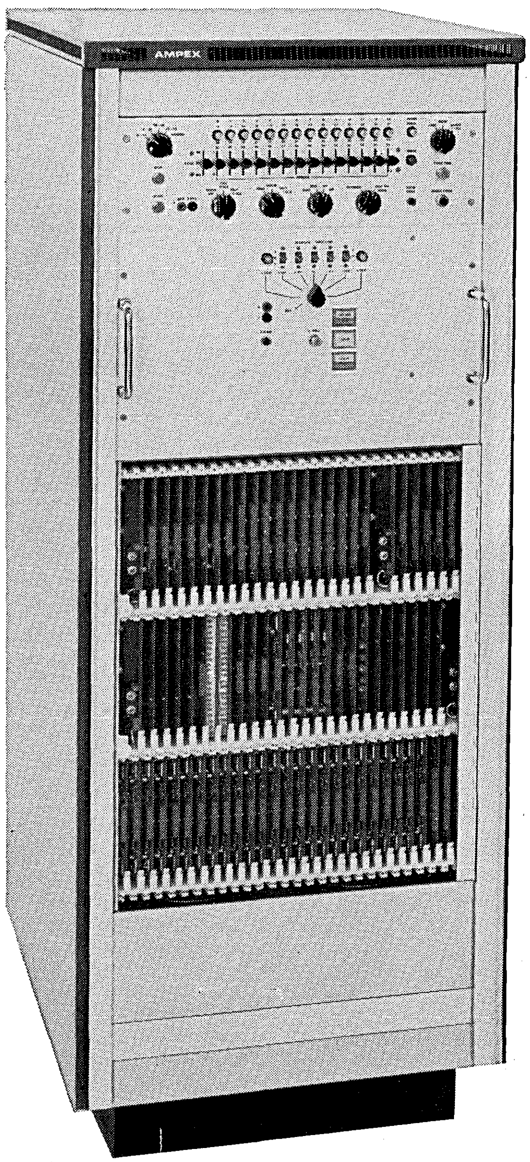
64

November

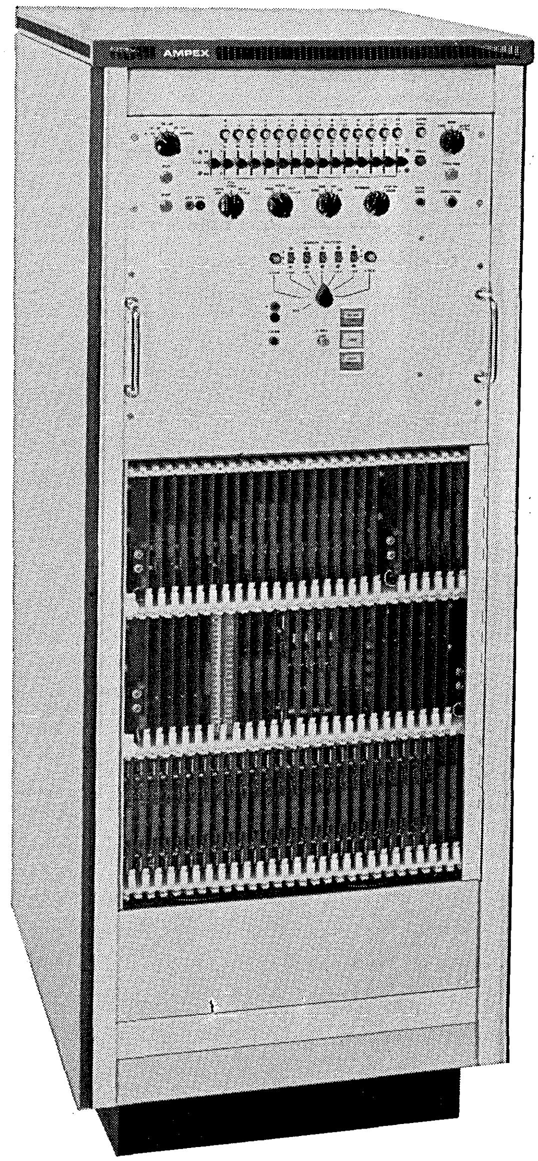
time sharing



1.8 *u*second



1.0 *u*second



NEW COINCIDENT CURRENT MEMORIES

Newest of the proven R Series systems, the RS and RZ are practically identical—except for memory cycle time speeds and price. And the price of the 1.0 RS averages only 20% higher than the 1.8 RZ. If your immediate requirements are for a 1.8 system, but you might need a 1.0 in the future, the RZ is an ideal selection. It can be quickly (24 hours) and economically converted into a 1.0 RS—just by plugging in a new stack, replacing two card types and making minor component changes. Both models have a capacity of 16,394 words—in 4,096 modules, 8 to 56 bits per word. (Illustrated is the 8 K model.) Temperature range is 0°C to 45°C without the use of current compensation, stack heaters or other compromising gadgetry. MTBF is high—a product of the stringent derating practices that have yielded unswervingly high performance from the Ampex R Series systems. Because of their modular magnetics, both systems are easily expandable with minimal circuit redundancy. A wide variety of options are available: data bus selection, zone control, parity generation and checking, built-in or remote tester. For details, call your Ampex representative, or write Ampex Corp., Redwood City, Calif.

AMPEX

CIRCLE 1 ON READER CARD

$28,500

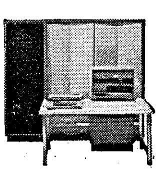
NEW DDP-116 COMPUTER
16-BIT WORD / 1.7 μSECS CYCLE / 4096 MEMORY

Standard features: Keyboard and paper tape I/O unit, comprehensive instruction repertoire, powerful I/O bus structure, multi-level indirect addressing, indexing, priority interrupt, extensive software package, diagnostic routines. Add time is 3.4 usecs. Options include high-speed arithmetic option, memory expansion to 32,768, direct memory interrupt, real time clock, full line of peripherals.

3C

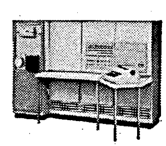
3C DISTRICT SALES OFFICES: NEEDHAM, MASS.; SYRACUSE, N.Y.; COMMACK, L.I., N.Y.; LEVITTOWN, PA.; CLEVELAND, OHIO; SILVER SPRING, MD.; DES PLAINES, ILL.; ORLANDO, FLA.; ALBUQUERQUE, N.M.; PALO ALTO, CALIF.; LOS ANGELES, CALIF.; HOUSTON, TEX.; HUNTSVILLE, ALA.

COMPUTER CONTROL COMPANY, INC.
OLD CONNECTICUT PATH, FRAMINGHAM, MASS. • 2217 PURDUE AVE., LOS ANGELES 64, CALIF.



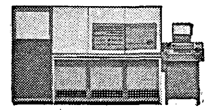
DDP-224/$96,000

24-bit word, 1.9 μsecs, 4096 word memory. 260,000 computations per second.



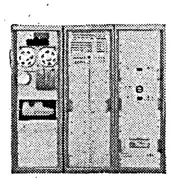
DDP-24,/$79,000

24-bit word, 5 μsecs, 4096 word memory. 100,000 computations per second.



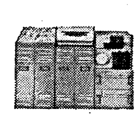
DDP-24A/$69,000

Same mainframe features as DDP-24 with modified I/O package.



DDP-24VM/$87,000

Functionally identical to the DDP-24. Rugged, compact, van mounted.



DDP-24P/Quotes on Request

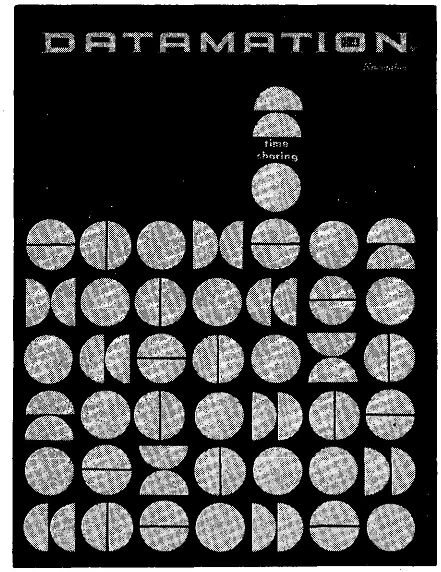
Ultra compact modular configuration for submarine installation.

CIRCLE 4 ON READER CARD

DATAMATION®
november
1964

automatic
information
processing
for business
industry & science

november
1964
volume 10 number 11



COVER

The interlocking comings and goings of time-sharing are the *raison d'etre* of the dynamic time segments in this month's cover design by Art Director Cleve Boutell.

President	**FRANK D. THOMPSON**
Publisher	**GARDNER F. LANDON**
Editor	**ROBERT B. FOREST**
Managing Editor & Art Director	**CLEVE MARIE BOUTELL**
Assistant Editor	**EDWARD YASAKI**
European Editor	**DON B. PRELL** F. D. Thompson Publications, Inc. 27 Baker St. London W.I., England
Washington Reporter	**JAMES McCROHAN**
Editorial Advisers	**DANIEL D. McCRACKEN** **LOWELL AMDAHL**
Technical Consultant	**ROBERT L. PATRICK**
Production Manager	**MARILEE PITMAN**
Circulation Mgr.	**ALGENE TRAINA**
Eastern District Managers	**FELIX CARNEY** **JAMES M. MORRIS** 141 E. 44th St., New York N.Y. 10017 MUrray Hill 7-5180
New England District Manager	**WARREN A. TIBBETTS** 112 West Haven Rd., Manchester, N.H., NAtional 5-9498
Midwest District Manager	**JOHN BRENNAN** 205 West Wacker Drive, Chicago, Ill. 60606 Financial 6-1026
Western District Manager & Vice President	**HAMILTON S. STYRON** 1830 W. Olympic Blvd., Los Angeles, Cal., 90006 385-0474

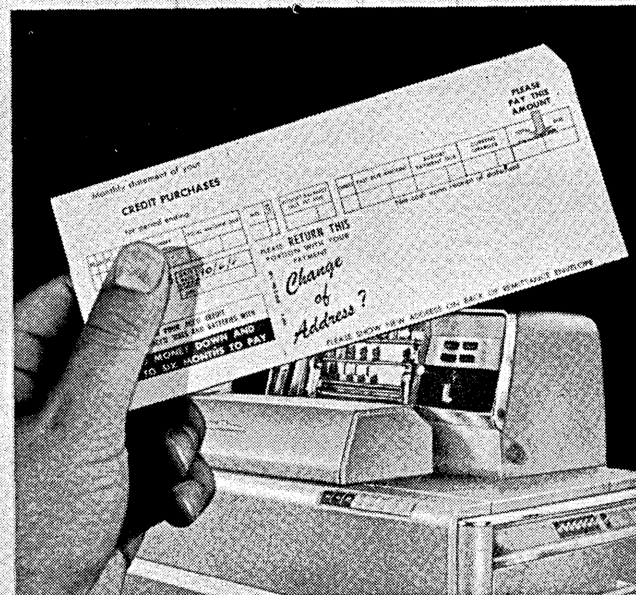
EDITORIAL OFFICES
1830 WEST OLYMPIC BLVD.
LOS ANGELES, CALIF. 90006

BPA Circulation audited by Business Publications Audit   Member, National Business Publications NBP

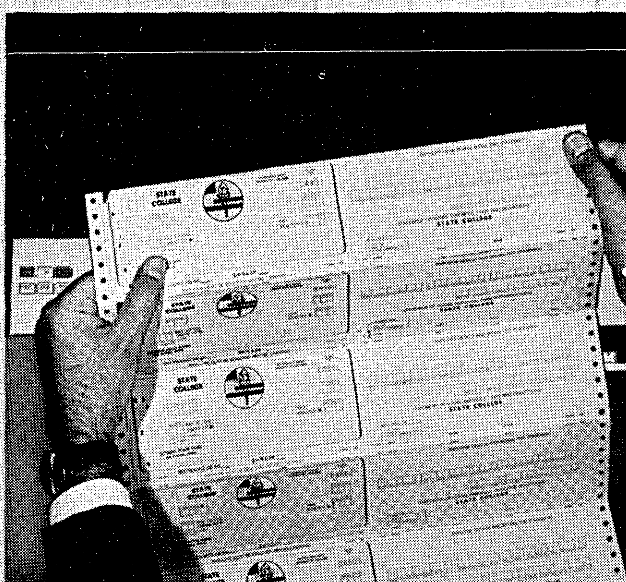
Microfilm copies of **DATAMATION** may be obtained from University Microfilms, Inc., 313 No. First St., Ann Arbor, Michigan. DATAMATION is published monthly on or about the tenth day of every month by F. D. Thompson Publications, Inc., Frank D. Thompson, President. Executive, Circulation and Advertising 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. 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. Form 3579 to be sent to F. D. Thompson Publications, Inc., 205 W. Wacker Dr., Chicago, Ill. 60606.
Printed by Beslow Associates, Inc.

THIS ISSUE— 53,059 COPIES

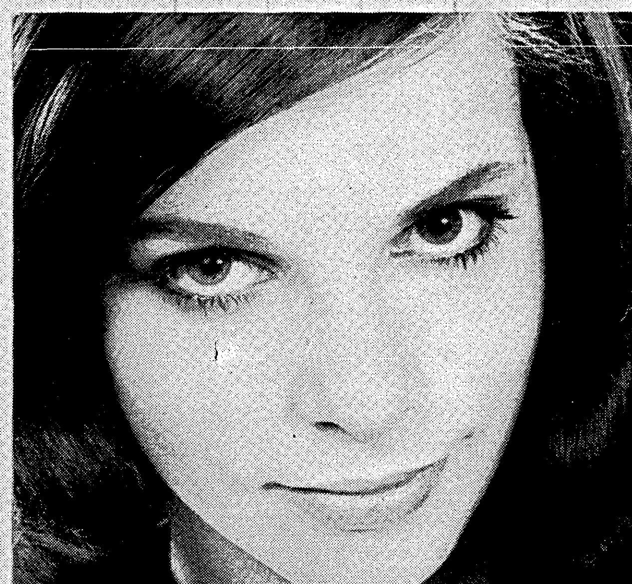
Moore Tab Card Forms



DOCUMENT CARDS, for statement billing and many other uses, have return stub as direct input into system, automating account reconciliation.



SIDE-BY-SIDE VOUCHER CHECK FORMS reduce the cost of payroll preparation and automate this function. Grain-long for efficient processing.



A WELCOME SIGHT in any data processing system, Moore tab card forms promote efficiency and smooth work flow with uninterrupted tab runs.

Look to Moore for every tab card requirement—for Moore offers tab cards for every forms construction. Ask your Moore man about:

ONE- AND TWO-WIDE LITHOGRAPHED CONTINUOUS CARDS—to give you high-speed continuous writing on automatic equipment. TWO-WIDE CARDS—cut machine time in half. DOCUMENT CARDS—with perforation to permit return stubs for automatic re-entry into your system. VOUCHER CHECKS—continuous tab card checks and vouchers without a spacer chip for use on any detacher, with check on top or bottom of voucher. TAB CARD SETS—provide for automation of source documents, for machine or handwritten use, with all copies needed for system use. 'PIGGY BACK' CARDS—combined with Continuous, Fanfold, or Speediset forms.

DATAMATION calendar

● A course on digital simulation for engineers, operations analysts, and technical management will be presented Nov. 13-20 at the Edgewater Inn in Long Beach, Calif. Sponsored by Control Technology, Inc., course prices range from $100 to $350.

● The California Assn. of County Data Processors will hold their fall business conference Nov. 18-20 at the Jack London Inn in Oakland, Calif.

● A conference on accounting applications of CPM and PERT procedures will be held by the National Assn. of Accountants at Cleveland's Statler-Hilton Hotel, Nov. 19-20.

● A workshop on data processing and computing in secondary schools will be held at the University of California, Davis branch, Nov. 20-22.

● Buying by computer is the title of the intensive seminar in dp for purchasing which will be presented in Boston, Nov. 30-Dec. 2; Los Angeles, Dec. 7-9 and, New York, Dec. 14-16. Sponsor is the National Defense Education Institute which is a joint enterprise of the National Security Industrial Assn. and Harbridge House, Inc.

● The second meeting of OPUS will be held on Dec. 2 at the State Department, Wash., D.C. OPUS, consisting of representatives of installations using the Datatrol OPCON system for the IBM 1401/60/10, was formed to provide a forum for exchange of ideas and common applications utilizing OPCON.

● Automation Training Center will conduct a course on "EDP Audit and Controls," Dec. 7-11 at the Hotel Westward Ho, Phoenix. For information, write to Director, Automation Training Center, Box 3085, Scottsdale, Ariz. 85257.

● Papers on the impact of batch fabrication on future computers are being accepted for the conference which will be held March 2-4 in Los Angeles. This event is sponsored by IEEE.

letters

more on npl

Sir:

Mr. McCracken mentions the lead-balloon career of FORTRAN IV. It is interesting to speculate as to what the current status of that language would be if all the statements in II were allowed in IV. (Such as is done by at least one company). I suspect that a capability of running old programs while slowly learning the fine details of IV would have made the latter a resounding success by now.

Since NPL is unique unto itself, two serious problems arise—retraining programmers (who are not very productive during training), and converting existing programs (a not inconsiderable task for large organizations). Both of these problems would be ameliorated considerably if the new language would accept FORTRAN II and IV, even inefficiently.

If another programming language needs to be added to the already considerable variety (and this point should be debated to considerable extent by the actual users and programmers before charging ahead), an approach considerably better than change for the sake of change as represented by NPL is necessary—preferably one that retains most of the results of past training and effort. I suggest that a much better contribution to the American computing field would be a complete ALGOL compiler which has additional capabilities where IBM has to be different.

Ernest Stiltner
Boulder, Colorado

Sir:

Mr. McCracken's stimulating article on the New Programming Language in the July issue was illustrated by an example of file processing (Fig. 4) which contained an error in logic. As the example now stands, the program will process the first transaction record, print the results, and then return to read the master file. The master file will then be read to the EOF. Termination will occur by way of the error exit.

Suppose the coding were changed so that the program returns to read the transaction file. When the program finished processing the first transaction record, it would return to read the transaction file, and then read the master file also. Unfortunately, if there are several transaction records with the same model number, the master will again be read to EOF, and termination will again occur by way of the error exit.

In coding this problem, another consideration is involved. A "merge" type program involving an input tape and a master tape is never programmed (at least not at Douglas Aircraft Corp.) without allowing for the possibility that an input item may not be found on the master file because of input errors, key punch errors, stacking errors, or redundancies.

Robert A. Grotz
Programmer, Douglas Aircraft Corp.

target too broad

Sir:

Mr. Fimple's article in the August issue would be more aptly titled "7090 hand modified FORTRAN II vs. 7090 COBOL for business dp on the 7090." Mr. Fimple claims to have analyzed the *language* when it is clear that he has compared a single implementation of COBOL to a highly hand-coded FORTRAN II system. The conclusions he reaches regarding the language are contrary to our experience with the use of COBOL on many types of equipment (including 7094's) over a 2½-year period.

A. J. Whitmore
Consultant, Business Systems
Westinghouse Electric Corp.

class conscious?

Sir:

The real Ned Kelly was a 19th century equivalent of Al Capone. He was idolized by the convict population of Australia who were jealous of the upper-class Australians. It is interesting, therefore, to have a new Ned Kelly on the computer scene of Australia. Could it be that some one is jealous of the upper-class computer company?

It is also interesting to note that the new Ned Kelly carefully avoided mentioning the reportedly massive discount CDC gave to clinch the 10 megabuck deal in Australia.

Raj Reddy
Computer Science Dept.
Stanford University

The pseudonym "Ned Kelly" was chosen with knowledge and malice aforethought. Rather than jealousy, we would like to suggest the traditional Australian trait of rooting for the best man, regardless of class. As for reported discounts, this is another tradition ... of those who lose orders, regardless of class.

gladly

Sir:

Could you kindly publish corrections to the table of FORTRAN characteristics in the August 1964 issue applying to Univac III.

Univac III FORTRAN does possess statements for magnetic tape; but does not have typewriter capability.

P. Ravitch
Manager of Technical Services
Univac III

reader scores

There has been enough confusion over whether K is 1000 or 1024. Let's not further the chaos by defining a "score" as 10 years instead of the usual 20 as was done in October's Editor's Readout.

It is indeed a magic time when you can squeeze two score years between 1964 and 1984.

Edward W. Ver Hoef
Systems Analyst
Defense Communications Agency
Washington, D.C.

(1) Model 33 Keyboard Send-[illegible] [illegible] printed copy at 100 wpm. (2) Model [illegible] [illegible] receives printed copy at 100 wpm. (3) Model 35 Automatic Send-Receive set sends and receives page copy and punched paper tape at 100 wpm. (4) Telespeed

WHICH TELETYPE MACHINES

All Teletype machines have the real-time capability of instant reception for transmissions of messages and data. Their speed and accuracy are essential to any system that continually adapts and relates changing facts to changing situations.

COMPATIBLE WITH BUSINESS MACHINES

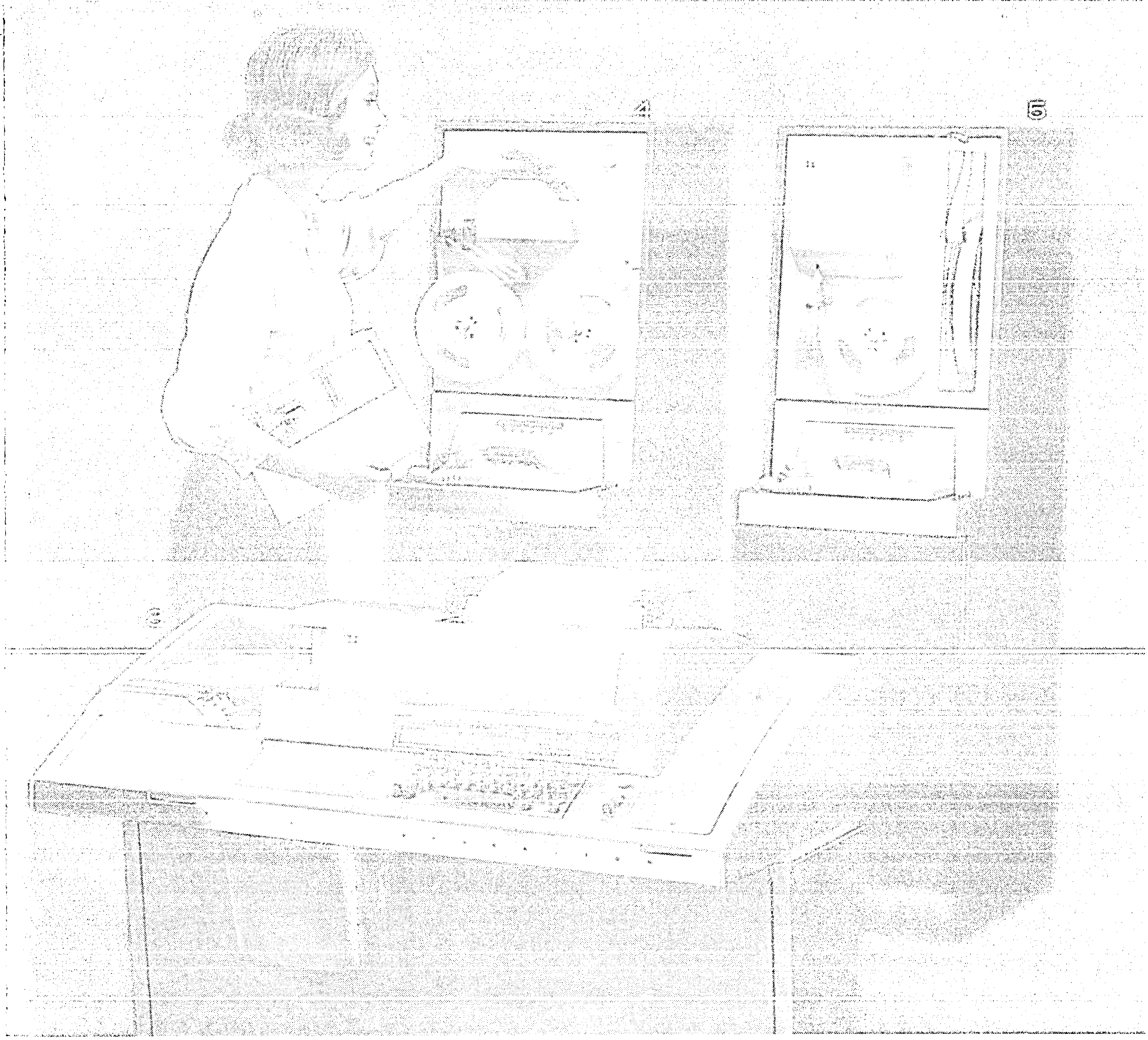
In addition, the Models 33 and 35 ASR sets and Telespeed tape-to-tape equipment transmit punched paper tape on the same code as most business machines. This means data acquisition and other computer functions can be initiated at distant locations and transmitted to the central computer.

DEPENDABLE REAL-TIME OPERATION

Built for dependable operation with a minimum of maintenance, Teletype communications equipment is used in real-time data processing systems, telemetering, and computer transmissions. For instance, real-time computers have been combined with Teletype terminal equipment in business and industry to help level off production peaks and valleys, to improve order processing and billing, and to quicken highly sophisticated systems like the following.

LINKS BRANCHES TO REAL-TIME COMPUTERS

A large electrical manufacturer uses Teletype machines



[illegible] punched paper tape [illegible]

HAVE REAL-TIME CAPABILITIES?

to link its nationwide offices, factories, and sales branches with a pair of real-time computers. As a result, they have cut order processing time from days to minutes. In addition, this real-time system is used for other data processing functions including audit and assembly of financial results, production of payroll [illegible] stock [illegible] [illegible]

PROVED RELIABILITY

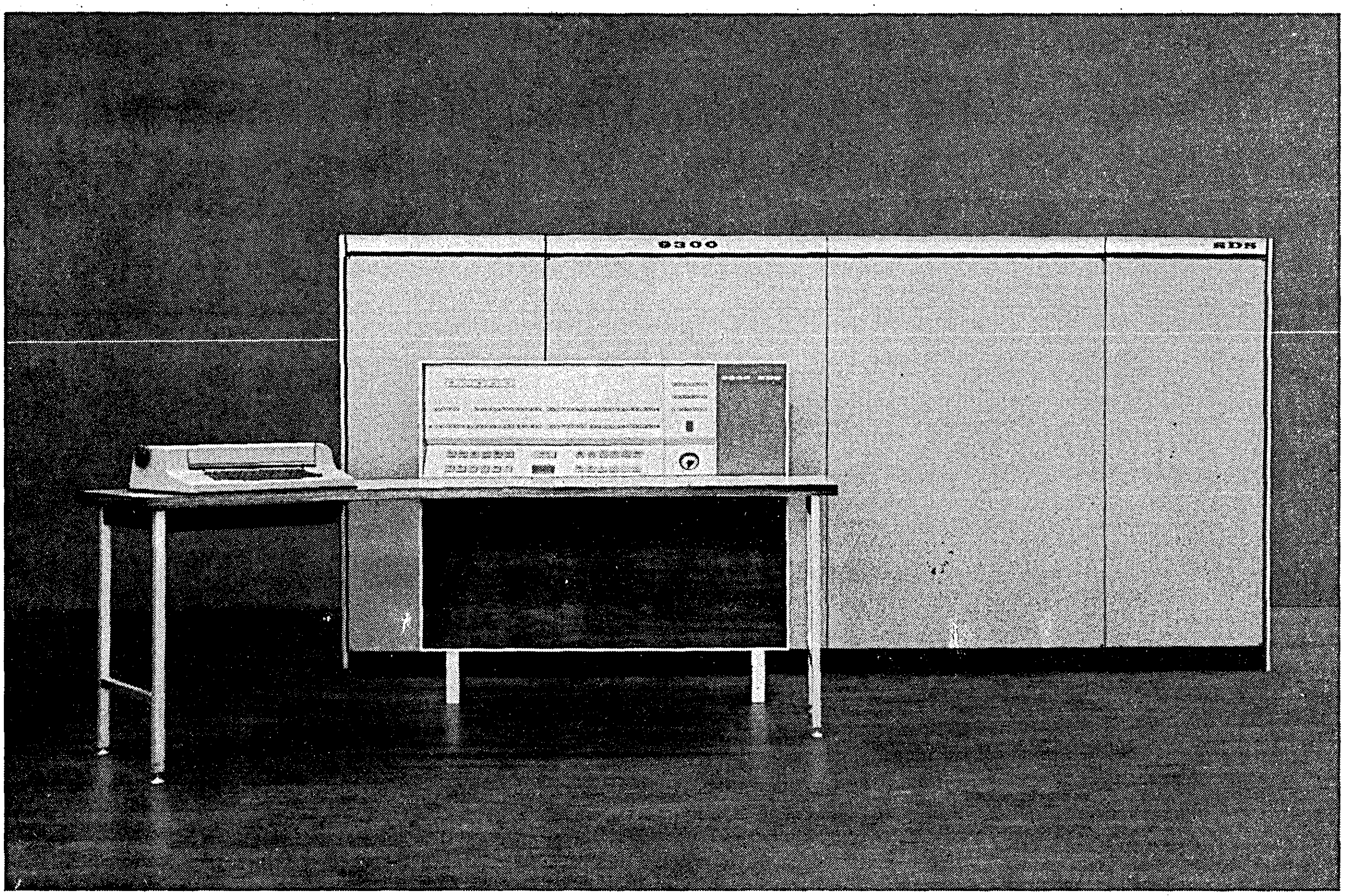
The reliability of Teletype equipment has been repeatedly proved over the years, and it's still the fastest, most accurate means of communicating written data. That's why this kind of equipment is made for the Bell System and others who supply the nation's communications services. Incidentally, the brochure he's reading is real, too. It contains information on the real-time uses of Teletype equipment and you can obtain a copy by writing: Teletype Corporation, Dept. [illegible], 5555 Touhy Avenue, Skokie, Illinois 60078

TELETYPE

If your application is real-time and you care about availability, reliability and cost...

There is exactly one scientific/engineering digita computer you can buy...

This one:



Availability: The SDS 9300 is here. It's running. It's real. It's in production. If you'd like to see one in operation yourself, come out to our plant and run your own problem on the 9300.

Reliability: How do *you* project reliability? Component count? The SDS 9300 has the lowest count of any machine in its class (up to five times less!).

Cost: The SDS 9300 costs less to buy and less to operate. You can have a complete 9300 system (8K memory, I/O, four tape stations, high speed printer, card reader and punch) for <u>$360,000</u>. You'd pay at least 20% more for a less powerful and flexible machine.

Do you have all the facts on the SDS 9300? A brochure awaits your inquiry.

SDS 9300 — $177,500 with 8,192 words of memory, Control Console, 6-bit Time Multiplexed Communication Channel, and Input/Output Typewriter.

Basic core memory of 4,096 words, expandable to 32,768 words, all directly addressable. One standard and as many optional buffered input/output channels as needed, all of which can operate simultaneously with computation.

Memory cycle time: 1.75 μsec

Execution times, including all accesses and indexing:

Fixed Point (24 bits plus a parity bit)

1.75 μsec	Add
3.5 μsec	Double Precision Add
7.0 μsec	Multiply
5.25 μsec	Shift (24 positions)

Floating Point
(39-bit fraction, 9-bit exponent)

14.0 μsec	Add
12.25 μsec	Multiply

SDS SCIENTIFIC DATA SYSTEMS 1649 Seventeenth Street, Santa Monica, Calif.

Sales offices in New York, Boston, Washington, Philadelphia, Pittsburgh, Huntsville, Orlando, Chicago, Houston, Albuquerque, San Francisco. Foreign representatives: Instronics, Ltd., Stittsville, Ontario; CECIS, Paris; F. Kanematsu, Tokyo; RACAL, Sydney.

CIRCLE 14 ON READER CARD

cai: software house in a hurry

With a flurry of acquisitions, Computer Applications, Inc., NYC software house, is closing in fast on the leaders in its corner of the industry. Formed in '60, the company began to take off when Jack de Vries, former IBM regional programming systems support manager, came in as president in the fall of '61. What was then a 15-man, nearly $200K operation has now grown into a 400-man organization which may gross $5-million in FY '64.

Within the past few months, CAI has swallowed two older firms, Orchard-Hayes and Computer Concepts. CC has a staff of 70, grossed $1.2-million in '63. Three other acquisitions are being consolidated into a suburban New Jersey service bureau. Main contracts include running the computing show for NASA's Goddard Institute for Space Studies in NYC, and NASA's Launch Operations Center in Cocoa Beach, Fla., the latter under subcontract to LTV.

Perhaps the most significant software contract is for the RCA 3301 COBOL compiler, but the firm has also done work for IBM, CDC, and Univac.

The company has an office in San Diego, and with the O-H and CC acquisitions, will inherit offices in Washington and L.A.

The acquisitions may push CAI ahead of Computer Sciences Corp., which grossed just under $4-million in FY '63, has some 250 employees. But CSC is reported to have swallowed the Communications System Division of ITT, and Doc. Inc., D.C. software firm.

banker/service bureau man takes his stand

Feeling perhaps a little schizoid is CEIR vp Bob Holland, whose boss, Herbert W. Robinson, is a key leader of the fight against banking's invasion of the service bureau business. Holland is also the founder of a Virginia bank and on the boards of others. He still maintains dp is no business for banks . . . not at the risk of depositor funds. Holland's bank, by the way, uses a GE service bureau.

Although not all bankers agree that they belong in the service bureau business, they are rapidly expanding the services they offer, will move into portfolio and investment analysis, perhaps stock analysis. One executive, Wells-Fargo's Bernard Hogan, said at the BEMA show last month that the service bureau business is extremely marginal,

ADVERTISERS' I

but that banks should and will get into the management and financial consulting game through their computer activities.

poor man's time-sharing pleases dartmouth

Our informal survey of time-sharing recently overlooked Dartmouth, where a GE 235/ Datanet 30 is serving users at 22 mod 35 Teletype stations. A simplified algebraic language — BASIC — has been designed for the system, and is taught in three one-hour sessions to nearly all undergrads. Users also can use a subset of ALGOL (without recursive features); a version of FORTRAN will be ready within a year. The system will be expanded to 30 remote stations. Computation Director Tom Kurtz says the system doesn't offer the flexibility and power of Project MAC's, but notes that "ours cost a lot less."

raytheon begins operation turnaround

Under new management and a new boss, the old Packard Bell computer operation tries to get off the floor. Head man Joe Ricca, responsible for the money-making Aeroneutronic Biax memory, is moving "with caution and humility." But he feels that the PB 440 can be sold, if its market is more sharply defined, is hiring salesmen to push it. Even the PB 250 may not be dead yet; nine were recently ordered by one firm. Investigations of a successor to the 250 are under way. Ricca plans to make use of Raytheon money and outside help to catch up with SDS, DEC and Three C's.

new faces of 1964 at univac

Some of the most significant moves in the recent Univac management musical chairs game: removal from the line of manufacturing and engineering vp Maury Horrell (former Bendix Computer boss) and marketing vp Retterer to an advisory board to president Forster. New key man: Fred Raach, who heads up the dp div., and who has brought in hot-shot government salesman Lee Johnson to take over marketing.

rumors and raw random data

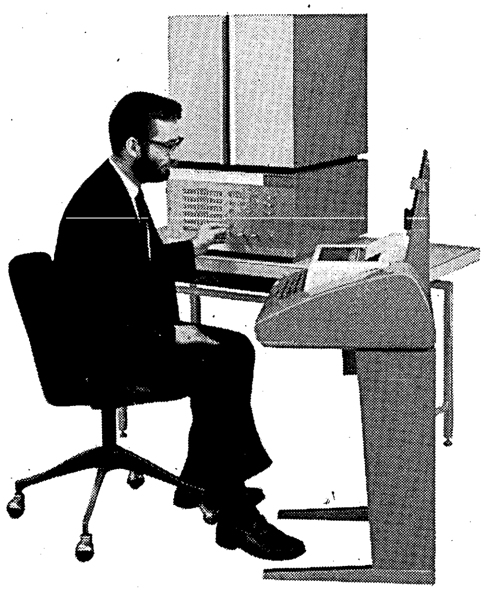
At least three bay area firms are moving up from 1401's to single-channel 1410's with 7330 tape drives allegedly as a transition to the 360. . . . One 1401 installation manager says it takes 33% of the original programming time to maintain his programs. . . . Lawrence Radiation Lab's U. of Calif. installation will replace its 7094's with either a 360/70 or a 6600 (which will later be upgraded to a 92 or 6800), plans to go the time-sharing route. . . . IBM has cut the extra shift rental cost of the 360 to 10% of basic rentals. An answer to GE's "one-class fare" plan? . . . There's a move afoot to establish a special-interest group for time-sharing in the ACM. . . . Burroughs seems to have no problem upgrading its 5000's to 5500's in the field. One changeover was made over a weekend. . . . California Research Corp. has retired its Datatron 204 after eight years' service. . . . Data Products Corp. plans to offer a disc file with one price, regardless of order quantities. . . . RCA's new family will be 360 programming-compatible. . . . Former SDC president M. O. Kappler has left CSC. . . . Finally, a COBOL you can swallow: a new drink developed by a dedicated crew of researchers at the Valhalla, in Sausalito: **C**ointreau, **O**range Juice, **B**randy **O**ver ice with a twist of **L**emon. Try your own proportions.

ANNOUNCING

TWO NEW COMPUTERS

WITH FLIP CHIP™ INTEGRATED CIRCUITS

PDP-8



- 1.6 μsec cycle time
- 3.2 μsec add time
- 12 bit word length
- 4096 word memory
  (expandable to 32,768 words)
- Fortran
- Field tested software
- 625,000 words/sec maximum
  transfer rate
- Reader punch and typewriter
  included

$18,000

PDP-7

- 1.75 μsec cycle time
- 3.50 μsec add time
- 18 bit word length
- 4096 word memory
  (8,192 words directly addressable,
  expandable to 32,768 words)
- Field tested software
  including Fortran II
- 570,000 words/sec
  maximum transfer rate
- 300 cps paper tape reader
- 63.3 cps paper tape punch

$45,000

OPTIONAL EQUIPMENT AVAILABLE

IBM compatible tape
DEC tape
CRT displays with light pen
Card equipment
Analog I/O
Data communication equipment
Bulk storage drums
Printers and plotters

digital
EQUIPMENT
CORPORATION
MAYNARD, MASSACHUSETTS

Washington, D.C. • Parsippany, N.J. • Los Angeles • Palo Alto • Chicago • Ann Arbor Pittsburgh • Huntsville • Orlando • Carleton Place, Ont. • Reading, England Munich, W. Germany • Sydney, Australia

FLIP CHIP IS A TRADE MARK OF DIGITAL EQUIPMENT CORPORATION

CIRCLE 16 ON READER CARD

washington report

DATAMATION® report

buyer wanted for specialty purchase

"For sale: One Communications Systems Division. Good condition. Will listen to any reasonable offer. Call PL 2-6000, New York City." This, in effect, is the offer that's been making the rounds of the nation's corporate and financial circles. The would-be seller is International Telephone & Telegraph, the giant Paramus, N.J., concern which has made a considerable penetration of the government market for communications hardware. Its Communications Systems Div., centered in Washington, does much work in network analysis, design for the Defense Communications Agency, raising for ITT the spectre of conflict of interest by being both advisor and supplier to DCA. Hence, CSD became expendable. Last word was that Computer Sciences Corp. of El Segundo, Calif., a computer services company which is hopeful of making a bigger splash in the government market, was close to a deal with ITT.

bubudget's computer brokerage service grows

BuBudget, apparently encouraged by results of its pilot programs in Philadelphia and Washington, will soon extend its computer exchange operation to 12 more cities. Denver will be first, followed by Atlanta, Boston, Chicago, Dallas-Ft. Worth, Kansas City, Los Angeles, New York, San Francisco, Seattle and St. Louis. Brokerage activities in these cities will be under the supervision of the GSA, indicating that BOB believes its computer exchange plan has proven itself. Under the program, a clearing house will be established in each city, through which federal agencies with excess computer time will be matched up with agencies which need computer time. The theory: through this exchange greater mileage can be gotten out of government-operated computers. Unlike the program now under way in Washington at the Bureau of Standards, no federal service center will be initiated in these cities.

brooks bill hearings back on the slate

The Senate Committee on Government Operations will conduct hearings early next year on the Brooks Bill, HR 5171, Sen. McClellan, committee chairman, recently announced. "It's the hope and expectation of the committee that early in the next session of Congress the bill will be reintroduced just as it passed the House, or as proposed to be amended by the Bureau of the Budget, and obtain as speedy action on it as possible," he said. Sen. Paul Douglas of Illinois, a champion of ADP economy, said he was "very glad the senator from Arkansas has now reassured us that it (the delay in scheduling hearings) does not mean defeat for the measure but merely postponement."

The Brooks Bill, which would delegate sweeping powers over federal computer policies and practices to the General Services Administration, has been a bone in many throats ever since its passage by the House in July 1963. Many have criticized the bill's language as being unclear

Continued on page 91

EDITOR'S READOUT

THE NEW WAVE?

It's slowly beginning to dawn on some people that the truly significant part of the 360 announcement last April was not the hardware, but a new programming language (called, with seeming immodesty, "New Programming Language"), which T. J. Watson, Jr. didn't even know about at announcement time.

As time marches on, however, it becomes increasingly apparent that NPL is one of the biggest guns in the IBM sales arsenal. Salesmen are pushing the one-language concept hard, if not always with equal expertise (one sales target says the IBM salesman walks in, writes NPL on the blackboard, and waits for him to swoon).

The sales effort reflects a tremendously important investment – some call it a gamble – on the part of IBM, which is putting plenty of bucks and even more prestige on the line. IBM is betting that business data processing and scientific computing are not-so-gradually merging, that a one-language shop will be an unmixed blessing . . . and that they can produce the software which will make these twin dreams come true.

The implications of this new language are huge. Partly because of the muscle behind it – anything No. 1 does is bound to make broad ripples – partly because of the promises a one-language shop offers: simplified training and documentation, the availability of one operating system, no more Babylon.

These promises may more than offset the complaints about the language, which range from the manner in which it was developed ("language by decree") to its lack of a strong theoretical foundation. Some believe that it will be difficult or impossible to develop an efficient NPL compiler. Others feel that, in trying to be all things to all people, it suffers . . . offers features not all can use, but for which all must pay.

These criticisms may be valid. But the weight of the force behind the new language almost guarantees its acceptance, in some form or other. Added to this influence is the absence of any strong counter-candidacy from COBOL. The triple victim of its own miserable heritage, lack of support, and an archaic committee system, COBOL looks to be less and less likely a candidate for anything. Nature may abhor a vacuum, but NPL will certainly not.

What we'd like to suggest is that it is not too late to influence the final form which this new language will take. Initially, it will be molded by IBM, certainly. But the user will have the final word. And competitive manufacturers – who must decide whether or not to add it to their language sampler – can now begin the studies which will help uncover any weaknesses in the language, any features which favor or ignore characteristics of particular hardware systems, or forget the needs of any applications. They might even be clever enough to improve NPL . . . and end up out-doing IBM.

We hope that users, competitors, user groups, standards bodies are all diligently studying the language. If it does indeed represent the new wave of programming languages, it is especially important that its refinement and standardization move ahead crisply, so that the language and its compilers can provide the man with the machine on his back – any machine – the best possible means of solving his already staggering problems.

INTRODUCTION TO TIME-SHARING*

by E. L. GLASER and F. J. CORBATO

During the past year a great deal of attention has been paid in the press and in various technical journals to the subject of on-line real-time systems, particularly those involving time-sharing systems of a general purpose character. By time-sharing systems we mean those systems in which the facilities of a computer complex are rapidly commutated among independent users who are each on-line at a remote console. Time-sharing of a central computer is not the sudden birth of a new idea which has come full blown from the head of the computer world Zeus, but rather, is a restatement of an idea that is relatively old in our field. It is interesting to note, for historic value if nothing else, that the early relay computers of the Bell Telephone Laboratories were capable of being operated by several different users from a distance. Of course, only one user could operate the computer at a time, but even in the early 1940's it was considered useful to have remote access to these computers as a problem solving convenience.

It is our purpose here to introduce the elementary features and characteristics which can be observed in contemporary operating time-sharing multiple-console systems. It is not our purpose to detail here all of the advantages of time-sharing inasmuch as several points of view are given among the references. Briefly, though, we are assuming that time-sharing when commercially available will be the mode in which most computers are used. This assumption of time-sharing is based in part on the economic savings of more efficient utilization of equipment, in part of the increased flexibility and convenience to computer users, and in part on evidence based on over a year of pilot time-shared operation of an IBM 7094 computer at M.I.T. that batch-processing can largely be replaced.

A major reason we believe most computers of the future will be time-shared is that current batch processing capabilities are a subset of any complete time-sharing system. In particular a user of a well-organized system should be able to either submit at the central computer a card deck prepared off-line or have the program typed in by himself or others at a remote on-line console. Similarly, although all jobs might be initiated at a console, the user could elect,

Prof. Corbato is with the Dept. of Electrical Engineering at MIT and deputy director of its Computation Center. He was formerly an assistant director at the center in charge of programming research, his principal professional interest being the techniques of computer usage. He has authored several papers and books in this area, is presently active in the development of time-sharing systems. He holds a BS from Caltech and a PhD from MIT.

Mr. Glaser is a research associate in the Dept. of Electrical Engineering at MIT, and holder of many patents in the field of digital computers. He was formerly manager of the Systems Dept. of Burroughs Research Div., and participated in the design of large-scale computers for IBM. A senior member of IEEE, he is a member of its Cybernetics Committee. He holds a BS in physics from Dartmouth College.

*Work reported herein was supported by Project MAC, and M.I.T. research program sponsored by the Advanced Research Projects Agency, Department of Defense, under Office of Naval Research Contract Number Nonr-4102(10). Reproduction in whole or in part is permitted for any purpose of the United States Government.

perhaps on the basis of job size, whether or not to await job completion. Moreover, the system should be such that the user has a choice of either immediate selective printing at his remote console from his program output file or of waiting a few hours for standard bulk output printed at the central facility by a high-speed line printer. Even magnetic tape units at the central facility can be made straightforwardly usable through remote consoles for those data processing applications which require vast linear files of information. The user of a time-shared system can, if he chooses, continue to do the equivalent of batch processing exclusively. However, our present experience indicates that most users will take advantage of the broader spectrum of service offered by this more flexible system. Finally since the cost to each user of a time-sharing system should be arranged to be the rental of just the processors, memory, etc. which each user is utilizing—not the entire computing complex as is of necessity the current practice with batch processing systems—it should be clear that arguments over cost are not a substantive issue when discussing time-sharing.

nature of a computer

To understand better some of the motivations behind time-sharing systems, the nature of computer use should be examined. Individuals often are not "solving problems" but rather, defining problems. In the conventional method of operating computation centers, the main use that the computer can be put to is solving a problem to which the user already knows the answer, in the sense that the method of problem solution is already well understood, and the computer is primarily used to determine specific numeric answers. Obviously there are exceptions to this broad generality; however, most of these exceptions come in the form of playing a very expensive question-and-answer game. In order to use a digital computer to help in the trial-and-error formulation of problems and the investigation of radically new areas, it is necessary for the man and the machine to be quite tightly coupled. Further, it is highly desirable that the person doing such work be able to have access to a computer at the time he needs it. In other words, the man wants to be able to use the machine with the same degree of facility that he now uses other immediate tools of intellectual activity, such as: blackboard, books, notepads, desk, etc. This type of activity is quite familiar to those who have had any dealings with the analog computer art. It is, however, at present unusual in the field of digital computers. Because of the broad range and scope of problems to which the digital computer can be applied, it is of interest to expand this interactive capability as rapidly as possible.

It is beyond the scope of this article to discuss in detail all of the necessary criteria to be met by a time-sharing system and, equivalently, to describe all the myriad factors that must be taken into account in order to have a functioning system. It is the intent to describe, in broad philosophical terms, a time-sharing information processing utility and indicate some methods by which such a utility can be brought into existence. In most cases, the methods described here are those initially developed at the M.I.T. Computation Center and which form the current operating system at Project MAC. The methods are not necessarily optimal and are not necessarily those we would use again. The virtue of these methods, however, is that they are not hypothetical and the current MAC system is a working example. Now that we have, to a small degree, described the motivation for at least our interest in information processing utilities, and further, have included the necessary escape clauses and qualifications, we can press on.

memory protection

In most fields of endeavor a very small change in a basic premise can affect, radically, the nature and method of system organization. Thus, in the field of computation, the concept of many programs operating in a computer with rapid switching between them, and further, that these programs are written by many different individuals at different times, requires a fundamental change in the method of operation of an information processing system. In particular it is necessary to protect the user programs from one another in order to preserve their independence. Of course, a program is never really debugged. What one means when one calls a program debugged is that the frequency of faults has dropped to a point that none has been observed in a reasonably long time. As a practical matter system programs must be treated as debugged and user programs as never debugged. We now see the necessity of there being hardware to guarantee that a user's program can never go outside those bounds established for it by the executive program. This is, in our opinion, one of the prime requirements of any hardware system into which we wish to place a time-sharing executive program. It should be observed that within a system in which many user programs are being swapped in and out of memory at a relatively high rate, it is often difficult or impossible, without such protective hardware, to determine the difference between a program that has run awry and a machine fault.

This protection hardware can take several different forms; that employed in the M.I.T. Computation Center and Project MAC IBM 7094's is a relocation register in conjunction with upper and lower bound limit registers. The lower and upper bound registers define regions in memory to which a user's program has access. Further, the relocation register serves the additional function of program relocatability. All programs are written as though they were in a memory of their own and starting at a location zero. Of necessity, when any form of protection hardware is employed, there must be two "modes" of operation, a user mode and an executive mode. In the user mode the protection hardware is effective and it is impossible to modify the protection state of the machine. It is also impossible to change the contents or even have access to the contents of the relocation and bound registers. Further, in this user mode it is often desirable that certain types of instructions that would in other ways interfere with the function of the system as a utility be prohibited. Such instructions as halt, input/output, etc., fall into this category. Any attempt to violate these restrictions or to arbitrarily enter the executive mode of the system, should cause an interrupt or trap into the executive program.

Another requirement of a time-sharing environment is the necessity for a combination of hardware and programming system that will guarantee each user's program access to the computer on some type of an equitable and convenient basis. The simplest form of such access is to allot each user in the system a certain fixed amount of time, at the end of which a program is automatically interrupted and the next program in turn is resumed. Thus, each user's program is like one point on a commutator. More complex scheduling algorithms than this may be used but are not an essential element. The variations of scheduling algorithms are themselves a very interesting topic, and when time-sharing is more common, could serve as the topic for an article in its own right. In any case, even the simplest scheduling algorithm requires an interval timer that can be set by the executive program which upon run-out causes a trap or interrupt. This feature is now quite common with many modern computers.

desired characteristics

Up to this point we've discussed those topics in the hardware and the software that are required as a framework for a time-sharing system. They represent to the time-shared computing system what the basic power supply and clock represent to the simple computer hardware. The system will not function without these elements; they form the bare minimum of mechanism required.

Before discussing the remainder of the elements of a time-sharing system, let us briefly examine what this system is to be. The best way to determine its desired characteristics is to look at the system from the standpoint of a user sitting at his console. To merely say that we will make a large-scale computer available to each user as though it was his own, is not sufficient. It is certain that the reader would be appalled, as are the authors, at the concept of having a personal large-scale binary computer without system programs available and with only typewriter access. Certainly more than this bare framework is requisite. A whole rationale of higher level operations must be built-in to the system so that the user has available, to him, very comprehensive "system commands" with which he can control the system from his console. System commands are required to do such simple things as: allow input to take place, that is to set the machine in such condition that it can receive information from a keyboard; create a file; edit a file; cause the contents of a file to be compiled; load and start the execution of a user's program; etc. It is even necessary for there to be a command for logging into the system to identify authorized users uniquely and associate them with their stored files. These various system commands are, in fact, small subroutines that are called either from the user's program or, more often, by specific instructions from the user's console. Thus, we can consider that there is a dormant state of the system with respect to each user. When this dormant state is in effect, with respect to a specific user, the system is constantly looking for some form of input from the user's console. During the user's time in the allocation schedule, if his program is in this dormant state and no input has been entered from his keyboard, no actual computer time is taken. In any case, the system should at all times be able to accept input signals from any user regardless of his position in the allocation schedule. The particular mechanization chosen is a function of the equipment to be employed in the time-sharing system.

In addition to the system command programs and the other basic control programs that have been described already, there is a set of programs required for what might be called housekeeping or administrative functions. Although these programs do not enter directly into the functioning of time-sharing, its smooth operation would be impossible without them. These include several categories of programs, such as system monitors, disc file updating programs and the like. Two functions that should be explicitly pointed out that fall in this category are those that authorize persons to use the system and those that allow file purging and maintenance. The first of these is a relatively obvious function and it merely requires that there be a specific file which is accessible only to one in authority. This file gives the name, problem number, resource allocations and other pertinent information for each person who is authorized to use the system.

file maintenance

The second housekeeping function, that of file maintenance has to do with the use of a large drum or, more commonly, a disc file which is generally associated with a time-sharing system. Although it is possible, under some limited circumstances, to use magnetic tape for the secondary storage of the system, this does not appear to be practical. The user's programs and data are kept on a disc file in the case of Project MAC. All files which are created are stored on the disc and subsequently accessed by name only and not by location. It would be wasteful for every user to have a fixed amount of storage assigned to him from the disc. Instead only a storage quota is assigned to him, with actual storage being allocated only as required. As a consequence, it is desirable to have a mechanism whereby it is possible to assign non-contiguous storage areas to the same user, and this is accomplished by means of list structuring each file. The files that belong to an individual user are then identified through a specific file which is his file directory. It is periodically desirable to reorganize files so that unnecessary motion of the disc access arm is eliminated. Further, it appears to be wise to take periodic dumps onto magnetic tape from the disc file so that the system can be reinitialized to a previous state in the event of a major malfunction. In addition, it may be desirable to remove stale files by placing them on a cheaper but more inaccessible storage medium.

Obviously, the classic view of a program run in a computer is no longer valid within this environment. Each user's program is nothing but one element of a system of programs whose exact size can vary rapidly in time. Efficient hardware utilization is no longer the prime measure of effectiveness. The service given to each user is the criterion that must be paramount. It is important to realize that the majority of the programs in the system are user programs and that they are inherently separate and independent. Moreover, if we are to maintain rapid response to the stimuli of each user, it is essential to be able to switch easily between user programs which are by design totally decoupled. Models of time-sharing systems have been designed and run in which the entire executive program and all its subsidiary programs have been structured as a single monolithic unit. This however is far from flexible and prevents evolutionary development of the system. It is desirable to segment the executive system to the point that each segment or module of the system is relatively decoupled from others, except at very rigorously defined and controlled interfaces. There are two reasons for this: first, it makes it possible to easily modify the system and update various elements as newer views of system organization gain sway and greater importance; second, it means that a fault in one part of the program is limited in its ability to propagate into the remainder of the system.

It is because of this rather loosely coupled nature of a large time-sharing system, that it is difficult to describe any one action that should be taken on the part of machine designers or equipment selection committees that will best facilitate an efficient system. A few dos and don'ts, however, can be observed. First of all, the machine must have those basic hardware protection features in one form or another as described earlier in this article. Second, the necessary compatible communications interface equipment and terminals must be available. It is suggested that terminals of the type used in communications are quite suitable here since, in point of fact, the user is carrying on a limited dialogue with the information processor and those facilities that have heretofore been useful in keyboard start-stop telegraphy are also useful here.

Third, the necessary complement of secondary and tertiary stores should be available. These include high-speed drums, disc files, and magnetic tapes. It may be that some of the new devices, such as magnetic strip memory systems, will have a place in future efforts. It should be mentioned that, since programs enter and leave the primary storage, the feature of easy relocatability of programs is highly useful. It means that a program can be placed anywhere in the core memory where sufficient room can be found for it, and it can be executed from that position whether or not it was previously in this position. Since its position change can take place at literally any time during its execution, this relocatability must be accomplished by means of hardware, as the attendant software conventions to make it feasible entail an inordinately large overhead burden.

Fourth, although multiprocessing has not yet been used in time-sharing, the availability of more than one processing unit that can access any part of memory in a system makes it possible to assign the processing resource in a highly flexible manner. In small systems this may be of little interest; however, on large systems, this can become of importance as it is one way of reducing the response time and thereby increasing the utility of the system. Further, it means that a high degree of system availability can be obtained without requiring astronomically high reliability on any one component in the system.

Fifth, the hardware should permit the type of loosely coupled programming system that has been briefly described in the preceding paragraphs. What is required is a flexible and highly generalized interrupt system and an I/O system that permits direct access to any I/O device without concern for the particular I/O channel that may be used. In the case of multiple processor systems, this intercommunication between I/O devices, memory units, and processors is critical, and can make the difference between a smoothly functioning system and one that barely works.

end to batch processing?

We are seeing the beginnings of what can be a radical change in the use of computers in the next several years. Time-sharing should make is possible for a single computing system to be made available to a number of independent users either within an organization or, in the not too distant future, from a number of totally different and possibly competing organizations. Further, it makes it feasible for the computer communication system complex to gather information where it is generated and to deliver results where they are needed. In such a system it is doubtful that many 60,000 line-per-minute printers will be used, although large numbers of 100 line a-minute printers will be needed, since the printing of a page takes place after the information has been routed to its proper destination. Further, it is doubtful that many users of a computer will really require the heavy fan-fold output that they now feel necessary for reassurance but rarely use. Instead, one would go to the keyboard and ask sophisticated questions about anv particular element in a file which is accessed by a central computer. This is not to say, however, that all computers will go predominantly time-sharing. On the contrary, there will still be batch processing for those problems not requiring intimate man-machine interaction; but even this, as was pointed out earlier, can be in conjunction with time-sharing.

It is difficult to draw any conclusions at this time since time-sharing is still in the advanced development stage. A few comments, however, are in order. The selection of proper hardware will have some effect on a time-sharing system although the nature of the programming system is of even greater import. Any programming group that is thoroughly competent in the field of system programming can certainly develop a time-sharing system. However, the organization that is only casually familiar with system programming had best wait till more of the problems are understood and techniques are more generally known. This should come about within the next few years. In any event, time-sharing is here. It is not a revolution; rather it is a more general method of making information processors available to the user. ■

REFERENCES

1. Strachey, C., "Time Sharing in Large Fast Computers," *Proceedings of the International Conference on Information Processing*, UNESCO (June, 1959), Paper BB. 2, 19.
2. Licklider, J. C. R., "Man-Computer Symbosis," *IRE Transactions on Human Factors in Electronics*, HFE-1, No. 1 (March, 1960), 4-11.
3. McCarthy, J., Lecture given spring 1961, *Management and the Computer of the Future*, M.I.T. Press (1962).
4. Corbato, F. J., "An Experimental Time-Sharing System," *Proceedings of the Spring Joint Computer Conference, 1962*, Spartan Books, Inc., Baltimore, Maryland.
5. Corbato, F. J., "The Compatible Time-Sharing System: A Programmer's Guide," M.I.T. Press, (1963).
6. Schwartz, J. I., Coffman, E. G., Weissman, Clark "A General-Purpose Time-Sharing System," *Proceedings of the Spring Joint Computer Conference 1964*, Spartan Books, Inc., Baltimore Maryland.
7. Fano, R. F., "The MAC System: A Progress Report," A paper presented at the Symposium on Computer Augmentation of Human Reasoning, Washington, D.C., June 16, 1964. Proceedings to be published by the Office of Naval Research.

THE SDC TIME-SHARING SYSTEM

facilities, services & potential

by JULES I. SCHWARTZ

In spite of all the power available in present day computers, a feeling of uneasiness, even disappointment seems to prevail among those concerned with paying for and using computers. These feelings stem primarily from the relative inaccessibility and inflexibility of the computer. The ability of users to produce computer systems that can be readily modified is frequently much less than supposed, and the ability of computers to afford quick and accurate solutions to a variety of even simple problems is surprisingly limited. Part of the reason for this dilemma is the formidable wall between users and computers in most installations. This wall is faced not only by managers and customers, but also by programmers.

One of the major reasons for the wall is the persistence of traditional techniques for applying computers. These techniques require that programs be prepared (in a language that is not usually oriented to application), be sent to the computer (where they are processed and executed with no intervention by the user), and, when processed, be sent back hours (or days) later. Aside from the fact that this approach can lead to numerous delays and wasted computer runs, there are many applications for which this detached operation is completely unsatisfactory. Thus the concept of continuous interaction by the user with his computer (*on-line* computer operation) promises to become an essential part of current techniques in computer application.

On-line use of a computer is not a new concept, for it has appeared in a variety of situations in the past; but these applications have been special purpose (e.g., SAGE) or quite limited (e.g., use of small computers such as the Bendix G-15 or the Digital Equipment Corporation's PDP-1). Complete user interaction may be available on small computers, but because of their properties, many desirable features of large on-line systems cannot be made available, such as access to many general- and special-purpose languages. Therefore, it seems, use of a large computer on-line would be desirable. Unfortunately, such use would introduce certain inefficiencies, without major changes in technique. For one person to preempt the total capacity of a large machine for long periods of time would be highly uneconomical if he could not keep the computer occupied all the time it was assigned to him. Time-sharing permits on-line use of the computer simultaneously by a large number of people by giving each user time when he requires it. This kind of system provides a direct and continuous working relationship between users and computer and keeps the computer busy most of the time by limiting the amount of idle time due to human thought or output from on-line devices.

In the Command Research Laboratory at SDC, a large percentage of the problems run are of the on-line, man-machine, interactive variety. Since these applications are virtually impossible to run in a serial fashion (one-at-a-time), the requirement to produce a system that would permit parallel running was a necessity. For this, and for the sake of further study of the time-sharing process itself, a time-sharing system was developed as the main program vehicle with which to run the laboratory.

characteristics of time-sharing systems

Given the general requirements for time-sharing systems, it may now be worthwhile to examine the properties (and hence the definition) of such systems. Four characteristics of time-sharing systems encompass most of their distinctive features; such a system is:

- Simultaneous—A number of people use the computer at the same time.
- Instantaneous—All users receive responses from their

Mr. Schwartz is head of the time-sharing project at System Development Corp., Santa Monica, Calif., with whom he has been associated since prior to its spin-off from RAND. His software development projects have included PACT, the Lincoln utility system, and JOVIAL (Jules' Own Version of the International Algebraic Language). He holds a BS in math from Rutgers and an MA in mathematical statistics from Columbia Univ.

programs or the system within seconds—or fractions of a second—of the completed computation.

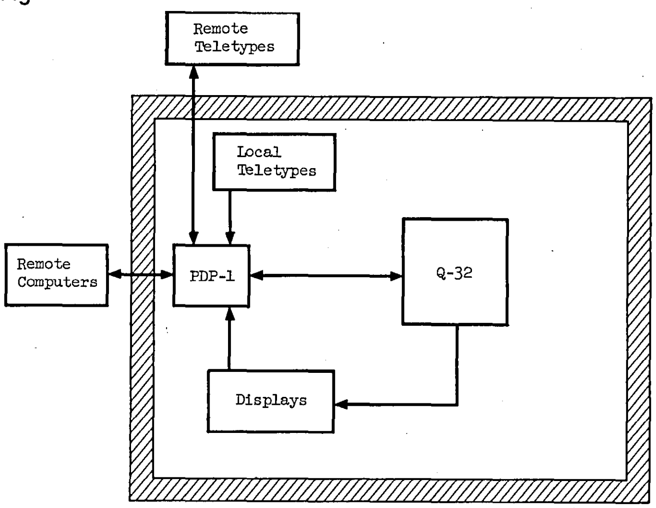
- Independent—Different programs, services, and devices can be in use separately or in combination during any given period of time.
- General-purpose—No restriction is placed on the kind of program or the application under the time-sharing system.

Various on-line systems have some, but not all of the properties just described. Others, such as the SDC system, do have all of these characteristics.

equipment configuration of the SDC system

Since on-line operation of a computer requires the use of input-output devices in addition to tapes, card-readers, printers, etc., on-line devices available to users of the system will be reviewed first. These devices include teletypes, typewriters, and cathode-ray tube (CRT) display

Fig. 2



the AN/FSQ-32 in Santa Monica and a CDC 160-A computer at the Stanford Research Institute, Palo Alto, Calif.

The basic characteristics of the time-shared AN/FSQ-32 system are listed in Table I. Assisting the Q-32 under time-sharing is the PDP-1, which serves as the major interface between the on-line devices and the Q-32. A simplified diagram of the complex is shown in Fig. 2. The local Q-32 configuration is shown in Fig. 3.

Fig. 3

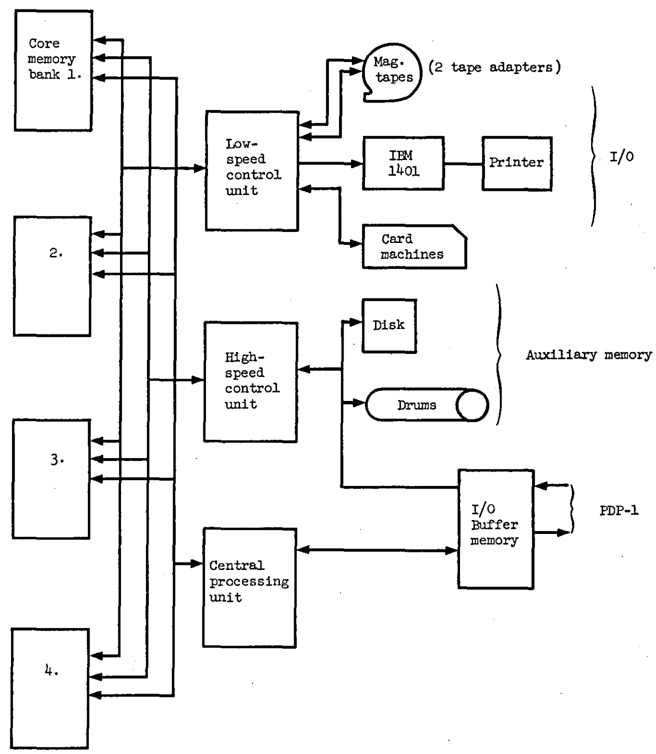


Fig. 1 Characteristics of the AN/FSQ-32 Storage Devices

DEVICE	SIZE	WORD RATE	AVERAGE ACCESS TIME
Core Memory	65K	2.5 usec/wd.	—
Input/Output Core Memory	16K	2.5 usec/wd.	—
Magnetic Drums	400K	2.74 usec/wd.	10 msec
Disc File	4000K	11.75 usec/wd.	225 msec
Magnetic Tapes	16 Drives	128 usec/wd. (High density)	5 to 30 msec (no positioning), depending on whether the tape is at load point, and whether it is being read or written.

consoles. The teletypes have been installed in various locations within SDC and are also used from numerous remote locations, as far away as Pittsburgh, Pa. Within SDC, there are eight model 28 Teletypes, 16 model 33's and three Soroban typewriters. There also exists capacity for eight remote teletype stations operating simultaneously. In addition to three types of keyboard devices, there are six display consoles currently available. These are all located within the Command Research Laboratory.

The SDC time-sharing system may also be used remotely via a 2kc line, the terminal to which may be a computer. That computer may in turn service displays and keyboards. This linkage permits the remote computer to communicate with a program in the SDC computer, thus permitting computer-computer-human interaction under time-sharing. Currently one such line is operating between

the time-sharing system

To understand the services available to a user of the time-sharing system, a review of the system itself would be useful. A simplified description of the system is as follows:

All programs requested by users are stored on an assigned section of the drum until the user quits.

A program is given one "quantum" of time to operate, which represents the maximum time for one operating cycle. The program's turn ends when it requests an input-output transfer, when the quantum of time ends, or when an error condition arises. When the turn ends, the executive system determines if another program is ready to operate. If not, the program will reside in core memory until another program is ready. When another program is ready to operate (and it occupies an area of core which overlaps that of the first), the executive writes the first back to its place on the drum, and brings in the new program to core.

The process of exchanging programs from core to drum is called *swapping*.

The basic cycle of an execution followed by a swap is fundamental to the time-sharing technique of operation. Controlling this process is a program called the *executive system*.

The executive system resides permanently in 16,000 registers of core. It has the following major functions:

- Responds to all computer interrupts and takes the required action.
- Interprets inputs and commands from teletypes and typewriters.
- Allocates both core and peripheral storage.
- Performs all input-output.
- Schedules the running of object programs (as in the above example).
- Performs a number of on-line debugging services.

Object programs, which can contain up to approximately 47,000 words of storage, are placed in core storage only when they are active (ready to compute). When they are not active (stopped or waiting for input or output), they reside on the drums, placed there by the executive. As shown in Figs. 4 and 5, they are put on drums by the system when given the LOAD command (see below - Basic Commands) via teletype.

The Fix Program, which occupies approximately 2,000 words of permanent storage, reacts to all computer errors, logs the error on a typewriter at the maintenance console, attempts to fix the error, then returns to the executive system with a description of the error condition. The system then isolates the user(s) affected by the error, notifies the affected user(s), concludes the affected program(s), and then continues the normal cycle.

A major function of the executive system is to interpret and respond to the various teletyped commands made by a user. It is essential that most users know these basic commands. They are used to sign in, load, and run object programs, stop programs, and sign out. These commands are:

- LOGIN: The user begins a run. With this command he gives his identification and a "job number."
- LOAD: The user requests a program to be loaded (from either tape or disc). Once this command is executed, the program is an object program to the system.
- GO: The user starts the operation of an object program or restarts the operation of an object program that has been stopped. Once the user gives this command, he can send teletype messages to his object program or the time-sharing system.
- STOP: The user stops the operation of an object program.
- QUIT: The user finishes a particular job. Upon receipt of the command QUIT, the time-sharing system punches accounting information into a card and removes the object program from the system.

Figs. 4 and 5 are examples of this sequence as it would appear on a user's teletype. It will be noted that to each command from a user there is a response on the teletype from the system. Until this response is received, the user cannot assume that the system has reacted to the command. Normally, this response is immediate.

In the brief description of the LOAD command above, it was stated that the requested object program is loaded from tape or from disc. It is assumed, of course, that the program exists in binary form—expected by the executive to perform the LOAD function. There are a number of ways in which a program can be put into this form. Some of the more common will be explained in the section on service routines.

The sequence of actions taken after the LOAD command is given is as follows:

The Executive examines the name of the program, which follows immediately after the command LOAD. If it finds that this program has been stored previously on the disc, it brings it in from the disc, places it on the drum, and responds with $LOAD OK as in Fig. 4. If the program is not on the disc, it types out instructions at the computer console for the computer operators to load this program, or the specified tape reel (optional) for this teletype. It also types $WAIT on the user's teletype. When this operation is completed by the operator, the program is stored on drum and disc (for subsequent loads), and the response $LOAD OK is typed on the teletype. The time between the $WAIT and $LOAD OK can be a matter of minutes.

Fig. 4. Load From Disc

```
LOGIN 0050 JCX.25
 $OK LOG ON 10
LOAD PRG2                   Object program on disc
$LOAD OK                    Object program on drum and disc
GO
 $MSG IN.
!STOP
 $MSG IN.
QUIT
 $MSG IN.
```

Fig. 5. Loading a Program From Tape

```
LOGIN 1123 JBX.30
 $OK LOG ON 10
LOAD TPRD 1852
$WAIT.                      Object program on tape
$LOAD OK                    Object program on drum and disc
GO
 $MSG IN.
!STOP
$MSG IN.
QUIT
 $MSG IN.
```

Fig. 6. User Talking to Object Program

```
LOGIN 0050 JDX.25           User to executive
 $OK LOG ON 10              Executive response
LOAD CALC                   User to executive
$WAIT.                      Executive response
$LOAD OK                    Executive response
GO                          User to executive
 $MSG IN.                   Executive response
 CALC READY                 Object program response
+ 2,2                       User to program
 RESULT +4.00000000         Program response
* 3,3,3                     User to program
 RESULT +27.0000000         Program response
!STOP                       User to executive
 $MSG IN.                   Executive response
QUIT                        User to executive
 $MSG IN.                   Executive response
```

"talking" on a teletype

It is quite clear that once the GO command is issued, it is necessary for the user to communicate over the teletype with the object program. Therefore, until otherwise notified, the executive assumes that any input on the teletype is to the object program, not the executive, and therefore simply passes the input to the program, which reacts in some way, probably with an output on the teletype. Fig. 6 shows a typical sequence of steps taken when a user communicates with an object program.

There usually comes a time when the user must start talking to the executive after he has communicated with his object program—to give the STOP command, for example. The user signals this intent by typing an exclamation point. This action and its consequences are demonstrated in Fig. 6. Correspondingly, it is frequently necessary to recommence talking to the object program, having once used the exclamation point. This is done by typing quotation marks, as shown in Fig. 7.

Fig. 7. User Talking to Both Program and Executive

LOGIN 0050 JDX.25	
$OK LOG ON 10	
LOAD CALC	
$LOAD OK	
GO	
$MSG IN.	
CALC READY	Program to user
+2,3,4	User to program
RESULT +9.00000000	Program to user
*4,4	
RESULT +16.0000000	
!DIAL 28 HELLO	User to executive
$MSG IN.	
"*3,3,3	User to program
RESULT +27.0000000	
!STOP	User to executive
$MSG IN.	
GO	
$MSG IN.	
—4,2	User to program
RESULT +2.00000000	
!STOP	User to executive
$MSG IN.	
QUIT	User to executive
$MSG IN.	

other executive services

In addition to responses to basic commands, the executive performs a number of other services for users. Included among these is a set of functions necessary to check out (debug) object programs on-line. The kinds of debugging commands available through the executive include the following:

- *Open*: Displays the contents of the given memory or machine register and uses this as a base address for other debugging commands.
- *Modify Open-Register Address*: Changes the address of the opened register by the given increment or decrement.
- *Insert*: Inserts the given values into the opened register.
- *Mask*: Inserts values by the given mask.
- *Mode*: Displays values according to specified mode (floating, decimal, octal, Hollerith, symbolic machine language).
- *Search*: Finds the register containing a specified value.
- *Breakpoint*: When a specified point in the program is reached, notifies the user, (on options) displays registers, and stops or continues the program. As many as five breakpoints are allowed simultaneously.
- *Dump:* Dumps a given set of registers, either on teletype or tape (to be printed off-line).

The actual commands to perform these functions usually include a symbol or address with one or two unique teletype characters.

Figs. 8 and 9 show examples of on-line debugging operations.

Fig. 8. Examining and Changing Words in Octal and Floating Point

40000'/	DBUG command
40000' = 0140000000040052'	DBUG response
DATAWORD/	DBUG command
DATAWORD = +4.56289757E+002	DBUG response
2.5 E-3 *	DBUG command
$IN	DBUG response

Fig. 9. Examining and Changing Words in Hollerith and Integer

MESGWORD (5) /	DBUG command
MESGWORD = CONTINUE	DBUG response
BETA/	DBUG command
BETA = —197	DBUG response
—495 *	DBUG command
$IN	DBUG response

Several commands are available to enable users to communicate with each other. These commands and their functions include:

- DIAL: Permits sending messages to any other user in the system.
- LINK: Initiates linkage of any two teletypes so that they act as one. Both teletypes can input and output to and from the same program or the systems. Also, both teletypes can type all the information being input or output on each other.

Commands to query the system about its status are available through the executive. Examples of these are:

- USERS: The response to this command is the number of currently active users in the system.
- TAPES: The response to this command is the number of tapes available for use by object programs.
- DRUMS: The response to this command is the amount of drum space available for object programs.

Quite clearly, object programs must have access to the on-line input-output devices that are part of the system. In addition to these, the system at SDC permits access to disc files and tapes as completely as is possible with the computer. Since a "traffic" problem exists with these devices (as well as with the on-line devices), all input-output units are assigned by the system. No object program makes an absolute (machine) reference to an input-output unit. In the object program, requests for input-output "files" include arbitrary names, for which assignment of a particular unit is required. The executive then assigns an actual (machine) unit or an area of a unit such as disc (if one is available). For subsequent reads, writes, or positionings of the unit, the object program requests the executive to move the file by using the name given in the initial request. Any attempt by an object program (in machine language) to read or write a unit directly will result in a computer interrupt that will stop the program.

A mechanism has been provided to permit the multiple use of one program by several users. An object program may in its input-output file declarations request more than one teletype or display console (in this case, specific units). Thus a single program can act as a recorder, monitor, situation generator, or play some other central role to a group of users in a game or other team effort. *(To be continued).* ■

time-sharing at rand

THE JOSS SYSTEM

An on-line problem-solving system, in which the programmer no longer is the intermediary between the man with the problem and the computer as the answer-box, is presently operating at The RAND Corp., Santa Monica, Calif. Eight remote typewriter consoles are linked to a time-shared JOHNNIAC computer, a small-scale and (by today's standards) slow computer designed in 1950-51 at RAND.

JOSS (Johnniac Open Shop System) was designed by J. C. Shaw to provide a personal computing service tailored specifically to the needs of the scientists and engineers at RAND—something that would enable members of the technical staff to sit at a typewriter (IBM 868 with a modified character set), state their problem, get their answer, and, if needed, try it again with altered sets of conditions or data. In short, Shaw has come up with a combination desk calculator/stored-program computer, but not a system for production computing.

Presently available to staff members—of whom more than 100 can be called qualified users—JOSS is reportedly experiencing its greatest response from engineers and scientists who have tried unsuccessfully to use a digital computer but were turned away by the mechanics of programming and the delays of batch processing. To use the system, however, one must learn the language and know how to type. The latter, alas, has been the biggest stumbling block: the user's facility with the system is in direct proportion to his typing ability. But with hardheaded nontypists, who are counted among the users, it's more a case of "I wish I had learned to type," rather than "I wish JOSS didn't require me to type." The Computer Sciences Dept. does no training of users; people learn by watching others, or trying and letting JOSS tell them their errors.

The software, consisting of some 6,000 (40-bit) words, operates in an interpretive mode. That is, the context in which the user's statement is executed is determined at the time of execution, and can change from execution to execution. Whenever there is a number in the language, a variable or an expression can be substituted.

When JOSS is interpreting steps in the internal mode and detects an error condition, it responds with an appropriate error message. The system then backs off so that the user can correct the offending step or steps, change any of the variables, define new variables, put in a part if one is missing, etc. Then by saying "Go" he asks JOSS to try again from the beginning of the execution of the statement which caused the error.

Other features: the user is completely sealed off from the computer configuration . . . decimal arithmetic throughout, with answers to nine decimal digits . . . high-quality printout (many users incorporate their output directly in publications) . . . JOSS's typeouts are in black, and the user's typeins in green.

What would be gained by having a JOSS system on a faster, larger machine? To get some idea of this, a routine

The designer of the JOSS system is J. C. Shaw of the RAND Corp., a member of the once-prolific writing team of Newell, Simon & Shaw. A mathematician and one-time actuary, Mr. Shaw has been programming with RAND since 1950. In addition to producing symbolic assemblers and interpreters for the 701 and JOHNNIAC, and time-shared programming systems, he has also investigated list processing and heuristic programming with the aforementioned A. Newell and H. A. Simon. Mr. Shaw holds a BA in math from UCLA.

added to RAND's 7090 FORTRAN monitor to print out statistical data on the nature of jobs, amount of core used, time of execution, time of compilation, etc., for each open-shop FORTRAN job that was run through its batching system. Studying these statistics, an observer commented: "If we had the same processor behind JOSS that we do behind our FORTRAN open-shop system, it looks as if over 50% by number of our FORTRAN open-shop jobs would be handled better in the JOSS interpretive mode than in the FORTRAN mode." This is on a straight-time and execution basis.

A RAND spokesman comments: "It is interesting to note that JOSS could handle about half of our jobs by number—which is certainly much less by time. I feel that this is a misleading figure, however, for the following reasons: I think that we came up with an average time in a 7090 FORTRAN job of, say, 40 seconds for compilation and perhaps 10 seconds for execution. We feel (we have no way of measuring this, of course) that of this 40 seconds of compilation, perhaps 35 seconds are spent compiling statements which were compiled on a previous run, and five seconds are spent compiling out the changes to correct a bug. Similarly on execution: out of that 10 seconds, eight seconds may get you back to where you quit in the previous run, and two seconds are spent going from the last bug to the next bug. We don't know how to measure this, but we feel that if we could, this would put the figure even more in favor of the JOSS system."

The power of the system and the simplicity of the language can be seen from the list of JOSS words and symbols below, and from the scenario of a filmed report on the system. The scenario is excerpted below. The principals are J. C. (Cliff) Shaw, a user (Fred Gruenberger), the narrator (C. L. Baker), and W. H. Ware, head of the Computer Sciences Dept. at RAND.

[User presses POWER button, and JOSS responds by typing:]

JOSS: 'IN' for JOSS service.

[User presses IN button.]

JOSS: Please identify yourself.

User: RAND

JOSS: 3-21-64 RAND*

User: Type 2+2.

JOSS: 2+2 = 4.

Shaw: An executive routine in JOHNNIAC serves up to eight users at a time by interpreting and carrying out commands given in a simple language. Let's watch Fred Gruenberger and Charlie Baker demonstrate some of the features of JOSS.

Baker: JOSS may be used in three ways: as a printing desk calculator; as a stored-program computer; and, most productively, as a combination of these two. First, we will demonstrate some of JOSS' arithmetic capabilities.

User: Type 2+2—3—3+4+4.

JOSS: 2+2—3—3+4+4 = 6

Baker: Multiplication must be indicated by a centered dot.

User: Type —1.2 • 1.2.

JOSS: —1.2 • 1.2 = —1.44

Baker: We divide by using a slash . . .

User: Type 7/2.

JOSS: 7/2 = 3.5

Baker: . . . and an attempt to divide by zero will result in an error message.

User: Type 7/0.

JOSS: Error above: Zero divisor.

Baker: An exponent is indicated by an asterisk.

User: Type 2*8.

JOSS: 2*8 = 256

Baker: Several expressions may be evaluated on one line, and a blank line is indicated by an underscore.

User: Type 3+2, 3—2, 3 • 2, ___, 3/2, 3*2.

JOSS: 3+2 = 5
3—2 = 1
3 • 2 = 6
3/2 = 1.5
3*2 = 9

Baker: The arithmetic expression shown in this chart may be evaluated by the addition of parentheses and brackets.

$$\frac{.7^{5}-2.3^{-.7}}{7^{6}-\left|9^{5}-5^{7}\right|^{2/3}} = ?$$

User: Type [.7*5—2.3*(—.7)]/[7*6—|9*5—5*7|*(2/3)]

JOSS: [7*5—2.3*[—.7)]/[7*6— |9*5—5*7|*[2/3)]
= —3.33630357 • 10*(—6)

Baker: A number of functions are available. The argument is enclosed in parentheses.

User: Type sqrt(49).

JOSS: sqrt(49) = 7

User: Type sqrt(sqrt(sqrt(sqrt(3*2*2*2*2)))).

JOSS: sqrt(sqrt(sqrt(sqrt(3*2*2*2*2)))) = 3

Baker: Trigonometric functions are provided, and we notice that numbers may be up to nine decimal digits.

User: Type cos (3.14159265/4).

JOSS: cos(3.14159265/4) = .707106781

Baker: Logarithms are to the base e.

User: Type log(2.71828183).

JOSS: log(2.71828183) = 1

User: Type y.

JOSS: Error above: Undefined.

Baker: JOSS is correct. The variable y is undefined. But a "Set" statement will serve to define it.

User: Set y = exp(10).
Type y.

JOSS: y = 22026.4658

Baker: JOSS functions are available to take the integer part of a number, the fraction part, the digit part, and the exponent part.

User: Type y, ip(y), fp(y), dp(y), xp(y).

JOSS: y = 22026.4658
ip(y) = 22026
fp(y) = .4658
dp(y) = 2.20264658
xp(y) = 4

Baker: To use JOSS as a stored-program computer, we may ask JOSS to store a statement rather than execute immediately. This is indicated by prefixing the statement with an identifying number.

User: 1.1 Type x, sqrt(x), log(x), exp(x), ___.

Baker: We will now ask JOSS to execute this step a number of times, varying x from 1, in steps of 1, to 100.

User: Do step 1.1 for x = 1(1)100.

JOSS: x = 1
sqrt(x) = 1
log(x) = 0
exp(x) = 2.71828183
x = 2
sqrt(x) = 1.41421356

---

*This line is typed by JOSS at the top of a new page (8½" x 11"). One-inch margins and similar date-lines are automatically supplied for each page.

DIRECT or INDIRECT:

```
Set x=y+w*2-cos(z/c).
Set x(3)=y(i)+z(j).
Set z(4,7)=3+c(i).

Do part 1.
Do part 1 for x=1,2,y,z.
Do part 1 for z=1(2)9(5)z,5,a(b)c.
Do step 1.1.
Do step 1.1 for x=1(-1)-9.

Type 2+3+5.
Type x.
Type x,y,z,cos(z),_.
Type X(2),Y,a(i,j).
Type x,y in form 2.
Type a,b,_,c(1) in form 3.
Type size.
Type "abcde".
Type step 1.1.
Type part 1.
Type form 1.
Type all steps.
Type all parts.
Type all forms.
Type all values.
Type all.

Delete x.
Delete a,b,c,x(i,j).
Delete all values.

Line.

Page.
```

DIRECT (only):

```
Cancel.

Delete step 1.1.
Delete part 1.
Delete form 1.
Delete all steps.
Delete all forms.
Delete all parts.
Delete all.

Go.

Form 1:
Any string of characters.
Form 2:
alt.=-__.____ vel.=.........

x=3+sqrt(7)
y=x+y-z*2/7
z=4
```

INDIRECT (only):

```
1.1 To step 1.5.
1.7 To part 3.

2.3 Done.

4.1 Stop.

6.1 Demand B(5).
6.2 Demand a(i,j).
```

RELATIONS:

$= \neq \leq \geq < >$

OPERATIONS:

```
+ = . / * () [] ||
```

CONDITIONAL CLAUSES:

```
if a=b
if a<b<c>d
if sin(x)>cos(y)
if x=5 and y=3 or z=2+x>3.
```

NUMBERS:

```
2
3.1415926
.003
4.535976·10*(-4)
7.9846·10*7
-4.6
-.06123·10*(3)
```

FUNCTIONS:

```
sqrt(x)
log(x)
exp(x)
sin(x)
cos(x)
arg(x,y)          (arctan y/x)
ip(x)             (integer part)
fp(x)             (fraction part)
dp(x)             (digit part)
xp(x)             (exponent part)
sgn(x)            (sign part)
max(a,b,...,z)
min(a,b,...,z)
```

```
PUNCTUATION and
SPECIAL CHARACTERS:

. , ; : ' " _ # $ ?

-__.______ indicates a field
   for a number in a form.
......... indicates scientific
   notation in a form.
# is the strike-out symbol.
$ carries the value of the
   current line number.
* at the beginning or end of
   a line kills the line.
```

```
        log(x) = .693114718
        exp(x) = 7.3890561
```

Baker: Our table would look much better with each set of values on one line. Let's interrupt JOSS by pressing the INTERRUPT button. [User presses INTERRUPT button.] JOSS has calculated several lines and will stop when they have been printed.

```
JOSS:   x = 3
        sqrt(x) = 1.73205081
        log(x) = 1.09861229
        exp(x) = 20.0855369
        x = 4
        Interrupted at step 1.1
```

Baker: We will change our program step by replacing it with one that calls for typing in a form.

User: 1.1 Type x, sqrt(x), log(x), exp(x), in form 1.

Baker: We must define this form.

User: Forml:*

Baker: JOSS does not require us to be perfect typists. A line may be killed by ending it with an asterisk, and JOSS will ignore this line.

User: Form 1:

```
        _____  __.__________  __.__________
        .............
```

Baker: We have corrected our mistake; we may now use underscores in the form to indicate the position of fixed point numbers, and a string of dots to indicate that scientific notation is desired.

User: Go.

Baker: "Go" tells JOSS to resume calculation at the point of interruption.

```
JOSS:   4   2.0000000   1.3862944   5.4598150 01
        5   2.2360680   1.6094379   1.4841316 02
        6   2.4494897   1.7917595   4.0342879 02
        7   2.6457513   1.9459102   1.0966332 03
        8   2.8284271   2.0794415   2.9809580 03
        9   3.0000000   2.1972246   8.1030839 03
        Interrupted at step 1.1.
```

Baker: Our output looks much better now. Let's dress it up with a heading. Again, we have interrupted.

User: 1.05 Do part 2 if fp(x/40) = 1/40.

Baker: JOSS will insert step 1.05 before our "Type" statement, and will request part 2 as a subroutine every forty lines.

User: 2.1 Page.
        2.2 Type form 2.

Baker: Part 1—that is, all steps numbered two-point-something—asks for a new page and a heading line. We must define this heading.

User: Form 2:

```
        X     SQRT(X)     LOG(X)     EXP(X)
        Do part 1 for x = 1(1)100.
```

Baker: This time we have asked JOSS to start all over from the beginning.

```
JOSS:   X   SQRT(X)     LOG(X)      EXP(X)
        1   1.0000000    .0000000   2.7182818 00
        2   1.4142136    .6931472   7.3890561 00
        3   1.7320508   1.0986123   2.0085537 01
        4   2.0000000   1.3862944   5.4598150 01
        5   2.2360680   1.6094379   1.4841316 02
        6   2.4494897   1.7917595   4.0342879 02
        7   2.6457513   1.9459102   1.0966332 03
        8   2.8284271   2.0794415   2.9809580 03
        Interrupted at step 1.05.
```

Baker: This looks much better, but an occasional space between lines would dress up our output nicely, so we have interrupted again. Step 1.07 should give us an extra space every five lines.

User: 1.07 Line if fp(x/5) = 1/5.
        Do part 1 for x = 1(1)100.

```
JOSS:   X   SQRT(X)     LOG(X)      EXP(X)
        1   1.0000000    .0000000   2.7182818 00
        2   1.4142136    .6931472   7.3890561 00
        3   1.7320508   1.0986123   2.0085537 01
        4   2.0000000   1.3862944   5.4598150 01
        5   2.2360680   1.6094379   1.4841316 02

        6   2.4494897   1.7917595   4.0342879 02
        7   2.6457513   1.9459102   1.0966332 03
        8   2.8284271   2.0794415   2.9809580 03
        [And so forth].
```

Baker: Now that our program seems to be checked out, we may ask JOSS for a copy of the program, the forms, and the value of all variables.

User: Type all.

```
JOSS:   1.05 Do part 2 if fp(x/40) = 1/40.
        1.07 Line if fp(x/5) = 1/5.
        1.1 Type x, sqrt(x), log(x), exp(x) in form 1.
        2.1 Page.
        2.2 Type form 2.
        Form 1:
        _____  __.__________  __.__________
        .............
        Form 2:
        X     SQRT(X)     LOG(X)     EXP(X)
        x = 100
        y = 22026.4658
```

Baker: We can now communicate with JOSS well enough to attempt a problem. An example is found in McCracken's *Guide to FORTRAN Programming*†. First, we'll need the constant 2π.

PROBLEM:

$$i = Ie^{-Rt/2L} \text{ SIN } 2\pi ft$$

STARTING AT $t = 0$,

EVALUATE FOR 100 INTERVALS

AT ABOUT 10 INTERVALS PER CYCLE.

†McCracken, D. D., *A Guide to FORTRAN Programming*, Wiley and Sons, Inc., New York, 1961.

User: Set k = 2•3.14159265.
Baker: Next, we can store a program to compute the required factors, and an approximate time interval . . .
User: 1.1 Set F = sqrt[1/(L • C)]/k.
1.2 Set f = sqrt[1/(L • C)−R*2/(4 • L*2)]/k.
1.3 Set I = k • F*2 • Q/f.
1.4 Set d = .1/f.
Baker: . . . and we'll type out these values.
User: 1.5 Type F, f, I, d.
Baker: Next, we need to enter the parameters for our specific problem.
User: Set R = 100.
Set L = .2.
Set C = .5 • 10*(−6).
Set Q = .001.
Baker: Now we ask JOSS to compute these factors.
User: Do part 1.

JOSS: F = 503.292122
f = 501.716868
I = 3.17220634
d = 1.99315603 • 10*(−4)
Baker: These look fine, with the exception of that rather messy looking time interval. We can change it to a clean one, and, knowing that our interval is small, our program can print the time in milliseconds rather than seconds.
User: Set d = .0002.
2.1 Type 1000 • t, I • exp[− R • t/(2 • L)] sin(k • f • t) in form 3.
Form 3:
___•___ms. = ___•__________ amp.
Baker: The program to compute and type consists of a single step which will print in a single form. The form may include any additional information we wish to print out with our answers. We can now ask JOSS to execute our step for 100 time intervals, from t = 0, in steps of d, to 100 times d.
User: Do part 2 for t = 0(d)100 •d.

JOSS:	.00 ms.	.000000 amp.
	.20 ms.	1.778902 amp.
	.40 ms.	2.733649 amp.
	.60 ms.	2.591195 amp.
	.80 ms.	1.508396 amp.
	1.00 ms.	− .026650 amp.
	1.20 ms.	−1.405807 amp.
	1.40 ms.	−2.136197 amp.
	1.60 ms.	−2.010678 amp.

[And so forth].
Baker: Since our program seems to be gold star, let's record it.
User: Type all.
[See Fig. 3].
Baker: The program is finished, so we might as well erase it, its forms, and all associated values from JOSS' storage.
User: Delete all.
Baker: JOSS can operate with indexed variables, and a stored program may call for values of variables to be entered during execution.
User: 1.1 Demand q(x,y).
Baker: We'll vary the value of the first index x from 1, in steps of 1, to y, . . .
User: 2.1 Do part 1 for x = 1(1)y.
Baker: . . . and vary the index y from 1, in steps of 1, to 8.
User: Do part 2 for y = 1(1)8.
Baker: We enter the value of each variable on the same line as it is requested by JOSS.

JOSS: q(1,1) =	3.456
q(1,2) =	−4.32
q(2,2) =	8
q(1,3) =	4.324567
q(2,3) =	−56.45
q(3,3) =	0.000987
q(1,4) =	4.44444444
q(2,4) =	−4.444433
q(3,4) =	

User: 3.456
−4.32
8
4.324567
−56.45
0.000987
4.44444444
−4.444433

Baker: When all these values have been entered, we may proceed to have JOSS calculate with indexed variables as would be required for problems more complex than we have been able to demonstrate here.
Ware: You have just seen a demonstration of many features of the JOSS system. As was pointed out, this is an experimental system designed to provide individual scientists and engineers with a personal, on-demand computing service. As a side aspect of the experiment, we hope to gain insight into the interface problem between a man and a computer.

The system shown here is currently available to RAND's professional staff. Our experience in using JOSS, our observations of user behavior, will be the basis on which to continue research. Later-model systems will certainly differ from what you have seen.

JOSS is one of many current efforts in on-line time-shared use of computers. The goal of the JOSS project has been to demonstrate that the ability to be on-line with the computer, given a reasonable language and almost continuous interaction, leads to a powerful computational tool. It is probably too early to forecast the ultimate effect of such systems, but for a certain class of problems, at least, the programmer as the middleman between the problem and the machine is no longer needed.

JOSS represents, we believe, a significant step forward because of its intimate interaction between man and machine. ■

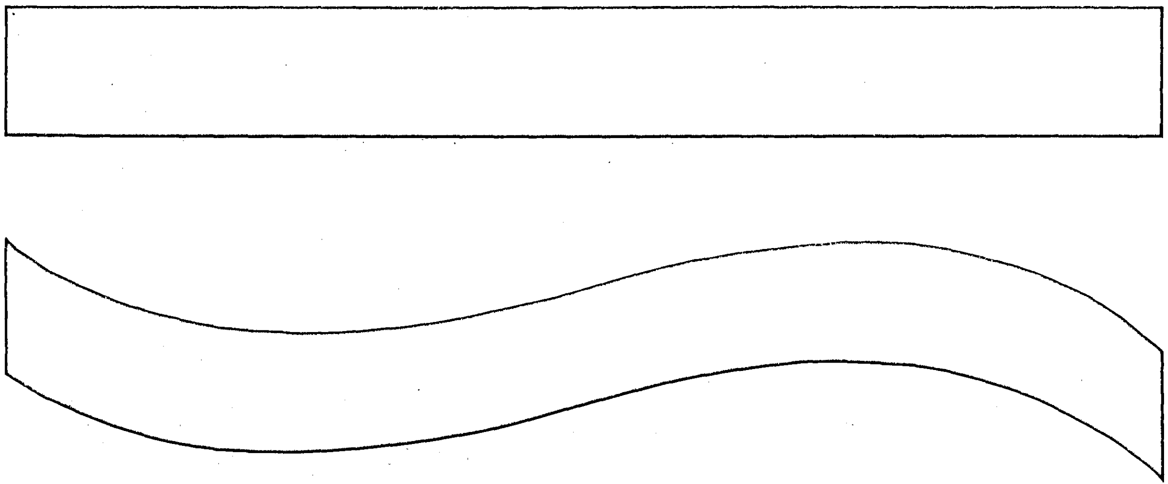
Fig. 3

```
1.1 Set F = sqrt[1/(L·C)]/k.
1.2 Set f = sqrt[1/(L·C)-R*2/(4·L*2)]/k.
1.3 Set I = k·F*2·Q/f.
1.4 Set d = .1/f.
1.5 Type F, f, I, d.

2.1 Type 1000·t, I·exp[-R·t/(2·L)]·sin(k·f·t) in form 3.

Form 3:
__.__ ms.    -_.______ amp.

      d =     2·10*(-4)
      f =   501.716868
      k =     6.2831853
      t =     2·10*(-2)
      C =     5·10*(-7)
      F =   503.292122
      I =     3.17220634
      L =      .2
      Q =     1·10*(-3)
      R =   100
```



A PANEL DISCUSSION ON TIME-SHARING

CORBATO: If we judge by the size of the audience, it is clear that we find ourselves with a subject which people feel is still an issue. It is not clear what is going to happen in this area of time-sharing, and indeed we find many differences of opinion even among experts, if there are any. The speakers on the panel were chosen for their active engagement with working systems or in considering the objectives of such a system. There are differences of opinion among the speakers, stemming, I think, from two causes. One cause is that people are looking at the same problem from different viewpoints although with similar objectives. A second cause is that there is actually a very large difference in the objectives, often leading to radically different statements. I hope we can reveal sharply in this discussion what these differences are.

The speakers are Prof. Alan Perlis of the Carnegie Institute of Technology, noted for having worked on the IT translator years ago and a former president of the ACM. Jules Schwartz, of the System Development Corp., has long been active in programming, has a language, JOVIAL, named after him, and is now major domo of the time-sharing system at SDC. Prof. John McCarthy is from Stanford Univ., is the originator of the LISP language for list processing, and has long been an advocate for time-sharing. Robert Patrick is a private consultant, has long been active in the programming field, and is a respected and articulate critic of blue-sky claims. As moderator, I feel I should indicate my bias in the sense that I've been long associated with the active development of time-sharing systems at the computation center at MIT. And now I'm deeply involved with the development and operation of Project MAC.

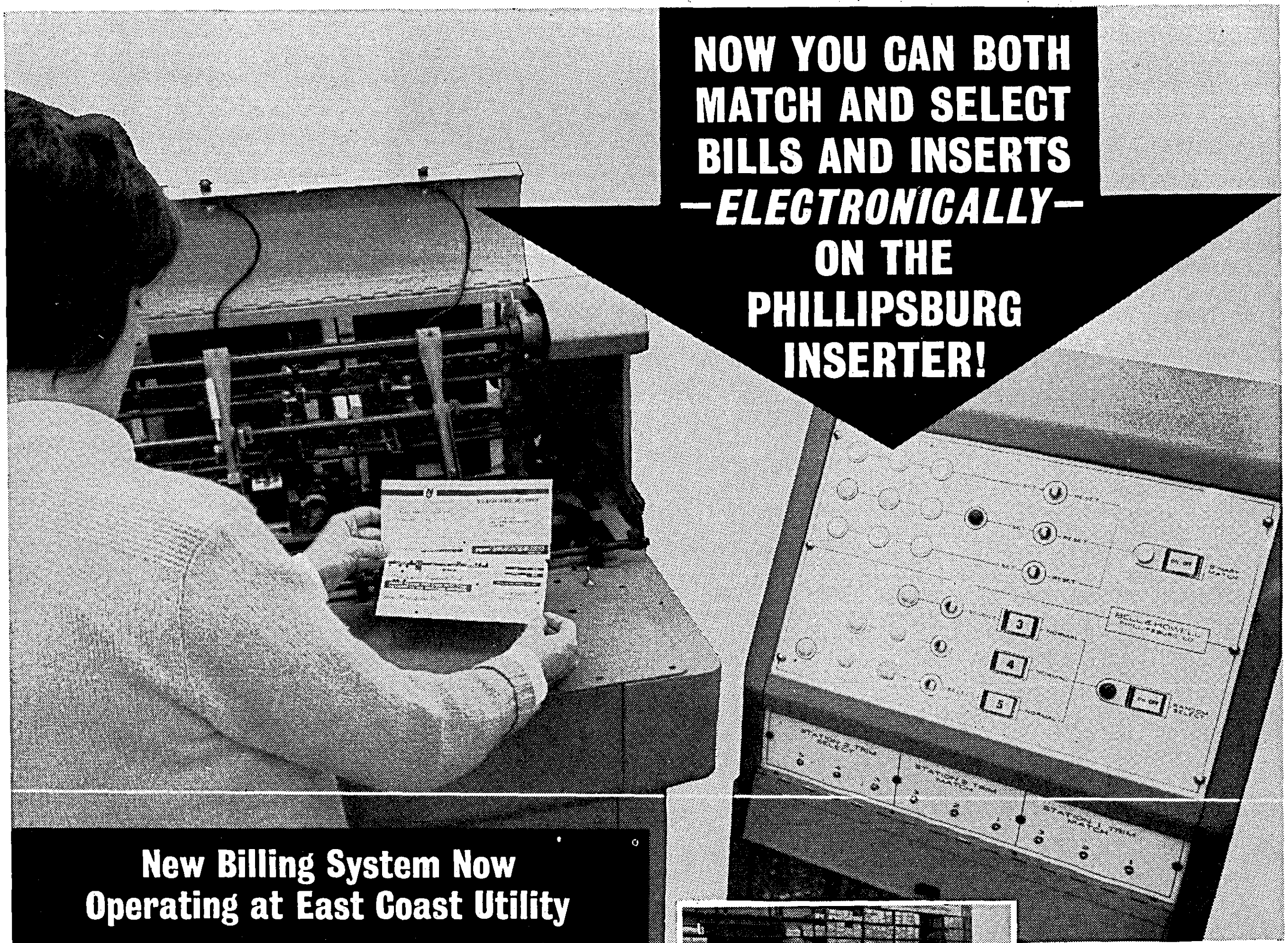
Mechanically, the systems represented by the panelists are different—they have different equipment and use different techniques. At Carnegie Tech, they have a G-20 computer which in part acts like a secretary for consoles that, by essentially a filing technique, accumulate and edit programs to be run between the normal jobs in the batch operation. In this sense, it is a Console system but Console jobs are run to completion. It's a very straightforward way of getting a lot of mileage out of a rather simplified control technique.

At Stanford, they are time-sharing a PDP-1, and at Project MAC and the Computation Center at MIT there are two separate 7094's being time-shared. At SDC we find a system similar to our own and similar in spirit to the PDP-1 but on a larger scale. All of these are interrupting jobs before completion, and actively commutating among users.

All the systems are roughly similar—they basically commutate among users and spread the computer service around so that each user feels that he has a part of a computer at his fingertips. There are a great many hardware compromises present in these systems and there are differences and similarities in both the objectives and philosophy. I think the hardware is less of concern because it's often the result of pragmatic issues than of philosophical objectives.

Let me start off the discussion of time-sharing by stating what is perhaps an extreme position of Project MAC's philosophical objectives, and this should lead to a statement by each speaker. We are trying to explore the notion of a computer utility with a general purpose computer placed at the fingertips of many users. First, service is the key point that we're emphasizing. It's facilities, and not hardware, that we're trying to spread around. We're contemplating a system much like the phone system where one may have excess capacity in order to gain service. It's not quite the same as the phone system, though, in that there are production jobs you can use to fill in the slack computer time. In any case, we visualize a system where, in the future, all jobs are started at a console even though they may later turn into production and the user may go home. Secondly, we feel that the user will ultimately be unconcerned with hardware in the programming or usage sense; he will not know the brand name of the central computer. The third point is that the programming systems, we feel, are everything. And we expect this to expand the base of system users and uses manyfold in the future.

PERLIS: The purpose of time-sharing is to take advantage of the fact that much important computation is of short duration and requires intermittent access to a computer. We are expending some effort in the development



THE PHILLIPSBURG INSERTER ELECTRONICALLY—

- **MATCHES code-punched cards with their correspondingly coded printed bills.**
- **SELECTS past-due notices, resident and business advertising, product literature, renewal notices, special subscription offers, etc.**

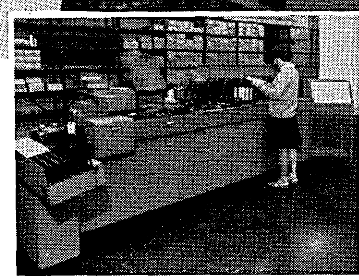
A new electronic breakthrough enables the Phillipsburg Inserter to accomplish two of the most important jobs in a billing operation: matching and selection of inserts. Imagine a volume billing operation in which it is possible to—

1. Guarantee that two mailing enclosures, such as a utility bill and cash card, are inserted in the same envelope,

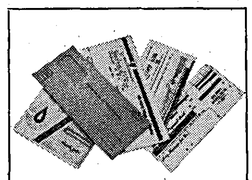
and, *at the same time*—

2. Automatically select one or more additional enclosures, such as an overdue notice, only for those recipients for whom it is appropriate. Contents of mailings can thus be varied, as required—and all without human intervention!

The system has extra dividends built in, too. Advertising material can be piggy-backed into the same mailing, at no increase in postage. There is no costly hand-sorting when payments are returned. The savings add up quickly. This is just the newest development in Phillipsburg's electronic-oriented program of research. There's always room for improvement—even in the Phillipsburg Inserter, the world's finest, most versatile inserting machine!



Operator holds matching inserts. Console for matching and selection is at right.



SEND FOR INFORMATIVE CASE HISTORY

We've wrapped up the entire matching-selection story in an actual case history of the New Jersey Natural Gas Co. Just mail the coupon for your copy.

Phillipsburg Division D 11 64
Bell & Howell Company
160 East Grand Avenue
Chicago, Illinois 60611

☐ Send me the Case History Report on Electronic Matching and Selection.

☐ Send me your brochure, "A New Advance in Mail Inserting Through Electronic Sensing."

Name____________________Title____________

Firm________________________________

Address______________________________

City__________State__________Zip code______

PHILLIPSBURG INSERTERS

160 East Grand Avenue, Chicago, Illinois 60611

A MEMBER OF THE DYNAMIC **Bell & Howell** BUSINESS EQUIPMENT GROUP

Micro-Data Div. **Rochester Film Div.** **Baumfolder Div.**
Micro-Photo Div. **Ditto Inc. Division** **Phillipsburg Div.**

Ditto of Canada, Ltd., Toronto, Ontario

CIRCLE 19 ON READER CARD

of programming systems for use in solving new classes of problems, which will include debugging techniques applicable to time-sharing computers.

"Time-sharing" is not accurate, although it's a good description and has a reasonably exact technical meaning. We are really sharing *facilities*, not *time*, unless we have several processors. We should distinguish between on-line use of a computer—that is, the real-time requirement for access to the processor and its facilities on demand—and what we might call a high convenience aspect: "I would like to get on the machine now, so you get off." The efficiency of this convenience depends very strongly on the distribution of job requirements. For example, if almost all jobs are of the same duration, then "time-sharing" tends to be inefficient and requires, in a sense, complete expansion of both the number of consoles and the storage and processor size almost indefinitely. Fortunately, in almost all cases, there is a distribution of job and response requirements. A principal difficulty in designing a time-sharing system is in finding the right way to measure the desired response time of a population of users so that those who require a fast response get it.

But besides the sharing of the central processor, other aspects of time-sharing research will have an important bearing on computer manufacturers and the general users. It will have a big effect, for example, on the way programming systems and languages are designed. We should realize that we will be sharing not only processors, but also programs; programs and processors are the same anyway. Thus, for a three-processor system capable of running simultaneously three programs, when each is running ALGOL, the system should permit three processors to simultaneously use the same program—rather than having three 12,000-word ALGOL programs taking up 36,000 words of core. So we're talking about sharing all of the facilities that are available in the system, and this intimates a different way of programming the language processors we design.

Another important aspect is hardware availability. Every time-sharing system that I know of uses equipment of many manufacturers, and consequently one of the most pressing problems that comes up is the development of interfaces between "foreign" equipments. It may be quite a while before one manufacturer will provide all the best tools for time-sharing: consoles are generally not standard equipment, a high-speed, high-capacity swapping drum may not be made by the manufacturer of your processor, and so on.

Also please note that as time-sharing develops and this concept of a central computer utility grows, so also grows the investment of every user in the stability of the system. This implies that the rate at which we replace computers has to be decelerated, or we have to find another way of rationally expanding the capacity to fit requirements (bound to grow) of a time-sharing system. Thus, if you have a 32K machine, how do you get it up to 65K? If you have one processor, how do you put on two, and then three? If you have a six-microsecond core memory on a given processor, how do you introduce a one-microsecond core? And not throw out all programs. These, then, become important problems in the design of a time-sharing system which are consequences of its existence as a utility.

New programming problems are also created. As soon as we introduce the remote use of computers, we reach the idea of the delay or deep freeze of a program being executed. If we modify such a program, having come across an error, do we recompile an entire program or do we do an incremental compilation? How many such compiling systems are available? It is easier to build interpreters to work this way, but you can't run 600 to 1,000 jobs a day off such interpretive systems.

Similarly for such systems, we must consider that everything may have to be done remotely: the maintenance engineer should learn how to maintain the computer from remote stations, and the people who design programming systems, hopefully including the monitors and executive systems, must learn how to debug them from remote consoles. In short, as we distribute this system further and further among more users distributed farther and farther apart, the necessity for everybody to behave like a remote user and for having guaranteed stability while growing becomes ever more important. I think these create very critical problems for the designers of these systems.

SCHWARTZ: In our time-sharing research, we're trying to study the advantages and disadvantages of this mode of operation, and trying to produce better techniques. For example, we're recording, as the days go on, how people and programs behave under time-sharing. From this, we're extracting, we hope, a lot of useful information. We're also simulating the time-sharing system with mostly our parameters and our computer characteristics built into it; it's proving quite useful for studies of future time-sharing systems. And we're carrying on an analytical study very similar to the one discussed in the paper presented today* —an analytical study based on a model corresponding to our system, rather than the one described today.

The on-line use of computers has great value for many things. The debugging of programs is one of the most popular uses of our system. Program preparation is another, although it is not one of the stronger aspects of our system. And another is information retrieval, such as for hospitals and police personnel, which are actually being run in our system now. These are some of the reasons that our time-sharing system exists.

The system itself is general purpose; users utilize teletypes, CRT displays, other computers, tapes and discs pretty much in the same general fashion as they would without time-sharing. There is a great variety of programming languages, and many types of work are done. The response is usually a matter of seconds; throughput and response time have not been a major problem yet.

McCARTHY: With regard to goals, ours are the same as that described by Corbato, namely to make computing into a public utility. In this respect, one must mention not only human interaction with a computer, but also the interaction of the computer with other kinds of equipment—for example with university laboratory apparatus. Two laboratories at Stanford have undertaken to connect their equipment to our system, and we'll see what their share of computer time will do for them in controlling their apparatus.

Now, I'd like to mention one thing which concerns this question of the range of sizes of jobs. We were quite proud of a previous version, since dismantled, called time-stealing on the '90. With that we were able, without stealing, to get the cost for a minimal LISP job down to 1.55 seconds of computer time. This was the complete time taken from the punching of one accounting card logging the job on, to the time it punched it logging it off—and don't think we didn't resent the tenth of a second or so taken by the card punch itself in punching the logging card.

* "Queueing in a Memory-Shared Computer," D. Fife and R. Rosenberg.

I'd like to mention one other distinction which I think some of the others made: the systems being described today are all general purpose systems, in the sense that the users can put their own programs in and run them. This compares to other sytems, described by their authors as time-sharing systems, which share a computer in some specific application such as airline reservations or simulating a desk calculator!

We must admit that difficulties in design of a system come from making all the kludges work, rather than the inability to solve theoretical problems concerning the expected distributions of waiting times. There are some parameters that seem to preclude theoretical design until we have more practical experience. I'd like to mention one of the worst problems confronting us and which concern other members of the data processing industry: the lack of inexpensive I/O equipment for time-sharing. The least expensive that can be visualized so far are directly-connected teletypes through interfaces, which come down to as low as $1,500 per user station. But if you want to go over any distance, or if you want to use the telephone company's facilities, you will get into squabbles with the telephone company.

PATRICK: My purpose here is to stimulate some thought so that we can understand and figure out where time-sharing fits in the overall scheme of things. Some people think that time-sharing will revolutionize the industry, that we'll all have consoles, and batch processing will rapidly die. I, on the other hand, don't think time-sharing will change our lives so radically. I view it as opening up a new market for computer services—one in addition to those we now serve with our large computers.

This (Fig. 1) is an illustration from a recent *Datamation* article, in which I tried to get a handle on the type of environment in which a computer operated. As I view the application of time-sharing now, it's an attempt to take

Fig. 1

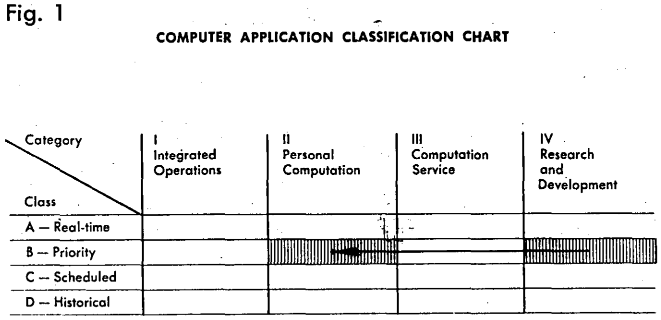
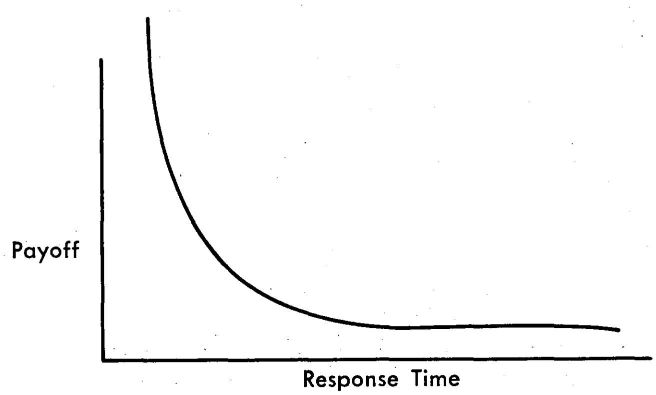


Fig. 2

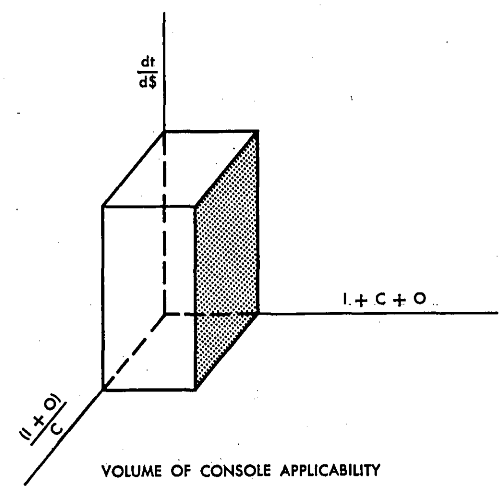


a well-proven, feasible technique, which handles priority applications in a research and development environment, and move it into a field environment where it will provide priority service for those who require personal computation.

Here (Fig. 2) is a payoff function from another issue of *Datamation*. I indicated in this article that I thought a value should be placed on the time of a man who has access to a time-shared system. It doesn't look like these things are going to be free; as a matter of fact, some of them are going to be moderately expensive. And if they are expensive, our managements will force us into giving them something in return for the added budget. The plot depicts the trade between the payoff and response time. As the curve shows, every job has some historical value regardless of when it gets done. However, some jobs have a much higher reward associated with early completion. Since it looks as if time-sharing is not going to be free, it appears that we're going to work on the left side of that plot, up where payoff versus response-time is fairly steep.

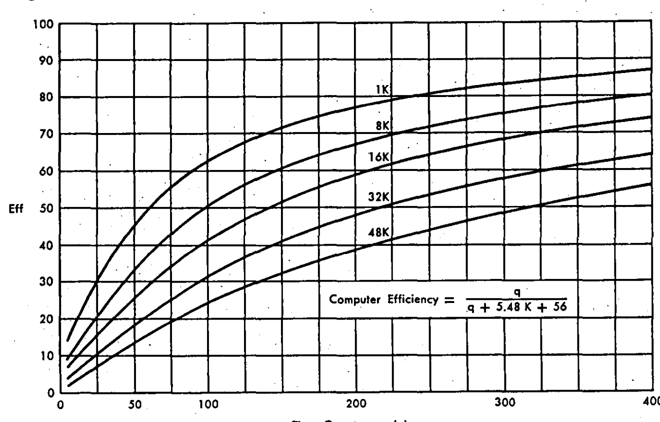
Here is a slide (Fig. 3) which I think depicts the volume of console applicability. The x axis is the sum of

Fig. 3



I + O + C. That's run-time to most of us. The y axis is the reciprocal of the sum of that payoff function; the payoff function is $\frac{d\$}{dt}$. The z axis is what I call the I/O balance; that's the ratio of input/output to the amount of compute you want to do. I see time-sharing for jobs which don't run very long, jobs for which there's a real payoff for finishing them up sooner, and jobs which have rather little input/output for the amount of running time they do require.

Fig. 4



This slide (Fig. 4) is a plot of the efficiency of the use of the computer itself. Efficiency is defined as the number of useful clock cycles of the mainframe you get, divided by the total number of clock cycles of the mainframe for the same period of time. The numbers were taken from the talk given by Jules Schwartz at this year's Spring Joint Computer Conference. I'm sure the efficiency figures can be improved somewhat by them by swapping better. . .

The problem is that you just flat run out of primary memory. There isn't enough core memory to load everybody who may have a program and is sitting at the consoles. So you have to take programs off the secondary storage device and bring them in, execute them, and put them back out where they resided before. This is very dependent on program size, the overhead required for scheduling and housekeeping, and the rotation rate of the secondary storage. Many of our systems are not completely outfitted with consoles; when they are, delays will occur and you'll measure to find out where the time went. With some of the current scheduling algorithms, the efficiency runs about 10% when you are reading card images from magnetic tape which you've pre-stored (so you're going to do a little data processing, which is moderately common nowadays—you want to read a few cards). If you compute a little longer, you can get the efficiency up to about 70%. Now, in no case are we making judgments about what the guy is doing with his time slice; we're just talking about machine efficiency.

Typically, a nice average might be about 40% machine efficiency. In such an environment, production programs will take 2½ times as long, elapsed time, than for batching. And if you distribute costs on the basis of CPU seconds, they'll cost 2½ times as much.

Now, we can consider machine interaction. From what I hear and what I've personally observed by sitting at consoles at SDC and at another time-sharing installation, you can't take this advanced machine interaction as proportional to the efficiency of the man who is sitting at the console. He can get more out. But the point I'd like to make is that he must get more out. He's got to get along with about a third of the computer time. The ratio is 130 net useful minutes per eight-hour day to 440 net useful minutes.

In addition, on a cost-efficiency basis, it looks like he's got to work to use the man *and* machine-time well. The break-even point is about seven to one.

PERLIS: I have no arguments with Jules and John, and I'm not sure that I have any with the last speaker. I merely wish to point out that it may well be that these time-sharing programs function a lot more than seven times more efficiently than they would in a batch processing environment—particularly when the batch systems are operating in very loaded systems where the turnaround time is 24 or 48 hours. It's quite difficult to measure the improvement. Certainly at our installation we can see significant improvement.

SCHWARTZ: I don't have any particular arguments with Professors Perlis or McCarthy. We can't go through the entire booklet [distributed by Patrick at the session] and discussion right now, but let me just point out the possible arguments. First, the advantages. The arguments here were quite clearly stated to be economic in nature—economic in a very particular way. There are some things that are hard to measure, but let's talk about them. There are many kinds of jobs and techniques which require on-line use of a computer for their existence, such as those I've mentioned. For these, it just is not economical to use a private computer.

This includes some solutions of problems by trial and error, and a number of others which require fast, rapid response human interaction. It turns out that there are a number of jobs we can do which do not require rapid human interaction, but because one had to use the time-sharing system he began to use it for that job and found that by goodness the time-sharing system actually helped. He approached it in a different way, but found that he was gaining something from on-line interaction. Now, it is true as has been mentioned several times that the scheduling techniques in the algorithm certainly leave something to be desired. Time-sharing is fairly new as things go, not as old as batch processing. We know of a number of techniques which can improve it.

The particular algorithm shown on the slide was one which tried to minimize response time without much thought given to the efficiency of computer use. It's an algorithm with some interesting points, but it certainly can be changed in a number of ways, some of which include the measure of efficiency as a factor in the quanta of time. And by the way, our record does show our efficiency in almost all cases is better than shown by Patrick.

You can't underestimate the value of on-line usage for such things as debugging. Once you get on to it, you find the measure of elapsed time—not the economics of computer use on the spot, the time it takes to get a job done, in days and weeks—goes down. Some people mention factors of hundreds, others will say factors of 30%. That's the minimum improvement I've ever heard. Then again, how efficient can batch processing be? It is true that they haven't improved as much as they should have, but who says they can? And there are a number of aids for time-sharing which users can get that you just can't get with off-line processing.

McCARTHY: Patrick is off by a factor of at least 30 in costs. First of all, a computer with performance comparable to the Q-32 in use at SDC can operate at costs closer to $100 an hour, rather than $1,000 an hour. That's my factor of 10. The other factor of three comes from the supposed efficiency of 27%. This factor of efficiency in a time-sharing system is based on the amount of memory and what use is made of it. By relating various functions to sizes of jobs, in my opinion, we can get the factor of 80%. The fact that any well-run job shop gets an efficiency of 90% is a statement that I find dubious. I must confess that I do not believe it to be the case at Stanford and I don't know where else they might achieve such a level except when they're running large production jobs.

Now, concerning the costs, we have figured that if we can get people to come in 24 hours a day over four years, we could charge people a fixed cost just to sit in front of the console. We could recover our costs, and so forth, at a dollar an hour. You must, of course, multiply this by some factor, say three, to get a first-shift charge. We will do some things to encourage people to come in during slack hours at a lower price.

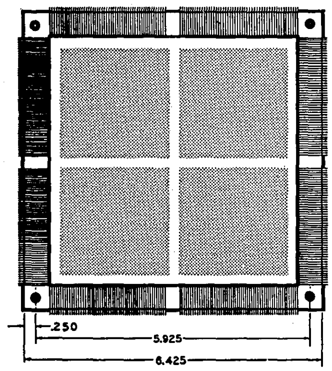
As to cost per station, Patrick mentions a system cost of $32,000 per station. I don't have our figures on cost per system, but I know it can be brought down to $1,500. That's what it is for our teletype stations.

PATRICK: I don't want anybody to think that I'm attacking Jules. He wrote down some numbers that I used. The *utilization* of systems, I'm sure, is 100%, but that isn't what I talked about. I was talking about the *efficiency* of use—net useful divided by total.

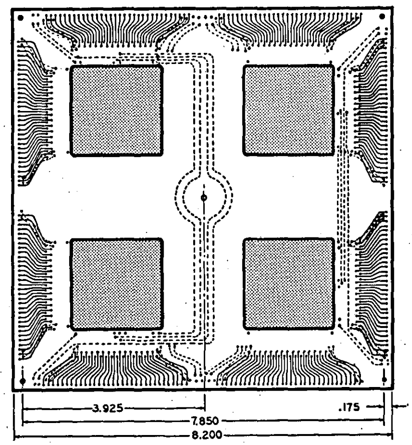
There are some good uses for time-sharing. I was at an Air Force test base recently, where they have to check out their large radars and test instrumentation before every test and recalibrate after each test. The computers are denied to the programming staff while they are supplying test signals to the radars. Here is a very good use of time-sharing, similar to the use that Jules puts his Q-32 to in driving the command systems laboratory displays. They want a little bit of a lot of computer for only a short period of time. Then, when they get ready to play with large simulations or to run a test, they want

Why look further for Core Memory Frames?

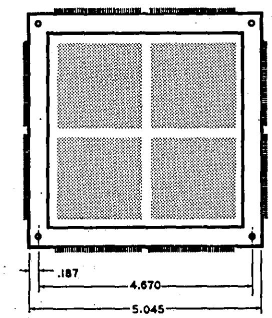
One of these three Fabri-Tek standards probably will fill your need!



Laminated frame with dip-soldered terminations
Four quadrants of 4,096 cores each per frame • 30-mil cores on 30-mil centers • Available in choice of three core-types • We'll stack it, package it, and terminate it to fit your requirements



Printed circuit quad-frame
4,096 cores per quadrant • 30-mil cores on 30-mil centers • Available in choice of three core-types • Solid copper shielding • We'll stack any number of planes and package the way you want



Molded frame with dip-soldered terminations
Four quadrants of 4,096 cores each per frame • 30-mil cores on 30-mil centers • Available in choice of three core-types • You choose the number of planes and packaging you want

The above frames are available in these core choices. Pick the one for your application!

Core Type	Full Drive Current @ 25° C.	"One" Output (4,096 core array)	Ts (max.)	Tp	Memory Cycle Time
A	450 ma.	30 mv.	0.45 usec	0.25 usec	2.0 usec
B	520 ma.	35 mv.	0.42 usec	0.22 usec	1.5 to 2.0 usec
C	640 ma.	35 mv.	0.40 usec	0.22 usec	1.5 to 2.0 usec

If one of these three doesn't suit your requirements, we have frames, stacks, and systems that are bigger, smaller, faster and slower. In fact, anything you want in a memory that doesn't move is available at Fabri-Tek. Call, write, or wire Robert E. Rife, Fabri-Tek Incorporated, Amery, Wisconsin. Phone: COngress 8-7155 (Area 715). TWX: 715-292-0900.

FABRI-TEK INCORPORATED

CIRCLE 21 ON READER CARD

all of a big computer. There's no quibble about this kind of an operation being very appropriate for time-sharing.

But as I recollect when I was a student, the student's time was the cheapest thing on the campus.

I don't understand how I can have a console for $1,500. That's just an I/O unit. To that I must add the telephone lines, which as an accountant I have to charge off or at least write off, and I've got to divide the central facility (which costs a million dollars or so) by the number of consoles it serves. That's where I get $32,000, as it says in the paper—for an IBM 360 model 50 with 90 consoles hung on it.

CORBATO: I think we observe a rather sharp difference of opinion. I'd like to suggest that different members of the panel are looking at different parts of the problems. Bob Patrick, I think, is concerned with hardware efficiency in a sense, while the other members are concerned with human efficiency. Now, the two are not related, obviously. I would like to suggest, too, from my own experience that this matter of cost is just like computer prices. Computer price has nothing to do with what it costs to build it. Similarly, what you choose to charge for the various services is rather arbitrary if the services are distinct and not comparable. It's quite easily done, and in fact we do it today in our own system—namely, charge the production runs, even though they are running within time-sharing, in such a way that they don't pay one iota more for the fact that they are being time-shared. In other words, time-sharing need not cost more. The issue, I think, is really more fundamental than that. It is whether one ought to be considering systems capable of serving all applications. Should we, as Patrick is suggesting, consider only some of the new ones, or, as some of the others and I suggest, all of them? Are there any more questions from the panel?

McCARTHY: What about this $1,000 an hour? Don't you think that's an unreasonable figure?

PATRICK: Well, let's ask Corbato what his twin '94 costs . . . what would that be on the market?

CORBATO: The 7094's we have at MIT are approximately 50% more expensive in rental than the traditional two-channel 94's that are on the average floor. [Ed. note: an hour of prime-shift time on a two-channel 7094 in Los Angeles costs $650.] They're overpriced because they weren't designed to be worked this way. There's no criticism here; the point is that time-sharing people are getting better service and thus should pay for it. They're getting few-second turnaround time; if 24-hour turnaround time is what one wants, it's very easy to get the price down.

PERLIS: I'd like to make a remark about batch processing. When you batch, a job enters, is run to completion, and exits. That ends the job, and anything after that is an entirely new computation. In point of fact, jobs run in batch processing are rerun several times. Each time the complete sequence of facility uses is followed including the finding and correction of a few more mistakes. So, when we are batch processing, it is just as though we were operating the computer in the interrupt mode except that we are doing a complete processing job every time. One has to examine the effectiveness of batch processing to find how many times we are re-running the same job with only minor changes. Certainly in open-shop computation this is done, I'd estimate, seven to 10 times per job.

PATRICK: We talked about people efficiency. With a large computing system, such as the ones we've described, if you need one minute of raw compute to produce an answer—an intermediate result, if you will—you have to sit at a console for 40 minutes with the way these things are time-slicing. Now, if you're going to have some man-machine interaction, you supposedly are popping an answer out every once in a while. The guy is making a decision whether to stop the computation and perhaps gain some of the benefits that Perlis just discussed or whether he wants to continue. So the programmer is tied up for 40 minutes at his console to get one minute of raw computer time.

Now, for some kinds of jobs which require lots of interaction, this is very good usage of the man because he is sitting there and making human-type decisions (i.e., those we can't program) all the time. Whenever something prints out, he says: yes, that's reasonable, and he elects to continue by not interrupting. However, if you're doing some differential equations work or some of the larger mathematical computations, that's quite slow. The turnaround time problem stems from the fact that our capacity is saturated. You can get good turnaround time if you have more computer capacity than you require. But if everybody immediately pounces on a console the minute the machine is turned on in the morning, then some of them are going to have to wait. If you have half as many consoles as you have programmers, then half the programmers are waiting for the other half to get off. I just don't understand how we can say that it improves our turnaround. The people efficiency is not necessarily much better. For those served, yes, but batch systems work well for those with high priorities—they get pretty good turnaround time now.

CORBATO: We've reached the end of the time. I think the speakers have spoken rather forcefully, and you can probably hear what you'd like to hear out of what's been said. I think it's clear that the issue and the subject is not closed and will continue in the future. ■

CIRCLE 22 ON READER CARD

. . . AT LAST — THE FIRST TRUE HYBRID COMPUTER

A completely new kind of signal processor, the Adage AMBILOG™ 200 computer is designed from the ground up to exploit the best of both analog and digital techniques. It combines parallel hybrid arithmetic with stored-program sequential operation.

High processing speeds (often ten times faster than comparably-priced conventional machines) and extensive input/output for both analog *and* digital data make AMBILOG 200 ideal for data acquisition, simulation, and information display.

PARALLEL HYBRID ARITHMETIC

AMBILOG 200 achieves high processing speeds through parallel organization of hybrid computing elements all operating simultaneously on analog and digital operands. Word length of digital operands is 15 bits; analog accuracy is .01%. A fast 30-bit digital accumulator augments the 15-bit hybrid arithmetic.

A typical configuration performs 12 additions, 2 multiplications and 1 division in ten microseconds.

Hybrid arithmetic, A/D and D/A conversion, comparison, and signal routing and conditioning are all carried out under direct control of the stored program.

SEQUENTIAL STORED-PROGRAM OPERATION

Fifteen and 30-bit data words are transferred at high speeds to and from memory and all other parts of the AMBILOG 200 under flexible stored-program control. Core memory word length is 30 bits; cycle time is 2 microseconds. Memory sizes up to 32,768 words are available, all directly addressable. Digital I/O devices include punched tape, typewriter, magnetic tape, and direct data channels. A unique multiple priority interrupt system permits complete servicing of interrupts in as little as 3 microseconds, and facilitates interconnection for multi-processor installations.

PROGRAMMING

Instruction word length is 30 bits, permitting simultaneous control of memory, source and destination selection, word rotation, and Boolean logical operations. The order structure includes provision for recursive indirect addressing and indexing, a number of conditional and unconditional jumps, and program traps.

The AMBILOG 200 Symbolic Translator permits programming in symbolic source language and is easily extended to accept any problem-oriented source language.

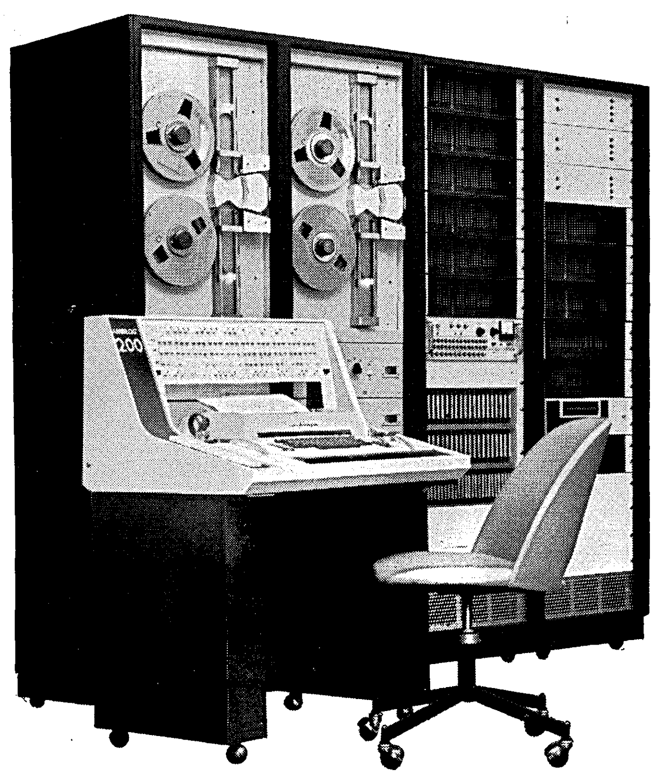
Software support includes programmer and maintenance training, installation and maintenance services, system programs, standard subroutines, and complete documentation.

BROAD APPLICATION

Flexibility, economy, and real-time processing power make AMBILOG 200 a "natural" for

Biomedical monitoring
Seismic data processing
Sonar and radar signal enhancement
Test stand instrumentation
Programmed automatic checkout
Factory test and inspection
Communications research
Process control
Space-vehicle simulation
Simulator trainers
Telemetry receiving stations
Visual display systems

For more facts, contact
I. R. Schwartz, Vice President
617-UN 4-6620.



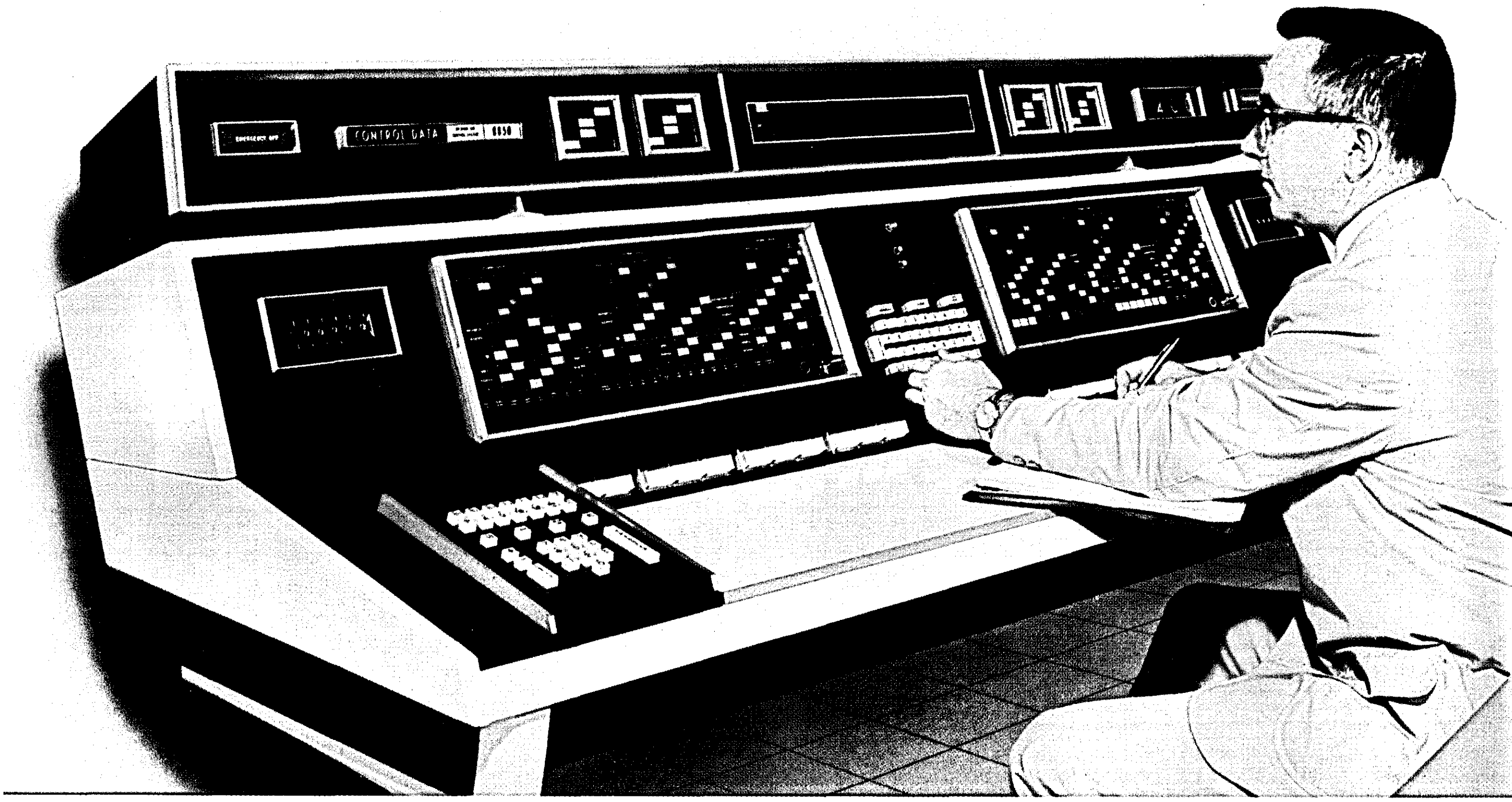
Adage, Inc. welcomes employment inquiries from professional engineers.

Adage INC

Main Office and Factory
292 Main Street
Cambridge, Massachusetts 02142

West Coast Plant
1145 East Ash Avenue,
Fullerton, California 92631

CIRCLE 23 ON READER CARD



Two computers are operated from this Control Data Corporation Console which uses six DATA•PANEL Information Displays.

NEW TEC-LITE DATA • PANEL

SOLVES FIVE INFORMATION DISPLAY PROBLEMS AT LOW COST!

Designers of computer consoles, display panels and status boards now have a strikingly beautiful, yet truly practical way to answer problems of visual impact, operator accuracy, appearance and display versatility. DATA•PANEL offers a totally new approach to custom designing and building information displays at a cost usually far below that for standard panels using conventional individual indicators.

TEC-LITE DATA•PANEL gives the designer almost unlimited freedom in size, style, color and arrangement of messages and indications within the display area. *Only when illuminated* are alpha-numeric messages and symbols visible in color behind planes of glare-free black glass. DATA•PANEL eliminates rows of ever-present individual indicators which restrict design freedom.

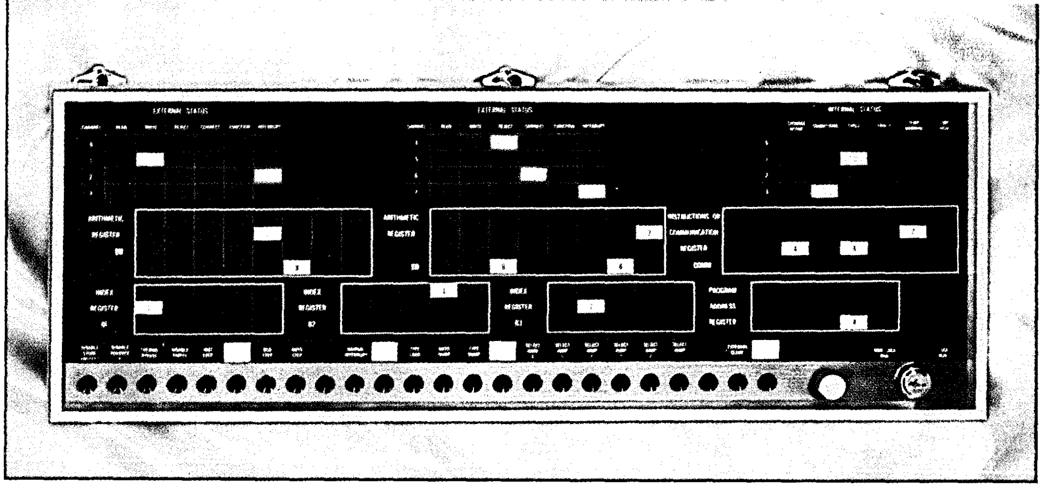
Operator accuracy increases sharply with the use of DATA•PANEL because only the important ON indications can be seen and read. OFF indications are invisible until illuminated and the operator is not confused by the presence of non-indicating indicators.

TEC-LITE DATA•PANEL is also electrically and mechanically versatile and is custom designed to fit the particular characteristics of any computing or control system. DATA•PANEL is designed and built by Transistor Electronics Corporation, originator and world's largest manufacturer of transistor controlled indicating devices. Send for 8-page brochure with design specification forms for more information on DATA•PANEL.

1 APPEARANCE—An entirely new look is given computer consoles and display panels by the distinctive and dramatic appearance of DATA•PANEL. Information is displayed with clarity, brilliance and readability never before available to designers. Indications, readouts, complete legends stand out emphatically, in full color behind smooth planes of glare-free black glass. As modern in concept as the systems it serves, DATA•PANEL is the key to clean, handsome console styling.

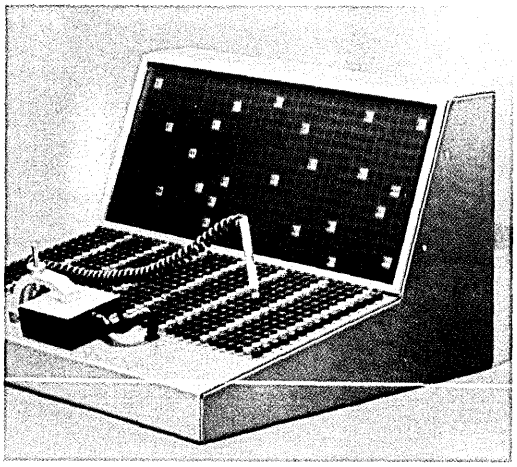
Status boards can be made in a variety of sizes using the DATA•PANEL Concept.

2 DISPLAY VERSATILITY—There are no restrictions, within practical limits, to the overall size of DATA•PANEL, or to the shape, color, size or arrangement of alpha-numeric messages, indications or digital readouts. Legends (which are photographically reproduced) may be as long or as large as required and unlike methods used with conventional indicator lights, are not limited by size of lens cap or cost of panel engraving. Each legend or symbol appears within a DATA•Module, which may be of many sizes or shapes.



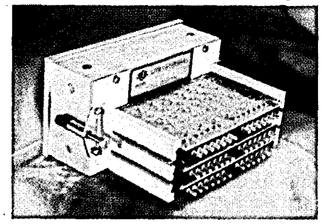
DATA•PANEL indications are visible only when illuminated. Permanently visible legends and grids are provided for operator orientation. Courtesy Control Data Corporation.

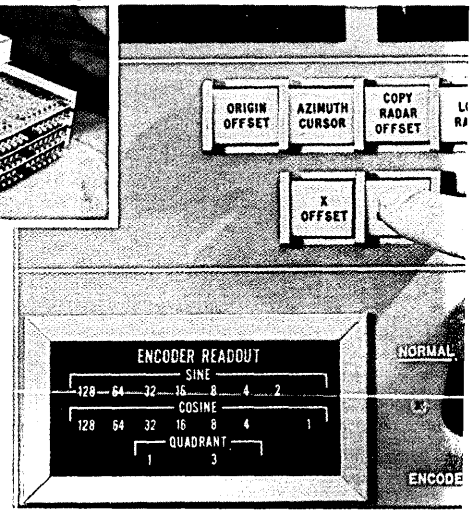
3 OPERATOR ACCURACY—
DATA•PANEL legends and indications in the OFF condition are totally invisible until illuminated. Operators, therefore, are not distracted or confused by the presence of ambiguous, non-indicating indicators, or, for example, by red lensed indicators that merely turn redder. Permanently visible grid lines and legends can be provided for visual orientation. Operators working with a visually clean panel suffer less fatigue and, consequently, make fewer errors.



DATA•PANEL, with 540 lamp display is used in Module Test Device at Litton Data Systems Division.

DATA•PANEL, 2½" high x 4½" wide, was custom designed for Litton Data Systems Division.

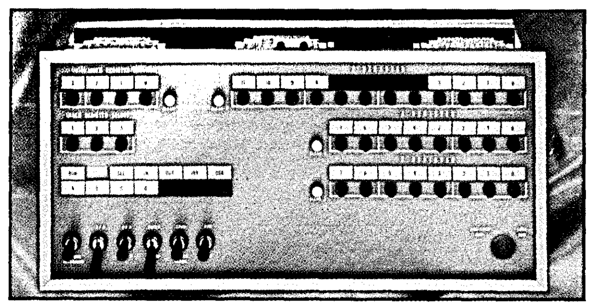




4 CUSTOM DESIGN—
Information display areas become an integral, complementary element of console styling when DATA•PANEL is used. Its extremely flexible visual and mechanical parameters give designers freedom never before available in display techniques. Built to designers' specifications, DATA•PANEL is a complete, self-contained display unit ready to mount. If you prefer, TEC-LITE DATA•PANEL industrial designers will provide console styling and display layout based on your requirements.

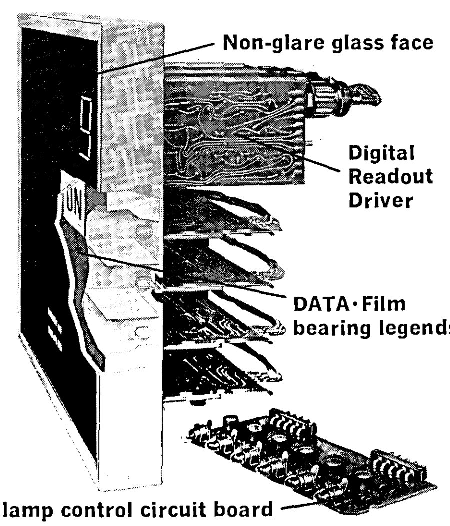
5 CUTS DISPLAY COSTS—Even though built to your design and offering unequalled appearance, DATA•PANEL usually costs less than conventional display panels using individually mounted indicators. DATA•PANEL costs less because it eliminates these costs: metal panel fabrication and finishing; hot stamping or engraving legends; mounting and wiring of many individual indicators. When transistor controlled lamps are specified, DATA•PANEL costs less because many lamp control circuits are placed on a single printed circuit board, common connections are used and simple lamp mounting devices employed. Eliminated are many expensive parts such as lenses, lamp sockets, nuts, bodies and molded terminal assemblies required for each individual-type indicator. DATA•PANEL production costs are reduced by standard tooling, hardware and component mounting techniques.

CUSTOM SWITCHES—
TEC-LITE DATA•Switches extend console design freedom to important control functions, too. They offer a variety of electrical, mechanical and visual options. Buttons may be produced in configurations appropriate to panel design. Conventional or transistor controlled indicators can be incorporated with, but isolated from, the switch to combine indication and control functions for space conservation and operator efficiency.



DATA•PANEL and switches are combined in this display-control assembly built for Control Data Corporation.

DATA•PANEL SOLVES ELECTRICAL-MECHANICAL PROBLEMS
DATA•PANEL is extremely flexible both electrically and mechanically. Replaceable incandescent or neon lamps are used and may be selected to operate from a wide range of supply voltages or controlled by solid state circuitry which is a part of the assembly. High current drain problems encountered with incandescent lamps are solved by transistorized circuitry that switches lamps ON and OFF with low current level signals usually found in solid state systems. High voltage problems inherent in neon lamps are confined to DATA•PANEL by use of self contained circuitry that operates from low level logic signals. Neon display tube, segmented, projection and other alpha-numeric readout devices can be mounted behind the glass panel as an integral part of the assembly. Mechanically, DATA•PANEL and its DATA•Modules may be of any practical size. The self-contained display panel can be mounted flush with adjacent surfaces or recessed below and at angles to surrounding surfaces. Rack mounting is also available.



TEC LITE INDICATING DEVICES

Transistor Electronics Corporation

Dept I, Box 6191 • Minneapolis, Minnesota 55424 • Phone (612)-941-1100

INDICATE 20 ON READER CARD


AUGUST 27, 1962
Mariner II interplanetary probe launched from Cape Kennedy; successful midcourse correction of orbit brings it close to Venus.

time-sharing hardware

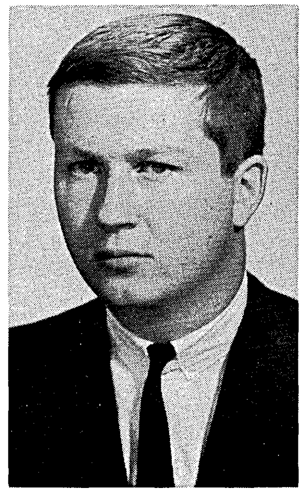
THE PDP-6

by R. P. HARRIS

The time-sharing concept embodied in Programmed Data Processor-6 evolved from more than two years' hardware and software development inside and outside Digital Equipment Corp. Inside, the PDP-1 was equipped with certain circuits and peripheral equipment to make time-sharing possible. Outside, programs and techniques were developed using the PDP-1 which demonstrated that time-sharing offered considerable promise. Some of this work is reported in the literature referenced at the end of this article.

From this experience, Digital laid down three broad criteria to be followed in designing the PDP-1. First, the computer must be accessible to a large number of users through locally and remotely located multiple user stations. The response time for the computer to service any one user must be kept as short as possible. Second, the system must be completely general purpose. The monitor and system programs must be written to be used from the multiple user stations. Third, the system must be completely expandable as users' needs increase, and the system must be able to incorporate equipment yet to be developed, such as faster memories.

The PDP-6 meets these criteria with a number of hardware and software features. Common to all is a modularity that permits PDP-6 systems to be specified for almost any application, to grow in size as requirements increase, and to be able to control new peripheral equipment as it is



Mr. Harris is an applications programmer in the sales department of Digital Equipment Corp., Maynard, Mass. He was formerly with Raytheon Co., engaged in software design for the Apollo guidance computer at MIT's Instrumentation Laboratory. He holds a BS in math from the Univ. of Massachusetts.

developed and made available. Multiple processor capability is designed into the system. Asynchronous operation of processors, memories, and I/O equipment is the prime reason for this modularity.

Basic specifications of PDP-6 hardware are: 36-bit word length, core memories up to 262,144 words directly addressable, 363 instructions, and up to 128 input-output devices. The arithmetic processor has 16 accumulators, 15 of which double as index registers.

From the point of view of the user at an input-output typewriter station, it is primarily the monitor that makes the time-sharing system effective. Because the monitor handles all scheduling and data manipulation problems, several users can be on-line with the computer, editing, debugging, and controlling I/O conversions. They can assemble, compile, and execute programs under direct control. Traditional batch processing can be run concurrently with these user activities, where the batch is treated as a non-reactive user.

For the purpose of this article, the memory can be thought of as divided into three distinct sections (see Fig. 1): I/O Service, Users Area, and Executive. These will be described in order.

input-output interrupt service

The effectiveness of the time-sharing system depends largely on the efficiency of the I/O interrupt service. To implement it properly, two hardware features were incorporated into the PDP-6.

1. A multiple-level priority interrupt system that can be program-assigned.
2. A fully buffered I/O system to allow the central processor to service I/O at a rate that approaches memory speeds.

The interrupt service is completely modular to allow a small system to carry only those routines directly related to its peripheral equipment. Thus the memory capacity of a small system is not over-taxed, yet expansion of the system is not restricted. As more I/O is added, the proper service routines are added into the system and buffer areas are extended.

The service is implemented in such a way that the programs are able to receive and transmit a sufficiently large quantum of data in a format common to all I/O equipment. Therefore, I/O service relieves the programmer of the burden of format translations and also insures that when a user receives or transmits a quantum of data, it will be large enough to be useful.

multiple user stations

The multiple user station interrupt service is an interesting example of how these features facilitate system operation. A data communications system was developed to multiplex up to 64 remote user stations into the I/O system. All stations interrupt on the same channel, use the same service routine, and provide the processor with the character transmitted and the station identification.

As interrupts are received, the I/O service routine first looks to see if a buffer area has been assigned to the user station causing the interrupt. If no buffer has been assigned and there is one available, the station is assigned the open buffer by means of a word pointer. If all available buffers are being used, the operator of the station is appropriately notified.

The normal quantum of data to be handled in the interrupt service buffers is a line terminated by a carriage return (or up to 75 characters). (It will be seen later, however, that multiple lines may be received or transmitted by the user program if desired). Therefore, characters are received or transmitted until the station buffer is empty, full or terminated by an appropriate character. When a buffer becomes empty in output mode, the I/O service sees if the user program has more data it wishes to transmit. This is determined by the user program specifying either continuous operation, in which all full buffers are serviced, or non-continuous, in which only one buffer is serviced.

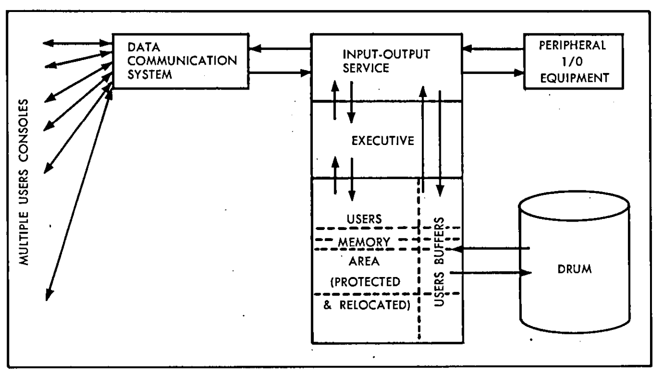
When a buffer becomes full or terminated in input mode, the I/O service transfers the buffer into the buffer area of the user program that requested the data. If the data has not been requested by any user program, it is transferred to the command decoder in the monitor.

user area in memory

The user area composes the largest bulk of memory and is under complete control of the monitor. Since many user programs will not have been debugged, several safeguards are used to insure there will be no unwanted user-user or user-monitor interaction, including (1) trying to halt the processor, (2) making memory references outside a monitor-defined area in memory, (3) trying to control or influence the I/O or priority interrupt system outside monitor control. For convenience and flexibility, however, the user must be able to input, output, control, and select peripheral equipment through the monitor. The user must be able to load several of his programs together with symbols, having them linked and relocated to fit properly into his assigned area.

The solutions to these user problems are built into PDP-6 hardware and further implemented in the monitor. When control is transferred into the user area, the user mode is entered, activating memory protect and relocation logic and returning control to the monitor should any of the illegal operations mentioned above be attempted.

Fig. 1 PDP-6 Hardware Software Interactions



The memory protect (MP) relocation (RL) registers function in the following manner. The user area is initially defined as starting at location 0 and ending at address N x 1024-1 (where N is a non-zero integer). The available area in memory where the program is to be loaded starts at M x 1024 (where M is a non-zero integer) and must extend N x 1024 contiguous registers. The monitor loads register MP with N and register RL with M. In the user mode, the following steps are carried out automatically by PDP-6 logic. For each memory reference, the high-order eight bits of the memory address are compared to the MP register. If the reference is greater than N, the memory protection flag is set and control is trapped to the monitor. In parallel with this function, the RL register is added to the high-order eight bits of the memory address register and used to select the proper location in memory. (The 16 accumulator locations are exceptions to this relocation).

With this hardware it is seen that programs can be

loaded into any available area in memory and can be from one to 255 blocks of 1024 words. While they are running it becomes impossible to reference any memory block lower than the one specified in RL; and any attempt to reference any memory block higher than the one specified in MP will be trapped and control transferred to the appropriate error sequence in the monitor.

Other circuits trap attempts to (1) execute a HALT, (2) leave the user mode, (3) influence the interrupt system, and (4) execute an instruction that interacts with the I/O system. All of these conditions trap to the monitor and are dispatched to the proper error routine and an appropriate diagnostic printed at the user console.

input-output control

There are 64 operation codes in the PDP-6 that are not decoded as instructions. These programmed operators trap to the monitor for dispatching. Thirty-two of these are left to the programmer to define additional software-implemented instructions. The other 32 become the user-monitor link for I/O control.

At load time, the user declares to the monitor the number of 1024-word blocks of storage required for his program. This must include sufficient memory to buffer his I/O data. He then uses programmed operators to initiate I/O transfers to and from his buffer area.

Four commonly used programmed operators are:

1. INITIALIZE — This performs several functions. The user includes with the command a device name, a device channel, a buffer area, and a set of condition bits. If the device is not available, control returns to the location following the call. If available, a set of ring buffers is set up in the area defined by the user. The device is assigned to the user and is initialized according to the conditions specified by the user. Control is then returned to the requesting program.
2. INPUT — Input is commenced from the specified device into the current buffer of the user's buffer area. If a "wait" request is made, control will not return to the user program until at least one buffer has been filled; otherwise control is returned immediately. If the user specifies "continuous operation," all contiguous empty buffers will be filled before input is terminated. If the input is not to be "continuous," only one buffer will be filled and input will terminate.
3. OUTPUT — Output is commenced on the specified device from the current buffer of the user's buffer area. The "wait" and "continuous" conditions also apply with this command.
4. RETURN — Return the specified device to the system.

The user may therefore control the I/O portion of his program, specify the amounts of data to be received or sent with one command, and may stay on queue to do further processing, or release time to another user while waiting for input or output.

executive

The executive is a collection of programs remaining permanently in memory to provide overall coordination and control of the total operating system. It must allocate facilities, handle all commands addressed to the system from the user console, and coordinate I/O between the user and the I/O service. Its principal function is to schedule all jobs to be processed from the run queue. Once a user program has been given control, it can be removed from run status by any of the following means:

1. The program is completed and does not require further processing.
2. The program has run for its allotted amount of time and must be re-scheduled.
3. The program has issued an I/O command and issues a "wait" request.
4. A program of higher priority must be run.

As a program is removed from run status, the executive must transfer control to another program as quickly as possible. In large memory, several programs can be waiting on the run queue and available immediately. If there is a large number of active users, a high speed drum provides secondary storage. Exchanges between drum and memory are made in parallel with processing. A direct access data control permits drum-to-memory transfers concurrent with the arithmetic processing.

The scheduling algorithm, to a large extent, determines the response time any one user will encounter when interacting with the machine at a user console. The most crucial parameter here is possibly the incremental run time — that is, the selected length of time during which program is allowed to run before being removed from run status. Obviously, as the time between switching from program to program becomes shorter, the time for the computer to service any one user decreases. However, the overhead time consumed by the monitor increases as a result. In the PDP-6, incremental run time is set at 100 ms, which is roughly equivalent to the maximum operating cycle of the user keyboards. With this type of response, the user would hardly ever perceive the difference between having the processor to himself and sharing it with several other users. This increment is variable, to be set for maximum system efficiency depending on system configuration, use, number of stations, etc.

At present, programs are scheduled in round robin fashion with programs returning from "I/O wait" requests given highest priority. In future systems, the scheduling algorithm will take into account such variables as: the length of time the program has to run before being completed or I/O bound; the time elapsed since it was last on run queue; and the priority associated with each program. With parameters such as these, the scheduling program will decide in what order to put programs on run status.

The first PDP-6 time-sharing system is operational now at Digital Equipment Corp. in Maynard, Mass. It consists of an arithmetic processor, 16,384 words of core memory, a magnetic tape system, line printer, card reader, perforated tape reader and punch, and a data communications system with four multiple user consoles. Using this system, larger and more sophisticated time-sharing systems are being developed in a time-shared environment.

The first PDP-6 time-sharing system produced for sale is scheduled for installation at the University of Western Australia in January, 1965. ■

REFERENCES

"A Time-Sharing Debugging System for a Small Computer," J. McCarthy, S. Boilen, E. Fredkin, and J. C. R. Licklider. *AFIPS Conference Proceedings*, Vol. 23, 1963 Spring Joint Computer Conference.

"The Time Sharing of Computers," Edward Fredkin. *Computers and Automation*, Nov. 1963.

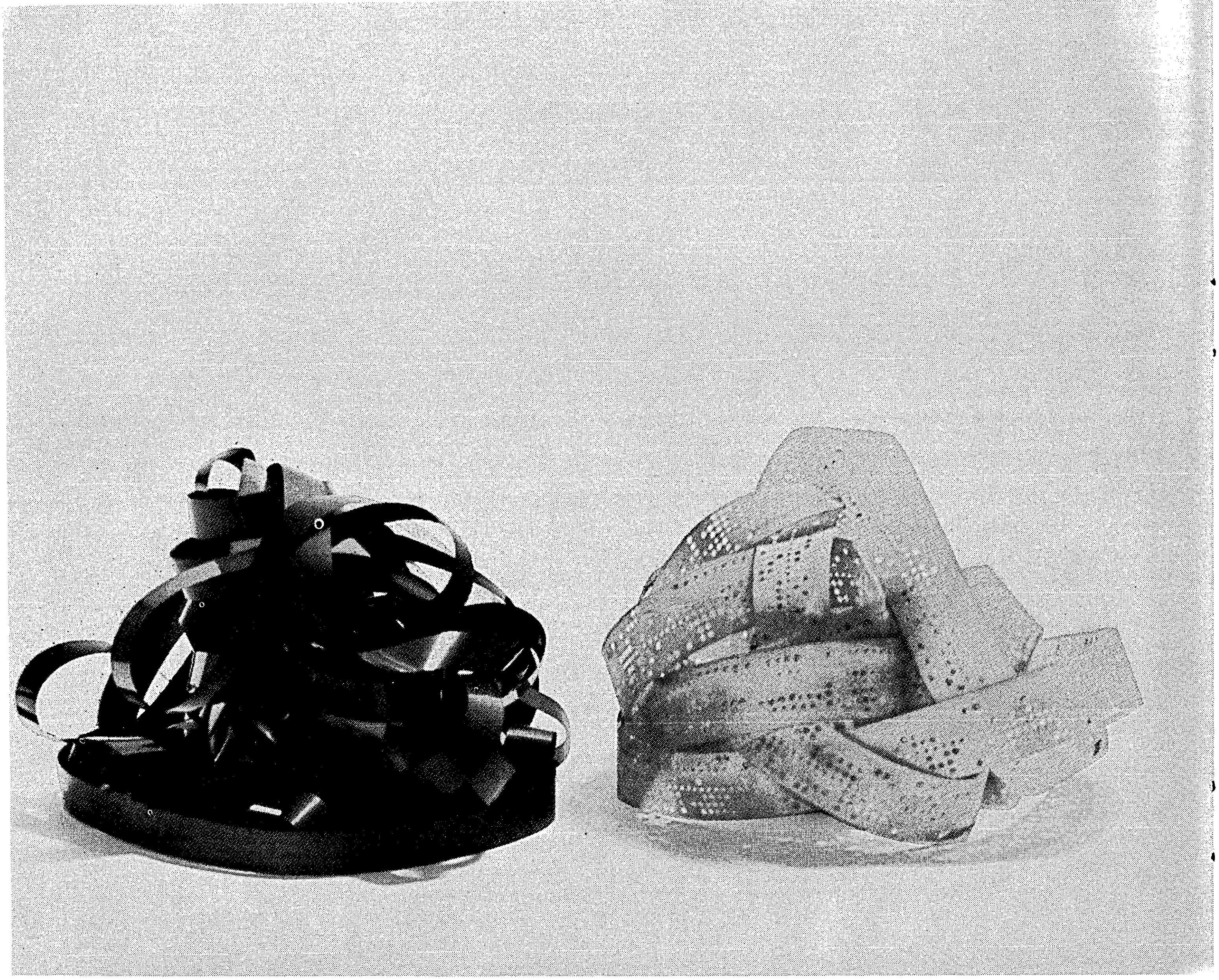
"Why Multi-Computers?" Walter F. Bauer. *Datamation*, September, 1962.

"Generalized Multiprocessing and Multiprogramming Systems," A. J. Critchlow. *AFIPS Conference Proceedings*, Vol. 24, 1963 Fall Joint Computer Conference.

"Time Sharing on the Ferranti-Packard FP6000 Computer System," M. J. Marcotty, F. F. Longstaff and Audrey P. M. Williams. *AFIPS Conference Proceedings* Vol. 23, 1963 Spring Joint Computer Conference.

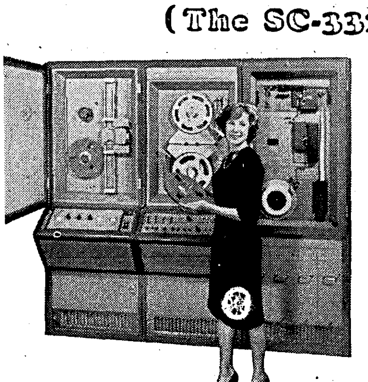
"The Compatible Time-Sharing System," *A Programmer's Guide*, The M.I.T. Computation Center.

"A Multiuser Computation Facility for Education and Research," Jack B. Dennis, Massachusetts Institute of Technology, Cambridge, Mass.



Why tie up expensive computers to duplicate or convert tapes?

(The SC-332A does it "off-line" faster, more economically.)



Many progressive users of data processing equipment find that converting magnetic tape to paper tape (and vice versa) or duplicating paper tapes can now be done far more efficiently with the GD/E SC-332A High Speed Bi-Directional Data Converter. You don't tie up expensive computer time—and its punch speeds are twice as ast as computers. Examples of its applications include:

Numerically controlled Machine Tools—The SC-332A precisely ranslates data on magnetic tapes to paper tapes used on umerically controlled machine tools. Permits adding or deleting special characters during code conversion.

Data Handling Systems for invoice preparation or inventory control—The SC-332A translates data received from remote locations on punched paper tape to magnetic tape for immediate use in computers.

Business Machines—The SC-332A accepts any punched tape from any peripheral data collection equipment and translates it for direct use in computers.

Speeds—Paper Tape punch speeds—280 characters/second average. Paper Tape reading speeds—900 characters/second average. 200, 556, 800 bit/inch input-output magnetic tape densities (selectable via patchboard).

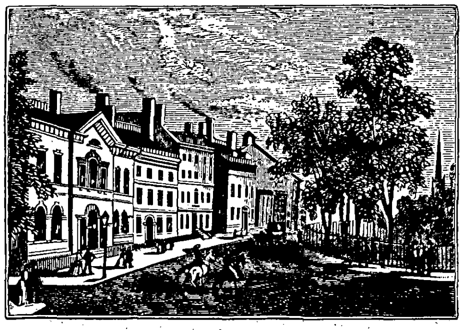
Send for complete operational data. Call (716) 342-8000 or write: Product Manager, Data Equipment, General Dynamics/Electronics, 1442 No. Goodman St., Rochester, N. Y. 14601.

GD

GENERAL DYNAMICS | ELECTRONICS

CIRCLE 27 ON READER CARD

part 2 . . . THE ADVENTURE OF THE MISSING BIT



by B. CONAN DOYLIE

Holmes watched from the window as our visitor took cab in the street below 221-B Baker Street. Holmes had filled his pipe with shag tobacco from the Persian slipper and the wreaths of acrid smoke swirled about his head as he stood watching the street below. As the cab bearing Dr. Charles Babbage rounded the corner, Holmes put his pipe down and sprang into action.

"Come, Watson," he said, "we yet have time to arrive at Crumbley-under-Lyme simultaneously with Dr. Babbage."

"But Holmes, I thought you were going to work here for the next several days, considering the case, and all that."

"Yes, indeed, Watson, but it is just as well that Dr. Babbage not know we are following him so that, should anyone else also choose to follow, he will also remain in the dark."

"You think the doctor is in danger, then?"

"Not in danger possibly, but there are deep waters here, Watson. Someone manifestly is interested in having the good doctor fail in his efforts. The reasons may be devious indeed."

And Holmes, cape flying about his thin shoulders was already halfway down the stair as I ran to keep up with him.

In Charing Cross Station, as we waited for our train, Holmes pulled me suddenly behind a pillar.

"Look there, Watson," he whispered, "one of our principals has already put in an appearance."

Across the station, headed for the same train which Dr. Charles Babbage had just boarded, strode the impressive, white-mustachioed figure of Sir Henry Glitch, KCB, special assistant secretary to the Chancellor of the Exchequer. In his hand Sir Henry carried a mysterious-looking black bag.

Sherlock Holmes waited until Sir Henry had boarded. Then he chose a compartment two cars to the rear of that chosen by the distinguished public servant. Holmes settled into his corner, selecting a long Manila cigar with which he soon succeeded in filling the compartment with drifting blue fog.

"Well, Watson, what do you make of it?" Holmes asked after an hour or so, gazing at me through half-closed eyes.

"I must say, Holmes, it seems a trifle confusing to me. I can't see why anyone would want to harm a machine. It seems like a strange way to injure Dr. Babbage. Further, I must confess I can't see why anyone would want to injure the good doctor. He seems like an excellent chap, surely not the sort to make enemies in this world."

"And yet, Watson, you must admit, there is a certain element of the bizarre about it. The room containing the logical machinery is always locked during testing of the Engine. The malfunctioning 'bit' only makes itself evident when Dr. Babbage is in the far laboratory. Three people have keys to the locked room, and all are, in the doctor's opinion, apparently to be trusted. And what do you make of the obvious hostility of Sir Henry who is charged with dispensing the funds so that Dr. Babbage can continue his work?"

"Inborn conservatism of a public servant charged with dispensing Her Majesty's funds, I would say. Men like Sir Henry are seldom known for their progressive outlook on matters of applied science."

"Yes, possibly, Watson, possibly, but surely we are slowing down for the station at Crumbley-under-Lyme. I suggest you remain here in the compartment with me until the good Dr. Babbage and the distinguished Sir Henry have both cleared the platform."

Holmes and I took rooms at the White Stallion Inn, having observed that Sir Henry Glitch had elected to stay at the Great Crumbley Hotel some distance up Cobolbury Street. After lunch we took cab to the laboratory of Dr. Babbage, just behind the Great Crumbley School. Holmes strode to the grounds in back of the building.

"Always an excellent idea to get the lay of the land, Watson," Holmes noted, kicking along through the mustard weeds which grew in back of the laboratory building. "You will note there the great Newcomen engine which undoubtedly powers Dr. Babbage's calculating machine. Observe how the belt runs jerkily through the slots in the wall. However, I sincerely doubt that a person could crawl through those slots in order to gain access to the building. All the windows are high above ground, and each seems to be securely barred by a steel grating. Further, we find no evidence of trampling of the weeds. I would say that no entrance to the laboratory has been gained from this quarter. Let us call on our friend."

Dr. Charles Babbage was most surprised when Holmes and I entered his office.

"Why, Mr. Holmes," said he, "this is indeed a pleasant and unforeseen visit. I had thought you had little interest in my problems."

"After due reflection, Doctor, it occurred to me that I could postpone my other work, since it is not of a pressing nature. Further, Watson and I can do with a day or so out here in the countryside. We spend entirely too much time in the confines of our great metropolis. But now, may we see your Calculating Engine? I am most anxious to know what such a remarkable device may look like."

"Why yes, indeed. In fact, I have just finished showing it to Sir Henry Glitch who was just here. He also surprised me with a visit."

"How interesting! And what did Sir Henry think of it?"

"Sir Henry is not pleased with my progress, I am afraid. He stated that he had evidence that my machine did not and would not work, and that he would personally recommend that no more funds be allocated toward its completion. I'm afraid this is the end of the road for my ideas, Mr. Holmes. Sir Henry was quite firm."

"Well, meanwhile," said Holmes, "let us observe it to see if we can note anything out of the ordinary."

"Very well, Mr. Holmes, if you will follow me into the room next door here, I have the console set now for operation."

Holmes and I followed Dr. Babbage into the controlling laboratory of the Computing Engine. Dr. Babbage sat in a wooden chair before a great array of levers. He selected one of a stack of cards, all of which seemed to be perforated with holes, and placed it into a device which somewhat resembled a cider press. He nodded to a young man standing beside the console. The young man pulled several of the levers.

"This is our technical man, Mr. Hollerith Powers," said Dr. Babbage. "Mr. Powers, Mr. Sherlock Holmes, and his friend, Dr. Watson."

"A pleasure, I'm sure," said Powers, continuing to work the levers.

"Where are your other associates?" asked Holmes.

"Miss Groper was in the office at the front of the building as you came in," said Dr. Babbage. "Dr. Eckley-Mauchert has gone up to the village for some nails with which to repair a part of the storage bank of the Engine.

"Now, Mr. Holmes, you will observe that by placing this card in the machine I have instructed it to seek the solution to the square of the hypotenuse of a sixty-degree right triangle with dimensions given for the two sides. There the indicator wheels are turning. There the counting ratchets are latching in order."

Through the laboratory wall we could hear faintly the wheezing of the great Newcomen steam engine as it drove Dr. Babbage's calculating machine.

"The first answer is correct," said Dr. Babbage.

"And the logical section of the machine?" asked Holmes.

"The laboratory doors are locked," Babbage said. "The logical control seems to be functioning perfectly."

"And no one is in the room?" Holmes queried.

"No one. I will try the second problem now."

Dr. Babbage proceeded with another large card in the cider press.

"Oh, bless my soul!" he said, suddenly. "There it is again. The same difficulty. Bit sub-four is missing from the control pattern."

Holmes gazed with intensity at the console.

"Tell me, Dr. Babbage," he said, if someone were in the logical control room, could he detect the fact that you are now operating the machine?"

"Why, yes, he could, Mr. Holmes," said Dr. Babbage.

"Will you trust me with your key?" Holmes asked.

"Why yes, of course."

"Then, my good sir, I will ask you to continue to sit here, operating the machine in normal fashion. Dr. Watson and I will perform an inspection of the laboratory containing the logical unit, with your permission."

Holmes stepped softly and quickly to the west door of the center laboratory. I observed that he had taken his Eley No. 2 revolver from his pocket. He held it in his right hand and turned the key in the latch of the laboratory door silently with his left. Suddenly Sherlock Holmes kicked open the door of the laboratory and stepped swiftly inside.

Over Holmes' shoulder I could discern that the laboratory was dark. Holmes took a small torch lantern from his pocket, lit it, and shined it about the room. The massive machinery was clanking regularly in the darkness. The pans of calcium chloride stood on tables here and there. Holmes aimed his torch along the floor.

"There, Watson!" he said.

The torch beam had illuminated a dark object sprawled on the floor alongside the logical device. There, in a pool of blood, his head crushed like an egg, lay Sir Henry Glitch.

After lighting the gas in the room, Holmes bent over the body of Sir Henry. The mysterious black bag lay beside the body.

"What do you make of this, Watson?" Holmes said.

"He's quite dead, in my opinion."

"Precisely, but I was referring to the insects. Did you see them scurry along the floor when I lit the gas? Since I presume that Sir Henry was not in the habit of transporting insects in his black bag, they must live here in the room, but hello! What's this?"

Holmes was bent over, scraping something from the floor with his penknife blade. The material looked like metallic powder. Holmes placed the silvery dust in a small envelope and put the envelope in his pocket. Then he began a careful inspection of the logical machine which was still operating. The bistable levers moved regularly up and down under the guiding of the pawls actuated from the master shaft which turned under power of the large belt from the Newcomen engine outside the wall.

Then Holmes carefully opened the black bag of the late Sir Henry Glitch. Inside was a sheaf of papers with the title sheet reading, "Report on the Utter Lack of Feasibility of the Project of Dr. C. Babbage." In the bottom of the bag was a key. Holmes compared it to the key given him by Dr. Babbage which fitted the laboratory door. The match was perfect.

"At any rate," Holmes observed, "we now know how Sir Henry got in here. I presume, Watson, you should notify our friends of the sad event of Sir Henry's death."

When the local constable learned of the importance of the late victim's position, he immediately notified Scotland Yard and Inspector Lestrade was dispatched up to Crumbley-under-Lyme on the next train. Meanwhile, the constable placed Dr. Babbage under arrest, charging him with suspicion of the murder of Sir Henry Glitch. Since Dr. Eckley-Mauchert was found in Sir Henry's room at the Great Crumbley Hotel, rather than purchasing nails at the ironmonger's, he too was held as an accessory. Miss Groper and Mr. Powers were questioned and released.

The following day Sherlock Holmes and I met Inspector Lestrade at the Great Crumbley Hotel for tea.

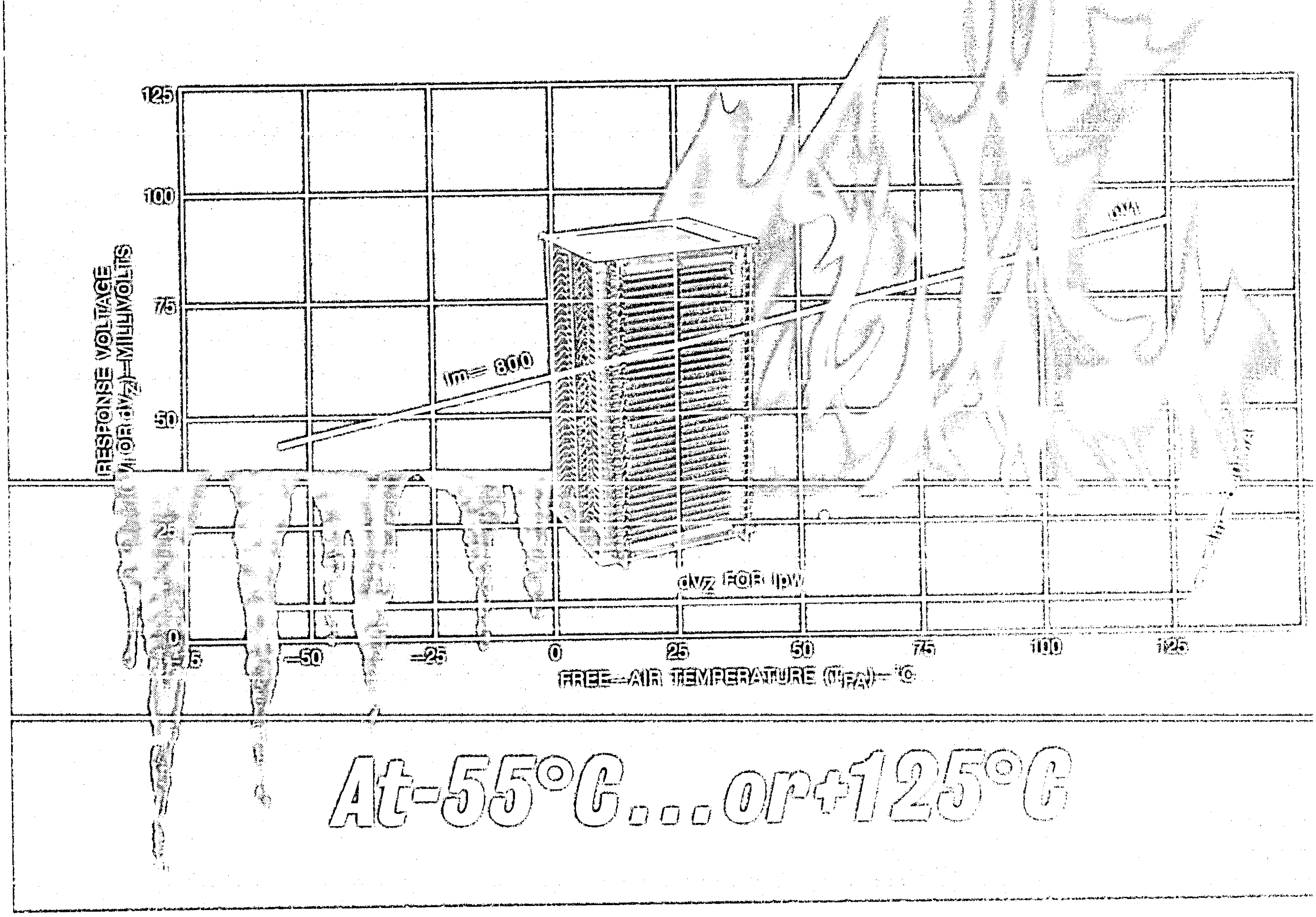
"Well, my dear Inspector," Holmes began, "am I to presume you have completed your investigation of the case?"

"Yes, Mr. Holmes," the Inspector replied, "I think the police will have no need of your fancy theories in this instance. The professor gentleman murdered Sir Henry for obvious reasons. Sir Henry was about to expose him as a quack so Baggage, or what's his name, did him in."

"And the murder weapon, Inspector? Have you found it yet?"

"I dare say we'll find it around the place somewhere. If

NOW YOU CAN ELIMINATE TEMPERATURE CONTROLS FROM FERRITE MEMORIES...



At-55°C...or+125°C

RCA high-speed, wide-temperature-range ferrite memory planes and stacks operate without temperature controls or current compensation over any 100 centigrade degree range between the limits of −55°C and +125°C.

Memory planes and stacks using RCA 0146M5 ferrite cores provide high-speed operation over the full temperature range with less than 480 ma partial-write current. At full driving current of 800 ma, these temperature-stable cores have a switching time of approximately 0.5 μsec, making them particularly suitable for memory systems with 2 μsec operating cycles.

RCA wide-temperature-range cores are also available for low-drive operation at slower speeds.

The full line of RCA cores, planes and stacks offers a wide variety of types for coincident-current, word-address, and switching applications at speeds to 250 nsec. Complete RCA memory systems range in size from 256 words x 8 bits to 16,384 words x 20 bits and offer speeds to 250 nsec.

Whatever your memory requirements, call your RCA Representative for a coordinated application service for RCA memory components and devices, ranging from cores and transistors to complete memory systems. For technical data, write: RCA Electronic Components and Devices, Memory Products Operation, Section F D-11, 64 "A" Street, Needham Heights 94, Mass.

The Most Trusted Name in Electronics

See RCA High-Speed Cores and Microferrites at SJCC... Booths 21-22

CIRCLE 28 ON READER CARD

you've a mind to bash heads, most any old thing will do," Lestrade said.

"But how do you account that Watson and I were with Dr. Babbage from the time he began to operate his machine until we found Sir Henry's remains upon the laboratory floor?"

"That's an easy one," Lestrade countered, "he was done in before you and Dr. Watson got there. The professor just invited Sir Henry in to look at his infernal machine, then bashed him one, just like that."

"I must confess, Lestrade, that the motive seems to belong to Dr. Babbage. There is little doubt that his project funds would have been cut off had Sir Henry had his way. However, there are a few minor matters that I would like to see cleared up. I wonder if you would be so kind, on my personal promise, as to release Dr. Babbage and his associate, Dr. Eckley-Mauchert, from the local gaol so that we might all gather in the laboratory this evening. It seems to me that one or two pertinent facts might possibly be demonstrated."

"Well, it's against the regulations, Mr. Holmes, but since you have been kind enough to help the police once or twice, I suppose I might do it."

"I shall be grateful," said Sherlock Holmes.

At half-past seven that evening we were all gathered in the laboratory under the watchful eye of Inspector Lestrade. Dr. Babbage seemed quite depressed, but, at Holmes' request, he turned on the Computing Engine and set the Pythagorean Theorem problem into motion. Following Holmes, we all went into the logical device room. The bistable levers were in motion, driven by the great belt from the Newcomen engine.

"Now, if I may," said Sherlock Holmes, "I would like to clear up one or two preliminary matters. You will note, Inspector Lestrade, that there are small metallic filings ground into the floor here alongside this section of the machine. Now, Dr. Eckley-Mauchert, if you will be so kind as to sit on that chair there and lift up your right foot, I would like the Inspector to examine the sole of your shoe."

The assistant to Dr. Babbage grudgingly complied.

"Well, I'll be, Mr. Holmes," said Lestrade. "The filings are ground into the sole of the shoe."

"You will find, I think, Inspector, that Sir Henry's shoes also show the same evidence."

"So," said Lestrade, addressing Dr. Eckley-Mauchert, "it was you that did Sir Henry in."

"I swear to God, no!" said Dr. Eckley-Mauchert.

"Why were you in Sir Henry's hotel room?" Holmes queried.

"All right, I admit that I was in league with Sir Henry. I was convinced as he was that Dr. Babbage was on the wrong track. I often stood with him here in the laboratory to point out design flaws. I went to the Hotel to give him my key so he could get in yesterday. I wanted the contract for myself. But I would have no purpose in harming him. Sir Henry had promised the contract to me when he cut Dr. Babbage off."

"Indeed," said Holmes. "Yes, I think we may believe him, Inspector. Dr. Eckley-Mauchert is guilty of some disloyalty, perhaps, but I think not guilty of murder. However, we will find the culprit here in this room."

"You sound quite sure of yourself, Mr. Holmes," Lestrade observed.

"There are, interestingly enough, two problems," said Holmes, choosing to ignore Lestrade's remark. "Let us consider the first one, since, in some ways it is the most interesting. Let me ask you, Watson, to extinguish the gas lights here in the laboratory. I am correct, am I not, Dr. Babbage, in presuming that this room was always kept dark unless it were occupied?"

"That is right," said Dr. Babbage, "I wanted to save gas, since it is expensive."

"Now," said Holmes, "if we may all be patient for a time."

After some five minutes during which we sat in total darkness, Sherlock Holmes lit his torch lantern. He shined it on the logical machine, aiming at the intermittently-failing bistable lever. A large cockroach perched on one end of the lever. It scurried for cover at sight of the light. "You may turn the gas on, Watson," Holmes said, "we have seen our culprit. You will note, Dr. Babbage, that while this particular logical lever seems to function perfectly, rocking readily back and forth on its fulcrum, the weight of a single cockroach was sufficient to hold it in the nought position. You may further observe that the same fine metal chips which were found on the floor are present here around this lever. A brief examination of them was sufficient to convince me that they were lead. Apparently some flaw in the mounting hole of the lever has caused the bird shot within it to grind away slowly and extrude onto the floor. It was this substance we found on the shoes of Dr. Eckley-Mauchert and the late Sir Henry. This logic lever no longer contains bird shot, and, hence, has lost its positive bistability. The weight of a cockroach was sufficient to cause failure in its action."

"But how does that point out the murderer?" Lestrade queried.

"It does not," said Holmes, "but it does solve the mystery of the failure in the logical section of the Calculating Engine.

"However, as I indicated previously, the murderer does stand in this room," Holmes said. "In fact, he stands there!"

With a sweep of his hand Holmes indicated the Calculating Engine.

"Mr. Holmes, you mean . . . ?" Dr. Babbage choked.

"Precisely," said Holmes. "See here! Watson, if you would be good enough to hold the light so that Inspector Lestrade can observe carefully. You will note, Inspector, that there are fibers adhering to this end of the two-to-the-twenty-third power logic bar, I believe it is. There is also a stain here which, I believe, you will find to be human blood; Sir Henry's blood, in fact. The fibers are human hair unless I miss my guess."

"Then, Mr. Holmes, you think Sir Henry stuck his head in the machine."

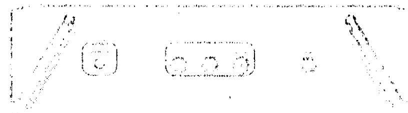
"There is no doubt of it, Inspector. Sir Henry returned here looking for more evidence to use against Dr. Babbage. He was so injudicious as to stand with his head too close to the logic bars. You will note that the actuating pawls, driven by the great Newcomen engine outside the wall, trip the levers with considerable vigor, imparting to them velocity sufficient to produce appreciable impact in view of the mass of the shot-loaded lever. Sir Henry was killed by the number twenty three flip-flop!"

"It seems to me the Doctor here owes you a debt of gratitude, Mr. Holmes," Lestrade said with some humility.

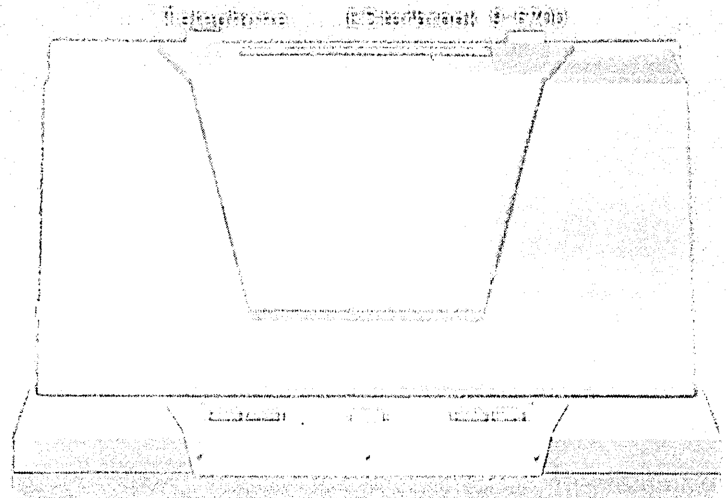
"Not at all," said Holmes. "It has been a most fascinating experience. I rather fancy, indeed, that there is a bit of poetic justice in it. Sir Henry Glitch set out to destroy this machine. Instead, the machine destroyed Sir Henry. *Sic eunt fata hominum.*"

Holmes took a cigar from his pocket and lit it.

"And now," said Sherlock Holmes, "I believe Watson and I have just time to catch the last train back to London. I note in the local papers that a rather mysterious series of events has attended the recent docking of the Dutch Steamship Friesland, and I suspect that Watson and I might well spend an interesting morrow on the banks of the Thames. We will, therefore, bid you all good night." ■



Military Electronics Division MOTOROLA



world report

DATAMATION®

honeywell/scotland delivers first mainframe

Bagpipes skirled and champagne flowed last month at Honeywell Controls Ltd., Newhouse, Scotland, as the first computer of a long line of H-200's and 400's was completed and shipped three months ahead of schedule. Final testing came 10 months after the firm established its manufacturing facility, geared to produce eight mag tape systems per month. To date, Honeywell admits to 50 computer sales (18 already installed) in the U.K.

new computer from france

Bull announced its new Gamma M40 scientific computer at the SICOB Exhibition in Paris last month. It will be marketed in the U.K. by De La Rue Bull Machines Ltd. When a De La Rue executive was asked if the new machine was related to the Gamma 40 (an RCA machine marketed by Bull), he replied, "No, the Gamma M40 is pure Bull."

germans move toward an equilibrium

German computer maker, Zuse A.G. has found the cost of keeping up with the Joneses too expensive, and has been sold to Brown-Boveri. Telefunken reportedly was among the bidders for Zuse which grossed $5-million last year.

french firms merge; have u. s. contacts

Two of France's oldest and largest electrical and electronics manufacturers, CGE (Compagnie Generale d'Electricite) and CSF (Compagnie de Telegraphic sans Fil), have merged their computer interests into a holding company to be owned 50-50.

CGE has a subsidiary, CECIS, which is licensed to sell and manufacture Scientific Data Systems gear. Although none is made abroad, 14 SDS 910-20 systems have been sold, three 9300's too (third of the total). CSF's interest is through its joint ownership with Bunker-Ramo of CAE, which markets its own (all French) medium-scale scientific computer, the CAE 510. Since announcing the system in October '63, they've received orders for 60, have delivered 16.

installations in japan on the up & up

By recent tally, over a thousand computers have been installed in Japan: 41% is reportedly IBM; domestic models, 34%; Univac, 20%. It is predicted that next year, Made in Japan computers will outnumber imports.

& odd bits

IBM is currently booked for 200 of its 360's in the U.K., is promising first delivery early in '65 . . . One of Britain's two largest companies, Imperial Chemical Industries Ltd. is standardizing on the 360, and is rumored to have ordered the first dozen systems. Buying U.S.A. won't win them friends.

Feel that muscle...and see why our system is the hottest contender for the title in the bout between the two leading low cost source data automation systems!

This is how the two systems shape up!

SCM TYPETRONIC		SYSTEM "X"
30 CPS	Reading Speed	10 CPS
30 CPS	Punching Speed	10 CPS
10 CPS	Printing Speed	10 CPS
X + 40%	Throughput Speed	X
Yes	Buffered*	No
Yes	Solid State Circuitry	No
Low	Noise Level	High
Complete	Modular Construction	Limited
Yes	Variable Carbon Control	No
Today's	Styling	Yesterday's
Electronic	Computations	Electro-mechanical
Photo-Electric	Type of Reader	Mechanical

*operates during the carriage motion

SCM TYPETRONIC® 2816 AUTOMATED DATA SYSTEM

SCM TYPETRONIC® 7816 AUTOMATED DATA COMPUTING SYSTEM

For a ring-side seat to see the **better** system, send the coupon back today!

SCM CORPORATION
DATA PROCESSING SYSTEMS
410 Park Avenue, New York, N. Y. 10022

Gentlemen:

I would like to learn more about the features of SCM TYPETRONIC SYSTEMS and their applications—no cost or obligation on my part, of course.

Name ....

Title ....

Company ....

Address ....

City .... State .... Zip Code ....

5

people IN DATAMATION

■ Paul Hachigian has been elected vp of the Data and Information Systems Div., ITT, Paramus, N.J. He is also chief engineer.

■ Arnold D. Palley has been named vp of Brandon Applied Systems, Inc., New York City. He was formerly vp with Griffenhagen & Associates.

■ Milton Sanders, vp, was named general manager for Bunker-Ramo Stamford Operations, Stamford, Conn. He joined the company in 1958.

■ Charles W. Johnson has joined Planning Research Corp. of Los Angeles as a senior associate in the firm's Information Systems Div. A specialist in design and specification of display equipment, he was previously with IBM.

■ Dr. Julius T. Tou, formerly head of the Computer Sciences Lab at Northwestern Univ., and currently adjunct professor of EE at Ohio State, has joined Battelle Memorial Institute as a senior scientist in information and computer sciences research.

■ Edward O. Boutwell has joined Amdahl Associates, Northridge, Calif., a computer consulting firm. He was formerly with Packard Bell Computer Div.

■ Lane L. Wolman has been named manager of engineering for Ampex Corp.'s computer products div., Culver City, Calif. He was previously director of dp system development for Librascope.

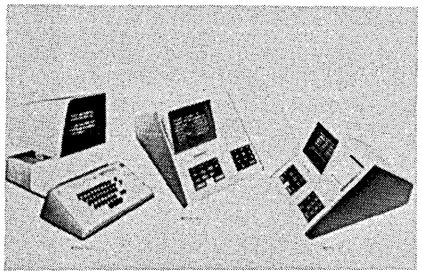
■ Herbert S. White has been appointed director of the scientific and technical information facility which Documentation Inc., Bethesda, Md., operates for NASA. He was formerly with IBM.

■ A newly created post, director of systems engineering at Digitronics Corp., has been filled by the appointment of Morton Siegelbaum. He has served in various capacities since the founding of the company in 1957.

NEW PRODUCTS

display consoles

Models 211 and 212 are terminal devices for use in office and reasonable factory environments, have key-



boards of your choice and small CRT. BUNKER-RAMO CORP., Chestnut Hill, Mass. For information:

CIRCLE 200 ON READER CARD

transmission control

The 1026 links a variety of remote data communications terminals to a computer, and is for smaller users: 1410, 40, 60 or 1240 computers. With a 210-character buffer it assembles I/O messages. IBM Corp., DP DIV., White Plains, N.Y. For information:

CIRCLE 201 ON READER CARD

tape certifier

The 690 is an off-line mag tape inspection and cleaning device for use by both large installations and manufacturers. There are channel location, tape stopping, and error tabulation facilities. CONTROL DATA CORP., Minneapolis, Minn. For information:

CIRCLE 202 ON READER CARD

multicolored traces

Data reduction shops can get their wiggly output in multicolor with a new photo-recording paper that features the production of well-saturated red, cyan and black traces against a white background. Datacolor 88 comes in 300-foot roll of 12-inch stock. CONSOLIDATED ELECTRODYNAMICS CORP., Pasadena, Calif. For information:

CIRCLE 203 ON READER CARD

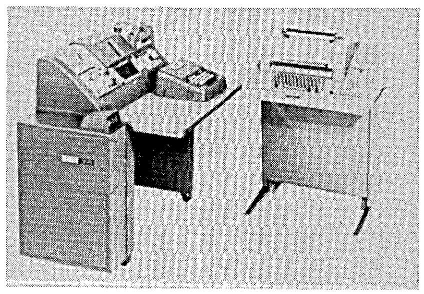
anti-static

Spray device prevents accumulation of static electricity, comes in 11 oz. aerosol can. Is non-toxic, non-flammable. ROBINS DATA DEVICES INC., Flushing, N.Y. For information:

CIRCLE 212 ON READER CARD

punched card transmission

Telepunch system allows transmission of punched card data via TWX; output is in the form of punched cards. System consists of a card transmitter interface between IBM 24 or 26 card punch and mod 35 Teletype, and a similar code converter at the receiving end which outputs on cards data



originated from cards, paper tape, or keyboard. SYSTEMATICS, DIV. OF GENERAL INSTRUMENT CORP., Hawthorne, Calif. For information:

CIRCLE 204 ON READER CARD

core memory

Model 52.02 has read/write time of two usec, access time of less than one usec, capacities to 16K (52-bit) words. Operating temperature is 0 to 65°C. FERROXCUBE CORP. OF AMERICA, Saugerties, N.Y. For information:

CIRCLE 205 ON READER CARD

additional core

Auxiliary core storage for the 1004 raises the number of directly-addressable characters to 1,922. UNIVAC DIV., SPERRY RAND CORP., New York, N.Y. For information:

CIRCLE 206 ON READER CARD

stored-program calculator

Mathatron can be used for fully automatic computation or as a desk calculator with ordinary math notations. Speed is 100 accumulations per second, and number range is $\pm 10^{-42}$ to

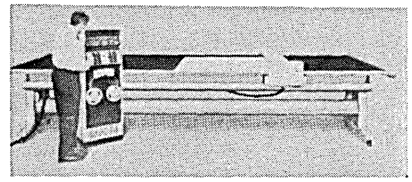
NEW PRODUCTS . . .

$\pm 10^{59}$. Has automatic decimal point location and parenthecation, pre-wired programs, and automatic evaluation of conventional mathematical statements. MATHATRONICS INC., Waltham, Mass. For information:

CIRCLE 207 ON READER CARD

automatic drafting

Series 600 systems plot surfaces up to 5 x 20 feet at speeds to 300 inches per minute and accuracies to ± 0.001 inch. Designed for on-line and off-line

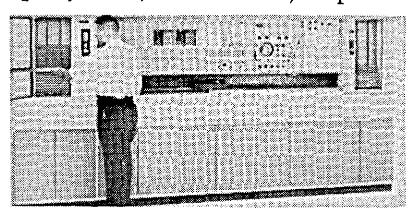


use and to meet requirements of EIA and NAS specs for N/C drafting machines. GERBER SCIENTIFIC INSTRUMENT CO., Hartford, Conn. For information:

CIRCLE 208 ON READER CARD

component tester

The TE 601 is for automatic analysis of diodes, RC modules, resistors and capacitors at up to 7,200 diodes per hour. Uses on-line computer (IBM tape systems, some others). Up to 35



different active and passive component tests at both elevated and ambient temperatures can be performed. IBM INDUSTRIAL PRODUCTS DIV., White Plains, N.Y. For information:

CIRCLE 209 ON READER CARD

paper tape reader

The RR-702 is a photocell paper tape reader and the RS-702 is the spooler, both operating at 700 cps, uni- or bidirectional. Rewind speed is 200 inches/second. Other features: self-adjusting brakes, no-tape and broken tape sensing. RHEEM ELECTRONICS, Hawthorne, Calif. For information:

CIRCLE 210 ON READER CARD

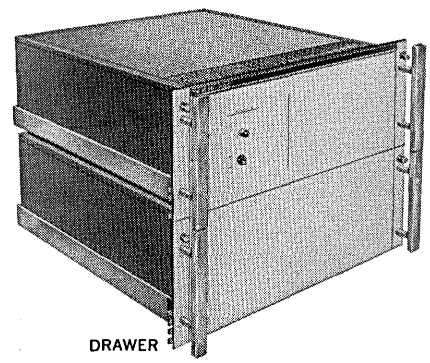
cartridge storage

Cabinets hold six or 12 disc pack cartridges in full-suspension drawers, each drawer having centering cups to keep cartridges stationary. Carrying case for one pack also available. WRIGHT LINE, DIV. OF BARRY WRIGHT CORP., Worcester, Mass. For information:

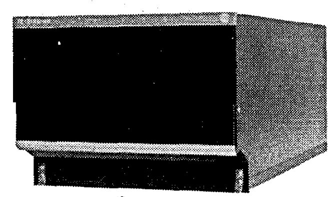
CIRCLE 211 ON READER CARD

Announcing a new line of adaptive microminiature modems with rates up to 6,000 bits per second!

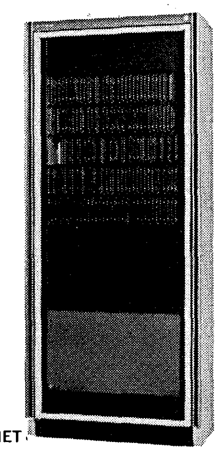
Available in three types: Drawer, cabinet or airborne. They're small; for example, the single drawer type is only 22½" x 17½" x 8¾". They employ Kineplex® techniques, operate on a 3 kc bandwidth over HF and wireline. Data rates can be controlled externally. Computer-assisted design techniques have reduced prices dramatically. If you have large amounts of data to move between computers or from remote sources to computers, contact us today.



DRAWER



AIRBORNE



CABINET

COMMUNICATION / COMPUTATION / CONTROL

COLLINS

COLLINS RADIO COMPANY, Information Science Center, 19700 Jamboree Road, Newport Beach, California, phone 714-549-2911 • International, Dallas

CIRCLE 31 ON READER CARD

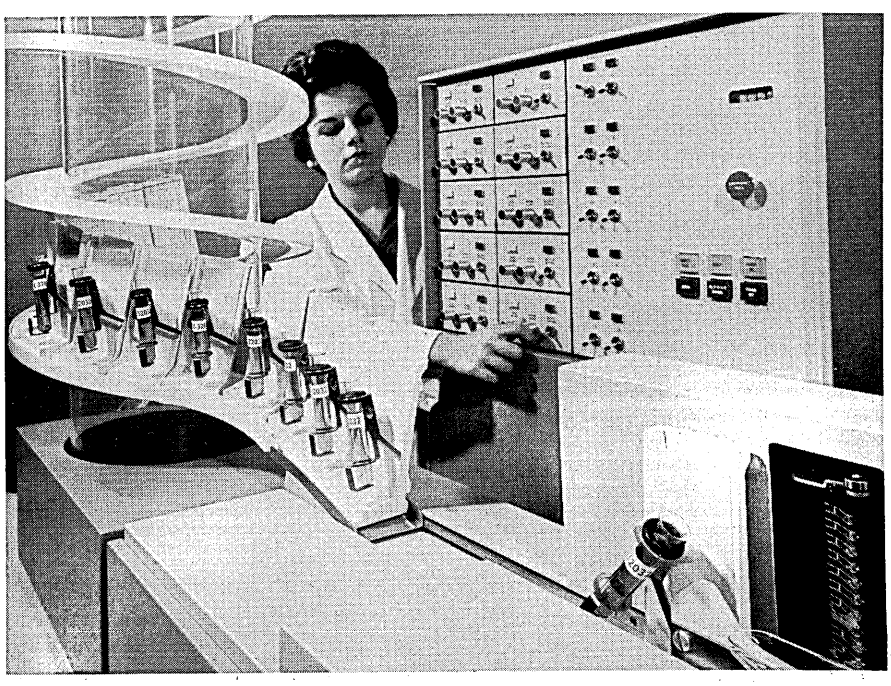
NEWS BRIEFS

DOCTORS SEE HOW COMPUTERS REDUCE CLERICAL TASKS

An experimental data acquisition unit for medical technologists who perform patient tests, and a pathology lab information system, were demonstrated by IBM's Advanced Systems Development Div., Yorktown Hts., N.Y., at the annual meeting of the College of American Pathologists and the American Society of Clinical Pathologists.

The experimental gear (see photo) makes certain lab tests and converts an analog reading into digital form for punching into cards. This data with patient information is then processed with a 1440, which prepares a report for review by pathologist and physician The equipment converts test analysis signals from up to 10 lab instruments.

The computerized information system extends from the bedside physician, ordering patient tests, to the hospital lab. Somewhere in between, though, there's need for a parttime keypuncher, which, slows the process while keeping costs down. Output includes results of tests and a patient summary report; the latter, at a quick glance, gives a physician a quick history of his patient's progress.



COMMITTEE FORMED TO STUDY NON-NUMERICAL PROCESSING

A multidisciplinary committee to study the application of computers to the processing of language data, including mechanical translation of foreign languages, has been formed by the National Academy of Sciences, National Research Council. In addition to mechanical translation, the group will look into computational linguistics, automatic abstracting, indexing and scanning of foreign language texts, and associated hardware.

Under Dr. John R. Pierce, Bell Telephone Labs, will be Dr. John B. Carroll, Harvard; Dr. David G. Hays, RAND; Dr. Charles F. Hockett, Cornell; Dr. Anthony G. Oettinger, Harvard, and Dr. Alan Perlis, Carnegie Tech.

GAS, ELECTRIC METERS READ AUTOMATICALLY VIA PHONES

It should make data acquisition specialists flip, footsore meter readers a thing of the past. In a 16-month field test on 38 meters, gas and electric meters were read automatically over telephone lines with equipment designed by Transitel International Corp., Paramus, N.J. Accuracy of 98% was reported, said to be "better than most men."

The system uses telephone company equipment called an automatic line insulation tester, which automatically dials a customer, interrogates the meter, and readies a recorder for incoming data. A byproduct paper tape is used to produce the billing. Requiring less than five seconds, the process does not ring the phone, jumps to another line if a busy signal is encountered.

The system can also be modified to operate a circuit breaker that could shut the power off or turn it on by telephone.

ST. LOUIS READIES POLICE INFORMATION SYSTEM

A metropolitan police information system loaded with such goodies as names and descriptions of wanted and missing persons and descriptions of stolen property has been announced by the St. Louis, Mo., fuzz. Applications include crime analysis reports, accounting functions, and direct inquiry capabilities.

Response time for the direct interrogator is on the order of three seconds but the cop on the beat, who phones or radios in his inquiry, must await reply from the dispatcher. Hardware scheduled for year-end installation includes an IBM 7040, a 7740 message control system, seven 1050 terminals, and a 56-million-character disc file. The force consists of 1,850 officers and 400 vehicles.

MEDICOMPUTER RESEARCH MOVES ON & ON

Reports on experiments in computer-translation of doctors' gibberish into processable form and computer-assistance in the diagnosis of illnesses were delivered at the sixth IBM Medical Symposium. Trying to overcome doctors' trepidation in using prescribed nomenclature in medical reports, Col. David Tatch told of a set of rules by which a computer could assign its own codes for each of 4,500 terms

NEWS BRIEFS

used in medical diagnosis. Of 2,500 diagnostic phrases, none was redundant.

In another on-going experiment, a data bank approach is used so that a doctor could input some symptoms and have output a list of possible diseases. Asked why a particular disease appeared on the list, the computer would supply a number of possible cause-effect relationships.

U-490 CONTROLS ANTENNA; TRACKS PLANETS, SATELLITES

A computer-guided, 120-foot-diameter microwave antenna is due to go on the air this year in Tyngsboro, Mass., near Boston. Called Haystack, the facility was established by MIT's Lincoln Lab with support from the Air Force's Electronic Systems Div.

Integration of a Univac 490 into the real time antenna control loop reportedly enables a non-skilled user to guide the antenna, communicating with the mainframe through a keyboard. Type MOON, and she's on target; the operator can request the antenna to point to designated coordinates or follow one of many preplanned tracking sequences. She'll track targets with angular rates as high as 3° per second.

The computations to find positions of stars, planets, the sun and moon are found by preliminary interpolation in stored mag tape tabulation of basic *Nautical Almanac* data. Positions of earth satellites and the West Ford belt are found by direct evaluation of the satellite equations from a starting point of orbit parameters. Once an object type has been selected by the operator, a main computer program cycle takes place every two seconds. The new position of the object is determined, superposition of selected scanning takes place, and the position transformed from celestial to radar coordinates. The system can also be run in speeded-up time to serve as a planning device.

GOSH, ANOTHER SMALL ONE: AN $18K PDP-8

Hopefully plugging price-gaps in the computer market is Digital Equipment Corp., Maynard, Mass., who recently introduced its latest, the PDP-8, a very competitive under $20K machine. For $18K, you get 4K (12-bit) words of core, the processor, and an I/O teletype with paper tape punch and reader. Memory cycle time is 1.75 usec, add time is 3.5 usec.

Compatible with the PDP-5, the 8 has FORTRAN and an on-line debugger, the DDT. There's also programmed floating point and eight auto index registers (memory locations). Tagged, "the integrated circuit computer," it has the firm's hybrid (they call 'em Flip-Chip) circuits. Delivery is in eight months.

INTEGRATED CIRCUITS ADDED TO SDS 92 COMPUTER

The SDS 92, announced in these pages last July (p. 61), will be dressed up with monolithic integrated circuits, first such use in a commercially-available computer. With circuitry changes, the machine has also been given an optional multiply and divide feature instead of the formerly announced multiply only, and a memory interlace option.

In integrated form are all the flip-flops, including all registers and memory control circuits. In addition to reducing the number of circuit cards by 40 or 50, the new circuitry is said to increase reliability and ease maintenance.

● The first of a projected 10 computer centers for educational dp and student teaching has been announced by California. Expected to be on the air by mid-65, the system will consist of a Honeywell 200 and 300 and communication linkage to distant schools.

● An optical processing service for its auto-dealer clients, replacing tape-punching adding machines, is being offered by the nationwide accounting systems firm of Reynolds & Reynolds. At the latter's five dp centers, NCR optical scanners will read tapes printed by NCR adding machines.

● Philco's Communications and Electronics Div. has received a $31-million contract to design, install, and run for a year the 10 overseas centers of AUTODIN, the Defense Dept.'s world-wide digital communications system. The contract was let by the Army Electronics Command.

● Yale Univ., whose computer center is housed in a structure donated by the widow and son (class of '42) of T. J. Watson, founder of IBM, has installed a 7094/7040 direct-coupled system. It replaces a 709, and complements a 1401, 1620, and 610 being retained for computer science and programming courses. The center, under Morris S. Davis, offers a 12-hour, non-credit programming course that's repeated through the year.

● A report on eight case studies made to determine factors affecting management decisions to automate is being made available in monograph form by the Labor Dept.'s Office of Manpower, Automation and Training. Studied by the Stanford Research Institute for OMAT were two banks' installation of computers, N/C machines in two electronics manufacturing firms, and automatic order-picking and conveyor systems in three warehouses; one other warehouse decided against automating. The study determines factors influencing management's decision, and expectations and results, particularly as they affect employment.

● Carnegie Tech has established a Center for the Study of Information Processing with a three-megabuck contract from DOD's Advanced Research Projects Agency. In addition to looking into the public utility concept of computer facilities, the center will add to its floor-space, get another computer (new total: five) and 18 more remote teletype stations (new total: 30). Anticipated budget for 1966-7 is $1.8 million with the equivalent of almost 100 fulltime staffers.

● What can you get for $650K nowadays? For that price, IBM's Federal Systems Div. made a thin-film-circuit 26-pound digital computer for the Navy's Bureau of Naval Weapons. Capable of processing 25,000 instructions per second, it's a parallel, binary, single-address computer with 80 thin-film circuit panels and 1,280 (18-bit) words of core. It was designed to accommodate 1,184 words of NDRO storage and 96 standard read/write addresses. The non-destruct feature is attained by removing cores from locations in the program words that contain zeros.

RANDOM ACCESS TAPES

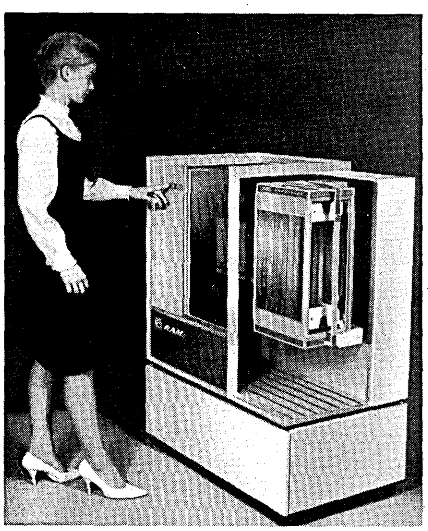
A random access storage unit using 16 loops of mag tape in an interchangeable cartridge was demonstrated last month at the FJCC by Potter Instrument Co. Inc., Plainview, N.Y. The RAM's cartridge capacity is 50.3 million bits, and average access time is 87.5 milliseconds. The device is said to be the first cartridge-loaded random access system to provide an immediate check-read-after-write capability. With one revolution taking 50 msec, this is an average saving of 25 msec per write.

In operation, the 30-inch loops of two-inch-wide tape travel at 600 ips, moving on a layer of air except at the capstan drive, which touches the unmagnetized side of the tape. Recording density is 1,000 bpi, and data transfer rate is 600,000 bps. The read/write heads are mounted in a common post between the two groups of tape loops, each loop having seven writing and seven reading heads.

The tape cartridge reportedly can be replaced in less than 20 seconds. "Disc packs currently in use require delicate handling," says Steve Keane, marketing vp. "A damaged disc cartridge can mean the total loss of information stored in it." By contrast, Keane says, data stored on tapes can be recovered even when the cartridge housing is damaged.

Price of the unit is from $12K to $18K, and the cartridge will go for $150 to $200.

CIRCLE 100 ON READER CARD



RANDOM ACCESS CORES

A mass random access core memory is due for delivery to a customer in April 1965 by Ferroxcube Corp., Saugerties, N.Y. The unit currently under construction has 64K (76-bit) words and a 12-usec cycle time. Target price is 1.5 to two cents per bit for a five-million-bit system; "well below" a cent-per-bit for 512,000 (90-bit) words.

The Ferroxcube approach employs a 30-mil (outside diameter) core. A coincident current write technique is used, coupled with a "word" or linear select read. This design permits a three-wire plane array to be used, and lends itself to low-cost stringing (since the stack is by far the most expensive part of the memory). The main core array is designed in a word-address configuration. Selection takes place through a switch core matrix—eliminating the costly diode selection techniques normally associated with word-select devices. The switch cores, in turn, are selected in a coincident current mode. To maintain simple design within the memory core stack, itself, the same wire is used for both sense and inhibit.

This system contains 256 x 256 words, each word driven by one switch core. Therefore, there are an equal number of switch cores—one on each intersection of 256 X-drive wires and 256 Y-drive wires. All switch cores are threaded by a third wire carrying a bias current, and by a fourth wire which couples the switch core to the 76 cores that make up a word. A word is accessed by selecting one of the X and one of the Y-wires.

A second design phase now in progress is directed toward a considerable reduction in cycle time. "In mass memories of this type," a spokesman reports, "it is completely feasible to achieve a cycle time as fast as four microseconds."

CIRCLE 101 ON READER CARD

NEW LITERATURE

HUMIDIFICATION GUIDE: For industry institutions and commercial buildings for comfort, for handling and processing hygroscopic materials and for preventing accumulation of static electricity are discussed in 28-page booklet. Also discussed is why air is dry and how to correct the condition with dry steam. Full data on the selection, sizing, location and installations for the system for area humidification and air handling systems is also presented. ARMSTRONG MACHINE WORKS, Three Rivers, Mich. For copy:

CIRCLE 130 ON READER CARD

DATA TRANSMISSION: Eight-page brochure includes performance figures, circuit description of the 26B Duobinary-Datatel data modem for transmission at 2400 bps over standard 3 kc voice channels. LENKURT ELECTRIC CO. INC., San Carlos, Calif. For copy:

CIRCLE 131 ON READER CARD

SOFTWARE: 16-page booklet describes modular 3301 system. Included are concurrent hardware and software development, operating system, modular software, growth system, sort/merge technique, service programs, program testing provisions, and scientific language facilities. RCA-CHERRY HILL, EDP. Camden, N.J. For copy:

CIRCLE 132 ON READER CARD

SOFTWARE PACKAGES: Eight ready-to-run packages for GE's compatible/400's are described in a 16-page brochure for dp managers. Coding examples are given for COBOL-61, GE-400 FORTRAN, I/O Supervisor, among others. GENERAL ELECTRIC CO., Phoenix, Ariz. For copy:

CIRCLE 133 ON READER CARD

OPTICAL SCANNER: Four-page booklet describes I-SCAN, a system for converting visual data into digital form. Information on various applications for the equipment in the electronic, medical, metallurgical and meteorological fields and the technical specifications and parometers of the system are presented. DIGITAL ELECTRONICS, INC., Westbury, N.Y. For copy:

CIRCLE 134 ON READER CARD

INTERVIEW HANDBOOK: Produced as part of Auerbach Corp.'s interviewer-training program, the handbook was developed for use in the company's study to determine how information is used to carry out scientific and technical tasks within the DOD. Defense Department contractors can obtain free copies from the Defense Documentation Center; companies not engaged in defense contracts can purchase copies for $5.60 from the Office of Technical Services, Department of Commerce, Wash. 25, D.C.

HIRING COMPUTER TIME: The important considerations and subsequent ramifications, in deciding whether time should be hired from a service bureau, are dealt with in a 19-page booklet, "EDP—The Use of Computer Service Bureaux." Cost: $.30. British Assn. of Certified Accountants, 22 Bedford Square, London, W.C.I.

HARDWARE-SOFTWARE LINE: Complete set (27 one- and two-page illustrated leaflets) of detailed specifications on ICT's new 1900 series dp system, including programming and peripheral equipment. INTERNATIONAL COMPUTERS AND TABULATORS LTD., London, England. For copy:

CIRCLE 136 ON READER CARD

PRESSURE-SENSITIVE, STOCK LABELS: Catalog designed for a wide range of uses with dp and systems equipment is available. The catalog has illustrations of typical press imprinting on stock sizes. A detailed index permits reference to labels by type or function. ALLEN HOLLANDER CO., INC., Bronx, N.Y. For copy:

CIRCLE 137 ON READER CARD

APPLICATIONS REPORTS: Five eight-page reports on the use of the H-200 in five different industries contain flow charts of systems designs, typical print-outs of generated reports and explanations of systems operation. Also shown are H-200 equipment configurations required for implementing personnel reporting, transportation revenue, accounting, refinery yield accounting, retailing sales audit and unit controls, and health insurance premium applications. HONEYWELL EDP, Wellesley Hills, Mass. For copy:

CIRCLE 138 ON READER CARD

MICROFILM DOCUMENT RECORDER: Four-page illustrated brochure describes the S-C 4400. Alphanumeric output is recorded directly from a computer or computer-generated tapes onto 16mm or 35mm microfilm for use in automated storage and retrieval systems. STROMBERG-CARLSON, San Diego, Calif. For copy:

CIRCLE 139 ON READER CARD

TECHNIQUE SELECTION: Automatic Selection of Digital Electronic Computers (ASDEC) for selecting and evaluating computing techniques for specific applications from existing machines is explained in 47-page book. Cost: $2. OFFICE OF TECHNICAL SERVICES, U.S. DEPARTMENT OF COMMERCE, Wash., D.C. 20230.

TOTAL SYSTEM: 16-page brochure describes approach which enables retailers to utilize the services of an NCR dp center for electronically processing original-entry transactions into detailed merchandise reports. Illustrations include typical merchandise inventory report, tax liability report, customer statements, profit and loss, and balance sheet. NATIONAL CASH REGISTER CO., Dayton, Ohio. For copy:

CIRCLE 140 ON READER CARD

PERT TEXT: 171-page book reviews basic PERT time concepts, and includes chapters on planning, scheduling, budgeting, control, updating plans, schedules and budgets, a review, conclusions, and a glossary. Cost: $6.50. SPECIAL SERVICES DEPT., FEDERAL ELECTRIC CORP., Industrial Park, Paramus, N.J.

WHY ARE 19 SYLVANIA LABORATORIES IMPORTANT TO AN ENGINEER WHO NEEDS ONLY ONE JOB?

These laboratories are linked together, within the integrated, nationwide complex of Sylvania Electronic Systems. Their activities traverse the entire gamut of advanced electronics. As a member of this variegated technological family, the individual engineer finds his advancement avenues and opportunities for interdisciplinary exposure have been multiplied, virtually by a factor of 19!

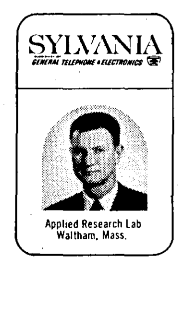
Cross-communications and cross-fertilization are excellent. Technical achievements, originating in, say, Mountain View, California—or in Williamsville, New York—or in Waltham or Needham, Massachusetts—are available to staff members across the Division. Examples of recent SES accomplishments range from survivable ground command/control networks for unmanned Minuteman sites (a Division-wide prime contract) ... to a laser breakthrough (result of a company-sponsored program). The laser wide-band modulator device developed has a theoretical capacity of handling simultaneously on *one* communications channel—more than 100,000 telephone conversations or 100 TV broadcasts.

Just how extensive SES' wealth of state-of-the-art knowledge is may be judged by the following:

In 1963, Sylvania ranked 19th among 500 DOD contractors (from all technical fields) in experimental, developmental, test and research work.

BEHIND THE SES ENGINEER ARE THE STABILITY AND FACILITIES OF GENERAL TELEPHONE AND ELECTRONICS.

The SES engineer has a special relationship with the parent corporation, General Telephone and Electronics, because SES acts in a Systems Management and Systems Program Development capacity for the entire corporation on major government contracts. In addition to operations which concentrate exclusively upon systems work in its most advanced military electronics aspects, SES headquarters houses the Division's Applied Research Laboratory which originates and advances electronic techniques required for tomorrow's more sophisticated systems.



WOULD YOU WORRY ABOUT "TECHNOLOGICAL OBSOLESCENCE" UNDER THESE CIRCUMSTANCES? We think not. Certainly not at all, if you took advantage of SES' professional self-development training programs such as in-plant seminars, symposiums and conferences plus university post graduate studies and professional society meetings.

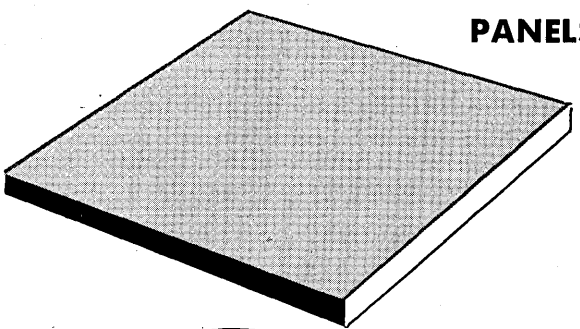
SYLVANIA ELECTRONIC SYSTEMS
Government Systems Management
for GENERAL TELEPHONE & ELECTRONICS

An Equal Opportunity Employer

BEFORE YOU CONTRACT FOR

ELEVATED FLOORING

IT PAYS TO TAKE A SECOND LOOK!

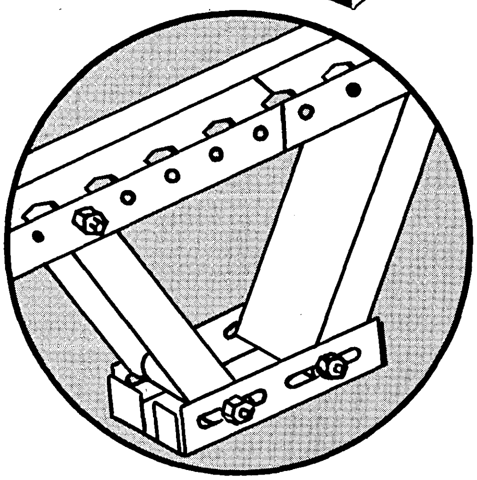


PANELS

Panels available from elevated flooring manufacturers vary considerably in terms of abrasion and indentation resistance.

Sample panels of A, B, and C might look alike. But what does the Taber Abraser (standard test for abrasion) show about elevated flooring panels? It proves the STRATO-FLOOR molded fiberglass panel to be *20 to 30% more mar-and-abrasion resistant and totally free of indentation!* Indentation tests, with compression pressures of up to 5000 lbs. through an individual caster wheel, show that all panels tested, *except STRATO-FLOOR*, had permanent identation.

STRATO-FLOOR panels, of course, are non-conductive and light-weight and are available in plain colors or mottled patterns.



SUB-STRUCTURE

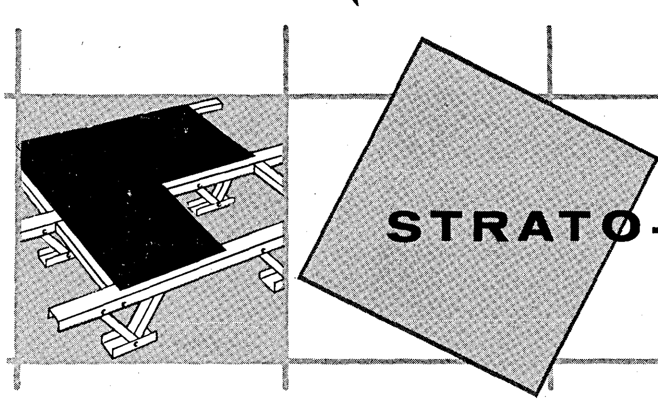
For maximum strength, minimum deflection, the STRATO-FLOOR sub-structure features the patented STRATO-TRI-JACK, which utilizes the strength of the triangle in its design.

Conventional types of jacks in other elevated flooring systems support the stringer and floor at one concentrated point. The STRATO-TRI-JACK distributes support of the stringer over a larger area, increases the load-bearing capacity and provides greater strength and rigidity. Supports a live load of 250 lbs. per square foot and a point load of 1000 lbs. with at least 4 factors of safety. 1/8″ galvanized steel channel is the strongest sub-structure on the market.

CONTRACTOR-RELIABILITY

STRATO-FLOOR elevated flooring is sold and installed by STRATO-FLOOR, INC. or by licensed STRATO-FLOOR distributors. They are reputable contractors, carefully selected by STRATO-FLOOR and known for reliable and conscientious work in the communities they service.

This is important. Before you contract for elevated flooring—get full details about STRATO-FLOOR. Write us for further information and the name of your STRATO-FLOOR distributor.



STRATO-FLOOR, INCORPORATED

795 East 152nd Street • Cleveland 10, Ohio

Phone: LIberty 1-4200

CIRCLE 47 ON READER CARD

washington report

Continued from page 21 . . .

and open therefore to very broad interpretation. The BOB-sponsored amendments referred to by Sen. McClellan are attempts to tidy up this language and insure that GSA will be something less than a final arbiter on how government agencies will spend money budgeted for computers. Outside contractors using government-financed computers have also taken umbrage over 5171, specifically at the phrase, "or at the expense of federal agencies" in defining those computers over which GSA shall have jurisdiction. Officials of North American Aviation, Aerojet-General, General Dynamics and Chrysler express a pretty unanimous opposition to any GSA gumshoes tracking up their computer rooms.

The Brooks forces, however, contend that the phrase, "at the expense of," is critical to effective legislation in this field since so much computing equipment procured at government expense is under contractor control. They may have a point. Informed sources estimate the amount of money spent on contractor-based computer equipment by the government is double what it spends in-house, totalling perhaps as much as $2 billion yearly.

government procurements continue apace

There's no shortage of new business for computer makers in Washington these days. Philco received a $31.4 million award from the Defense Communications Agency to supply special purpose computing gear for its 10 new AUTODIN overseas message switching centers. Even at what some consider to be a low, low price, it's the biggest bit of computing business to come Philco's way in many a moon.

Honeywell chalked up another major sale with its lease to the Air Force of eight 800/200 systems to replace outmoded equipment in use at major air commands. The deal represents about $1.5 million in annual revenue for Honeywell. Even more significant, Control Data reportedly has breached the walls of what was considered to be a staunch IBM outpost at the U.S. Weather Bureau, where a 6600 will be going in next year replacing a STRETCH and a 7094 mod II. A second 94 will remain, at least for the time being. The 6600, a full-blown system which rents for about $150K monthly, is expected eventually to take over all the chores, and then some, now handled by the three IBM systems, which rent for about $350K.

The deal underscores what many consider to be the vulnerability of remaining IBM lease installations in the government.

Correspondingly, greater rue is being expressed by non-IBMers over the mammoth $230 million purchase of leased machines last winter by the Defense Dept., the majority of which were 1401's, 90's, and 94's.

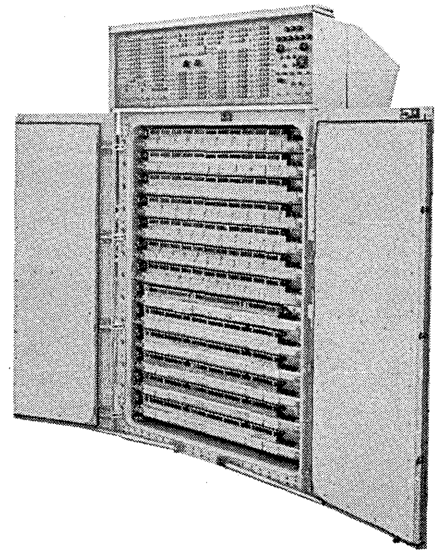
"We didn't realize then just how seriously this action would affect our market," said one D.C. computer sales executive, "although we knew many 'soft' installations were included. Were it not for that purchase, a great many more IBM systems would now be in serious trouble."

A goodly number of 1401's, however, do remain on lease in government agencies, and about these, reports of replacement swirl constantly.

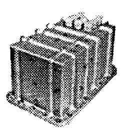
The Navy is said to be readying specs for bids to replace a fleet of 26 1401's, the Post Office has the same intentions concerning 12 of its 1401's.

The big question, of course, is whether IBM will be able to defend its vast corps of purchased 1401's when these become eligible for replacement in 1966 and 1967.

If you're interested in advancing high speed data processing techniques, consider current progress at UNIVAC-Twin Cities



in Clipping Nanoseconds



UNIVAC engineers and scientists recently proved out the feasibility of 200 nanosecond memories. Now work is under way on memories with cycle times of less than 100 nanoseconds.

This is typical of practical advances resulting from basic breakthroughs in thin films and microelectronics made by UNIVAC professional staff.

But, achieving a leading edge in computer technology is only half the story. At UNIVAC-Twin Cities we view computers in a systems context — we're primarily concerned with complete systems, where high speed data processing forms the heart of the operation.

This total-systems orientation demands a multi-discipline staff with a wide range of capabilities and specialized talents — people knowledgeable in the many facets of aerospace and surface based computer systems technology and analysis.

Development activities are mounting; heavy staff increases are in order. Current programs include multi-processor computers and computer memories for the Army's Nike-X anti-missile missile system; A-NEW — anti-submarine warfare system; MWRT — Mobile Wing Reconnaissance Technical Squadron; others. Opportunities are open in these areas for people with a professional degree and appropriate experience:

COMPUTER SYSTEMS DESIGN & ANALYSIS
AEROSPACE SYSTEMS ANALYSIS & DEVELOPMENT
ADVANCED DEVELOPMENT & ELECTRICAL ENGINEERING
IN MEMORY AND DATA TECHNOLOGY
SCIENTIFIC PROGRAMMING/ANALYSIS
COMPUTER HARDWARE PRODUCTION ENGINEERING

Write to Mr. R. K. Patterson, Employment Manager, Dept. L-12, UNIVAC Division of Sperry Rand Corp., Univac Park, St. Paul, Minn. 55116. An Equal Opportunity Employer.

UNIVAC
DIVISION OF SPERRY RAND CORPORATION

COMPUTER PROGRAMMERS & SYSTEMS ANALYSTS

ready to

MOVE UP!

Here at the U.S. Army Data Support Command in Washington, D.C., MOVE UP means growth potential in three important ways:

1/ Expand your capabilities to the utmost by tackling wide-ranging programs at one of the most complete and advanced computer activities in the country. Using IBM 1401, 1460, and 7080 and RCA 501 digital computers, you will provide vital management information to the Army staff in personnel, budgets & accounting, supply, procurement, and other fields—on a worldwide basis.

2/ Advance to higher positions as rapidly as you can qualify. Salaries start between $7,220 and $10,250—with Civil Service benefits, of course,—but EDP technology is so vital to the Army's operations that there is a constant need (and opportunity) for programmers and analysts at ALL levels.

3/ Enjoy the added benefits of an "edp-oriented" environment. Not only are there several government computer complexes in the area, but a surprising number of private edp firms are moving in and taking hold.

If you have your degree and sufficient digital computer experience (including at least two years in systems analysis or programming) to work independently on large-scale projects, please send resume or SF-57 (available at the Post Office) to: Mr. J. F. Mackrill, Staff Civilian Personnel Division, Office of the Chief of Staff, Department of the Army, The Pentagon (Room 1E429), Washington, D.C., 20310. The Army is an Equal Opportunity Employer.

CIRCLE 88 ON READER CARD

FREE Data Processing Opportunities Bulletin

Shows the positions you could have in the field of Data Processing.

Cadillac Associates, the nation's largest executive and professional placement service, represents the majority of the nation's top companies in Data Processing. Their best jobs, at salaries from $6,000 to $75,000 appear in our monthly Data Processing Opportunities Bulletin.

Both the Bulletin and our completely confidential placement service are available to you absolutely free of charge. Client companies pay our fees.

For your free Bulletin without *any* obligation, circle Subscriber Service Card No. 85. Please use *home* address only.

LON D. BARTON
President
Cadillac Associates, Inc.*
29 E. Madison Bldg.
Chicago 2, Illinois
FI 6-9400

**Where More Executives Find Their Positions Than Anywhere Else in the World*

In Los Angeles— Lon Barton Associates 3275 Wilshire Blvd.
In San Francisco—Lon Barton Associates, 120 Montgomery St.

CIRCLE 85 ON READER CARD

WANTED

Men to match our mission

Do you want to be *motivated* ☐,
or just *moved* ☐?
Are you ready for a *challenge* ☐,
or merely a *change* ☐?
Do you want to *promote* your competence ☐,
or simply *preserve* it ☐?

Answer these questions objectively. If you have checked the first box in each case, let's get acquainted. We want to tell you about our work and aspirations involving the design of information systems for administrative, engineering, and scientific applications, and to hear the same from you. We are interested in people with competence in

- Operations Research
- Systems Design and Engineering
- Systems Analysis
- Computer Programming
- Software Design
- Applied Mathematics and Statistics

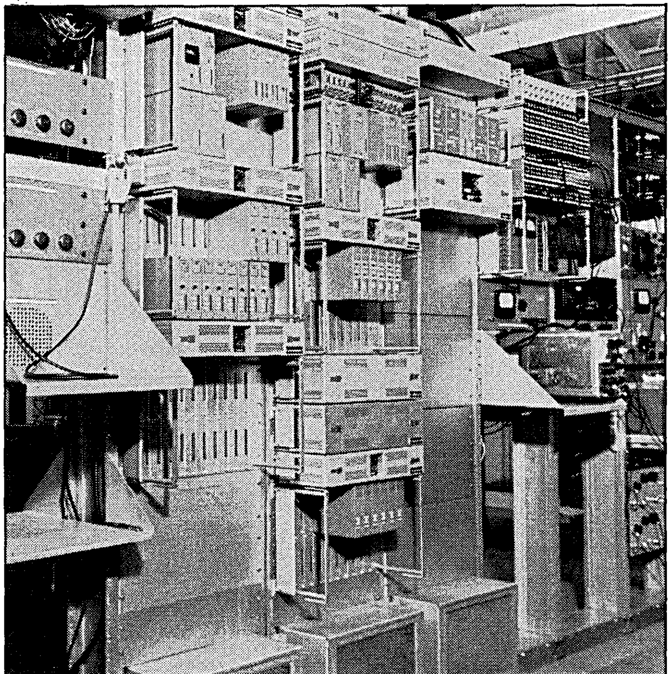
Now is the time to send your resume to

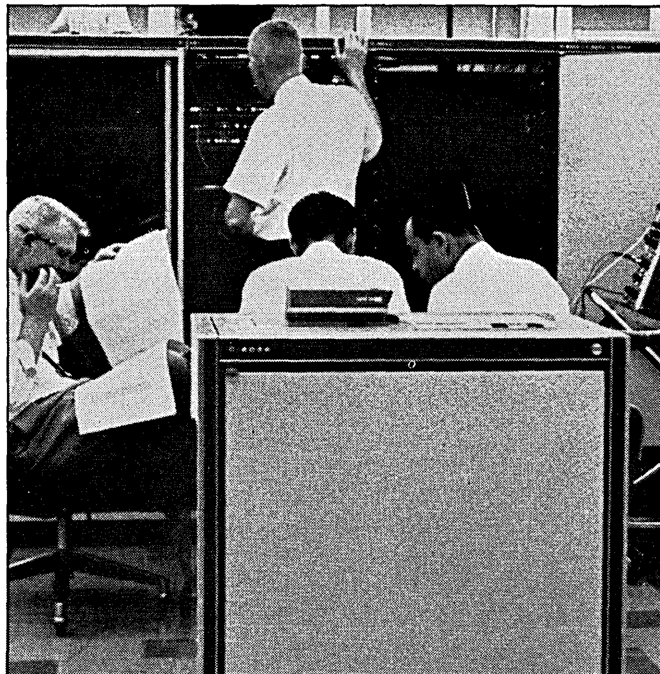
Mr. Charles G. Calderaro, Director
Arizona Research Center
URS Corporation
Box 1025, Sierra Vista, Arizona

An equal opportunity employer—*by choice*

CIRCLE 87 ON READER CARD

Has your 1955 project been this successful?





In 1955, Collins was on the very edge of the data business, concentrating on transmission techniques. Culmination of research which began in the '40's was Kineplex®, a system which employed the methods of kinematics with multiplexing. This signaling and detecting system was achieving 4,000 words/minute or 3,000 bits/second in a band of less than 3 KC. A converter for punched card input/output was being developed, to be followed by equipment for magnetic tape.

Today, we are in final development stages on the first all-purpose computer designed from the ground up for high-speed information and control systems, and are looking beyond present computer techniques.

In this new era of Applied Information Science, we are working in the area of adaptive control of communications. Our new stored logic machines have the versatility to handle business and scientific data, communications switching control and process control. We are working with integrated logic circuits and high-speed memories in the 10-500 nanosecond range. We are concentrating on increasing the speed and integrity of information transfer within the computer as well as in the outside communication media. Improved system performance is achieved through the application of model analysis, simulation, switching theory, and logic synthesis.

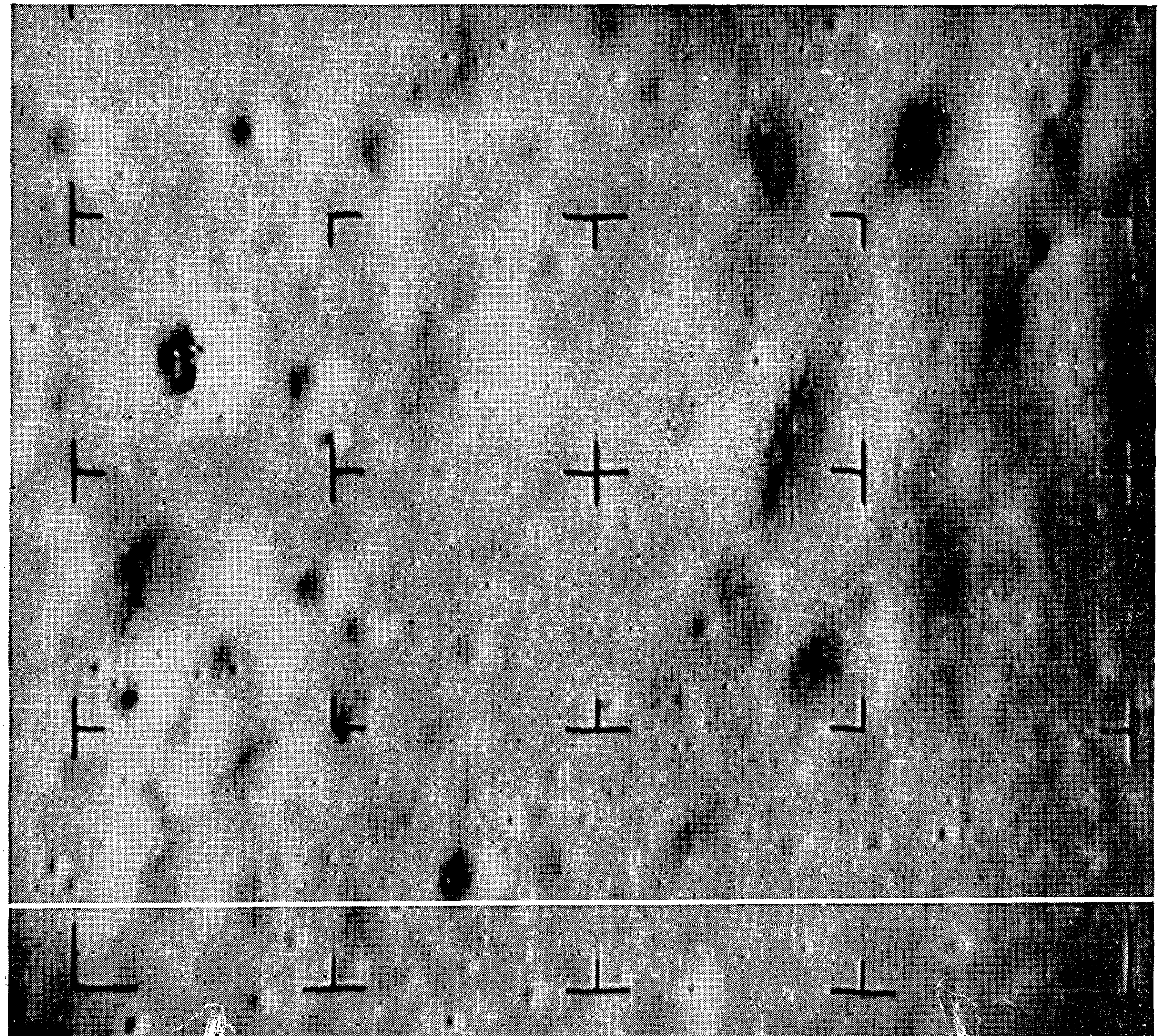
Further, we apply the experience gained from operation of the largest and fastest switching center in the world, handling some 12,000,000 words a day virtually without loss. We welcome the help that top-level, open-minded people with logic design, machine organization, or software experience can give us. Top-level people with an open mind and experience in broad areas of software and programming. People who want to stay with the most advanced project and succeed with it. Contact E. M. Moldt, Collins Radio Company, Cedar Rapids, Iowa.

COLLINS RADIO COMPANY
Cedar Rapids, Iowa • Newport Beach, Calif. • Dallas, Texas

COLLINS

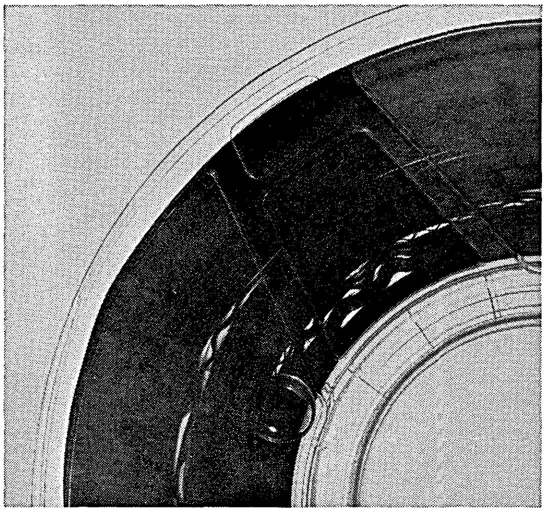
An equal opportunity employer

CIRCLE 89 ON READER CARD



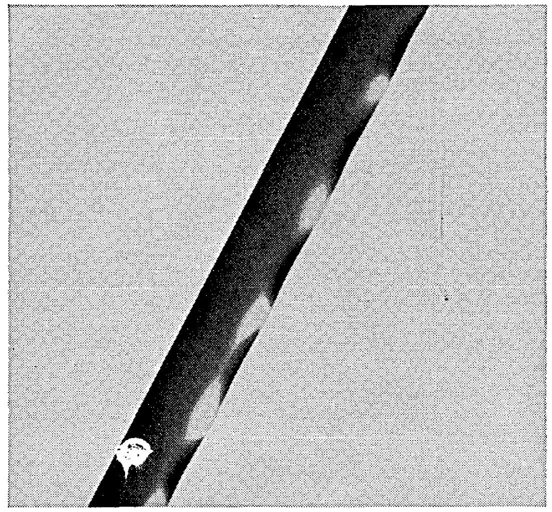
4 COMPLAINTS ABOUT COMPUTER TAPE

(And how Memorex solves them!)



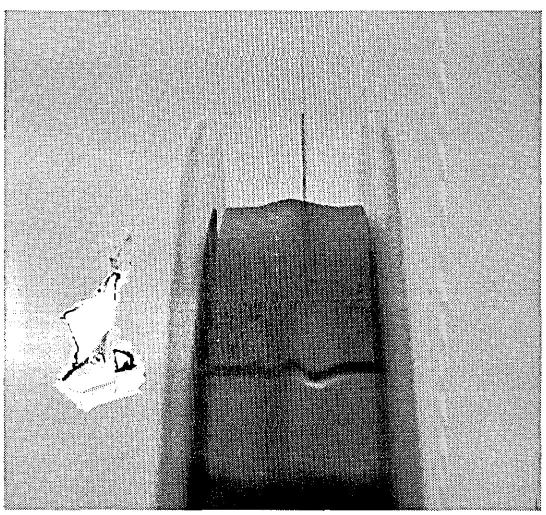
Complaint. Cinching during shipping, use or handling results when reel is wound under improper tension or exposed to temperature extremes.

Solution. Precision winding, special packing and careful shipping are examples of attention to detail that insure cinch-free delivery every time.



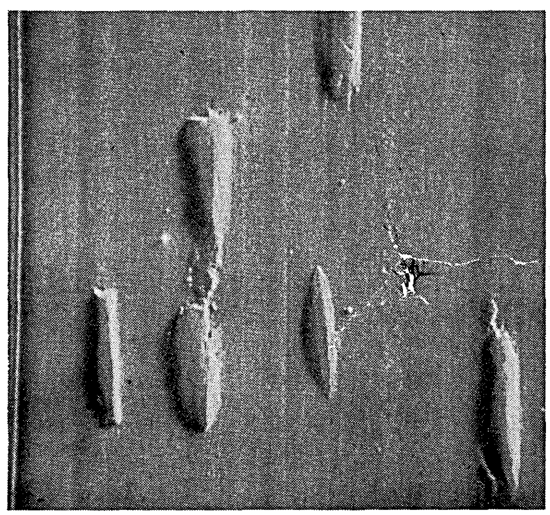
Complaint. Wavy edge caused by improper slitting.

Solution. Specially designed Memorex slitters and microscopic edge inspection of every reel prevent wavy edges. Fifty-one other quality control checks (many performed only by Memorex) guarantee every Memorex reel pre-test perfect.



Complaint. Semi-permanent ridging and loss of contact caused by microscopic scratches produced in manufacturing or use.

Solution. Memorex-designed manufacturing facilities include equipment unique to the industry which eliminates all fixed friction surfaces that potentially produce scratches.

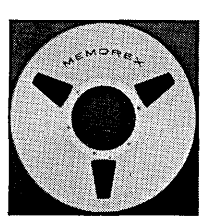


Complaint. Dropout-causing clumps of redeposited coating (50X magnification).

Solution. Memorex has developed coating formulations and processing methods to achieve superior bond between coating and base, extra toughness, high flexibility, and a smoother surface. Result: Memorex tape is essentially redeposition-free.

Memorex tape is premium tape. No need to pre-check it. You can place Memorex computer tape directly in service — reel after reel.

Memorex certification means what it says: Memorex computer tape is error-free. Extra care, extra steps and scrupulous attention to every detail make it that way. We know the importance to you of having a tape you can depend on.



Are you on our mailing list to receive the Memorex Monograph Series of informative technical literature? Write 1172 Shulman Road, Santa Clara, California.

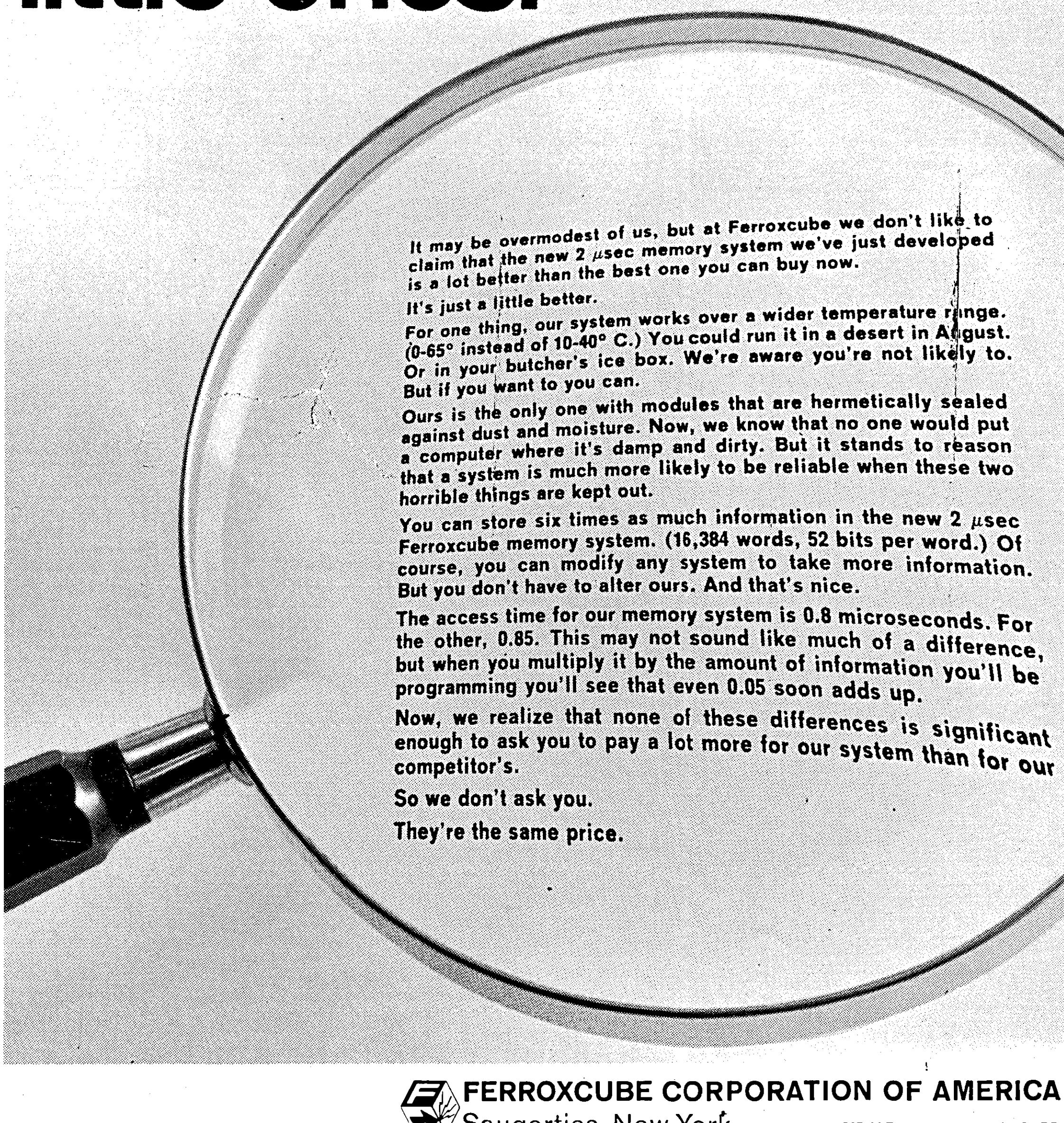
MEMOREX

PRECISION MAGNETIC TAPE

CIRCLE 2 ON READER CARD

When you're dealing with things as small as 30 mil cores, it's hard to make any big improvements in a 2 µsec memory system.

So we've made some little ones.



FERROXCUBE CORPORATION OF AMERICA
Saugerties, New York

CIRCLE 3 ON READER CARD