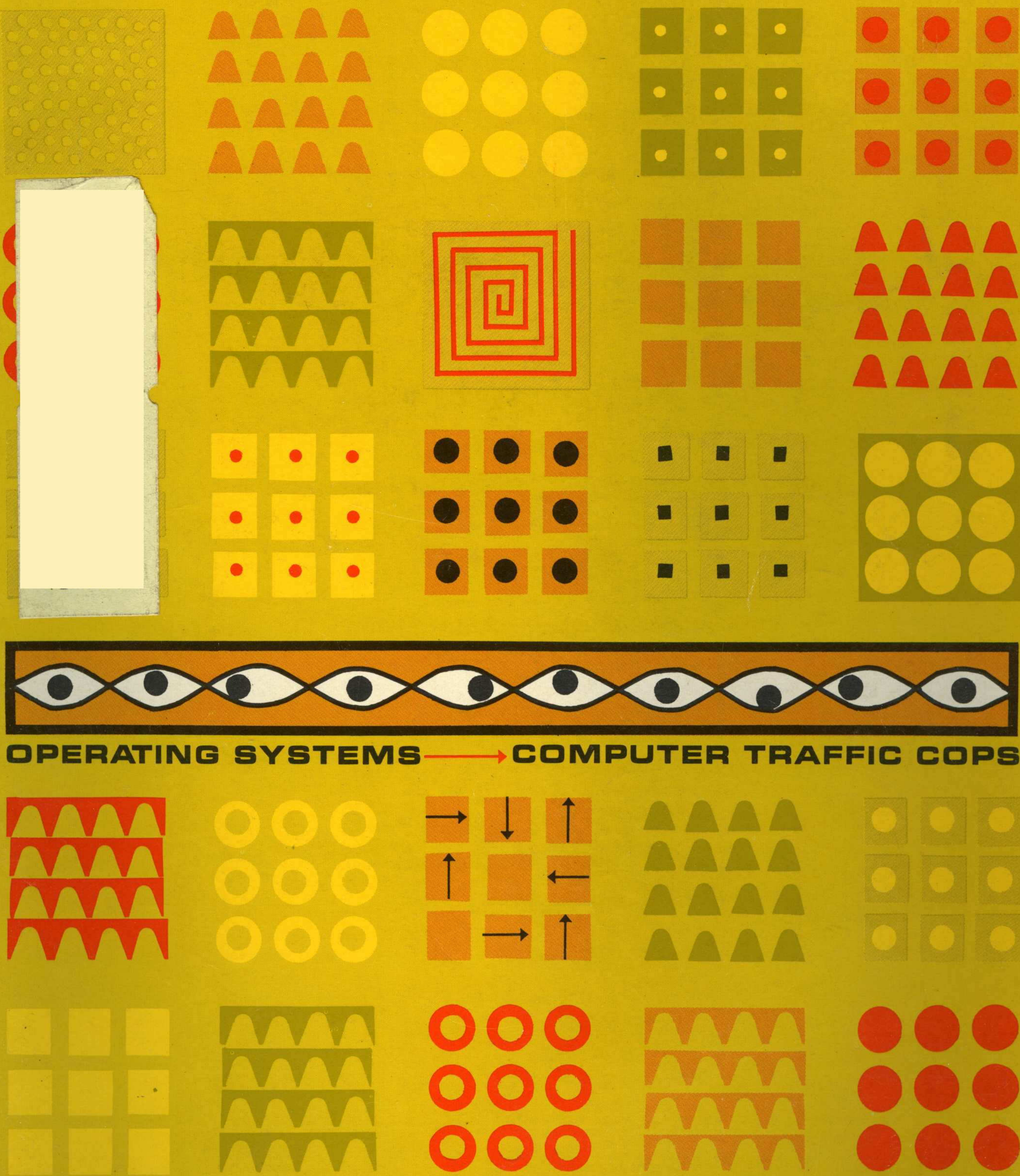
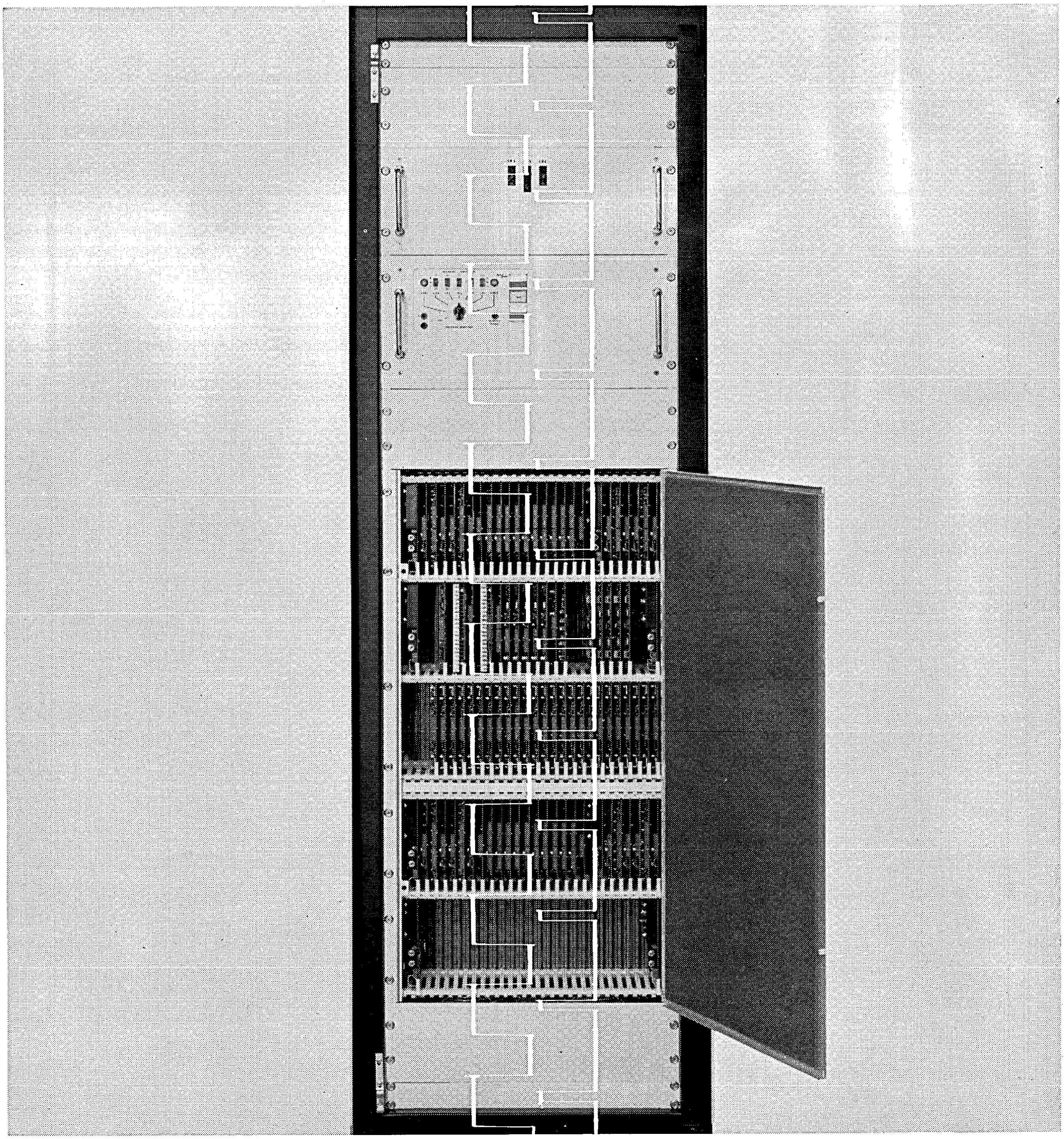
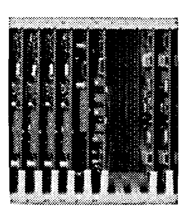
DATAMATION

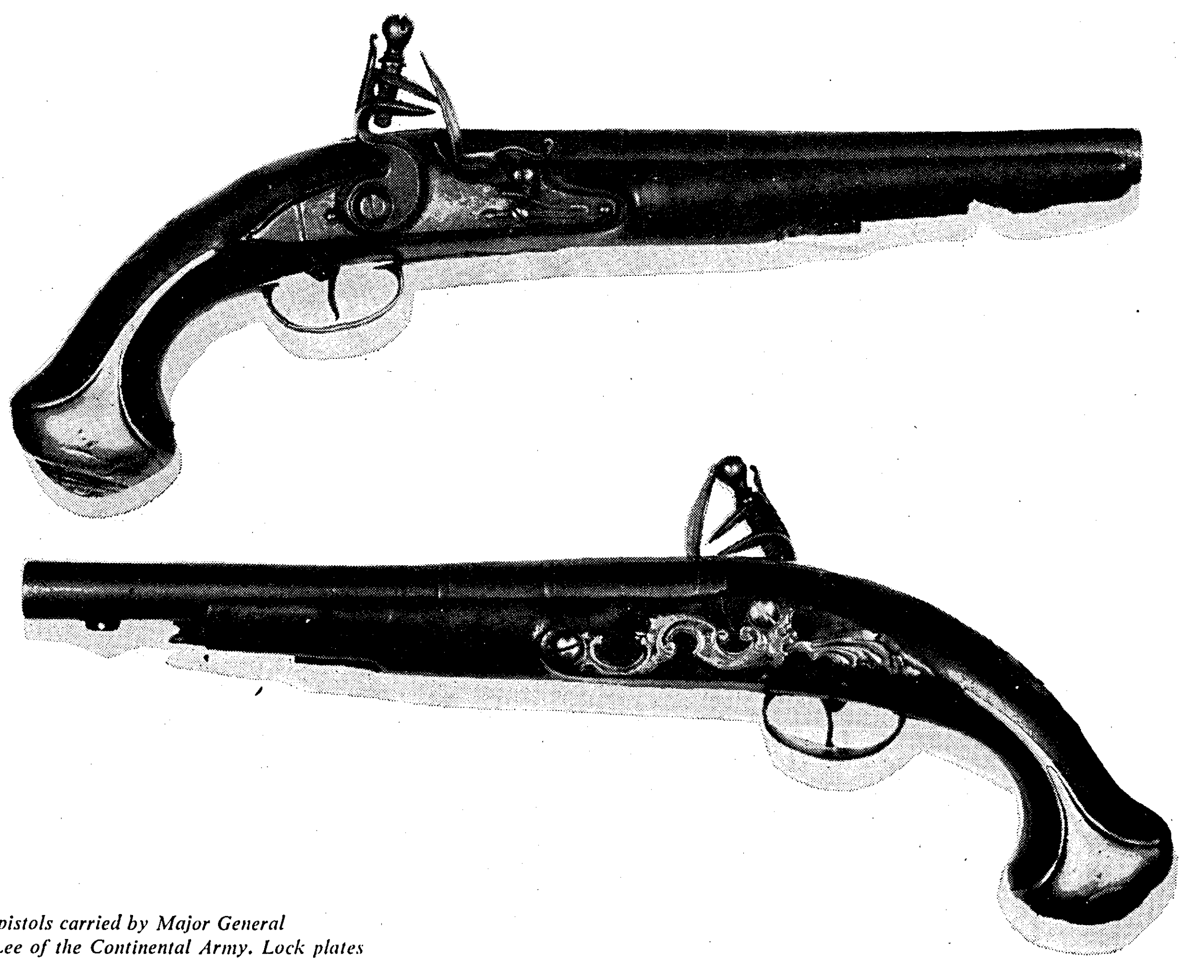
64

May

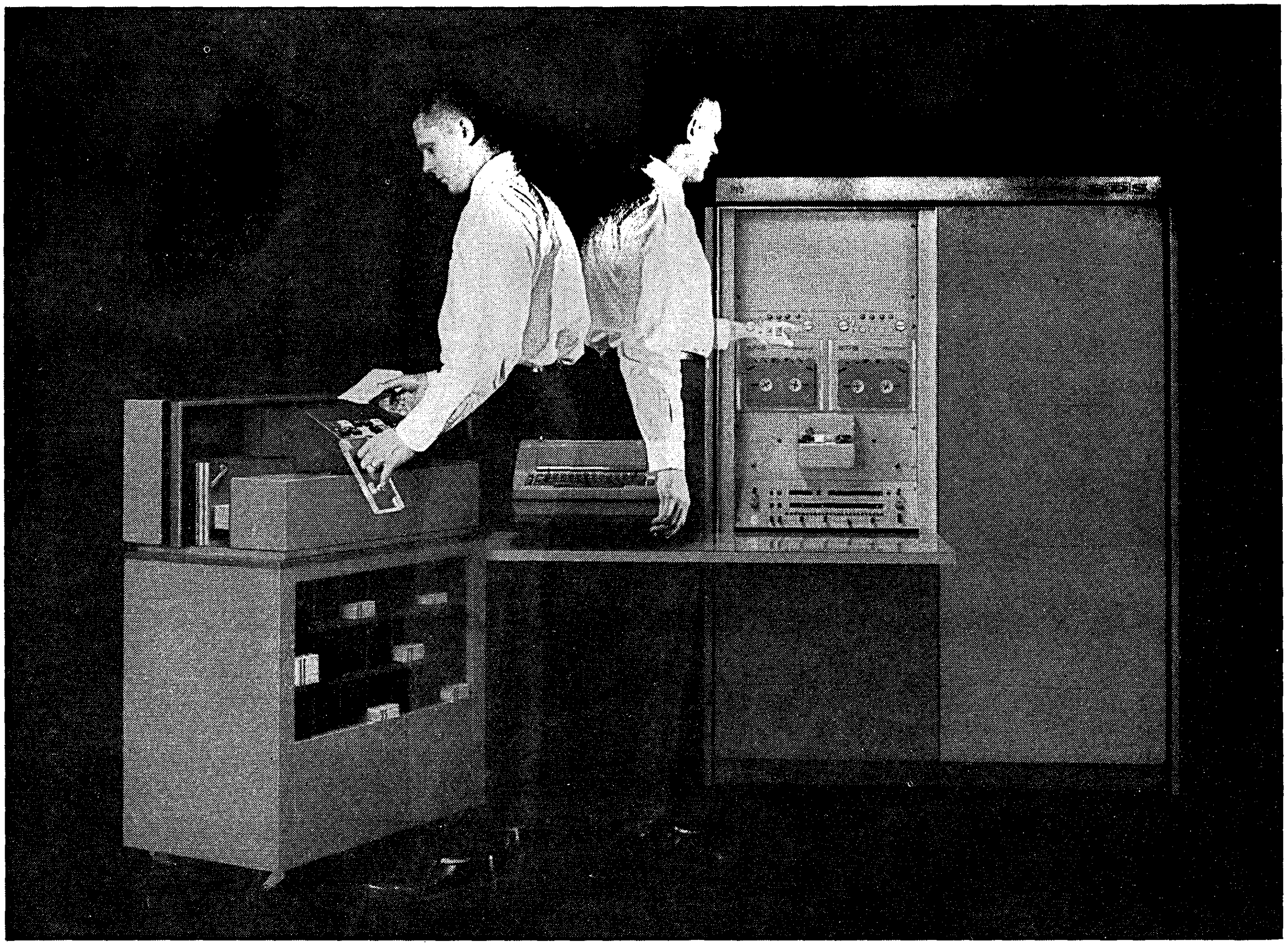








OF SMALL COMPUTER OPERATION



WITH MAGPAK

To assemble a program

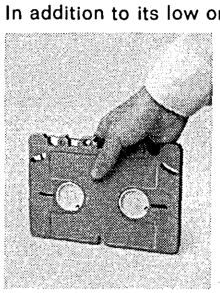
1. **Load single card deck containing source program and data.**

2. **Execute the program.**

performance costs twice as much, yet operates up to ten times slower.

Magpak convenience saves money

In addition to its low original cost, Magpak greatly reduces operating costs. Most small, scientific/engineering computers are used on an open shop basis. They are generally operated by high salaried technical personnel with only moderate computer usage skills. This results in lost time and operating errors. With Magpak, both the time required to operate the computer and the opportunities for human error are greatly reduced. And the easy to carry, easy to store, easy to mount Magpak tape cartridges eliminate the wear, tear and sequencing problems inherent to paper tapes and cards. With Magpak, you simply load a punched card source program and execute your problem in a manner identical to operating a large, expensive computer system.



For more information

If you would like additional information about SDS 900 Series Magpak systems, contact your nearest SDS sales and service office, or write on your letterhead for our Magpak Data Bulletin.

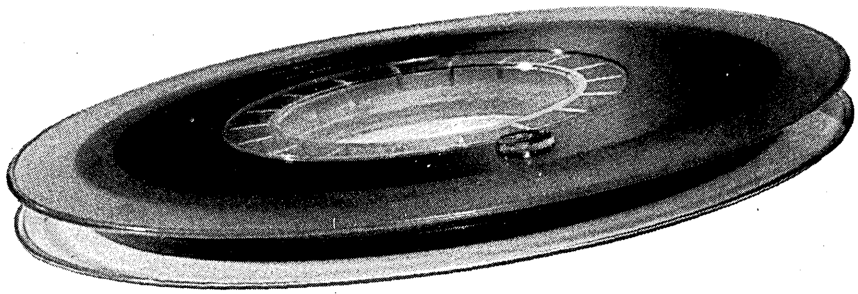
SDS 900 SERIES/MAGPAK

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

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

CIRCLE 8 ON READER CARD

This is Computape reel #8741-11



Case History: Reel #8741-11 was part of an order shipped to the Engineering Department of a leading University. There it was selected at random and subjected to a continuous wear test. In 268,853 passes no permanent dropouts were experienced.

Exceptional? Yes, outstanding . . . But another example of the longer wear qualities of Computape. Longer wear, less static build-up, greater reliability than ever before.

Test it yourself. Against any other tape.

COMPUTRON INC.
122 CALVARY STREET, WALTHAM, MASSACHUSETTS

CIRCLE 9 ON READER CARD

DATAMATION 64

the automatic handling of information

volume 10 number 5

Feature Articles

Departments

Staff

President **FRANK D. THOMPSON**

Publisher **GARDNER F. LANDON**

Editor **ROBERT B. FOREST**

Managing Editor & Art Director **CLEVE MARIE BOUTELL**

Assistant Editor **EDWARD YASAKI**

News Editor **VALERIE ROCKMAEL**

Washington Reporter **JAMES McCROHAN**

Editorial Advisers **DANIEL D. McCRACKEN**, **ROBERT L. PATRICK**, **LOWELL AMDAHL**

Technical Consultant **DON B. PRELL**

Production Manager **MARILEE PITMAN**

Circulation Mgr. **ALGENE TRAINA**

Eastern District Managers
FELIX CHARNEY
JAMES M. MORRIS
141 E. 44th St., New York 17, N.Y.
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
201 N. Wells St., Chicago 6, Ill.
Financial 6-1026

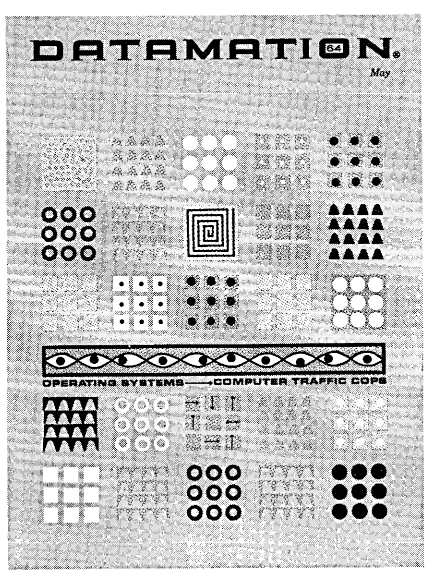
Western District Manager & Vice President
HAMILTON S. STYRON
1830 W. Olympic Blvd.,
Los Angeles 6, Cal., 385-0474

EDITORIAL OFFICES

1830 WEST OLYMPIC BLVD.
LOS ANGELES 6, CALIF.

THIS ISSUE — 46,800 COPIES

Cover



Keeping a directive eye on traffic flow into, out of, and through a computer is the operating system, featured this month with three descriptive articles preceded by a tutorial piece. Suggesting the various functions of these complex software systems is the cover, designed by Art Director Cleve Boutell.

Circulation audited by Business Publications Audit

Member, National Business Publications

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.

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

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 St., New York 17, N.Y. (MUrray Hill 7-5180). Editorial offices, 1830 W. Olympic Blvd., Los Angeles, 6. Published at Chicago, Ill. Controlled Circulation paid at Columbus, O. Form 3579 to be sent to F. D. Thompson Publications, Inc., 201 N. Wells St., Chicago 6, Ill.

Printed by Beslow Associates, Inc.

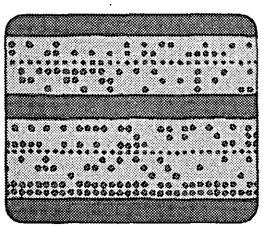
WHICH PAGE PRINTER DO YOU NEED?

With several new and different lines of Teletype send-receive and receive-only sets, how do you decide which is best for you? While they may seem alike at first glance, there are actually many differences.


4-row keyboard

Let's begin with the most obvious difference. The Model 32 send-receive set has a 3-row keyboard, like the familiar Model 28, but the Model 33 and 35 have 4-row keyboards. Why the shift to a 4-row keyboard when the 3-row keyboard has been standard for many years? To avoid the shift! That is, to avoid the need to shift in order to type numbers and common punctuation marks. Besides saving strokes and cutting down on errors, the 4-row keyboard is familiar to any typist.

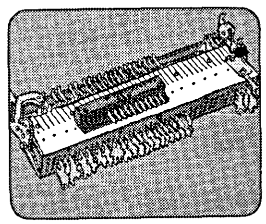
Now, for the not so obvious differences. The Model 32 page printer operates on the 5-level, 7.50 unit code, but the Models 33 and 35 handle an 8-level, 11-unit message and data code which conforms to the newly approved American Standard Code for information interchange. Both the 32 and 33 units are available with a friction feed platen. The Model 35 can have either a friction or sprocket feed platen, the latter for use with multiple-copy business forms.


5- and 8-level tape

The Model 35 has the exclusive "stunt box" that can perform a wide variety of switching duties including "editing" incoming messages, as well as turning on and turning off unattended Teletype equipment. The Models 32 and 33 are equipped with a simplified "stunt box."

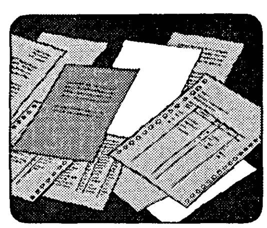
What do these Teletype machines have in common? Many things. They operate at 100 wpm; have pneumatic shock absorbers for smooth, quiet carriage return; all-steel clutches that engage at high, positive pressures to insure slip-free operation; and quieter nylon gears and pulleys that last longer and cut maintenance to a minimum.

What about optional features? The Model 35 offers many modification kits to serve individual needs. Included are vertical tab, horizontal tab, page feed-out, and a variety of different width platens for printing on business forms.


stunt box

Now, which machine should you use? Though your specific needs have to be taken into consideration, the following will serve as a guide: Where the traffic ranges from normal to light, Models 32 and 33 are the most economical. The Model 35 is designed for handling a much larger volume of messages and data, as well as offering increased versatility for on-line and off-line communications.

What are some typical applications? This Teletype equipment is used to handle a variety of business needs, such as to link sales offices with customers, production plants with company headquarters, warehouses with distribution outlets, purchasing with outside suppliers.

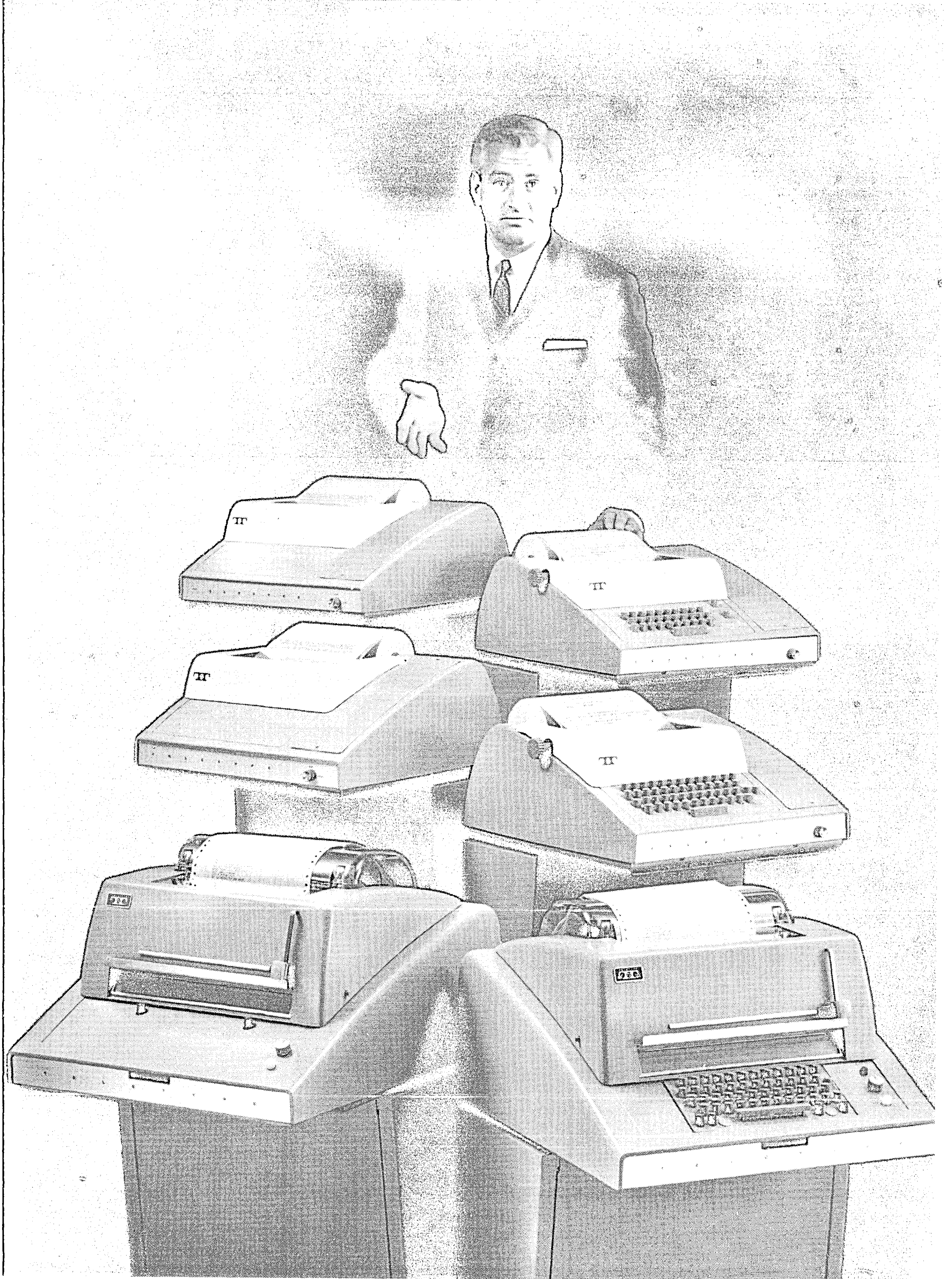

multiple-copy forms

All business forms can be typed on this equipment, such as invoices, payroll checks, personnel records, sales orders, freight bills, tracers, and reservations. In addition, they can be used to gather information for sales reports, expense figures, production schedules, and account facts.

This kind of equipment is made for our parent company, Western Electric, Bell System affiliates, and others who require the utmost reliability at the lowest possible cost.

To obtain a new and informative brochure on the uses of the Model 32, 33, and 35 equipment, write: Teletype Corporation, Dept. 81E, 5555 Touhy Avenue, Skokie, Illinois 60078.

TELETYPE ®

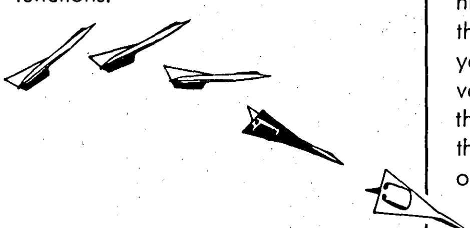


Solution by simulation

CONTROL SYSTEM OPTIMIZATION ATTAINED IN RECORD TIME WITH HYBRID SIMULATION

A new generation of problems have emerged in recent years, exceeding the complexity and accuracy requirements of those which had previously challenged equipment and personnel capabilities of major computing laboratories. Traditional computing techniques, i.e. analog or digital, were unable to handle them. New techniques and equipment were required.

One form of hybrid computation is the association of a general purpose analog computer with a complement of general purpose digital logic, memory and conversion components. Its genesis was rooted in the desire to exploit the high-speed integration capability of the analog computer. The consequent need to control and change the analog's program on the basis of previous results at correspondingly high-speed, led to the adoption of parallel digital devices programmed to exercise the necessary logic, control, timing and data storage functions.



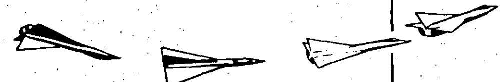
The employment of such a hybrid computing system significantly extends the simulation capabilities of the modern computing laboratory. It permits the actual "automating" of certain classes of problems and achieves impressive savings in computation time and engineering evaluation effort.

Optimizing the Real Time and 1000 X Real Time Simulations

Optimizing response to pilot input commands can significantly improve the performance of high-speed aircraft. A recent study at the EAI Princeton Computation Center confirmed this. An aircraft adaptive control system was simulated on a HYDAC 2000 hybrid computing system.

The analog portion of HYDAC 2000 simulated dynamics of the aircraft both in real time and at 1000 times real time. A reference model representing the idealized aircraft response was also simulated at 1000 times real time. The digital portion of the computer made an efficient multi-parameter search through a decision making sequence accomplished with programmed logic elements.

For a given pilot command, the response of the reference model and the actual aircraft were computed at high speed and compared. Gains in the control loop for pitch, roll and yaw were systematically varied until values were determined which gave the greatest correspondence between these two high-speed systems. These optimized values were determined



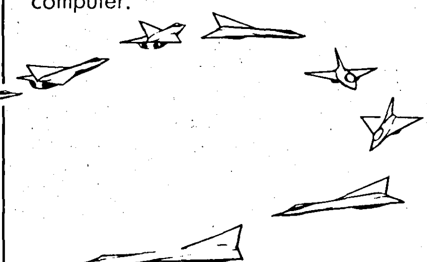
within 0.1 second after the pilot initiated his command. The gain settings were then implemented in the real time model of the aircraft. This process continued to occur with gain values being upped ten times a second as the input signals changed.

The study is described in EAI Application Study 3.4.5H, available on request.

Advanced Hybrid Computing Systems

EAI undertook in 1961 the development of a hybrid system that would meet the requirements specified in its own and customer computer laboratories. These advanced computing tools extend the science of simulation by: 1. *enabling complete system simulation.* Onboard digital computers, actuators, etc., may be readily included within the simulation. Transport delay simulation, accurate function storage, multi-variant function generation, partial differential equation solution and other complex functions and calculations are now practical. 2. *increased simulation laboratory efficiency.* Logic control, high-speed iterative techniques permit automatic parameter search, optimization, stability studies, parameter sensitivity analyzing, etc.

EAI's series of integrated hybrid computers includes: HYDAC 2000 — comprising the 231R-V analog computer and the Series 350 digital operations system; HYDAC 2400—comprising the 231R-V, DOS-350 and the 375 (DDP-24) general purpose digital computer.



For detailed information on HYDAC 2000 or 2400 and their application, or for research support services available at EAI Computation Centers, write today.

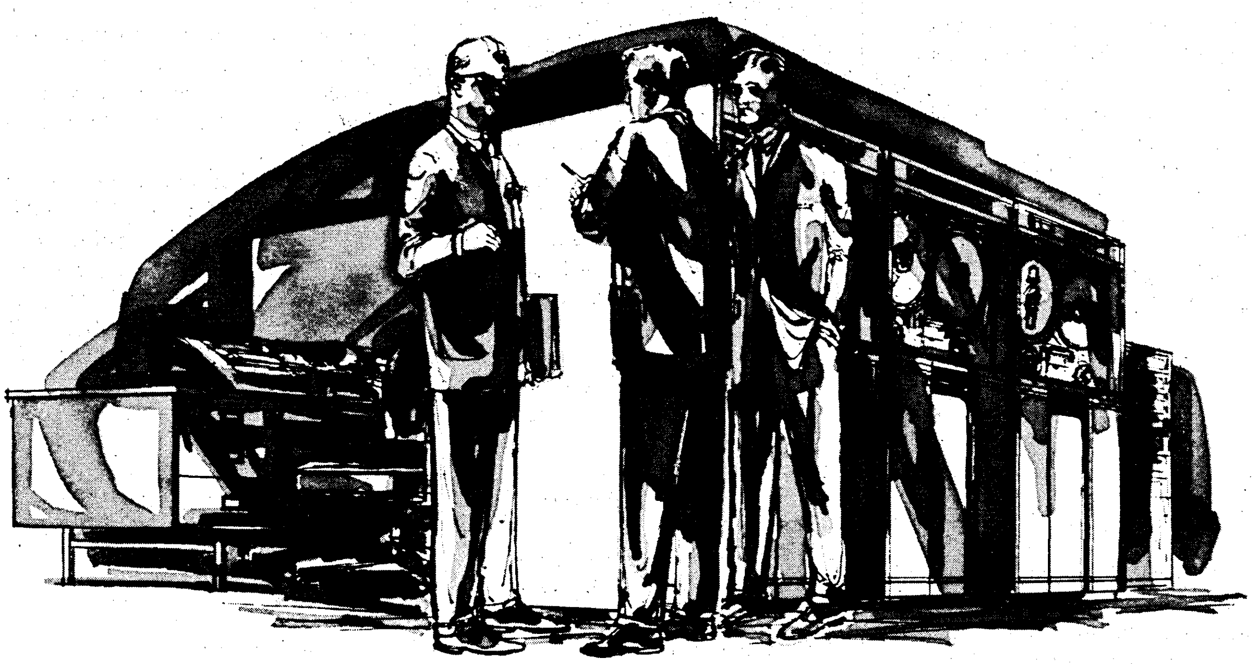
EAI®

ELECTRONIC ASSOCIATES, INC., *Long Branch, New Jersey*

ADVANCED SYSTEMS ANALYSIS AND COMPUTATION SERVICES/ANALOG COMPUTERS/HYBRID ANALOG-DIGITAL COMPUTATION EQUIPMENT/SIMULATION SYSTEMS/SCIENTIFIC AND LABORATORY INSTRUMENTS/INDUSTRIAL PROCESS CONTROL SYSTEMS/PHOTOGRAMMETRIC EQUIPMENT/RANGE INSTRUMENTATION SYSTEMS/TEST AND CHECK-OUT SYSTEMS/MILITARY AND INDUSTRIAL RESEARCH AND DEVELOPMENT SERVICES/FIELD ENGINEERING AND EQUIPMENT MAINTENANCE SERVICES.

Learn what others say about it

“ ”



then find out why they rely on MAC PANEL COMPUTER TAPE

One reason is that MAC Panel puts extra measures of quality into the production of its Heavy-Duty Computer Tape. And it's not just a white glove treatment. It's white hat, coat and shoe covers as well. Overdoing it? Maybe by some standards, but these extra steps help assure reliable tape performance. MAC Panel's gravure coating process, MAGNE-FLO coating, is another step that results in uniformly consistent signal output, better wear characteristics. In short, greater tape reliability. Your MAC Panel representative has more facts up his sleeve . . . and testimonials, too.

MAC PANEL COMPANY MAC PANEL **High Point, North Carolina**

Representatives throughout the United States, Canada, Latin America and Europe

CIRCLE 12 ON READER CARD

DATAMATION calendar

- Automatically Programmed Tools (APT) and numerical control technology will be made available to industry in a series of seminars. The first will be held on symbolic control technology, May 21-22, at the ITT Research Institute, Chicago.
- The Assn. of Management Consultants Inc. will hold its annual conference, May 21-23, at the Hotel Continental, Chicago.
- The National Telemetering Conference will take place at the Biltmore Hotel, Los Angeles, June 2-4.
- The Systems Engineering Conference and Exposition will be held concurrently at the New York Coliseum, June 8-11.
- The Univ. of Southern Calif., Los Angeles, will hold a short course on hybrid computation, June 8-19. Topics will include partial differential equations, system optimization, sampled-data control systems and man-machine systems.
- Five intensive courses on digital computers in real time, digital computer engineering, automatic programming, numerical analysis and automata theory, sponsored by the Univ. of Michigan, Ann Arbor, will be held June 8-19.
- The Graduate School of Business of the Univ. of Chicago will conduct a faculty seminar, sponsored by the Ford Foundation, June 22-July 31. Mathematical models and digital computers in business is the subject of the seminar.
- A course in Advanced Techniques of Programming Digital Computers, sponsored by Physical Sciences Extension and Engineering Extension, UCLA, Los Angeles, will be held June 22-26.
- The Fifth Joint Automatic Control Conference, sponsored by the IEEE, will be held at Stanford Univ., Stanford, Calif., June 24-26.
- The 19th annual meeting of the Association for Computing Machinery will be held at the Sheraton Hotel, Philadelphia, August 25-27.

letters

knights in real timeland

Sir:

As a Champion of the Underdog, I, Sir Modred, shall undertake the defense of the ogres slain by the gallant knight encased in Junior Steel ("The Fabulous World of Real Timeland" by T. B. Steel Jr., March, p. 24). A sword moulded from what Copi calls "the Irrelevant Conclusion" should be easy to shatter. Consider the plight of the Naive Beginner who was attempting to relate system time to "real world" time. Our hero concluded that all systems are subject to external time constraints, simply because a job must be finished in a reasonable period of time. But surely our Naive Hero should have concluded that our Beginner expected there to be a continual series of time constraints which may even be clearly defined. By an "extremely clever act," the Humble Author rendered the Naive Beginner speechless.

What of the Practical Realist? He is unfortunately not a "common fellow," for he is the one who follows the Holy Grail, PROFIT. Of course, the future is of great concern to the *successful* Practical Realist. Here the lance of Specious Reasoning is sharpened to a fine point. Is it true that if one says there are time constraints from the external environment, one implies it is hard to meet the specifications? Surely neither statement follows from the other. Perhaps the system can easily meet the external time constraints, even though these constraints are of great concern to the user.

How can we revive that slain dragon, the Knowledgeable Expert? This terrible creature was crushed by one irrelevant question: "Can we quantify the deadline and save the phenomenon?" The answer given is an incorrect "No," rather than a correct "Sometimes no." But in those cases in which one can't quantify, perhaps one can find a dependence between deadlines, or perhaps one can plan for surprise deadlines.

I am glad that the "Sin of Sloppy Thinking" has not affected Sir Galahad. I agree with him in spirit, but not in practice, that deadlines imposed by system designers are not important. But a rejection of deadlines imposed by the customer or user is tragic. This rejection is based on the assumption that the customer is a "liar" or perhaps only a "fool." Perhaps the user is a blind machine who is not interested in a "ton of listings," but is ready to use the results, if presented in Braille.

Louis A. Raphael
Scientific Data Systems
Santa Monica, California

Sir:

To restrict our thinking to the idea of a mere deadline seems very strange indeed. Rather, we must think in terms of computational rates, input rates and output rates that must be compatible with the mechanical rates of a plant.

A definition of real-time systems incorporating the idea of dynamics accepts a computer and a business corporation as such a system . . . I am sure that any company that has had a computer shift the digits on a cheque several places to the left appreciates this coupling more than Mr. Steel.

James D. Kendall
DCF Systems Limited
Malton, Ontario, Canada

Sir:

From the taxonomical jungle of Real-Timeland, Humble Author Steel has brought back, alive and squirming, a definition of "real-time" that includes payroll applications and excludes airline reservation systems. Such a definition is contrary to traditional usage, to say the least.

. . . Perhaps the proper approach toward definition would be to abstract from a sample of exemplary systems those attributes uniquely "real-time." Here are some attributes I find useful in telling the real-time systems from nonreal-timers.

1. Real-time systems are continuously operating systems. They cannot be arbitrarily interrupted without the high probability of seriously degraded performance.

2. . . . are those in which the current time (not just the date) enters as a factor into the calculations. They have to keep track of minutes and seconds, not just months and days. (An hour-counting system, then, would be a borderline case).

3. . . . are those whose inputs are automatically collected from their sources and whose outputs are auto-

matically distributed to their users.
4. . . . are those whose inputs cannot adequately be scheduled and whose outputs are constrained by response-time limitations (severe enough to make processing delays highly undesirable).

I should be very much surprised if, from this short list of attributes, there could not be constructed a useful definition, in high agreement with traditional usage, of that much abused term, "real-time."

CHRISTOPHER J. SHAW
System Development Corp.
Santa Monica, California

norbert wiener: in memoriam

Sir:

The death of Norbert Wiener in Stockholm will leave a void in the world that may not be assessed for many years. He lived as prophets always do: ignored by most in his own country; venerated by his disciples; often ridiculed by the press and its readers; and rarely believed. Outside his own country, Wiener has been revered as one of the very few very great men of this century.

We, at home, who take some pride that we revered his genius and who have known his prophecies to be the truth, shall miss the great master. We feel keenly that we have lost his guidance, his inspiration and his wisdom. But we also know that he will live as long as there is a civilization in which human beings can strive to achieve what he worked for: the human use of human beings.

ALICE MARY HILTON
New York, New York

fortran VI

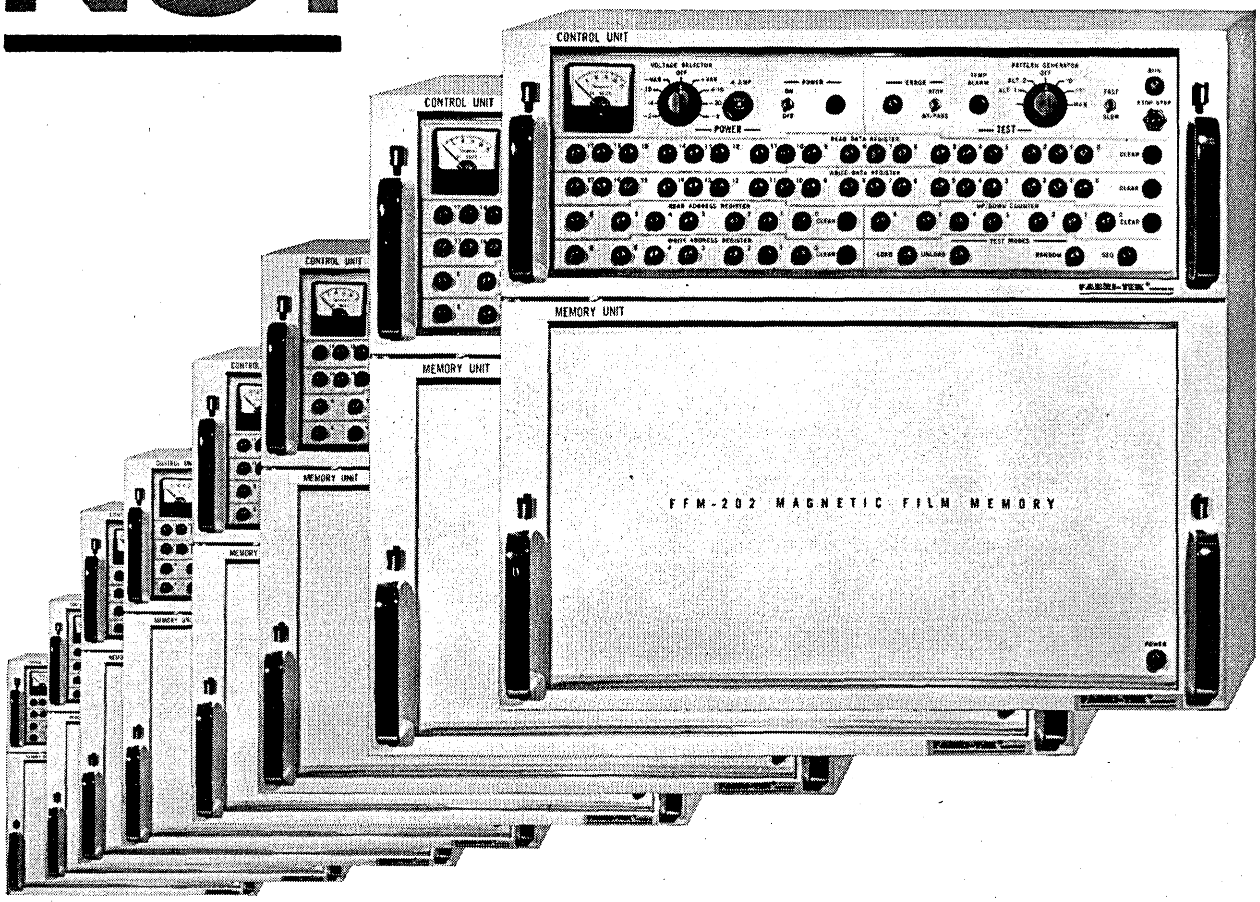
Sir:

Regarding your story on the IBM/SHARE committee (April, p. 17), on March 6, the SHARE Executive Board by unanimous resolution advised IBM as follows:

> The Executive Board has reported to the SHARE body that we look forward to the early development of a language embodying the needs that SHARE members have requested over the past 3½ years. We urge IBM to proceed with early implementation of such a language, using as a basis the report of the SHARE Advanced Language Committee.

BEN R. FADEN
For the SHARE Executive Board
North American Aviation
El Segundo, California

This is NOT a new product!



Fabri-Tek's 300-nanosecond thin-film memory system has been on the market for well over a year now, so we refuse to call it "new." Some people are just catching up to the memory system technology we offered last year, but they still haven't caught up with the Fabri-Tek Series FFM-202 Magnetic Film Memory System.

If you have such applications as scratch pad storage, index registers, real time data processing, or any of the new exotic data processing problems where high speed with reliability is needed, then here is your answer.

• 300 nanoseconds full cycle • 200 nanoseconds read or write only • 150 nanoseconds access time • Read only, write only, read-restore, read-modify-write operating modes • Capacities up to 512 words of 36 bits each • Full Fabri-Tek quality, realiability, and maintenance ease

Why wait for others to announce their new products when you can have the answer to your high-speed memory problems today. Write, call, or wire Robert E. Rife, Fabri-Tek Incorporated, Amery, Wisconsin. COngress 8-7155 (Area 715). TWX: 715-292-0900.

FABRI-TEK

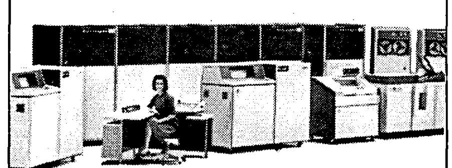
CIRCLE 17 ON READER CARD

Multi-Processing

Multi-Pr

they're starting to copy us

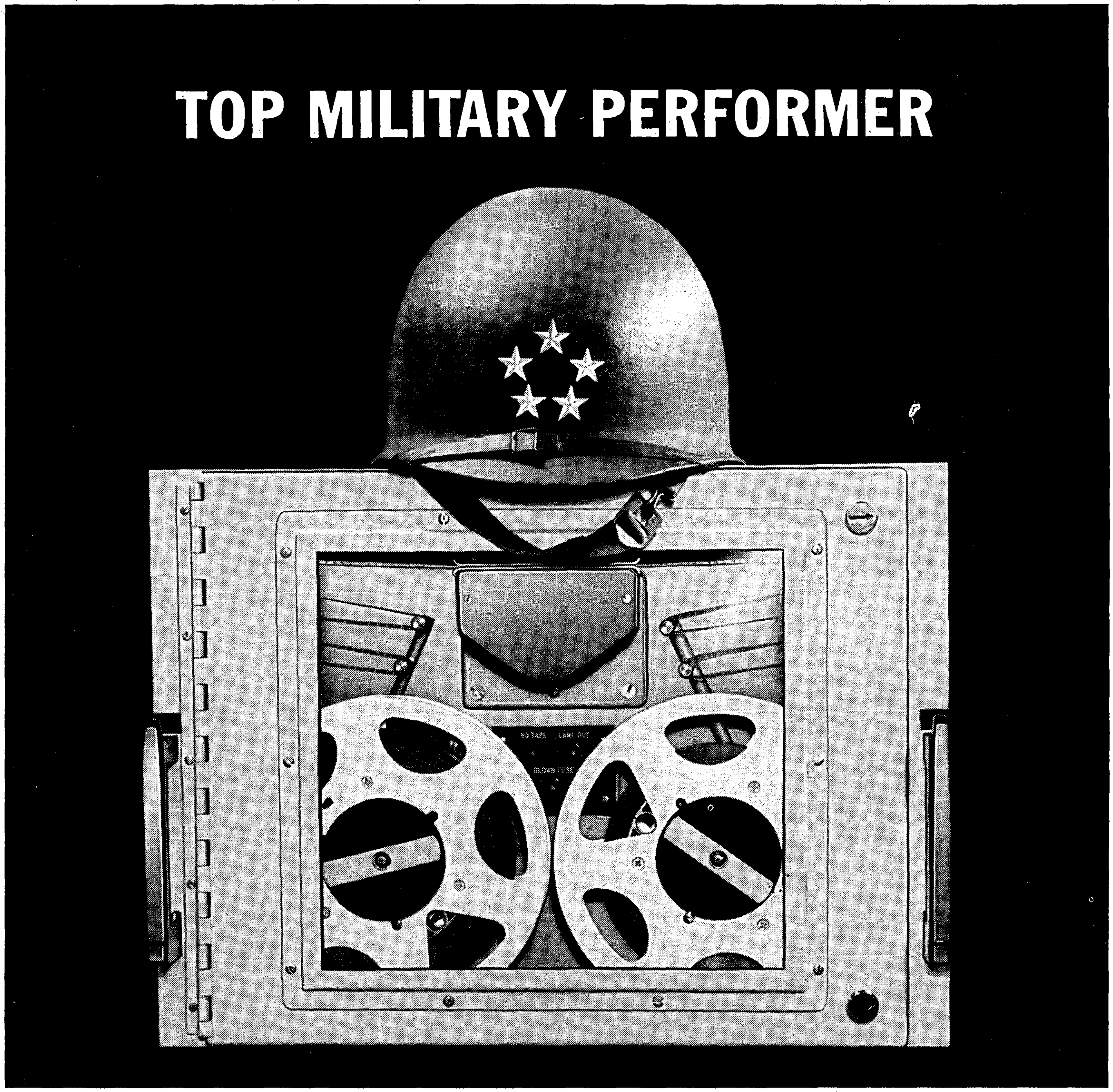
It seems that in proving to business and government that multi-processing really works, we convinced the competition too! They used to knock us. Now they're knocking themselves out trying to develop multi-processors of their own. But remember this: only the Burroughs B 200, B 5000 and D 825 are right now routinely running production programs for their users, utilizing effective multi-processing. The proof is yours for the asking.



Burroughs Corporation
Detroit, Michigan 48232

Burroughs—TM

CIRCLE 18 ON READER CARD



The new fully-militarized 500 RM Tape Reader

At last! . . . a reader conceived, designed and built specifically for military applications — the tape reader that meets all applicable MIL specs including requirements for low temperature operation.

It's Photocircuits' new high-speed 500 RM . . . the fully-militarized photoelectric reader that gives you reading speeds up to 1,000 char/sec. and 8″ reels for maximum data storage. The new 500 RM's exclusive printed-motor direct-capstan drive eliminates completely all the brake and pinch-roller problems typical of today's hybrid or conventional units — giving you accuracy plus reliability unequalled by any ordinary tape reader.

Details and specifications on the fully-militarized 500 RM are yours for the asking . . . or, if commercial/industrial applications are your interest, ask about Photocircuits' new high-speed 500 R. Write today to:

Photocircuits
CORPORATION
TAPE READER DIVISION
GLEN COVE, NEW YORK

CIRCLE 19 ON READER CARD

BUSINESS & SCIENCE

COMPUTERS, COMMUNICATIONS IN HEAD-ON CLASH

ASCII, established as an American standard code for information interchange last summer, may be headed for trouble in its implementation cycle. Although ASA's BEMA-sponsored X.3 has voted to publish a proposed standard for high-order-first transmission, low order can't be counted out yet. And even if high order gains the standards nod, it's doubtful that the federal government or the communications industry would use it.

GSA has specified low order first for a big '64 Western Union communications system; DOD has specified low order for its successor to FIELDATA; the Bureau of Standards favors low order, as do international communications interests. Even with publication, no consensus for high order is foreseen. But if it did pass, it might be a non-standard standard.

FORTRAN VI VERY MUCH ALIVE

The SHARE executive committee says its carefully worded statement to IBM (see this month's letters) on npl (new programming language, or F-6) was not tentatively negative, but "as enthusiastic as we could make it" after only three or four days of study. "It's definitely a good language ... contains nearly all we'd been asking for," says one member.

The committee hurried its recommendation to IBM so work could proceed; no floor motions were made to shoot it down, despite some criticism resulting... maybe from confusion about the aims of npl, which originated from desires to extend and improve FORTRAN. Now it looks to SHARE leaders like something quite different ... not a FORTRAN replacement, but a language for scientific applications with data processing overtones. They doubt it will prove attractive to GUIDE, a user group with stronger COBOL leanings.

The npl specs were made available to ASA's X.3.4 programming languages subcommittee, but even serious preliminary standards study is months away. Although SHARE says it did not consider the question of standards implications in making its evaluation, the implications loom large. IBM competitors must now decide to push implementation of npl for their machines ... or wait to see its fate as the third language for the 360. Decisions, decisions.

360 TRIGGERS ANNOUNCEMENTS, NEW MARKETING GIMMICKS

Evidence of 360's impact was strong at the SJCC, where several new machines were unveiled, including 3 C's DDP-224 and DEC's PDP-7. The 7, featuring a 1.75-usec memory, is program-compatible with the PDP-4, sells for maybe 10% more. With a $95K price

tag for a typical system, the 7 will compete with the likes of the 224 ... SDS 920 and 930.

GE, meanwhile, privately pushed its new family, the "600/compatibles," featuring new airborne (A-605) and military machines (M-605, M-625), to go along with the previously leaked 625 & 635. GE hopes the 600 will help it capture a hunk of the command & control market, estimated at 900 megabucks/yr by '65 and $1.2-billion by 1970.

Honeywell was busy. Besides demonstrating its 200 Liberator and discussing the new 1.75-usec-memory 300, the company set up a weather service. A recorded message at a NYC phone number predicted sunny skies over Washington, "clearing fog." Earlier messages had been more pointed: "Temperatures of minus 360 degrees in Poughkeepsie," falling due to a cold front caused by customers asking if the 360 was designed to satisfy their needs or the manufacturer's. Privately Honeywell says it will be building 60 200's a month by the end of the year.

SJCC SOCIETY GLEANINGS

AFIPS re-elected its officers (chairman is Don Madden), decided to continue its search for a voice-of-the-industry secretary. ... ACM heard the report of its committee of thoughtful persons, who think the Association might want to change its name, also start worrying about dwindling membership. Why? Fear that time sharing, other new techniques, will automate the programmer out of existence. The answer: an education program to upgrade applications programmers to systems programmers. ... The IEEE merged the old computer committees of its founding societies (IRE, AIEE) into a new one: Society for Computer Sciences, headed by Keith Uncapher. ... ASA's X.3.4 subcommittee made a recommendation that the U.S. representative to the forthcoming international standards meeting recommend concurrent and equal-favor development of ALGOL, COBOL and FORTRAN standards. ... The recently revised COBOL committee may be revised again.

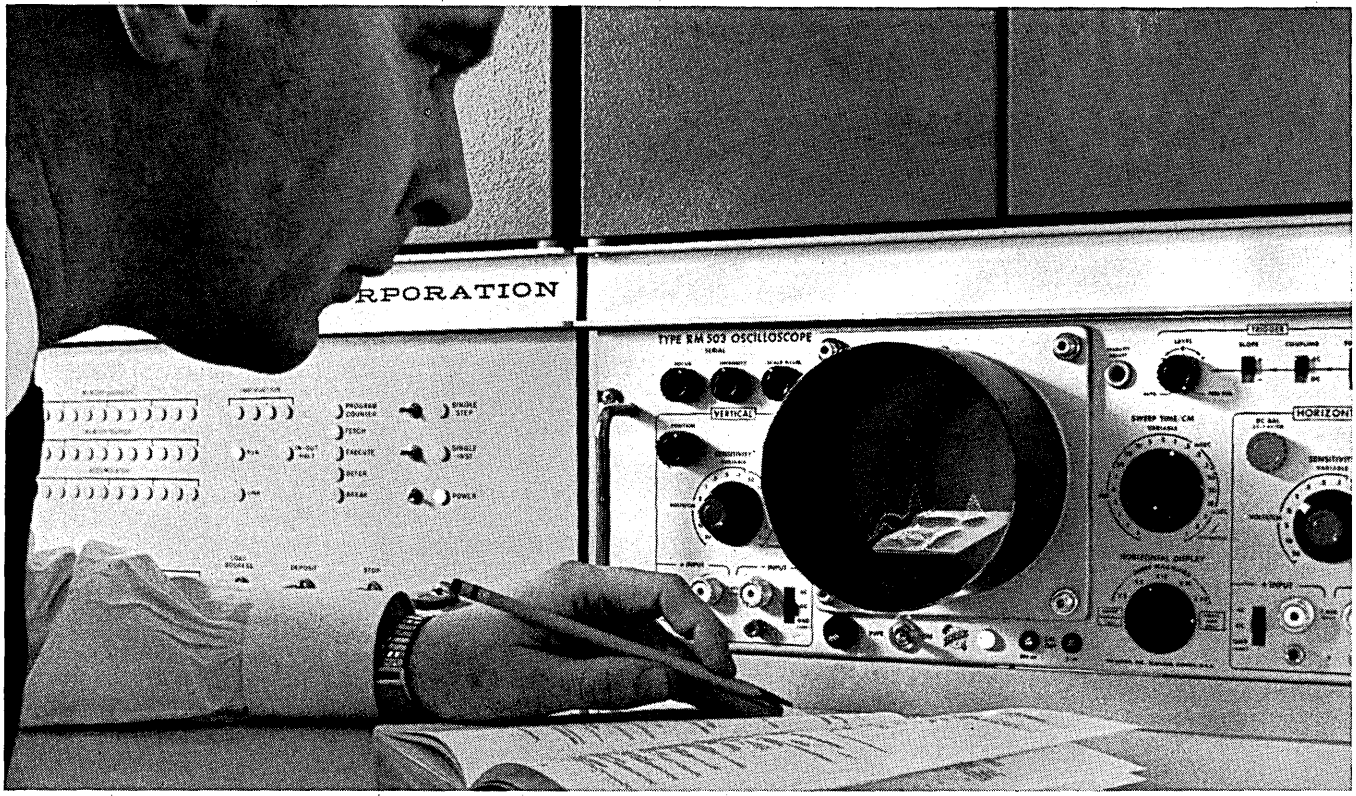
NEW FACES

EA Industrial Group,NYC, has been formed to design, manufacture and peddle industrial control systems. The firm is also sole U.S. distributor for the computer products of England's Elliott Automation. Now quoting Elliott computers in its bids, EAIG plans to announce its own digital control computer "soon." Now 10 people strong, the firm will manufacture out of LA.

Another new firm, U.S. Magnetic Tape, Chicago, is getting ready to market a new "super tape."

RUMORS AND RAW RANDOM DATA

Digital Equipment Corp. says it's sold more PDP-6's since the 360 announcement than before ... Ampex will build an IBM-compatible nine-channel tape unit which should be available this year. Look for all else to follow suit. ... In addition to starting its own institute, CEIR will offer to market products of information processing firms. The deal would include sales promotion, advertising and PR, would represent the industry's first nationwide rep effort. ... IBM is negotiating for contracts to build two super 360's (mod 90), probably for the AEC. ... A DEC employee is said to have met a minister in an Australian pub who asked him his line. When the American said computers, the minister said, "My grandfather was in that line." That's right, he meant Charles Babbage. ... One 7090 user estimates it might cost him $3-million at each of three installations to reprogram for the 360.


RPORATION
TYPE RM 503 OSCILLOSCOPE

WASHINGTON REPORT

THE SERIES 360: SALES BY OSMOSIS

No wild enthusiasm was kindled among Government computer users by IBM's 360 announcement. The 18 to 24 month hiatus before any hardware will be seen made for a somewhat soggy initial reaction. But Government folk, like their commercial counterparts, are well attuned psychologically to acceptance of IBM's latest as the "logical next step."

One sidelight: The 360's modularity features may in part answer cries of "oversold" that frequently emanate from the General Accounting Office. It's even possible that leasing, a bad word lately around Government agencies, might come back in style.

Will the transition to the 360 be automatic? That's probably the way to bet, but since the 360 is such a departure from its immediate predecessors, especially in software, a number of Government users are thinking radical thoughts about other manufacturers. "When the time comes to turn in our 7094," noted one astute Federal computer shop operator, "our operating continuity will be disrupted in any event by the 360, so we may just as well look over what the other manufacturers have got to offer. Maybe we'll find a computer that'll do our job better."

The openminded approach, however, is likely to be confined to the more sophisticated scientific-type users within the Government.

NEW TARGET FOR GAO

The General Accounting Office has focused its gimlet eyes on the costs borne by the Federal Government on behalf of Government contractors who maintain large in-house computing complexes and bill their costs back to Uncle Sam. Ordinarily the expense involved in maintaining these computers is buried in overall project costs, but GAO gumshoes have been dredging up some of the fiscal facts. Big dough is involved. People who ought to know say that the amount of money spent on computers in this fashion by the Government exceeds the amount it spends directly for EDP gear, which places the figure in the $700-million-annually ball park.

First large contractor to be tied to the stake and baited on these charges in a GAO report was Martin Marietta Corporation, at whose Aerospace Division facilities GAO claims the Government will be paying unnecessary costs of $7.7 million over a five-year period, $13 million over a six-year period. This, says GAO, is because Martin Marietta leased, rather than bought, several oodles of IBM equipment. ''Nonsense,'' Martin Marietta has in effect answered, but the Congressional watchdog agency remains convinced it's on the trail of something big.

The GAO report also contains its usual recommendation that the President establish a central source, facility, agency or something for Government computer acquisitions and their utilization. H. R. 5171 – the Brooks bill – meanwhile languishes in

Continued on page 101

EDITOR'S READOUT

LEST WE FORGET

SEAC retired last month with all the fanfare befitting the last official public appearance of a vigorous teen-age pioneer. Put to work at the National Bureau of Standards in 1950 (what were *you* contributing to computing then?), SEAC has since performed nobly, with dutiful precision – if not always with grace and elegance – in a wide range (perhaps not quite 360°) of important problem solving and information processing experimentation. SEAC graduates constitute a Who's Who of numerical analysis and computing. The machine helped in the development of linear programming . . . and the H-bomb.

The farewell party was fun. Old SEAC friends told stories that would have made a human retiree laugh and cry. Like the day they proclaimed a "Very Good Friday" because "SEAC can add!" The next Monday, said Dr. Ralph Slutz, they discovered "it couldn't subtract." This was corrected, as was the strange malfunctioning characteristic which could be resolved only by jumping on the wooden floor supporting SEAC. There was the day the typewriter ran out of paper. So the printout of prime numbers was recorded on toilet paper.

To those who worked with the machine at all hours of the day, SEAC took on a human personality . . . female, of course. She became the "good old girl" who performed faithfully even though her adder was designed backward and upside down, whose weak joints sometimes made jiggling fatal.

Ida Rhodes, first studying the description of the machine, couldn't figure out the meaning of the Greek letter mu before the word "seconds." The week that was to be devoted to a study of a "scatter-brained scheme" stretched to three years, and a beautiful relationship with "that beautiful, precious, precocious SEAC . . . a little darling who performed beautifully from the start."

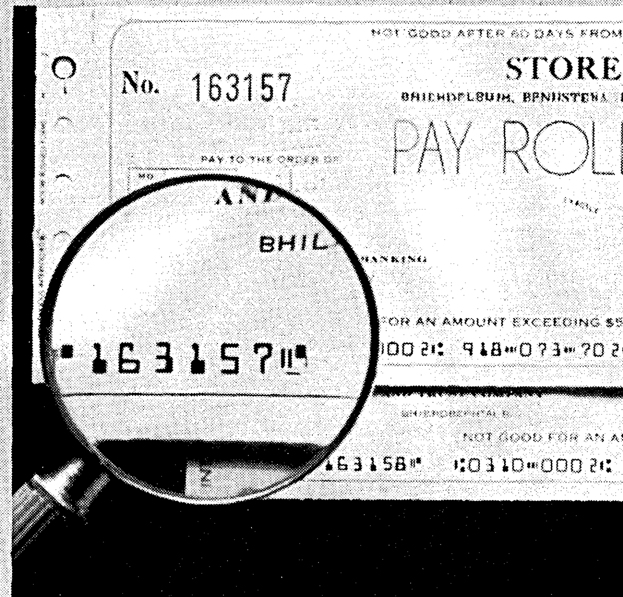
"Bewildered, bewitched and bothered" by what she called a miraculous birth, Ida felt it was "a mirage, a trick . . . that her senses were being duped. But," she added, "it has been for the past 15 years a most diligent, faithful and competent servant." And it taught her that the efforts of dedicated people can create what their senses tell them is impossible.

To this enchanted observer of SEAC's swan song, the ceremony underlined a couple of key ideas. John Todd, chief of the computer lab at NBS in SEAC's early days, stressed one of them when he pointed out that the first computer user was an engineer or mathematician. Today he is a bank manager. He described modern computing as a "mail order business," fit for "spades and shovels" . . . and added that "the best and most interesting work will be done with the mathematician close to the console."

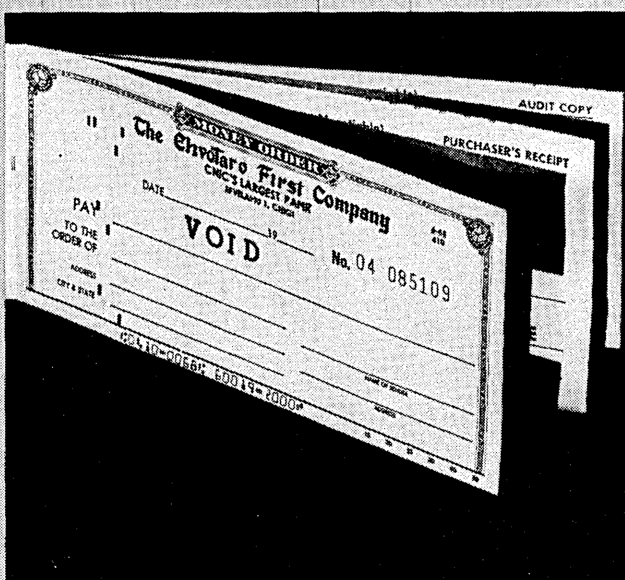
Time-sharing and the remote console open a door to a closer relationship between man and machine than has been possible lately. But we doubt that it will ever approach the good old days. But if it allows the people with the problem to do some direct, dynamic, creative work without hours of coding or educating a programmer, it will be a step backwards in the right direction.

The farewell party re-enforced another of our pet notions: that it's the people behind the machine who count. We learned that the bright, creative people who built, developed and tutored SEAC were especially warm human beings . . . not the stiff, learned stereotype egghead of popular legend. To these people – to the Sam Alexanders and Ida Rhodes' – the computing profession owes continuing gratitude.

Forms that drive machines



MOORE MCR consecutive numbers and encoding are precise, for accurate reading and MCR sorting. Minimizes rejects and errors in runs.



MOORE TAB CARD SETS provide precision-printed source documents that are machineable, to automate bookkeeping, filing, and reference.

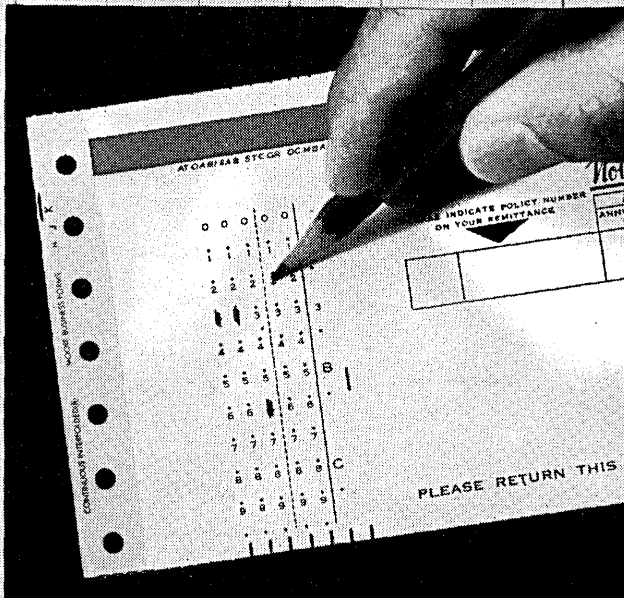
QUALITY MANUFACTURE helps cut downtime, mis-reads, errors. This saves your staff's time, promotes efficiency, and speeds work flow.

Data processing is speeded up when source documents are used as input to activate machines, whether by MCR, OCR, or punched cards. Extremely close tolerances in input documents are a 'must' for perfect results.

Moore forms are manufactured to serve as accurate input media for all types of data processing equipment. For example, Moore Tab Card forms are made with the grain long, offering maximum rigidity along card length, affording greatest dimensional stability.

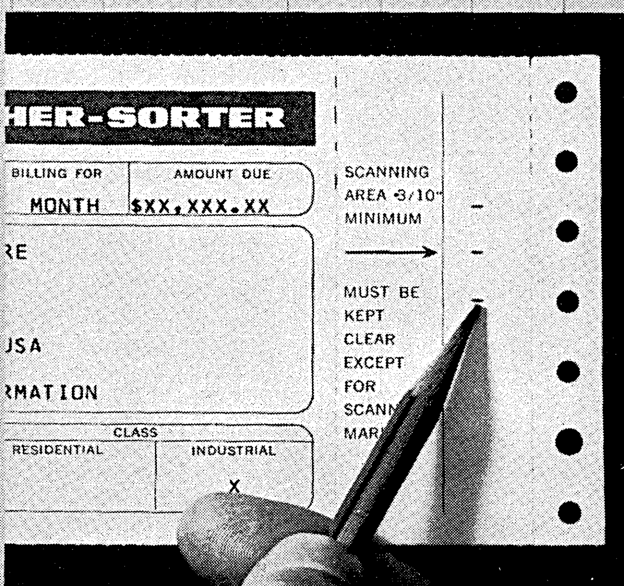
Moore MCR printing meets exacting ABA specifications for accuracy in reading-sorting operations. OCR forms made by Moore meet the highest standards for correct readings by all optical scanning equipment.

Moore precision products are the result of years of



MOORE OCR CHARACTER PRINTING and mark-read positioning provide accurate readings and processing by optical scanning equipment.

MOORE GRAIN-LONG CARDS provide for efficient preparation of statements, plus a returnable voucher for direct input into the system.



FOR SPECIALIZED EQUIPMENT such as the Detacher-Sorter, Moore forms eliminate manual sorting, improve forms-handling efficiency.

with Moore quality built in

experience in practically every type of form construction. 32 modern plants, close to you for service, have the most modern equipment, the most knowledgeable personnel.

The Moore man knows equipment and the proper form construction for each machine. Rely on him for recommendations on forms and forms systems to get the utmost from your equipment.

Ask the Moore man for suggestions—he can help. If you work with forms, we can show you how to make forms work for you.

'The right business form for every form of business'

Niagara Falls, New York • Park Ridge, Ill. • Denton, Texas
Emeryville, Calif. • Over 500 offices and factories in North America

MOORE BUSINESS FORMS INC

CIRCLE 23 ON READER CARD

OPERATING SYSTEMS

boon or boondoggle?

by T. B. STEEL, JR.

When a group of computing old timers gathers together, the conversation often turns on some harrowing experience of console debugging. Viewed through the mists of eight or 10 years such experiences tend to take on a poetic character. One is led to the conclusion that only here is found true *rapport* between men and machines; only in this way is real computing done. There is a key truth in this point of view and we shall explore it in a moment. Naive advocacy of this position, however, ignores an elemental fact of life. Scores of man-hours can be purchased for the cost of one machine hour.

Operating systems were born from this observation. The sole purpose of early operating systems was elimination of idle time. Modern operating systems attempt a great deal more than minimization of idle time, and the history of the development of operating systems is largely a story of this changing philosophy. To see this development in perspective it is important to keep in mind the original problem.

Even a casual observer could easily discern the primary sources of idle time in an environment where the programmer marched into the machine room with his card decks and listings, preparatory to an extended session of playing with the console keyboard. First, an inordinate amount of time was wasted whenever the machine hung up and the programmer scratched his head, trying to figure out what to do next. Second, nothing at all happened—not even constructive thinking—when a job was done. The departing programmer had to gather all his material and insure that nothing was left undone. The new user had to get himself and his material properly emplaced.[1] The growth of problem size and the attendant increase of magnetic tape usage further complicated the problem of logistics.

The initial step toward a solution to this problem was obvious: get the programmer back to his coding sheets by hiring professional machine operators and instructing them to take a standard action on unanticipated machine stops. This helped by streamlining the logistics and eliminating head scratching at the console,[2] but even operators took a noticeable amount of time to remove one job and initiate another. The next step—automation of as many of the operators' functions as possible—is apparent today; in

Mr. Steel is a computer systems scientist in the Advanced Technology & Research Directorate of System Development Corp., Santa Monica, Calif., and a senior research leader of the Information Processing Research staff. He is particularly concerned with applications of mathematical logic to dp theory. He is AFIPS rep to the IFIP TC-2 Programming Language committee, and secretary of X3.4 Common Programming Languages subcommittee. He holds a BA and MA in math from U. of California.

[1] It was during this hasty transfer of authority that the worst calamity would occur—a dropped card deck (often unsequenced). If this deck belonged to the oncoming user, idle time mushroomed. Paper tape has its virtues.

[2] It certainly helped to cut down the time that the run light was off. It is problematical whether it really cut cost. To the author's knowledge no study was ever made to see if the machine time saved was worth the cost of the operators. It doesn't really matter because total machine time available had already become a problem by the time operators were widely employed.

the early fifties its conception was an imaginative accomplishment.

A corollary for the automation of the machine operator is the necessity for rigid control of access to the machine console. As might be expected, this has caused a great deal of complaint. Most of the objections come from programmers who feel that they are unable to debug their programs without intimate interaction with the machine. Experience proves this false. Also lost, however, is the capability for a user to monitor the progress of calculation in his problem. In certain cases this loss is keenly felt and cannot be repaired by programming alone until heuristic programming or some other aspect of artificial intelligence technology succeeds in reproducing human decision-making in problem-solving situations.

Assignment of priority for the invention of an idea has been a hazardous undertaking since the time of Newton. The author doesn't know—and really never expects to know—who first thought of an operating system. First thoughts are unimportant in this sort of business anyway. The man who counts is the one who implements and publicizes an idea. Perhaps the first serious semi-public discussion of the concept took place at an informal meeting of 701 users in Herb Grosch's hotel room during the 1953 Eastern Joint Computer Conference in Washington. Among those present were representatives from the General Motors Research Laboratories, the installation that can claim the credit (or assume the blame)[3] for introducing the operating system to the computing world.

General Motors developed and used an elementary operating system late in their 701 experience. The effect of the introduction of this system on their machine room productivity was such that they felt compelled to construct a similar system for their next machine, a 704. The Los Angeles Division of North American Aviation had been impressed with the effectiveness of the GM system and was also planning to obtain a 704. In 1955 the two installations joined forces to produce an operating system for the 704. The unparalleled success of the 704 spread the GM-NAA concepts into all corners of computing through the medium of the concurrently developing SHARE organization. Soon, most 704 installations had operating systems of their own, and not long thereafter the idea jumped the boundaries of machine type. Today operating systems are used almost universally in connection with large computers.

This rapid dispersal of the operating system concept is a striking example of the influence of cooperation and communication on the information processing business. In the absence of SHARE and the other user groups, a few operating systems would have been developed by individual installations in splendid isolation, but the suddenly acquired ubiquity of the notion would not have been forthcoming. Aside from the user groups there are two other potentially unifying forces in the information processing community: the professional societies and the manufacturers. By and large the pragmatic orientation of operating system development has prevented interest in them from breaking through the theoretical bias of the professional societies, and the manufacturers have been quite slow in understanding the requirement.[4] The original problem attacked by operating systems—minimizing idle time—is not really a computing problem at all; rather, it is a problem of installation management. It is a users' problem and was solved by users.

The basic idea of a primitive operating system aimed at the elimination of idle time is job stacking (or batching).[5] Rather than being loaded into the machine independently and immediately executed, a collection of jobs is gathered into an input batch and the programs necessary for each job, together with the relevant input, are *all* loaded onto an input file. There is a program, normally kept in the main store, whose function is to load the next job from the input file, halting only when there are no more jobs to process. All jobs are required, upon completion, to transfer control to this program, and the machine operators are instructed to execute the appropriate manual transfer if anything goes wrong. Aside from trivia, like a routine that sets all of the unused portion of the main store in such a fashion that a wild transfer of control in some job causes the next job to be brought in automatically, nothing else can be done to dispose of *inherent* idle time. The rest of the paraphernalia of a modern operating system is designed to serve different purposes.

In addition to inherent idle time, deriving from the fact that people and machines perform at different rates, there is another kind of idle time which can plague a computing installation. It comes from conflicts that arise due to the independent decisions made by programmers in the selection of machine components. A simple example of such a situation is the case of programmer N + 1 who wants to use tape unit A as an intermediate scratch tape just after programmer N has used the same unit as an output tape. While operators run about dismounting and mounting tapes—or, at the very least, changing switches to rename tape units—everything in the machine must come to a halt.

Given just the simple loading program described above, however, the only thing standing in the way of the virtual banishment of idle time is this complex logistics of input and output. In order to make an operating system function effectively, it is necessary that each program conform to a set of input-output constraints in the interest of prevention of conflicts and the possible destruction of files. The rules governing these constraints are complicated, difficult to explain and impossible to enforce by management fiat. The solution is simple. An uncomplicated, obvious and enforceable rule is adopted to the effect that *all* input and output is done through standard routines that are part of the operating system. In this way the constraints are embodied in a single program and universally applied. Additionally, since the bulk of machine failures in the past decade have manifested themselves in magnetic tape systems, centralized control of all input and output permits standard and effective recovery procedures.[6]

An operating system that minimizes idle time through

[3] The possibility that operating systems are a mistake is a serious one and will be dealt with below. That they are a mixed blessing is unexceptionable as the current furor over turn-around time makes clear.

[4] There are facts to support this contention. For example, at the beginning of 1956 three 704 customers—RAND, Lockheed, and North American—pooled their free checkout time to improve its utilization. The IBM people were startled at the idea of the simple program written on the spot that allowed the stacking of assemblies so that operator intervention was not required from one to the next. If such a trivial departure from stacking input to an applications program was surprising, it is no wonder that the more sophisticated idea of stacking whole *jobs* did not occur to them.

[5] "Batching" is really the better term. Commonly, jobs are stacked on a serial file and executed in the order of loading. With the advent of random access mass stores, a batch of jobs can be deposited in the store and executed according to some priority scheme that is quite independent of the order of loading. Thus, the job file ceases to be an ordered stack and becomes an unordered batch.

[6] This is not *logically* true. The individual programmer, knowing what his own program is up to, can, in principle, do a better job of error recovery than can a standard routine. The trouble is that the programmers usually don't bother to try.

a job sequencer and centralized input-output has accomplished its assigned task. Such a system provides a bonus value, however, for it makes possible a number of things that could not sensibly be done in its absence. This is a direct consequence of the fact that an operating system provides a focal point inside the machine for capitalizing on information derived from the inter-job transition. For example, if a machine is equipped with a real-time clock, appropriate routines can automatically log jobs, compute machine time charges and prepare machine utilization reports. Also, on a machine with parallel input-output channels, it is often possible for input of the next job to begin prior to completion of output for the last job. While this sort of thing may seem old hat to the reader today, it was far from obvious not so long ago and, furthermore, was (and still is) hard to do right.

Channeling all jobs through a single focus permits other economies and procedural simplifications. An obvious and almost universally employed strategem is the collection of frequently employed service routines, such as assembly programs and compilers, into a single file. The principal virtue of this procedure is not to save loading time,[7] but rather to provide a single, common copy of each service program. Thus, everyone uses the same versions of the service routines, modification and maintenance being reduced, thereby, from a hopeless nightmare to an unpleasant chore. Additionally, establishment of a system file leads to the possibility of *phasing*.

Phasing is perhaps the least understood and most maligned aspect of operating system technology. The essentials of the concept are simple. First, input translation—assembling, compiling, decimal to binary data conversion, etc.—is done for *all* jobs in a batch, the results going to an intermediate file. Then, execution of all jobs takes place with all results recorded in binary form on another intermediate file. Finally, output translation—binary to decimal data conversion, formatting, etc.—is done for all jobs in the batch.[8] The original purpose of phasing was to save storage space. Input and output routines seem to consume a great deal of space and on machines with small stores it is possible to free a large part of the main store for calculation by relegating the input and output portions of the program to separate phases of processing. This gain is paid for, of course, by addition of the time required to write and subsequently read the intermediate file. As main stores have increased in size, the marginal gain from phasing has become less important. We might say that phasing is a passing phase.

Before leaving the subject of phasing it is worth noting two additional matters. First, it is possible to save a fair amount of time in the input and output phases if the pertinent service routines can *all be kept in storage*, thus eliminating repetitive readings of the system file. To my knowledge this has never been successfully done for an input phase, largely due to the inordinate size of compilers at the time of major interest in phasing. It has, however, been done for output translation and proven quite effective. The second observation is that phasing represents the high point (or low point, perhaps) of the divorce between user and program. In a phased system there is no longer a continuing span of time during which a given user's job is run. It is this fact more than anything else that led to loud objections to the whole concept of phasing.

Those aspects of operating systems discussed above have all dealt with the problems of running a job shop, and, as such, are concerned with machine efficiency and management information gathering. Recent operating systems have been cloaked in a panoply of devices aimed at aiding the programmer. Thus, routines whose original purpose was to handle input-output logistics have been turned into input-output control systems that virtually eliminate the need for a programmer to know anything about machine input-output. Integration of compilers and the loading routines of operating systems has provided some storage allocation capability. The vast array of debugging aids incorporated into today's operating systems is an attempt to atone for pushing the programmer out of the machine room.

Today, operating systems are being developed to deal with all the problems mentioned above and, in addition, are being designed to handle the requirements of new hardware capabilities—direct data inputs, miscellaneous interrupt conditions, multiprocessing, and so forth. It should be evident that a modern computing installation—at least to the extent that it is a job shop—could not function effectively in the old fashioned environment. This does not mean that operating systems are wholly beneficial, however. The problem of removing the man from the machine has already been mentioned. Associated with this is the problem of turn-around. In many shops it takes far too long for a job to pass through the system, and we begin to see the converse of the problem operating systems were designed to solve. A great deal of man time is now expended to gain a little machine time. Finally, to date operating systems have been invariably ad hoc programs, highly machine-dependent.

Solutions to both problems are on the horizon. The notion of on-line programming—providing the user with direct access to the machine—made economical by multiplexing users (time sharing), is rapidly gaining adherents. It promises to bring back the old *rapport* between the user and his program without the associated inefficiencies. Also, operating system functions are not as machine-dependent as one might suppose. Work is now underway to implement an *almost*[9] machine-independent operating system. So far there seems to be no work on a machine-independent, time sharing, on-line system but there is no compelling reason why such a system can't be built. I will stick my neck out far enough to predict that this will be the wave of the future, with time sharing coming first.

Now we can return, briefly, to the question, "Are operating systems a good thing?" It is now my belief that they are not—to the extent that the emphasis on job batching has taken the user away from interaction with his program. They have certainly been a *necessary* evil, however, machine costs being what they are. Perhaps the traditional operating system has been an essential detour in the attainment of efficient on-line operation, for this too has its control system and, while different from a batch job sequencer, is no less an overhead program common to all users. And it seems certain that large machines will rarely be used hereafter without *some* central control program. ■

---

[7] Whether the service routine is on a system file or on the input file has no bearing, generally, on the time it requires for loading. Time *is* saved, of course, in the peripheral activity of originally loading the input file.

[8] The emphasis on binary in this discussion suggests that phasing only makes sense on binary machines. This is not entirely true, although phasing was invented for binary machines and is more advantageous on them.

[9] "Almost" means that approximately 95% of the lines of code in the system be in a machine-independent procedure language and will not require change for a new machine.

"admiral"

OPERATING SYSTEM FOR THE 800/1800

by JACQUES BOUVARD

The development of an operating system for a large scale digital computer is a challenging undertaking. The designer has to cope with a variety of so-called "desirable" characteristics which too often appear partially conflicting. Underlying these conflicts is the fact that the operation of a computer brings into play three separate, but intimately related, elements—namely, the computer user, the machine and the software. Each of these elements has its own requirements, capabilities and limitations.

Recognizing that the computer is essentially a tool whose problem-solving capabilities are enhanced by the software, the basic task of the operating system is to help the user in exploiting the capabilities of the hardware-software complex. On a standard computer, this is achieved by relieving the human operator of the burden of detailed execution supervision. The operating system provides for automatic transition among the successive jobs to be processed. Automatic program set-up involves library search, loading into memory, data distribution and output collection. The need for operator intervention is further reduced by the availability of debugging aids, restart facilities, logging procedures, etc.

On a computer equipped with built-in multiprogramming capabilities, the functions of the operating system go far beyond these standard services. Automatic time-sharing of the central processor and input-output facilities among several programs eliminates the imbalance between processing and input-output activities. Sharing of memory space

Mr. Bouvard is a staff analyst in the programming systems division of Honeywell EDP, Wellesley Hills, Mass. His activities there include design and implementation of the Executive system for the 800 and 1800, systems analyst for FACT, and project manager for Admiral. He was recently appointed project manager for the development of all software for the H-200. He holds an MS in electrical engineering from the Univ. of Paris, and an MA in math from Harvard.

The Honeywell 800/1800 systems consist of a central processor, magnetic tape drives, control units, peripheral input-output devices and an operator's console. The central processor includes core storage, arithmetic and control sections, and a traffic control and multi-program control capability for parallel processing of up to eight programs simultaneously.

THE HONEYWELL 800/1800 SYSTEMS .....

Traffic control is a device which monitors all peripheral activities and provides proper channel connections at proper times between central processor and peripheral units. Up to eight input and eight output channels may be operated simultaneously. Multi-program control coordinates demands of up to eight completely independent programs running simultaneously and in parallel.

Parallel processing directs time-sharing for up to eight active programs. Each program may start, proceed and stop independently of other programs. The total gain in efficiency of this technique will depend on the nature of the programs being run simultaneously.

Parallel processing is a hardware feature of the central processors of both the H-800 and H-1800 systems. The H-800 central processor has a minimum of 4,096 words of core storage, the H-1800 a minimum of 8,192 words. (A Honeywell word consists of 54 bits, six of which are used for checking, 48 for information in normal usage). Both central processors also contain a 256 register control memory, which is divided into eight identical groups of 32 special registers. The special registers have numerous functions, including word masking, indexed addressing, indirect addressing, and supervision of parallel processing tasks.

Memory speeds for both core and control memories is two microseconds per word in the H-1800 and six microseconds per word in the H-800, which is equivalent to 120,000 three-address instructions per second and 30,000 three-address instructions per second, respectively.

The systems, while designed primarily for business computation, also possess scientific capabilities, including a floating point unit that operates at nanosecond speeds.

and external devices provides for the absorption of any excess capacity and eliminates idle system time during inter-job setup. Finally, parallel processing permits the complete integration of the system so that it can be effectively coordinated from a single location. This simplifies the overall system operation and increases the equipment utilization. The operating system thus becomes a computer management tool capable of automatically controlling the processing of the work load while optimizing the hardware utilization. At the same time, the operating system must remain sufficiently flexible so that it can adapt itself to the operating policies of the computer user; it must be capable of responding instantaneously to any command issued by the human operator.

This quality of adaptability to the operating environment must also apply with respect to the hardware and software characteristics. The operating system must be capable of operating on any system configuration. It must also be compatible with all available programming systems without imposing upon them any artificial restriction which could impair their efficiency or ability to perform a job. Finally, all of these characteristics must be implemented at the lowest possible cost in terms of hardware and computer time overhead.

the admiral operating system

These considerations have been used as guidelines in the development of the new ADMIRAL operating system for the Honeywell 800/1800. Both of these systems embody the concept of parallel processing as a standard hardware feature. Earlier developments in the field of multiprogram operating systems include the Honeywell Executive System. This initial package, designed for a minimum H-800 configuration, utilizes the static scheduling approach. In this system, the schedule is prepared during a preliminary phase. During execution, a small supervisory routine controls the processing of the work load according to the predetermined schedule.

Another multiprogram operating system developed at the Metropolitan Life Insurance Co. embodies the concept of dynamic scheduling. In this scheme, scheduling is performed during execution and thus the work load need not be specified ahead of time. This approach provides for a more flexible operation at the expense of a somewhat larger supervisory routine.

With the availability of larger H-800 system configurations and with the announcement of the H-1800, the development of a more sophisticated operating system called ADMIRAL was undertaken at Honeywell.

In an attempt to create a system well fitted to the user needs, preliminary specifications of the system were drawn by a design committee and reviewed with some 13 large Honeywell 800 users and potential Honeywell 1800 users. Following a full month of marketing survey, recommendations were submitted to a design and implementation team which drafted final specifications, incorporating many of the user suggestions.

dynamic scheduling

ADMIRAL's major function is to schedule the processing of the workload to continuously maintain maximum utilization of the available equipment configuration. This is achieved by always attempting to saturate the system, i.e., by keeping as many programs as possible in simultaneous operation. The scheduling algorithm must, however, recognize a number of practical considerations:

1. Dependence. To conform to the natural flow of the work load, ADMIRAL recognizes the concept of jobs. Each job consists of a series of programs which depend upon each other for input-output. Programs within jobs can only become eligible for execution once all of their logical precedents have been completed.

2. Urgency. The relative urgency of each job is described by means of a relative priority coefficient assigned by the human operator. These coefficients govern the order in which jobs are examined by the scheduling function for selection and execution.

3. Hardware Requirements. Another feasibility criterion recognized by the scheduling algorithm is that, at any instant, the total amount of equipment allocated to the programs in parallel operation cannot exceed the total available configuration.

At the beginning of a shift or at any time during execution, the operator may submit new job requests. Job descriptors, usually in the form of a small card deck, are immediately input and stacked into an internal queue known as a job list.

The scheduling function is activated each time a change occurs affecting the job list or the system utilization. The job list is then examined, one job at a time, by order of priority.

The hardware requirements of the first program in each job are determined and matched against the current available equipment. If enough equipment is available, the job is selected for execution and its first program is started. If, however, a conflict develops between the hardware requirements of the first program in a job and the current unused system capacity, the job is passed over and the scheduling function proceeds to inspect lower priority jobs. To avoid permanently bypassing a job with large hardware requirements, each time a job is passed over, its priority is automatically raised, up to a maximum priority. When such a job exists in the job list, further scheduling is momentarily suspended until sufficient equipment is released back to the system to accommodate that job.

Another important consideration is that once a job has been started, it should be completed as rapidly as possible. For this reason, whenever the scheduling function is activated following the completion of a program, it first attempts to select another eligible program from the jobs in current execution. If and only if all eligible programs in the current jobs have been started, the scheduling function considers selecting new jobs from the job list by order of priority. This method prevents starting too many jobs at once and having to interrupt jobs frequently during execution, which might entail a considerable amount of tape file saving and remounting operations.

Processing of high priority jobs with little or no delay implies the capability to interrupt jobs having lesser priority and conflicting equipment requirements. This is achieved by means of so-called absolute priorities which are recognized by the scheduling function. There are two types of absolute priorities: TOP and SUPER.

TOP priority jobs are immediately selected from the job list and started as soon as enough equipment becomes available. No other program in any other job can be started as long as there exists an eligible program waiting in a TOP priority job.

SUPER priority jobs are handled in the same manner, except that the scheduling function will interrupt as many programs as necessary to insure continuous processing of such a job. This scheduling scheme insures automatic and optimal system utilization while providing the operator with the possibility of instantaneous manual override.

hardware allocation

The scheduling function also involves hardware allocation for each program. As soon as a program becomes eligible, ADMIRAL automatically determines by inspecting the program the amount and nature of equipment that it requires to operate. These requirements (in terms of memory space, tape units, and other peripheral devices) are matched against the currently available system configuration to determine if the program can run now and, if so, to perform detailed allocation.

To fit a new program in memory, ADMIRAL attempts to find the smallest unused area capable of accommodating the program. This scheme avoids splitting memory into a number of small spaces. If, however, no single empty space large enough to hold the program exists and if, on the other hand, the total unused space exceeds the program's size, all other programs currently in memory are compressed toward the lower end of memory. This process creates a gap of available space at the upper end of memory into which the new program can be fitted.

In allocating tapes, ADMIRAL attempts to distribute the load among the available input-output channels. The tape allocation scheme is also aimed at minimizing the amount of tape handling by the operator between programs. When several programs of the same job manipulate the same logical file, inter-program setup can be reduced by insuring that the same physical unit is assigned to this file throughout the execution of the job. ADMIRAL interprets a statement on the job descriptor which specifies a logical equivalence between the symbolic addresses of such a file in the various programs of the job and assigns a single physical tape unit to this file.

To enhance the flexibility of the scheduling system and achieve greater equipment utilization, ADMIRAL also provides for dynamic hardware allocation throughout execution. Should the hardware requirements of a program vary during the course of execution, equipment allocation may be performed only when the equipment is actually needed, upon request of the program itself. Similarly, equipment may be released back to the system when no longer needed. For example, a program may initially require the use of a card reader to enter a few parameters, or it may use a large number of tapes during a sorting phase and only a few during a reporting phase. Such a program can request the necessary equipment when it is needed, use it only as long as necessary, and release it for use by other programs. This dynamic allocation feature is also particularly useful when dealing with programs whose requirements cannot be accurately determined in advance because they depend upon the particular data being processed.

checkout facilities

ADMIRAL is a checkout as well as a production operating system. It includes a complete set of debugging aids such as a program correction facility, test data distribution, pinpoint dynamic dumps with flexible editing format, and memory difference dumps (in which only those registers which changed since the previous dump are edited). Dumps can be requested when the program is submitted for execution or they can be obtained upon operator command at any time during the run.

ADMIRAL incorporates a number of automatic restart features and manual recovery procedures. Restart or anchor points are set upon program request when reaching convenient logical breakpoints. Intermediate results may thus be preserved as a guarantee against possible system malfunction. Restart points are also set automatically when a program must be interrupted. ADMIRAL dumps the contents of memory allotted to that program and records the current positions of the external files. Later resumption is achieved without loss of computer time by restoring the program to the status it was in at the time of setting the restart point.

To keep the human operator informed of the progress of the run, ADMIRAL logs the identification of each program at the start and at the end of the program. It instructs the operator of any decision it may take concerning job selection or hardware allocation and requests manual operations where necessary. The operator may override ADMIRAL's decisions through a set of commands issued from the console. Such commands may consist of changing job priority, adding or deleting jobs, interrupting, dumping, or restarting a program (or system of programs), modifying the system configuration, etc. ADMIRAL's response to these commands is immediate.

internal structure

Adaptability of ADMIRAL to a wide variety of hardware configurations and software systems is achieved by means of a completely modular structure. The system is placed under control of a master coordinating function, permanently retained in memory and capable of recognizing service requests issued by the operator or by any of the running programs.

Each call is analyzed and the proper function executed through a number of independent sub-functions which can be brought in and out of core memory as required. At the start of the run, the operator assigns to ADMIRAL a certain area of core storage. This area is automatically divided into three portions. One of them is allotted to the Coordinator; another is used to stack up the job descriptors. The rest constitute a pool-storage which is divided up among the various sub-function and associated communication arrays. New sub-functions are placed at the highest available locations within the storage-pool while arrays are assigned starting at the lowest locations. In this manner, the available space is concentrated in the middle of the storage-pool and can be used either to accommodate new sub-function or new arrays.

Control transfer between sub-functions is accomplished through the Coordinator, and can be one of three types—open, closed, or return. Open transfers merely pass control over from one sub-function to another. They are used to link the phases of a sequential function. Closed transfers define a new depth in nesting. They are used to call a sub-function in the manner of a closed subroutine. Conversely, Return transfers cause control to be returned to the next upper nesting level.

Data communication between sub-functions is accomplished through overlapping arrays associated with each sub-function. Storage allocation for these arrays is performed by the Coordinator each time a new sub-function is activated. In the case of an Open control transfer, the array of the calling sub-function is totally released and the corresponding area assigned to the array of the called sub-function. In the case of a Closed transfer, the array of the calling sub-function is not released, but may be partially overlapped by the array of the called sub-function. The calling sub-function can thus preserve intermediate results while allowing the called sub-function to access the necessary data. Finally, when a sub-function performs a Return transfer, its array is totally released.

This scheme eliminates the need for a fixed common array. It also provides for allocating arrays only when they are needed, for only the amount necessary and for only as long as they are actively used.

internal segmentation

Each sub-function consists of a string of coding which remains invariant throughout execution. Moreover, this coding only involves index or indirect addressing and thus is not sensitive to actual allocation. Sub-functions can, therefore, be retained in memory as long as sufficient space remains available and be re-activated at will. They can also be moved in memory without any need for code relocation. These properties provide for an extremely efficient, while simple, dynamic segmentation scheme.

When a new sub-function is called for and there is insufficient space available within the storage pool, one or several inactive sub-functions must be released. A simple overlay technique is used. Each sub-function is assigned a "weight" coefficient which denotes its relative frequency of use. This weight is automatically raised by the Coordinator each time a sub-function is activated. To make room for a new sub-function, the Coordinator merely scans the storage-pool and determines which sub-function is currently used the least frequently. This sub-function is released. The gap thus created is filled by "squishing" all adjoining sub-functions toward the end of the storage pool. The additional space obtained is thus automatically added to the center area of the storage-pool, and the regular space allocation procedure can be used for the new sub-function and associated array.

Because of the extreme simplicity of this segmentation scheme as well as efficiency of the storage allocation process, the cost of ADMIRAL in terms of memory space and computer time is kept to a minimum. Furthermore, this modular structure provides for accommodating various programming systems without imposing upon them any artificial restrictions. Local service functions, such as program loading and relocation, segmentation control and internal restart procedures which may vary from one programming system to another, are handled by different sets of sub-functions. These various sets of sub-functions, known as Local Monitors, are specifically tailored to meet the requirements of each programming system. Other more general functions, such as work load scheduling, are accomplished by a unique set of sub-functions called Master Monitor.

example of internal operation

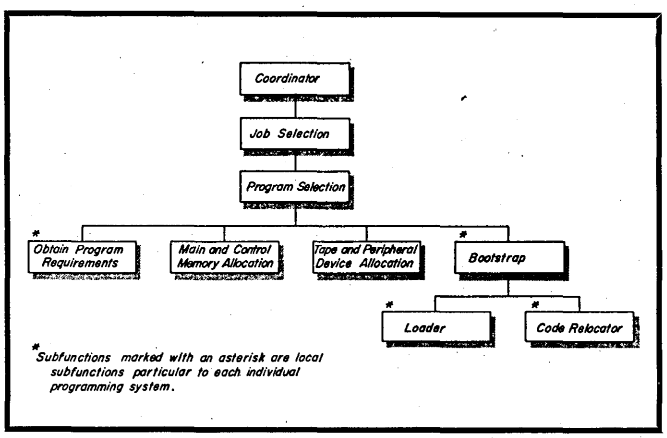
The interrelation between the Master Monitor and the local monitor sub-functions is illustrated in Fig. 1. This example involves the operation of the scheduling function, which is activated by the termination of a program.

When a program terminates, the Coordinator calls in the job selection sub-function to determine which job is to be examined first. Once a job has been chosen, the program selection sub-function checks to see if there exists any eligible program in that job. If so, a local monitor sub-function is called upon to examine the eligible program and determine its hardware requirements. Memory and peripheral allocation sub-functions are then used to decide whether sufficient equipment is currently available and to make detailed allocation. This information is then passed on to the bootstrap sub-function, which locates the first segment of the program, loads it, and relocates it by means of a local loader and relocator.

operating characteristics

It is worthwhile noting that ADMIRAL's services are only required intermittently, mostly as part of each program's setup. The job of time-sharing the central processor among the active programs is automatically handled by the standard multi-program control feature built in the hardware. This feature operates on an anticipatory basis, out-of-phase with the main memory, and therefore does not require additional memory cycles. The operation of ADMIRAL is thus quite similar to that of a real-time program; it normally remains dormant, becoming active only in response to a call issued by the operator or by one of the running programs. The cost of ADMIRAL in terms of computer time is further reduced by the fact that it itself operates in parallel with the rest of the work load. Only the calling program is interrupted when ADMIRAL becomes active; other programs proceed normally.

Fig. 1 The Admiral Scheduling Function



The experience has shown that, on the average, ADMIRAL requires less than five seconds of computer time for scheduled program.

The problem of queueing the calls to the Monitor is automatically resolved by the hardware with no time penalty. Whenever ADMIRAL is activated, an interlock is set which prevents further calls until completion of the current function. The method used is analogous to that employed by the machine to stack the demands on input-output channels. Calling programs are stalled as long as the Monitor remains in operation. Special provisions are made, however, to enable the operator to override this interlock when necessary.

Job descriptors are entered into the waiting list either via the console or by means of any conventional input device. Depending upon the situation, this input device may be assigned to ADMIRAL either permanently or only when the need arises to enter additional job descriptors. Full-time assignment of an input device is well suited to the case where the work load consists of a large number of short jobs; temporary assignment would probably be preferred in a production type of operation.

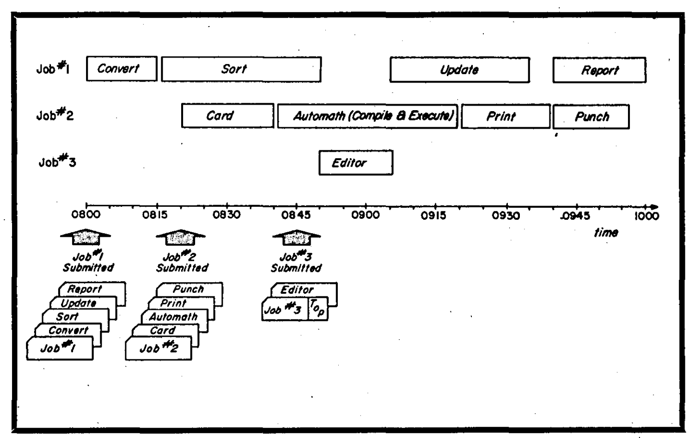
Job descriptors specify the identification and nature of the programs which compose the job. They may also include optional information such as absolute hardware assignments, tape consistency data, program corrections, dump requests, or test data. Test data is automatically written onto a tape assigned to the job.

The ADMIRAL system includes a library of utility programs which are available to the user for instantaneous call. These programs are scheduled and run like any other programs. They perform standard service operations such as tape handling, data editing, or peripheral conversion.

example of operation

The dynamic scheduling feature of ADMIRAL is illustrated in Fig. 2, using a simple example which involves only three jobs. The first is a master file updating job consisting

Fig. 2 Example of Dynamic Scheduling



of four distinct programs: CONVERT, a card-to-tape conversion program which writes a transaction file on tape; SORT, which arranges the transactions in master file order; (scientific) subprograms to be compiled and executed. This job includes the programs CARD, which converts the source decks and associated test data onto a common input tape; AUTOMATH, which compiles and executes the source programs, writing dumps and program results onto a common output tape; PRINT, which edits and UPDATE, which uses the sorted transactions to produce an updated master file; and PRINT, which creates a printed report. The second job consists of a batch of AUTOMATH prints the output tape on the printer; and PUNCH, which punches the compiled subprograms on binary cards. The third job consists of the single program EDITOR, which edits and prints a report from a magnetic tape file.

In order to illustrate the way in which ADMIRAL handles the scheduling of these three jobs, it is necessary to make certain assumptions about the times that the jobs are submitted and about the running times of the various included programs. In Fig. 2, it is assumed that the master file updating job is entered into the job list at 8:00 in the morning. Since all of the required equipment is available, the job is immediately scheduled and started. At 8:15, the card conversion is completed, and the sort is immediately selected by ADMIRAL and started.

At 8:20, the operator submits the scientific compilation job. Again, all of the required equipment is available, so the card-to-tape operation is started immediately, and at 8:40 the AUTOMATH compiler begins the series of compilations.

The operator next learns at 8:45 that a report to be produced by the program EDITOR is urgently needed. The third job is accordingly submitted with a TOP priority assignment. However, this job is temporarily delayed due to the unavailability of equipment. When the sort terminates at 8:50, the scheduling function, recognizing the existence of a TOP priority job in the waiting stack, interrupts job #1 and immediately schedules the report generator.

When, at 9:05, the TOP priority report is completed, job #1 is resumed with the program UPDATE. At 9:20, the series of AUTOMATH compilations and executions terminates, and the editing of the common output tape is started. Next, the UPDATE program is completed at 9:35; however, due to the unavailability of the printer, the program REPORT cannot be started until 9:40. At this time, the two programs REPORT and PUNCH are scheduled in parallel.

ADMIRAL may use a variable amount of equipment, depending upon the type of operation. It requires as a minimum one control group, the first 1,216 words of memory in any bank, and one magnetic tape unit. More equipment enables ADMIRAL to operate even more efficiently and to perform additional functions. For example, more memory can be used to hold more sub-functions in core storage and to reduce the need for access to tape. An additional tape can be used for dynamic dumping, logging, and storing restart information.

conclusion

In July 1963, a preliminary skeleton system was released to a selected sample of users for field test. The following December, a second package was delivered which included the bulk of the system. Additional releases incorporating the most advanced features are due by mid-1964.

It may be necessary to wait another few months in order to evaluate accurately the full impact of ADMIRAL over computer operations. Nonetheless, the experience to date has confirmed that the concept of a large-scale computer system equipped with multiprogram capabilities and placed under the automatic and continuous management of an operating system is feasible, practical and economical.

The case of Honeywell's Aeronautical Div. is typical of the kind of computer productivity which becomes possible under this new concept. Computer processing at Honeywell Aero reflects most accurately the pace and complexity of today's aerospace industry. A 32K, 15-tape H-1800 is used for producing both regularly scheduled and "special study" management information reports; the computer is also used as an integral tool in the research, design engineering and evaluation loop of product development. The work load may, therefore, be divided into four categories:

1) scheduled runs
2) planned "fill in"
3) requested "fill in"
4) "on demand"

Without parallel processing, a balance of central processor and magnetic tape speeds could be achieved only at the expense of artificial programming restrictions. Without dynamic scheduling, effective exploration of the parallel processing capabilities required rigid and sometimes impractical scheduling practices.

The combination of the parallel processing features of the H-1800 and ADMIRAL now allows the operating staff to use the scheduled runs and planned "fill in" as a base to which "requested fill" and "on demand" work can be dynamically added.

On the basis of the experience gathered to date, it is estimated that, thanks to ADMIRAL, it will be possible to cut central processor hours and increase the average number of tape units in simultaneous use, as shown in Fig. 3.

Fig. 3

	Without ADMIRAL	With ADMIRAL	Relative Change
Central Processor—Hours	100	60	—40%
% Central Processor Capacity Utilized	60%	97%	+37%
Tape Hours	750	900	+20%
% Total Tape Hours Utilized	50%	90%	+40%

The utilization rate of the central processor will increase from 60 to 97%, thus realizing a 40% cut in the required central processor time. Tape time increases by 17% and the tape utilization rate from 50 to approximately 90%.

To the ADMIRAL user, a substantial saving in central processor time may result in one or several consequences: a cut in overtime rental; an increase in capacity, and a decrease in "turn around" time which can be vital for scientific installations.

acknowledgement

The author wishes to extend credit to all of those at Honeywell who contributed to the development of ADMIRAL, and especially to T. F. Hatch Jr., who has been associated with the project since its conception. Many valuable suggestions were also received from Mr. B. L. Neff of the Metropolitan Life Insurance Co. and from Honeywell's Aeronautical Div. which had gathered considerable experience in the field of multiprogram operations. ■

OPERATING SYSTEM FOR THE 1410/7010

by NELSON L. BARNETT and A. KENNETH FITZGERALD

The problem of inefficient machine room operation—spending as much time mounting tapes as processing, and remounting a job after each phase—has been fully recognized during the past several years. Many of the monitor systems designed to solve this problem have actually resulted from some fairly astute retrofitting of monitor to previously written compilers and installation programs.

Designers of the new 1410/7010 Operating System, on the other hand, enjoyed the opportunity of planning and building an entirely new and integrated system. The design effort was initiated by laying down certain system concepts. It was agreed that the system would be:

- Developed and issued as relocatable modules with no absolute patches,
- Symbolic throughout, to eliminate absolute references and simplify the communication problem,
- Recursive so that the system itself can generate any modification of itself,
- Standardized at the interface of all subsystem elements with the monitor, to reduce monitor core requirement and make programs independent of one another during development and use,
- Open-ended for adding new hardware and functions, and
- Operated with one set of procedures for both IBM and user programs.

With these basic concepts in mind, several system objectives were designated as design goals.

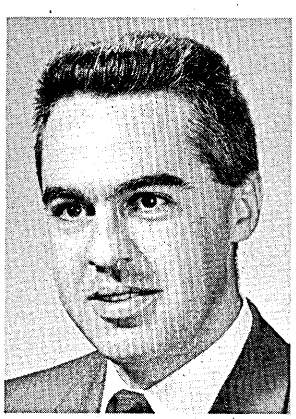
- Primary goal was to provide programming support for a wide range of machine configurations and processing applications, penalizing none with functions not desired, nor lacking those functions desired by a minority.
- The supervision and control provided would be as compatible with user-written programs as with the compilers and utility programs provided with the system.
- The monitor would enable users to batch process any mix of scientific, commercial, testing and production jobs.
- The operating system would be so constructed that the compilers would not have to compromise speed for supervision.
- Diagnostics would reduce the number of jobs rejected because of minor program errors.
- Compile-and-go facilities would be included for any number of subprograms constituting a given program, to simplify operations and reduce the number of machine passes required to debug a program.
- The loader would have the ability to gather subprograms from different language compilations and assemble them into a single program.
- Input-output assignments would be made on a symbolic basis so that tape drives, for instance, could be scheduled at execution time to suit the convenience of the machine room.
- All programs would permit a telecommunications interrupt at any time, without any special coding being written in the programs themselves.
- Programming standards, testing tools and operating procedures would be established to reduce elapsed time from problem definition to production status.

The operating system designed to meet these objectives is an integrated set of programs using a common operating procedure. It consists of the control programs, both resident and non-resident, and the compilers, production and utility programs which are dependent upon the control programs for an operating environment. By the nature of some of the objectives, such as continuous processing, some monitoring services must always occupy main storage. However, many other functions are transient in nature—thus the reference to resident (in main storage) and non-resident control programs. Basic and rigid standards were established early and maintained throughout system development for such functions as index register usage, calling sequences, subroutine definitions, and control card nomenclature. These standards helped assure the system's modular integrity.

system control

Three major elements make up the system control. These are the resident monitor, the non-resident transitional monitor, and the non-resident linkage loader.

The resident monitor contains routines for error checking and channel scheduling for I/O, a communication area,



Mr. Fitzgerald is a technical advisor to the 1410/7010 Programming Systems Group of IBM's Data Systems Div. in Poughkeepsie, N.Y. He helped coordinate development and improvments for SOS for the 709 and 90, and has been with the firm in the Scientific Computing Center, Service Bureau Corp., and with Applied Programming and Programming Systems. He holds a BS from Queens College, an MS from NYU, both in math.

Mr. Barnett is manager of the 1410/7010 Programming Systems Planning Group of IBM's Data Systems Div. in Poughkeepsie, N.Y. Previously, he contrbuted to the design of scientific and commercial languages and compilers and monitoring systems for the 705 and 709 at the Allison Div. of General Motors in Indianapolis, Ind. He has a BS in math from Franklin College in Franklin, Ind.

console inquiry routine, I/O tables, load routine, and end-of-program routines. These remain in core storage while the operating system is functioning. In a communications-equipped system, a Tele-processing* supervisor is part of the resident monitor as well.

The transitional monitor contains routines required to permit transition from one job or program to the next during batch processing. It includes I/O assignment routines, part of the console inquiry routine and a control card interpreter. It modifies and augments the resident monitor as required. Those duties which do not have to be performed while the user's program is operating are performed by the transitional monitor. This is carried to the extent that many routines in resident monitor are initialized by the transitional monitor instead of by the resident monitor. This results in a trade-off between flexibility at operating time vs core storage at operating time. The console inquiry is one such routine.

Though considered as a control function, the linkage loader is called into operation and functions as any normal batch program. What sets the linkage loader apart from the other elements of the system is its preparation of dependent programs—those that must operate under the monitor—for execution, and its unique relationship to the control programs during generation of a specific operating system. The linkage loader converts relocatable programs into absolute format for execution by resolving symbolic linkages and data references. Independently compiled relocatable programs are permitted to refer to locations in each other with direct linkages resolved by the linkage loader. The loader permits partial or complete overlays of program segments or phases.

The linkage loader assigns a subprogram to a particular area of storage in accordance with several factors: the size of the installation's resident monitor, the size and location of any other subprograms to be loaded at the same time, and control card information specifying unique storage assignments.

batch programs

Compilers, Sort Definition and utility programs written for this system come under the category of batch programs. Any dependent program is written with the knowledge that the system's IOCS and monitor do in fact exist and control the flow within the machine. Source programs written in Autocoder, FORTRAN, COBOL, and Sort Definition programs are translated into relocatable subprograms in a format acceptable to the linkage loader. The linkage loader converts these programs into an absolute program for subsequent execution.

Both COBOL and FORTRAN processors compile source language programs directly into machine language.** This process bypasses the intermediate stage of a lower-level symbolic language program common to many language compilers, and in part accounts for the significant compilation speeds achieved by the compilers. Since the compilers produce relocatable modules, subprograms from various language compilers can be combined into a single program, large programming projects can be segmented into smaller programming efforts, and subprograms can be shared among several projects.

sort definition program

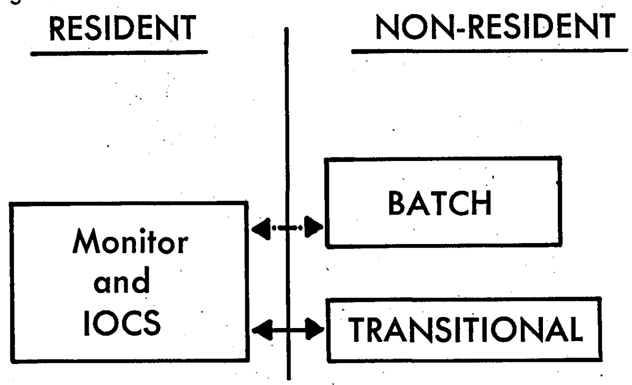
Sort functions are easily introduced into object programs through the sort definition concept employed by the system. A sort is defined by the programmer through the user of sort control cards. The Sort Definition program then tailors a specific sort to fit the functions desired, (e. g., fixed or variable length records; true sort or merge; two, three, four or five way merge). The sort control cards cause the Sort Definition program to select from 55 prewritten modules a combination of routines or modules that provide the most efficient sort procedure. The routines are properly combined into a working sort through the linkage loader's control cards.

system flow

The overall system flow is controlled by control cards placed at appropriate points within a collection of jobs to be processed. System flow can be altered by receiving an interrupt signal from a communication network.

A simplified flow of information within the operating system is shown in Fig. 1. The dotted line between the various elements listed in Fig. 1 represents how control is passed from element to element within the system. The solid line shown between the resident monitor and the transitional monitor depicts the inherent connection required between these elements. This provides the necessary resident monitor communication with little or no extra overhead required. Due to this direct and intimate connection between transitional and resident monitor, several functions otherwise destined to be in core at all times, have now been relegated to reside at least in part in the transitional monitor.

Fig. 1



Aside from the control programs, the compilers, Sort Definition and production programs are essentially free of dependence upon one another and as such, get identical treatment from the monitor. This is because control cards are efficient and because IBM and user-written programs do operate at the same level and meet the resident control at a common interface.

Development of compilers for this system, in fact, took advantage of this facet of system flow to speed the work. Autocoder was used to compile modules of the FORTRAN compiler; the linkage loader was then called into action and built a FORTRAN compiler; the newly created FORTRAN compiler was executed and the process was then repeated. Any complex of programs can be constructed to form an integrated job in order to reduce the critical turnaround time often experienced during development of a complex system.

A user would probably take advantage of this facility when using the Sort Definition program. A sort, quite often, can be considered a phase of a large program. On this basis, it is possible to: execute Autocoder and assemble process one; execute Sort Definition which tailors a specific sort; execute Autocoder and assemble process two; and then execute linkage loader, which in turn builds a production program of the form, "process, sort, process," with the sort being the middle phases of the program. This illustrates how a sort can be integrated into a program. It is equally feasible to build a sort as a stand-alone program. Furthermore, it shows how a program destined

*IBM trademark

**The new processor offers approximately four times compilation speed improvement over FORTRAN 40K, and 18 times over FORTRAN 20K. The times below are for a batch of 26 customer programs, comprising a total of 2,687 source cards. Speed improvements for the COBOL processor on the 1410 and 7010 (four programs, 5,600 source cards) range from 16-1 to 33-1.

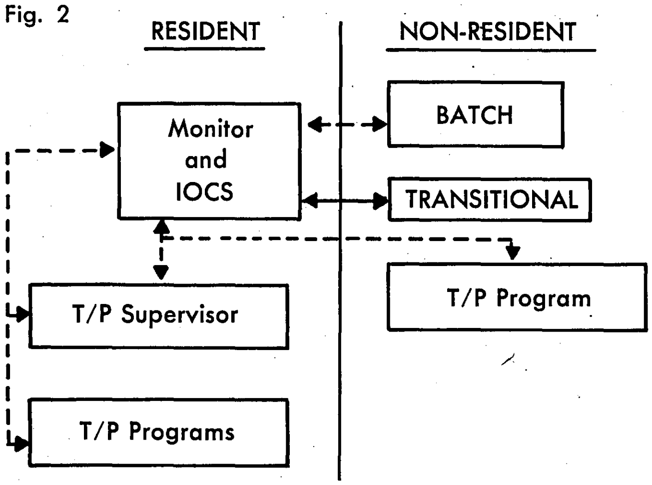
to be on an installation's system operating file can be tested before it is ready to be stored on that file.

telecommunications system

The Tele-processing Supervisor and Tele-processing programs are optional resident functions, as seen in Fig. 2. This relationship illustrates how the resident monitor can be augmented with user programs since Tele-processing programs are written by the user to meet his own needs.

The dotted lines between the resident monitor and Tele-processing Supervisor and between the Supervisor and Tele-processing programs illustrate how the overall system flow is changed upon receipt of an interrupt from a communication network. The system offers facilities for dumping the batch program and for calling in a Tele-

Fig. 2



processing program which then gets control. When the communication service is completed, control is restored to the batch program at the point of interruption.

Complete flexibility is given to the user by permitting him to retain some Tele-processing programs in core permanently while less frequently used programs are called as needed into the batch program area in core or into an area reserved for Tele-processing programs.

In a system with a communications network, detection of an interrupt signal while the resident monitor, transitional monitor, compiler, sort or any other program is in operation, results in temporary suspension of processing.

Registers and indicators being used by the batch program are saved in memory as a data block by a combination of IOCS and Tele-Processing Supervisor routines. No special coding is required in any user's batch program to satisfy communication demands.

When one of the Tele-processing programs is in operation and a second communications signal is received, the second interrupt will be remembered and properly serviced after the first is serviced. After the communications service is provided satisfactorily, which may require calling one or more Tele-processing programs into operation, control will be passed by the monitor back to the batch programs at the point where the interrupt occurred. Any pending batch I/O requests continue to be serviced during a Tele-processing program operation.

The I/O assignment scheme uses "indirect addressing" to achieve flexibility with a minimum resident core storage, thus avoiding dial spinning, tape dismounting or remounting, and unnecessary disc reformatting. Symbolic references are written in data files. Assignment cards inserted in the control decks equate symbolic names with physical units. When a program begins execution and requests the IOCS to "open" a data file, the IOCS, in cooperation with the monitor, links the appropriate data file in the user program with the physical unit to which the symbolic name has been equated. I/O assignment is accomplished by two tables in resident monitor and two corresponding tables in transitional monitor. The construction of I/O tables is such that no wasted storage is expended in resident monitor. Direct linkage to one entry in one of the resident monitor tables from each data file within a program as well as the more intricate linkage from transitional monitor to these tables are resolved by the linkage loader.

The IOCS uses list processing procedures for queueing I/O requests and service activities. Complete provision is given to a user for intercepting the IOCS at various points in its internal operating cycle. Exits such as I/O-operation-being-completed, errors-uncovered, and standard end-of-file and end-of-reel exits are among those provided.

List processing works on a priority of "first in, first out" for batch requests. This is altered when a communication interrupt occurs, at which time priority is given to the communications equipment. Any batch request queued at the time of interrupt continues to be serviced on a first in, first out priority during the processing of the communication request.

system generation

The system generation is designed to facilitate the creation of a unique operating system that provides an efficient vehicle for processing the work load within an installation. This is realized by providing an integrated package containing the necessary control programs and dependent programs in an integrated fashion to minimize cost in time and equipment. In general, for each installation there exists some optimum combination of system elements which, when integrated with user programs, forms the best possible operating system for that installation. It is to this end that the system generation process was directed. In the past, the concept of system generation has been employed by so-called editing programs which permit the removal of system elements, as well as the addition of user written programs.

Here, system generation creates a set of direct linkage points for the resident monitor and their corresponding symbolic equivalents. These are stored in tabular form and made available to the linkage loader for subsequent object program tie-ins. This form of communication, rather than the vector list scheme, was used because of its space and time-saving characteristics. It is only necessary to regenerate the resident monitor when new functions or additional equipment are added to the system. Object program changes, such as in the compilers or in any user-written program, do not require regeneration of the total system. Through the use of the linkage loader, this process provides a completely symbolic system for the user with no dependence on absolute references between functional modules, and no critical points anywhere in memory even for the resident monitor.

System elements (approximately 300 modules) are contained within a relocatable library and can be contained, in whole or part, in the system operating file or on a separate tape or disc file. The libraries are referred to by the linkage loader in building programs during the generation process and building absolute programs for immediate execution.

In the system generation process, the user is permitted to select, add, delete, or modify any modular element in order to satisfy his equipment and functional requirements. To allow this, the system was designed with a minimum of required control functions and with interfaces that permit adding additional control modules and dependent program modules. There is no penalty in performance or core requirements because of functions not selected. A mandatory set must be generated, but this can be replaced with user-written functions having the same external interface, though different internal logic.

Those modules of the monitor and IOCS not needed at a particular installation, such as modules required to provide system residence on disc, can be discarded. Those portions of the resident monitor, including IOCS, which are extremely variable are generated from macros, so that little or no excessive characters are generated for a given configuration or function. Resident monitors can range in size from a minimum 8.7K characters of memory to as much as 35K in an extreme case, meeting the needs of 40K to 100K users. The installation without communications, but with compile-and-go capabilities in a 40K tape environment, requires approximately 11.5K for the monitor.

System generation is a process which utilizes the Autocoder compiler, the linkage loader, the Sort Definition, and the inherent system flow capabilities in tailoring a unique system for a specific installation.

One can only expect that future operating systems will be even more effective, more powerful supervisors of data processing systems. Certainly with the significant trend toward real-time data handling and processing, operating systems must be capable of responding to increasingly complex equipment configurations and functional requirements.

Although this article is directed specifically at the 1410/7010 Operating System, the recently announced System 360 is conceptually consistent and in most cases has taken the next logical step in providing a truly integrated and flexible set of programming packages. ■

master control program

OPERATING SYSTEM FOR THE B 5000

by CLARK OLIPHINT

Two of the major B 5000 design objectives were (1) that all programming was to be done in ALGOL and COBOL, and (2) that the operation of the system was to be directed by a Master Control Program (MCP) which would relieve the operator and—especially—the programmer of virtually all the inefficient and error-causing details of peripheral unit designation, memory area assignment, and so on. The simultaneous and coordinated design of the computer and the programming system has produced a hardware-software system so well integrated that all B 5000 users employ the standard programming system (with minor modifications for special applications in a few cases). It has been the experience of B 5000 users that the exclusive use of compiler languages in programming gives advantages in documentation, program preparation, and debugging which cannot be over emphasized.

Another major design objective was that processor time unused by one program during input-output operations was to be used to process other programs, and that the availability of this feature was to have no effect on program preparation. In other words, the system was to operate in such a way that the programmer need have no concern about relative processor-I/0 times. This objective has far-reaching implications. For example, if one program is to be run while another is waiting on input or output, then both must be in memory at the same time. Since the system was designed to free the programmer from houskeeping chores, and to allow any simultaneous mix of ALGOL and COBOL compilation and execution, *all* allocation of memory and peripheral units must be done *at the time the programs are executed.*

operation of the hardware

One would expect that this approach to computer design would result in a different breed of computer, and this has indeed been the case. For example, one of the unusual

For the past 2½ years, Mr. Oliphint has been manager of the B 5000 Master Control Program project with Burroughs Corp. in Pasadena, Calif. He has had various assignments in the systems programming and product planning areas. He holds a BA and MA in physics from Rice University.

features of the B 5000 is the "stack," which is used to store operands, subroutine return information, subroutine parameters, and temporary values used by subroutines. Each program has its own stack in memory. As operands are brought from memory they are placed at the top of the stack, pushing down all other items in the stack. Arithmetic operators use the top two values in the stack as operands, remove them from the stack, and leave the result in the top of the stack.

The stack actually consists of two arithmetic registers, A and B, in the processor, and consecutive cells in core storage. The S register in the processor is initially set to the first address of a group of words to be used for the stack. If A and B are both full and an operator is executed which causes an operand to be added to the stack, then (as part of the execution of the syllable) S is increased by one, the contents of B are stored in the word now addressed by S, and the contents of A are transferred to B. The A register is then free to receive the operand being added to the stack. If B is empty and an operator is executed which requires two operands, then the reverse action occurs: the contents of the word addressed by S are brought to the empty register, and S is decreased by one. Analogous operations take place if A or both A and B are empty. This technique achieves the effect of a push-down stack, and makes it unnecessary to move any word in the stack from one memory cell to another. Except for the arithmetic registers A and B, which are shared by all programs, each program has its own stack. Thus, in generating programs to do series of arithmetic operations, the compiler does not have to make provision for storing temporary values; they simply remain in the appropriate stack until they are needed. Since all parameters for subroutines, *including the return address,* are in the stack rather than in the body of the subroutine, recursive subroutines are executed as easily as non-recursive subroutines. It has proved to be truly gratifying, in our experience, how this implementation of the stack concept has made recursion almost trivially simple.

Since a program and its data may be anywhere in memory when the program is executed, it is useful to have an easily accessible table into which the MCP may place absolute addresses for those segments of the program or its data which are in core storage. Such a table, called the Program Reference Table (PRT), is created for each program. The R register in the processor unit is set to the base of the PRT, and all memory references in the program are relative to the R register. Operands, descriptors of data segments, and descriptors of program segments are in the PRT.

If a segment is in core storage, its descriptor contains the base address of the segment. The program may read or store any element of a data segment by using the data descriptor and an index of the element. Branches from one program segment to another, or subroutine entries, are performed by using a program descriptor. If a segment is not in core storage, a bit in its descriptor is set to zero; any reference to that segment then causes the program to be interrupted so that the MCP can bring the segment from a drum* into core storage.

When an input or output operation is executed by the B 5000, an input-output descriptor containing the peripheral unit number, the type of operation to be performed, and a base address in core storage, are sent to an I/O channel. The I/O channel then executes the operation independently recording errors if they occur. When the operation is completed, the I/O channel sets a bit to interrupt the processor and stores a result descriptor in memory.

There may be as many as four I/O channels in a B 5000 system. No extra complication arises, however, from the presence of multiple channels; neither memory modules nor I/O units are permanently connected to any specific channel, nor is the programmer concerned in any way. The connection is made when an I/O descriptor is sent to an I/O channel, the first channel available receives the descriptor and executes it, regardless of which memory module or peripheral unit is involved.

Two important MCP functions are made possible by the timer, which is an integral part of the B 5000: to enable log-keeping, and to prevent a program from getting caught in a loop and staying there until the operator recognizes the fact and stops the program. The timer is a six-bit counter which is incremented by 1 every 60th of a second; after 64 counts, overflow occurs and causes an interrupt. After four such interrupts, the MCP compares the elapsed processor and I/O time with the (optional) estimate supplied by the programmer. If the running time exceeds the estimate, the program is removed from memory and the operator is notified. As implied above by the word "optional" in parentheses, the programmer may omit the estimated processor and/or I/O times; in this event, the MCP inserts a time "estimate." As of this writing, the MCP word reserved for this estimate is then arbitrarily filled with 1's, yielding an "estimate" of approximately 9.7 hours. Naturally, if one wishes to employ the time-keeping abilities of the MCP, he begins with a pessimistic estimate and refines it as he gains experience.

A program being executed will be interrupted when any one of 40 conditions occurs; some of these have been described above. When a program is interrupted, the contents of all the pertinent registers are stored in the upper part of that program's stack, and control is transferred to a location associated with the condition causing the interrupt. At that location is the beginning of an MCP routine which will take the proper action.

operation of the standard programming system

The standard programming system, including the MCP and the ALGOL and COBOL compilers, is sent to users on magnetic tape. A small card deck is used to load the MCP on a drum. Either compiler, or both, may also be loaded on a drum. If a compiler is not in drum storage, the MCP will find it on the tape and read it in when it is required. The MCP and ALGOL compiler together require about half of one drum; if the COBOL compiler is included, about one-and-a-half drums are needed. Each drum contains 32,768 words.

The MCP uses one magnetic tape, called the Program Collection Tape (PCT), to store programs scheduled to run but not yet brought into memory. All other tapes are available for programmer use, but are individually recognized (if labeled) and selected by the MCP as required by various programs. The programmer need not (and indeed cannot) designate *specific* magnetic tape units, although the operator can. The operator is notified when a reel is to be removed or put on a given unit; the message also gives the name of the file and of the program.

The MCP allocates core storage for a program as the program requires it. If the storage requirements of the program exceed the available amount of core storage, the MCP automatically overlays segments of the program or its data to make room for other segments. Because of this feature, programs can be executed on the B 5000 even though they apparently require more core storage than is available.

Obviously, any program—and a compiler is no exception—requires some minimal amount of core storage for its execution. In order to execute the MCP, the compilers, and most object programs, three out of a possible eight memory

*One drum is required in a minimal B 5000; a second is optional, even in a two-processor system.

modules are required on a B 5000. However, overlaying program segments and retrieving them later will require processor and I/O time, thus adding to the time required to execute a program. Considerable increases in program speeds can be achieved by simply adding one memory module. No program changes are required for adding or deleting memory modules or peripheral units, unless a drastic deletion results in an unworkable situation (e.g., attempting a three-tape sort with but two magnetic tape units on line).

mcp functions

The major functions of the MCP are listed below in the order in which they are usually invoked.

1. Scheduling and loading programs
2. Reading program and data segments from a drum when required by a program
3. Allocating and overlaying core storage
4. Assigning peripheral units and input or output buffer areas for each program
5. Initiating input and output operations, and recognizing the completion of these operations
6. Removing a program from memory when it is finished or when certain error conditions occur
7. Adding programs to or deleting programs from a library
8. Maintaining a log of system operation, which is written on any unit when requested.

The MCP is composed of 31 segments, some of which are discussed below. Depression of the LOAD button on the B 5000 console reads 512 words of the MCP from a drum into core and transfers control to the initial portion of the MCP. The Initialization Routine in the MCP then reads into core those MCP segments which are to remain permanently in core, and initializes all MCP tables.

Four segments of the MCP remain permanently situated in core: the Program Control Routine, Input-Output Initiate Routine, Input-Output Complete Routine and Storage Allocation and Overlay Routine. All other segments are read from a drum as they are required, and remain in core until the space is needed for some other purpose.

The Program Control Routine is used to transfer control to any segment which may not be in core. If the required segment is not present, the Program Control Routine will obtain space for it and read it from the drum. The Program Control Routine also has the function of deciding which program is to be executed if more than one is in memory ready to run. After an interrupt has been processed by an MCP routine, control is returned to the Program Control Routine, which in turn resumes (or initiates) processing of one of the programs in memory. The Program Control Routine chooses the highest priority program that is ready to run. (A program is not "ready to run" if it is waiting on an I/O operation).

program scheduling and loading

Programs are scheduled and loaded by two segments of the MCP called Collection and Selection. Collection adds programs to the schedule as they are specified by program header cards. A header card may call for a program to be compiled and executed, in which case Collection will add the compiler to the schedule, and, when compilation is completed, will add the compiled program to the schedule. If a header card calls a compiled program from a library, Collection will add that program to the schedule. In every case, a program is merged into the schedule in priority order behind all programs of equal priority already in the schedule. When processor time is available, Selection is notified by the Program Control Routine, and will attempt to find a program in the schedule which can be loaded into memory and executed. It will check information collected by the compiler as to memory and peripheral unit requirements, and will choose the first program in the schedule which will run in the amount of system now available. It then loads the selected program into memory and leaves it ready to begin operation when chosen by the Program Control Routine. The entire program is written on the drum in this process, and the beginning segment is also read into core.

When space is required in core storage for any purpose, the Storage Allocation Routine is used to obtain it. All available core storage is in a linked list, with the smallest space first on the list and the largest last. When core storage is requested, the first available section of core large enough to satisfy the request is removed from the list, the required amount of space is assigned in that section of core, and the remaining amount is linked into the available storage list. If there is not enough core storage available, the Overlay Routine is called to make more storage available. Since there is a copy of all MCP segments and program segments on a drum, it is not necessary to write these segments on the drum when they are removed from core. Data segments, however, are written on the drum if they are overlaid.

The order in which assigned memory is overlaid is: first, "overlayable" MCP segments (all segments except the four permanently in core); second, program segments of the lowest-priority program; third, data segments of the lowest-priority program. The second and third steps are then repeated for all remaining programs, proceeding from lowest to highest priority. The overlay operation continues until either the required space has been made available or there is nothing more which can be overlaid. In the latter case, one of the programs in memory actually requires more core storage than the compiler has estimated. It is then too late to take corrective action; an error message is typed out, and all programs in process are terminated.

When either a program segment or a data segment is overlaid, the space is linked into the available storage list, and the descriptors associated with the segment are marked absent so that any future reference to the segment will cause a "presence bit" interrupt.

As a program is executed, it may, from time to time, branch to program segments not in core, or try to refer to data not in core. In either case a presence bit interrupt occurs, and the Presence Bit Routine in the MCP must find the required segment on the drum, obtain core space for it, and read it into core. When this is completed, the segment is marked present and the program is left ready to run as soon as it is chosen by the Program Control Routine.

input-output

The assignment of peripheral unit numbers and buffer areas is made at the time a program first reads or writes a file. The MCP maintains a record of the file names of all input files and the units on which they are mounted, and the units on which output files are mounted. The Four-Second Routine in the MCP is responsible for maintaining this table. This routine is so called because it is normally executed on every fourth timer interrupt (approximately every four seconds). In the course of its operation, it also determines the status of all peripheral units. If the unit is a magnetic tape unit, the Four-Second Routine reads the first record on the tape and stores the name of the file mounted on the unit. Once the file name has been recorded, the tape will not be read again unless the Four-Second Routine detects a change in the status of the unit. After such a change (e.g., from ready to not ready), it is necessary to read the tape label again when the unit is returned to a ready status. If a tape is mounted with a write ring, the Four-Second Routine also checks a purge date in the label to ascertain whether the information on the tape can be destroyed. If the purge date has not been passed, a message to that effect is typed out, and the operator has a chance to correct a possible error.

When a file is opened, an input-output descriptor is created which the program then uses in the execution of any operation with that file. To perform a read or write operation, the program executes a Program Release operator, causing an interrupt which requests the MCP to perform the specified operation. After checking the availability of the requested unit and any one of up to four I/O channels, the Input-Output Initiate Routine in the MCP will either initiate the operation or put it in a waiting list of I/O operations, and then call the Program Control Routine to choose a program to run. The program requesting an I/O operation is eligible to run, since it may not need the information just requested for some time. If it tries to use the information before the operation is complete, it will be interrupted and set "not ready" until the operation is complete.

When an I/O operation is completed, it causes an interrupt. The I/O Complete Routine in the MCP then initiates a waiting operation if possible, sets the program waiting on the operation just completed "ready," and calls the Program Control Routine to choose a program to run.

When a program is finished, it executes a communicate operator, which causes an interrupt. The Communicate Routine in the MCP closes any files not closed in the program; returns all core and drum storage assigned to the program; calls the Log Routine to store information about the program; and, if the program finishing is a compiler, calls Collection to add the compiled program to the schedule. It then calls the Program Control Routine to choose another program to run.

library tapes

The library tape maintenance system consists of a group of routines to add or change programs on a library tape. The header card calling for compilation may specify that the program is to be added to a library or is to replace a program of the same name already on a library tape. Another control card may be used to call a program from a library, either for addition to another library or as a replacement for a program with the same name in a library. All programs to be placed in a library are written on the Program Collection Tape (PCT) when the above cards are encountered.

A library tape is created on recognition of a library tape maintenance control card. This card may either cause a specific library tape to be copied with additions and replacements as noted above, or it may cause a new library tape to be created containing only those programs previously specified to be added to a library. After the tape is created, a message on the Message Printer gives the operator the name of the library and the unit on which it was created.

log

At any time, the operator may request that the log information be written on the peripheral unit of his choice. The first line of the log for each program shows, from left to right, the name of the program, whether the program was compiled (COM) or executed (RUN), remarks from the header card, the date, start time, finish time, number of seconds the processor was used for this program, reason for stopping the program (EOJ= normal program finish; ERR = error finish), and the number of seconds the peripheral equipment was used for this program. Each succeeding line gives information about a file declared in the program. This information, from left to right, is multi-file identification, file identification, preparation date, cycle number, number of seconds this file was used by the program, number of seconds this file was assigned to the program, number of errors encountered in reading or writing this file, and, for those files actually opened, the type of peripheral unit on which the file was located. To conserve table space, peripheral unit names are abbreviated: CR means card reader, LP signifies line printer, and MT indicates magnetic tape; additions now being incorporated will include PR for paper tape reader and

Fig. 1. Typical B 5000 Log Output

```
 TERM14  COM COMPILATION CARD                    64049 0016 0016 00023 EOJ 00024
0000000 DATA    00000 01 00004 00019 000 CR
0000000 OUTPT   00000 01 00010 00016 000 LP
MCPPCT  CODEFIL 00000 01 00001 00012 000 MT
0000000 SOLT    00000 01 00000 00000 000
0000000 SOLT    00000 01 00000 00000 000
0000000 LIBRAR  00000 01 00000 00000 000
 TERM14  RUN                                     64049 0016 0017 00017 EOJ 00006
0000000 TAPEREC 00000 01 00002 00016 000 MT
0000000 CARDREC 00000 01 00000 00016 000 CR
```

PP for paper tape punch. The last column is blank if the file was declared but never actually used in the program. A typical log is shown in Fig. 1.

results

As was originally planned, all programming for the B 5000 is being done in ALGOL or COBOL. Both compilers contain features which have proved to be highly satisfactory in checking out programs. Estimates of increased programmer productivity resulting from the use of compiler languages indicate a manpower saving of about a factor of 10 in programming, checking out, and documenting typical programs.

In our own experience, both compilers were written in ALGOL, and we realized an increase in programmer productivity of four or five to one; i.e., if the compilers had been written in assembly language, it would have taken four or five times as many man-hours as were actually required to complete the task.

The increased productivity of the B 5000 resulting from the use of normally idle processor time is highly dependent on the type of programs being executed. If a group of programs are all processor-bound, no time can be saved by running the programs together; it is not unusual, however, to find two or three programs which run together in very little more time than that required to execute the longest program of the group.

Most cases fall between these two extremes. In the usual case, a group of programs can be executed in less time than is required for the same group if each program is run alone, but will require more time than that required for the longest program. (This should not be taken to imply that the shorter programs will not be finished before the longest one). The actual time saved depends on the amount of idle processor time which would be available if the programs were executed serially. ■

ELECTRONIC SYSTEM DIVISION

edp system selection

by ROBERT B. FOREST, Editor

Discharged as an Infantry major after WWII, Col. McCloy joined the Air Force in '47 and has since held comptroller and data systems management positions. For 3½ years, prior to his present assignment, he was responsible for dp equipment selection and software development for the AF Logistics Command, and has been with ESQ since its inception. He holds a BS in Commerce from Iowa State and an MS in Business Administration from Ohio State.

DATAMATION: Could we begin with your name and title?

MC CLOY: Colonel Edward McCloy, chief, EDP Equipment Office, Hanscom Field, Bedford, Mass. The Office is part of the Electronic Systems Division of the Air Force Systems Command, which has headquarters at Andrews AFB near Washington, D.C. Our code symbol is ESQ.

DATAMATION: I wonder if you could describe briefly the goals of the EDP Equipment Office—which we can refer to as ESQ—and perhaps the scope of this activity?

MC CLOY: Let me take them in reverse order. The EDP Equipment Office, ESQ, was established in January 1963 to be the single EDPE selection agency in the Air Force for business and scientific systems and for some selected additional systems. We are involved only with those which are to be competitively procured. If sole source selection is approved by Hq USAF, which sometimes does occur, where a sole source situation exists, we don't get in the act. Our primary goals are to insure that we get equipment that can do the job effectively and economically for the Air Force, and that we have full competition and objective selection to meet the Air Force data systems requirements.

DATAMATION: Is this scope or are these goals the same as they were when ESQ was established?

MC CLOY: Yes, except that we have an added function of evaluating all unsolicited EDPE proposals.

DATAMATION: And you don't anticipate any other change in the goals or the scope?

MC CLOY: No.

DATAMATION: How are you organized to achieve this job?

MC CLOY: We have three divisions. One, quite small, is a Plans Division, which establishes basic procedures and handles management controls. We have two additional divisions, one of which is systems oriented (systems now meaning data rather than equipment system orientation). Its role is to work with the user to insure that we understand the system requirement, and that this requirement is properly specified to, and satisfied by, the manufacturer. The other division is hardware and software oriented, and is made up of technicians with a programming background. The two divisions jointly go through the evaluation. The work is handled on a project basis, taking those skills and people necessary to handle the particular selection. As such, a project will have people from both divisions.

DATAMATION: How many people do you have in ESQ?

MC CLOY: The authorization is 45. As of this morning we have 43. Of those, six are secretarial, the remainder are either supervisory or technical. And the makeup of the technical staff is about 50-50, military and civilian. You might be interested that, for the roughly 20 technical civilian positions we have, we had over 150 applicants; I'm very pleased with all of the people here.

DATAMATION: I was going to ask you about the quality of your people and how you hold them. I presume that you can't always compete with private industry's salary level.

MC CLOY: I've been pretty fortunate; we have had recognition of the skill requirement here, and have been able to compete fairly well. We went on a rather agressive recruiting campaign, and had applications from all over the United States, as well as the U.S. Army in Europe. I think the people here for the most part are glad that they came. In some ways, I can't think of a better opportunity for an individual who has gone a fair ways in the data systems business and wants to broaden himself. He will be exposed here to a wide range of applications, particularly of management-type or business systems. The Air Force, as you probably know, is the world's largest user of these equipments, and they do have a great variety of applications. Our analyst also is exposed to a wide range of equipments. We take the proposals and break them down in great detail, analyze them very closely, so he gets to become aware of the capabilities of various manufacturers' equipment and various categories of systems.

DATAMATION: What kind of background is required for the work?

MC CLOY: We structured ourselves to civil service classifications; one of these is the digital computer system analyst, and those in the systems division are then defined as system analysts. Many of the analysts, of course, began as programmers. In the technical division, we have designated them as programmers. An independent survey was made of our group recently, just to see how we were progressing, and the survey team was quite surprised at the range and depth of experience of the civilian employees here. The average experience is about eight years.

DATAMATION: Is the system analyst in a higher position than the programmer?

MC CLOY: No, these work in parallel. We were able to get recognition here of the skilled programmer. There has been a tendency sometimes, in some organizations, to "graduate" from programming to analysis, but we don't look at it this way. These are two skills that are parallel and complement each other.

DATAMATION: How do you handle what might be a problem with your military personnel? Are they always able to achieve a rank that allows them to earn an amount equal, or similar, to their civilian counterpart?

MC CLOY: This is a dilemma. It depends on the individual and the rank. The recent pay raise helped in the lower grades. At the captain level, for example, each competes fairly effectively for his age. One of the programs the Air Force has today is to encourage service careers for second lieutenants. They have a mandatory tour of four years, and after that they do a lot of soul-searching whether they are going to stay in service or get out. But if we can talk them over that hump (and by that time they are usually first lieutenants), we're in. They can see some rapid promotions on the civilian side sometimes, so it's a perpetual problem—no different, I guess, in my function than anyone else's.

DATAMATION: Now, could you describe your function here, and the process by which a bid is evaluated?

MC CLOY: Well, about two years ago Hq USAF found that it had three major problems in the data systems area. The first was a lack of effective systems control. A user could solicit hardware proposals from manufacturers and make a selection before Hq USAF really knew what he was up to. He would, however, have to get Hq USAF final approval. Second, there was a lack of consistency and objectivity in applying the criteria by which equipment was selected. The third was the need to standardize systems among commands. It became apparent that it was highly desirable to have one skilled organization do the equipment selection in relation to standard systems approved by Hq USAF. So they really did the two things simultaneously: put systems under control and centralize equipment selection for approved systems.

If a user has a bright idea of something that he wants to put on a computer, he starts with a Data Automation Proposal, or DAP. This goes up to Hq USAF. If this is blessed and if there are implications of new equipments, then we get a copy of the DAP so that we can begin to become familiar with the problem. If new equipment is not required and the DAP is approved, the user proceeds to develop the system on in-house equipment. As I said, where there's an implication of new equipment then we become familiar with the project. Now we don't work with it very closely, but only intermittently. We don't help the user design the system; that's his task, as we don't have this many people. As we gain experience, we can give some guidance to the scope of the project, the possibilities of the kind of capabilities which are coming down the pike, etc. For the most part today, we have insured that the user has documented his system in such a way that after it is approved by Hq USAF, we in turn can put this out into a Request for Proposal to manufacturers. The user's final product is called his data systems specification, which is the documentation of his requirement in more detail than his original proposal. The DAP is a little less than the classical feasibility study; the Data System Specification is a little more. The specification again goes up to Hq USAF; if it is approved the systems specification is sent to us. We finalize it from an equipment selection standpoint, and put in other data which is of the evaluation type —questionnaires for the vendor, how he is to respond, what formats he should use, workload data so he can time the system out, etc. We then finalize the Request for Proposal (RFP).

Prior to issuance of this RFP, there are certain things we have to go through. Hq USAF has directed that all selections be done by what are called System Source Selection Board techniques. This means that there is a final committee, of which I am usually the chairman, which makes the selection. It also says that the criteria for these selections will be established prior to the issuance of the RFP. In other words, by the knowledge that our people have gained about the specification by working with the user, we establish the things which will be evaluated and how they will be weighted in relation to the total. The RFPs are then sent out to the manufacturers. We're open for questions during a period of time. If one manufacturer asks a question, we answer to all manufacturers (unless he asked about a proprietary item) so they have the benefit of the same information.

They then come in with their proposal. In some cases they must then actually run a problem, which we call a benchmark problem, on their own equipment. The proposal and the benchmark data are then evaluated by what we call an Evaluation Group, which is an element under the Systems Source Selection Board. This evaluation group in turn has sub-groups, each of which analyzes an area. For example, we could have a cost sub-group, a systems sub-group, a software sub-group, etc. These each work toward the pre-established criteria, but without access to the results of the other groups; this assures objectivity. In other words, no one on this Evaluation Group can put together the results and say, "I know who won this one." For example, the systems sub-group does not know what the impact of system timing might be on costs because they don't know the costs. So this assures objectivity by keeping the evaluation sliced up. Each of the evaluation sub-groups then comes up with its validation and evaluation against these pre-established criteria, without weights. These are given to the Board; the Board applies the pre-established final weights, and the final values then become the basis for the Board decision.

I might make one other comment here, in light of the original question, that has to do with validation and analysis. I think the vendors are finding out that we are going into the ability of their equipment to do the job in much more detail than any customer, at least Air Force user, has ever done before. We're getting down to the rock bottom of equipment capabilities and their response to the proposal. We're not buying brochure treatments. My impression is that the vendors are impressed with the degree to which we are searching through the proposals to insure the selected equipment can do the job.

We can't afford to have a near miss. The manufacturer cannot afford to have a near miss. It has been taking us roughly six to eight weeks in our analysis before we come up with a winner.

DATAMATION: Would it be possible for a particular manufacturer to help a user develop a DAP in such a way that it almost pointed to his equipment?

MC CLOY: Among my responsibilities is the evaluation of unsolicited proposals. This goes along the same route of system approval. We have asked the manufacturers—not required them, but asked them—not to participate with our users, and Hq USAF has directed our users not to use vendor assistance in any way prior to equipment selection. We have stressed that what normally happens, if the manufacturer makes such a study, is that by the time the DAP has gone through the selection or development cycle, it turns out his equipment isn't the right one to do the job. Additionally, one of my basic responsibilities is to insure that a specification is not "manufacturer-oriented." If a user has inadvertently oriented the specification to given equipment, we will modify the specification to eliminate those elements. We insure that the RFP is competitive.

DATAMATION: What procedure is there to help the user make sure that he has evolved a sensible system under the DAP?

MC CLOY: Well, I think it occurs in two ways. First, we have a rather deep reservoir of experienced EDP people in the Air Force. The first computer installed by the Air Force, I think, was in 1952, and they were rather widespread by 1958. Consequently, the initial analysis is usually done by competent people. Second, there is a series of review channels; the final one for the DAP and the specifications is up to Hq USAF, where they have a comprehensive review process, with a final buy-off by what is called the Data Automation Panel. This is a part of the system review that I talked about, and it's quite comprehensive.

DATAMATION: Now, you mentioned a board which sits on the results of the various evaluations sub-groups. What is its composition?

MC CLOY: The System Source Selection Board usually consists of five to seven senior personnel from my office, primarily my division chiefs and myself. After going through this two or three times we found it was a good idea to have a consistent board, getting experience, whereas in many AF boards a man may serve once in his lifetime. Then we also go elsewhere in ESD to bring new blood and new ideas to the Board. And in turn we hope to give some experience to the rest of the ESD in some of the problems in equipment selection. Now, in those cases where we are getting multiple equipment, such as equipment for 152 installations, we may establish a formal SSSB. This is a Hq USAF Board, chaired by a general officer, and with representation of the various major air commands affected. In that case, ESQ serves as Evaluation Group, in support. This probably will occur about once a year. So far we've had one case of this sort.

DATAMATION: What steps do you take to make sure that a bid is open to as many qualified people as possible?

MC CLOY: We have a list of all known manufacturers of EDP Equipment which we refer to for each requirement. In addition, we give it widespread publicity by synopsizing in the Department of Commerce Bulletin. Also, we put out normal press releases—every once in awhile you pick it up in *Datamation!*

The Air Force has a policy that any vendor responding must be able to respond to the total system requirement. That is, a vendor cannot say, "Let me furnish a printer for the system." He must respond to the total system requirement. Of this list of 34 manufacturers, some really cannot react, so they may try to sub-contract to a major manufacturer. If they make, say, a CRT display system and we have a CRT requirement, they might come in with a firm that lacks display capability. So we get proposals from a considerably lesser number than the total list.

DATAMATION: Might any of these be considered a systems management or consulting firm?

MC CLOY: No. We have had some consultant firms ask for our RFP. We have always responded stating the systems requirement, but, today, we have not had a second letter.

DATAMATION: Are any steps taken to ensure that small

firms have a chance to bid against large companies?
MC CLOY: Yes. First, the list I mentioned contains many of the smaller manufacturers, and we reach additional potential suppliers through the Department of Commerce Bulletin and other media. Second, our analysis techniques protect the smaller company. A big company, for instance, might be able to bring in more people, spend more time in preparing their proposals, etc. We insure that the capabilities of each equipment proposed is measured against the stated system requirements.
DATAMATION: One of the factors considered in your evaluation are software capabilities. How do you measure this?
MC CLOY: The ideal way is to run the system; this gives you a better reflection of the software capability than anything else. This is quite expensive, and we can't do this on a capricious basis. It's also time-delaying to the user, who is anxious to go ahead with the selection. So you try to pre-establish what you think will be wanted; for example, you might say you have to be able to sort in a certain number of sort keys in a certain sequence.

Now, you really have two kinds of requirements. One is that which I will call a go-no go. The RFP says you must have a report generator; this you either have or you don't. If you don't, you don't compete; you are considered non-responsive. And these we state in the RFP in what we call Mandatory Requirements. You must be able to respond to this. Beyond that, some of them are effective in degrees. For instance, we may say that you must have a COBOL compiler for required COBOL. Then we might say for each optional feature of COBOL that you have, you get some extra consideration.
DATAMATION: How direct a concern to you is programming costs, and how do you evaluate or measure it?
MC CLOY: We have found that competitive equipment for similar system requirements have approximately equal programming costs. This is particularly true in the Air Force where policy calls for COBOL and a standard scientific language. We are looking for more precise techniques for measuring programming costs without requiring actual programming of systems. I guess we are all familiar with the lack of common data in this area.
DATAMATION: Do you have any evidence that COBOL, for instance, is lowering programming costs?
MC CLOY: I specifically have no such experience. Hq USAF recently put in Air Force-wide a single program on a variety of equipments and provided programs in COBOL. Some commands had machines with COBOL compilers; some did not, but used the statements as a specification language. It was estimated that the conversion by those with COBOL compilers took approximately one-fourth of the manhours of those that did not.
DATAMATION: How wide are the variations in the COBOL's that you're receiving as parts of proposals?
MC CLOY: For the most part the differences are coming in the options—they're basically complying with the mandatory requirements. Again, one of the things we're working on is standard COBOL benchmark programs, which will enable us to compare equitably competing equipment.
DATAMATION: You also mentioned vendor support in training, backup capability, maintenance support, etc. How do you measure or evaluate a company's training capability?
MC CLOY: To date, this has ended up, not intentionally, as a go-no go requirement. That is, we make as a requirement that you will provide a certain level of training, and either are willing to do this or not. Now we could, for example, say, "And I want to know how good your instructors are." How can I really evaluate that? One manufacturer may have the best instructor in the country, acknowledged as such, and the vendor may say, "He will instruct your people." But that individual could quit before the training occurs. So we specify a mandatory training requirement.

Now, we are beginning to develop a system—our own data system, you might call it—to collect data as to the effectiveness with which the manufacturers have met these requirements. One of the points we have made strongly to our user is that we expect to get data as to how both the manufacturer and Air Force met the commitments as specified in the RFP and proposal, and then we can build a file of data on the manufacturers' true reliability.
DATAMATION: How do you decide when to use the benchmark problem in your evaluation procedure?
MC CLOY: There are a variety of considerations, of course. A prime one is the importance of the system. For example, a recent proposal set a maximum monthly rental of $7,000. Well, you can't ask the manufacturer to do a lot of systems analysis and programming for a maximum monthly rental of $7,000. However, if the system is to operate at multiple locations, we want to be sure it will run effectively, and will require a benchmark. Then there are some applications for which it is quite simple to run a benchmark, primarily the scientific computers. Most of them have compilers which, when given the problem in standard language, take no time at all to compile, with little cost to the manufacturer. On the other hand, when we decide to run a benchmark we limit our selection to announced, physically available equipment and software, and exclude announced equipments not yet physically available. We also must be careful that we don't drive out the manufacturer who has good equipment but lacks the working capital that he is willing to risk. In the base supply package for approximately 150 computers, where they did run a benchmark, we estimated the cost to run in the hundreds of thousands of dollars to participate. As I said, we can't require this type of expenditure capriciously.
DATAMATION: Who selects the benchmark problem, and determines that it fully measures the capabilities of the hardware?
MC CLOY: The user writes the problem to our specifications, and we ensure that by running the problem the equipment will also do the job.
DATAMATION: Do you make recommendations to lease or purchase equipment?
MC CLOY: We always consider lease and purchase as options, and indicate where the break-even point would be. This decision is based upon the anticipated life of the system in the Air Force inventory. Hq USAF has to balance this in terms of fund availability and the latest potential for change in projected system life. It's a volatile world!
DATAMATION: The Air Force order for 150 computers, to which you referred, was for an inventory control application at numerous bases. What led to this "fleet order" decision?
MC CLOY: The Air Force, from the start, tried to establish and maintain a standard base supply system. But it began with punched card systems, acquired tube-type computers and, when transistorized hardware came out, had about five or six different pieces of gear. We found that when we tried to convert a standard system to non-standard gear, we didn't end up with a standard system.

What we found was that each major command had a programming office, and this was expensive. An airman who moved from one command to another had to be retrained for the different supply system established there. And when the Logistics Command, the granddaddy of the supply system, tried to get information on a common basis from all commands, they couldn't get it because the computers were producing different kinds of products. So their first problem was to have a standard supply system.

DATAMATION: Do you foresee a trend, then, toward more and more standardization?

MC CLOY: This is a general Air Force policy today. If you have a common requirement and a common system, this is often best satisfied with standard equipment.

DATAMATION: What kind of relation do you have with such organizations as MITRE and System Development Corporation?

MC CLOY: We have explored various automated techniques to be applied to this business with the help of MITRE. We have had some training provided by System Development Corporation, but no assistance in the equipment selection process from either.

DATAMATION: You have indicated your attempts to study the possibilities of "automating" the selection process. Are outside firms studying this for you?

MC CLOY: Unfortunately, our budget does not allow this. And also, I think it's healthy for our people to interlace this function with their other work, though I have found that this is difficult to do. The need to concentrate on day-to-day business negates a man's staying on research, so we try to rotate our personnel between necessary research and the normal selection work.

DATAMATION: What are your plans and hopes for future evaluation techniques, including an automated system?

MC CLOY: I feel strongly that evaluation is based upon system throughput capability and ability to do the job. I am very concerned about what I call the dimensions of the system. If I had the choice of having a system big enough or having a precise statement of running time, I'd be more concerned with having a system big enough for the job. The running time is subject to many vagaries such as programming capability.

We have found that our highly technical people have spent a fair amount of their time in drudgery, as in detailed timing analyses. I think it's highly important they use their skills for judgment rather than drudgery so that we need an automated timing and analysis capability of some sort. I think in order to do a better job, the best job we can conceivably do, we must have either a benchmark or simulate a benchmark through some system of automated analysis. I hope to do this. We've been looking at the kinds of things which are available, and hope within three of four months to come up with a specific plan on how to solve this one.

DATAMATION: I assume that you do not intend to automate out the human factor or final judgment, but only to take out, as you said, some of this drudgery.

MC CLOY: That is correct. This drudgery means translating general performance specifications into throughput, for instance—where five manufacturers who have all proposed equipments specify hardware characteristics and internal speeds. We get other factors on such things as software efficiency, make system inputs, and end up with a result. There are so many unknowns that are hard to anticipate—such things as multiprocessing, memory utilization, segmentation, a variety of these kinds of things that if timed blindly can produce an anwer that isn't reasonable. Once you get into these things, you find a myriad of decisions to make. An automated system forms a basis for judgment; it doesn't substitute for it.

DATAMATION: Is there a trend of leaning more toward one manufacturer? As you get a certain number of one kind of equipment for inventory control, and then possibly tie in the inventory control with a slightly larger system for producing management reports, and maybe that would then tie into a large system at Hq USAF, I wonder if this might tend to "lock in" one manufacturer?

MC CLOY: There are several factors which preclude this going to the degree you suggest. In the first place, there is a certain amount of compatibility growing in industry. As you know, for example, just about every manufacturer has the capability of using IBM tapes. So industry has, to some degree, taken care of the problem for us. The Air Force bends over backwards, as do we, in our review of specifications to prevent a specification which would preclude competitors. We could pay a cost for this—it can cost you money to convert, and so forth—but we would think over the long haul that this cost would pay us back in terms of increasing competition in the industry and, through competition, obtaining more for our AF dollars. Also, the Air Force has a communication system called AUTODIN. One of the functions of AUTODIN is to take in data in one mode, depending upon the data processing capabilities at the input station, and put it out at whatever mode they have at the other end. This takes care of the "different levels" problem you posed.

DATAMATION: What effort is made here to select equipment from different manufacturers—spread the work around?

MC CLOY: This does not affect our selections. When this office was set up, one of the ground rules given me was that we could not get involved in that type consideration. For example, we've already made a selection involving approximately 150 computers from one manufacturer. Do we make sure he does not get the next one? That's not our role in any manner, shape, or form; nor is it Air Force policy to select in or out any vendor except on the merits and cost of the case in hand. We make each selection on its own merits, based on the preestablished criteria I spoke of.

DATAMATION: I wonder if, say, a management consultant could take over this function for commercial enterprises?

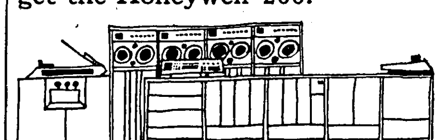
MC CLOY: Probably, but it would be useful only for a user who genuinely wants a competitive selection.

DATAMATION: What have you learned in your first year of operation?

MC CLOY: We've gained a lot of valuable experience from the four system selections we've made to date, and preparation of additional RFP's enough to feel establishing the office was a good idea. We've also learned that the entire area of equipment selection and evaluation has been rather naively approached by the industry and by users, and not enough attention has been given to it. Third, this is a very tedious and difficult task if the job is to be done adequately; we have to get into a great amount of detail. There are many unanswered problems: What is multiprocessing? What is the capability of equipment in multiprocessing? What is the implication on a given system requirement? Is what the manufacturer said realistic without actually running the problem? I think we're doing a pretty fair job in the light of the state of the art and the timetable we are operating on, and we're confident of the selections we have made.

DATAMATION: How have the users responded? Do they resent this as a usurpation of some of their functions and responsibliities?

MC CLOY: Much to my surprise, they're quite enthusiastic. In the past, the user was getting his lumps when he made selections. As I said, he wasn't skilled in it, and had no opportunity to become skilled in it. At a recent conference, to my surprise, the users said, "Thank God for you. It's your head now, not mine." ■



treading corporate waters

NON-DECISION THEORY

by DAVID BOURLAND, JR.

In the course of a recent project in naval operations research, it occurred to the writer that it was high time to apply one of the fundamental tenets of operations research: for the team to visit and experience personally the operation under study. Accordingly, we undertook a visit to the seashore to have a look at the ocean. The ocean was quite interesting, and not at all as we had imagined it to be. Surely this experience will greatly enhance our future effectiveness in naval operations research: "We were *there*."

One particularly curious thing happened. While observing the ocean in actual operation, one of our number stumbled upon a journal which had been washed up on the beach. This journal contained the fruits of obviously extensive deliberations, some of which had been obscured by sea water. However, it was possible to make out the basic structure of a theory of executive action which seems too important to keep from general circulation. The balance of this paper contains the fragments of what seems to be a "Non-Decision Theory."

the fundamental theorem

Still preserved, fortunately, is the most important portion of this work: the fundamental theorem upon which all else rests. The formal statement is:

> *The scientific administrator should never make a decision.*

The proof is quite straightforward (as, indeed, proofs are supposed to be, unless they can be made to be "elegant;" these latter are never straightforward), and is of two parts. Firstly, we observe that a decision is either right, or wrong. If it turns out that the wrong decision was made, it is clear that the theorem is trivially true. On the other hand, suppose that the correct decision was made. This would merely place the administrator in the position of having to make another decision on a future, more complex problem (which he would most likely botch). Hence, even the right decision leads to unfortunate results except for a set of measure zero. And therefore the theorem is true almost everywhere.

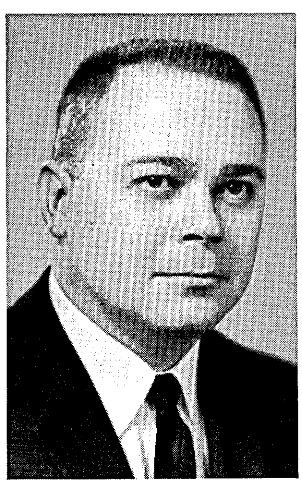
the command and control theorem

We turn next to the important Command and Control Theorem:

> *Subordinates should always be instructed, in difficult situations, to "Play it by ear."*

This follows directly from the Fundamental Theorem. Alas, it is becoming almost impossible to find competent junior personnel these days.

Since each person, in a given job, has only a single decision to make (this decision is of course apt to be wrong, and he will then be fired), a more appropriate unit is desired. This is needed because no matter how effectively an honest-to-God decision is avoided, one must still *seem* to make decisions. We therefore adopt for our basic unit of Non-Decision Theory 1/1000-th of a decision, or the



Mr. Bourland is senior staff engineer with Martin Marietta's Electronic Systems and Products Div. in Baltimore. He has been engaged in systems and operations research activties with Booz, Allen Applied Research Inc., Univac's Advanced Systems Research, and MIT's Operations Evaluations Group. He is a member of ORSA and the Institute of General Semantics, and holds an MBA from the Harvard Graduate School of Business Administration.

milli-decision, symbolized by "?". This symbol has been chosen in recognition of the uncertainty and mystery that always surrounds decisions (i.e., 1000 milli-decisions). This brings the present theory in line with the empirical law that:

What normally passes for a "decision" (i.e., a few milli-decisions) is actually three orders of magnitude away from the real, genuine article.

the dynamic case

It is asserted that:

$$\frac{ad}{at} = \Sigma D_i \text{ Ln } Di, \ O < Di < 1,$$

where D represents decisions, measured as usual in ?. The proof is left (thank goodness) as an exercise for the student. As a hint, it is pointed out that the quantity on the right is negative; the significance of this negativity should be apparent to those who have been following the emerging spirit of Non-Decision Theory.

hebert's last theorem

The following is quoted verbatim from Drew Pearson's column of 18 November 1961 as it appeared in the Washington *Post and Times Herald:*

[A contract of $296,000 has been let for]"Research in the General Area of Decision-Making under Uncertainty. Attention is to be primarily directed at decision situations characterized by the desire to optimize the value of some measure of accomplishment.

"Obviously," commented Congressman Edward Hebert, "this is about how to make a decision when there is no decision to be made."

This theorem (however apocryphal it may be) should stand for years as a challenge and inspiration for us all.

three back-to-the-wall theorems

It sometimes happens that the situation deteriorates so badly that a statement *has* to be made, involving a significant number of milli-decisions. The first step is to apply the following Meteorological Theorem:[1]

It always helps to know which way the wind is blowing in the front office.

This is, in effect, the inverse of the Command and Control Theorem. The proof is affected by merely changing the arguments of the direction cosines from x_i to $(\pi - x_i)$, for all i.

The following two theorems are asserted without proof; their elementary theoretical nature does not require it, despite their great practical importance. Firstly we give the Electronic Warfare Theorem:

If possible, get an expensive electronic device (i.e., a computer) to make the decision; if the decision turns out to have been wrong, one of its tape units can be disconnected and two programmers fired in retribution.

In the fullness of time, one's activities will eventually impinge unwantedly upon the preserve of some other person or group. Then appeal should be made to the Droit-de-Seigneur Theorem:

When it becomes obvious that a decision will mess things up significantly for someone else, the decision should be advertised as having been made by Higher Authority.

a danskin "interest" theorem

There is no least interesting decision.

The proof here is by contradiction. Suppose there were some least interesting decision, D_{MIN}, and some initial decision, $D_0 > 0$. Upon reflection it may be seen that D_{MIN} would be a *most* interesting decision indeed. Since it therefore would be more interesting than an appropriately chosen D_0, the theorem is proved.[2]

two probability theorems

It is probably desirable to include two theorems concerning Large Decisions. They apply not only to Bernoulli Trials, but also retroactively to Bernoulli's Tribulations. Let us first give the Weak Law of Large Decisions:

The larger the decision, the less necessary it is to make it, for it will be Overtaken By Events.

Consider the dynamic case, given above. Since the first partial derivative with respect to t is negative for all decisions, all tangents to the curve must cross the t-axis. Therefore for each tangent there is some t such that D=0 and hence it has been Overtaken. It is obvious that when D=0 the milli-decision expenditure rate is also minimal for real values of t. Hence the Fundamental Theorem is satisfied, and so the theorem is true.

We now generalize the above into the Strong Law of Large Decisions:

A sequence of milli-decisions is said to obey the Strong Law of Large Decisions if it converges, but to a value significantly less than is required for some action to be taken.

The proof involves showing that the Ault Criterion holds and that the third Chandler-Hamby lemma applies. It can be seen that the Weak Law of Large Decisions is a special case of the Strong Law.

future developments

The initial paper on this theory, based on the battered manuscript mentioned above, appeared in a trip report prepared as a Memorandum for the Director of Research in December 1961. In this day of explosive technological growth, it is to be expected that the state of the art provided by the original manuscript would soon be outgrown. This development is confirmed by the following items.

- Today, in research centers across the country, experiments are being conducted to reveal the secrets which lurk in micro-decision areas. The promises of nano-decisions lie just over the horizon.
- Rumors have leaked out concerning graphical techniques which make explicit use of non-decision space in order to take a variety of non-parametric methods into account.
- Several particularly promising doctoral candidates at the Harvard Graduate School of Business Administration are out collecting case material even now for a course to be offered soon entitled "Non-Decision Theory for Middle Management."

Perhaps the most telling indication that confirms the wide acceptance of Non-Decision Theory is provided by a small brochure in last week's mail. A West Coast university will offer, next fall, a two-week cram course on this topic. Taught by obvious experts in the field, the enrollment fee seems quite reasonable at only $4500. ■

---

[1] Sometimes called the Buys Ballot *Theorem*. One form of the completely unrelated Buys Ballot's *Law* goes like this: If the wind is blowing on my back (i.e., the plane of back is perpendicular to the axis of wind flow in horizontal projection), and the low pressure area is to my left, then I am in the Northern Hemisphere.

[2] This theorem, and its proof, constitute a Non-Decision theory analog of a theorem in number theory due to the mathematician J. M. Danskin, which concerns the non-existence of a "Least interesting number." Taken together these two theorems point the way toward a non-existence theory, which suggests an accompanying non-uniqueness theory. So far as the writer knows, Dr. Danskin has not published his result in number theory.

FLOORS FOR COMPUTERS

selection and care

by C. T. POTTS*

Often overlooked in the purchase of electronic data processing equipment that costs hundreds of thousands of dollars is adequate preparation of a room to house the equipment. This oversight can cause the purchaser a great deal of inconvenience and expense.

Many customers fail to realize that complicated electronic equipment requires a special raised or pedestal floor. If the floor is not ready by delivery, the customer is faced with the prospect of paying for high priced equipment that remains idle until the site is completed. In addition, the customer may end up with a flooring pattern that does not particularly suit him because he has to take what is immediately available.

Pedestal floors were developed specifically for use with data processing equipment. As the name implies, a pedestal floor is simply a raised floor set on pedestals, or pedestals and rails. The space between the floor and the subfloor is needed to contain the electronic cables and, in many cases, acts as the plenum chamber for air conditioning.

In an electronic data processing room, the floor has many jobs to perform. A pedestal floor must be high enough to contain the electronic cables and cable heads—usually six to eight inches. It must be strong enough to support heavy static loads, and have removable panels so that there is easy access to the cables for repairs, installing new equipment or changing locations of existing equipment. Further, the floor should have an even fit to contain the conditioned air and to prevent dust from filtering between the joints. In some cases, it might be desired to have the floor grounded to carry off static electricity.

At the minimum, pedestal floor manufacturers require six to eight weeks to manufacture and install a floor as specified by the customer or architect. Floors have been installed in much less time on rush orders, but the customer then has little choice in the style and design of floor finish, and usually has to take what materials are available. In general, the longer the customer delays ordering a pedestal floor, the more difficulty he is going to encounter in getting a floor installed to his satisfaction.

There are a number of pedestal flooring systems available but essentially they fall into two types—stringer and stringerless.

In the stringer system, pedestals or jacks are adhered to the subfloor and stringers or channels are connected from one pedestal head to another. Floor panels are then set on top of the stringers (see Fig. 1).

In the stringerless or pedestal system the panels are set directly on top of the pedestal heads and held in place by various locking systems (Fig. 2.).

Panels can be easily removed in both systems by merely attaching a suction cup device and lifting. While the panels must be strong enough to support machines, they must be light enough for easy lifting.

Pedestal floors are made of aluminum or steel as well as combinations of materials such as aluminum and steel, or steel and plywood. Most popular sizes are 24″ x 24″ and 18″ x 18″.

Most panels are surfaced with resilient tile flooring, either vinyl-asbestos or homogenous vinyl. A strong contact adhesive is used to bond the tile to the panel.

Armstrong Cork Co., which manufactures both types, recently conducted personal interviews with representatives of the six leading pedestal flooring manufacturers in the East and found a growing trend to the use of vinyl-asbestos tile.

Three basic reasons were cited:

1. Economy. Vinyl-asbestos tile sells for less than half the cost of vinyl tile, and is just as durable. One pedestal flooring manufacturer called it "The best floor for the money."
2. It is easy to maintain.
3. It is familiar in appearance. Vinyl-asbestos is probably the most widely used flooring materials for commercial buildings.

There is a wide choice of color and design in both vinyl-asbestos and vinyl, provided the customer orders the pedestal floor in plenty of time to enable the manufacturer to obtain the selected tile. Basically, however, the

Fig. 1
Stringer system offers stronger support, decreases lateral movement. Note suction cup lifting device.

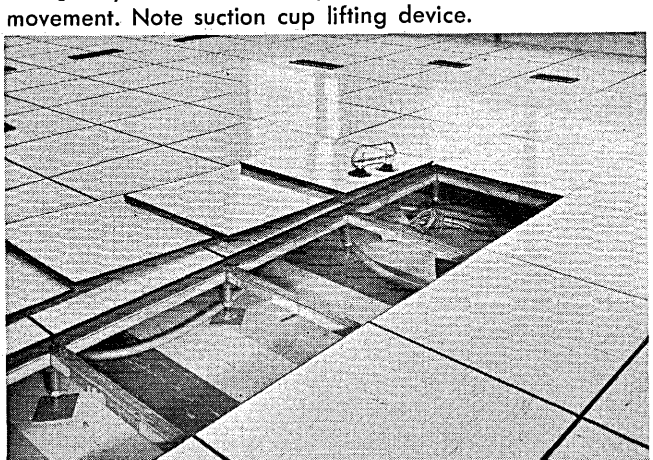
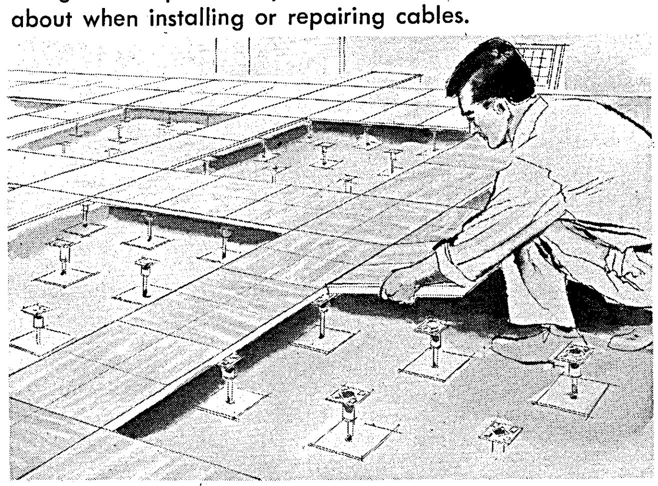


Fig. 2
Stringerless or pedestal system has no cross sections to worry about when installing or repairing cables.



*Mr. Potts is manager of Fiber and Surfacing Products Sales of the Industry Products Div., Armstrong Cork Co., Lancaster, Pa.

trend is for subdued colors to complement the electronic equipment—neutral colors with a subtle graining.

maintenance

When rolling equipment into place over newly-installed floors, some form of protection should be provided to prevent dynamic load damage to the tiles. A sheet of plywood or similar material, laid down in the path of the moving casters, will help distribute the weight of the machine.

Use of hard rubber or aluminum caster cups beneath the legs of heavy computer equipment will help prolong the lift of the resilient floors. To assure limiting the load to 200 pounds per square inch, the size of cups or furniture rests used must be calculated on the basis of the diameter and width of the caster and the weight of the hardware.

Resilient flooring—the generic term for linoleum, rubber, cork, vinyl, etc.—is the easiest of any material to maintain but, because a number of maintenance materials can be harmful to electronic equipment or wiring connections, special consideration is required. For example, floor wax, steel wool, excessive amounts of water, use of poor quality buffing pads, and cleaners containing ammonia or chlorine should be avoided.

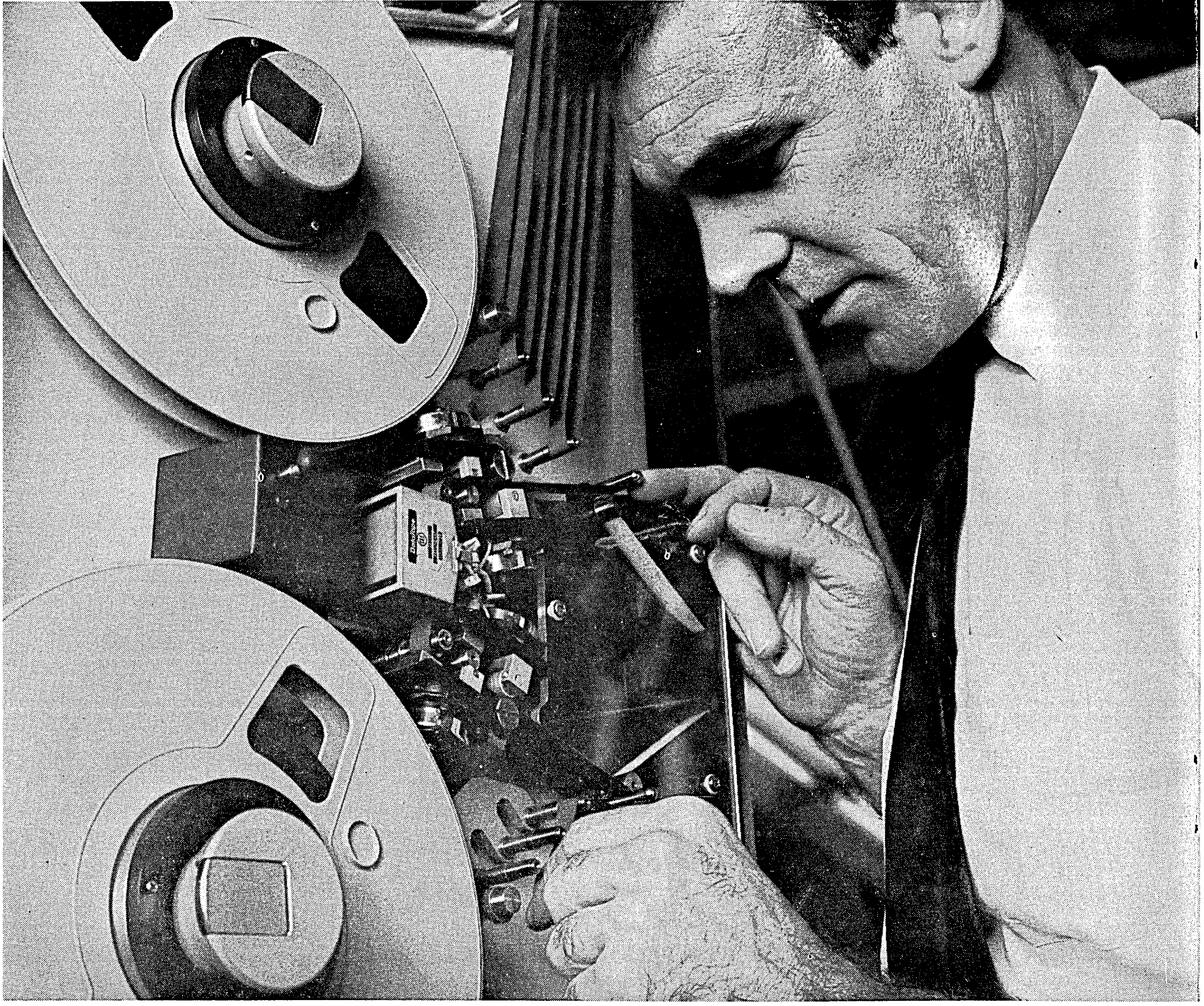
Two thoroughly-tested methods are recommended to provide safe, practical maintenance for computer areas:

Using standard maintenance equipment

Cleaning—Apply a small amount of good commercial floor cleaner diluted approximately one part cleaner to 40 parts cool water, with a mop, being careful not to allow the solution to seep between the individual panels. Next, scrub the floor with a fine (polishing) grade nylon pad used under a power machine. Take up the spent cleaner with a mop or wet vacuum. Let the floor dry thoroughly.

Finishing—Apply a thin coat of a heavy duty commercial floor finish with a mop by dipping the mop in the finish and squeezing out the excess liquid with a squeeze-type wringer. Mop on the finish using a side-to-side motion, turning the mop frequently. Repeat this process until the entire floor area is covered with a thin film of polish.

Maintenance—The floor area should be swept daily with a treated dust cloth or mop to pick up loose dirt or dust particles. Burnish the tile occasionally with a fine grade nylon pad under a power machine. Then go over the burnished area with a dry vacuum pickup unit to remove any dust. For best results, this should be done as a two-man operation. Frequent dry



burnishings will reduce the frequency of floor cleaning.

Using an automatic floor maintenance machine

This method is recommended for larger floor areas.

Cleaning—Apply a small amount of commercial floor cleaner diluted approximately one part cleaner to 40 parts cool water, with an automatic floor maintenance machine. The machine will loosen soil, pick up the spent solution, and dry the floor in one operation. If desired, maintenance machines that will also rinse the floor can be obtained for an easier, one-pass operation.

Finishing—Same as method one.

Maintenance—Follow the steps in method one, except that the burnishing can be done with the automatic floor machine which also will serve as a dry vacuum to pick up dust from the nylon pad.

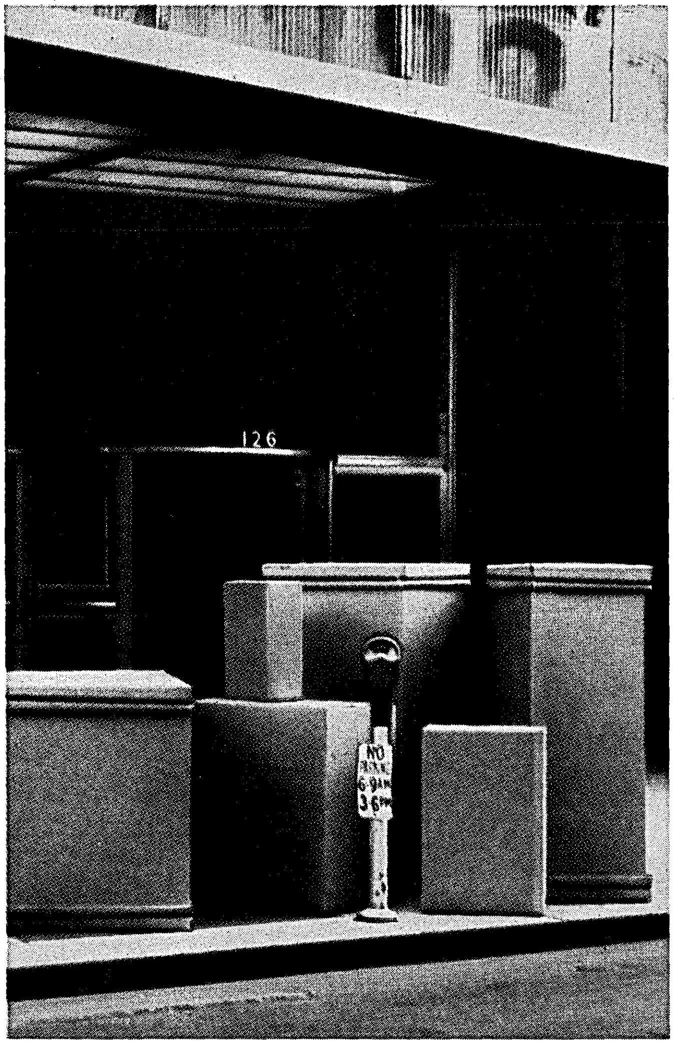
For heavily soiled floors (caused by neglect or heavy wear), the following steps are recommended:

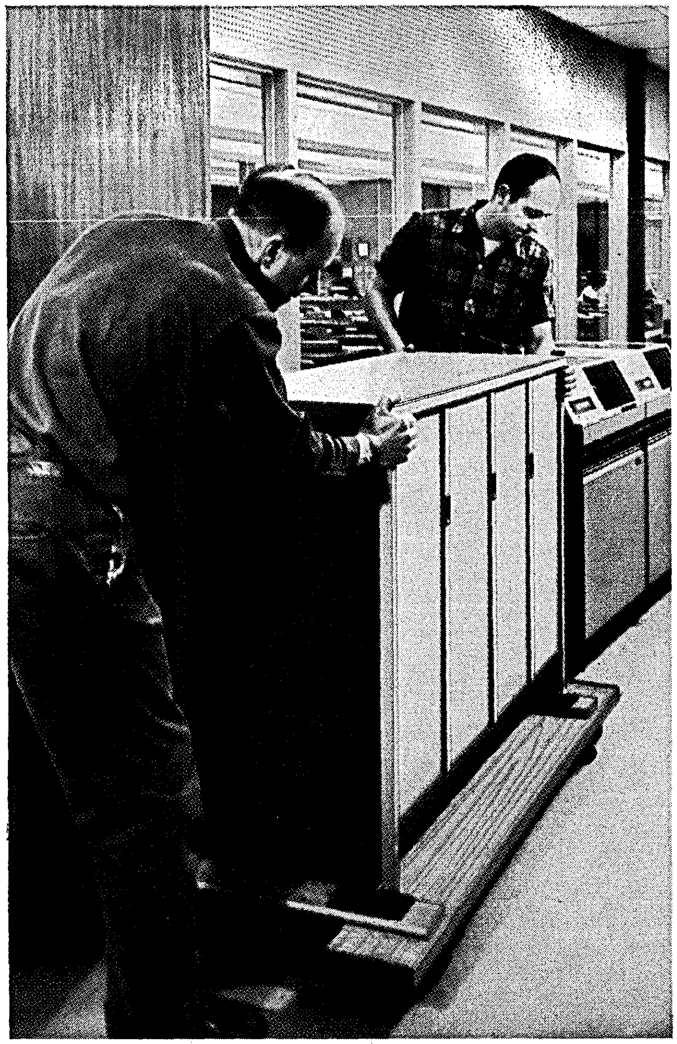
1. Clean the floor as previously described. Rinse and let dry.
2. Apply Clorox full strength with a mop that has been wrung out in a squeeze-type mop wringer. Do not allow the liquid to seep between the panels.
3. Allow to dry. Then rinse with clear water (again, keeping the liquid from seeping between the panels.) The Clorox will harden the surface and make it more resistant to both heavy traffic and further soiling.
4. When all traces of the cleaner have been rinsed from the floor, let dry, and apply a heavy duty commercial floor finish according to the directions in method one.

If the floor is so heavily soiled that the first treatment does not restore its appearance, repeat the above method after 24 hours. No floor polish should be applied until after the second treatment.

Companies installing data processing equipment need not become experts on pedestal flooring systems and their maintenance. However, they can eliminate a lot of future problems by ordering their pedestal flooring well in advance of equipment delivery. Basically, all they need to know is that a pedestal floor is necessary, the types of pedestal floor systems available, the types and patterns of floor tile desired, and that maintenance of a computer room floor requires certain precautions. Contact the pedestal flooring manufacturer for help in selection, then give him the lead time that lets him provide a trouble-free installation. ■

We go farther than the sidewalk:





Datacase is sold, installed and serviced locally!

■ If the service you've been getting from your DP auxiliary supplier stops at your door, you'll be delighted to do business with your local Steelcase dealer. He's the one who handles Datacase—the color-coordinated auxiliary units with the "computer look" made by Steelcase. He has the experience and knowledge to help you plan or amplify your DP department, to create offices and machine rooms that are as efficient as they are attractive. And, as we said, he not only sells, but delivers, installs and services locally. Call him as your machine room needs arise. He's in the Yellow Pages. Steelcase Inc., Grand Rapids, Michigan, and La Mirada, California; Canadian Steelcase Co., Ltd., Don Mills, Ontario.

DATACASE®/BY STEELCASE

CIRCLE 29 ON READER CARD

people IN DATAMATION

■ George W. Fenimore has been appointed international vp of the Bunker-Ramo Corp. He will be in charge of all overseas activities which include the manufacture and marketing of industrial control computers and associated equipment.

■ Ray W. Macdonald has been elected executive vp of the Burroughs Corp. He was formerly vice president in charge of the International Div.

■ James J. MacIsaac has been appointed vp and general manager of the Data Processing Systems Div. of SCM Corp. He was previously director of business planning for the Diebold Group Inc.

■ John J. Kramer has been appointed managing director of Control Data Corp. in Europe. Previously, he served as eastern regional sales manager for RCA's Data Processing Div.

■ William A. Ogletree has assumed the post of acting general manager of the Advanced Scientific Instruments Div. of Electro-Mechanical Research, Inc.

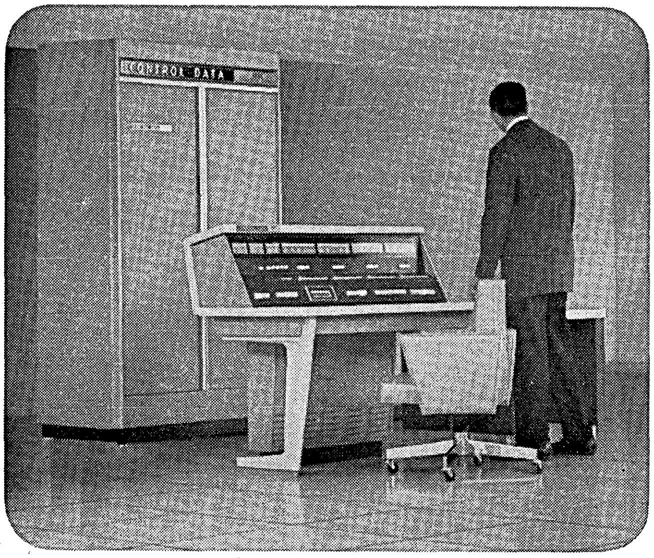
■ Norman F. Schneidewind has been appointed manager of commercial data processing projects for Planning Research Corp. He was formerly manager of central data processing at Title Insurance and Trust Co.

■ James E. Connors has assumed the position of manager of RCA's New Systems, EDP Div. Connors will be responsible for marketing systems support for new products. He was previously with Crucible Steel Co. of America.

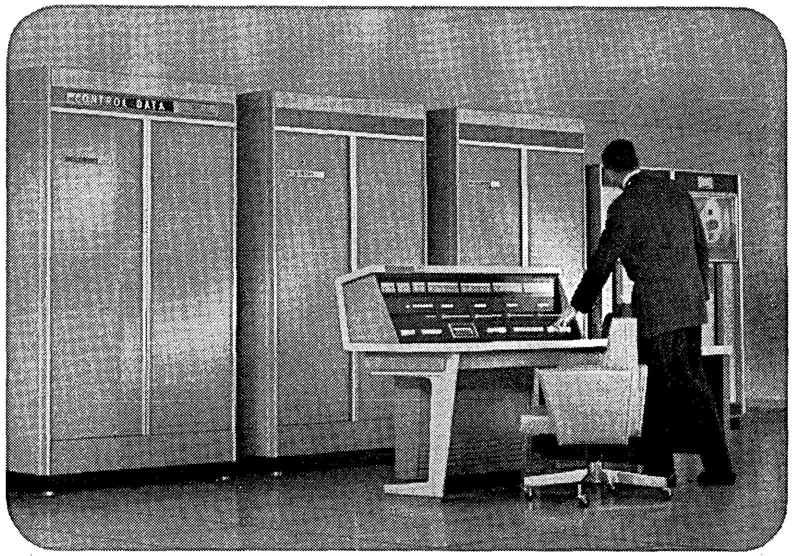
■ Seymour Schoen, former manager of the Computer Dept. of Litton Industries, has been appointed director of the computer systems laboratory at the Guidance and Control Systems Div. of that company.

■ Fred Cornish, a specialist in computer-based management sciences, has joined the Information-Management Sciences Div. of the Auerbach Corp. as program manager.

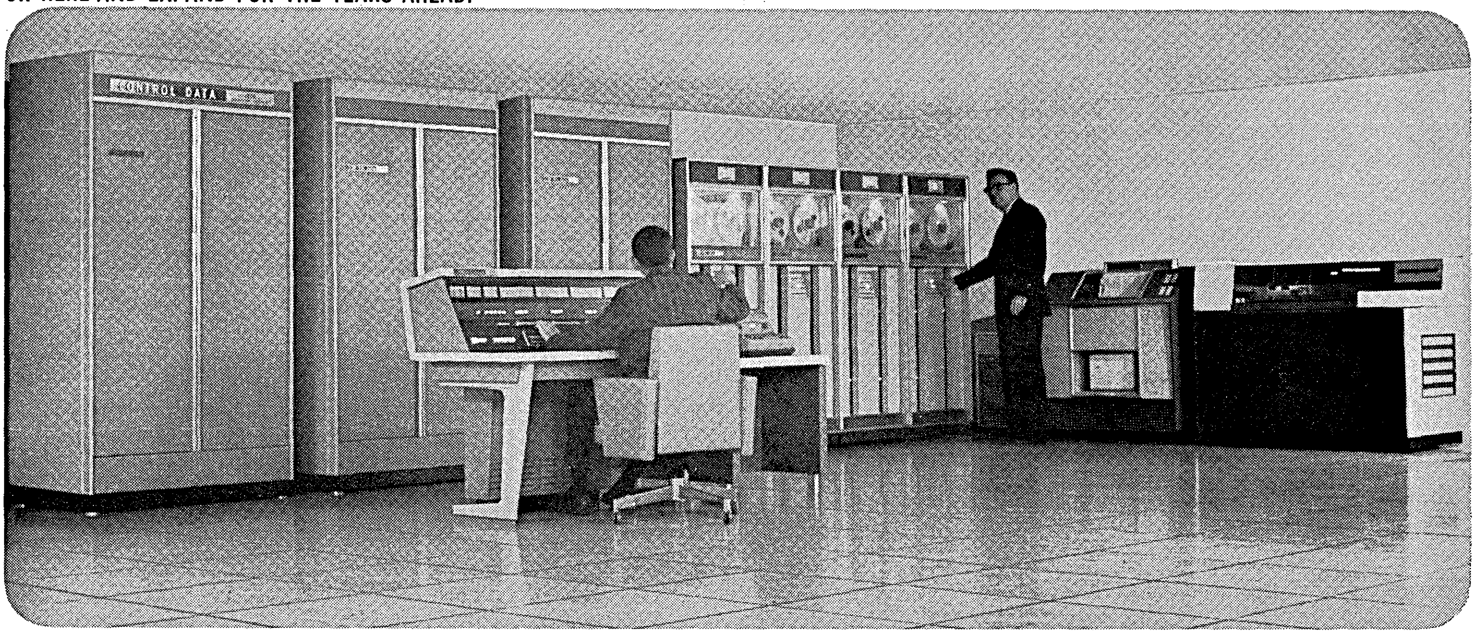
START HERE WITH A BASIC SYSTEM . . .



OR HERE WITH EXTENSIONS . . .



OR HERE AND EXPAND FOR THE YEARS AHEAD.



THE CONTROL DATA® 160G ANSWERS YOUR QUESTIONS ABOUT COMPUTER SIZE AND COMPUTING ECONOMY

Here is how the **CONTROL DATA 160G** Computing System answers some of the most pressing questions about computer size and computing economy:

How much computing power do you need to handle your present data processing operations?

Depending on the scale of problems to be solved, you can select, by modules, the exact size 160G system you need—according to computing capacity, memory size and input-output arrangement.

How about system growth to accommodate your problem growth?

Complete modularity of the **CONTROL DATA 160G** makes available required system capability whenever you need it—as many as nine compute modules can access one memory module; magnetic core storage is expandable from 8,192 to 131,072 words with 1.35 microsecond cycle time; you can expand input-output data channels on one compute module to give you seven buffered channels and one non-buffered.

How can you maintain a high rate of computations-per-dollar with an expanded 160G system?

Expansion of the 160G is economical, does not interrupt the operation of an existing system and requires minimum rearrangement of your site environment. Also, an existing **CONTROL DATA 160-A** system can be upgraded in computing capacity without the retraining or reprogramming required when a totally different system replaces one in service.

If yours is a "new use" problem, system replacement or system expansion, investigate the advantages of the **CONTROL DATA 160G,** which fully utilizes all of the field-proven 160-A programming packages. Contact your nearest Control Data representative.

CONTROL DATA CORPORATION

8100 34th AVENUE SO. • MINNEAPOLIS 20, MINN.

OFFICES: Alamogordo • Albuquerque • Atlanta • Beverly Hills • Boston • Cape Kennedy • Chicago • Cleveland • Colorado Springs • Dallas • Dayton • Denver • Detroit • Downey Honolulu • Houston • Huntsville • Ithaca • Kansas City, Kan. • Minneapolis • Newark • New Orleans • New York City • Oakland • Omaha • Philadelphia • Phoenix • Pittsburgh • Sacramento • Salt Lake City • San Bernardino • San Diego • San Francisco • Seattle • Washington, D.C. • Frankfurt, Germany • Hamburg, Germany • Stuttgart, Germany • Lucerne, Switzerland • Zurich, Switzerland • Melbourne, Australia • Canberra, Australia • Athens, Greece • London, England • Oslo, Norway • Paris, France • Stockholm, Sweden • Mexico City, Mexico • Ottawa, Canada • Tokyo, Japan

A FEDERAL EDP STUDY

to set policies, legislation

A study of automatic data processing in the executive branch of the federal government has gotten underway with the formation of a project staff, headed by Carl W. Clewlow. A report to the director of the Bureau of the Budget is scheduled for the end of May, with ultimate transmission to the President and the Congress by June 30, 1964.

The study is a culmination of the legislative activity around Capitol Hill over the past two years, which resulted in the so-called Brooks Bill, H.R. 5171, and the hearings conducted by the House Post Office and Civil Service Committee under the chairmanship of Congressman Arnold Olsen of Montana. The latter committee, after extensive hearings, recommended to President Kennedy in September 1963 that a study of the type mentioned above be conducted and the reports to the Congress made by June 30.

main objectives

Objectives of the study, announced last December, are the recommenda-

tion of policies, administrative organizational improvements and legislation "in the public interest."

The broad range scope of the study includes all federal dp activities and establishments, areas in which relationships with state and other governmental bodies are involved, as well as contractors whose computing activities are in part federally supported. It includes the complete range of information processing equipment.

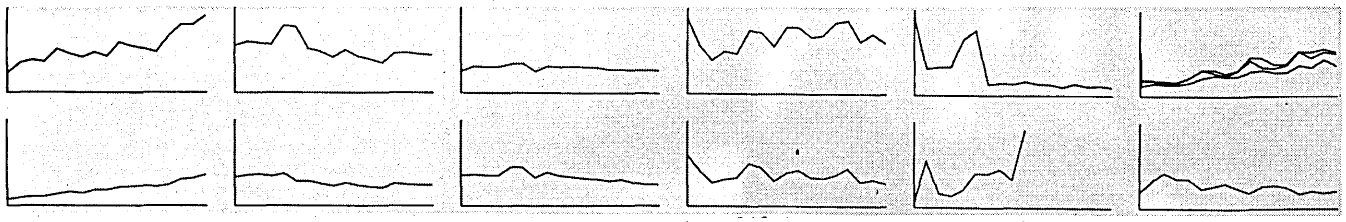
four topics to be studied

Topics to be studied include edp organization and management, systems planning (with emphasis on questions concerning centralized control and standardization), manpower and personnel (including unemployment effects), and equipment management. The last item includes contractual arrangements; purchase vs. lease; criteria for selection; "prudent use;" system efficiency; maintenance; redistribution or disposition of excess gear; compatibility and standardization.

heads staff of eight

Mr. Clewlow formerly served in several high-level administrative capacities for the Treasury and Army departments, and is on leave for the duration of the study from Arthur Young & Company, management consultants. He heads a staff of eight representatives of federal departments and agencies. These are: Joseph F. Cunningham, associate director of Data Automation, Department of the Air Force; W. Howard Gammon, assistant to the director, National Bureau of Standards; Martin Hochdorf, chief of the Computing Center, Tennessee Valley Authority; Donald B. Rock, Chief, ADP Evaluation Branch, Federal Aviation Agency; Richard G. Shook, chief, Program Development Section of ADP Branch, Office of the Controller, Atomic Energy Commission; William H. Smith, assistant commissioner for Planning and Research, Internal Revenue Service, and Clark R. Renninger, management analyst, Office of Management and Organization, Bureau of the Budget.

The report will be reviewed by the Advisory Committee, headed by Robert Ramspect, former Congressman and chairman of the Civil Service



FOR PENNIES, YOUR COMPUTER CAN PLOT DATA DIRECTLY IN EASY-TO-READ GRAPHIC FORM

Why try to visualize the significance of masses of digital data—or plot it laboriously by hand? The new EAI DATAPLOTTER® Series 3110/3120/3130 automatically plots digital information on-line from almost any digital computer or from punched cards or punched tape. It will plot lines, points or symbols with extreme accuracy at speeds up to 120 points per minute on charts up to 10" x 15". YOU CAN RENT IT FOR $3.50 AN HOUR.

If you want to get more out of your computer-generated data, write for information on the new EAI DATAPLOTTER today.

EAI®

ELECTRONIC ASSOCIATES, INC., *Long Branch, New Jersey*

ADVANCED SYSTEMS ANALYSIS AND COMPUTATION SERVICES/ANALOG COMPUTERS/HYBRID ANALOG-DIGITAL COMPUTATION EQUIPMENT/SIMULATION SYSTEMS/SCIENTIFIC AND LABORATORY INSTRUMENTS/INDUSTRIAL PROCESS CONTROL SYSTEMS/PHOTOGRAMMETRIC EQUIPMENT/RANGE INSTRUMENTATION SYSTEMS/TEST AND CHECK-OUT SYSTEMS/MILITARY AND INDUSTRIAL RESEARCH AND DEVELOPMENT SERVICES/FIELD ENGINEERING AND EQUIPMENT MAINTENANCE SERVICES.

CIRCLE 28 ON READER CARD

Committee. It is comprised of agency heads, academicians and industry representatives.

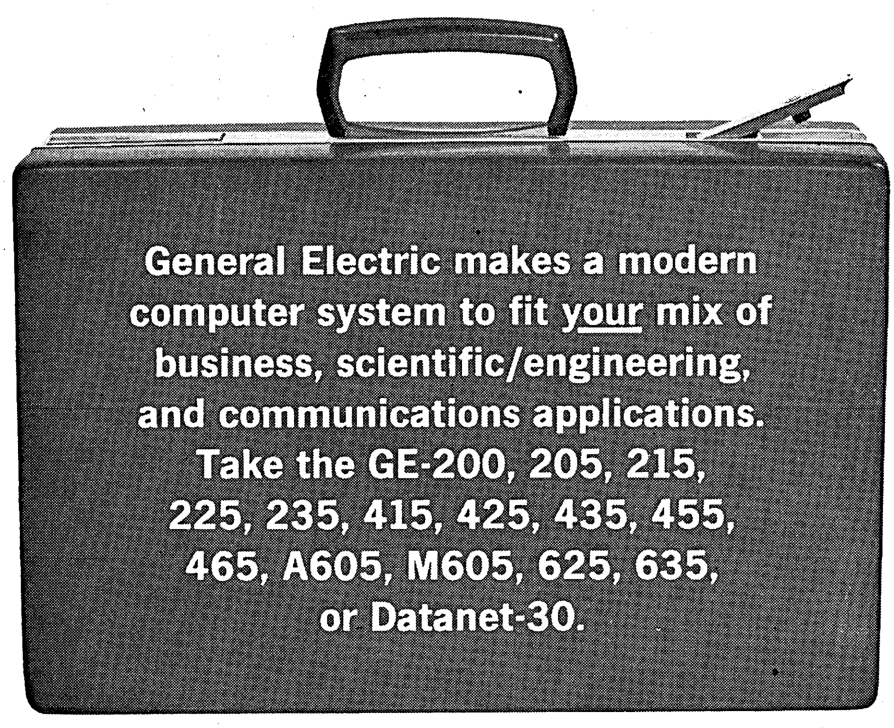
study staff reviews installations

The study staff has been visiting various government installations, government installations operated by contractors, using organizations, software firms, corporations whose data processing activities are as far-flung as those of the federal government, research organizations and individuals prominent in the dp community to review the various problems which the federal government faces and to solicit advice on any and all phases of the ADP management problem. The study staff has available a complete inventory of all of the equipment of the federal departments and agencies which was concurrently in the process of being reported as a result of a directive from the Bureau of the Budget. Representatives from all the manufacturing organizations are being contacted to determine: (1) Areas in which the government can improve its present methods of operation and (2) At the policy-making level of the organization, the impact which a major modification in government practices or policies would have on the industry as a whole. Presumably, Olsen's report, plus the various reports of the General Accounting Office, and other information provided voluntarily to the committee, will be used as a basis for the study.

In addition to the project staff members above, other government employees have been assigned to task forces working on various phases of the overall study.

how to manage

At the core of this activity is a determination whether or not automatic data processing in the federal government should be managed in a way different from the general management scheme of other assets which are used in the discharge of the government program responsibilities. If so, it can be anticipated that some major changes in government approach to ADP will be recommended by the project staff. Any thoughts or suggestions might be sent to Clewlow at the Bureau of the Budget, Washington 25, D.C. ■



the fun begins

SYSTEM/360's INITIAL IMPACT

by ROBERT B. FOREST

The giant dropped the other shoe last month. As usual, the thud was less shattering than the anticipated version. But then what could possibly equal the power and the glory of the mythical machines created in the minds of computerites by imagination working on the partial and misleading information in the weekly trade press?

It's difficult this early in the game to assess the real, the long-range impact of IBM's new System/360 . . . harder yet to separate fact from ballyhoo. But for what it's worth, here is a report on some of the initial reactions to what has to be the industry's biggest announcement yet, whether it's measured by the weight of the hefty press kit, or the number of units introduced. (Estimates are that IBM has spent 300 megabucks to date on the project, expects to see this rise to $500-million . . . hopes to sell 20,000 systems).

The internal IBM reaction could be characterized as quiet, smug elation. One office is supposed to have sold its yearly quota on A (Announcement)-Day. In Texas, a man allegedly interrupted the 360 presentation to demand he be allowed to order one right then . . . which sounds like a combination plant and a new version of the rich Texan jokes.

The competition helped the daily press get their stories filed with the expected statements on their reactions, then turned quickly to a more intensive study of their next marketing target. Inside comments varied somewhat from the official. "The 360 looks like a clean machine," said one competitor. "And it took a great deal of courage for IBM to break with its past product line."

The courage involves opening the door to that chamber of expense horrors, reprogramming. (Cost estimates range from $100K to several $-million). To ease the transition to the 1401, IBM offers stored-logic circuitry which will allow "most" 1401, 1440, 1460 programs to run as is on the models 30 and 40. GE offers the same technique in its 415. But these 1401 programs will not take maximum advantage of the 360, and should probably be reprogrammed eventually. By using the Honeywell 200 Liberator software approach, the user allegedly gets his reprogramming done automatically . . . although it's not clear that such new programs will be as efficient as those which are hand-coded.

The problem for the 7000-series user is a little trickier. Although IBM is "looking at" a hardware simulation technique similar to that available for the 1401, odds are against it. This is partly because the 90—most popular in the 7000 line—has more peculiarities, is harder to simulate than the 1401. IBM will suggest what they call "graduated transition" . . . replacement of a peripheral 1401 with a mod 30, which can be used to debug 360 programs as they are developed, and expanded gradually as the 360 program library grows.

The tricky transition question is clouded by the addition to IBM software of NPL (New Programming Language) which stemmed from the work of a six-man joint IBM/SHARE committee.

But reprogramming considerations aside, the 360 announcement has to worry the competition considerably . . . partly because anything new from IBM creates an automatic bandwagon effect, partly because the completeness of the new line offers less reason for people to look outside.

The outfit which looks on the surface to be least hurt is GE, with its "400 compatibles" line. The 415 is price competitive in the 32K-memory, four-tape configuration. But GE lacks a mass random access device, a shortage which will probably be corrected within a year or two. With a brand new marketing director, the company has to clarify its marketing philosophy, figure out a way to focus its impressive technological and financial muscle on computers in a highly decentralized, profit-oriented operation. The company so far seems unwilling to recognize that selling computers differs significantly from selling any complex electronic gear.

Unlike GE, RCA and NCR have random access devices. RCA points out its 3488 has greater capacity than the CRAM and more speed than IBM's announced counterpart. RCA and NCR, however, lack a complete, compatible family. This despite the former's claim that its 3301 covers "a major portion" of the 360 line, and that its cost/performance ratio beats that of models 40 and 50.

Honeywell can probably still be expected to make some inroads into 1401 territory, if at a less accelerated pace, if its Liberator lives up to advance billing. The low end of the 200 is only slightly more expensive than its 360/30 counterpart; expanded punched card and small mag tape configurations are slightly cheaper . . . and delivery favors Honeywell here.

Control Data has the beginnings of a family with some compatibility in the 32-34-3600 line, and can be expected to come out with something under the 3200 this year. At the upper end, the 6600 may have a year—or two at the most—of breathing room before the first 70 is delivered. But 6600 marketing life will undoubtedly be shortened.

The biggest question mark: the smaller competitors . . . SDS, Packard Bell, ASI, DEC and the like. At first look, the extensive IBM line appears to hit these people pretty hard. Their biggest hole card: the ability to marshall a small engineering crew to create in six months to a year a new system, without massive R & D overhead.

360 technology

IBM underplayed somewhat the "solid logic technology" of its new line. Despite some half-hearted "third generation" talk, the main claim to fame of the new hybrid (as opposed to integrated) circuitry is its amenability to low-unit-cost mass production. This, plus greater (but hard-to-measure) maintainability and reliability, seem to more than offset at present the integrated approach, which offers some potential speed advantages, but remains unproven on the other grounds.

user view

The man who will have the final word on the impact of the 360, the user, viewed its announcement with pretty much the same mix of curiosity, delight and criticism which has characterized reactions to previous unveilings . . . with perhaps a shade more skepticism showing. Many who have placed multiple 360 orders point out that this is a standard delivery protection device . . . that a lot of hard study, evaluation and question-answering must precede installation.

One '90 user with a heavy machine-language program investment is especially doubtful of his management's willingness to spend reprogramming bucks. Another feels his company has enough '90 power to satisfy its requirements for the next four to five years. Characterizing the 360 as "more muscle, a giant step backward," he maintains that what is needed now is "smart"—development of techniques to take maximum advantage of what's now in house. But his company has ordered some 360's anyway.

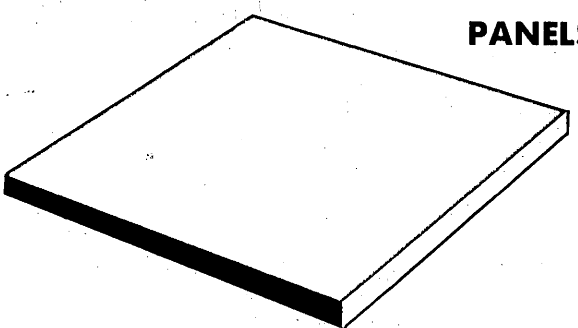
A commercial data processor who has ordered two 30's and a 40 (added to earlier orders for current IBM, Honeywell and Burroughs gear), says he's "somewhat impressed" by the 360's I/O flexibility. He'd use the 30's to replace a 1401/1460 and almost halve his rental. And he likes the compatibility, which should keep him from reprogramming himself "into a hole" as he's done in the past. He's still talking to the competition, but feels that some of them, as "relative newcomers" must do more to prove themselves than IBM, which has done a better job of delivering on its promises so far.

Another feels that the economic incentive (rental cuts of 50 per cent for 7080, 7090) will force him down the 360 route. And he thinks 360 interrupt features will open the door to real-time applications which can be approached on an incremental basis impossible before. The large core doesn't excite him—"we've misused core"—except for linear programming work.

One maverick doesn't share the enthusiasm of his company, which ordered "plenty" of 360's within an hour of the announcement, without price agreements. He likes the idea of the mass random access memory, but em-

BEFORE YOU CONTRACT FOR ELEVATED FLOORING

IT PAYS TO TAKE A SECOND LOOK!

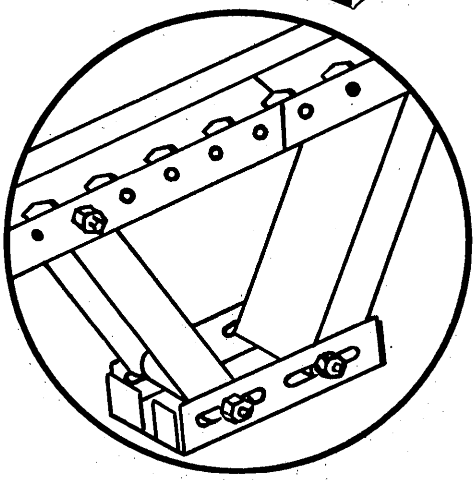


PANELS

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

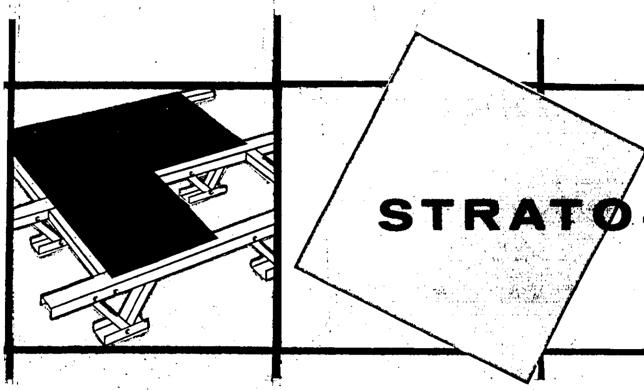
For maximum strength, minimum deflection, the STRATO-FLOOR sub-structure features the patented STRATO-TRI-JACK, which utilizes the strength of the triangle in its design.

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. ⅛" 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 26 ON READER CARD

phasizes he hasn't seen it in action yet, as he has RCA's unit. Head of a decentralized operation, he sees the possibility of reducing the configurations of his remote installations. "Suspicious by nature," he can't imagine nothing better will appear within 10 years.

One careful man thinks its 'way too early to decide anything about the 360, except that this kind of gear calls for a "good, hard systems look. Does it mean we just shift what we're doing now over, or use it for the first pass at a true systems approach," he asks. And he wants to see what's in the compilers.

experts anonymous

Consultants, independent hardware and software experts also want more information. "It's not the beast the '90 was," says one, "but the I/O gear has been available elsewhere for years. And they haven't paid sufficient attention to multiprocessing. The bulk core, at five cents a byte, looks interesting, but by the time problems get that large, 8 usec will kill you." Software? "They've got a lot of tough programming to do." (IBM says it will spend over $10-million in programming development).

An independent software manufacturer thinks IBM is overlooking techniques which allow development of smaller compilers. "The 360's compatibility should permit one compiler (per language) and one operating system for the whole line," he concludes.

A time-sharing enthusiast and command-and-control software expert likes the 360. He's curious about how they'll solve the reprogramming problem.

If there was less than unanimous enthusiasm for 360 technical prowess, observers generally admired IBM's marketing skill. "They touched all the bases," says one. By announcing now, the company fights off increasingly penetrating raids on current installations. At the same time, 18-24-month delivery insures their life for that much longer at least.

And we'd guess that the last 360 announcement is not in. You can look for models 10 and 20 . . . and maybe something *really* big above the 360/70. Later, as competition appears, IBM can be expected to hop up core speeds.

The competition may feel that with the dropping of the other shoe their target is out in the open and clearly defined . . . but it's an awfully tough target.

"Glad we bumped into you," Marv said as I rubbed my bruises.

"I want you to meet Carson Maim, who is coming to work for us. Carse, this is Paul Daly, our senior research engineer in charge of data reduction."

"My pleasure!" An over-diameter young man with a high C.G. grabbed my hand and ground it into what felt like a gruesome tangle of pulp and splintered bone.

"I'd appreciate it, Paul," Marv went on, "if you'd show Carse the D.R. system."

"Glad to," I gritted between clenched teeth. "I've just got time before I'm due at Emergency Hospital."

"Fun-nee," Marv said. "Carse, would you wait over there for a minute?"

"Glad to," said the megawatt power supply, moving off down the aisle.

"What's up, Marv?"

"His bowling score. Up around two twenty."

I was impressed. "Makes up for losing Evans, huh?"

"We had to do something to stay on top of the league. Now listen, Paul. Carse may not be much of a data handling engineer, but I'm counting on you to turn him into one before Mr. Robertson gets around to looking him over. Otherwise we'll be minus our only chance of snatching that trophy for keeps."

I started the guided tour at the tape transport cabinets.

"We receive the FM-FM flight test telemetry on these tapes," I told Carse.

He made a noise that sounded like "nngg." I hoped it meant I was getting through to him.

"First we put it through these discriminator banks and this analog multiplexer."

"Nngg."

"Next," I said, "we come to the Analog-to-Digital Converter where the pulse-amplitude data is digitized before it goes to the buffer that—Woops! I beg your pardon, Mr. Robertson, I'll get right out of your way."

"Wait, Daly," the Chief said as I made a futile attempt to keep Carse out of sight. "Fanning and I were just coming to look for you. Who's this young man?"

Well, there goes the league trophy, I thought.

"This is Carson Maim," I said. "Our new associate data handling engineer. Carse, meet Mr. Robertson, Vice President in Charge of Research."

The Chief stuck out his hand and said, "It's a pleasure to mmmmwwwooowww!"

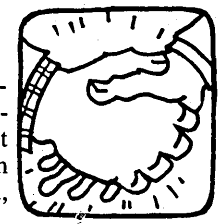
"Sorry, Mr. Robertson," I said, dragging Carse back. "He was just trying to be friendly."

"I see," said the Chief, rubbing his hand. "As I was saying, Fanning seems to be having trouble mating the buffer to the A-D Converter. One minute it's in sync and the next minute it's either ahead or behind."

I said, "Why don't we—"

"Just a minute!" the Chief said with a vengeful gleam in his eye. "Let's see what this young man has to say about it."

"Nngg," said Carson Maim.



Oh, brother, I thought.

"Well?" the Chief said.

"Nngg, with this kind of an A-D Converter, you only have a two or three microsec interval in which the Converter output register contains a legitimate word. Then it starts being replaced by the next word, a bit at a time."

I found my mouth was hanging wide open, and managed to get it closed.

"Go on," the Chief said, looking interested.

"I'd say as long as the buffer only has such a short time to grab off the word, you're going to have problems."

"And you know some trick for lengthening the interval?" the Chief challenged him.

"Sure," Carse said. "If you were using the A-D Converter EECO makes, you could plug in a second output register. Then each word transfers to the second register and stays available while the first register is filling up with the next word. It gives the buffer at least eight times as long to snatch each word, and lets you process data faster."

"EECO, huh?" the Chief said. "I know their products have a good reputation. How's the price?"

"Their A-D Converters start at under two K,"[1] Carse told him.

"Then what's the tradeoff?"

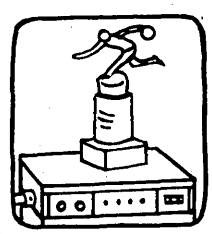
"None that I know of. In fact, EECO ADC's have some other extras. For instance, if an external trigger command is received during the conversion cycle, an EECO ADC will store it until the cycle has been completed, and then act on it. And you can get their converters with a gated display for looking at one channel at a time. In case you want to moniter the signal from a certain transducer."[2]

"That's good?"

"I leave it to Mr. Daly."

"It's good," I agreed.

"Also," Carse went on, "every EECO A-D Converter has an automatic mode. This gives you continuous digital display of the incoming analog data. It sure simplifies calibration and test procedures when you're trying to isolate problems in a system. You see, EECO has done a lot of system work. That's why they have this interface approach that eliminates a lot of problems before they start."



"That right, Daly?" the Chief asked me.

"It sure is," I said. "In fact, we've got an EECO ADC on order right now." We didn't, actually, but we would have as soon as I could arrange it, if Carse's facts checked out.

They did.

"Carse, where did you learn so much about EECO A-D Converters?" I asked him that night as we watched Marv trying to pick up a split. "Did you use to sell them?"

"No. I bowled in the same league with the EECO team. We beat them pretty bad, but I learned a lot about A-D Converters[3] in between turns. There comes your ball, Mr. Daly."

By the way, when you're in our neighborhood, drop in and see our bowling trophy. Also our EECO A-D Converter. They're both permanent.

1 He was talking about the EECO 761 ADC with resolution to 10 bits binary or 12 bits BCD. Converts at 18 KC at max. resolution. If you need higher speed, the EECO 760A ADC gives resolution to 12 bits, converts at 44 KC, is priced from $3700. That's the best deal we've heard of—how about you?

2 Also ask about EECO Multiplexers with simultaneous sample and hold for use with EECO ADC's. EE 4-5

3 So can you. Just circle this footnote and mail to Electronic Engineering Co. of Calif., Mail Station 1700, Box 58, Santa Ana, California. You'll receive **FREE** a handy table of powers of two. Table also includes quasi-software format programming of EECO 751 Computer Format Control Buffer.

CIRCLE 30 ON READER CARD

THE RCA 3488

The RCA 3488, which made its debut as RACE but now is referred to as the random access computer equipment, is an on-line storage device for the 301 and 3301 computers. The basic storage unit is a flexible magnetic card, and the average access time is 350 milliseconds, reducible by programming.

Each 16 x 4.5-inch card has a capacity of 166,400 characters; 256 cards make up a magazine (an enclosed, removable unit), and eight magazines comprise the 3488 storage unit with its own read/write head. Two units, however, can be linked in tandem to share a read/write head, and eight 16-magazine units can go on-line. "If you go through the multiplication," an RCA spokesman comments, "you'll find that you have capacity for more data than you know what to do with." Total: 5.4 billion characters—sufficient to record 25 characters of information for each person in the U.S., and (the company enjoys pointing out) equivalent to 1,000 on-line mag tape drives.

Data on the cards are recorded longitudinally on 128 tracks divided into 64 separately-addressable bands of two tracks each. Characters are recorded serially, two tracks being recorded in parallel. Each band (two tracks) of recorded information is divided into four addressable blocks of 650 (seven bit) characters each. Recording density is 700 bpi.

Computer addresses are delivered to the 3488 through a control module. The 3488 includes two full address registers for the selection and pre-selection of cards, and a magazine address which is used to return a card to its original magazine. Each address specifies one of 16 magazines to be accessed, one of 256 cards to be selected from the specified magazine, one of 64 bands on the card, and one of four blocks from the selected band. Addresses are compared and verified before reading or writing occurs.

The accessed card is power-selected in the magazine by electronically-controlled bars and grippers, and pulled along a raceway to the read/write assembly. This consists of a revolving drum, a movable read/write assembly, and a re-circulate gate (for repeated passes under the heads). Reading and writing are accomplished by eight heads which can be moved to one of 16 locations. Once positioned, the assembly is capable of reading or writing to one of four addressable bands, each containing four addressable blocks.

Head positioning for the proper band occurs simultaneously with card selection. A pre-selection technique permits a second card to be ready, timed to the leading edge of the first card, reducing effective access time. The timing mechanism clocks the block address automatically each time a block gap passes under the read/write heads. If the address requires the next head position, the heads are moved during the next card gap, permitting the data on an entire card to be transferred by one read/write instruction. Time required to transfer data on an entire card is 3.8 seconds, and the data transfer rate is 80KC. Data-checking capability include address verification, parity read-after-write, and lost bit fill-in.

The 3488 software system uses the "bucket" concept for handling and storing data, eliminating extensive chaining methods and preventing multiple overflow. It is said to simplify the programming to handle blocks of data as if they were on mag tape. A bucket consists of one or more blocks, and accommodates records of varying sizes. In order to reduce scanning and access time, a "tag" specifying a bucket address is used by the software to determine where an overflow record is stored on or can be located. Thus, no more than two seeks are necessary to retrieve any record randomly stored in the unit.

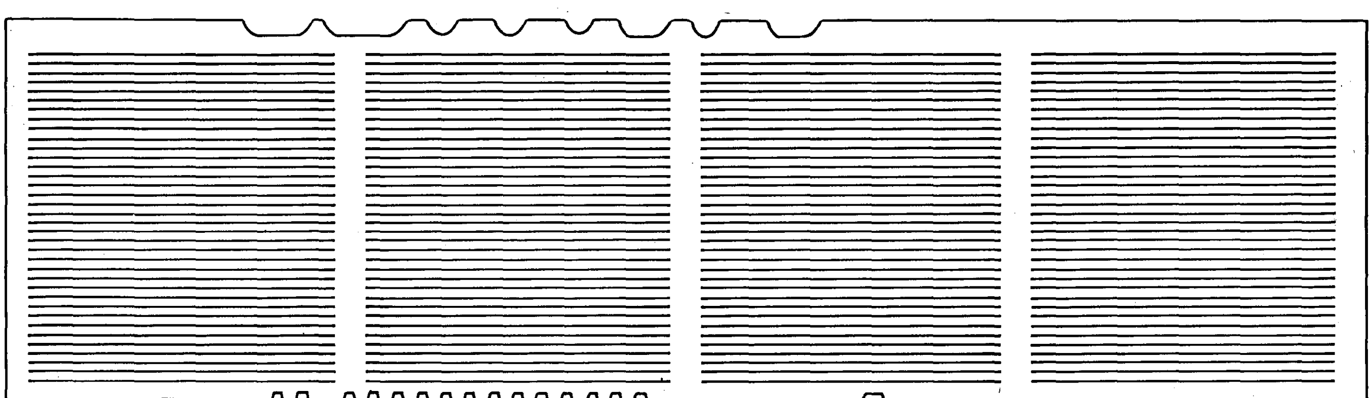
Complete software control is provided at the system, magazine, card and bucket levels, and may be supplemented by the user's own programming control.

Generally-used methods of organizing file data are applicable with the device, as are grandfather-father-son maintenance techniques. In sequential processing—one item after another order by primary number of key—use of the 3488 is facilitated by directories with "high-hit" summary information and a generalized sort system. In serial processing—by consecutively-located records or buckets—data transfer can be continuous with one instruction for bulk processing, storing programs and tables, and for statistical analysis, report generating, or duplicating master files. Full data control is effected with maps of file locations and with indices or tables for individual file entries. ■

The following 3488 lease prices are based on 176 hours usage with the 301 computer:

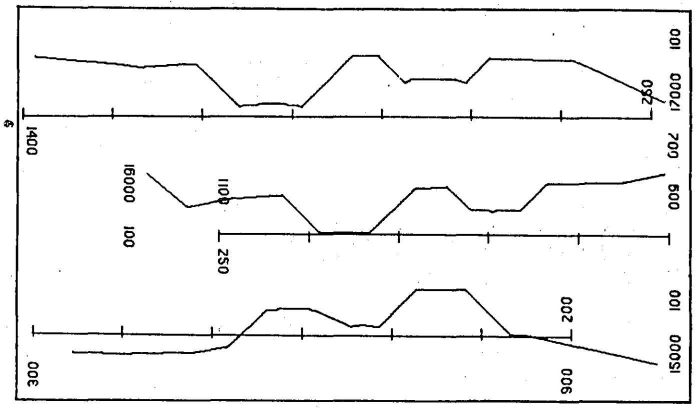
First unit, 340 million characters	$2,770.00
The add-on unit	1,385.00
The control	610.00
The magazine with 256 cards	340.00
Single replacement card	2.25

Outline drawing of magnetic card shows notched coding, data tracks, four addressable blocks.

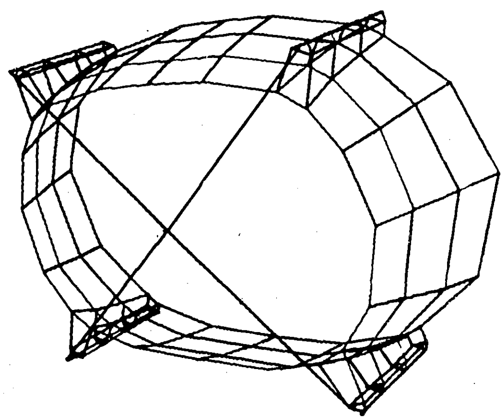


Write Tape in 5 Minutes...

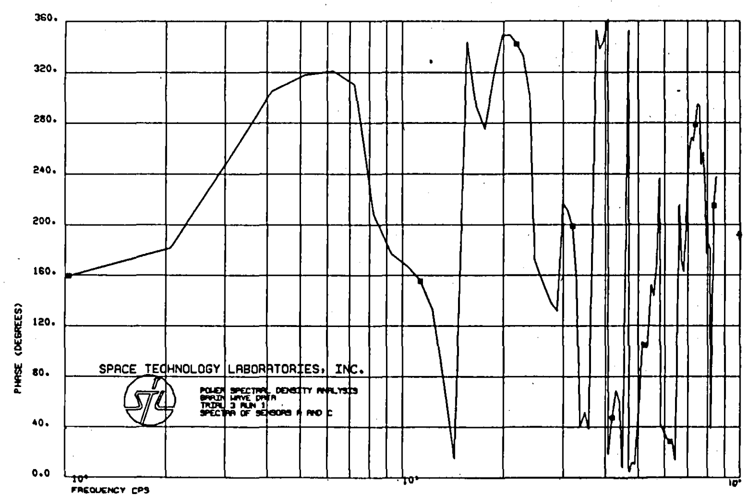
Plot 8 Hours at High Speed



HIGHWAY CROSS SECTIONS—plot 384 from one 2400-foot reel of magnetic tape.



ENGINEERING DRAWINGS—plot up to 100 from one 2400-foot reel of magnetic tape.



DATA REDUCTION GRAPHS—plot 240 from one 2400-foot reel of magnetic tape.

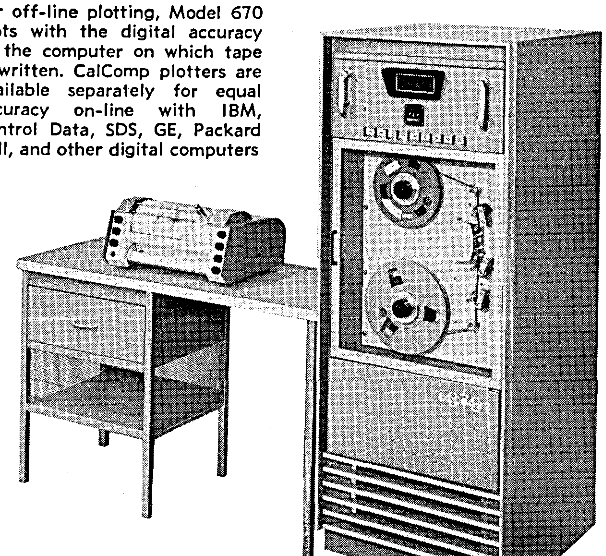
NEW CALCOMP 670 CUTS COMPUTER TIME ...INCREASES MAGNETIC TAPE OUTPUT

ACCURATE AS THE COMPANY IT KEEPS

CalComp's Model 670 Magnetic Tape Plotting System provides 8 hours of high speed plotting from a single 2400-foot reel of tape. Tape requires less than 5 minutes to write on most large, high speed digital computers. Computer sub-routines (an extensive library available) generate automatic annotation of graphs, charts, curves, drawings and maps . . . all symbols, letters, numerals or scale markings required.

- From tape written at a density of 556 bits per inch (278 plotter commands per inch) Model 670 drives the plotter continuously at full speed and imposes no restriction on record size.
- Model 670 handles tape densities up to 800 bits per inch, plots from single or multiple addresses, provides automatic search for selected addresses. Block addresses are visually displayed during plot and search modes.
- Solid state electronics in roll-out, tilt-up drawer and test panel provide ease of access and maintenance. The system operates on 50/60 cycle, 115/230 DC and occupies floor space of a secretary's desk.
- Model 670 is designed for use with all CalComp Digital Incremental Plotters, producing graphic data displays to 29½ in. wide x 120 ft. long.

For off-line plotting, Model 670 plots with the digital accuracy of the computer on which tape is written. CalComp plotters are available separately for equal accuracy on-line with IBM, Control Data, SDS, GE, Packard Bell, and other digital computers

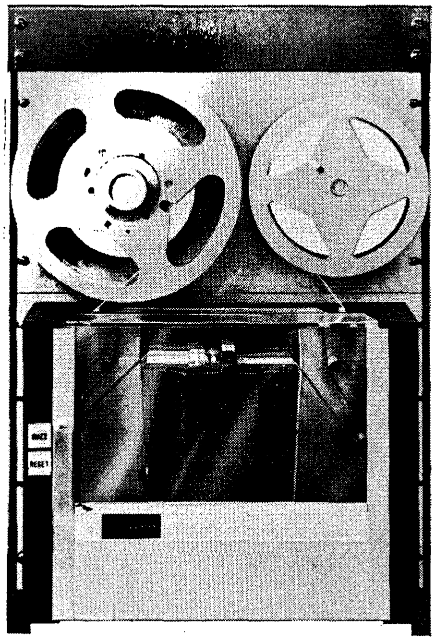


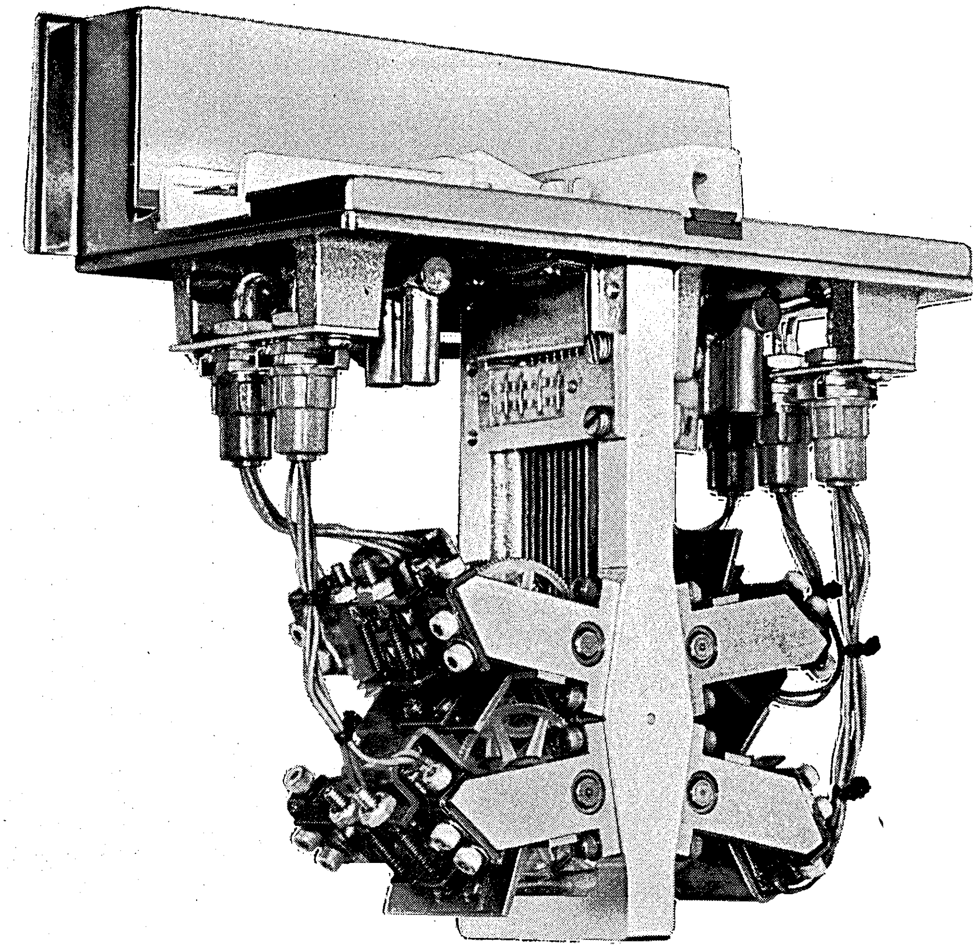
CALCOMP

Specialists in Digital Technology

CALIFORNIA COMPUTER PRODUCTS, INC.
305 Muller Avenue, Anaheim, California

CIRCLE 31 ON READER CARD





You can trust the P-150; it corrects its own mistakes

All of Tally's perforators (over 3,000) have made mistakes. We haven't built a perfect machine and neither has anyone else.

But Tally has built the next best thing—a perforator that can tell when it makes a mistake and can correct it—all at 150 characters per second.

Tally calls this new perforator the P-150. Even before we added the self-correction feature, it was a dependable machine. Now it's by far the industry's most trustworthy high speed perforator.

How You Can Use It

Computer and machine tool programming, photo composing, and many data logging applications demand accurate paper tapes. With the P-150, you don't need to design a system which includes post reading the tape for error or going through a cleanup routine which takes time and more equipment. Instead, you can design around the perforator's accuracy. Tally gives your system these lines to communicate with the perforator: (1) 8 data lines and a sprocket line; (2) a line to introduce the parity test signal; (3) two lines to check parity condition, one odd and one even; (4) a line for advancing tape; (5) a line for reversing tape.

With the P-150 your system can produce clean tape in less time, with less equipment, and at lower cost.

The Price is Right

The cost of the industry's most trustworthy perforator is: mechanism, $2,850; drive electronics, $1,400; tape handler (1,000-foot), $450.

For additional details on how Tally's P-150 perforator can solve your design problem, please write to Tally Corporation, 1310 Mercer Street, Seattle, Washington 98109.

TALLY®

CIRCLE 32 ON READER CARD

INTERNATIONAL DP CONFERENCE

The year's largest meeting for the Data Processing Management Assn., the International Data Processing Conference, will be held June 23-26 at the Jung Hotel in New Orleans. Convention theme is "Space Age Challenge." Featured will be a reconvening of the Executive Forum held at last June's DPMA meeting in Detroit. The topic, "How Can Executive Management and Data Processing Best Assist Each Other?" will be discussed by a panel.

The registration fee for association members is $75, for non-members, $85.

Some 30 other seminars are scheduled for all day Thursday and Friday morning. The keynote speaker on Wednesday will be Warren C. Hume, president of the DP Div. of IBM, and luncheon speakers on Friday will be Calvin D. Johnson and Justin Wilson.

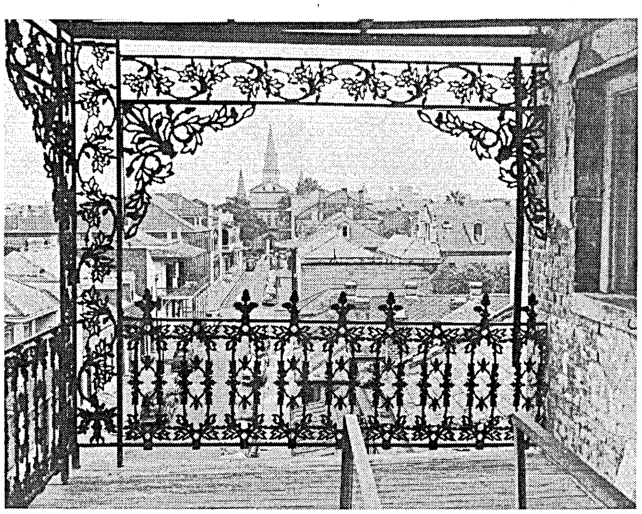
Some 60 exhibitors will participate in the Business Exposition. Exhibit areas will be open on Tuesday from 1-6 p.m., on Wednesday and Thursday from 9 a.m. to 6 p.m., and on Friday from 9 a.m. to noon. And for wives, a full ladies program includes tours of the French Quarter and the Uptown and Garden district, and fashion shows. The fee for this is $28.

In addition to seminars on specific industries and applications, the following sessions are planned:

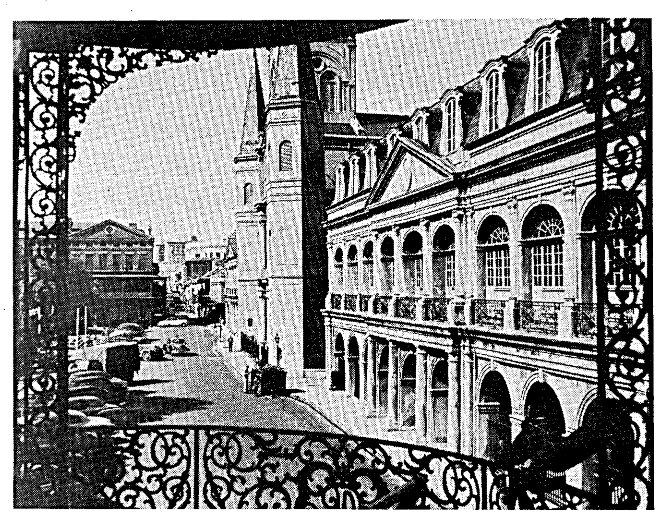
A Look Into the Future; Selection, Training & Evaluation of Personnel; Management Sciences; How to Conduct a Feasibility Study; COBOL Revisited; EDP Within the Corporate Structure; Equipment Comparison; Management Consultants Pro & Con; Centralized vs Decentralized EDP; Statement-Type vs Machine-Oriented Programming Languages; and Equipment Purchase vs Rental.

Other topics: Petroleum; What's New in Bio-Medical Research; Improved Marketing by Computer Control; Manufacturing; DP in Volume Retailing Operations; Transportation; Insurance; Government; Education; Utilities; Banking; Hospital Administration; Data Communication; Management Information Retrieval & Dissemination; Simulation Techniques, Uses & Validity; Controls & the Audit Trail; PERT & CPM; Random vs Serial Storage; and Zip Code & the Transition Stage. ■

A view of Orleans Street looking toward the rear of the famous St. Louis Cathedral.



A view through the iron grill of the lower Pontalba Apartment balcony, looking up Chartres Street.



COMPUTING IN THE SECONDARY SCHOOLS

what, why and how

by FRED GRUENBERGER

For a comprehensive discussion of the role of the computer in the high school curriculum, we should, of course, consider more than *Why* and *How*. We might go into the *What:* What should we try to teach about computers and with computers? And we might also toy with the *Who:* Who should do the teaching and to what groups of students?

But let's take these things in order; the first one, logically, is *Why*.

The question is *not* why we should teach computing. We must assume that knowledge of computing and computers is good, and that large numbers of people must be trained and educated each year from now on. The question is, do we want this training and education to begin prior to college? It is obvious that many of us believe computing belongs in the high school, but we should examine the reasons. Many subjects are competing for time, space, and money in our school systems. It can be argued that if computers come in, something else might have to leave. If this is so, there had better be powerful reasons for what we are proposing.

The first argument is common to many subjects. It goes like this: The pace of technology is speeding up, young people will emerge into a complex society, and we should accelerate their training in the lower grades. Essentially, this says: "We're in a hurry, so let's cram faster." Thus we see calculus moving from the 14th grade toward the 12th (though admittedly only for the brightest students). We also see the same argument applied to driver training courses. It can be argued that driver training is literally training for survival, and perhaps *my* survival, which touches me. The same argument, if it is valid, applies to computing, and perhaps we could make a case for computing being a more basic tool (and of wider application) than the calculus. I would prefer to restrict the discussion to "Where *should* computing be taught?" I am trying to point out by analogy that we should always question new demands on the time or money of either the school or the pupil. I believe that computing as an academic subject can stand on its own. It is not a fad; it is becoming a vital part of almost everyone's education.

For any subject, there is a threshold point, it seems to me, below which you cannot go. Thus, you can't teach algebra to five-year-olds, no matter how bright; their minds just are not mature enough to absorb the symbolism and abstraction. I would seriously question efforts to introduce the calculus to the 17-year-olds. It is easy to demonstrate the learning of the formulas and mechanics; I am questioning the learning of the content and its meaning. We must be able to demonstrate that more understanding has taken place than could be exhibited by, say, a tape recorder or a parrot. So in advocating computing as a subject for the 12th grade, I must determine where the threshold is.

Admittedly, for any given subject, the threshold is different for different people. The subject we are considering, though, is proposed for mass education; we must worry about the threshold point for all students who might arrive at the fourth year of high school mathematics. Many experiments have established that fair understanding of computing can be realized with gifted 12-year-olds (i.e., 7th or 8th graders). It seems reasonable to assume that we would be well over the threshold at the 12th grade. With computing, we can prove that understanding has developed in the child (and proof is difficult with almost every other subject). When a student can take a real computer problem and carry it through all its stages independently (namely, analysis, flow-charting, coding, debugging, testing, and production), then there is little question but that he has mastered much of the subject. We can insure that the problem he works on is, indeed, a good computer problem by having him select from a limited — but very large — set. If so, we have at least established feasibility.

The threshold concept is not new. Consider the prob-

Mr. Gruenberger is associate mathematician at the RAND Corp., Santa Monica, Calif., and the author of five books on computing. He is presently gearing up to teach, for the fifth consecutive year, a class of 12-year-olds (see Jan. '63 Datamation). He holds a master's degree in math from the Univ. of Wisconsin, where he also served as project supervisor in the Numerical Analysis Lab, later as numerical analyst for GE in Richland, Wash.

lem that a mother cat faces in giving her kittens the training they need for survival. The lessons she gives, on stalking and pouncing, and in how to go through a doorway, are basic and vital; the kittens who fail to learn stand a good chance of not becoming grown-up cats. The first thing she does (by instinct, to be sure) is wait for them to reach minimum age—the threshold for the course she is about to give. It comes at about four weeks of age, and it would be futile to teach it earlier. She then teaches by example—for instance, by going through a doorway in the time-tested manner. They then learn by doing. Usually they do it all wrong—once. They are promptly punished, by being knocked five feet or so by mommy's big paw. They get one more chance, and on the second try they get it right, and are promptly rewarded with lunch. This is known as the *Felix Domesticus* school of teaching. Cats don't seem to need any more modern methods, since this one works splendidly. I have digressed; the important point is that threshold effect.

But now I argue that the learning of computing is analogous to the learning of a foreign language. In fact, the first third of a course in computing *is* the learning of a foreign language, and moreover is conducted in that language. It is well established that foreign languages are best learned young (in the case of languages, the younger the better. In some parts of the world—China, for example—small infants learn idiomatic Chinese readily). For aged people, no amount of time and effort seems to be enough to learn a foreign tongue without an accent and with full command of the word order in a sentence.

This, to me, is the strongest argument—namely, that it's simply easier and more fun to learn this subject in high school than in college. Other arguments:

a) The high school student has less preconceptions and misinformation that will only have to be erased. He tends to bring a blanker slate to write on, in other words.

b) The college-bound student who has had some computer training in high school has a distinct advantage on his arrival on campus. It is becoming very difficult these days to find a college that is computer-free.

The day may not be far off when the roles of the entering freshman (science or engineering student) and the college admittance officer will be somewhat reversed. The student will be asking, with justification, whether he should put *this* college on his list of acceptable institutions. He will ask how much computing capacity the college has, and how much of it, per week, he can get.

c) The computer can be used as a teaching device to make the learning of other things (e.g., mathematics) easier or more attractive. There is also a prestige and reward factor to consider in having a computer course available in the senior year.

Those of us in computing claim—and believe—that the learning of computing has a beneficial effect in regard to clear, logical organization of one's thinking on new problems. We are claiming the transfer effect, and we have, to date, little or no evidence to support it. I recall (in slightly different context) the same claim made some years back for the learning of Latin—namely, that besides all its other advantages, it was good for the learning of English. I think the transfer argument is as weak for computing as it was for Latin; at least, we should demand proof that the transfer actually takes place. The best I can say right now is that I think it has made *my* thinking more orderly—but you can see what an illogical argument that becomes.

So much for *Why*. I have already tacitly implied the *Who*: The computer course should be embedded in the mathematics department. Specifically, if it is to be in the regular curriculum, I think it should be in the trigonometry course. A significant portion of a traditional trig course deals with the mechanics and dog work of computation, and the computer can pay back most of the time it takes to teach its fundamentals. We can hope, of course, that the computer will not be labelled "exclusive property of the mathematics department," but that its use be extended to other departments. As an example, all teachers have common chores of bookkeeping in connection with grades. The computer might offer some relief from such chores as a starter; eventually its power can be extended to more sophisticated uses. The foreign-language teacher, for instance, might easily be lured into trying a machine approach to word-for-word table look-up translations of simple texts. The teacher of business practice should be attracted to some of the simpler business applications of the computer, and so on.

As to the *What* of computer instruction, there are three important areas:

1) Traditional problems. We can use the computer to solve many problems that have always been included in secondary math classes. There was mentioned the solution of triangles (i.e., all the stock trigonometry problems); there is also the solution of systems of equations, and so on. Of great importance, along these lines, is the topic "What should we *not* compute?" In other words, what problems are not good computer problems? Let me cite two examples of what I regard as not-good computer problems.

> a) The calculation of 17^5, which is not only trivial, but a one-shot problem. A good computer problem demands repetition.
>
> b) The playing of tick-tack-toe. This can be great fun, and the programming for it can be challenging, but it's the wrong thing to compute. Every possible board situation in the game can be stored in any computer, and the correct move can simply be looked up. The world is jammed with good and real computer problems; why allow students to waste their time and the machine's time?

2) Advanced problems. There are many problem areas in advanced mathematics—calculus, number theory, college algebra, analytic geometry—which can be successfully tackled with a computer, if only by cut-and-try methods. For many problems, it is possible to show that the problem is not yet solved—and the student may be the one to show us—or that the known solutions are limited and he can move the boundaries of knowledge back a bit. This is the most challenging motivation I know for the good student. By making a solution feasible, the student is led to inquire into the analytic methods. Inquiring students are always good, sometimes almost as good as inquiring teachers. In my experience, the computer acts as a catalyst to produce both.

3) Entertaining problems. The whole field of computer work is somewhat akin to a large jigsaw puzzle, and it seems to attract the same kind of addicts. The world is full of puzzles that are ideal for computer attack. Everyone's experience along these lines has been the same—namely, that if you furnish a computer to bright kids, you need only stand back and let them go. High school students will not tolerate any waste of the machine's time, nor any work on trivial problems. They will demand answers to their questions. The high school teacher who finds all of this slightly frightening need not worry. He has several things in his favor:

a) He has several years to learn the subject, and has always the assurance—as with any subject—that lots of stupider people have done it before him.

b) His students will learn the subject at high speed, with or without him, and will be delighted to show him the fine points.

c) As with no other subject, he can respond to most questions with: "Why don't you try it and see?" You can't hurt a computer through its console buttons, and there are no mysterious hidden elements. Both the teacher and the student can personally check out every topic. With any sort of luck, they can find a better way to do many things.

And that brings me to the question: *How*. How do we couple our students to these wondrous machines? It is well known that the machines are frightfully expensive.* I would suggest that many schools are unaware they've been moving in the wrong direction. It isn't really necessary (however desirable it would be) to move a machine to the students; why not move the students to the machine? It's a rare school these days that isn't within 10 miles of a computer, and I have yet to hear of a computer user who wouldn't encourage an earnest group to visit his installation and use some machine time. In most such cases, the school will acquire a free teacher for lagniappe, and one who works long hours with great enthusiasm. It's not the ideal solution, but it's a fine way to operate today. Maybe tomorrow we can get the manufacturers to make an inexpensive machine for the schools. The manufacturers are understandably shy about an undemonstrated market, but, on the other hand, when the market develops (there are 11,000 high schools of 1,000 or more students in the country), they want to be there with the goods. Meanwhile, I should sound a warning against inexpensive non-computers, of which there are already too many on the market. Schools should not waste their precious dollars on paper clip gadgets that purport to be "just as good" as a computer. Everyone is well aware that adding the word "programmed" to the cover of a book doesn't automatically make it into a programmed text. Similarly, no matter how big the word "computer" appears on the nameplate or the brochures of a cheap gadget, that doesn't make it a computer, or a computer simulator, nor a device to "do the same things as the giant electronic brains" (sic). A computer is an internally programmed calculator; so far, all of them are electronic. The secondary schools shouldn't settle for less, particularly when there is probably access to one nearby for nothing.

The real computers *are* getting cheaper, and quite rapidly. Coincidentally, textbooks are appearing that are designed for high school use, with every promise of many more to come. The only real shortage right now seems to be trained teachers.

It is much too early to try to define exactly *what* about computers and computing should be taught to high school students. We need, and should take time, to explore a wide range of curricula and many diverse approaches. Here, there is a comforting fact: there is no record of an unsuccessful course in computing — at any level. This only demonstrates, it seems to me, that the real teacher, ultimately, is the computer itself—and it is superb. ■

*There is such a thing as a "free" machine, occasionally, but it may be a white elephant . . . old and unreliable. Cost of maintenance may be prohibitive. No one sets out to teach the use of an inoperative machine.

IBM reports to the industry—the new SYSTEM/360

All-purpose system solves variety of problems

IBM SYSTEM/360 can solve data processing problems in every area of business, science and industry.

You can use this one system for inventory simulation, market forecasting, linear programming, statistical analysis and other jobs.

Two types of channels help handle these jobs at faster speeds. Selector channels control high-speed I/O devices such as tapes and files. A multiplex channel serves multiple I/O devices such as printers and communications terminals. All channels are overlapped and have data rates of up to 1,200,000 characters per second or 2,400,000 digits per second.

SYSTEM/360 uses an eight-bit rather than six-bit character (plus one check bit in every character). Two 4-bit numeric characters can be "packed" in each 8-bit character. This saves storage space and increases the data transfer rate.

The new character size and the instruction set provide greater flexibility for manipulation of individual bits within a character, making it easier for the programmer to set up switches, masks and similar logic instructions for a program.

IBM SYSTEM/360 features new I/O devices

With SYSTEM/360 you can choose from the widest selection of I/O devices ever offered with one system. You select only the equipment you need. When you need more power or capacity, just add new I/O devices.

The new IBM 2400 Tape Drives (with a standard ½″ tape) utilize a 9-track, single-density, 800-bit-per-inch tape format...with data rates 22.5 K to 90 K per second. With an optional read-write head, the same tape drives work with conventional seven-track tapes.

For even faster tape throughput, the new IBM 7340 Hypertape Drive gives you rates up to 340,000 characters and 680,000 digits per second.

Other new devices: With the 2841 Storage Control Unit, every record on disks and data cell units may have an individual format.

The 2311 Disk Storage Drive gives you direct access to over 7,000,000 characters of data on each interchangeable disk pack.

The 2321 Data Cell Drive lets you put billions of characters on-line.

The new 2301 Drum Storage Unit reads or writes data at 1,200,000 characters per second with a drum latency to any record of only 8.6 milliseconds.

Using the vast assortment of I/O devices offered with SYSTEM/360, you can put together a large variety of configurations...to handle any data processing problem.

Memory ranges from 8 K to over 8 million characters

With SYSTEM/360, IBM offers virtually unlimited memory at low cost.

In this new system, the main core memory alone ranges from 8 K to 512 K. With large, high-speed memory like this, less programming is required.

Big jobs now go faster—particularly when they involve tables, rates, large matrices and subroutines.

Up to 8 million characters of bulk core storage can be added, one or two million characters at a time. Every character is directly addressable and up to eight characters can be accessed in only eight microseconds.

The programmer uses the same instruction set for any SYSTEM/360 configuration. You don't have to revise most of your programs.

This new type of core memory gives you the largest, low-cost memory ever offered.

SYSTEM/360's operating system is tailored to your needs

The operating system of SYSTEM/360 is a comprehensive package of control and processing programs which support a wide range of configurations. From this modular package, you select the elements which match your needs.

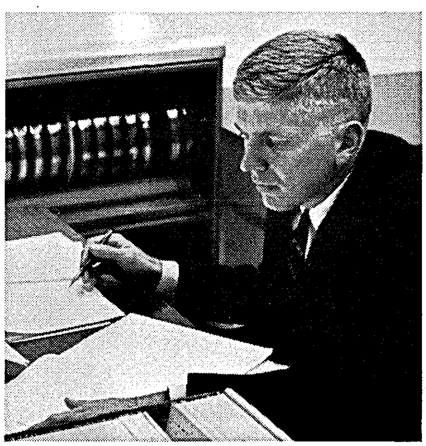
Control programs initiate job operations, load programs, assign I/O units, handle stacked job processing and job scheduling.

Control routines perform all I/O functions required by the processing program. These routines also take care of channel scheduling, control of buffer areas in main storage and standard file labeling.

The operating system includes improved compilers for programs written in FORTRAN and COBOL as well as symbolic assemblers with macro-instruction capability.

A new programming language has been developed especially for applications which require both scientific and commercial capabilities.

Problems can be stated in terms related to English, mathematics or symbolic language. The operating system automatically translates source programs into machine language and then executes the program, if desired.



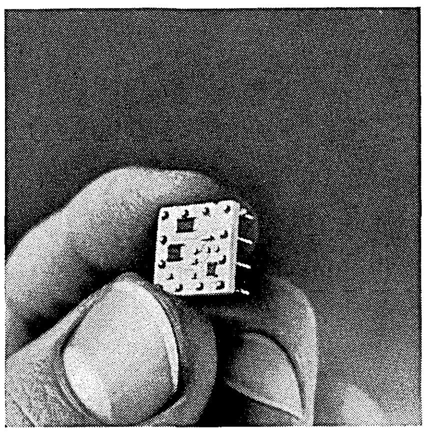
You get four new easy-to-use programming languages with SYSTEM/360—an assembler and compilers that speed both programming and processing.

New high-speed circuits used in IBM SYSTEM/360

In SYSTEM/360 we use a new "solid logic" technology, built around miniature circuits.

We manufacture tiny chip transistors, then assemble them into printed circuits only a half inch square—a fraction of the size of previous circuits. Finally, we permanently seal each circuit.

These tiny micro-circuits pack more computer logic in smaller space. And they speed operation inside the central processing unit.



New high-speed circuits contribute to the increased processing speed of SYSTEM/360.

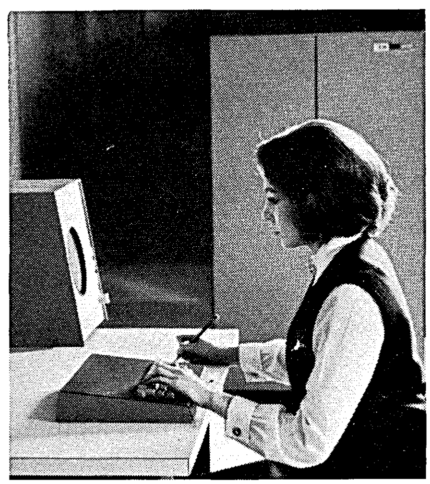
System offers complete data communication capabilities

SYSTEM/360 is designed to handle data communications and data processing operations.

It can be expanded, in stages, to take input from one to 256 communication lines. You may start with only one line and add as many as you need.

The system allows message data to arrive simultaneously with normal processing operations. Its fast, multiple interrupt plan minimizes the time needed to make a program switch, identify the interrupt and act on it. Control programs automatically relocate programs when necessary.

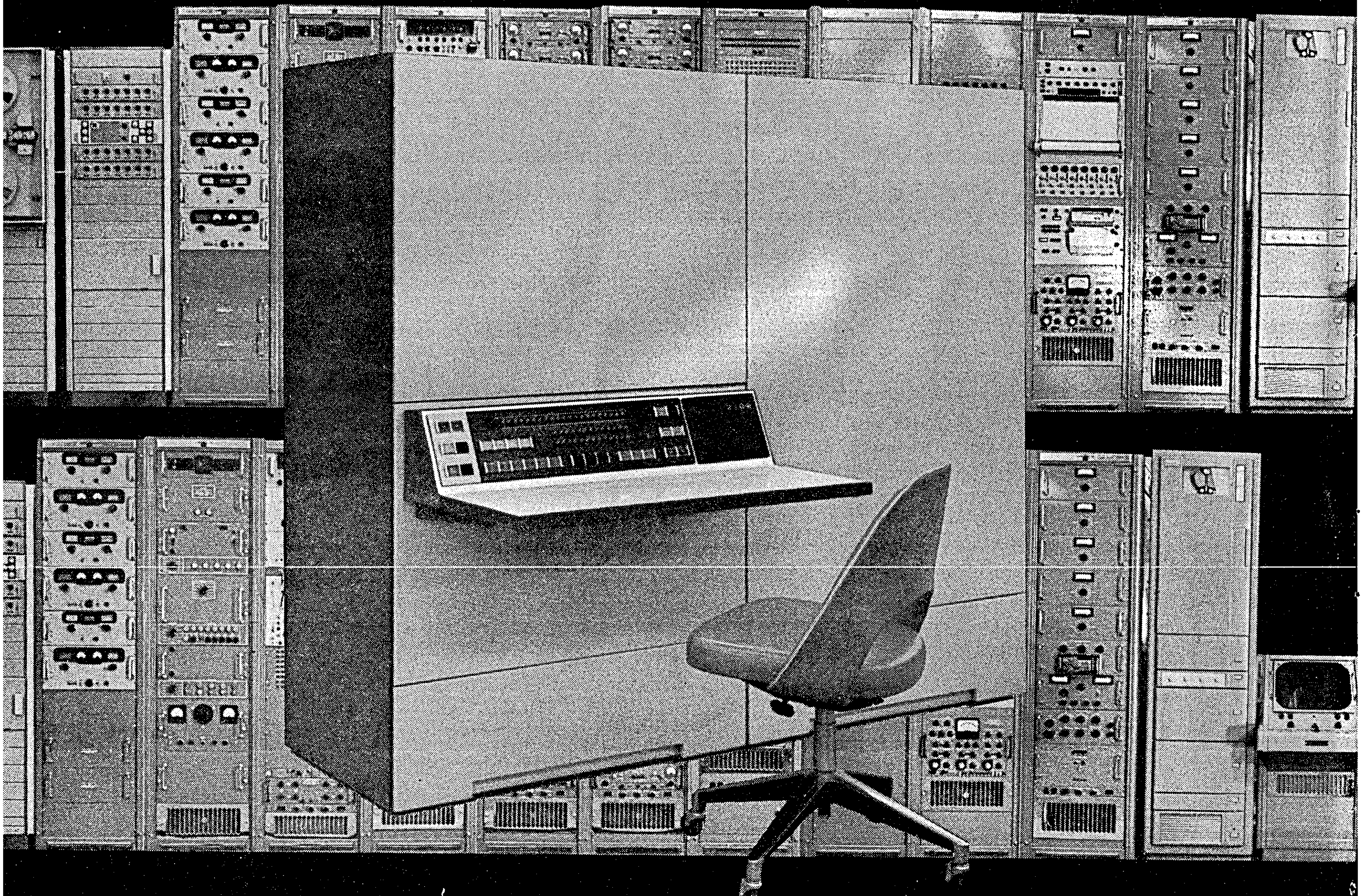
SYSTEM/360 also offers memory protection, dynamic storage allocation, direct access files of any size and speed, expandable core storage and multi-code conversion.



I/O devices like the IBM 1015 Display Unit keep you in direct contact with your central processor from any location.

IBM®
DATA PROCESSING

CIRCLE 34 ON READER CARD



NEWS BRIEFS

CYBERNATION FEATURED IN TRIPLE REVOLUTION PAPER

The tendency toward greater use of automation and cybernation should be encouraged by public policy and not resisted, according to a 27-page document, *The Triple Revolution,* sent to President Johnson and Congressional leaders. Signed by leading economists, labor leaders and journalists the paper discusses three simultaneous revolutions occurring in the U.S. which require fundamental changes in public and private policy—cybernation, weaponry, and human rights. Most discussion is with the first.

The report suggests that promises of jobs to Negroes today are a "cruel and dangerous hoax" because of their special vulnerability to cybernation. "The demand of the civil rights movement [for freedom and jobs] cannot be fulfilled within the present context of society," the statement declares. "The Negro is trying to enter a social community and a tradition of work-and-income which are in the process of vanishing even for the hitherto privileged white worker."

The paper also recommends federal and state programs for the transition period presently being experienced. Single copies are available while they last from The Ad Hoc Committee, Maurer, Fleisher, Zon & Associates, 1120 Connecticut Ave., N.W., Washington 36, D.C.

BIOMED COMPUTER CENTER GOES TO WASHINGTON UNIV.

A Computer Technology Center for Biology and Medicine, destined for MIT under a 2.8-megabuck grant from the National Institutes of Health, has been transferred to Washington Univ., St. Louis, Mo. Working with the School of Medicine and the Engineering School's department of computer sciences, the center will study advanced computers for biomedical research.

Also making the move are William N. Papian, named director of the center and associate dean, School of Engineering, and Wesley A. Clark, research professor of computer science. The two are responsible for LINC (Laboratory Instrument Computer), desk-top hardware for the biomed researcher in the lab.

AFTER 10 YEARS, SABRE IS ON THE AIR

As alert air travelers know, the American Airlines reservation system, SABRE, already is operational in some communities and should be on-line nationally this year. The 10-years-in-the-making, 30.5-megabuck system stores not only seat inventories but also passenger names and phone numbers.

By year-end, 1,000 agent sets will be on-line to an IBM 7090 system housed at Briarcliff Manor, N.Y. The configuration includes two 64K 90's, 12 disc files (total capacity: 532.8 million characters), six drums (7.3 million characters), and 12 tape drives.

Among SABRE's automatic functions: notify sales agents when special action is required, such as calling a passenger to inform him of changes; maintain and process waiting lists of passengers desiring space on fully-booked flights, as seats become available; send teletype messages requesting space on other airlines, follow up if no reply is received, and answer same requests from other airlines; and advise agents to follow up on a reservation when a passenger hasn't picked up his ticket within the time limit.

PRODUCT DEVELOPMENT LEADS PERT APPLICATION GROWTH

The fastest growing commercial use of PERT is in the management of new products, which nevertheless ranks in applications below R&D projects and construction, according to a study of 183 companies by the consulting firm of Booz, Allen & Hamilton Inc. By the end of '63, half the companies surveyed were using PERT (Program Evaluation and Review Technique) only for commercial work, and another 35% for both commercial and government projects.

The firm also notes a change in management's attitude toward the technique. In 1959, the top brass considered that 80% of PERT's value lay in its planning benefits. Now, 75% of users vote for operating benefits. Like time savings. Of those that answered this question, three-fourths believed that 6-20% of total project time was saved by using the technique.

Cost? Some 47% of respondents regarded the costs as minimal, 45% as moderate, and 8% as high. The Air Force indicates that PERT runs 0.5% of the total cost of an R&D project and 0.1% of other major programs.

Almost 100 companies reported that they processed PERT by hand; almost 80% of the manually-processed projects had 200 or fewer activities. Difficulties encountered with using the technique: 64 mentioned securing realistic time and cost estimates; 57 mentioned obtaining operating personnel's acceptance and use; and 41 noted personnel training problems.

SAVINGS, QUALITY CONTROL CITED BY TWO USERS

Westinghouse, whose computer-controlled teletype network is headquartered in Pittsburgh, reports it spent 16.2 megabucks in 1963 for its company-wide computing effort. The firm says computers are producing a net savings of 30% of its outlays. The Tele-Computer Center reportedly pays for itself in direct savings. Effects of communications net: closing of four field warehouses and eight-megabuck reduction of stocks in all warehouses.

At three Chrysler Corp. plants, a computer-based quality control system is on the air and is scheduled for its remaining assembly lines nationwide. Central hardware is an IBM 1710 computer, linked to data collection units at assembly points. The system keeps tab on defective work and parts reported, and reminds those farther down the line of corrective action not taken. Byproduct: report for management and parts manufacturers.

17-YEAR-OLD OKLAHOMAN WINS $150 SOFTWARE PRIZE

A 17-year-old Altus, Okla., student has won the "Most Original Computer

NEWS BRIEFS . . .

Program" contest sponsored by the Assn. for Educational Data Systems for junior and senior high school students. The $150 prize went to Steve Doughty who, with his teacher Earl Newberry, was flown to Santa Barbara, Calif., for the presentation at the second annual AEDS convention. There were 55 entries in what is slated to become an annual contest.

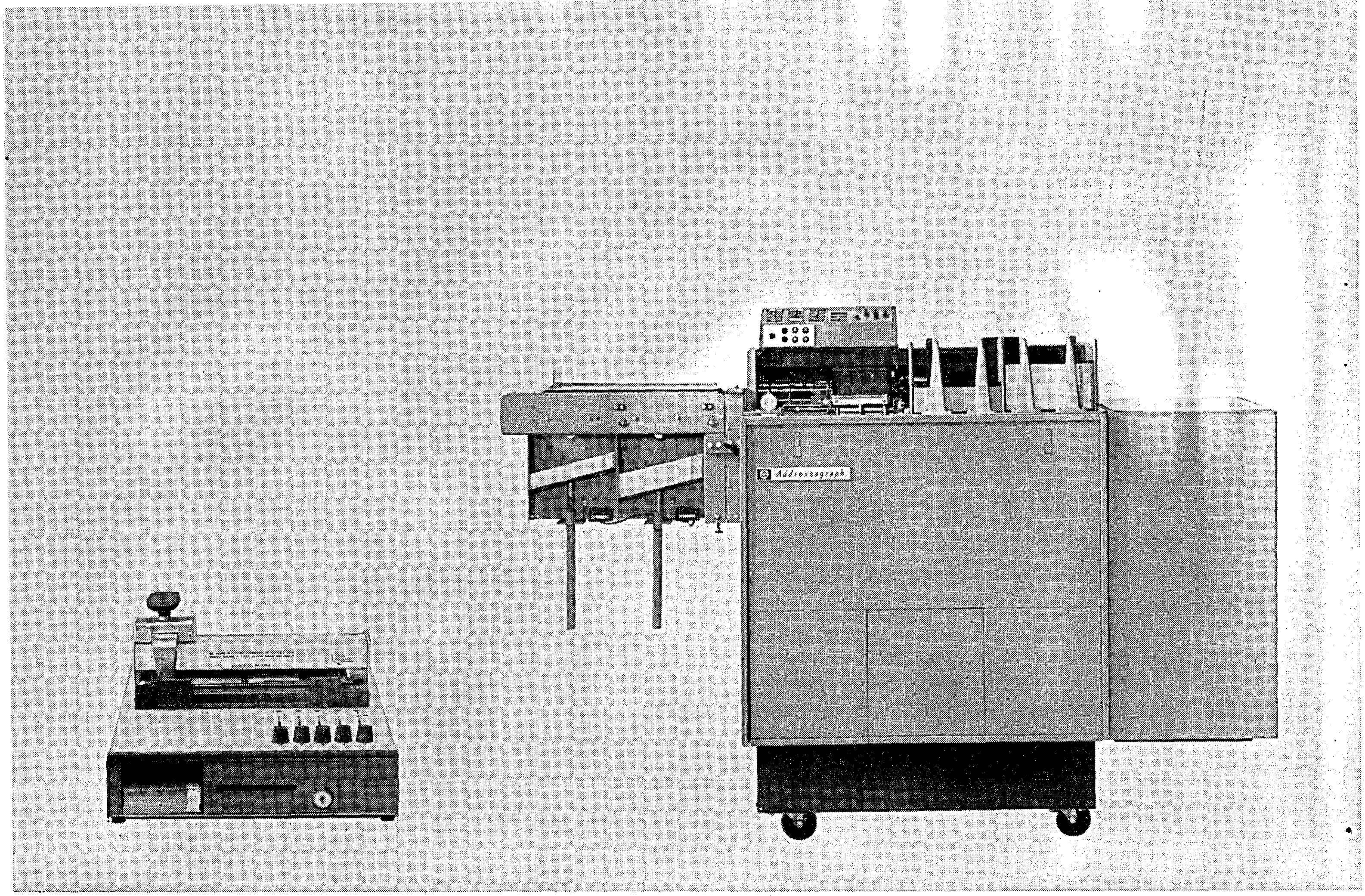
The software, written in FORTRAN and run on a 1620, enables irrigation workers to determine the number of acre-feet of water being delivered hourly through a six- and eight-inch pipe by observing the water level in a pilot tube. The youth is attending a Saturday course in computer math at the Univ. of Oklahoma.

Six second-place prizes of $50 were awarded, as well as 10 awards of a year's subscription to *Datamation*. The prizes and air fare were paid by Control Data Corp. Judges were Fred Gruenberger, RAND Corp., and Dr. Richard Andree, Univ. of Oklahoma.

● RCA has received a "multimillion-dollar" contract from Western Union for additional work to expand the DOD's Autodin system. Built by WU in 1963, the worldwide military communications network presently consists of two RCA 601's, and the new work is scheduled for completion in mid-66.

● About 10% of Hughes Dynamics personnel has survived the reorganization which the firm calls a shift of emphasis from software to hardware and which includes the sale of AIS (see below). H-D has been exercising its 10-year option to purchase the majority stock of Dashew Business Machines, retains the large credit reporting service Consumer Credit Clearance, and is trying to figure what to do with numerous other holdings.

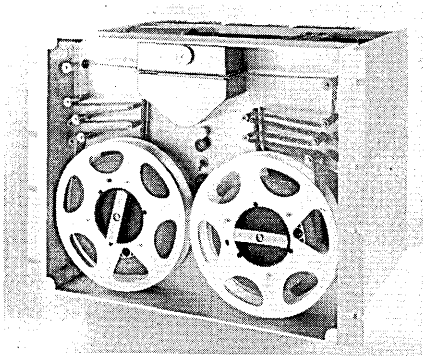
● The old Advanced Information Systems in Los Angeles, acquired by Hughes Dynamics a year ago, has been purchased from the latter by Informatics Inc., Culver City, Calif. No purchase price has been announced. Switching allegiance is a group of nine headed by John Postley, who has been named director of Advanced Information Systems, and also including Arnold Anex and Dwight Buetell. The former head of AIS, Dr. Robert Hayes has joined UCLA's Library School. The purchase also includes a contract with the Metropolitan Data Center to develop a data bank of land use information for urban renewal projects and other metropolitan studies.



NEW PRODUCTS

paper tape reader

The 500R is photoelectric, bidirectional, and has proportional reel servos which keep the tape in contact with the capstan and read head, even in the wind mode. Read speeds are field



adjustable to 500 cps, in wind/search modes to 1,000 cps. PHOTOCIRCUITS CORP., TAPE READER DIV., Glen Cove, N.Y. For information:

CIRCLE 200 ON READER CARD

terminal buffers

The Local and Remote Universal Buffers each handle up to 50 MIMO terminals, and are linked to a computer I/0 channel and telephone channel, respectively. Both handle message assembly and disassembly, code and format conversion, line identification, I/0 timing, and communication line monitoring. DATA TRENDS INC., 1259 Rt. 46, Parsippany, N.J. For information:

CIRCLE 201 ON READER CARD

gp family

The 1050 family of small and medium-scale computers consists of main frames III and IV which, with assorted modules, become the 1050 Card series, Tape series, Mass Storage series, Satellite series, Real-Time series, and the 1004 On-Line series. Access time with Mod. III is 4.5 usec per character, and two usec with Mod. IV. Time to add five decimal digits to five digts is 117 usec with the III, 35 usec for the IV. The III has 4-32K characters of core, the IV has 8-64K characters. The family is software-compatible. UNIVAC, DIV. OF SPERRY RAND Corp., Sperry Rand Bldg., New York, N.Y. For information:

CIRCLE 202 ON READER CARD

cores & stacks

Nondestruct read-only rope memories in a variety of sizes are available. Cycle times are from five to 15 usec. Other planes and stacks have 30 or 50-mil cores, and are for memories from two to seven usec. BURROUGHS CORP., ELECTRONIC COMPONENTS DIV., Plainfield, N.J. For information:

CIRCLE 203 ON READER CARD

tape cleaners

Models E-4 and E-2 are economy and automatic units, respectively, which use carbide steel blades to shave away error-causing dirt particles from mag tape. CYBETRONICS INC., 132 Calvary St., Waltham, Mass. For information:

CIRCLE 204 ON READER CARD

monrobot software

Library additions for the Monrobot XI include a symbolic assembly system, as well as programs for such users as engineers, designers, surveyors, statisticians, and teachers. These are for traverse closure, multiple regression, matrix inversion, least squares approximation, etc. MONROE CALCULATING MACHINE CO., 555 Mitchell St., Orange, N.J. For information:

CIRCLE 205 ON READER CARD

check printer

The 1445 prints both the E13B font and conventional type. Linked to a 1440, 1401 or 1460 computer, it permits insurance companies to produce preauthorized checks with other processing jobs. Standard speed with 56-character set plus MICR is 190 lpm, with 42 characters 240 lpm. IBM DATA PROCESSING DIV., 112 E. Post Rd., White Plains, N.Y. For information:

CIRCLE 206 ON READER CARD

If your company is growing fast but you've been timid about computers...

NEW LITERATURE

HIGH SPEED LIGHT PEN: Bulletin describes type 370 designed for use with cathode ray tube displays with rapid plotting rates. DIGITAL EQUIPMENT CORP., Maynard, Mass. For copy:
CIRCLE 130 ON READER CARD

MEMORY DRUM: Description, special capabilities and system specifications for the SD-1000 are included in brochure. LITTON INDUSTRIES, 5500 Canoga Ave., Woodland Hills, Calif. For copy:
CIRCLE 131 ON READER CARD

DATA TRANSMISSION: The 27A, specifically designed for use on HF radio, is the subject of this brochure. Included are performance figures, circuit description and block diagrams. LENKURT ELECTRIC CO., INC., 1105 County Road, San Carlos, Calif. For copy:
CIRCLE 132 ON READER CARD

A-D CONVERTER & MULTIPLEXER: 12-page catalog gives technical discussion of features of units and the 8000 series. Block diagrams, model charts and complete specifications are given. TEXAS INSTRUMENTS, INC., P. O. Box 66027, Houston, Tex. For copy:
CIRCLE 133 ON READER CARD

DP AND COMPUTER AUXILIARY: Catalog 15 contains 60 illustrated pages of data processing auxiliary equipment. Catalog 16 covers computer room equipment made by this company. TAB PRODUCTS CO., 550 Montgomery St., San Francisco 11, Calif. For copy:
CIRCLE 134 ON READER CARD

INFORMATION HANDLING: 32-page brochure details the company's lines of analog acquisition systems, PCM acquisition systems, digital dp systems, special purpose data systems, digital command systems and cards and modules. RADIATION, INC., Melbourne, Fla. For copy:
CIRCLE 135 ON READER CARD

SPECIALITY CARDS: Brochure describes company's complete line of speciality cards. MIDWEST TAB CARD PRODUCERS, INC., 2411 West Howard St., Evanston, Ill. For copy:
CIRCLE 136 ON READER CARD

MAG TAPE CENTERS: Brochure outlines company's mag tape centers and includes descriptions of equipment used and services provided. GENERAL KINETICS INC., 2611 Shirlington Rd., Arlington, Va. For copy:
CIRCLE 137 ON READER CARD

PDP-6: 16-page booklet includes system description, type 166 arithmetic processor, i/o equipment, instruction times and illustrations. DIGITAL EQUIPMENT CORP., 146 Main St., Maynard, Mass. For copy:
CIRCLE 138 ON READER CARD

PRC offers new JOVIAL compiler

...with cost and delivery advantages

Here's how PRC can provide a JOVIAL compiler that costs a lot less than you're used to paying—and can be delivered a lot quicker. Digitek Corporation has recently developed revolutionary techniques of compiler construction that produce FORTRAN at ½ the cost and ½ the delivery time of existing compilers. As a co-operative venture with Digitek, PRC can now apply these same techniques to JOVIAL to accomplish comparable savings in cost and time.

PRC's broad experience in command and control plus Digitek's breakthrough in compilers mean the latest state-of-the-art approach to your JOVIAL compiler.

The Digitek compiler method has provided FORTRAN for Daystrom, Hughes Aircraft Company, Honeywell, and Scientific Data Systems. The compilers can be implemented on all 8K and most 4K computers.

TYPICAL EXAMPLE: PRC would apply to your JOVIAL compiler the same techniques that resulted in a FORTRAN II for Scientific Data Systems 920 which was delivered on time at a fixed price.

Features included: All statements—accepts the full language · Several times faster than comparable compilers · Efficient machine code · Optional generation of program diagnostics.

Call or write now for complete information on how the combined capabilities of PRC and Digitek can save you time and money on *your* JOVIAL compiler. Planning Research Corporation, 1333 Westwood Boulevard, Los Angeles 24, California.

PLANNING RESEARCH CORPORATION

LOS ANGELES / WASHINGTON / NEW YORK / NORFOLK / SAN DIEGO / HONOLULU

Providing assistance to Government and industry in systems engineering, economic research, and information systems—for more than a decade.

CIRCLE 43 ON READER CARD

EXPERIENCED PROGRAMMERS

room for achievement at IBM®

Opportunities:
New concepts in programming languages—exploratory research and development of machine-independent languages for scientific and commercial applications. R & D of ways of easily defining a number of varying special application areas: investigation of the nature of programming languages, their specifications, form and meaning, their essential syntax and semantics; comparative analysis and evaluation of language features; development and documentation of language standards.

Advanced programming techniques—research and development in such areas as: new compiler techniques, operating systems, symbolic debugging of source language programs, non-numeric algebraic manipulations in computers, information retrieval systems, and time-sharing systems.

Business-oriented programming—advanced development of sorting techniques, merging, report generators, and file-maintenance programs.

Qualifications: a bachelor's or advanced degree—at least one year's advanced programming experience.

Additional information: the art of programming at IBM is exceptionally challenging. Liberal company-paid benefits include relocation expenses. These opportunities are located mainly in Poughkeepsie, a suburban environment about 70 miles from New York City. Other programming facilities are located in White Plains, N. Y., New York City and Boston, Mass. IBM is an Equal Opportunity Employer.

If interested, please write, outlining your experience and qualifications, to: **James D. Baker, Dept. 701S IBM Corp., P. O. Box 390, Poughkeepsie, N. Y. 12602.**

COMPUTER PERSONNEL

Following are some of the positions for which we are currently conducting recruiting and search assignments:

■ Senior EDP Consultant	$18,000
■ Lead Systems Analyst	$20,000
■ Lead Systems Analyst	$20,000
■ Lead Systems Analyst	$10,800
■ Senior Systems Analyst	$10,000
■ Senior Systems Analyst	$10,000
■ Senior Systems Analyst	$10,000
■ Lead O/R Analyst	$18,000
Senior O/R Analyst	$20,000
■ O/R Analyst	$12,000
■ Senior Programmer	$ 8,500
■ Programmer	$ 8,200
■ Programmer	$ 8,000

All fees are assumed by our client organizations. All negotiations are handled in complete confidence. Registrations of computer professionals not actively seeking a position change but sincerely interested in considering opportunities for which they are qualified are cordially invited.

Computer Personnel Consultants, Inc. specializes exclusively in the recruitment of and searches for professional computer personnel; computer personnel appraisal; and, computer personnel administration consulting.

CPC is composed of professional staff members with extensive computer experience with national management consulting or CPA firms, with computer manufacturers or with computer users.

CPC is the *only* personnel firm that has engaged in field research of computer personnel characteristics. Call, write or visit us to discuss these and other opportunities.

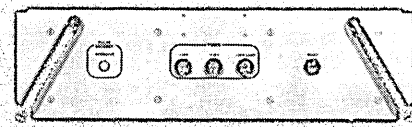
T.M.

CPC

COMPUTER PERSONNEL CONSULTANTS, INC.

135 South LaSalle Street—Suite 1460
Chicago, Illinois 60603—Phone 312—236-0900

CIRCLE 76 ON READER CARD



Motorola's TP3000 non-impact electronic teleprinter has been selected for more than 500 critical military and scientific readouts. Reasons: *RELIABLE*... offers higher MTBF predictions than any comparable equipment. *FAST*... up to 3000 WPM or 2400 bits per second with standard 8½" or 72-column page width. *QUIET*... inaudible under normal room environment. *VERSATILE* ...translator can selectively drive 30 or more remote printers. *FLEXIBLE*... accepts up to 8 level codes, generates up to 64 characters, symbols or figures. Typical applications: **NASA**—at McDonnell Aircraft Corp. as AGE monitor and readout on Gemini program, and at JPL for space flight data systems. **ARMY**—at Ft. Huachuca as readout for BASIPAC and MOBIDIC. **NAVY**—as readout for an experimental missile control system aboard the Navy electronics test ship, USS Norton Sound. **AIR FORCE**—as readout for numerous Air Force "L" systems, assigned official USAF nomenclature—Electrographic Teleprinter TT-352 and Pulse Decoder KY-417. Please call or write for technical information package that includes data sheets, printing samples and teleprinter rate calculator.

MOTOROLA TP3000 HIGH-SPEED TELEPRINTERS SERVE

ARMY

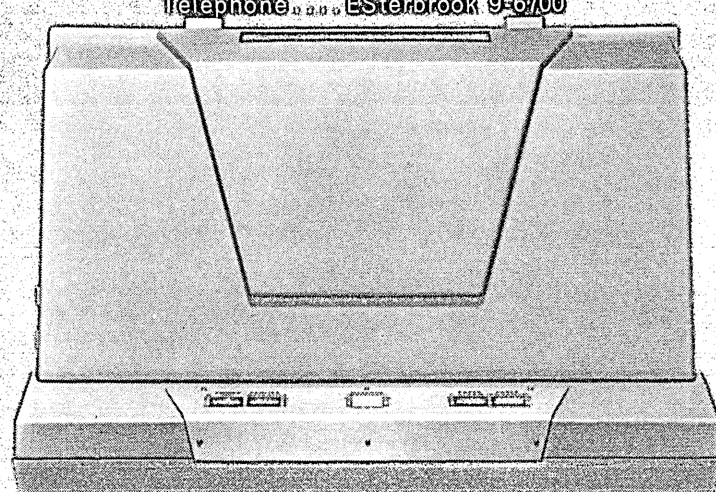
AT FORT HUACHUCA ELECTRONIC PROVING GROUND

The U.S. Army Electronic Proving Ground at Fort Huachuca, Arizona, is using the TP3000 teleprinter system for high-speed, hard copy readout from the BASIPAC and MOBIDIC computer systems. The teleprinters are being operated both directly out of the computers and remotely through the Army's high-speed digital data terminals.

CIRCLE 45 ON READER CARD

Military Electronics Division MOTOROLA

CHICAGO CENTER / 1450 North Cicero Avenue / Chicago 51, Illinois
Telephone..... ESterbrook 9-6700



Continued from page 21 . . .

a Senate pigeonhole, awaiting the end of the Civil Rights filibuster and the completion of a comprehensive Gov't. adp study by the BOB's Clewlow Committee.

ENTER THE 17-YEAR COMPUTER

Burning Industry question: What has Univac got, now that after 17 long years it's got a patent on ENIAC, first electronic computer and reknowned brain child of J. Presper Eckert and Dr. John Mauchly.

The patent — No. 3,120,606 — weighs one pound, seven ounces, makes 148 claims. Patent office fees paid for the grant amounted to $158, though Univac undoubtedly ran through at least several barrels of money in legal fees and related costs. From 1947 to February 4 of this year, the patent survived eleven major "interferences" which serve to make the patent that much more iron-clad.

The end product is a document which requires a platoon of Philadelphia lawyers and electronic engineers to comprehend. According to the patent office, the patent is "quite broad in scope" — suggesting that it establishes Univac's legal and proprietary title to basic mechanical and electronic concepts found in nearly all electronic computers.

Univac is closemouthed on the patent and its possible implications for licensing or royalty arrangements with other manufacturers. "That's usually the reason for a patent being secured in the first place," commented a patent office examiner who worked on the claim. "I don't know of any reason why Univac wouldn't be considering such action." The possibility is real enough to send most computer makers scurrying back to their law books.

NEW CASH FOR PROJECT MAC

Project MAC is due for some fresh funding. MAC, a probe into new ways to utilize computers underway at MIT, will exhaust its initial budget – believed to be about $2 million — in 15 months (against a postulated 18). The project's angel, DOD's Advanced Research Projects Agency, isn't quibbling with MIT over this profligacy, however, since it's well pleased with research results. And provision was made for accelerated expenditure in the initial contract. The new arrangement, to begin in August, is now being negotiated. It's expected the new contract will run for two years and contain a nice raise for MIT, to perhaps $3 million.

AUTODIN, AIR TRAFFIC CONTROL DECISIONS DUE

The word on who gets the next big AUTODIN plum, providing computers for its projected ten overseas message switching centers, is expected early in July. RCA recently made a clean sweep of the computer end of AUTODIN domestic updating.

And noises are being heard that the GSA's private wire system, a pint-sized AUTODIN system contemplated for non-military agencies, is being merged, as in swallow, with its big brother...A decision is reported imminent by the Federal Aviation Authority on its choice for a prototype computer(s) for its Atlantic City test facility, a major step on the way to its proposed super-duper air traffic control system. Could be that the successful bidder will have his foot in the door for the follow-up business -- an assignment that may carry a price tag of as much as $40 million for air terminal computers.

COMPUTER SOFTWARE SPECIALISTS

YOU SHOULD LOOK AT CONTROL DATA... AS A CAREER... WHY?

***BECAUSE** of the worldwide acceptance of CONTROL DATA'S technological accomplishments in general and special purpose computer systems, and . . .*
***BECAUSE** of its outstanding advantages . . . nationwide locations, creative work climate, opportunities for recognition and advancement.*

If you have medium or large-scale computer experience and a B.S. degree, please examine the following opportunities which exist at all experience levels and find out "firsthand" how the worldwide acceptance of CONTROL DATA'S product family has created once-in-a-career opportunities.

SYSTEMS EVALUATION: Participate in the development of quality assurance techniques for general purpose programming systems. These positions require a good understanding of system programming techniques and creative imagination. Positions located in Palo Alto and Minneapolis.

SYSTEMS INSTALLATION: Represent CONTROL DATA technically at various nationwide customer sites. Responsibilities will include orientation, training, programmer consultation and software systems installation for CONTROL DATA® 3600, 3400, 3200 and 1604 computer customers. Positions located in Palo Alto and Minneapolis.

ADVANCED SYSTEMS: Participate in the development of experimental programming systems in such areas as general purpose translators, multi-processing systems and new, high-level programming languages. Positions located in Palo Alto.

PROGRAMMING SYSTEMS: Systems programmers required with a minimum of two years' experience to assist in the development of parallel processing and multiple console access operating systems. Opportunity to assist in machine specifications in development of combined hardware-software systems. Continued openings in scientific and business compiler development aré also available. Positions located in Los Angeles, Palo Alto and Minneapolis.

SOFTWARE DOCUMENTATION: Assist in development of reference manuals, teaching aids, sales aids and other forms of documentation for programming systems. Positions located in Los Angeles, Palo Alto and Minneapolis.

SALES SUPPORT ANALYSTS: Consult with CONTROL DATA'S hardware customers to analyze their problems for computer applications in both pre- and post-sale situations. Experience required in scientific or business data processing programming for medium or large-scale computers. Scientific experience preferred. Positions located at all nationwide CONTROL DATA sales offices.

COMPUTER SALES ENGINEERS: Digital computer experience in sales engineering and/or applications programming to sell CONTROL DATA computers and related industrial product lines. Positions located at all nationwide CONTROL DATA sales offices.

PROGRAMMING INSTRUCTORS: Teach beginning and advanced programming to both CONTROL DATA employees and customer personnel. Must be experienced in scientific or business data processing programming. Scientific experience preferred. Positions located in Minneapolis, Los Angeles, Washington, D.C. and Palo Alto.

DIAGNOSTIC PROGRAMMERS: Develop and run diagnostic maintenance routines for 1) multi-computer systems or 2) in conjunction with regular CONTROL DATA computer checkout and field installation work. Positions located in Minneapolis.

PROGRAMMER ANALYSTS: Analyze Data Center customer problems for computer applications. In addition, you will be involved in sales support work and the preparation of programming proposals. Positions located in Los Angeles, Palo Alto, Washington, D.C., Minneapolis and Long Island, New York.

TECHNICAL APPLICATIONS: We need experienced programmers with advanced degree in math or physics for work in long-range market support. This area includes nuclear applications, APT codes for computer controlled machine applications and general mathematical applications using linear programming. Positions located in Minneapolis.

DATA CENTER SALESMEN: Data processing sales experience required. Will sell Data Center computer time to customers. Must have thorough knowledge of computer applications. Positions located in Los Angeles, Palo Alto, Washington, D.C., Minneapolis and Long Island, New York.

REAL-TIME PROGRAMMERS: Experience in real-time industrial computer control systems development and programming in the power, nuclear, petrochemical, oil and gas production and metals industries. Positions located in Minneapolis.

TO ASSURE PROMPT REVIEW OF YOUR QUALIFICATIONS AND INTERESTS PLEASE SEND RESUME TO ONE OF THE FOLLOWING AREA STAFFING REPRESENTATIVES.

PALO ALTO: B. CRIPE, 3330 HILLVIEW, PALO ALTO, CALIF. ■ LOS ANGELES: P. WEBER, 5630 ARBOR VITAE, LOS ANGELES, CALIF. ■ EASTERN REGION U.S.: J. FETTIG, 11428 ROCKVILLE PIKE, ROCKVILLE, MD. ■ MINNEAPOLIS AND MIDWEST: T. OLDHAM, 8100 34TH AVE. S., MINNEAPOLIS, MINN.

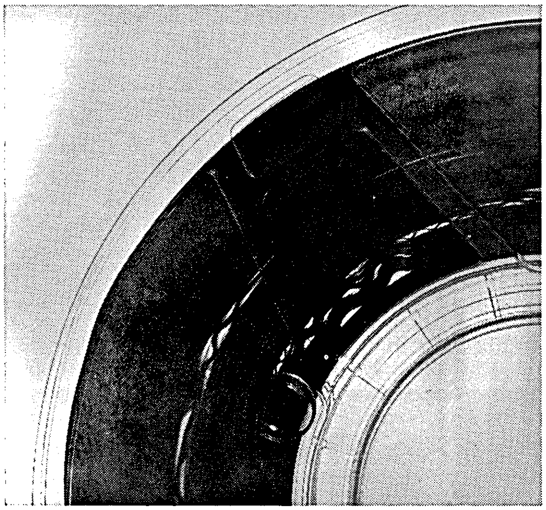
NOTE: When sending resumes, be sure to mention position or positions of interest as listed above.

CONTROL DATA
CORPORATION
An Equal Opportunity Employer

ADVERTISERS' INDEX

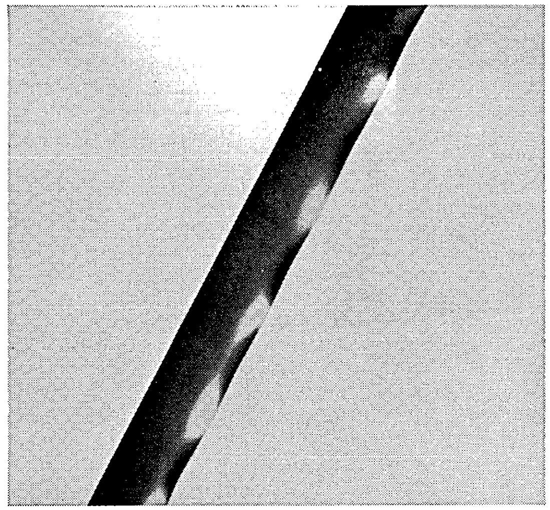
4 COMPLAINTS ABOUT COMPUTER TAPE

(And how Memorex solves them!)



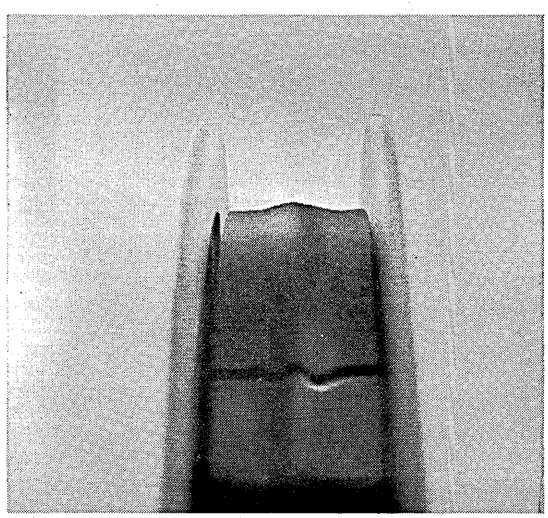
Complaint. Cinching during shipping, use or handling results when reel is wound under improper tension or exposed to temperature extremes.

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



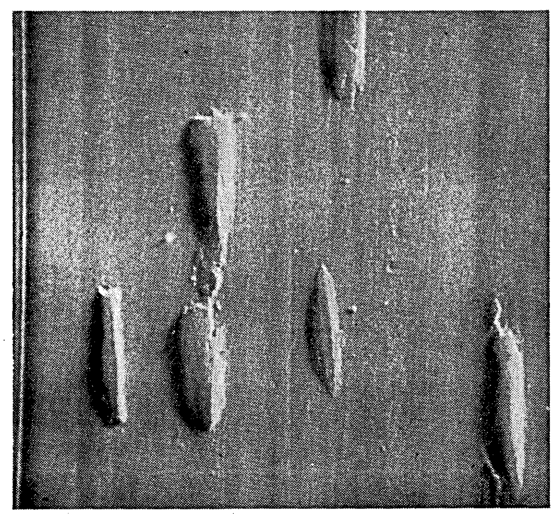
Complaint. Wavy edge caused by improper slitting.

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.

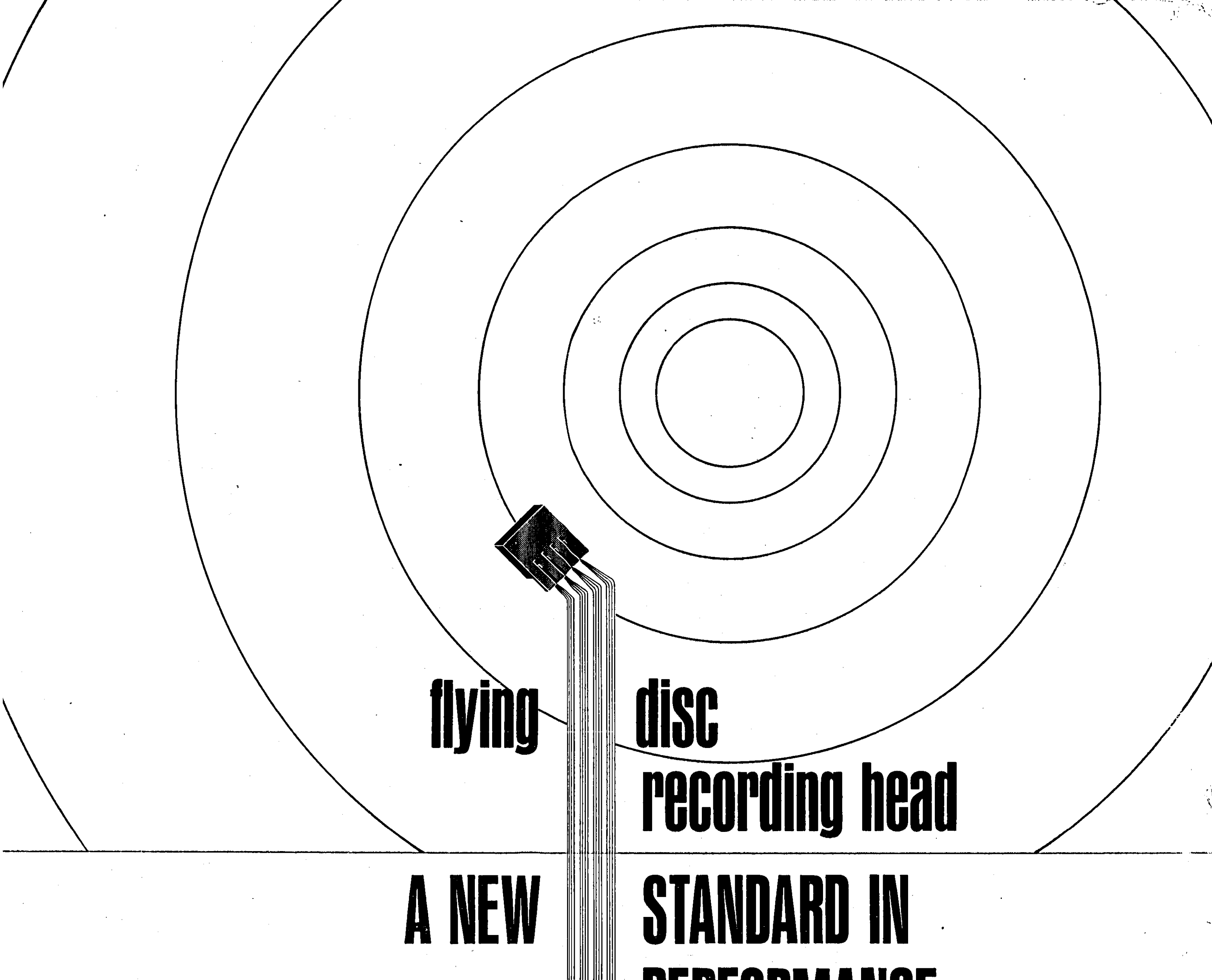
Memorex certification means what it says: Memorex computer tape is error-free. Extra care, extra steps and scrupulous attention to every detail make it that way. We know the importance to you of having a tape you can depend on.

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



flying disc recording head

A NEW STANDARD IN PERFORMANCE

An optimized floating head design employing recent breakthroughs in ferrite technology is now offered as a finished product. Utilizing high density ferrite throughout, and fused by molten glass, the assembly is controlled to far more precise tolerances than those yielded by previous methods—and at substantial cost reductions!

Higher bit densities and **broader operating margins** are a direct benefit of this control. Stability over wide temperature ranges and long term electrical and mechanical stability are unprecedented.

This flying disc head design is offered in configurations of one to twelve tracks, 10 mils wide—all with the same mechanical and aerodynamic properties. Shown below are actual test results obtained in our laboratory on a plated disc.

1000 Bits per Inch Packing Density
40 MV Output at 1.6 MC
150 Micro Inches Flying Altitude

If you are concerned with a new disc file design, look into the use of this new readily available head. Our application engineers will gladly work with you to meet your requirements.

FERROXCUBE
CORPORATION OF AMERICA
SAUGERTIES, NEW YORK

CIRCLE 3 ON READER CARD