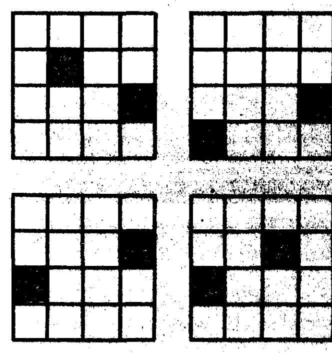
DATAMATION 62
October

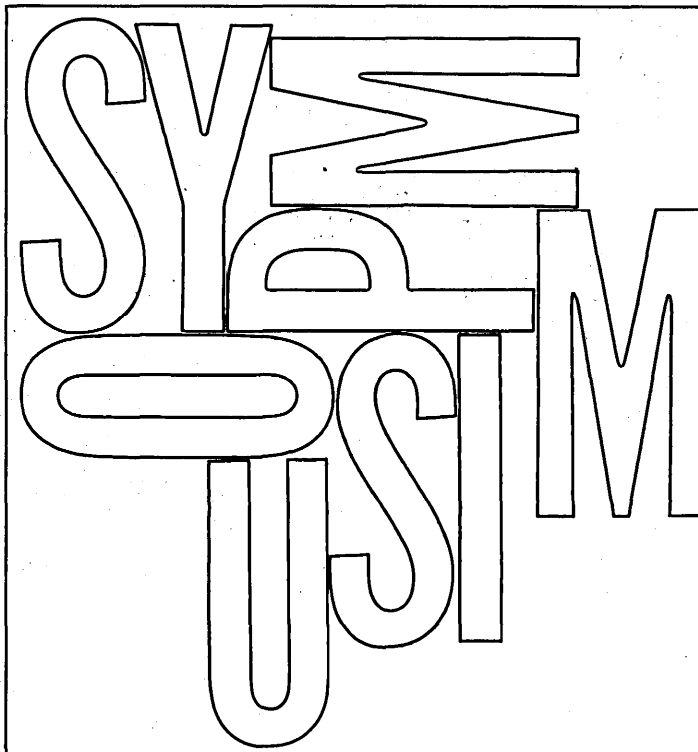


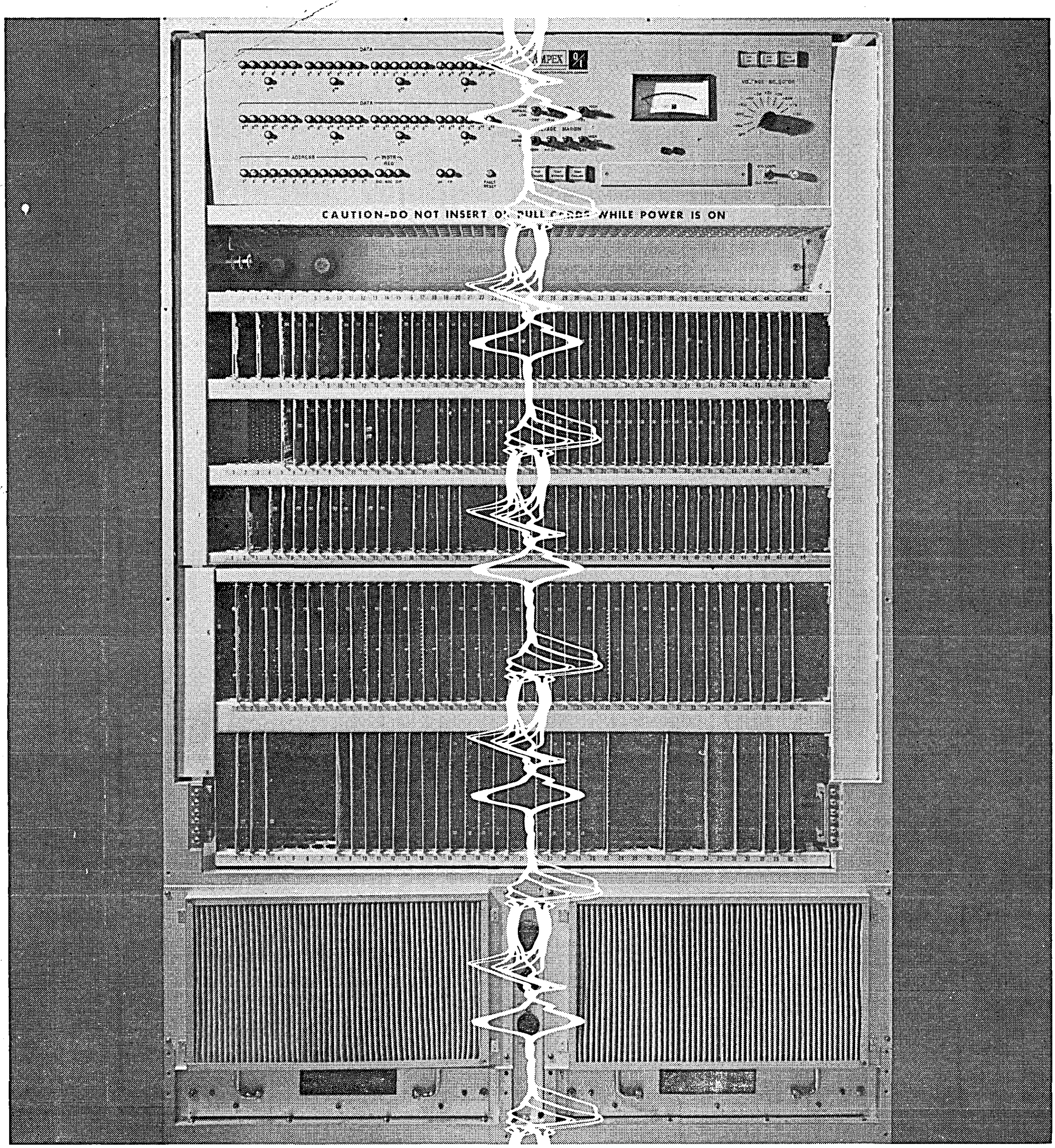
# RAND SYMPOSIUM

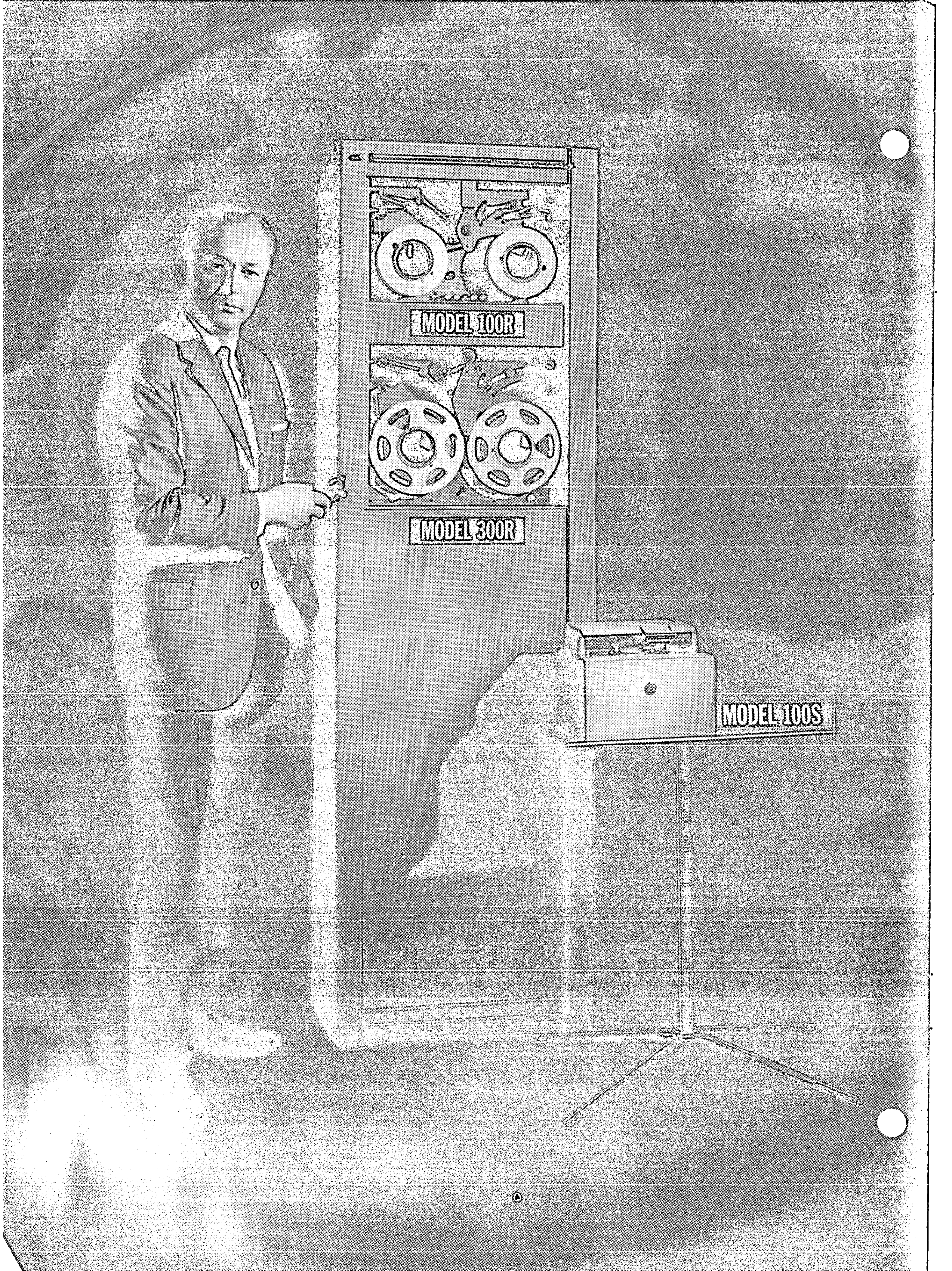


PROGRAMMING LANGUAGES
PROGRAMMING LANGUAGES
PROGRAMMING LANGUAGES
PROGRAMMING LANGUAGES
PROGRAMMING LANGUAGES
PROGRAMMING LANGUAGES
PROGRAMMING LANGUAGES
PROGRAMMING LANGUAGES
PROGRAMMING LANGUAGES
PROGRAMMING LANGUAGES



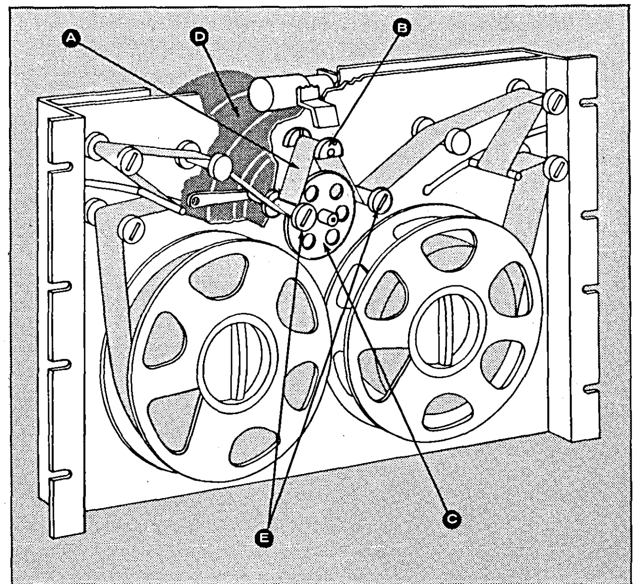





MODEL 100R
MODEL 300R
MODEL 100S

# WHAT MAKES THESE TAPE READERS THE BEST IN THE WORLD?

**Mechanical simplicity...which yields a degree of reliability unattainable by any other paper tape reader! Simplicity made possible through the utilization of the revolutionary PMI printed motor direct drive servo. Movement of the tape through the read head is achieved by merely starting and stopping a printed motor. The brakes, clutches and pinch rollers that cause big trouble and down time in conventional tape transports are completely eliminated.**




Line by line cycle: movement of tape (A) over read head (B) is controlled by drive capstan (C)—attached directly to shaft of PMI printed motor* (D); spring-loaded rollers (E) hold tape gently against capstan, keeping tape movement in exact accord with capstan rotation; advance command pulse accelerates motor, capstan, and tape; as read head detects next sprocket hole, a reverse pulse to motor halts capstan and tape with next character perfectly aligned in read head.

**U.S. Patents of Printed Motors, Inc. Pending.*

Photocircuits' 100% solid state readers perform positive line-by-line reading at up to 300 c/s; can change speed and alter it infinitely from 0 to 300 c/s without mechanical adjustment; are easily loaded; will transport and read crude splices; and can be clock synchronized with a computer up to 300 c/s, eliminating the need for buffer storage.

Console, rack-mounted, reel-servoed, uni- or bidirectional transports in a low (up to 60 c/s) and high (free-running at up to 400 c/s) speed range are available. Strip readers and separate spoolers, various tape widths, block type reading, special input-output ratings, militarized units, and remote control can be supplied.

Phone, wire or write for complete information.

Photocircuits
CORPORATION
TAPE READER DIVISION
Glen Cove, N. Y.

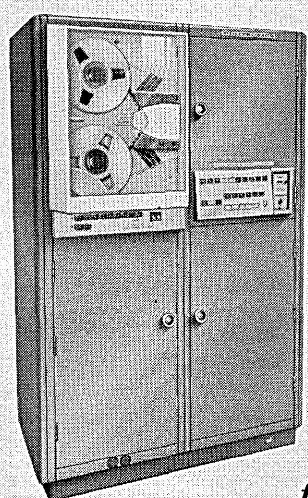
CIRCLE 5 ON READER CARD

at DIGITRONICS

they chose

# POTTER

## TAPE TRANSPORTS



### *for the Dial-o-verter System*

The Digitronics Dial-o-verter System links computer and data processing centers from coast-to-coast by transmitting data at high speed over voice grade telephone lines, broadband or microwave facilities, complete with error detection and correction features. It also offers interchangeability of media (magnetic tape to paper tape, etc.).

Key to this highly reliable system is the Potter M906 II Magnetic Tape Transport. This high-speed record/playback transport is equipped with all solid-state electronics mounted on individual plug-in circuit boards. A tension arm slack loop system combined with patented dual vacuum buffer provides smooth cycling of the tape at stepping rates of 200 per second.

In addition to its application in telephone line transmission systems, the versatile M906 II is extremely useful in modern high density recording systems, providing data transfer rates of up to 500kc.

To learn how Potter Magnetic Tape Transports can be applied to your data processing problems, write for details today.

T.M.

POTTER INSTRUMENT CO., INC.
*Sunnyside Boulevard • Plainview, New York*

CIRCLE 6 ON READER CARD

# DATAMATION

*the automatic handling of information*

*volume 8, number* 10

## Staff



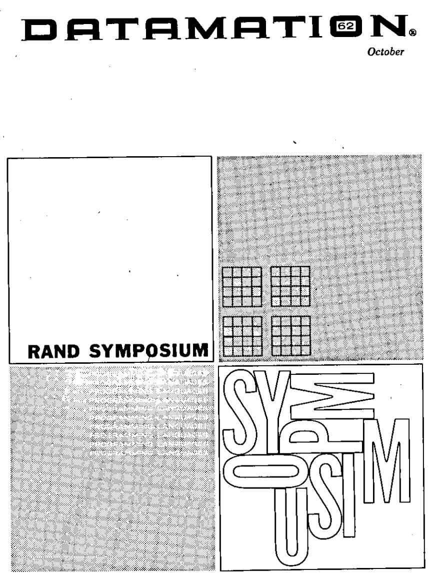
## Feature Articles



## Departments



TIHS ISSUE—41,198 COPIES




## Cover

An annual event of major scope in the computing profession, DATAMATION is pleased to offer part one of The RAND Symposium. Participants in this year's session point their verbal gambits at programming languages. Cover design is by Jim Brooke of Playa del Rey, California.

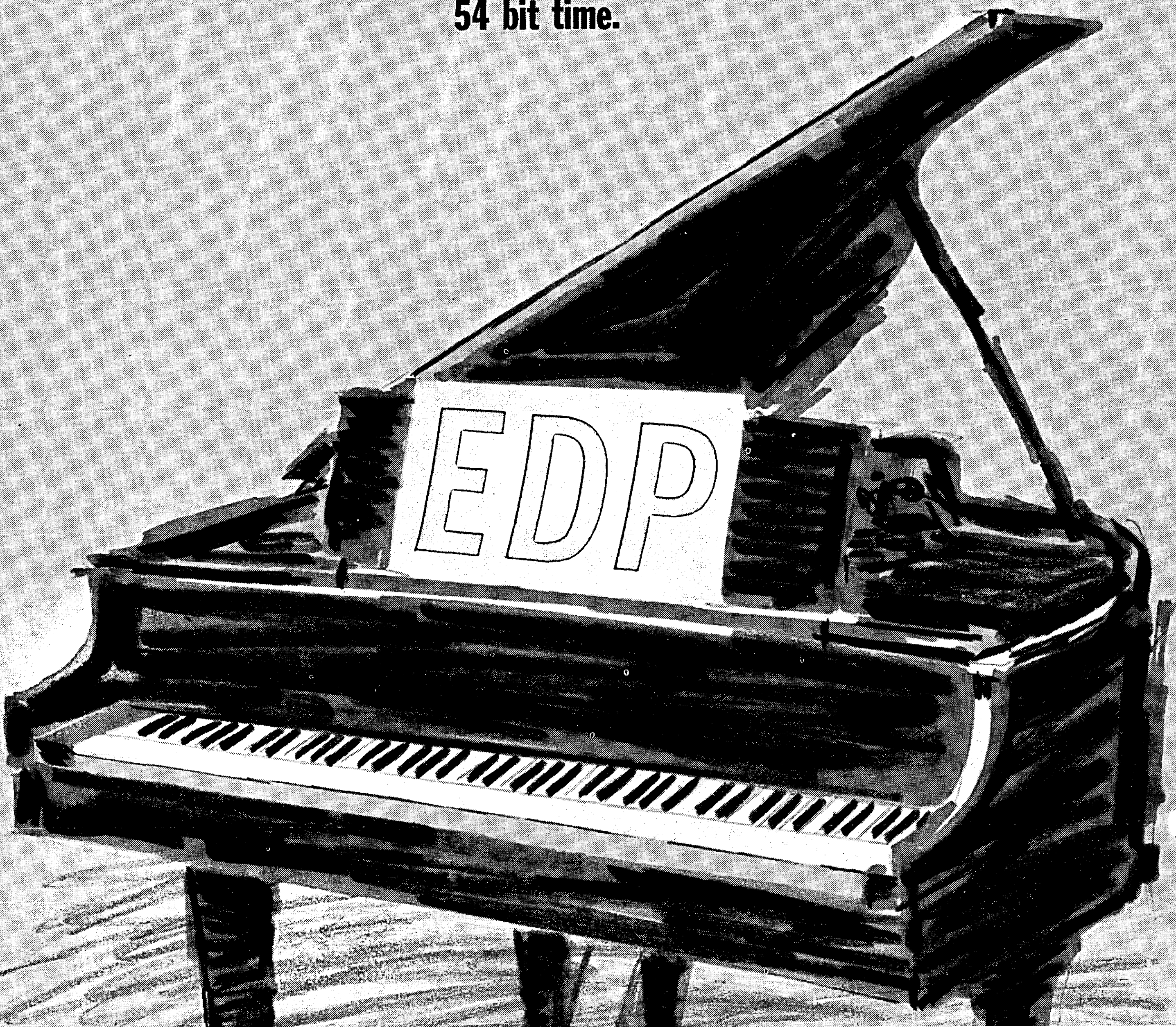
Circulation audited by Business Publications Audit BPA

Member, National Business Publications NBP

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, 10373 West Pico Blvd., Los Angeles 64, Calif. (BRadshaw 2-0817). Published at Chicago, Ill. Accepted as controlled circulation 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.

# Plays any tempo:

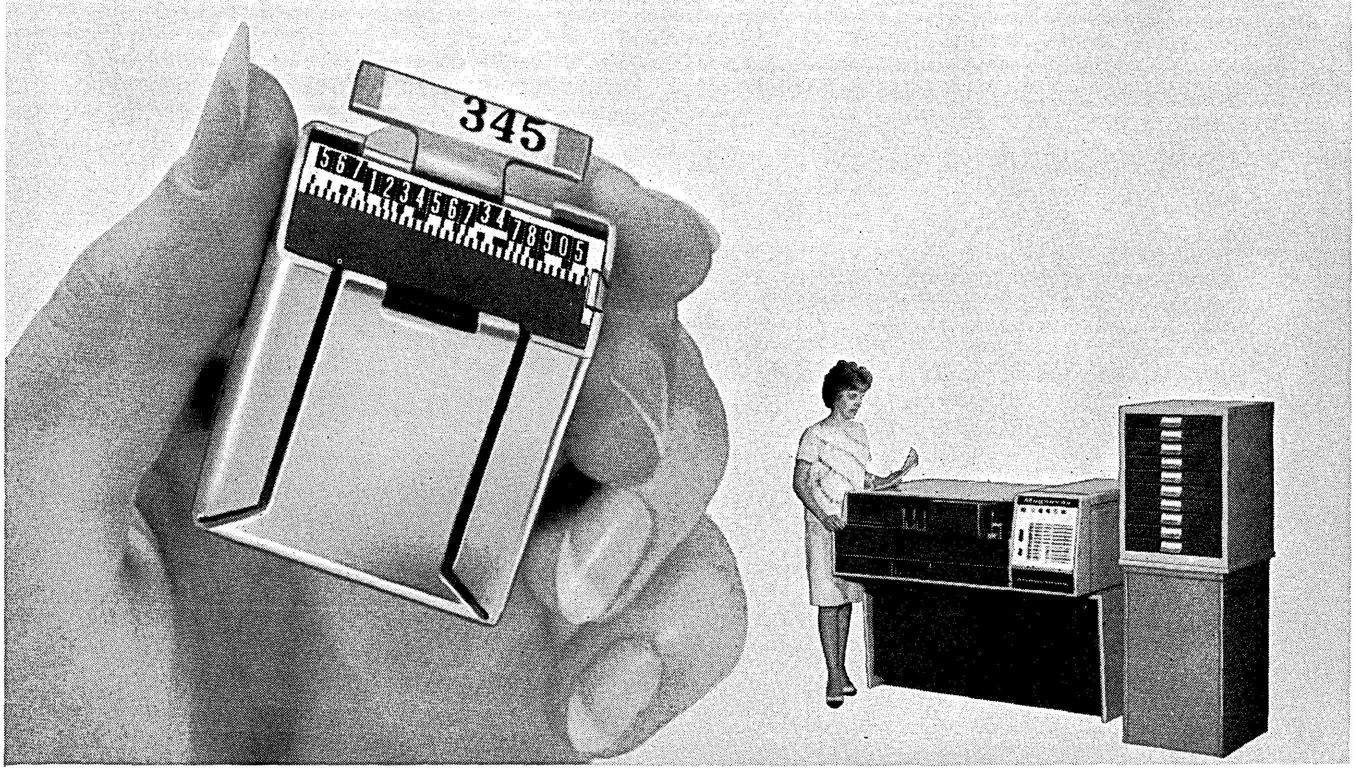
**12 bit time,**
**24 bit time,**
**36 bit time,**
**48 bit time,**
**54 bit time.**



# important DATES

- Armour Research Foundation is sponsoring the 1962 Computer Applications Symposium, to be held Oct. 24-25 at the Morrison Hotel, Chicago.
- The Ninth Institute on Electronics in Management, sponsored by the American University, Washington, D. C., will be held October 29-November 2, at the International Inn, Thomas Circle, Washington.
- The 15th annual International Systems meeting is scheduled for Oct. 29-31 at the Hotels Statler Hilton and Sheraton Plaza, Boston, Mass.
- A Conference on Spaceborne Computer Engineering will be held Oct. 30-31 at the Disneyland Hotel, Anaheim, California.
- The Operations Research Society of America will meet from Nov. 8-10 at the Sheraton Hotel, Philadelphia, Penna.
- The 1962 Fall Joint Computer Conference will be held on Dec. 4, 5 and 6th at the Sheraton Hotel, Philadelphia, Pennsylvania.
- The AIEE Winter General Meeting, Jan. 27-Feb. 1, 1963, will feature special sessions on Artificial Intelligence. The meeting is scheduled for New York City.
- The AIEE/IRE International Conference on Nonlinear Magnetics will be held at the Shoreham Hotel, Washington, D.C., April 17-19, 1963.
- The 1963 Spring Joint Computer Conference will be held May 21, 22 and 23rd, 1963, at the Cobo Hall, Detroit, Michigan.
- The 1963 ACM National Conference will be held Aug. 28, 29, and 30th in Denver, Colorado.
- The 1963 Fall Joint Computer Conference will be held in the Las Vegas, Nev., Convention Center, Nov. 12-14, 1963.
- The 1964 ACM National Conference will be held in Philadelphia, Penna., Aug. 25-28, 1964.
- The IFIP Congress 65 is scheduled for New York City in May, 1965. It is the first International Congress scheduled for the United States.

# MEDIA(TM) *another first from* Magnavox




# Just 60 seconds to retrieve and make hard copy of any one of 30,000,000 documents

This new, high speed, electronically controlled filing system provides an entirely new concept in document storage and retrieval.

Called **MEDIA** (Magnavox Electronic Data Image Apparatus) it locates and produces a full size black and white hard copy of any page in a file of 30,000,000 pages or more, in less than a minute. As a result, you can realize substantial savings in time, floor space and office overhead.

Material is filed in an absolute minimum of space since each tiny metal capsule holds up to 600 pages. Documents are never removed from the system which consequently assures you of a completely inviolate file, and entirely eliminates the problem of lost documents and refiling errors.

**Magnavox**

*Half a century of excellence in electronics*

Automatic purging of obsolete documents, random updating, and low cost duplicate files are some of the many additional advantages offered by **MEDIA**

*For complete information on equipment and prices, please fill out and mail coupon below.*

The Magnavox Company, Dept. 31102
**MEDIA** Sales Department • Fort Wayne, Indiana

Send me complete information and prices on **MEDIA**. My requirements will involve the filing of approximately_____ documents, averaging_______pages per document. These pages are a maximum of_______inches in size and are printed on the front only ☐ front and back ☐

*Name*____________________

*Title*____________________

*Firm*____________________

*Address*____________________

*City*____________ *Zone*____ *State*________

CIRCLE 8 ON READER CARD

# letters

## debugging a term

*Dear Sir:*

While wishing in no way to quarrel with the essence of Harry Cantrell's remarks on "Where Are Compiler Languages Going?" in the August issue of DATAMATION, I do believe he has used the term "source language debugging" improperly. I am sure many readers would agree that the facility which Harry describes is "Symbolic Modification" and that "Source Language Debugging" is that feature of a Compiler and/or Operating System which enables it to return to the programmer in source language form diagnostic information generated during a checkout run.

The latter may also include the facility for stating debugging requests in the source language, i.e., a special application of symbolic modification!

THOMAS G. SANBORN
*Associate Manager*
*Data Systems Dept.*
*Computation and Data Reduction Center*
*Space Technology Laboratories*
*Redondo Beach, Calif.*

## skenekadee

*Dear Sir:*

Eye wil not mispel Schenectady.
Eye wil not mispell Schenectady.
Eye wil not misspell Schenectady.
Eye will not misspell Schenectady.
Aye will not misspell Schenectady.
I will not misspell Schenectady.

A CAREFUL READER
*Warren, Michigan*

## the need to learn

*Dear Sir:*

I would like to make a comment on Mr. Cantrell's excellent article in the August issue, "Where Are Compiler Languages Going?" The requirements for ease of learning are quite realistic! But, why should it be necessary in most instances for people to learn to *read* the language without learning to really use it (this is often dangerous)? If we remove the restriction that anyone be able to quickly assimilate the language enough to read it, we can abandon the verbosity and other drawbacks of modeling it after English or even algebra and pick the best language for the purpose. It is of interest to note that the simpler selfcompiling compilers (see NELIAC and JOVIAL, etc.) look not too different from modern assemblers (see UNIVAC-CSC's UTMOST and also the suggestions of George Mealy in the July ACM letters).

We have a lot to learn before standardizing computer languages. But when the day comes, I suspect the language(s) will be as incomprehensible to the layman as a text in higher mathematics is today.

CHARLES SWIFT
*Computer Sciences Corp.*
*Palos Verdes, Calif.*

**fan mail**

*Dear Sir:*

How many other readers do you suppose are bearing in silence the heavy-handed attempts at humor of J. W. Granholm? Granted, that a touch of humor is desirable to relieve the seriousness of DATAMATION. But can't you find something better than Granholm? This man really isn't very funny. His humor is not only forced, but is nowhere near an adult level. The latest attempt, "How to Hire a Programmer," seems to rely for humor on the single device of maligning the intelligence and actions of every sort of individual connected with computers from the purchaser to the equipment supplier and coder.

May I repeat, there must be better humor around.

THOMAS L. GERBER
*White Plains, N.Y.*

(Editor's Note: Frankly, we have inquired in and around your geographical area with only limited success. To be truthful, we feel Granholm is just fine.)

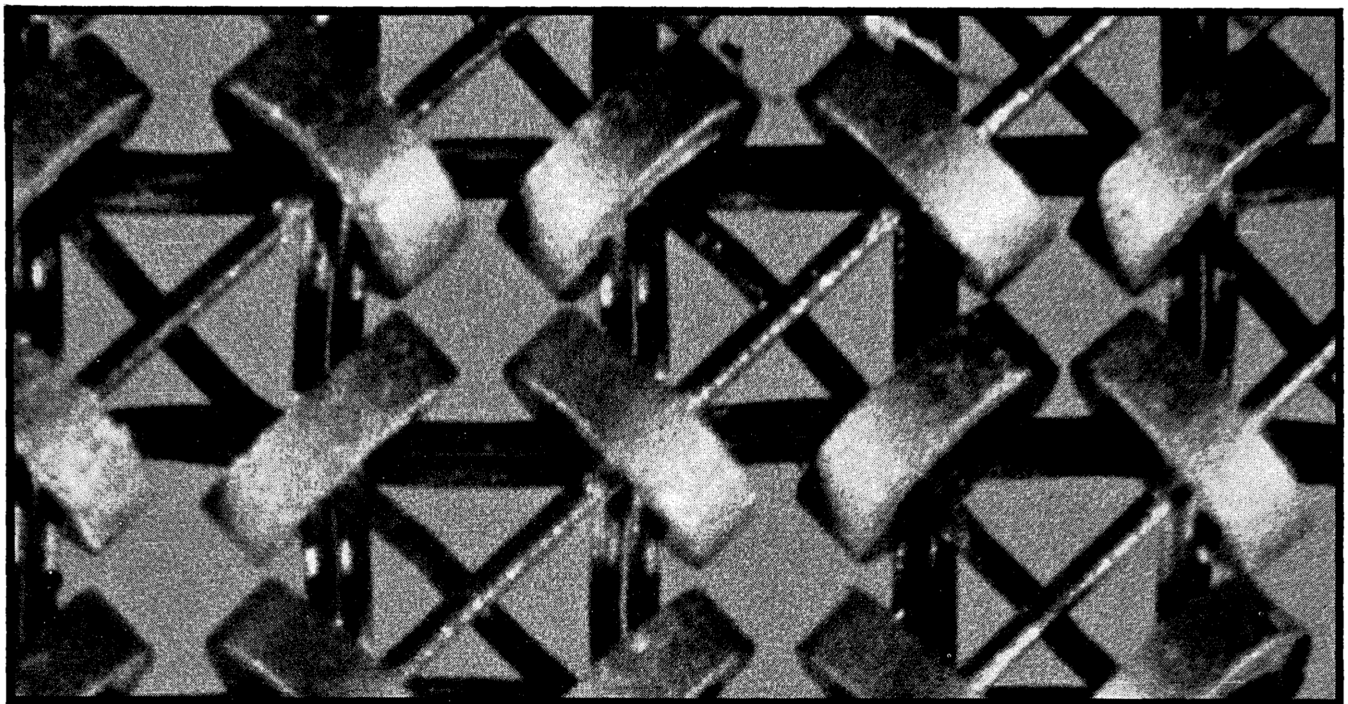
**myopia**

Sir:

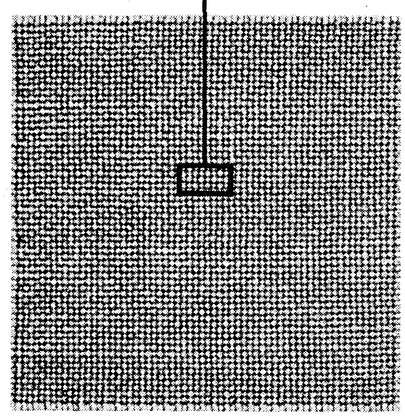
In your editorial "A Long View of a Myopic Problem" you mention that ". . . discontent with one's job and working conditions is increasingly evident . . . and the quality of individual productivity has suffered proportionately." Your ads in the back of your publication verify that the industry knows this and tells you why.

Take a look at some of these quotes: "(He) soon began to feel tied down by little things . . . what with all the magnetic red tape, he couldn't seem to get his career off the ground . . . and what he'd hoped would be a job with broad horizons proved too confining," and "Our engineer scientist oriented management has taken every step toward providing the type of environment that increases technical creativity."

CRAIG BELK
*Douglas Aircraft*
*Santa Monica, California*



50 TIMES MAGNIFICATION



5-WIRE COINCIDENT 64 X 64 MATRIX
(ACTUAL SIZE)

# It takes big minds to think this small

. . . good, sound engineering minds with that added spark of boldness that frequently solves the seemingly unworkable problem. The core memory frame above is a case in point. A 5-wire coincident, 4,096-core matrix less than 2 inches square seemed impossible to make on a production basis. But Fabri-Tek engineers devised special production aids and techniques, and we now produce them in volume.

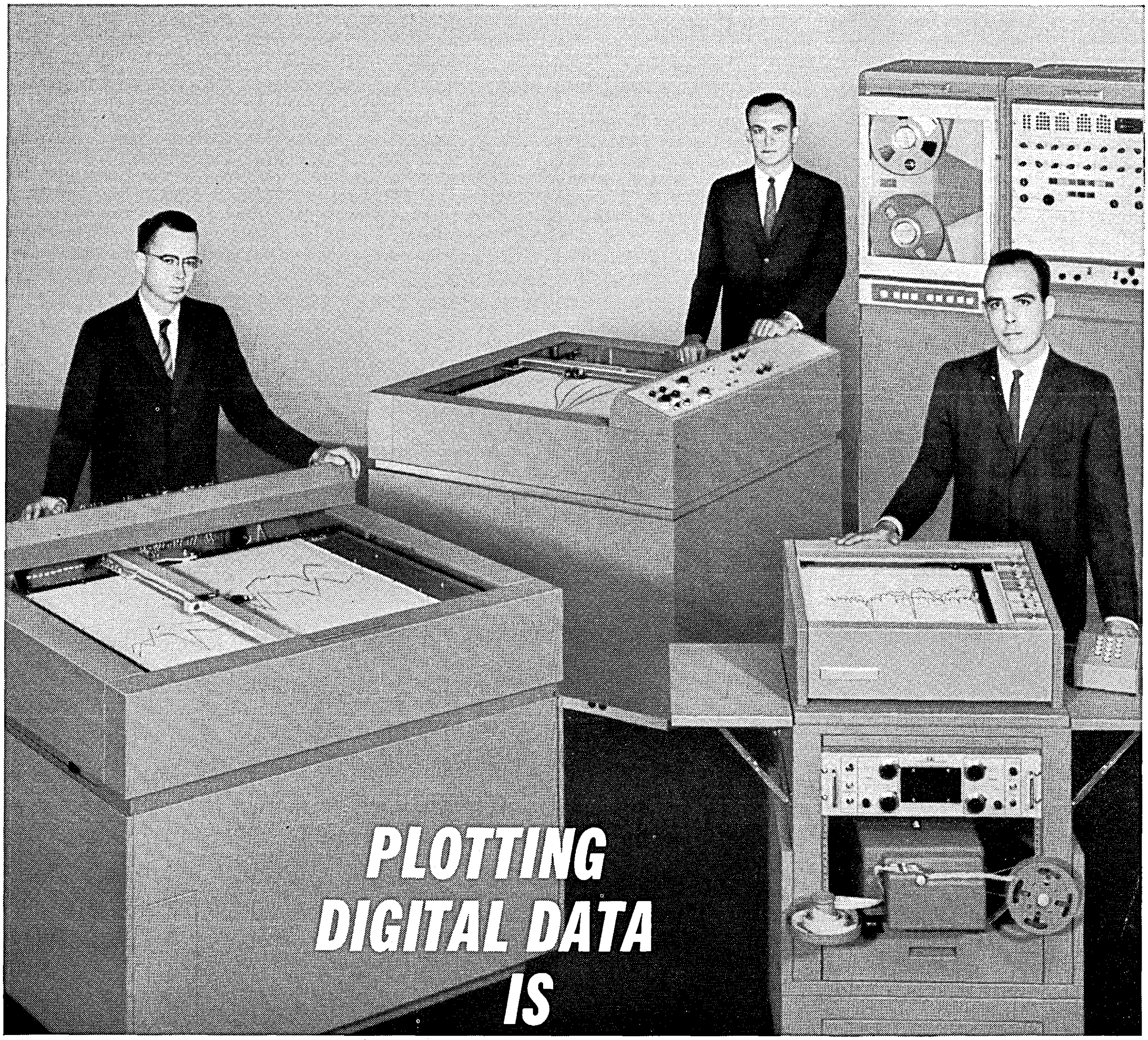
Fabri-Tek engineers fit core memory frames and stacks to your systems. Cores from any quality manufacturer can be used to match your speed and environmental requirements. You get exact electronic and physical fit.

Complete memory systems from Fabri-Tek have been designed with simplified, standard circuits for reliability, ruggedness and easy maintenance. Other features include solderless connections on all inner wiring, test points on printed circuit modules, address and data information indicators, matched input, output and control levels, and optional self-testing features.

**New color brochures** describe printed circuits, memory frames, memory stacks and complete memory systems. Write today for your free copies. Fabri-Tek, Inc., P. O. Box 8046, Minneapolis 16, Minnesota.

FABRI-TEK®

CIRCLE 11 ON READER CARD



# PLOTTING DIGITAL DATA IS OUR SPECIALTY

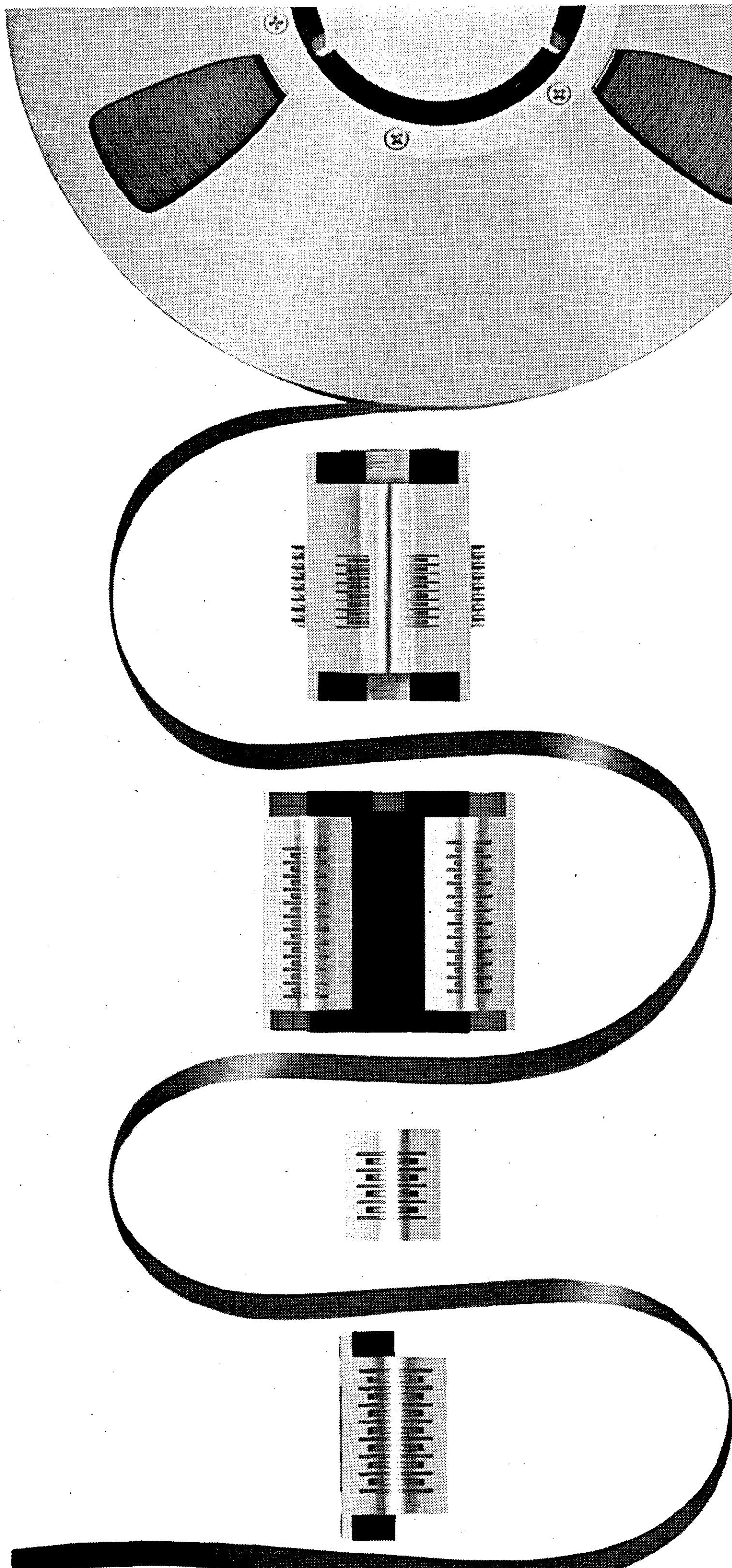
*Since 1952, EAI plotting equipment has been applied to a steadily lengthening list of data reduction applications — from simple, manual point plotting to high-speed magnetic tape input contour plotting. Again and again, the flexibility, speed and extreme accuracies of EAI plotters have dictated their selection over competing instruments. ◻ EAI standard plotters include 11″x 17″, 30″x30″, and 45″x60″ boards. Operation can be either off-line from punched cards, punched tape and magnetic tape, or on-line with various computers. Output modes include point, line, symbol, and contour plotting. Plotting speeds up to 4500 line segments per minute can be provided. Reliability is assured by solid-state circuitry and superior mechanical design. ◻ You can draw upon EAI's wide application and design knowledge by describing your requirements. Write for information, detailing your needs, today.*

**EAI**

**See this product demonstrated at our booth No. 1372 at the ISA Show in New York City, October 15-18.**

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

CIRCLE 12 ON READER CARD



# Whether it's 1 or 1,000 every Brush magnetic head is 100% tested to insure system performance

Detailed mechanical and electrical test data is on file for every head manufactured. Complete production drawings allow duplication at any time. Modern quality control and production facilities are unequaled in the Industry. Couple this with the largest full time technical staff concentrating on magnetic heads and you'll understand why Brush cannot be matched in the sum total of capabilities necessary to stay in advance of your system requirements. Visit Brush and see for yourself, or write for full details.

**brush INSTRUMENTS**

DIVISION OF CLEVITE 37TH AND PERKINS, CLEVELAND 14, OHIO



CIRCLE 13 ON READER CARD

CIRCLE 15 ON READER CARD

# BUSINESS & SCIENCE

## CONGRESSMEN QUERY NAVY ADOPTION OF JOVIAL

Despite the Navy's recent adoption of JOVIAL for Command and Control (See Datamation, p. 17, August, 1962), the issue is hardly at rest; in fact, it may soon develop into a political hassle of major scope.

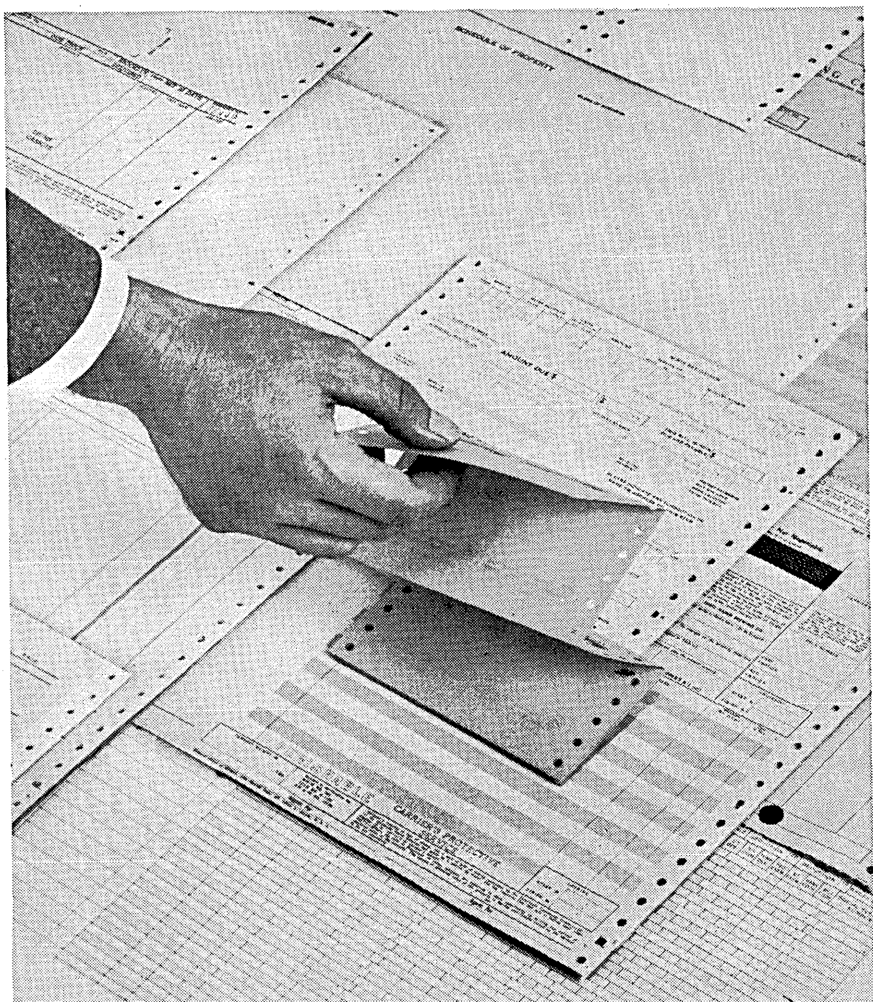
The Navy's position has been questioned through inquiries lodged with Senator Clair Engle and Representative Jeffery Cohelan, and based on the following contentions:

1. The adoption of JOVIAL occurred as one of the results of a meeting chaired by an applicant for a Civil Service position with NAVCOSSACT who was on a vendor's payroll at the time of the meeting.
2. Navy officials did not pay sufficient attention to a $22,000 study they commissioned in which Computer Usage Corp. concluded that NELIAC operated at considerably faster compiling and executing speeds than JOVIAL.
3. The question was raised as to the present operational state of the JOVIAL compiler and the cost of its maintenance to the Navy. NELIAC advocates argue that their compiler was completed under Navy funds and is, in fact, currently in use at the Naval Electronics Lab in San Diego.
4. The initial results of a still-to-be-completed Air Force compiler test (see below) indicate that compile times for NELIAC are about eight times faster than JOVIAL.

The Navy's reply to these inquiries stated that their new technical director was officially employed by the Navy on April 17, after a decision had been reached to adopt JOVIAL. His relationship in the decision-making process or his affiliation with the vendor is not discussed in the Navy reply. Two of the Navy's reasons for adopting JOVIAL are that JOVIAL may be used as part of the standard language to be adopted for a unified command, and that JOVIAL is a more powerful compiler than NELIAC. The cost of maintenance for JOVIAL was not volunteered in the Navy reply.

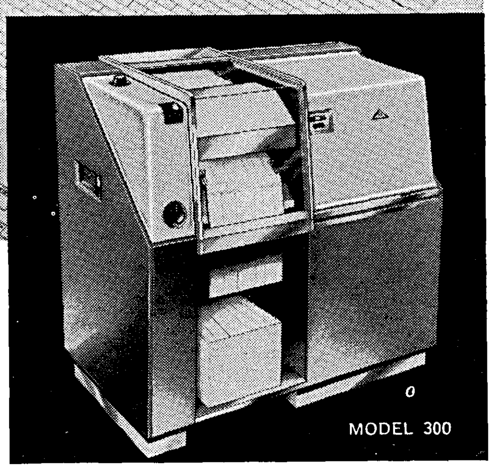
## AIR FORCE TO REDO COMPILER TEST

The first stage of a compiler test on the 7090 has been completed by the Department of the Air Force. However, the test will be redone since it was felt



# BIG PRINTER CAPABILITY

# SMALL PRINTER COST




For banking, commercial, control and scientific applications... designed for use with small to medium range computers.

▲Printing rate 300 lines per minute ▲Accommodates single or multiple part fanfold forms 3″ to 19″ wide and 22″ deep ▲Handles heat transfer and pressure sensitive papers; also card stock ▲Prints up to 136 columns, 10 columns per inch; 26 letters, 10 digits and 12 symbols per column Provides a choice of line spacing 6 or 8 per inch ▲Solid state circuitry used throughout ▲Soundproofed housing contains Print Head, Power Supplies, Buffers and Logic

ANELEX

SYMBOL OF QUALITY

Further Information Available on Request

**ANELEX CORPORATION**

155 Causeway St., Boston 14, Massachusetts

CIRCLE 16 ON READER CARD

# THE SAD COMPUTER


NO. 1 IN A SERIES

**I am a sad computer. You'd be sad too if someone told you to do a job and didn't explain what it was all about. That's the problem a lot of us computers have. Some people just don't understand us. They don't realize that three very important steps . . . Consulting, Systems Design, Programming . . . must be successfully completed before we can possibly go to work effectively. It's not that we're temperamental or anything like that; these are just the facts of life. I've heard via the "grape line" that all of the computers at the McDonnell Automation Center are always happy. I know my boss won't let me go there to live, but maybe he'll write them or call them up and ask how they can make me happy too.**

Call or write today for literature on Consulting, Systems Design, Programming and Data Processing.

**MCDONNELL AUTOMATION CENTER**

*Division of MCDONNELL AIRCRAFT*

St. Louis, Missouri

Phone: PErshing 1-2121

CIRCLE 41 ON READER CARD

that execution times were not entirely meaningful. The same algorithm was not used in preparing all of the programs for the test although the same flow chart was used for all languages except CL-1.

The following results were not released by the Air Force but by a member of the committee participating in the test.

It has been pointed out that readers should be careful in drawing conclusions from the following results since the ability of the individual programmer was felt to carry more significant weight than the language itself.

AIR FORCE
COMPILER TEST
7090

| | Compile time | # of source statements produced | # of object instructions produced |
|---|---|---|---|
| SCAT | 40 secs. | 1,750 | 2,309 (macro capabilities used) |
| CL-1** | 5 mins. 54 secs. | 500 | 2,003 |
| FORTRAN*** | 1 min. 30 secs. | 380 | 2,480 |
| JOVIAL* | 3 mins. 47 secs. | 484 | 1,362 |
| MAD | 52 secs. | 353 | 3,153 |
| NELIAC* | 29 secs. | 289 | 1,020 |

* Object speeds of both JOVIAL and NELIAC were 50/60th of a sec.

** The same flow chart was used for all languages except CL-1.

*** For FORTRAN programming about 7% was written in assembly language.

Others were written 98-100% in the compiler language specified.

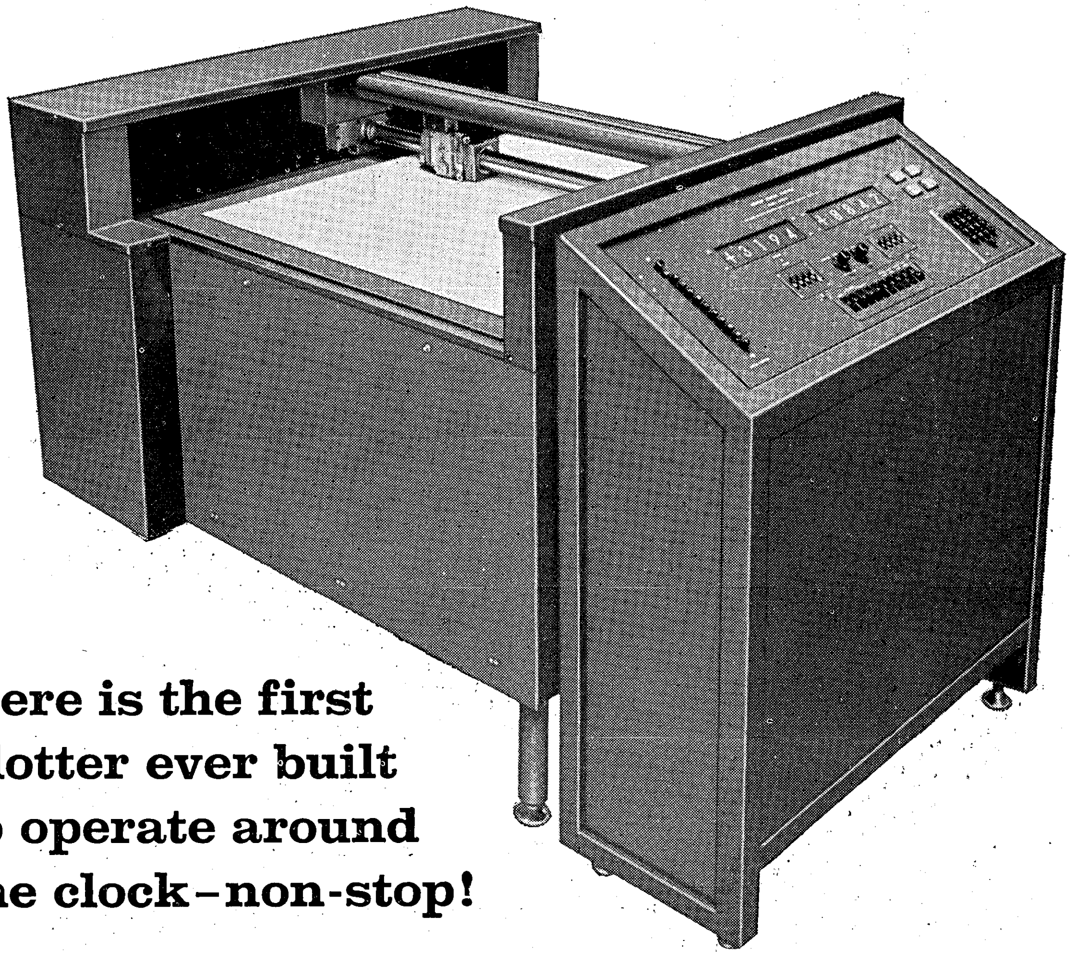
A C-E-I-R COST CUTTER

Pointedly aimed at cutting overhead expenses, C-E-I-R has realigned its seven major offices into three, budget-conscious districts. The new Western region includes Los Angeles and San Francisco. It is headed by J. Paul Walsh. Houston is included with Washington in the Southeastern region to be managed by Jack Moshman. New York, Boston and Hartford have been joined into the Northeastern region managed by George Wulfing.

REQUIEM FOR A LIGHTWEIGHT

General Mills, never a major contender in the computer industry, has thrown in the towel. Limited to a few preliminary encounters--a few sales to government agencies--the Minneapolis firm does not intend to deliver any more machines from inventory, nor is it planning any future production. The stance which General Mills will take will be one of "re-evaluation of our position in the computer market"--a somewhat plaintive utterance, usually heard following the ten-count.

# UNPRECEDENTED RELIABILITY!



## Here is the first plotter ever built to operate around the clock–non-stop!

Sure, Gerber's new plotter is as accurate as any you've ever seen. Plots to ±.002″. Repeats to ±.001″.

But even more important, the Model GP-30-D operates at pinpoint precision 24 hours a day, seven days a week. No inaccurate plots, no need for test decks every few hours. Absolutely no drift.

<u>In short, the GP-30-D will plot your data as fast and as long as your computer turns it out!</u>

And reliability is just one of the many new features built into this new plotter. Here are a few more:

**Maximum slewing speed:** 20″ per second.

**Plotting speeds:** • 50 points per minute with points spaced up to 4½″ apart from IBM cards.
- 42 points per minute when plotting at random.
- 80 points per minute with paper tape input.

**Digital Logic:** No digital to analogue conversion used. The digital logic is all solid state, plug-in cards, which actually make up a special purpose computer.

**No potentiometers:** Magnetic-type discreet shaft encoders eliminate all drift and warm-up time.

**Drive system:** Precision ball screws driven by ¼ hp. motors.

**Choice of tables:** Plotter available with a vertical table as well as the horizontal table shown.

**For plotting accuracy and reliability, there is no "or equal." Write for full details.**

Six printing wheels,
each with twelve positions.
Prints any combination up to six digits,
letters or symbols side by side.

GSI

## THE GERBER SCIENTIFIC INSTRUMENT CO.

**P. O. Box 305 • Hartford, Conn.**

Branch sales and service offices at: Washington, D.C. RE 7-6992 • Los Angeles, Calif. MU 1-5745 • Toronto, Canada AX 3-7011

CIRCLE 17 ON READER CARD


CIRCLE 18 ON READER CARD

# EDITORS' READOUT

## THE GAME OF LIMITED EXCHANGE

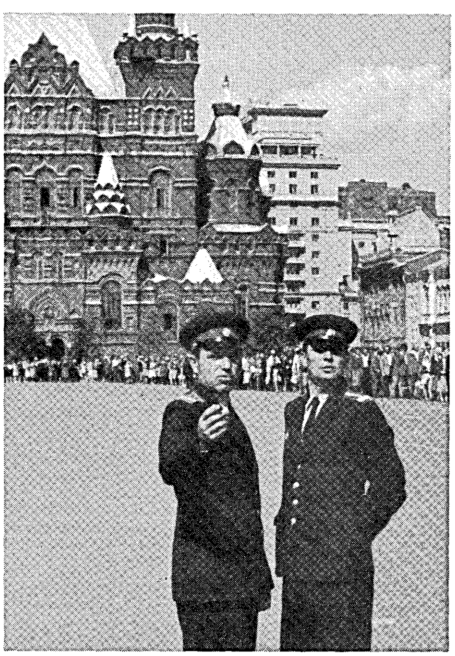
Having recently concluded a brief, eight-day visit to the Soviet Union, *Datamation* was permitted to view only a limited cross section of Russian computer technology. A more comprehensive report on this August trip will be published after translation and interpretation of collected materials has been completed.

However, one conclusion which was reached as a direct result of preparations for the trip as well as experiences inside Russia, should be made known during this interim period.

Simply stated, an awareness of Soviet computing technology is vastly limited in this country by the Russians' obvious timidity at broadcasting details of their specific scientific attainments and certainly, their problem areas. The converse however, is not true; in fact, it may be presumed that the Russians have a rather complete dossier on computing in this country from the comparatively free access which they presently have to a proliferation of technical papers, magazines and reports published.

As proof of this pudding, the Soviet willingness to communicate with Western observers on computing development is noticeably limited. This conclusion is a rather obvious one: the "game" or "balance" of free exchange is very definitely tilted in favor of the Soviets. Unfortunately, obvious conclusions are rather easily overlooked particularly when they present a thorny problem with a rather difficult, but necessary solution.

In May, 1959, a delegation of U.S. computer technologists visited the Soviet Union and the following year their report was published in the IRE Transactions. In searching available literature for *Datamation's* recent visit to the USSR, it was surprising to find that this report represents the latest survey of the Soviet computing art available in the U.S. *It is almost three years old.*



In the interim, occasional U.S. observers have been able to visit the Soviet Union and obtain a series of interviews in specialized areas within the general field of automatic information processing. Ed Feigenbaum's report on Soviet cybernetics is one such instance and was published last year in ACM *Communications.* In *Datamation's* August issue, a brief, updated report from Czechoslovakia was presented as the most current source on Soviet computing.

On the whole, however, the dearth of information is appalling. To be sure, there have been numerous articles and speeches purporting to survey this area but their actual content level has been shockingly low and even the kindest evaluation of this form of reporting suggests misrepresentation.

Of course, these facts could hardly be of much concern to the Soviets except that they are probably pleased by them. Their position with regard to releasing information is immeasurably strengthened by the continued, virtually unrestricted input from U.S. and other Western sources and the well-publicized lack of information entering this country.

In correspondence with the Soviets prior to this recent trip, replies were largely negative in the matter of granting interviews or arranging visits to computing installations. Reasons for this attitude ran a gamut from a heavy incidence of summer vacations to the surprisingly frank statement, "It is now not the practice of our institute to grant interviews." Many of our letters remained unanswered.

Upon arriving in the USSR, the situation proved far worse than may have been presumed from the preceding correspondence. Unfortunately, the timing of this trip coincided with the recent flight of two Soviet cosmonauts and to some extent, a measure of credibility may be attached to their excuses. But in general it seemed curious indeed, that *all* computing technologists were suddenly on vacation or in state committee meetings and that all computing facilities were closed.

One incident may be worth relating. At a public exhibit of a small transistorized machine, the Russians were asked for a sample of paper tape from an operating perforator. Their reply was that this request must be approved by the workers of the factory. When we pointed to a waste basket containing both paper and mag tape and obviously awaiting incineration or similar disposition, the same reply was forthcoming.

At the Munich Congress, less than half of the expected Russian delegation was able to attend. Many of their scheduled papers were cancelled.

In a screening of what little Russian literature is available, we have found reports of U.S. programming languages copied verbatim without even the implication of credit.

Finally, on several occasions during *Datamation's* visit to the USSR, it was noticed that copies had been made of several elements of U.S. peripheral equipment including card readers, sorters, key punches, etc.

What can be done to tilt the information flow in an opposite direction? In a free society, probably very little. Magazines will continue to be published, papers written and reprinted, conferences held and Russians invited; in fact, in most instances, acceptances by Russians will be reflected as rare accomplishments by the conference planners.

The solution is not an easy one and at best, may be only partially effective. But an effort must be made and in this respect, we would suggest the following:

1. For publications, circulation lists should be screened and copies sent behind the iron curtain only when there is a direct return of a comparable publication. *Datamation* is now employing this policy.

2. For conferences, invitations should be extended only for legitimate contributions to knowledge rather than a polite skirting of academic fringes. In addition, invitations to Soviet delegates should be offered only if and when a similar invitation is extended to the Western world for a Soviet conference (such as a recent Fall conference in Moscow on automatic programming to which no Western delegates had been invited or a similar conference in Warsaw last year to which no invitations were extended outside of iron curtain countries.)

3. If an exchange delegation is to be arranged, we could insist that America be first to participate, rather than last, as was the case in 1959 when the Soviets visited the U.S. and our return visit was purportedly based on what they had been shown here.

4. Manufacturers and societies should be asked not to engage in a "friendly" exchange unless there is some actual substance to the exchange and it is bi-directional in fact as well as in expectation.

5. In general, we should refrain from holding the Soviets in a position of awe and recognize their unwillingness to communicate for precisely what it is, rather than as an attractive halo at the end of a dark tunnel.

It is unlikely that even a determined, planned effort of limiting our output to the USSR will have a serious effect on Russia's ability to obtain this information. It would be virtually impossible nor would it be desirable to place as tight a band of secrecy about the Western world as presently exists for Iron Curtain countries. However, a firmer attitude "may" induce a greater willingness to communicate on the part of the Russians or at least might serve as recognition on our part, that the game of limited exchange can be a two-sided one. ■

The RAND Symposium convened in Santa Monica, California, for the fifth successive year one day before the Spring (formerly Western) Joint Computer Conference. Departing from the previous year's format, where the participants were free to select a series of topics, the 1962 Symposium was limited to a single subject: programming languages.

The Symposium, while having no legislative effect, is a meeting of individuals prominent in the industry who spend a day at RAND in the sharing of ideas.

The all-day session was tape recorded, a 208-page manuscript prepared, expurgated by participants and further edited by DATAMATION to meet space requirements.

Chairman of the session was Robert L. Patrick, a RAND consultant. Other participants included: Paul Armer, RAND; George Armerding, RAND; Phil Bagley, MITRE; Howard Bromberg, RCA; Tom Cheatham, Computer Associates; Dick Clippinger, Honeywell; Joe Cunningham, USAF; Bill Dobrusky, SDC; Bernie Galler, Univ. of Mich.; Barry Gordon, IBM; Fred Gruenberger, RAND; Jerry Koory, SDC; Jack Little, RAND; Brad MacKenzie, Burroughs; Dan McCracken, consultant; Ascher Opler, Computer Usage; Charles Phillips, BEMA; and Dick Talmadge, IBM. Opinions expressed do not reflect individual corporate viewpoints.

# THE RAND SYMPOSIUM: 1962

## part one — on programming languages

PATRICK: The object of today's discussion is enlightenment on the subject of common languages. Let's look at Figure 1. This chart will indicate why we should all be concerned. It is rumored that the DOD intends to standardize on a language for Command and Control. This chart shows an area characterized by the name, Business, usually characterized by a high ratio of input and output to computing. On the left we also have Scientific jobs which are characterized by just the inverse. We have all heard of the famous Los Alamos job in which you read one card in the morning, compute all day, and print one line in the afternoon. Of course, this job doesn't exist; in fact neither of these areas exist in pure form, but sections and subsections of jobs do fall into these categories. There is an overlap of the two as indicated by this Venn diagram intersect. This overlap is seen in areas like data reduction and heavy actuarial work. In these area you may think of it as scientific, but it has many characteristics of a business job. Command and Control has characteristics which are common to these two and also many characterisics of is own. Real time applications have characteristics common to Command and Control and also characteristics unique to real time problems. Consequently, if anything were to happen such as a standardized language for Command and Control, it would affect the entire field.

The lower portion of the chart shows a simple equation. The equation is an expression for total cost. (A similar expression could be written for time.) As the equation shows, total cost is made up of dollar amounts spent for training, programming, coding, compilation, assembly, testing, production, maintenance, and documentation. All of these factors are involved in every single job that is done on a computer. (In some cases the coefficient of a given term may be zero. For example, if you have the facility for compile-and-go, then the assembly phase does not appear.) But for most problems, all the terms exist with non-trivial coefficients.

Sales literature and some of the "technical" literature that we read implies that we can reduce some of these terms without at the same time increasing some of the other terms. Gullible citizens believe these statements. I believe that we will be doing ourselves and our country serious damage if we fail to see through these claims.

Now, would anyone like to open up the informal discussion with a fact?

OPLER: I would rather open with a question. Do you distinguish between a common language and a standard language?

PATRICK: I think we would like to if you would care to describe them for us.

OPLER: A common language is one that would be acceptable to two or more disparate machines. A standard language is one maintained and enforced as a standard language which requires some sort of committee agreement.

Some manufacturers had developed languages which could be processed on several of their machines. These would be common languages, but they would not be standard languages.

LITTLE: Are we saying that a common language may not be the standard, but a standard language, by definition, must be common?

BROMBERG: I feel it is unnecessary to labor this distinction. For our purposes here, it seems to me we can discuss only common languages, by which we mean those which can function on more than one machine type.

PATRICK: Maybe it would be appropriate to discuss whether we want one. I think there are some of us who do, but then there are also some of us who don't. Dick, would you like to discuss this point?

TALMADGE: It's my opinion that we don't want a com-

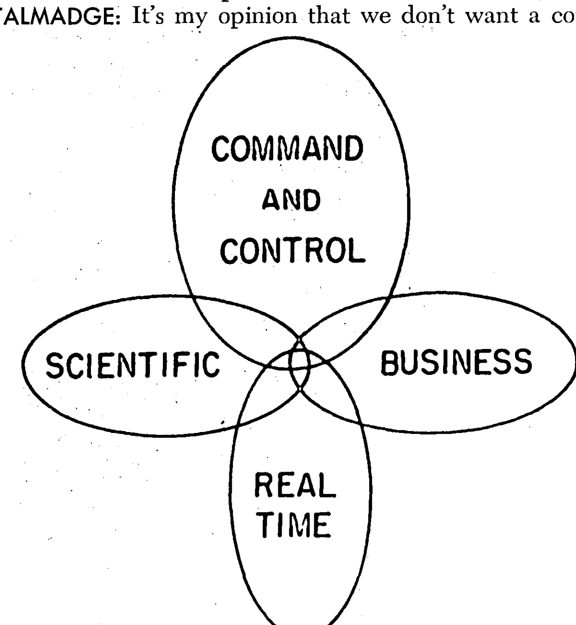



Figure 1

COST = \$TRAIN + \$PGM + \$CODE + \$COMP + \$ASSEM + \$TEST + \$PROD + \$MAINT + \$DOC mon language in the sense that most people use that term; that is, a language that everybody would use. To draw an analogy, mathematics, which people tend to think of as using a standard language, in reality uses a standard set of notations that are widely understood. Furthermore, a mathematician may use his own notation within very elastic limits, as long as the meaning is clear. One of the most acute problems in computing is that we have not yet developed procedures which allow such a standard notation. Until we know more about linguistic structure, it is unlikely that we will. In any event, a common language (whatever that may be) does not seem to be the answer.

I think we have a lot to learn about how to use a language to express a given class of problems; and how to determine what problems are amenable to a given language.

BROMBERG: It strikes me that the same comment can be made with reference to every single human endeavor; we have a lot to learn.

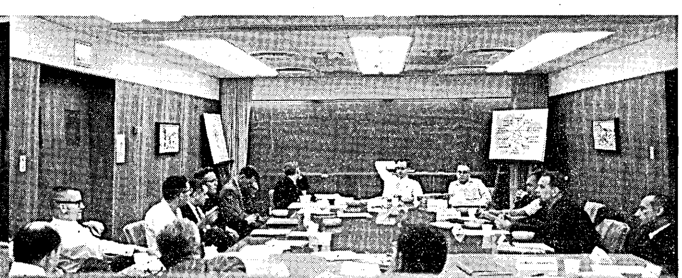
TALMADGE: I didn't intend to imply that one should not use whatever tools are available. I merely meant that I think the present tools are inadequate for the job of forming a permanent language; rather, they should be used to develop more powerful tools.

BROMBERG: I assumed that the question was, "Is this the time for a common language?" and I would say this is certainly the time.

GORDON: I would like to point out, Howard, that there is a significant difference between utilizing and standardizing.

I think everyone is in favor of standardization in much the same way everyone is in favor of motherhood; but also, like motherhood, if it occurs prematurely, it can cause a great deal of inconvenience.

BROMBERG: Let me give you a little of the background of what is going on in ASA, BEMA, and the X-3.4 Subcommittee on Common Programming Languages. There has been established a survey director whose function it is to survey all the common programming languages. An interesting part of this survey is that assembly languages like SAP, Autocoder, and X-1 have by definition been precluded. They are not incorporated in the survey because of obsolescence or the level of the language. The only kind of common language admitted to the survey is that which is effectively non-machine oriented; that is, it does not look like any known machine order code. The entries are essentially problem languages as close to natural language as we can get.



McCRACKEN: How many common languages have you listed?

BROMBERG: There must have been about 90. I can't be too sure.

PATRICK: Howard, I can't think of the *first* one, if you set up the requirement that it must be machine independent.

BROMBERG: Of course, it all depends on what you mean by machine independent. Must it be totally machine independent?

PATRICK: Well, if you are going to cut across machine lines, it seems like it ought to be.

McCRACKEN: It doesn't have to be perfect in order to be better than what we now have. That's going to be my standard speech today.

GORDON: But the new language we adopt should be significantly better than what we now have.

ARMER: Maybe that's why we're having so much trouble now. We don't have enough standardization at the trivial end. For example, on character sets.

PATRICK: I seem to hear three different cheers for that sentiment.

LITTLE: Languages may evolve or standards may evolve. I think that languages can get to be standard or common in a lot less exotic ways. If a big enough user uses a single language without regard to the rest of the world, then you are sort of saddled with that language. DOD may be going down this road and whether you like it or don't like it you may have the situation with you very shortly. I would like to hear an outline of some methods that can be used to help insure that the language we get is a good one. I am sure that we are going to get one. But I don't think we're going to get one that somehow magically everyone will like.

There are certain things that you want built into it. For example, you want it to have the ability to change – to make itself better, or for people to make it better.

PHILLIPS: I'd like to go back and add something to what Bromberg said about the ASA-X3 program. One part of that program concerns the development of a standard language. I'm quite sure (although this may be just an opinion) that the 18 members of BEMA or the general interest and user groups that are supporting this program would not feel that they are bound to adopt and use, to the exclusion of everything else, a language that might be developed as a result of the program. By the same token, I don't think that Defense, in supporting the COBOL program, had any intention of making it a required program for use throughout Defense unless there was good reason for it and there were definite advantages or needs for communication that it could satisfy.

PATRICK: Well, that sounds very high and lofty but it seems to me that if you insist on COBOL before you will consider a machine from a given manufacturer that this kind of implies that you won't get COMTRAN.

GORDON: Or FACT.

PATRICK: Well, if we're looking to the future it just doesn't seem that with the money press that is coming on the manufacturers that they can afford to do two of these languages. So if the DOD says, "I want COBOL," or "I want JAZZY," or some other darn language, it looks like we're all going to be using it whether we like it or not. I would dearly like someone to convince me that I'm wrong.

LITTLE: Isn't it true that if you just get enough people implementing COBOL, either you accept it at that point of time as common or as a standard, or what you develop thereafter as common or standard must be compatible with COBOL?

MacKENZIE: Good, bad, or indifferent, I think we ought to be able to rigorously define the language in question.

PATRICK: A la the syntactical chart?

MacKENZIE: Not at all, for the charts are merely visual aids. We should be able to find ways of producing rigorous descriptions of languages — ones that people can read and ones that represent the authority rather than ones which require inferences to the authority based on observation of a machine representation in action.

McCRACKEN: I would tend to agree with that. Suppose you have two languages that do about the same thing. One of them has a lot of effort behind it; that is, a lot of people working on it. The other is what you might call an offshoot of the same thing but just a bit better — but not widely used, and not being heavily worked on by large numbers of people. It's my opinion that the computing world would be better off to settle on the one that is more widely accepted and work within the framework of that one language to improve it, than to push for acceptance of offshoots which may, in some way, be better.

CLIPPINGER: Has someone brought up the fact that X-3.4 has selected three languages to consider for standardization: FORTRAN, ALGOL, and COBOL. X-3.4 has requested from IBM a statement as to their position on this matter and IBM has responded favorably. They have done some work in providing as a starting point an initial draft of a form of FORTRAN which could be used as a kick-off. X-3.4 is about to set up a group to go to work on FORTRAN along with the other two.

GALLER: ALGOL and FORTRAN are quite similar in that they both cover what we have loosely called the scientific area. You really don't need both of them. So if there's a move to make them both standard, it would be interesting to see why. And here it seems to me that FORTRAN is being looked upon as a standard, not because it's so wonderful but because it is already so common. ALGOL, on the other hand, doesn't have much claim to commonness yet but it does have something to offer. I don't think there is a real tremendous need for two languages to be standard but it is interesting that both of them are being looked at.

McCRACKEN: But then why look at three of them? I'm not referring to COBOL but, for example, why do we also need MAD?

GALLER: MAD is a language that is not used very much yet. It seems to me that we cannot decide that any one of the languages we have now is *it*.

Look back to when FORTRANSIT was distributed. The covering letter for it said something like "everyone recognizes that FORTRAN is the language that we're going to use from here on out so you'd better get on the bandwagon." Simply and historically we know that that was not true. I don't even want it to be true. There's a difference between standardization at a point in time where you could get great benefits from it, and abandoning the search for something better. We should always be prepared to go on to something better. We may choose to develop a language that is different, and the fact that various people are using MAD would seem to indicate that there is *something* there. There are people who are choosing not to take it on because of the cost of deviating from FORTRAN and ALGOL and so on. We are very happy that people are looking at it, but some people who have looked at it have rejected it, and that's fine.

McCRACKEN: The question was whether in the process of looking at the new languages we have to create a new name at the same time. Can't we work within the framework of what is widely accepted? Can't we try to improve what we already have or do we have to go off to the side and create something new that is not compatible?

GALLER: Sometimes to make progress we have to go off to the side and start fresh.

PATRICK: Bernie is making use of the unique position he is in, where he can experiment without severly changing the course of the field unless he just happens to find something very good. Perhaps IBM cannot do this and the Department of Defense cannot do this. If either IBM or the DOD (and they are the two powers) do this, their experiments will have a profound effect on the field.

ARMER: God help us if our universities can't experiment.

McCRACKEN: That isn't what I said. They've got to experiment.

ARMER: What would you have them do differently? Somehow I feel that you're picking on Bernie.

McCRACKEN: No. Bernie is a friend. But if they do find something good by experimenting in the universities what do they do about it? Do they go out and say, "Let's everybody buy my language now? It just happens to be called MAD and it's not ALGOL." Or should they get back into the ALGOL effort, having found a good thing? *Of course* they should experiment.

ARMER: But it seems to me that their experiment has to involve a fairly large number of users. They can't just develop it and then not use it.

GALLER: We've got 2,000 people on campus just writing programs. We use it.

ARMER: Yes, but it seems to me McCracken was arguing that MAD is not really going to change the world and now you should put it on the shelf and do something in ALGOL. Is that what you were saying Dan?

McCRACKEN: No, I was asking a question. The question is, should they now go out as salesmen for MAD? Alternatively, should they go to the ALGOL people and sell them on the improved features that they have developed in MAD?

PATRICK: Selling something to the Secret Society is a difficult job.

McCRACKEN: There is now a well established committee of IFIPS for the maintenance of ALGOL.

GRUENBERGER: One of the things that impresses me about the attempt to have a language become common is the very short half-life of such a venture. FORTRAN is a fine example. From time to time FORTRAN gets to be a little bit frozen. A new tape comes out from IBM; we all get a copy of the tape and we all have the same language for about five minutes. Then everybody goes off in different directions again. Our experience here at RAND is probably quite typical. We diverged from FORTRAN-17 (or whatever is currently kicking around) about two years ago. And we're miles removed from what everyone else uses. It's no longer common in any sense except that a certain amount of training can be transferred from person to person. But our FORTRAN codes can't run anywhere else.

GORDON: Excuse me, Fred, but I'd like to ask you to refrain from using the word "everybody" to mean 7090 users. There is more to FORTRAN than 7090 use. However, if anything, that makes the situation worse than you just said.

GRUENBERGER: Of course, the same thing applies to FORTRAN among the 1604 users and everybody else. These people get a package and they're common with everyone else for a few milliseconds and then they think of a goody to put in or a patch to make and they're not common any more.

CLIPPINGER: I'd like to say a little bit about the process of standardization. We in X-3.4 who are trying to do something about it are not all convinced that we are going to succeed. Standardizing a programming language is an extremely complex business. Fred's point about half-life I think is quite pertinent. I don't think we are going to learn how to do it fast enough to achieve any results on any of the languages we've selected. And yet, we're hard at work trying to move in that direction. I feel a need for a von Neumann to get to the heart of the problem. We need a definition of the syntax of a programming language. The right kind of brilliant man could lay a firm foundation there which might enable us to mechanize on a computer the determination of whether you have a properly specified language. He could provide us with tools to enable us to determine whether we're ready to consider extensions; to give us further information to enable us to resolve ambiguities. Without some sort of firm foundation I think we're just going to spin our wheels and the languages are going to move faster than we can move to catch up to them.

PHILLIPS: I'd like to comment on what Clippinger has just said and take exception to it. He made essentially the same statement that we heard just now when he was a practicing member of the COBOL effort. He didn't think then that COBOL would ever get off the ground.

CLIPPINGER: Well, it is off the ground but it's different for every implementation and we have no experience as yet in its use. I would guess that when the 1410 users discover what it is they have, they will be so sick that there will be a severe reaction. We'll get the same reactions from other small computer users (it has nothing to do with the 1410). The fact is that the language is so complex that it is extremely difficult to put it on any computer with a small memory.

GORDON: Some users of big machines are going to be sick too.

CLIPPINGER: There's at least some hope here. With a larger machine you have more degrees of freedom. You can improve what you first get (although it won't be good enough) and eventually converge toward something which is so good that the users will want to use it. But we have a lot of learning to do before we know where we stand or can evaluate what COBOL is really worth.

PHILLIPS: I'm not suggesting that COBOL has arrived, in any sense. I am suggesting, Dick, that you are surprised at our rate of progress compared to what you thought it would be two years ago.

CLIPPINGER: That's probably true. I think everyone here recognized the weight of that DOD mallet.

PATRICK: I'm not sure that's true and that may be why we're here today. I understand that the DOD says:

> "For an electronic digital computer to be installed on or after December 31, 1962, computers will be selected from those for which a COBOL compiler is available, compatible with the equipment delivery, unless it has been determined that the intended use of a particular computer would not benefit from the availability of a COBOL compiler."

I don't think you'll find a military officer in his right mind who would buck that. I don't think these boys have guts enough to stand up to you and say, "Charlie, my compile time is gigantic and my throughput is way down. I can't stand it." I don't think they'll speak to you the way I will.

PHILLIPS: Remember I don't work for the government any more.

CUNNINGHAM: I don't agree on Bob's comment. I'd like to invite you to come and sit in my office some day and listen to the way our data processing people sound off about things they don't like. And this is certainly one of them.

PATRICK: But the big push was toward the COBOL effort. The secondary areas will take care of themselves because the directive is published and there isn't a man alive who is man enough to say "I made a mistake" and rescind it. I think this is the legacy we received from Charlie Phillips.

McCRACKEN: Are you saying that COBOL is a big mistake? There are also people who have gone the first round and are happy with it, so far.

PATRICK: I am saying that the requiring of COBOL *in its present state of development* could be a hell of a big mistake.

CUNNINGHAM: Well let me come back to that point. You read us an excerpt from a defense directive. That directive did not say that it required COBOL. It said that equipment to be furnished had to provide a compiler for COBOL.

PATRICK: Do you think if they were bringing out the H-800 again they would do FACT in the light of that development?

CUNNINGHAM: I don't know.

ARMER: In the light of that development look what's happened to COMTRAN. COMTRAN and FACT are both dead.

PATRICK: Yes, in the light of that one statement. Now is that good?

CUNNINGHAM: From our standpoint in the Department of Defense, yes, it is good. Not that either or both are dead but that we only have one language. It would be hard to estimate the cost to us of not having the ability to select updated equipment due to conversion costs. Common or standard languages fit the pattern developing throughout the Department of Defense in which the department will be dictating standard practices for use throughout the department.

PATRICK: Don't you think that you're shoeing the wrong end of the horse? It looks as though your problem occurs. . .

CUNNINGHAM: Our problem has occurred already.

PATRICK: The problem gets refreshed every day every time you order a computer. Thirty-six months from now you probably won't have anything installed that you have installed now. In other words you'll get a fresh start in 36 months.

CUNNINGHAM: Gee, I wish you'd talk to the Government Accounting Office. But, have you considered the conversion problem in an installation where there are between 500-600 routines (all dynamic) with over a million instructions and the problems and cost of changing these programs and equipment?

PATRICK: These are facts to the computing field. How many UNIVAC-I's do you still have running and don't you wish you could get rid of them? UNIVAC-I's and 704's are about the only two machines that are still around that have been here more than 40 months.

CUNNINGHAM: There are still a lot of 705's, 650's, etc. that have been around a long time too.

PATRICK: Yes, but they're probably now mod 3's which are incompatible with the mod 2's and the mod 1's. I think your main motivation for a common language stems from the fact that you want to order one computer from every manufacturer and set them side by side. This is bound to give you trouble. This is like trying to drive a stagecoach pulled by a zebra, a kangaroo and a mule, all in harness together and we're going to go to San Francisco. It doesn't seem to me that the solution of your basic problem is through a common language.

LITTLE: I think there's a point here, though. You're expecting your common language to cut across machine types and classes. I know from experience in the Air Force that the different depots, for example, have different kinds of machines which are expected to implement the same inventory control system. What happens then is that you are not only forced to take the same problem and implement it for different depots, but you are then forced to make the language compatible across the machines too. This is a big job, the way things sit now.

GORDON: There are only three problems the way I see it. We're trying to standardize the wrong thing. We're doing it the wrong way. And we're doing it at the wrong time.

Let's take the way we're doing it. Traditionally, standards become recognized as such by virtue of the fact that they become standard through usage. After long use people look around and say "Gee, this is a good thing; let's give it the label 'standard'." What we're trying to do now is sit back and pull out of the blue sky something brand new, that we will call a standard, which will wipe out everything that went before. This is why I say we're doing it the wrong way.

Secondly, we're trying to standardize the wrong thing, which is what Armer said before. There are two different levels at which you can standardize. Galler mentioned SAP before, which was a local standard, a standard among a small group of one-machine users. This was a standard; it just wasn't a world-wide standard. You had standardization within an installation, within a group of machine users, within a single manufacturer, etc. You had geographic levels of standardization.

Another kind of standardization lies in things like character sets, tape formats, and module standardization.

**BROMBERG:** We *are* looking at character code standardization. You can't put on a set of blinders and close your eyes to all the progress that is current because it doesn't happen to start at the point that you have defined as the beginning.

**GORDON:** Unhampered by knowledge, we're going ahead and laying down some firm, rigid things that are likely to be a bit of a problem after a while.

**BROMBERG:** The way that we're approaching – the matter of commonness of a language – is perhaps untimely. I would maintain that there are two reasons for having a common language. The first is that it provides an effective means for the specification of problem solution. You can't really say anything against this. Everyone is looking for a good usable vehicle for expressing problem solution.

The second is this ability (that DOD is waving) of being able to take a given problem specification in this common language and run it on another machine.

**GORDON:** You talk about the ability to communicate effectively with the machine. You need a good problem stating language. For all we know, FACT may be the best problem stating language yet devised. But we are not going to have a chance to find out because it is not going to get sufficient usage. It will get *some* usage (as did Commercial Translator) but the blunt fact of the matter is that we are never going to have any really effective usage because of the fact that the standard preceded the evolution.

**BROMBERG:** Barry, in the creation of any one of these languages (the creation of the actual specifications themselves) there is never any of the wisdom of experience and use.

**GORDON:** That is correct. Consequently, some of them don't quite work out. I agree. You have to have a chance to find out whether they will work.

**BROMBERG:** That's what we are doing. Consequently, by finding out what their so-called standard does, you have every opportunity in the world to come up to the language maintenance committee and say "Look, this damn thing doesn't work. We have a function in Commercial Translator that is much better. Why not consider it?"

**OPLER:** Sometime in the late summer of 1961, the SHARE COBOL committee solicited improvements in COBOL that people might want. Someone offered the following suggestion: Let's modify the COBOL statement that says "Add A and B and C to D" to "Add A and B and C to D and E;" that is, to permit the result of an addition to be put in more than one place. I think everyone agreed that this was a sensible suggestion. When the SHARE COBOL committee agreed on it and the SHARE Executive Committee approved it, the suggestion was forwarded to the appropriate COBOL committee. To the best of my knowledge, this suggestion (which was made in August of 1961) will not see the light of day in a COBOL compiler until late 1963 or 1964 if approved by everybody down the line. Thus, a simple, logical suggestion takes about three years to appear in the processors that people use.

**PATRICK:** Bromberg said a moment ago that there are two reasons for standardizing. One is to have a language to communicate with and the other is that the DOD wanted to put the same problem on two machines. It seems that the first could be handled by a standard language for communication purposes and the second could be handled by not putting two dissimilar machines back to back.

**GORDON:** The first requirement can be handled by any one of a number of languages or any one of a group of languages. If the impetus is to get a good programming language there is no requirement at all that it be the same one that is being used down the block.

**BROMBERG:** I disagree. Suppose you grant that any one of these unknown languages is adequate for the job. You have training costs to begin with. If you're using two different languages you have to do twice as much training as though you use one language.

**PATRICK:** Only when I interchange people.

**McCRACKEN:** May I get in here? I heard several times the statement made that only the DOD has the problem; that they are the only ones that have different machines that have to talk to each other. This is not true. I know of a certain user who had a machine, let's call it Machine A. They needed a bigger machine, so they surveyed the market. One of the available machines was Machine B, made by the same manufacturer. Machine B would accept the programs of Machine A. Their evaluation of the machines involved indicated that they did not want Machine B. It was not the best machine, for the price, for their job. They ordered Machine B anyway because they figured they couldn't afford the reprogramming cost, which, by their estimate, was about half a million dollars. So they went ahead and got the machine that they didn't want just because of this non-common language business. It isn't just DOD.

**PATRICK:** I don't see where that has any bearing at all.

**McCRACKEN:** If they had been in COBOL in the first place they wouldn't have had this problem.

**PATRICK:** If they had been in COBOL in the first place – present state-of-the-art, now – they might have been spending twice as much to get their programs in.

**GORDON:** And also they might have needed the added capacity of Machine B sooner.

**GRUENBERGER:** I think all that McCracken was pointing out was the rebuttal of the statement that only DOD has the problem. We have seen examples lately of other people having the problem; Westinghouse, for example. They have recently stated that they are going to use COBOL company-wide. At least in Westinghouse the guy in charge of a little installation may pick a machine with a fair amount of freedom.

**CUNNINGHAM:** In the first place the COBOL effort is made up of a lot of different people besides Defense, so a lot of other people must at least think they have the same problem. I'm not speaking only of manufacturers. There are a variety of users in the COBOL effort who must recognize a need and be interested in getting on top of it.

Secondly, I don't think that you'll find, even in the Defense Department, two different machines in the same installation back to back doing the same job. You won't find the 501 and the 705 working back to back. What you will find is the 501 in one place and the 705 in another place.

**PATRICK:** What is at David Taylor?

**CUNNINGHAM:** I don't know.

**PATRICK:** It seems to me David Taylor has a LARC, a UNIVAC, etc.

**CUNNINGHAM:** I presume that the variety of work dictates a variety of equipment. I'm saying they're not doing the same jobs back to back.

**ARMER:** How about logical back to back, where you have one machine at one site doing, say, inventory control and a different machine at a different site doing the same inventory control problem?

**LITTLE:** You can still be hurt by having different machines in the same installation because you do not have the freedom to pick up the load of one from the other.

**CUNNINGHAM:** Agreed we've just gone through the same analysis in the Air Force that McCracken was talking about; the one that says that we didn't want to pick up a new machine because of the programming investment involved. A machine that we might really want we would never get to because we couldn't afford the reprogramming costs (both dollar and time). The question we face is "How much longer can we afford to pay what might be four times the cost of the job to wait to get to the optimum position?" So we decided that the optimum for the moment is this swap we went through when we changed 705's for 7080's.

**LITTLE:** There's a very good point here. Is a language really going to solve this problem? Do we really design languages for use by what we might call professional programmers or are we designing them for use by some subhuman species in order to get around training and having good programmers? Is a language ever going to be an effective substitute for really good people?

**McCRACKEN:** It won't be perfect but it will be better than what we have now.

**CLIPPINGER:** One of the reasons that users would like to get a good programming language is the freedom to change from one machine to the other. McCracken pointed out that the switch is pretty tough when you are in assembly language. Now there's a fact here which seems pretty obvious but it ought to be stated: if problems were formulated in an English type language (COBOL, or FACT, or Commercial Translator), the statement of the problem contains all the information necessary for the computer and therefore contains most of its own documentation.

If you want to switch that same problem from COBOL to FACT or FACT to Commercial Translator, or even from COBOL A to COBOL B you have about the same order of magnitude of work, which is ten times less than changing from 705 Autocoder to some other assembly language. So there seems to be an intrinsic value in the English type language. This advantage could even be gained in a language that is not English narrative as long as it's some symbolic language that is suitable for stating problems. The point is that compilation produces documentation. We're all aware that the big problem in moving a computer solution from one machine to another when you are dealing in assembly language (we're talking about very large problems) is that the people involved in producing that checked-out code are no longer around.

**GALLER:** You may have the same problem with English type languages too.

**PATRICK:** Yes, it's not clear that you've changed it any. If you have poor management and you haven't kept your source decks up to date you have the same problem.

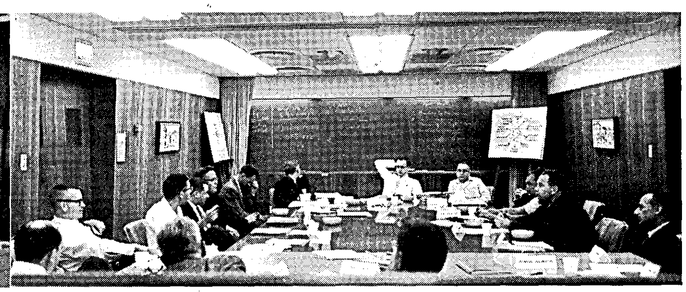
**CLIPPINGER:** With FACT, for example, you compile and you don't patch because it's too difficult to patch. You simply recompile when you redo the problem. Therefore the final version which is running is supported by a document. And you know exactly what that program does.

**BROMBERG:** Jack raised the question "For whom is a language designed?" A language like COBOL is designed for two kinds of people. First, it is designed for the implementers – the guys who are actually going to interpret the language specifications for a particular computing device. Second, it is designed for the salesman, so that he has something to go out and talk about. It is not really designed for the user, per se. The secret for the effective utilization of all these languages is the recognition by the user that he cannot exist in the common programming language cosmology as a clod.

**McCRACKEN:** I am strongly impressed by Howard's remarks. I ran a little program to prove that COBOL can't be used by clods. I ran it on the 1105 at the Air Force Logistics Command. There were two versions, both of which had exactly the same procedure division. They produced exactly the same results. The only difference between them was in the data division, and the change was minor.

We compiled the first one and the running time was about eight seconds. The second one, which was also a legitimate COBOL program (a slightly modified data division, the same procedure division and the same results) took *80* seconds to run. I think this goes to show that you can't be a clod. You do have to know something about machines. It also goes to show, I think, that the efficiency of the object program is not just a function of the language, or of the compiler.

I think it also shows that you don't have to know *very much* about the machine. I had never worked with an 1105 before. All I used in designing this horrible example was knowledge of how alphabetic information is stored in a binary machine.



**GORDON:** I'd like to point out that the point Dan *made* was not the one he *claimed* to make. It seems to me that he has demonstrated that COBOL *can* be used by clods, but with disastrous results.

**OPLER:** I was talking to an installation manager who uses COBOL. He decidede to adopt COBOL because he figured that his problems divided into two classes: those that would fit into COBOL and those that were rather unconventional. For the latter, his best programmers would write in assembly language. They went ahead on this basis, and found to their consternation that the following situation had developed: the programmers who were trained in

COBOL (but who did not know much about the machine), were writing programs in COBOL. After compilation, the object programs would not run. They would then call the other group of programmers to patch the COBOL object program. Eventually, they found that they were using both teams full time on COBOL. Maybe this will be the final division of people in such installations. Groups A and B will sit on top of each other to help each other.

BROMBERG: On the other hand, there are many cases where people are using COBOL and using it very well and they have never compiled a COBOL program.

GORDON: That's probably the best way.

ARMERDING: No one yet has said anything about compile times.

PATRICK: We're talking only about the language at the moment.

ARMERDING: And I think that's one of the main troubles with people who design languages. They seem to forget about things like the compile times and execution times.

When they finally see the result running on their machines they throw up their hands in horror. As Clippinger said, when the 1410 users see what COBOL is going to do for them and to them they're going to be horrified. You can't really predict this. You can't say what the effect of COBOL is going to be on a machine that hasn't even been announced yet.

LITTLE: I think I'd feel a lot better if I believed that some of the users would have enough sense to scream. Some of them are going to put a high level language on a machine and not know any better.

ARMERDING: Yes, that's also true.

PATRICK: Barry, do you hear any screams?

TALMADGE: You'll hear some screams from users of FORTRAN on the 7070, but there are an awful lot of 7070 users of FORTRAN who don't scream; and this worries me more.

GORDON: I take exception to your statement, George, that language designers do not consider compile times. Some of them do; that's a fact.

ARMERDING: They can't if the machine hasn't even been designed yet.

GORDON: Sometimes you're lucky if you're working on a language for an old machine.

GRUENBERGER: You mean like one of these days we're going to get FORTRAN for the 7090, huh?

GORDON: Can we delete Fred's remark?

Basically language designers will take into account (as best they can) things like compile times. But there are many other things they have to take into account. For example, when you design a language, some poor slob is going to be stuck with the problem of teaching it to people. There are problems of implementation schedules to be met. You aren't going to design something that you can't implement for the next thirty years; not deliberately, anyway.

ARMERDING: I agree. For example, compile time is not nearly as important as efficiency of the object code. You are probably willing to buy a certain amount of long compilation if you'll get out an efficient code at the other end.

ARMER: Sometimes you have one goal in that equation of Patrick's, sometimes you have another.

In the first version of FORTRAN they devoted so much attention to object code efficiency that the compile times were way high. That was on the 704. Now they seem to be going the other way, on the 7090.

TALMADGE: I don't think it's true that you need accept long compile times in order to get good object code efficiency. Very often, both in the scientific and commercial field, the life of a job is only one or two runs. In this situation, it is most important to be able to connect small parts of a program rapidly; that is, compile efficiency is much more important than run efficiency. This point was kept in mind in designing Commercial Translator.

As to the comment that a language can force a long compile time, I believe it to be true to some extent, but not to the extent many people believe. Too often the processor designer loses sight of the fact that his system is going to be run on a particular machine, in a particular environment. It is more important to design an efficient, simple, *total* operating system than to turn out a super efficient object code rather than just and average object code.

PATRICK: If I make a compiler efficient in the way you just mentioned I make it machine-dependent again.

TALMADGE: You're talking about a processor now.

PATRICK: If I take an easy language and restrict all my symbols to six characters or less so that they'll fit your machine then I can't run Clippinger's programs which were designed for his 48-bit machine. And it makes your compiler a great deal faster: that is machine-dependence.

BROMBERG: On the other hand, there are some of us who believe that one of the functions of a compiler is to do error-checking. We can certainly design a fast compiler (call it a User Beware Compiler) that lets you effectively write anything that you'd like. And you can thereby increase compile times by a factor of 10 and still keep it within the area of machine independence.

TALMADGE: If you're talking about computer independence then you're talking about the language description. But you can't talk about computer independence when you talk about the processor which translates from that particular language to produce an object code on the machine. Then one cannot be independent, any more than one can be machine-independent when writing a data description for a commercial problem. Dan's example is a good one: when one writes the data description to take advantage of the binary machine then the object program is much more efficient.

PATRICK: The point I was getting at was that if you are implementing a processor you must be doing it for a specific machine. There is a threshold point at which it makes a great deal of difference whether you put restrictions on the language or not. It's the same language; the syntax is the same, the verbs are the same, it has precisely the same meaning, but if I can put some restrictions on the use of that language by the source programmer it will make a lot of difference in the performance of the compiler.

TALMADGE: I agree with you on that, Bob. There are certain things which are critical as far as compilation times are concerned and which buy practically nothing in language facility. The point has been made many times that one shouldn't try to be too general in a given language because the cost of generality is frequently high in terms of compile time.

The point I'd like to make, however, is that I don't think this is true until one gets to a fairly advanced stage in the language development.

PATRICK: But they're not machine-independent anymore.

TALMADGE: In what sense?

PATRICK: Well, take the example I used before. If someone has used 30 character names from the COBOL made for Clippinger's machine, I can't compile them even though they are supposedly written in the same COBOL.

DOBRUSKY: It seems to me that this discussion right here borders on the difference between standard and common. The fact that they're standard means that both of you can handle them in one way and another. If you write them generally machine-independent, (acceptable within the constraints of the grammar and syntax of the language) you're going to pay for it on any machine. This is typical of every compiler and every language. However, I am sure that there is a standard subset of COBOL that both of you could use; the same is true of Commercial Translator.

McCRACKEN: I'd like to ask the assembled implementers a question. I talked to the man in charge of software construction for a certain machine. He has 50 people working for him (give or take one or two). I asked him what kind of people these are. He said that he had six group leaders who had been in the programming business for three or four years. For the other 44 the average experience is under one year. At the time I was talking to him this group had just finished, say, COBOL for some machine. I wonder how much of our trouble these days is simply due to the fact that we haven't grown up yet.

"Look," I asked this guy, "would you be better off with three people?" And he said, "Of course I would, but I can't get them." He said, "There is nothing I can do but let them grow up."

LITTLE: Or do it with the six group leaders. How about that?

ARMER: I'd be interested in asking Cheatham how many man-years you estimate the task you mentioned (building compilers for the Burroughs D-825, Control Data 1604, and the IBM 7090) will take and what kind of people are going to be doing it.

CHEATHAM: Well, for one thing, I never put more than two or three people on one compiler.

GRUENBERGER: What kind of people?

CHEATHAM: The average man, for example, has six-eight years experience.

LITTLE: You're talking about implementing compilers. I'd like to ask Jerry their experience in implementing systems.

KOORY: What do you mean by systems?

PATRICK: Come on, you can't avoid the question like that.

KOORY: No, but it was worth a try.

LITTLE: I'm talking about doing a job for a customer as opposed to doing a compiler.

KOORY: You're asking, for example, how many people do we apply to building a Damage Assessment Model for the DOD?

LITTLE: Yes.

KOORY: Well, of course, this is a function primarily of what we consider to be the requirements of the system. We have just finished a Damage Assessment Model which will operate on the 1604. It was written in JOVIAL. For the actual writing and check-out we used on the order of 25 programmers, as I recall.

LITTLE: Do you have any idea of the level of these 25 programmers?

GORDON: Before or after?

PATRICK: And which way does it go?

KOORY: I would say on the average between one and two years. We are fortunate enough to have four or five of them who each had four or five years of experience. We have had a fair enough number of brand new folks, you might say brought in off the street (that is, just out of college). We had to train this latter group.

BROMBERG: I don't know if I'm speaking to McCracken's quesion or not, but is not this vast number of inexperienced programmers hurting us? I have concluded that it is inevitable that we are always going to have such numbers of inexperienced programmers. When we deal with language implementation for new computers I think the first thing we should do is turn out a compiler and then perhaps a year later turn out a good assembler. I reached this conclusion just because of this problem that Dan brought up. Those customers who are new to the machine are quite similar to those implementers who are new to the business and to that particular machine. Only through use and experience is the customer going to get down to the measure of the efficient use of the machine. At that time they should be given an assembler which allows them to make this efficient use.

McCRACKEN: Leave us just pray that the customers just don't get so furious at the compiler they receive by that process that they give up on COBOL.

PATRICK: Yes, I seem to remember the time when a person would polish like crazy if he was doing utility work, because you would say to yourself, "Gee, if I can save three instructions here, that will be three instructions off everyone's use when they go to use this particular sine routine." What happened to that philosophy? Seems like we're kind of galloping into second shift rental with the philosophy you were just mentioning. We're saying, "It only takes an extra 10,000 instructions so we get into second shift rental in the third month and who cares?"

OPLER: When you write an application and you make a slight error (e.g., your loop is one instruction too long) you have hurt that application *only*. When you make the same sort of error in writing a compiler everyone who uses that compiler gets hurt. I feel very strongly about this. When you're writing a compiler, you must have your best people on it exerting their best efforts to squeeze everything out of it they can.

GORDON: This sort of thing should make very happy those people who argue that you can put up with a little inefficiency. We hear a lot from them. What Bromberg said is a good answer to these guys who say, "We want it yesterday and we're willing to put up with a little inefficiency to get it." Howard's approach would give it to them.

MacKENZIE: There's no real basis for comparing the efforts or experience levels. I don't think there is any real argument on this because you have to achieve design control one way or the other. You had best do this with a small experienced group.

CLIPPINGER: What is the subject of this conference?

PATRICK: The pros and cons of common languages. And what we're after is enlightenment.

CLIPPINGER: I'm not sure we've been talking about that subject.

PATRICK: It might be appropriate to review what we've covered this morning.

We started out talking about visceral feelings. These were things that appealed to us, like why common languages were good and why they were bad. These are God and Mother categories on a high plane. Along the way we uncovered just a few facts. We have made some statements that such languages are not completely machine-independent. As of the present state-of-the-art, we don't seem to know exactly how to do this. In the implementation stage you made them, perhaps, a little more machine-dependent in order to get some efficiency. We pointed out that you pay for generality. We noted that it was great to have a language that you can translate across all machines and have it efficient on every machine. I think it is our consensus that we don't quite know how to do this just yet.

The last topic we were on, I think, was whether it was better to have a high quality small staff or a young mob. With all deference to Bromberg, I think he was talking about a pyramid of a staff, with a genius at the top and levels of priests and sub-priests down to the machine clerks at the bottom. It's rather difficult to administer such a staff because the design process extends all the way down to the key-punch stage. ■

# BENCHMARKS IN ARTIFICIAL INTELLIGENCE

**for negativists only**

by FRED GRUENBERGER, The RAND Corp., Santa Monica, Calif.

Various writers, in discussing the problems of simulating intelligence, have bumped into the problem of furnishing examples of thinking. Consider this quotation from Armer [1]:

> "Many of the negativists seem to say that the only evidence of machine intelligence they will accept is an achievement . . . seldom achieved by man. For example, they belittle efforts at musical composition by machine because the present output compares miserably with that of Mozart or Chopin. How many *men* can produce music that compares favorably? The ultimate argument of this kind occurred at a recent meeting in England during which a discussant stated that he would not accept the fact that machines could think until one proved the famous conjecture of Fermat, better known as Fermat's last theorem. From his logic one can also conclude that, to this date, no man has been capable of thinking, since the theorem remains unproven."

or this one from Hamming [2]:

> "Very frequently 'thinking' is defined to be what Newton did when he discovered gravitation. By this definition most of us cannot think! As an exercise, I suggest you try framing a test that is the least, or close to the least, which you will accept as demonstrating that a machine can think. I have been unable to devise one that would suit myself, let alone others, and have tentatively considered the hypothesis that 'thinking' is not measured by what is produced, but rather is a property of the way something is done."

Armer, in the same paper cited above, advanced the same suggestion:

> "It would be useful to have at hand some milestones for the future, to assign a metric . . . Turing supplied one such milestone but additional ones are needed. To this end we need a clearly defined task which is at present the exclusive domain of humans (and therefore incontestably 'thinking') but which we can hope will eventually yield to accomplishment by machines."

It seems clear to me that a machine need not surpass all men, nor even exceed the work of the rare brilliant man. It need not even be original. The goal (at least a first goal) is to duplicate what a man does when he claims he is thinking. This nearly always reduces to some decision process which can discriminate between two or more possible choices. What we need are some well-defined, low-level decision processes that are common to most men. To be useful as benchmarks, these tasks must be outside the scope of computer solution now. It is the purpose of this paper to attempt to perform the exercise Hamming suggests.

1. Let me begin with a fairly trivial example and one, moreover, that is well within present capability to implement. It is illustrative only, of things readily done by humans which are not easy to do by machine.

When I talk on the telephone, having handy a pad of ruled paper and a pencil, I habitually doodle. The doodling, in this case, is formal and well defined; it proceeds rapidly and at a very low mental level. An example appears in Figure 1. During an extended long distance call, I can develop a full page of it.

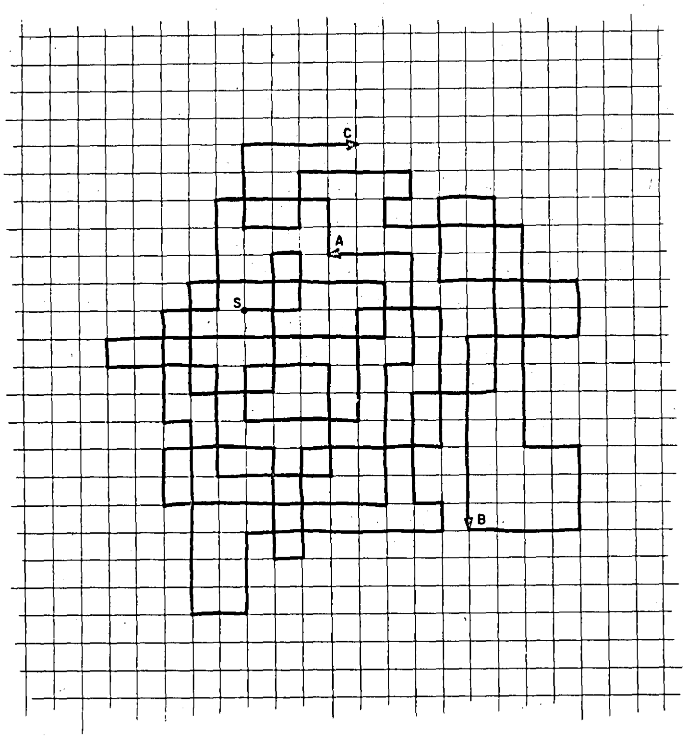
The characteristics are these: starting at S, the doodle lines follow the printed lines of the cross-section paper without overlap. The pattern's mass tends to centralize around S (i.e., no fair going away). The points marked A, B, and C represent places where decisions were made. Notice, for example, that I could not turn south at A, since there would be no way to "get out." Similarly, I could not turn west at B, or south at C. Yet I can create this sort of nonsense effortlessly, and apparently endlessly. Can we program a computer to produce such patterns? Unquestionably, the answer is yes, but it hasn't yet been done, and it's not easy. But remember, we seek tasks that any normal human can do that (presently) computers cannot. The example here may be trivial, but it illustrates the principle. (Should the reader decide that it is too trivial, he is invited to flowchart the decision logic.)

2. Example 1 illustrated a situation in logical decision making. The classic example in this category is game playing; specifically chess. The problems and goals involved in game playing by computers are extensively discussed in the literature [3, 4, 5,]. Since, at the present time, we have no computer program that can rank as a chess master, there is a clear benchmark involved. Those concerned (on the negative side) with the question of artificial intelligence should prepare themselves now to take a stand: will the eventual existence of a chess master program cause them to concede success?

We may have in this example something more than a simple benchmark. Consider the following remarks by John Williams [6]:

> "It seems inescapable to me that the brain of man, like that of other vertebrates, is an item, of random design, to meet one basic purpose: survival. There is some doubt these days whether it will in fact meet this criterion, but there is no reason to suppose that it is well designed to perform intellectual man's proudest function: namely, to think. The fact that it has outthought things like saber-toothed tigers is no evidence that it is particularly apt for abstract thinking. It may be argued that I dismiss the survival-tested mechanism too quickly, and perhaps I do. The question arises when we ponder how to motivate advanced machines. I am confident we will discover means to do so, and I will be astonished if we are reduced to threatening them with tigers."

FIG. 1 — DOODLING




In the context of the chess problem, these sentiments have great significance: we may be due for a big surprise. It is conceivable that man does not know how to play chess well *at all;* our total experience can prove only that some men play better than others. Thus, we have here two benchmarks. The first is the computer program that plays better than any man. The second is, possibly, a program that plays better than any man *could.* The bland assumption that, whatever thinking is, man does it superbly may turn out to be fallacious. We may come to regard man's ability to create original ideas or to make decisions properly as rudimentary.

3. Another currently classic problem in logic and decision making is that of machine translation of languages (for which most people immediately substitute Russian-to-English translation). (A similar, but much simpler, example is the translation of ILLIAC computer code to 7090 code. In this case we are dealing with two languages, both of which are absolutely precise. And we can't do this job by computer, either.)

For this problem, our benchmark is obvious, though seldom stated. Success will be achieved when a computer can duplicate the work of human bilingual interpreters. We must be able, someday, to feed a machine idiomatic Russian text, taken at random (from a specific field, if you wish); the computer must then produce idiomatic English text having the same meaning; this, in turn, must be fed to the inverse program which will output Russian text closely corresponding to the original. The benchmark I have set up here is stringent indeed. I am not at all sure that many human translators could pass it (note that the human analogy requires at least two independent persons). Hamming [2] has stated the problem this way:

> "Besides designing and using artificial languages for machine-human use, we have also attacked the problem of translating from one natural language to another, such as from Russian to English. Both the great success and the great difficulties left shed a good deal of light on various aspects of language. To me, a comparative outsider, it seems that the major stumbling block centers around the vexing question, 'What did the author mean?' Perhaps this is a real question; and perhaps it is not, but it seems to me to be where progress in machine translation of languages is presently in difficulties."

As with the chess problem, though, it may turn out that there are actually three benchmarks involved here:

(a) We learn to perform reciprocal translations with at least the facility of well-trained humans.

(b) We discover that the nub of the problem is to learn how to say what it is we mean in some better way than the sloppy one we have developed over the centuries.

(c) We then learn how to translate much better than humans now do. Again, I would expect some surprises, as this research continues, that are not flattering to man's old way of doing business.

4. We turn now from problems involving mainly logic to one whose central property is that of information retrieval. Information retrieval is another of those broad fields for which many benchmarks are implied (such as achieving any sort of success whatsoever); let us state a specific problem:

In this library, which is known to contain all English poetry, find for me the author of the work containing the line "... and that simple arch, which bridges roaring rapids . . ."

This states the worst case of one aspect of IR, since I just invented the quotation. (So its author is now known, but I can readily compose another quotation.) The machinery is already at hand to tackle this problem. We need only store all English poetry on tape and search it, word by word, for the desired reference. But that isn't the way thinking humans operate. What's more to the point, that would be an inefficient IR method. The first goal, remember, is to try to duplicate what thinking humans do; later we can attempt to surpass that level.

The IR people are concerned about the problem I have stated. In this, as in many other problems in artificial intelligence, some of the research follows this pattern:

a. Is there any way at all we can do the job?

b. There is, but we don't like it. We are guaranteed a solution, but at horrendous cost.

c. Human beings (in this case English professors and librarians) can do the same job, with fair speed and reliability, and at trivial cost.

d. Well, we'd better investigate how they do it. There is some (presently) mysterious process (call it, say, thinking) at work here. (When we discover the process, of course, it won't be mysterious at all.)

5. Consider a problem in pattern recognition. I frequently answer my office phone and hear the question "Have you got a minute?" The words are almost always in this canonical form (and hence the information content of the words themselves is zero). The speaker assumes that he and I are on familiar terms. The problem is to recognize his voice. I can do this for perhaps fifty voices (out of a much larger population of voices that come to me via telephone). A computer should be able to do this job. The message takes almost exactly one second; not counting the content of the words, it uses about 1000 bits of information expressed in terms of pitch, overtones, timing, etc. It seems reasonable that a computer should be able to analyze a thousand bits of information for patterns that match known samples.

There is a form of this problem that is even worse. For about three voices, I can do a good job of identification given only the one word, "Hi!"

6. Here's another pattern recognition problem. I am told that my department in RAND will hold a buffet-dance Sunday evening, at a country club just off the Pacific Coast Highway, about halfway between Malibu and Ventura. There will be a sign beside the road to indicate the turnoff.

I can find that sign. I have little idea of just what it says, or the style of typography, or the size, or the color, or whether or not it's spelled correctly, or which side of the road it's on. And in the miles of highway that I must search there will be hundreds of other signs to distract me. But I can find that sign.

7. Here is a signature on a check. Authenticate it. This is pattern recognition again, somewhat akin to the problem of the roadside sign but with a bigger payoff. Millions of check signatures must be authenticated every day. The situation is obvious: humans can do the job and (currently) machines can't. What is more precious to my thesis is that humans are well known to botch this job frequently. Our first success at the task may very well surpass the best human endeavor.

What can we conclude from these exercises? Can we expect that when any number of such tasks are successfully performed by a computer that the argument will be won?

Somehow, I doubt it, which makes for an interesting point. The ability to doubt is common to all men, and by itself constitutes an act of thinking. Perhaps it should be added to the list. We will not have moved any closer to an admission of thinking capability on the part of machines; we will only move some set of people across the line between denial and acceptance. To quote Marvin Minsky [7]:

> "Intelligence is a chimera. You regard an action as intelligent until you understand it. In explaining, you explain away. Conversely, provided you do not understand, it is presumably intelligent."

We have already been around the circle many times. In Newton's day, the ability to solve a set of ten simultaneous linear equations with twenty digit precision in ten seconds would probably have been labelled thinking without question. Indeed, many of the tasks performed routinely today by computers would have been regarded at one time as in the realm of thinking, provided the decision had been made before the fact.

What has been suggested here does not conflict with either the criterion of the negativists (quoted earlier), or with Turing's criterion as given in Armer's [1] paper:

> "A. M. Turing was one of the first to expound at length on the question of machines' thinking, in an erudite article published in 1950 [8]. Although he discusses the basic question of whether machines can become capable of thinking, he recognizes the difficulty of defining properly the words 'machine' and 'thinking.' To circumvent this problem, he examines instead the question of a game wherein an interrogator who can communicate with a human and a machine via a Teletype but does not know which is which, is to decide which is the machine."

If a computer can be programmed to compose a symphony, so much the better. Or, if I can't tell the computer from the human on the other end of a Teletype line, then I can rightly conclude that there is intelligence at the other end. I won't feel hurt if it turns out that the intelligence is artificial. In fact, I'll feel elated; I have one or two problems, for a better intelligence than I command, to brood on.

My arguments amount, then, to an appeal to those who are inclined negatively toward the idea of artificial intelligence to take a stand. What would it take to convince you? Remember, you are constrained to select a task that is common to large numbers of people who exhibit natural intelligence. No fair picking a task that less than one per cent of the population can perform.

I have tried to select such tasks. If you don't like my selection, pick your own. But pick it now. ■

## REFERENCES


1. Armer, Paul, *Attitudes Toward Intelligent Machines,* The RAND Corporation, P-2114, September 30, 1960. (This paper contains an extensive bibliography of the whole field of artificial intelligence.)
2. Hamming, R. W., *Intellectual Implications of the Computer Revolution,* Bell Telephone Laboratories Paper.
3. Kister, J., P. Stein, S. Ulam, W. Welden, and M. Wells, "Experiments in Chess," *Journal of the Association for Computing Machinery,* April 1957.
4. Newell, A., J. C. Shaw, and H. A. Simon, "Chess-Playing Programs and the Problem of Complexity," *IBM Journal of Research and Development,* October 1958.
5. Samuel, A. L., "Some Studies in Machine Learning Using The Game of Checkers," *IBM Journal of Research and Development,* Vol. 3, No. 3, July 1959.
6. Williams, J. D., *Toward Intelligent Machines,* The RAND Corporation, P-2170, December 29, 1960.
7. Minsky, M., "Methods of Artificial Intelligence and Heuristic Programming," *Proceedings of the Symposium on the Mechanization of Thought Processes,* NPL, November 1958, Her Majesty's Stationery Office, 1959.
8. Turing, A. M., "Computing Machinery and Intelligence," *MIND,* October 1950. Also available in *The World of Mathematics* Vol. 4, p. 2099.

As the first conceptual paper on an internally programmed computer, the following presentation of Burks, Goldstine and von Neumann's *Preliminary Discussion* appears in two parts, the first of which was published in Datamation's September issue. This is the concluding segment.

To the best of our knowledge, this is the first formal publication of *Preliminary Discussion.* The work was originally accomplished at Princeton's Institute for Advanced Study in June, 1946. It appears here as a condensed version with much of the detailed mathematical support deleted. However, readers may soon reference the complete paper in one of the forthcoming volumes on the writings of von Neumann to be published this Fall by Pergamon Press, N.Y.C.

# PRELIMINARY DISCUSSION OF THE LOGICAL DESIGN OF AN ELECTRONIC COMPUTING INSTRUMENT

**part two**

by ARTHUR W. BURKS, HERMAN H. GOLDSTINE, and JOHN von NEUMANN

5.11 . . . In discussing the multiplicative organs of our machine we must return to a consideration of the types of accumulators. The static accumulator operates as an adder by simultaneously applying static voltages to its two inputs – one for each of the two numbers being added. When steady-state operation is reached the total sum is formed complete with all carries. For such an accumulator the above discussion is substantially complete, except that it should be remarked that such a circuit requires at most 39 rise times to complete a carry. Actually it is possible that the duration of these successive rises is proportional to a lower power of 39 than the first one.

Each stage of a dynamic accumulator consists of a binary counter for registering the digit and a flip-flop for temporary storage of the carry. The counter receives a pulse if a 1 is to be added in at that place; if this causes the counter to go from 1 to 0, a carry has occurred and hence the carry flip-flop will be set. It then remains to perform the carries. Each flip-flop has associated with it a gate, the output of which is connected to the next binary counter to the left. The carry is begun by pulsing all carry gates. Now a carry may produce a carry, so that the process needs to be repeated until all carry flip-flops register 0. This can be detected by means of a circuit involving a sensing tube connected to each carry flip-flop. It was shown in 5.6 (Part 1) that, on the average, five pulse times (flip-flop reaction times) are required for the complete carry. An alternative scheme is to connect a gate tube to each binary counter which will detect whether an incoming carry pulse would produce a carry and will, under this circumstance, pass the incoming carry pulse directly to the next stage. This circuit would require at most 39 rise times for the completion of the carry.

At the present time the development of a static accumulator is being concluded. From preliminary tests it seems that it will add two numbers in about $5\mu$ and will shift right or left in about $1\mu$.

We return now to the multiplication operation. In a static accumulator we order simultaneously an addition of the multiplicand with sign deleted or the sign of the multiplicand and a complete carry and then a shift for each of the 39 steps. In a dynamic accumulator of the second kind just described we order in succession an addition of the multiplicand with sign deleted or the sign of the multiplicand, a complete carry, and a shift for each of the 39 steps. In a dynamic accumulator of the first kind we can avoid losing the time required for completing the carry (in this case an average of 5 pulse times) at each of the 39 steps. We order an addition by the multiplicand with sign deleted or the sign of the multiplicand, then order one pulsing of the carry gates, and finally shift the contents of both the digit counters and the carry flip flops. This process is repeated 39 times. A simple arithmetical analysis which may be carried out in a later report, shows that at each one of these intermediate stages a single carry is adequate, and that a complete set of carries is needed at the end only. We then carry out the complement corrections, still without ever ordering a complete set of carry operations. When all these corrections are completed and after round off, described below, we then order the complete carry mentioned above . . .

. . . 5.13 We might mention at this time a complication which arises when a floating binary point is introduced into the machine. The operation of addition which usually takes at most 1/10 of a multiplication time becomes much longer in a machine with floating binary since one must perform shifts and round-offs as well as additions. It would seem reasonable in this case to place the time of an addition at about ⅓ to ½ of a multiplication. At this rate it is clear that the number of additions in a problem is as important a factor in the total solution time as are the number of multiplications.

5.14 We conclude our discussion of the arithmetic unit with a description of our method for handling the division operation. Before proceeding let us consider the so-called *restoring* and *non-restoring* methods of division. In order to be able to make certain comparisons, we will do this for a general base m = 2, 3, . . . .

Assume for the moment that divisor and dividend are both positive. The ordinary process of division consists of subtracting from the partial remainder (at the very beginning of the process this is, of course, the dividend) the divisor, repeating this until the former becomes smaller than the latter. For any fixed positional value in the quotient in a well-conducted division this need be done at most m-1 times: If, after precisely k = 0, 1, . . . , m-1 repetitions of this step, the partial remainder has indeed become less than the divisor, then the digit k is put in the quotient (at the position under consideration), the partial remainder is shifted one place to the left, and the whole process is repeated for the next position, etc., etc. Note that the above comparison of sizes is only needed at k = 0, 1, . . . , m-2, i.e. before step 1 and after steps 1, . . . , m-2. If the value k = m-1, i.e. the point after step m-1, is at all reached in a well-conducted division, then it may be taken for granted without any test, that the partial remainder has become smaller than the divisor, and the operations on the position under consideration can therefore be concluded. (In the binary system, m = 2, there is thus only one step, and only one comparison of sizes before this step.) In this way this scheme, known as the *restoring* scheme, requires a maximum of m-1 comparisons and utilizes the digits 0, 1, . . . , m-1 in each place in the quotient. The difficulty of this scheme for machine purposes is that usually the only economical method for comparing two numbers as to size is to subtract one from the other. If the partial remainder $r_n$ were less than the dividend d, one would then have to add d back into $r_n$-d in order to restore the remainder. Thus at every stage an unnecessary operation would be performed. A more symmetrical scheme is obtained by *not restoring*. In this method (from here on we need not assume the positivity of divisor and dividend) one compares the signs of $r_n$ and d, if they are of the same sign, the dividend is repeatedly subtracted from the remainder until the signs become opposite; if they are opposite, the dividend is repeatedly added to the remainder until the signs again become like. In this scheme the digits that may occur in a given place in the quotient are evidently $\pm1, \pm2 \ldots, \pm(m-1)$, the positive digits corresponding to substractions and the negative ones to addition of the dividend to the remainder.

Thus we have 2(m-1) digits instead of the usual m digits. In the decimal system this would mean 18 digits instead of 10. This is a redundant notation. The standard form of the quotient must therefore be restored by substracting from the aggregate of its positive digits the aggregate of its negative digits. This requires carry facilities in the place where the quotient is stored.

We propose to store the quotient in the Accumulator Register AR, which has no carry facilities. Hence we could not use this scheme if we were to operate in the decimal system.

The same objection applies to any base m for which the digital representation in question is redundant —i.e. when $2(m-1) > m$. Now $2(m-1) > m$ whenever $m > 2$, but $2(m-1) = m$ for $m = 2$. Hence, with the use of a register which we have so far contemplated, this division scheme is certainly excluded from the start unless the binary system is used.

Let us now investigate the situation in the binary system. We inquire if it is possible to obtain a quasi-quotient by using the non-restoring scheme and by using the digits 1, 0 instead of 1, -1. Or rather we have to ask this question: Does the quasi-quotient bear a simple relationship to the true quotient?

Let us momentarily assume this question can be answered affirmatively and describe the division procedure. We store the divisor initially in the *Accumulator* A, the dividend in the *Selectron Register* SR, and wish to form the quotient in AR. We now either add or subtract the contents of SR into A, according to whether the signs in A and SR are opposite or the same, and insert correspondingly a 0 or 1 in the right-hand place of AR. We then shift both A and AR one place left, with electronic shifters that are parts of these two aggregates.

Multiplication required an ability to shift right in both A and AR. We have now found that division similarly requires an ability to shift left in both A and AR. Hence both organs must be able to shift both ways electronically. Since these abilities have to be present for the implicit needs of multiplication and division, it is just as well to make use of them explicitly in the form of explicit orders. It will, however, turn out to be convenient to arrange some details in the shifts, when they occur explicitly under the control of those orders, differently from when they occur implicitly under the control of a multiplication or a division.

The process described above will have to be repeated as many times as the number of quotient digits that we consider appropriate to produce in this way. This is likely to be 39 or 40.

## 6.0 the control

6.1 It has already been stated that the computer will contain an organ, called the Control, which can automatically execute the orders stored in the Selectrons. Actually, for a reason stated in 6.3, the orders for this computer are less than half as long as a forty binary digit number, and hence the orders are stored in the Selectron memory in pairs.

Let us consider the routine that the Control performs in directing a computation. The Control must know the location in the Selectron memory of the pair of orders to be executed. It must direct the Selectrons to transmit this pair of orders to the Selectron Register and then to itself. It must then direct the execution of the operation specified in the first of the two orders. Among these orders we can immediately describe two major types: an order of the first type begins by causing the transfer of the number, which is stored at a specified memory location, from the Selectrons to the Selectron Register. Next, it causes the arithmetical unit to perform some arithmetical operations on this number (usually in conjunction with another number which is already in the arithmetical unit), and to retain the resulting number in the arithmetical unit. The second type order causes the transfer of the number, which is held in the arithmetical unit, into the Selectron Register, and from there to a specified memory location in the Selectrons. (It may also be that this latter operation will permit a direct transfer from the arithmetical unit into the Selectrons.) An additional type of orders consists of the transfer orders of 3.5. Further orders control the inputs and the outputs of the machine. The process described at the beginning of this paragraph must then be repeated with the second order of the order pair. This entire routine is repeated until the end of the problem.

6.2 It is clear from what has just been stated that the Control must have a means of switching to a specified location in the Selectron memory, for withdrawing both numbers for the computation and pairs of orders. Since the Selectron memory (as tentatively planned) will hold $2^{12} = 4096$ forty-digit words (a word is either a number or a pair of orders), a twelve-digit binary number suffices to identify a memory location. Hence a switching mechanism is required which will, on receiving a twelve-digit binary number, select the corresponding memory location.

The type of circuit we propose to use for this purpose is known as a decoding or many-one function table. It has been developed in various forms independently by J. Rajchman and P. Crawford. It consists of n flip-flops which register an n digit binary number. It also has a maximum of $2^n$ output wires. The flip-flops activate a matrix in which the interconnections between input and output wires are made in such a way that one and only one of $2^n$ output wires is selected (i.e. has a positive voltage applied to it). These interconnections may be established by means of resistors or by means of non-linear elements (such as diodes or rectifiers); all these various methods are under investigation. The Selectron is so designed that four such function table switches are required, each with a three digit entry and eight ($2^3$) outputs. Four sets of eight wires each are brought out of the Selectron for switching purposes, and a particular location is selected by making one wire positive with respect to the remainder. Since all forty Selectrons are switched in parallel, these four sets of wires may be connected directly to the four function table outputs.

6.3 Since most computer operations involve at least one number located in the Selectron memory, it is reasonable to adopt a code in which twelve binary digits of every order are assigned to the specification of a Selectron location. In those orders which do not require a number to be taken out of or into the Selectrons these digit positions will not be used.

Though it has not been definitely decided how many operations will be built into the computer (i.e. how many different orders the Control must be able to understand), it will be seen presently that there will probably be more than $2^5$ but certainly less than $2^6$. For this reason it is feasible to assign 6 binary digits for the order code. It thus turns out that each order must contain eighteen binary digits, the first twelve identifying a memory location and the remaining six specifying an operation. It can now be explained why orders are stored in the memory in pairs. Since the same memory organ is to be used in this computer for both orders and numbers, it is efficient to make the length of each about equivalent. But numbers of eighteen binary digits would not be sufficiently accurate for problems which this machine will solve. Rather, an accuracy of at least $10^{-10}$ or $2^{-33}$ is required. Hence it is preferable to make the numbers long enough to accomodate two orders.

Our numbers will actually have 40 binary digits each. This allows 20 binary digits for each order, i.e. the 12 digits that specfy a memory location, and 8 more digits specifying the nature of the operation (instead of the minimum of 6 referred to above). It is convenient to group these binary digits into *tetrads*, groups of 4 binary digits. Hence a whole word consists of 10 tetrads, a half word or order of 5 tetrads, and of these 3 specify a memory location and the remaining 2 specify the nature of the operation. Outside the machine each tetrad can be expressed by a base 16 digit. (The base 16 digits are best designated by symbols of the 10 decimal digits 0 to 9, and 6 additional symbols, e.g. the letters a to f.) These 16 characters should appear in the typing for and the printing from the machine.

The specification of the nature of the operation that is involved in an order occurs in binary form, so that another many-one or decoding function is required to decode the order. This function table will have six input fiip-flops (the two remaining digits of the order are not needed.) Since there will not be 64 different orders, not all 64 outputs need be provided. **However, it is perhaps worthwhile to connect the outputs corrresponding to unused order possibilities to a checking circuit which will give an indication whenever a code word unintelligible to the control is received in the input flip-flops.**

The function table just described energizes a different output wire for each different code operation. As will be shown later, many of the steps involved in executing different orders overlap. (For example, addition, multiplication, division, and going from the Selectrons to the register all include transferring a number from the Selectrons to the Selectron Register.) For this reason it is perhaps desirable to have an additional set of control wires, each of which is activated by any particular combination of different code digits. These may be obtained by taking the output wires of the many-one function table and using them to operate tubes which will in turn operate a one-many (or coding) function table. Such a function table consists of a matrix, as before, but in this case only one

| | SYMBOLIZATION Complete | Abbreviated | TABLE 1 |
|---|---|---|---|
| 1. | S(x) ⟶ Ac+ | x | Clear Accumulator and add number located at position x in the Selectrons into it. |
| 2. | S(x) ⟶ Ac | x− | Clear Accumulator and subtract number located at position x in the Selectrons into it. |
| 3. | S(x) ⟶ AcM | x M | Clear Accumulator and add absolute value of number located at position x in the Selectrons into it. |
| 4. | S(x) ⟶ Ac M | x −M | Clear Accumulator and subtract absolute value of number located at position x in the Selectrons into it. |
| 5. | S(x) ⟶ Ah+ | x h | Add number located at position x in the Selectrons into the Accumulator. |
| 6. | S(x) ⟶ Ah | x h− | Subtract number located at position x in the Selectrons into the Accumulator. |
| 7. | S(x) ⟶ AhM | x hM | Add absolute value of number located at position x in the Selectrons into the Accumulator. |
| 8. | S(x) ⟶ Ah M | x −hM | Subtract absolute value of number located at position x in the Selectrons into the Accumulator. |
| 9. | S(x) ⟶ R | x R | Clear Register* and add number located at position x in the Selectrons into it. |
| 10. | R ⟶ A | A | Clear Accumulator and shift number held in Register into it. |
| 11. | S(x) XR ⟶ A | x X | Clear Accumulator and multiply the number located at position x in the Selectrons by the number in the Register, placing the left-hand 39 digits of the answer in the Accumulator and the right-hand 39 digits of the answer in the Register. |
| 12. | A÷S(x) ⟶ R | x ÷ | Clear Register and divide the number in the Accumulator by the number located in position x of the Selectrons, leaving the remainder in the Accumulator and placing the quotient in the Register. |
| 13. | Cu ⟶ S(x) | x C | Shift the Control to the left-hand order of the order pair located at position x in the Selectrons. |
| 14. | Cu′ ⟶ S(x) | x C′ | Shift the Control to the right-hand order of the order pair located at position x in the Selectrons. |
| 15. | Cc ⟶ S(x) | x Cc | If the number in the Accumulator is ≥ 0, shift the Control as in Cu ⟶ S(x). |
| 16. | Cc′ ⟶ S(x) | x Cc′ | If the number in the Accumulator is ≥ 0, shift the Control as in Cu′ ⟶ S(x). |
| 17. | At ⟶ S(x) | x S | Transfer the number in the Accumulator to position x in the Selectrons. |
| 18. | Ap ⟶ S(x) | x Sp | Replace the left-hand 12 digits of the left-hand order located at position x in the Selectrons by the left-hand 12 digits in the Accumulator. |
| 19. | Ap′ ⟶ S(x) | x Sp′ | Replace the left-hand 12 digits of the right-hand order located at position x in the Selectrons by the left-hand 12 digits in the Accumulator. |
| 20. | L | L | Multiply the number in the Accumulator by 2, leaving it there. |
| 21. | R | R | Divide the number in the Accumulator by 2, leaving it there. |

* Register means Arithmetic Register

of the input wires is activated at any one time, while various sets of one or more of the output wires are activated. This particular table may be referred to as the recoding function table.

The twelve flip-flops operating the four function tables used in selecting a Selectron position, and the six flip-flops operating the function table used for decoding the order, are referred to as the *Function Table Register*, FR.

6.4 Let us consider next the process of transferring a pair of orders from the Selectrons to the Control. These orders first go into SR. The order which is to be used next may be transferred directly into FR. The second order of the pair must be removed from SR (since SR may be used when the first order is executed), but can not as yet placed in FR. Hence a temporary storage is provided for it. The storage means is called the *Control Register*, CR, and consists of 20 (or possibly 18) flip-flops, capable of receiving a number from SR and transmitting a number to FR.

As already stated (6.1), the Control must know the location of the pair of orders it is to get from the Selectron memory. Normally this location will be the one following the location of the two orders just executed. That is, until it receives an order to do otherwise, the Control will take its orders from the Selectrons in sequence. Hence the order location may be remembered in a twelve stage binary counter (one capable of counting $2^{12}$) to which one unit is added whenever a pair or orders is executed. This counter is called the *Control Counter,* C.C.

The details of the process of obtaining a pair of orders from the Selectron are thus as follows. The contents of CC are copied into FR, the proper Selectron location is selected, and the contents of the Selectrons are transferred to SR. FR is then cleared, and the contents of SR are transferred to it and CR. CC is advanced by one unit so the Control will be prepared to select the next pair of orders from the memory. (There is, however, an exception from this last rule for the so-called transfer orders. This may feed CC in a different manner, cf. the next paragraph below.) First the order in FR is executed and then the order in CR is transferred to FR and executed. It should be noted that all these operations are directed by the Control itself, not only the operations specified in the Control words sent to FR but also the automatic operations required to get the correct orders there.

Since the method by means of which the Control takes order pairs in sequence from the memory has been described, it only remains to consider how the Control shifts itself from one sequence of control orders to another in accordance with the operations described in 3.5. The execution of these operations is relatively simple. An order calling for one of these operations contains the twelve digit specification of the position to which the Control is to be switched, and these digits will appear in the left-hand twelve flip-flops of FR. All that is required to shift the Control is to transfer the contents of these flip-flops to CC. When the Control goes to the Selectrons for the next pair of orders it will then go to the location specified by the number so transferred. In the case of the unconditional transfer, the transfer is made automatically; in the case of the conditional transfer it is made only if the sign counter of the Accumulator registers zero.

6.5 In this report we will discuss only the general method by means of which the Control will execute specific orders, leaving the details until later. It has already been explained (5.5) that when a circuit is to be designed to accomplish a particular elementary operation (such as addition), a choice must be made between a static type and a dynamic type circuit. When the design of the Control is considered, this same choice arises. The function of the Control is to direct a sequence of operations which take place in the various circuits of the computer (including the circuits of the Control itself). Consider what is involved in directing an operation. The Control must signal for the operation to begin, it must supply whatever signals are required to specify that particular operation, and it must in some way know when the operation has been completed so that it may start the succeeding operation. Hence the control circuits must be capable of timing the operation. It should be noted that timing is required whether the circuit performing the operation is static or dynamic. In the case of a static type circuit the Control must supply static control signals for a period of time sufficient to allow the output voltages to reach the steady-state condition. In the case of a dynamic type circuit the Control must send various pulses at proper intervals to this circuit.

If all circuits of a computer are static in character, the control timing circuits may likewise be static, and no pulses are needed in the system. However, though some of the circuits of the computer we are planning will be static, they will probably not all be so, and hence pulses as well as static signals must be supplied by the Control to the rest of the computer. There are many advantages in deriving these pulses from a central source, called the *clock*. The timing may then be done either by means of counters counting clock pulses or by means of electrical delay lines (an RC circuit is here regarded as a simple delay line). **Since the timing of the entire computer is governed by a single pulse source, the computer circuits will be said to operate as a synchronized system.**

The clock plays an important role both in detecting and in localizing the errors made by the computer. One method of checking which is under consideration is that of having two identical computers which operate in parallel and automatically compare each others results. Both machines would be controlled by the same clock, so they would operate in absolute synchronism. It is not necessary to compare every flip-flop of one machine with the corresponding flip-flop of the other. Since all numbers and control words pass through either the Selectron Register or the Accumulator soon before or soon after they are used, it suffices to check the flip-flops of the Selectron Register and the flip-flops of the Accumulator which hold the number registered there; in fact, it seems possible to check the Accumulator only. The checking circuit would stop the clock whenever a difference appeared, or stop the machine in a more direct manner if an asynchronous system is used. Every flip-flop of each computer will be located at a convenient place. In fact, all neons will be located on one panel, the corresponding neons of the two machines being placed in parallel rows so that one can tell at a glance (after the machine has been stopped) where the discrepancies are.

**The merits of any checking system must be weighed against its cost. Building two machines may appear to be expensive, but since most of the cost of a scientific computer lies in development rather than production, this consideration is not so important as it might seem.** Experience may show that for most problems the two machines need not be operated in parallel. Indeed, in most cases purely mathematical, external checks are possible: smoothness of the results, behavior of differences of various types, validity of suitable identities, redundant calculations, etc. All of these methods are usually adequate to disclose the presence or absence of error in toto, their drawback is only that they may not allow the detailed diagnosing and locating of errors at all or with ease. When a problem is run for the first time, so that it requires special care, or when an error is known to be present, and has to be located — only then will it be necessary as a rule, to use both machines in parallel. Thus, they can be used as separate machines most of the time. The essential feature of such a method of checking lies in the fact that it checks the computation at every point (and hence detects transient errors as well as steady-state ones) and stops the machine when the error occurs so that the process of localizing the fault is greatly simplified. These advantages are only partially gained by duplicating the arithmetic part of the computer, or by following one operation with the complement operation (multiplication by division, etc.), since this fails to check either the memory or the Control (which is the most complicated, though not the largest, part of the machine).

The method of localizing errors, either with or without a duplicate machine, needs further discussion. It is planned to design all circuits (including those of the Control) of the computer so that if the clock is stopped between pulses the computer will retain all its information in flip-flops so that the computation may proceed unaltered when the clock is started again. This principle has already demonstrated its usefulness in the ENIAC. This makes it possible for the machine to compute with the clock operating at any speed below a certain maximum, as long as the clock gives out pulses of constant shape regardless of the spacing between pulses. **In particular, the spacing between pulses may be made indefinitely large. The clock will be provided with a mode of operation in which it will emit a single pulse whenever instructed to do so by the operator. By means of this, the operator can cause the machine to go through an operation step by step, checking the results by means of the indicating-lamps connected to the flip-flops.** It will be noted that this design principle does not exclude the use of delay lines to obtain delays as long as these are only used to time the constituent operations of a single step, and have no part in determining the machine's operating repetition rate. Timing coincidences by means of delay lines is excluded since this requires a constant pulse rate.

6.6 The orders which the Control understands may be divided into two groups: those that specify operations which are performed within the computer and those that specify operations involved in getting data into and out of the computer. At the present time the internal operations are more completely planned than the input and output operations, and hence they will be discussed more in detail than the latter. The internal operations which have been tentatively adopted are listed in Table I. It has already been pointed out that not all of these operations are logically basic, but that many can be programmed by means of others. In the case of some of these operations the reasons for building them into the Control have already been given. In this section we will give reasons for building the other operations into the Control and will explain in the case of each operation what the Control must do in order to execute it.

**6.6.7 One basic question which must be decided before a computer is built is whether the machine is to have a so-called floating binary (or decimal) point. While a floating binary point is undoubtedly very convenient in coding problems, building it into the computer adds greatly to its complexity and hence a choice in this matter should receive very careful attention. However, it should first be noted that the alternatives ordinarily considered (building a machine with a floating binary point vs. doing all computation with a fixed binary point) are not exhaustive and hence that the arguments generally advanced for the floating binary point are only of limited validity. Such arguments overlook the fact that the choice with respect to any particular operation (except for certain basic ones) is not between building it into the computer and not using it at all, but rather between building it into the computer and programming it out of operations built into the computer.**

Building a floating binary point into the computer will not only complicate the Control but will also increase the length of a number and hence increase the size of the memory and the arithmetic unit. Every number is effectively increased in size, even though the floating binary point is not needed in many instances. Furthermore, there is considerable redundancy in a floating binary point type of notation, for each number carries with it a scale factor, while generally speaking a single scale factor will suffice for a possibly extensive set of numbers. By means of the operations already described in the report a floating binary point can be programmed. While additional memory capacity is needed for this, it is probably less than that required by a built-in floating binary point since a different scale factor does not need to be remembered for each number.

To program a floating binary point involves detecting where the first zero occurs in a number in A. Since A has shifting facilities this can best be done by means of them. In terms of the operations previously described this would require taking the given number out of A and performing a suitable arithmetical operation on it: for a (multiple) right shift a multiplication, for a (multiple) left shift either one division, or as many doublings (i.e. additions) as the shift has stages. However, these operations are inconvenient and time-consuming, so we propose to introduce two operations (L and R) in order that this (i.e. the single left and right shift) can be accomplished directly. These operations make use of facilities already present in A and hence add very little equipment to the computer. It should be noted that in many instances a single use of L and possibly of R will suffice in programming a floating binary point . . .

6.8 In this section we will consider what must be added to the Control so that it can direct the mechanisms for getting data into and out of the computer and also describe the mechanisms themselves. Three different kinds of input-output mechanisms are planned.

*First:* Several magnetic wire storage units operated by servomechanisms controlled by the computer.

*Second:* Some viewing tubes for graphical portrayal of results.

*Third:* A typewriter for feeding data directly into the computer, not to be confused with the equipment used for preparing and printing from magnetic wires. As presently planned the latter will consist of modified Teletypewriter equipment.

6.8.1 Since there already exists a way of transferring numbers between the Selectrons and A, therefore A may be used for transferring numbers from and to a wire. The latter transfer will be done serially and will make use of the shifting facilities of A. Using A for this purpose eliminates the possibility of computing and **reading from or writing on the wires simultaneously. However, simultaneous operation of the computer and the input-output organ requires additional temporary storage and introduces a synchronizing problem, and hence it is not being considered for the first model.**

Since, at the beginning of the problem, the computer is empty, facilities must be built into the Control for reading a set of numbers from a wire when the operator presses a manual switch. As each number is read from a wire into A, the Control must transfer it to its proper location in the Selectrons. The CC may be used to count off these positions in sequence, since it is capable of transmitting its contents to FR. A detection circuit on CC will stop the process when the specified number of numbers has been placed in the memory, and the Control will then be shifted to the orders located in the first position of the Selectron memory.

It has already been stated that the entire memory facilities of the wires should be available to the computer without human intervention. This means that the Control must be able to select the proper set of numbers from those going by. Hence additional orders are required for the code. Here, as before, we are faced with two alternatives. We can make the control capable of executing an order of the form: take numbers from positions p to p + s on wire No. k and place them in Selectron locations v to v + s. Or we can make the Control capable of executing some less complicated operations which, together with the already given control orders, are sufficient for programming the transfer operation of the first alternative. Since the latter scheme is simpler we adopt it tentatively.

The computer must have some way of finding a particular number on a wire. One method of arranging for this is to have each number carry with it its own location designation. A method more economical of wire memory capacity is to use the Selectron memory facilities to remember the position of each wire. For example, the computer would hold the number $t_1$ specifying which number on the wire is in position to be read. If the Control is instructed to read the number at position $p_1$ on this wire, it will compare $p_1$ with $t_1$; and if they differ, cause the wire to move in the proper direction. As each number on the wire passes by, one unit is added or subtracted to $t_1$ and the comparison repeated. When $p_1 = t_1$ numbers will be transferred from the wire to the Accumulator and then to the proper location in the memory. Then both $t_1$ and $p_1$ will be increased by 1, and the transfer from the wire to Accumulator to memory repeated. This will be iterated, until $t_1 + s$ and $p_1 + s$ are reached, at which time the Control will direct the wire to stop.

Under this system the Control must be able to execute the following orders with regard to each wire: start the wire forward, start the wire in reverse, stop the wire, transfer from wire to A, and transfer from A to wire. In addition, the wire must signal the Control as each digit is read and when the end of a number has been reached. Conversely, when recording is done the Control must have a means of timing the signals sent from A to the wire, and of counting off the digits. The $2^6$ counter used for multiplication and division may be used for the latter purpose, but other timing circuits will be required for the former.

If the method of checking by means of two computers operating simultaneously is adopted, and each machine is built so that it can operate independently of the other, then each will have a separate input-output mechanism. The process of making wires for the computer must then be duplicated, and in this way the work of the person making a wire can be checked. Since the wire servomechanisms cannot be synchronized by the central clock, a problem of synchronizing the two computers when the wires are being used arises. It is probably not practical to synchronize the wire feeds to within a given digit, but this is unnecessary since the numbers coming into the two organs A need not be checked as the individual digits arrive, but only prior to being deposited in the Selectron memory.

**6.8.2 Since the computer operates in the binary system, some means of decimal-binary and binary-decimal conversions is highly desirable. Various alternative ways of handling this problem have been considered. In general we recognize two broad classes of solutions to this problem.**

*First:* The conversion problems can be regarded as simple arithmetic processes and programmed as *sub-routines* out of the orders already incorporated in the machine. The details of these programs together with a more complete discussion are given fully in Chapter IX, Part II, where it is shown, among other things, that the conversion of a word takes about 5 milliseconds. Thus the conversion time is comparable to the reading or withdrawing time for a word – about 2 milliseconds – and is trivial as compared to the solution time for problems to be handled by the computer. It should be noted that the treatment proposed there presupposes only that the decimal data presented to or received from the computer are in tetrads, each tetrad being the binary coding of a decimal digit – the information (precision) represented by a decimal digit being actually equivalent to that represented by 3.3 binary digits. The coding of decimal digits into tetrads of binary digits and the printing of decimal digits from such tetrads can be accomplished quite simply and automatically by slightly modified Teletype equipment.

*Second:* The conversion problems can be regarded as unique problems and handled by separate conversion equipment incorporated either in the computer proper or associated with the mechanisms for preparing and printing from magnetic wires. Such convertors are really nothing other than special purpose digital computers. **They would seem to be justified only for those computers which are primarily intended for solving problems in which the computation time is small compared to the input-output time, to which class our computer does not belong.**

6.8.3 It is possible to use various types of cathode ray tubes, and in particular Selectrons for the viewing tubes, in which case programming the viewing operation is quite simple. The viewing Selectrons can be switched by the same function tables that switch the memory Selectrons. By means of the substitution operation $Ap \rightarrow S(x)$ and $Ap' \rightarrow S(x)$, six-digit numbers specifying the abscissa and ordinate of the point (six binary digits represent a precision of one part in $2^6 = 64$, i.e. of about 1.5% which seems reasonable in such a component) can be substituted in this order, which will specify that a particular one of the viewing Selectrons is to be activated.

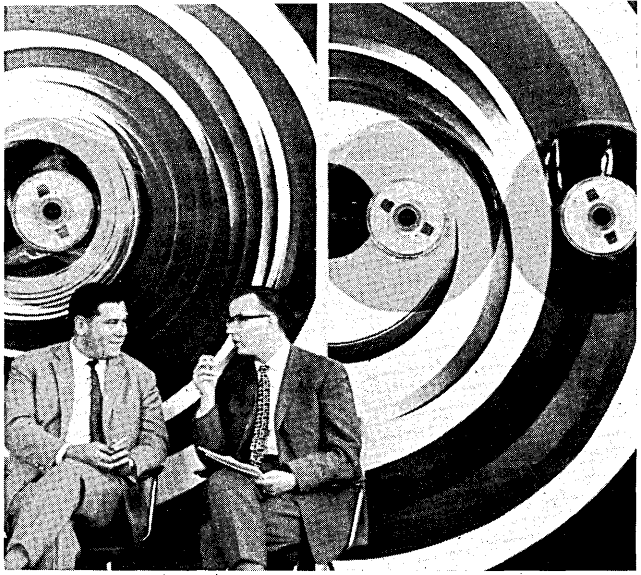
6.8.4 As was mentioned above, the mechanisms used for preparing and printing from wire for the first model, at least, will be modified Teletype equipment. We are quite fortunate in having secured the full cooperation of the Ordnance Development Division of the National Bureau of Standards in making these modifications and in designing and building some associated equipment.

By means of this modified Teletype equipment an operator first prepares a checked paper tape and then directs the equipment to transfer the information from the paper tape to the magnetic wire. Similarly a magnetic wire can transfer its contents to a paper tape which can be used to operate a Teletypewriter. **(Studies are being undertaken to design equipment that will eliminate the necessity for using paper tapes.)**

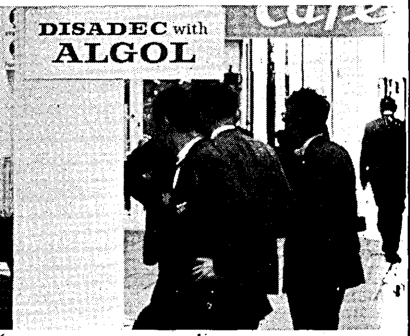
The statement of a new problem on a wire involves data unique to that problem interspersed with data found on previously prepared paper tapes or magnetic wires. The equipment discussed in the previous paragraph makes it possible for the operator to combine conveniently these data onto a single magnetic wire ready for insertion into the computer.

It is frequently very convenient to introduce data into a computation without producing a new wire. Hence it is planned to build one simple typewriter as an integral part of the computer. By means of this typewriter the operator can stop the computation, type in a memory location (which will go to the FR), type in a number (which will go to A and then be placed in the first mentioned location), and start the computation again.

**6.8.5 There is one further order that the Control needs to execute. There should be some means by which the computer can signal to the operator when a computation has been concluded, or when the computation has reached a previously determined point. Hence an order is needed which will tell the computer to stop and to flash a light or ring a bell.** ■








# IFIP REVISITED...or back to

**by HAROLD BERGSTEIN, Editor**

Good sausages were hard to come by at this year's IFIP Congress but in other respects the Munich meeting may be considered a resounding success: attendance was estimated at about 2,800 with representatives from 41 countries participating; the organization of the Congress was as high as the best Teutonic standards would allow; exhibits were attractive if not informative, and an occasional paper was well worth hearing. In addition, the flavor of the Congress was as Bavarian as the travel posters had promised. Perhaps it was just not a good year for sausages. One can't blame Auerbach for everything.

While American papers seemed to dominate the technical program, the conference itself impressed most delegates as European in concept, planning and execution. Less than 450 American participants attended and by far the largest percentage of these were directly involved in the program as panel members or speakers.

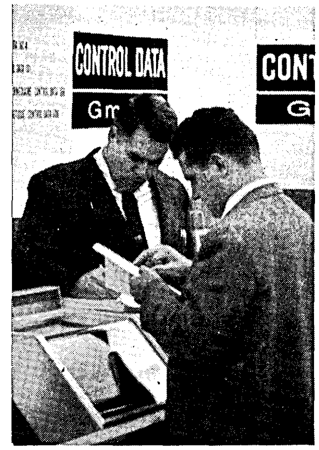
In virtually all respects, IFIP was a programming-oriented conference. Papers on hardware, circuit design, advanced components, etc., drew the smallest attendance while sessions on ALGOL, artificial intelligence, information retrieval, were presented to capacity audiences. Attendees at these sessions differed from their American peers in their application and interest. Few speakers were troubled with delegates wandering in and out of the sessions and almost all audiences seemed to be occupied in comprehensive notetaking as opposed to a rather carefree, seemingly labored attentiveness common to U.S. conferences.

Undoubtedly the most noteworthy aspect of the conference were the planners' efforts at organization, most of which met with remarkable success and only a few bordered on hilarity. The registration packet for example, was truly a monumental collection of materials: several outlines of the program, maps of Munich, the exhibit area, travel folders on Bavaria, a catalog of European equipment, a catalog of a book exhibition, preprints of the technical papers in English, French and German, abstracts of the technical papers, a list of preregistrants, a beer party ticket and a conference badge, all of which were enclosed in a vinyl covered loose leaf folder complete with ball point pen.

At the Technische Hochschule where the technical program was held, a profusion of arrows and signs ushered delegates through a succession of corridors and hallways which could properly be labelled "awe-inspiring." Somehow, during the course of the conference, a few strategic arrows were slanted or turned in directions contrary to their original position and delegates complained of wandering into wash rooms or broom closets when their goal was indeed, the main auditorium.

There are many, many hotels in Munich and it seemed that all of them were used for the Congress from \$4 per night pensions to \$18 plus breakfast, U.S.-style accomodations. Almost everyone had a room and at times it appeared that almost everyone was registered at a different hotel. Hallway conversations soon stretched into the alleyways of downtown Munich but somehow persons with like interests seemed to attract one another. As one delegate phrased it, "It worked out *despite* the organization."

Traffic in the exhibit area appeared light during most of the conference although the exhibit was well worth the viewing. One amusing incident in this respect occured during and after the opening session. Exhibit guards had been told not to admit delegates directly into the show




# Bavaria with photos and prose

from the main auditorium. However, some of the American delegates insisted that they be admitted and the German guards appeared equally stubborn, resulting in what history may someday record as one of the last, minor engagements of World War II.

Survivors able to gain admittance to the exhibit were pleased to find an exceptionally handsome, flower bedecked display of some of Europe's best entries in the main frame contest, and a number of American firms pointing their best toe forward into the burgeoning, hopefully soon-to-be-common market. The interest of Europeans in ALGOL was exemplified by numerous signs accompanying equipment exhibits and of course, in frequent conversations throughout the Congress. In Europe, FORTRAN is generally viewed as "that other language."

The need for small to medium scale business and scientific machines was apparent throughout the exhibit, as was interest in process control and real time systems. American equipment on exhibit included several elements of CDC periphearal gear, Ampex tape transports, Friden typewriters and data collection devices, Hewlett Packard recorders, registers and a-d converters, Potter tape transports, IBM's 1620, CalComp's plotter, NCR CRAM, General Precision's LGP-30, and RPC 4000. UNIVAC displayed a memory core stringing technique and pamphlets on the 1107. . .

At Ferranti, the ATLAS installation at Manchester was reported as still *not* on the air . . . ICT has recently merged with SEA computers . . . Telefunken offered some interesting line printers and mag tape units . . . The Zuse exhibit attracted much attention with a reconstruction of the first program controlled computer, the Zuse Z3 and the new Z23, a 10K machine on the air for the show . . . At this booth, the Packard Bell 250 was also shown . . . Bull reported that the Gamma 60 was fully operational . . . The size of the Siemans booth drew special attention for its high speed printer and data transmission equipment . . . Unrelated to Zuse was a historical exhibit from the Deutsche Arbeitgemeinschaft fur Rechenanlagen in Darmstadt which also laid claim to developing the world's first computer . . . The French exhibitors were arranged in a collected body of 10 firms and associations including the French equivalent of ACM. At one of these exhibits, the passing visitor could receive a presentation in a choice of several languages by pressing an appropriate button . . .

In general, most of the exhibits showed operating equipment and were informative. Salesmen were well briefed and able to communicate in several languages.

Two unfortunate incidents of transportation were reported: the CDC 606 tape drive was dropped en route from the U. S. as was a main frame from the Danish Institute of Computing in Copenhagen. Dominating the exhibit area by the sheer weight of its space and sales force was IBM World Trade.

Many sidelights to the Congress were almost as interesting as the Congress itself. On exhibition was the PERM computer of the Technical High School in Munich, an interesting display of literature and hardbound editions drew many visitors and the press room of the Congress headquarters provided an impressive sight of its own. On a small table in this room there appeared on the second day of the show, the following material *only*: a copy of Auerbach's opening address to the Congress, a selection of photographs of Ike Auerbach, a press release based on Auerbach's opening address and a biography of the newly reelected chairman of IFIP, Isaac L. Auerbach. ■

# The far-reaching possibilities of NCR's amazing new reading machine!

Consider the promise of NCR's revolutionary optical scanner.

Now, it makes possible the direct processing of printed data without having to convert that data into "machine language" such as punched cards or punched tape.

It saves time and money by processing data that was obtained automatically in making original entries.

It reads and analyzes cash register tapes at the speed of 520 characters per second.

It processes printed data from adding machines and accounting machines as well.

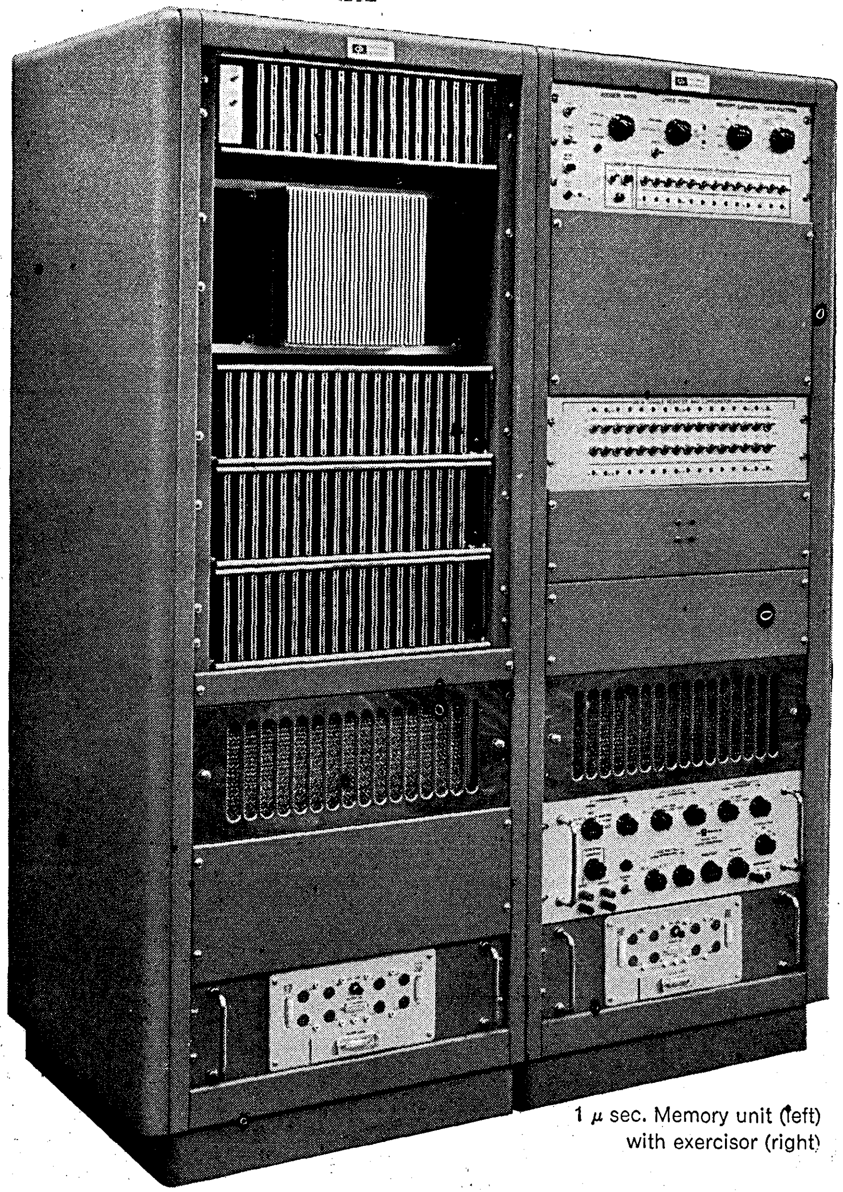
Optical Scanning unlocks a whole new dimension in the simplification of paper work—for businesses large or small. NCR's reading machine shortens the time required to prepare reports... and helps answer *today's* problems ***today.***

This exciting new development is another example of NCR's dedication to providing the finest in Total Systems to make your business life better. Let your local NCR representative explain how this can help your business.

NCR PROVIDES TOTAL SYSTEMS—FROM ORIGINAL ENTRY TO FINAL REPORT—
THROUGH ACCOUNTING MACHINES, CASH REGISTERS OR ADDING MACHINES, AND DATA PROCESSING
The National Cash Register Co.•1,133 offices in 120 countries•78 years of helping business save money

NCR

CIRCLE 20 ON READER CARD

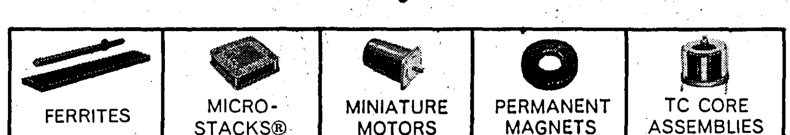


1 $\mu$ sec. Memory unit (left) with exercisor (right)

# 1 $\mu$ sec. word organized Memory System

**. . . joins the broad line of Indiana General memory systems.** "Terminal" switching — a new, more efficient switching technique developed by IGC engineers — offers you high read/write speed plus the time-proved reliability of FERRAMIC® core memory elements. ■ Sense windings and increment windings are wound to cancel increment noise pickup in the sense windings. ■ Drive currents are derived from a constant current source and switched to drive lines on command to eliminate problems of high current pulsing. ■ One-volt noise rejection built into all circuits. ■ All Logic Circuits utilize the Norlogic technique. ■ Sense amplifier has a wide band pass with a high signal-to-noise rejection ratio. ■ System is furnished with capacities in multiples of 2,048 words in all standard access modes — with shortest possible delivery. ■ Wide range of optional equipment available. ■ **Magnetic storage systems are our business.** We offer a wide range of main and buffer memory systems with virtually unlimited capacities and bit lengths, wide range of cycle speeds. **Write today for engineering data file on our new 1 $\mu$ sec. memory system to** Indiana General Corporation, Electronics Division, Keasbey, New Jersey.

**INDIANA GENERAL**




CIRCLE 21 ON READER CARD

# any questions?

triton®
ELECTRONICS, Inc.
62-05 30th Avenue
Woodside 77, N. Y.
Computer Tape Certified 556 Bits

## No question with Triton Magnetic Computer Tape ABH-5560

Every reel is boxed with its own individually registered test chart. Certified assurance that your reel has passed the stringent test of 112 million error-free test pulses at 556 bits per inch*. In addition, Triton's exclusive Trionized® process achieves significantly less oxide shed, less abrasion, and longer life. This assured quality and consistency can be yours tomorrow. Simply write, wire, or phone today.

TAPE TEST CHART
TAPE NO. ABH-5560-D11
TESTER NO. 7
DATE Sept 5, 1962
OPERATOR J. L.

*AN ERROR IS ANY REDUCTION BELOW 50% OR ANY INCREASE IN NOISE EXCEEDING 10% OF NORMAL SIGNAL AMPLITUDE. WHEN RECORDING 556 NRZ PULSES PER INCH ON SEVEN .035 INCH TRACKS PER 1/2" TAPE

CIRCLE 22 ON READER CARD

**with end-product emphasis**

# PROPOSED: AN R&D CENTER FOR EDUCATIONAL DATA PROCESSING

by ALVIN GROSSMAN, Bureau of Pupil Personnel Services,
Dept. of Education, Sacramento, California

The rapid increase in the number of school pupils in the last decade has created serious problems in processing vital educational data. For each of the millions of pupils in the secondary schools, literally hundreds of pieces of essential data are generated, processed, recorded, copied, reprocessed, filed, evaluated, collated, and indexed. Information that was formerly not even gathered is now deemed essential because of the introduction of data processing equipment and its possible application in the areas of planning and evaluating instruction, curriculum studies, guidance, counseling, and other ancillary processes. Because of these factors, the burden of data problems has become increasingly heavy and will become heavier as more and more data are gathered about more and more pupils.

The introduction of computing equipment is initiating a minor revolution in the structural relationship among educational agencies which consume vast amounts of data and is making advances possible which would be out of the question without such equipment. More and more school districts across the nation are rapidly moving into the use of educational data processing services. However, too many educators have become pre-occupied with data processing tools, and too little concerned with the information product. It appears that it is vitally necessary for some agency to provide substantial assistance or consultation to local districts and state departments of education and that it is financed to explore more effective methods of data processing.

The California State Department of Education, through its Bureau of Pupil Personnel Services, has been studying the impact of data processing on the public schools in California through a pilot research project involving five school districts in the San Francisco Bay Area. The study was prompted by the major problems described above and by the necessity for relieving professional educators of the increasing burden of clerical work involved in handling educational data.

As a result of the experiences in the research project the possibility of extending or facilitating a local development of the types of services provided by the pilot project throughout the state was considered. A State Advisory Committee to the Department of Education recommended that as major support for such local centers, and as a stimulus for improving the effectiveness of all such systems, a common center for development in data processing as applied to educational services be established. It was recommended that the center be established within the State Department of Education so that the organization and control of such a unit be under state direction.

The operation of the proposed center would serve the following major purposes:

1. *Design and development of model systems for collecting and handling data to meet the requirements of educational functions, including the analysis of existing systems and a planned transition to a more effective system.*

Basic to the development of educationally oriented systems is the analysis of the specific requirements of data processing systems. Caution needs to be exercised so that modifications can be made at different levels of local control. The restrictions and conditions under which the new system will operate must be determined in detail and in advance. Most of these restrictions will be technical in nature, influenced perhaps by equipment considerations, but more important conditions will be imposed by regional, financial, psychological, administrative and scheduling considerations that affect system design. Other foreseeable conditions must be provided for; there must be adequate staff orientation to enable educators both to make optimum use of the system potentials and to play their part in a system with understanding and cooperation. *At all times, educational objectives and standards must dictate the nature of the processing system rather than vice versa.*

It should be emphasized that the activities of the center would focus on development in data processing itself, rather than on studies which would incidentally use data processing as an instrument. There is, at present, no agency outside the laboratories of manufacturers which is focused on this vital area.

A research and development center could develop improved designs for a pupil accounting system already using machine processes, as well as to experiment with areas which have not yet been adapted in any degree to machine data processing. For example, the design of class and pupil schedules and student registration would improve provisions for eliminating scheduling conflicts of students, a system making possible the maximum use of school facilities, place teachers in optimum teaching situations, and provide for the necessary records for office and teacher use. The procedure would be basic to the entire pupil accounting system, since it would tie in closely with the attendance system, report cards, cumulative records, and many varied reports, lists and statistical studies.

2. *Research and development in facilitating computer applications by means of programming aids such as compilers, and report generators.* One of the most serious obstacles to effective use by educators of the proper resources of internally programmed computers is the apparently forbidding problem of learning to prepare programs. In recent years manufacturers and technicians have worked cooperatively to develop programming aids such as assemblers, compilers, and report generators which considerably facilitate the programmer's task. Compilers, in particular, make use of linguistic, logical and algebraic forms which it is hoped, may ultimately be written in an easily readable style which are easily learned by a beginner. Some of these "languages" are also designed to be machine independent. However, as in the case of the computers themselves these languages are problem-oriented; the needs of scientists, engineers, and other specialists are given greater attention. What is needed is an educational compiler based on existing computer languages but embodying intrinsic procedures, functions and routines most often used in educational applications. It was felt that COBOL would be the most logical language to use. Again, the proposed center can sponsor and support the development of such systems by working in conjunction with technical specialists who are professionally trained to develop and compile the languages of manufacturers who must prepare the internal translators which perform symbolic programs into the object languages used by the particular machines.

3. *Conduct and support of simulation studies in such areas as data processing (pre-trial of new systems) and in specific applications, such as scheduling, guidance, validation of counseling procedures, curriculum changes.* It is now quite evident that at least certain types of complex decision processes can be performed or better assisted by computer systems and often in a more efficient manner than by humans. The potential of these quantitative decision-making techniques appears to be great. The efficient use of data would allow educators to make better decisions and for the first time allow them to experiment with techniques of solution before expending funds on untried programs or services.

The center could experiment with methods of performing decision-making functions on computing equipment which would involve use of "theoretical models" of the consequences of decisions. These theoretical models can become quite complex and may require many man-months or even man-years for their development.

4. *Establishment and maintenance of a library and clearing house for information relevant to educational data processing methods, systems, and equipment.* Some school systems have already prepared manuals describing their procedures and one can make a sizeable collection of different card and tabulating forms which may provide valuable guidance and inspiration for other districts. There is also a growing literature of data processing applications which need to be reviewed, summarized, collated and made available through a central clearing house, since much of this literature is available only in sources not usually consulted by educators, or distributed by manufacturers only to their own customers.

Development of computer programs creates, in a sense, capital items of expense; unlike furniture or buildings, however, once these programs have been perfected, they may be shared at little or no cost with many districts, thus increasing the efficiency of each district's system and reducing the unit cost of program development.

5. *Development of proposals for cooperative establishment of compatible systems for the mutual benefit of local school systems.* Each district has established its own system, usually to the accompaniment of considerable pressure to get work done and hence, with little time or encouragement for original and developmental study of the methods themselves. Thus, a wide variety of systems have emerged, each differing in major or minor details from other systems. Little attention has been given to the need for exchanging, collating, and summarizing valuable educational information, except in the particular and often transitory problems. Consequently, there is also much wasteful duplication of developmental efforts and expense. There is of unquestionably great potential value to educators throughout the country if feasible methods can be developed to promote a variety of educational information to meet requirements at local, regional, state, and national levels.

6. *Evaluation and analysis of available data processing equipment and the development of specifications for required capabilities of new equipment for educational use.* Until recently, manufacturers have designed equipment for commercial, scientific and military use and rarely tried to meet the specific and unique needs of educators. Recent and current developments in the technology of data processing make it increasingly difficult for educators who are inexperienced in this field to obtain from an objective and single source, reliable information about the nature of existing equipment and the configurations appropriate to their local systems. Collection and analysis of essential and critical performance characteristics of available equipment can best be performed by an agency such as the proposed Research and Development Center.

As the volume of data processing services increases throughout the country, it will become increasingly important that educators be able to express with adequate technical understanding and detail, their needs for the modification or creation of equipment. Schools are potentially large consumers of equipment and should be able to focus manufacturers' attention on the specifics of educational needs. There is already evidence of a strong interest on the part of some manufacturers in obtaining such guiding information from educators. Too often the very nature of school data processing systems has been based upon the assumption that schools must use what exists. If the systems are approached without regard to such limitations, new needs may be discovered which can be made known to enterprising manufacturers.

7. *Provision for consultation to local school systems.* At present, a school district which wishes to establish or modify a data processing system is limited in obtaining effective help to a few personnel from other districts or from a few manufacturers' representatives who may be qualified in educational matters as well as in technical knowledge of data processing. There is a great need for a source of consultant help, even for districts already well advanced in data processing, and the wide range of experience which the staff of the proposed center could obtain would be even more valuable than the assistance of persons from single districts who are familiar only with their own methods.

The Research and Development Center headquartered in the State Department of Education would use the live data from the pilot project. Five school districts have been participating in the activities of the center over the last two years. The districts cooperating in the pilot study range in size from a small secondary district of 800 students to a district with a secondary enrollment of 17,000 students. Each district is located in a different county to permit working with the various offices of the county superintendents of schools. There are a total of 35,000 students in 23 secondary schools.

Among the first efforts of the Research and Development Center would be the refinement of the punch-card procedures and techniques now used at the pilot center into an integrated electronic data processing system. An education compiler system would be written and tested as it is developed on the RCA 301 computer, and then through the cooperation and assistance of various other equipment manufacturers, these programs would be run on other types of equipment. Once these programs have been developed for the 301 as well as for other machines, it would then be established as a part of the program library available to all school districts.

Staff members of the research and development team would be assigned to the specific projects listed above. The staff would work in small teams on a particular problem area. "Experts in residence" would be called to the center to contribute their particular talents to the problem under consideration. These experts would be persons with extensive experience in the area under study, appropriate field people as well as representatives of equipment manufacturers. In addition, assigned graduate studies will be carried on in conjunction with the project under the supervision of the Director of the Research and Development Center and cooperating professors at local colleges and universities.

While the experts in residence are at the center, field people who are vitally interested in the problem area would be called in to work on the solution of their mutual problem. Further work would be accomplished at local colleges and universities and school districts following the sessions at the center. This "off-line" activity would be supervised by appropriate members of the California State Advisory Committee on Data Processing.

The Research and Development Center is designed to develop and demonstrate an effective education and information system through the use of a computer. The major emphasis will be on the end products to be created, and not on the tools that are needed to create them. ■

---

# ACM's 17th ANNUAL

## a reflection of sorts

The ACM's 1962 national conference opened in Syracuse on September 4th and closed three days later. Fortunately, a large number of the society's members were unable to attend. This is perhaps the kindest reflection one could make on the show. It was bad.

There were, however, a number of positive aspects which rightfully, should receive attention. Accomodations at the Hotel Syracuse were excellent. Laundry and dry cleaning service was punctual and the work was well done. Breakfast in the dining room was superb. Eggs Hawaiian, for example, featured in addition to two fried eggs, a generous helping of ham, pineapple and bananas. Demi-tasse coffee which preceded breakfast was most welcome.

Despite inclement weather, a clambake held on the second night of the conference, drew satisfactory crowds and all reported eating many clams.

As for the specifics of the conference, the papers were appalling. One speaker on artificial intelligence was questioned on an omission of an important credit. His reply (slightly paraphrased), "I don't attend many sessions where papers are given but if the author will send me his paper (it was a book), I'll look it over."

Another paper which was titled "History of Writing Compilers" was actually a history of one individual's rather limited experience with *a* compiler.

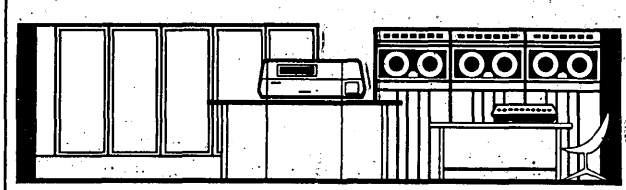
At the exhibits, Bendix poker won the prize for demonstrating the best looking gals fully operative and somehow diminishing the effectiveness of their G-20 mockup. An innovation at this year's show were a few, rather timid representations of computing at several universities. U. of Pitt takes the grand prize in this respect for taping the output of a music generator written for the 7070 and blasting off to the tunes of Dixie and the Pitt Fight Song, a particularly embarrassing note for speakers in the adjoining room who found the accompaniment to their papers rather ludicrous although frankly, quite persistent throughout the conference.

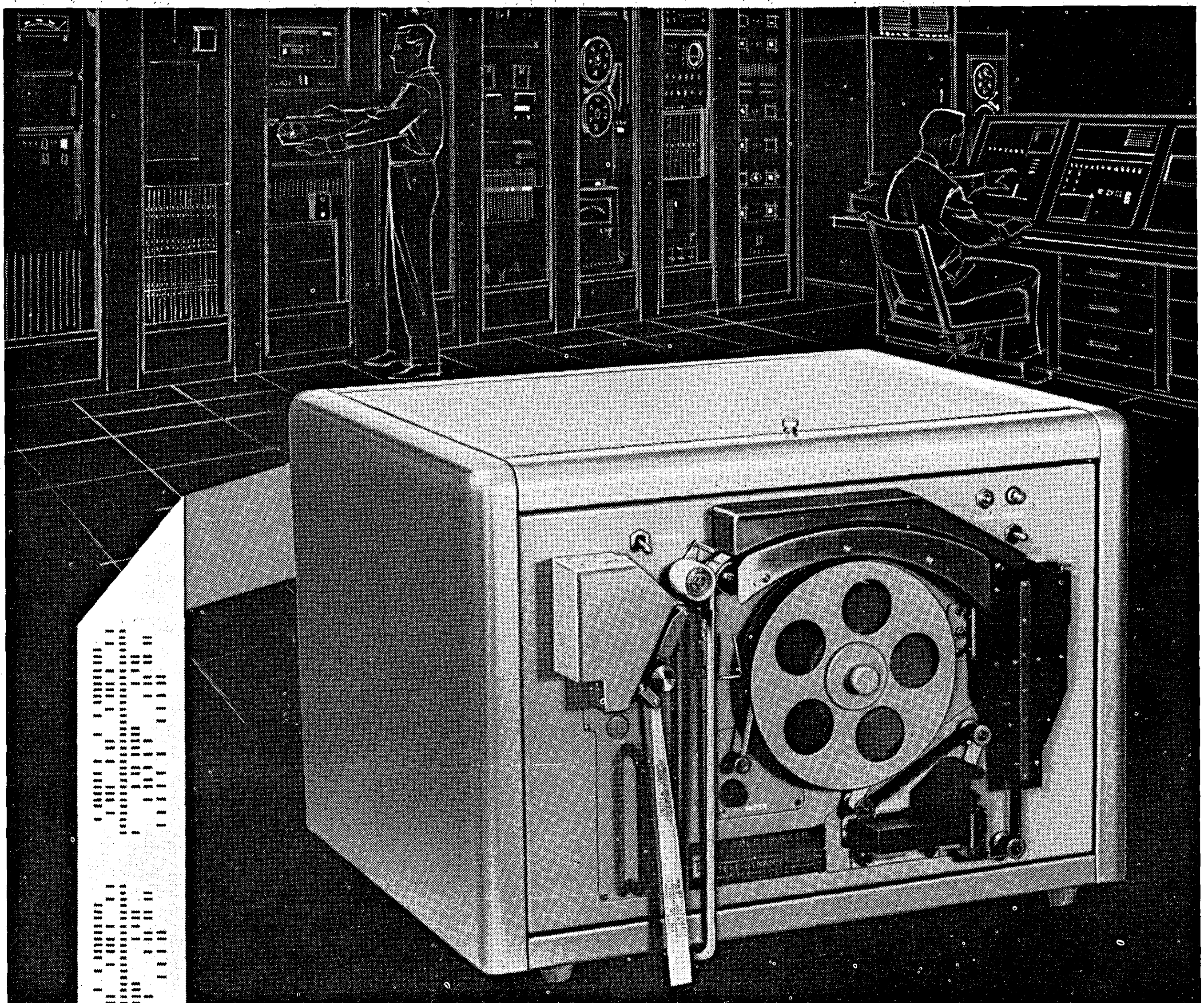
It was widely recognized by virtually all of the 1,100 registrants at Syracuse that this ACM conference was not the best ever. Undoubtedly much constructive work took place in the planning and execution of this show. We regret that much of this labor did not blend well to produce a satisfactory result. Several constructive comments might prove helpful: timing should be watched more closely since this event was scheduled immediately after the Munich meeting and just prior to a gathering of the SHARE clan. In addition, major population centers may have some virtue as was demonstrated last year in Los Angeles when attendance figures, number of exhibits and profits more than tripled the Syracuse balance sheet. Finally, greater care should be exercised in screening abstracts and abstracted speakers. Perhaps a larger number of papers could be invited.

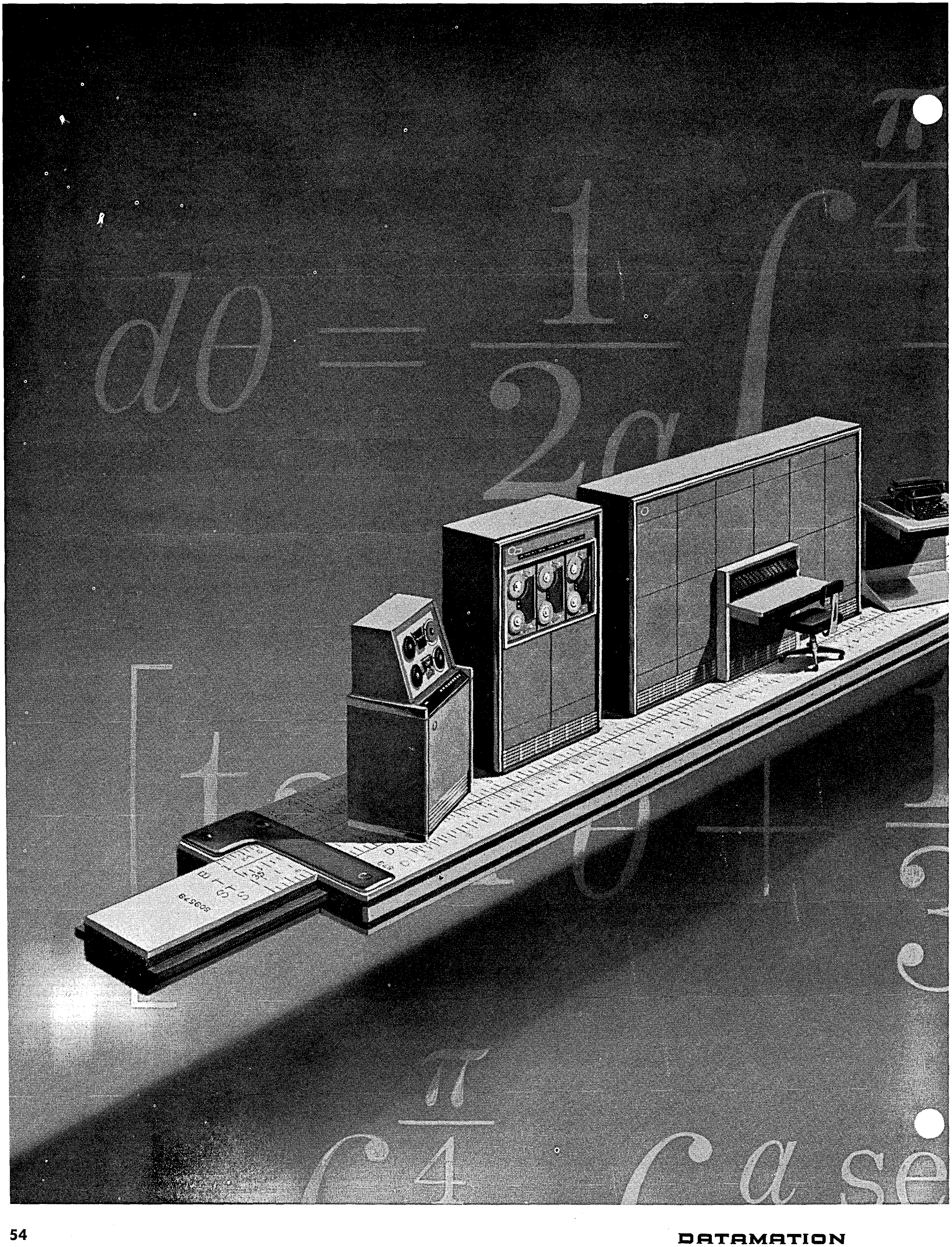
In conclusion, *Datamation* apologizes to those who may be offended by the facetious, earlier remarks in this commentary. We hope, however, that a properly dignified point has been made. ■

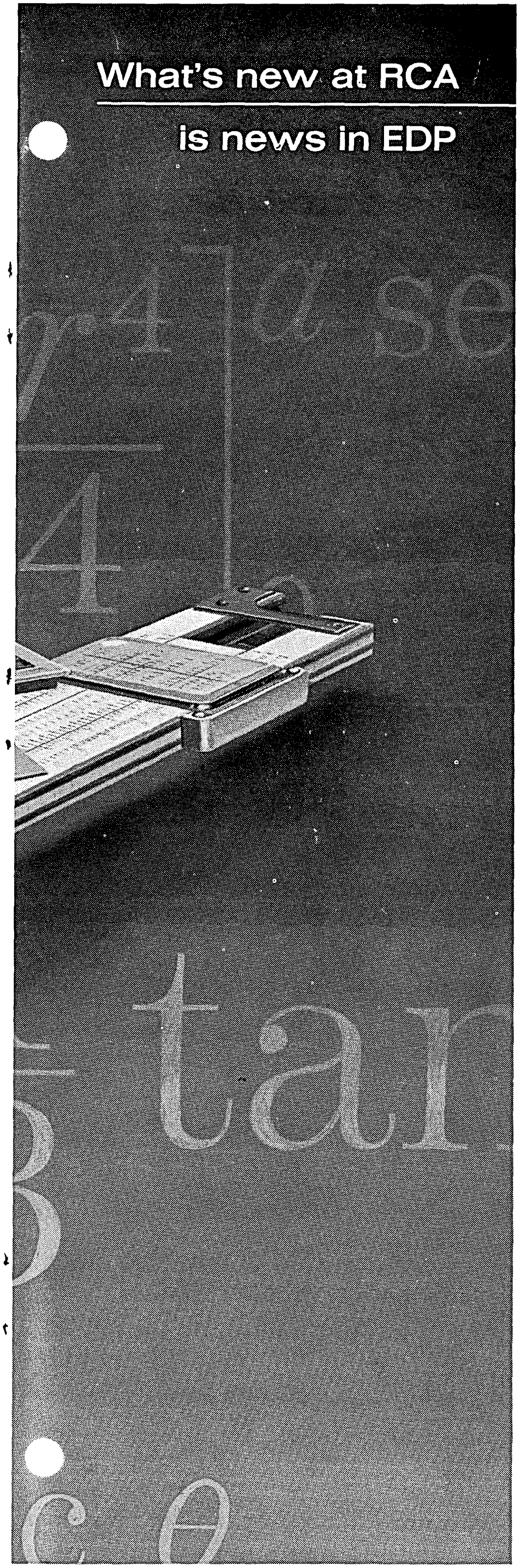
# Honeywell 400 Cobol: mighty strong language

COBOL, by definition, is a common business-oriented programming language for computers. As intended, there is a relatively high degree of compatibility between COBOL compilers designed for specific computers. It does not follow, however, that all compilers are of equal caliber.
Just as there are differences in computers, COBOL compilers vary in scope, speed and effectiveness.










# RCA ADDS NEW SCIENTIFIC CAPABILITY TO THE LOW COST, HIGH CAPACITY 301

RCA's 301, today's best investment in low-cost commercial EDP, is now the best buy for commercial *and* scientific computer power at low cost.

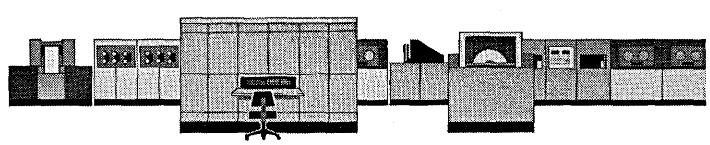
**TOTAL EDP CAPABILITY.** The new RCA 301 gives you *total EDP capability.* With this capability you get greater efficiency for your rental dollars, can schedule time for *both* technical and business assignments. Use the 301 for all your routine EDP business accounting needs. And use it for statistical, analytical and control problems. Assign mathematical tasks to the new 301 and free professional time for creative work.

**MANY NEW WAYS TO USE RCA 301.** Open up new possibilities in your day-to-day data processing with this new 301. For instance—replace several smaller, less efficient computers with one 301. Or—if you have conflicts with time-sharing a large computer, give critical departments their own 301. Or—if you have an outdated computer that's slowing down work, replace it with a 301. You may well save half the cost to do the same job!

**EXTENSIVE SOFTWARE COVERAGE.** RCA makes available a variety of scientific sub-routines—for matrix operations, linear programming, statistical analysis, curve-fitting, double precision floating point, etc., plus Scientific and Bell Interpreter systems, UMAC (an Algebraic Compiler which employs Fortran mathematical statements), and 301 Fortran.

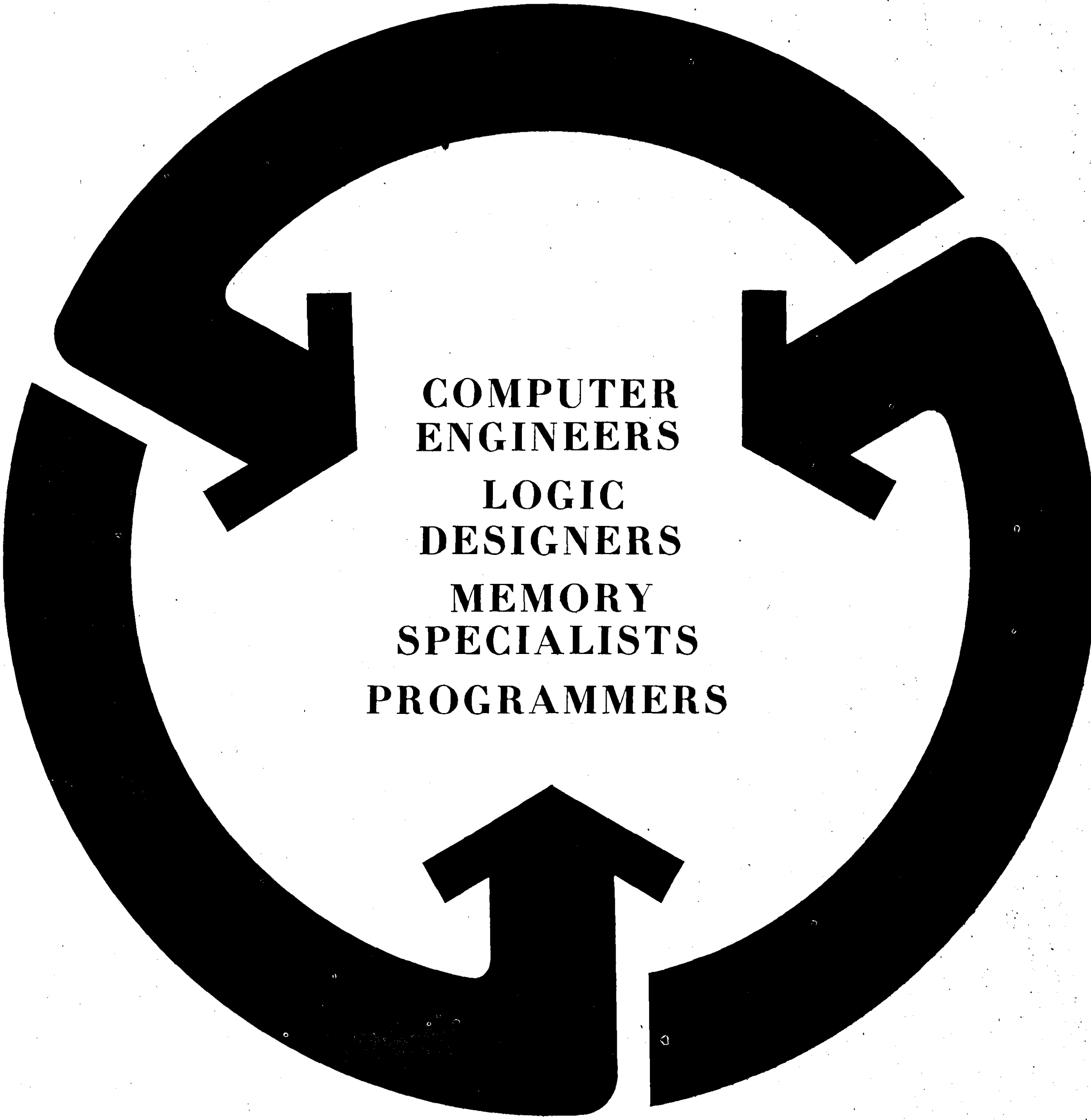
Check the specifications below and find out what the new 301 features can do for you. Then contact RCA Electronic Data Processing, Cherry Hill, Camden 8, N.J.

- An entire family of new fast circuitry arithmetic instructions, including:
  - fixed point
  - floating point
  - 16 digit accumulator manipulation and shifting facilities
- This new instruction format allows the use of bit indicators for storage of operands and for address field modification.
- Approximately 6000 floating or fixed point eight digit add/subtract and 2000 floating or fixed point multiply/divide operations per second. (With 2 digit exponent in floating point.)
- Three index fields for address modification.



**The Most Trusted Name in Electronics**

CIRCLE 26 ON READER CARD




Collins Radio Company has immediate openings for experienced digital computer design engineers at its plant in Cedar Rapids, Iowa. New programs in data processing and message switching offer outstanding opportunities for growth and development with a leader in the electronics field.

COLLINS

Qualified applicants interested in a career with a future, send your resume to:

L. R. Nuss, Manager
Professional Employment
**COLLINS RADIO COMPANY**
Cedar Rapids, Iowa

An equal opportunity employer

CIRCLE 75 ON READER CARD

"Dull" is one adjective which could never be applied to the past and recent activities of RCA's EDP Division. In some measure, the corporate machinations of this major hardware contender have garnered far more than an equal or deserved share of the public spotlight.

The following taped interview was conducted by DATAMATION in an effort to bring a clearer focus to a sometimes muddled scene and to eliminate if possible, some erroneous reflections.

**some reflections on status**

# RCA AND EDP

based on a taped interview with E. S. McCOLLISTER, V.P., Marketing
by HAROLD BERGSTEIN, Editor

**Q:** At present, what is the total investment that RCA has made in its EDP division?
**A:** A figure in the area of 100 million dollars has been mentioned.
**Q:** What is the current state of your general purpose, commercial business?
**A:** We have booked in excess of 340 301 systems, all of them in the 80 column punched card/magnetic tape market and 93 501s are either installed or on order.
**Q:** What is your projected break-even date for the EDP division?
**A:** A statement recently published by David Sarnoff and Elmer W. Engstrom reported that we are going to cut our losses in half in '62 and we will cut them in half again in '63. In '64 we are going into the black. That is our plan and we are on it.
**Q:** How would you rate RCA EDP among the various competitive companies in the field?
**A:** You could look at this in a number of ways such as value of products shipped in a given year or cumulatively in terms of number of installations, etc. This year we will be in the top three in terms of value of products shipped.
**Q:** What are your plans for the 601?
**A:** We are going to sell the 601. We have five of them in production at present, and we'll ship at least two during the latter part of this year.

However, we had made a decision to ease up on the 601 sales effort because there had been so many doubts raised and so much printed concerning the 601 that it made the sales job more difficult and the program was stretched out.

We reached a point where we felt the best way to sell this machine was to have a product that we could demonstrate. That will automatically remove all doubts.

Over and above that problem, you have to remember that a business such as ours is finite in size. We have a fixed number of sales personnel and if these people are spending their time selling 601s, they can't be spending it on something else. From a purely business point of view, we decided to concentrate sales on the 301s and 501s which are in production and are being shipped.

This decision will probably limit the number of sales we will make in the 600 series because time is going by, but if you look at it from a dollar and cents standpoint, it was probably the wisest thing to do. We're not as concerned about product mix as we are about income. It's the amount of dollars of revenue that you can bring in!
**Q:** There has been much discussion concerning the loss of the Cal Tech installation of a 604. From RCA's viewpoint, what happened?
**A:** The computer people at Cal Tech wanted a 604. They felt that the 604 was the best machine for the university but we ran into a situation in which we were uncertain as to when we could deliver one to them so we had to tell them that this was the case.

In addition, we struck a Profit and Loss statement on this installation and because it was an educational discount system, and because the 604 costs a great deal of money, it appeared that over a period of time it would be very costly to RCA. For the overall success of our business, we had to evaluate where an expenditure of this size would do the most good and we felt that we could not justify a 604 at Cal Tech particularly since certain offsetting revenue that we had hoped for did not materialize. I personally went to Pasadena and told them we were going to be late. The straight story is that we asked Cal Tech to be released from the contract.

Q: Within RCA, a capability has been developed in tunnel diode research through Project Lightning work. Will this specific capability be utilized in RCA's next generation of computers?

A: Project Lightning is only one of the things that will be taken advantage of in our next generation of equipment.

Q: How soon will the industry be able to look toward RCA for a new family of equipment?

A: As far as the industry is concerned, the next generation of computers will look quite a bit differently than they do at present. The numbers one generally hears around the industry in this respect are deliveries in late 1965 or the first half of 1966.

Q: Do you have any particular direction in mind as far as engineering advances (i.e., tunnel diodes, cryogenics, thin films, etc.)?

A: Actual selection hasn't been made as yet but we are interested in all of these things as possibilities. There are studies going on right now between thin film memory and microferrites as to relative cost, and in the near future we should have some satisfactory answers.

Q: Quite recently, RCA announced the commercial availability of microferrites. Wouldn't this imply that other, competitive manufacturers may be using advanced components from RCA, while your EDP division does not.

A: I certainly hope not but then, only time can tell. If I may get in a plug, I think that one of the basic resources that supports both RCA's interests and RCA's capabilities in the computer business is the fact that we are doing as much work of as many different kinds throughout the corporation as we are in devices, components, techniques, etc., that have a direct applicability to computing and data processing.

Q: RCA seems to have gone in and out of the process control field. Why?

A: Almost anything you do in anything related to the computer business requires an investment, and you have got to put your chips where you think the greatest return is going to be. At the same time you have to see to it that you protect your main line interest whatever it is and obviously, ours up to this point, has been data processing.

All of the statistics about the industrial computer business indicate that at this time it is a relatively small market. Secondly, there are quite a few people in it; I believe there were seven or eight bidders on a TVA job some months ago. Thirdly, there is not a great deal of profit in it within the foreseeable future. Finally, there are other companies in that business who have an interest extending far beyond just the computer itself. Westinghouse and General Electric are two cases in point. As a result, RCA feels that if the control business is going to grow, it is always possible to get back into it. Right now we've got more important bases to cover.

Q: In the area of programming support, what is the current status of COBOL in RCA?

A: RCA was among the first to produce a COBOL compiler and there are quite a few of our installations using COBOL, probably more than any other manufacturer. Of course, work is still being done on COBOL.

Q: Is work still progressing on FORTRAN for the 601?

A: We had suspended FORTRAN for the 601 but we are now at the point of reinstituting and completing this compiler.

Q: What are your plans for implementing ALGOL?

A: Everyone in the industry says that the algebraic compiler most in demand is FORTRAN, at least at this time and for the foreseeable future. Therefore, if you are concerned with providing your customers and prospects that which they view as the most desirable, one would have to side with FORTRAN. Whether this situation will prevail indefinitely, I

couldn't say, but ALGOL has to be generally accepted in this country before it is going to make good business sense for manufacturers to implement the language, except perhaps to offer it as an alternative to FORTRAN, if you've got to have one.

Q: Have you subcontracted your software development?

A: Yes, I have suggested to our software people that we buy about 30 percent of our total software from outside sources. There are two reasons for this policy: first, to avoid the peaks and valleys of staffing an in-house operation and secondly, to be in a better position to keep informed on the state-of-the-art as practiced elsewhere.

Q: How many employees are presently in RCA EDP?

A: About 2,000 but this does not include the manufacturing of the 601 which is being performed by the Camden plant. Our Service Bureau is also for the most part separate from EDP. The Camden facility is part of Defense Electronic Products (DEP).

Q: How large is your marketing department in EDP and what does it include?

A: About 950 persons are in marketing which includes product planning, advanced programming, field operations and other related functions.

Q: What future directions do you envision for RCA EDP?

A: Obviously, we have already made a significant penetration into the data processing business. I think it is axiomatic in both military science and business science that once you establish a beachhead, you don't abandon it and so you can bet your life we are here to stay.

Recently, we have made inroads into such fields as data communications and we are getting into data acquisition. In addition, the total systems approach is becoming more important and as a company, we have a good deal to gain from increasing emphasis in this area. At present, RCA has a 500-megabuck business in defense which by and large is total systems.

# THE KONSCIENCE OF A KOMPUTER KONSERVATIVE

**a Kludge fable**

**by I. V. GOODY, KKK**

It all started at Konsolidated Knockwurst Kompany when one of our technical people wanted to use a linear program, at least that's what I think he called it, to optimize the mixing of salami, knockwurst, baloney, and other of our products. Said he needed a small desk-size computer that rented for about $1,000 a month to make his calculations. Was going to save us a fortune each year. Had it all figured out, even knew how to program the brain. Well, we prided ourselves on progressive management at Konsolidated Knockwurst Kompany so we went along with it and placed a letter of intent. Besides, one of our competitors had a computer somewhere and we couldn't let them make us look old-fashioned.

The first indication of trouble came when our financial people decided that they merited the prestige of running the computer more than the technical people. Since $ carried more weight than brains in this organization, it was pretty easy for them to swing this. I have to admit they went about it in a business-like way. They made a lot of studies and since they had had several applications in mind they cancelled the letter of intent on the small machine. The man they put in charge of computers was an old tab man. The system he was using had to pass 500,000 cards a day, so naturally he wanted a card-to-card computer. He said he could understand holes in cards, but not spots on magnetic tape. He placed an order with some outfit he had been dealing with for thirty years. He was quite loyal to them since they had gotten him the job at Knockwurst. Can't remember their name, but he said they were pretty reliable. They promised to buy a lot of knockwurst for their plant cafeteria too.

Things seemed to be in pretty good shape until the boss' nephew moved in. He's a pretty smooth operator, but unfortunately quite a playboy. What he knew about computers you could put on one rod of an abacus, but he had "The Connection." Because of his playboy habits they kept him out of trouble by putting him in charge of long-range planning. Confidentially, this planning was so long-range we would all be eating algae rather than knockwurst by the time his plans went into effect. He had no know-how or staff, but he sized up this computer jazz as a good way to move up in the organization and get in line for his uncle's job. He immediately went to a seminar and picked up some terminology. Said what this business needed was integration across functional lines, the total systems approach. We didn't know what he was talking about, but he really had a smooth line. Besides, like I said, he had "The Connection."

First thing he did was cancel the card-to-card order with Old Reliable. Everybody had one of their machines, so putting one in didn't buy you much glory. Besides it was too small a system. You know, prestige is directly proportional to memory size and monthly rental. Also he heard that you didn't rate without magnetic tapes. Well, he was a cagey enough guy to cover his tracks, so to reduce his risks he called in a consultant, who threw the job open for bids. The consultant was to resolve the knotty problem of cards versus tape, among other things. You should see some of those computer manufacturers' proposals! The most conservative one said we could eliminate 90% of the work force. Managers, of course, would not be eliminated. In fact, six more managers would be added.

Well, this consultant couldn't resolve the card-tape problem on technical grounds, so he made the decision the democratic way. He was a very nice guy even if he didn't know much about computers. He took a poll of the 2,000 workers we expected to displace and a majority voted a preference to be displaced by a tape system rather than a card computer. That settled that. Playboy then dropped a hint to the consultant that you could get a lot of prestige with a first-of-its-kind machine. This consultant joker knew which side his bread was buttered on, which is how we got in bed with Kludge Komputer Korporation. The Kludge salesman said he could code our major application *himself* in four weeks. Famous last words! We did get some nice concessions out of Kludge Korp. since our lawyer negotiated the contract—4,000 hours of free debugging time, 20 on-site maintenance men, and a good deal more. The deal covered 28 single-spaced pages. It turned out later all those maintenance men weren't a concession after all—in fact 48 wouldn't be a concession.

Playboy started to build his empire. He picked up 4 managers, 11 guys doing "systems work," and two kids just out of high school that he hired as programmers. 15 chiefs, 2 Indians. Kludge Komputer, after dragging their feet for a while, finally set up a programming course. As instructor they sent in a maintenance man who had never taught a course before. Poor guy, the hardware and software specs hadn't even been firmed up yet on the Kludge, although we didn't know it at the time. Funny thing was, our people had so little background they didn't know the difference. This instructor was pretty conscientious and he wanted to have some sort of a graduation ceremony. Kludge Komputer couldn't afford a luncheon, so this instructor on his own gives everybody 10 nickels and sends them down to the Automat. Said it fit in with the theme of automation. Our chief financial officer was invited as guest speaker and gave out diplomas. He got twelve nickels.

Kludge Komputer had promised us a systems engineer as a consultant to help put in the system. Said he would be an expert on the knockwurst industry as well as on computers. What he knew about knockwurst he got from eating a sandwich the day before he arrived, which was one day after Kludge hired him. Some long term employee! He knew even less about computers than he did about knockwurst, but at least he had a nice personality. He was an honest kid, always told you he didn't know the answer when you asked him a question rather than beat around the bush. He reported to the salesman, the joker who said he could program our job in nothing flat. This salesman didn't even know how to press the "start" button! Kludge Komputer Korp. had a nice habit of sending accountants out on the scientific jobs and scientists out on the data processing jobs. They said it increased objectivity and led to cross-fertilization. After watching this kid they sent us operate in our office, I can see where they get the last part of their theory.

We figured that if the kid could just pass us enough information on the Kludge we could get by. We were pretty naive, because even after the design was frozen we couldn't get any information since Kludge was so disorganized that even at their home office nobody communicated with anybody, least of all with their people in the field. We finally outfoxed them, though. We sent some people to Kludge headquarters supposedly on debugging trips, though we never did get much coded. They wandered around the Kludge plant and picked up any interesting looking writeups or preliminary manuals they saw on peoples' desks. They also grabbed all the card decks they could find and brought them back. Trouble was our people didn't know what to do with the stuff once they got it. They also brought back some software schedules, and *that* got us into real trouble. We didn't know that they were target dates and that you had to add two years to them to get real dates, for the routines that did get released, and most of the routines never did get released.

Playboy didn't believe in schedules, but after about a year and a half without seeing any action, it was obvious to everybody that we were in trouble. We really didn't have any programming staff qualified to use the Kludge Assembly Program. We had counted a good deal on the Kludge compiler (KLUDGTRAN) which was very late. With KLUDGTRAN, we had been told, even our janitors would be able to program the Kludge with an hour of instruction. Our lawyers had put some nice hookers in the contract, as I mentioned before, so we lowered the boom on Kludge Komputer Korp. for being late with the compiler. We demanded a turnkey job from the manufacturer. We had them dead to rights, so they had to send in a small army to do our job. They moved pretty fast when we threatened to put up a "KKK KLOBBERED BY KLUDGE" banner at the next industry convention.

Our systems people had studied the thing to death.



CIRCLE 300 ON READER CARD

They had so many different approaches for a new system that we finally decided to mechanically transfer the old tab system to the computer. When Kludge took over the job, though, our people started to change the system on them. Every day they had some new idea or change. Drove the Kludge boys frantic. Our systems people came up with a system called Scatter And Gather All Media Inputs (SAGAMI), but top management didn't buy this approach. They then redesigned it in the form Scatter And Lose All Media Inputs. SALAMI our management could understand, and things settled down.

Kludge Komputer Korp. was running up some pretty big costs on several of their jobs so they finally cut their support to us in half, and used some of the other half to fund an industry-wide "Emphasis on Support" advertising campaign. Kludge's public relations chief was the same genius who when the Alabama watermelon crop failed, dreamed up the campaign about eating the "cute little watermelons." That was the year it didn't rain much and the melons didn't get any bigger than baseballs.

Two years after we placed our order with Kludge a milestone occurred. We got our first software routine. It didn't do much, only about 100 instructions, but it made us think that Kludge Komputer Korp. was making real progress. This routine was followed in six months by a correction, which was followed in six weeks by an addendum to the correction, which was followed in six days by an errata to the addendum, followed in six hours by a telegram giving the corrections to the errata and then the sincere apologies of Kludge by telephone. We began to lose faith again when KLUDGTRAN arrived. We had been promised 90 percent efficiency compared to coding in assembly language but even the KLUDGE consultant admitted it was more like 10 percent. This really bugged the manager of Kludge's compiler team who sent us a routine coded in KLUDGTRAN that ran 90 percent as fast as the same routine coded in machine language. It turned out to be a routine that advanced the real-time clock and had no input/output. In any event, we never could get through to the compiler boys about what we wanted. They didn't want us Knockwursters profaning their compiler on routine production jobs anyhow.

Things really got interesting when we began to cut over to the new system. Talk about a conversion fiasco! Playboy hired a bunch of high school girls on vacation to key punch some of the manual files. Well, the files were in pretty poor shape to start with. A lot of skeletons started to tumble out of the closet. Then these kids couldn't punch worth a damn. Somebody thought we could save money, so none of the stuff was verified. Any self-respecting data that survived this then had to go through the Kludge Kard Reader. I think some of the Kludge engineers must have seen our meat grinders because this Kard Reader really shredded cards at the drop of a hat. The ones that got through were misread anyhow. What we had in the way of card files wasn't in too good shape either and they had to go through the card grinder too. After we were into this conversion phase for about two weeks the whole business ground to a halt. We didn't have our old files up to date any more and we had no decent files on the computer system. We wound up hiring 162 Manpower Girls and it took them 41 weeks to straighten it out.

We had one big laugh out of this whole mess. Our chief financial officer quite a few years back had said "we shall never pay out Konsolidated Knockwurst's good money on the strength of a bunch of holes punched in a bunch of cards." That was before we got progressive in our management thinking. Well this same officer paid out $200,000 when the Kludge off-line printer taking data from magnetic tape jammed on a payroll run and printed out 1,000 copies of one employee's pay check. The employee was a pretty smart cookie, cashed them all, and was down in Brazil before we knew about it. It turned out that our people didn't have time to build any controls into the system.

About six months after installation, things began to settle down. We had a beautiful showcase on the physical end of it, with very modern decor, and we ran thousands of people by it. Unfortunately, the technical people refused to use the Kludge on their LP problem because they claimed they were shafted by Finance and Playboy's power plays, and were agitating to get a computer of their own. Our key financial work was running, even if it did cost us twice as much as before, but we had a lot of prestige. The Kludge salesman, the guy who predicted four weeks of programming, had been promoted


"....AND TO THINK THAT I BOUGHT THIS SYSTEM TO REPLACE SIX BOOK KEEPERS!"

to Branch Manager, even though it turned out that our job took 40 man-years. I was beginning to think we had just about weathered the computer storm when, because of some program bug, 1,100 of our employees were paid too little. That brought on a devastating strike and the Konsolidated Knockwurst Kompany Board of Directors lowered the boom.

How do I know about all this? Well, my real name is Kurt K. Koldwater, former President of Konsolidated Knockwurst Kompany. I really had my nose rubbed in this computer situation, that is before I was retired prematurely. Funny thing, they didn't pick Playboy to replace me. He didn't look too good either, even if he was the Chairman's nephew. They were looking for a man who knew computers and modern management techniques. This I got for them, but I fixed them good. My last official act as President was to appoint as Executive V. P. the guy who used to head up Kludge Komputer Korporation! He is running the show now at Knockwurst. Guess what I'm doing now? That's right. I'm a management consultant at Kludge Komputer Korp. and my main job is to cut out all the baloney over there. ■

**An Open Letter to**
**Kludge Komputer Kompany Kustomers:**

*Dear Sirs:*

It has come to our attention that certain software systems (KLUDGETRAN, KLAP, and KLAP-TRAP in particular) have been distributed to you by the Kludge Kompany in such a form as to create a considerable amount of extra income for the Kompany. This has been done in a number of ways, using methods which are blatantly obvious to those who wish to use the Kludge efficiently, rather than as a device to demonstrate how much work is being accomplished in an installation. A brief listing of these methods follows: along with some examples:

1) Extraneous comments on the attached 5 line-per minute Komment Printer. (ex. KLAP CAN NEVER REACH THIS POINT IN THE SYSTEM. CONDITION IGNORED.)

2) On Komputers equipped with variable speed tape units, the normal mode is low-speed. A set of 17 KONTROL KARDS are required to make each unit high-speed, and this must be done with each job run.

3) Superfluous Testing and Stops. (ex. TODAY IS NOT SATURDAY, AS SPECIFIED IN THE #*$DAY KONTROL KARD. PLEASE ENTER KORRECT DAY OF THE WEEK AND RESTART ENTIRE SYSTEM.)

4) Redundant System Tape Movement: The Kludge Monitor (KLUMON), containing the above three systems and twenty-three obsolete systems, not only has KLAP-TRAP and KLUDGETRAN at the end of the tape, but when going from one subsystem to the other, no position indicators are saved and a complete rewind of the KLUMON tape is required.

5) International System Errors. These require the rerunning of many jobs after receipt of a KLUMON KORREKTION (which may itself contain errors).

It should be borne in mind that these are not simply the results of poor programming, but deliberate efforts on the part of the Kludge Kompany to MAKE MONEY. The word to be remembered is KAUTION.

Signed: *A Friend*

# NEWS BRIEFS

## HUGHES ENTERS COMPUTER FIELD

The Hughes Tool Co. has created a new division, Hughes Dynamics, which will develop proprietary items in both software and hardware, and will ultimately manufacture such hardware. Initial plans call for the division to provide management services to industry.

Examples of proprietary software, described by John T. Pettit, a Hughes Dynamics senior consultant, were systems for solving complex management, production or accounting problems; as proprietary hardware, information retrieval and peripheral devices.

The division will be engaged in consulting services in the areas of real time control systems; management control systems; plant automation; simulation; operational analysis; and data processing service center operations.

Headquarters will be in Houston and Los Angeles, with offices in principal cities.

## ARMOUR SYMPOSIUM

The Armour Research Foundation of the Illinois Institute of Technology is sponsoring a Computer Applications Symposium, which will feature discussion of present and potential capabilities of computers. The Symposium, to be held Oct. 24th and 25th at the Morrison Hotel, Chicago, will hear Reed C. Lawlor, Los Angeles attorney speak on the impact of computers in the legal profession, and Richard F. Clippinger, Honeywell, whose subject is the current progress in standardization of data processing.

## WEBER NOMINATED FOR 1ST IEEE PRESIDENCY

Dr. Ernest Weber, president of Polytechnic Institute of Brooklyn, has been nominated for president of the Institute of Electrical and Electronic Engineers (IEEE). The group is composed of the merged American Institute of Electrical Engineers, and Institute of Radio Engineers. Named to serve as vice president was Dr. B. Richard Teare, Jr., dean of the College of Engineering and Science Carnegie Tech.

Approval of Weber and Teare, as well as a list of 25 directors, will be voted on by the 160,000 members of the IEEE by October 1.

## MIT TO USE H-1800 FOR SPACE NAVIGATION

The Instrumentation Laboratory of MIT will utilize a Honeywell 1800 for the design of computer circuitry to be used in Project Apollo. Also planned is check out of machine logic and the guidance program of the space-borne computer, and simulation of full operation before the lunar flight takes place.

Scheduled for third quarter, 1963, delivery, the 1800 configuration will consist of a 32K memory, floating point unit, random access disc file, six high density mag tape transports, high-speed printer, card punch and reader.

The laboratory has been using a Honeywell 800 since December, 1961.

## SITES CHANGED FOR JOINT COMPUTER CONFERENCES

The 1963 Fall Joint Computer Conference will be held at Las Vegas, Nevada, instead of Los Angeles, and the 1964 Spring Joint Computer Conference has been changed from Boston to the new Washington Hilton Hotel, Washington, D. C., which is scheduled for completion in October, 1963.

Dr. Willis H. Ware, chairman of the governing board of the American Federation of Information Processing Societies (AFIPS), sponsor of the conferences, said that "due to the growth in exhibitors and attendance at the Joint Computer Conferences, it has become necessary for us to look for more space than conventional conference facilities offer."

Another factor which Dr. Ware said led to the change was that it was to the advantage of both attendees and exhibitors that all activities be kept under one roof. The main auditorium at the Las Vegas Convention Center seats 8500 and the center includes 90,000 sq. ft. of exhibit space and 17 meeting rooms. The Washington Hilton will have 50,000 sq. ft. of exhibit area adjacent to a ballroom planned to seat 4,000.

## CDC SALES UP 107% IN '62

Control Data Corporation's sales and service income for the year ending June 30, 1962, amounted to $41,034,009, representing a gain of 107.4% over 1961's figure of $19,783,745.

Profits, reported William Norris, president of CDC, were $1,542,622, 83.1% higher than in 1961, when $842,524 was recorded. On the basis of the number of shares of common stock outstanding (adjusted in 1961 for the three for one stock split), per share earnings were .39 in 1962, compared with .24 the previous year.

The backlog of orders on June 30, 1962, was $49,410,000; on the same date a year ago, the backlog was $24,220,000.

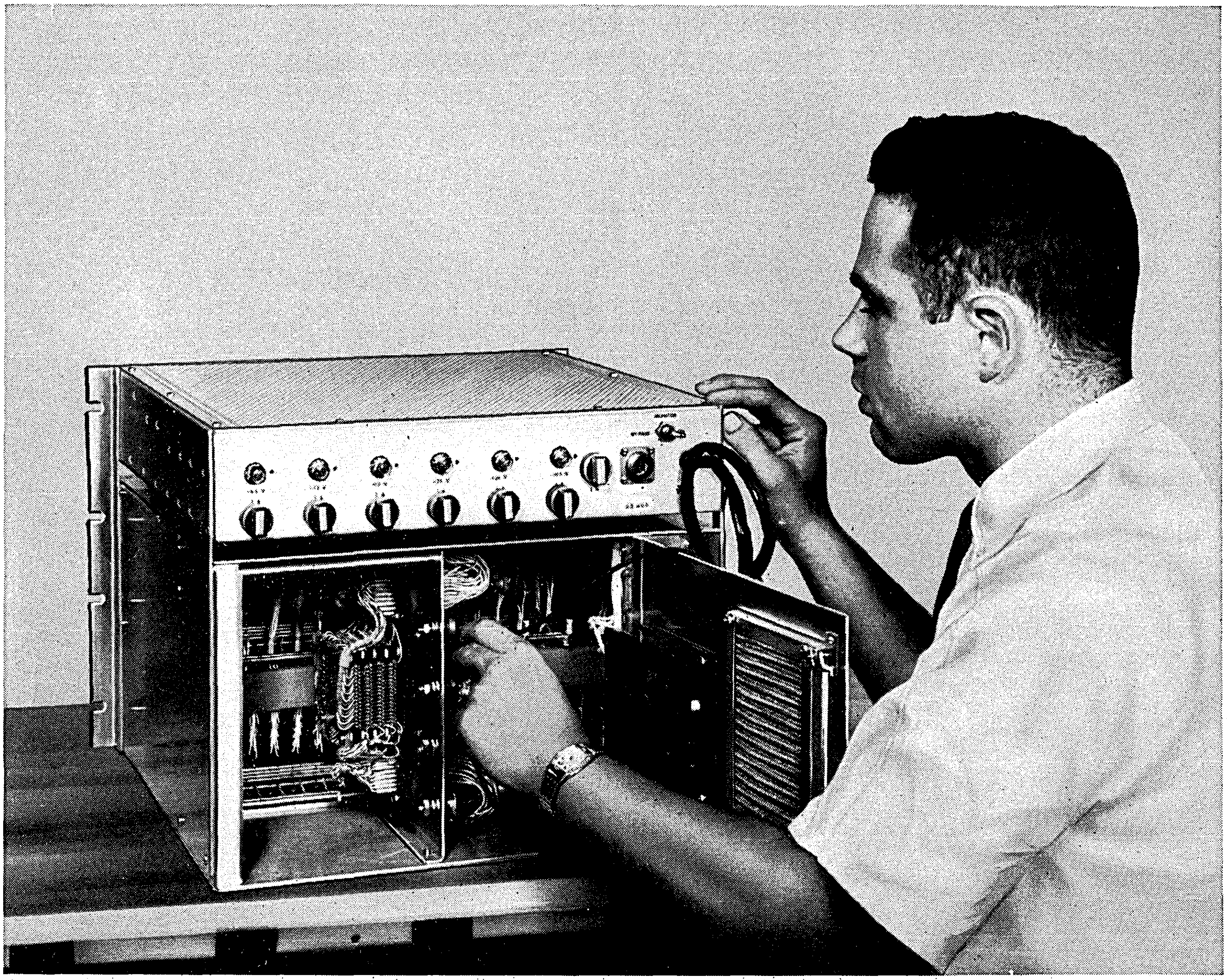
## NO NEWS DEPT.: IBM SALES UP TOO!

Net earnings of IBM were $116,309,000 for the first six months of 1962, reported board chairman Thomas J. Watson, Jr., as compared with the earnings of the corresponding period a year ago of $10,859,439. The after-taxes figure is equivalent to $4.21 a share; last year, the per-share amount was $3.67.

Gross income from sales, service and rentals in the U. S. for the period ending June 30 was $931,705,051, compared with $811,163,397 in the same period during 1961.

## O'SEAS DATA TRANSMISSION

Data transmission at a rate of 1,500 words per minute is being offered by RCA Communications, Inc. The new service for international computer users, called Datatelex, was recently inaugurated when data equivalent to approximately 10K words was transmitted successfully in a 10-minute period between New York City and London. Data in any form—mag tape, punched cards or paper tape may be transmitted at the rates of $40 for the first ten minutes and $4 for each subsequent minute.



# Here's Custom Memory Flexibility in a Standard Package

# RCA 5 $\mu$SEC MS/Series MEMORY SYSTEM

**Compact, high-capacity systems now available with the word lengths you specify—available for quick delivery**

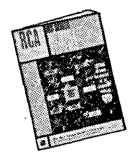
Now you can bring a new standard of operating efficiency, servicing flexibility and plug-in convenience to your computer designs with RCA's complete modular memory system. It utilizes new, compact modular design throughout. All circuits and elements, as well as the complete system plug in for easy maintenance and reduced computer down-time.

This rugged RCA memory system, encased in an aluminum cabinet, has a capacity of 128 to 4096 words and up to 32 bits per word. It incorporates RCA ferrite memory cores and planes, with specified wide margins of operation...up to ±8%...to cope with broad variations of voltage levels over a wide temperature range.

All RCA Complete Memory Systems are available in a broad range of standard types and custom designs. RCA is ready, willing, and able to meet the most stringent custom requirements for voltage, current, temperature range, and speed of operation to make the memory system fully compatible with your equipment.

Here are some of the outstanding features of complete RCA Memory Systems:

- Specified Wider Margins of Operation...Up to ± 8 per cent...to cope with broad variations in voltage levels.
- Custom Design Service...RCA's engineering staff will custom-design a memory system to your specifications.
- Complete Information Retention...even in case of primary power loss.
- Wide Temperature Range...0°C to 50°C.



**FREE BROCHURE** describes complete MS/Series Line including operating modes and options.

Write RCA Semiconductor and Materials Division, Commercial Engineering, Section FD10, Somerville, N.J.

**THE MOST TRUSTED NAME IN ELECTRONICS**

**RCA SEMICONDUCTOR & MATERIALS DIVISION FIELD OFFICES... EAST:** Newark, N.J., 744 Broad St., HU 5-3900 • (Camden-Philadelphia Area) Erlton, N.J., 605 Marlton Pike, HA 8-4802 • Syracuse, N.Y., 731 James St., Rm. 402, GR 4-5591 • Baltimore, Md., EN 9-1850 • **NORTHEAST:** Needham Heights 94, Mass., 64 "A" St., HI 4-7200 • **SOUTHEAST:** Orlando, Fla., 1520 Edgewater Dr., Suite #1, GA 4-4768 • **EAST CENTRAL:** Detroit 2, Mich., 714 New Center Bldg., TR 5-5600 • **CENTRAL:** Chicago, Ill., Suite 1154, Merchandise Mart Plaza, WH 4-2900 • Indianapolis 5, Ind., 2132 East 52nd St., CL 1-1405 • Minneapolis 16, Minn., 5805 Excelsior Blvd., WE 9-0676 • Denver 11, Colorado, Continental Terrace Bldg., Suite 301, 2785 N. Speer Blvd., 477-1688 • **WEST:** Los Angeles 22, Calif, 6801 E. Washington Blvd., RA 3-8361 • (San Francisco Area) Burlingame, Calif., 1838 El Camino Real, OX 7-1620 • Seattle 4, Wash., 2250 First Ave. S., MA 2-8816 • **SOUTHWEST:** Dallas 7, Texas, 7905 Carpenter Freeway, ME 1-9720 • **GOV'T:** Dayton, Ohio, 224 N. Wilkinson St., BA 6-2366 • Washington, D. C., 1725 "K" St., N.W., FE 7-8500 • **RCA INTERNATIONAL DIVISION,** 30 Rockefeller Plaza, New York 20, N. Y. Cable Address: RADIOINTER, N. Y.

CIRCLE 29 ON READER CARD

## NEWS BRIEFS . . .

### 1620, 1410 SPEEDS INCREASED

Memory reference time has been reduced in the 1620 model 2 from 20 to 10 usec, while a new accelerator feature for the 1410 raises internal operations up to 20 percent. CPU digital transmission in the 1620 has also been increased, with two digits being transferred in three 10 usec cycles instead of four 20 usec cycles.

Deliveries of the 1620 model 2 will start in the fourth quarter, 1963. Rental of a card system with 20K storage positions will be about $2,825 monthly, and will cost $129,600.

New circuitry for the 1410 boosts memory speed from 4.5 usec to 4 usec and permits faster addition.

● Electronic Associates, Inc., has developed two special plug-in components which enable the EAI TR-10 analog computer to perform computations in nuclear physics normally performed on larger equipment. The new components, called reactor kinetics network and transport delay simulator, are interchangeable with standard TR-10 components and do not effect the general purpose capabilities of the computer.

CIRCLE 106 ON READER CARD

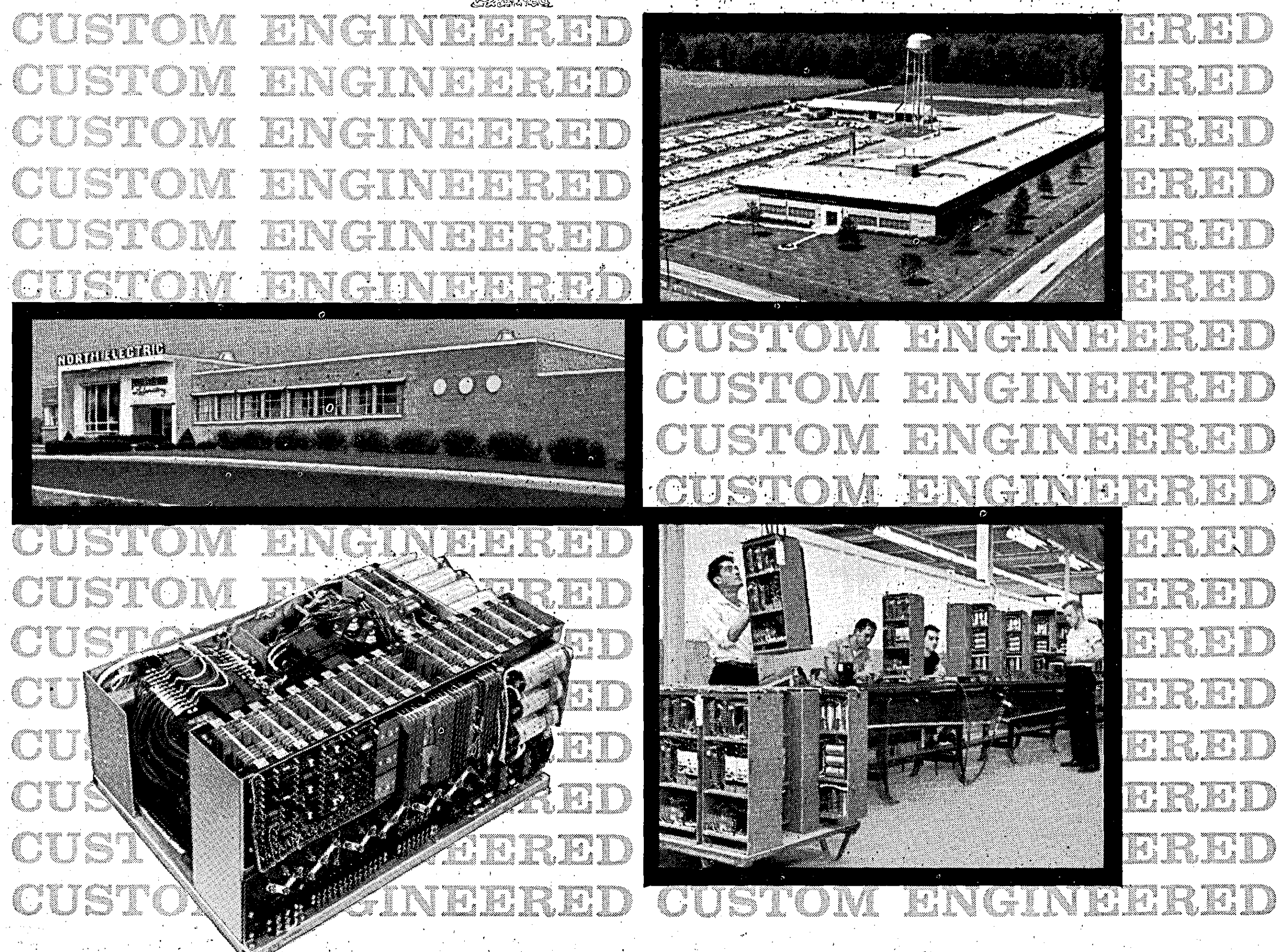
● A bi-directional data converter, made by General Dynamics/Electronics Rochester for the Rocketdyne Division of North American Aviation will convert data generated by an IBM 7090 into punched paper tape that will control the operation of complex machine tools. Data received from paper tape will be converted into mag tape for computer input or tool control. Speed from mag tape to paper tape is 250 cps; in the reverse process, 500 cps is possible.

● Standard Federal Savings & Loan Association, Chicago, has purchased a UNIVAC 1004 Card Processor, which will be used for mortgage and savings accounting. Delivery is scheduled for early 1963.

CIRCLE 107 ON READER CARD

● A two-week course in information storage and retrieval is being offered by the University Extension of the University of California, Los Angeles. Subjects to be covered include introduction and summary of present system design work, mathematical theory of file organization, application of mathematical tools, and system implementation. Fee for the course, which starts on October 29, is $200. Additional information may be ob-

**NORTH ELECTRIC** VISIONEERING ACHIEVEMENT SINCE 1884 . . . INTO THE FUTURE




# POWER SUPPLY SYSTEMS

## ...from the finest, most complete facilities

The modern, extensive facilities of the Power Equipment Division (**PECO**) of North Electric are the largest in the industry. They are devoted to the design, engineering and manufacture of power supplies to each customer's individual requirements.

**PECO ENGINEERING** will custom design a power supply equal in reliability to your own equipment.

**PECO LABORATORY and PROTOTYPE SERVICE** will place a custom built prototype in your hands in the shortest possible time.

**PECO PRODUCTION** will keep pace with your production requirements.

**AND PECO** custom tailored power supplies are competitively priced.

**PECO** custom-engineered power devices, from low power miniaturized and transistorized power supplies to unusually high current capacity battery chargers, range from unregulated AC & DC to highly-regulated AC & DC for commercial, industrial and military computer, communications and automatic control systems.

The **PECO** capability, recognized since 1935, is at your service. We welcome your inquiry. Write for brochure "Custom Engineered Systems."

OTHER DIVISIONS:
DEFENSE SYSTEMS — *Electronics • Command and Control Systems*
ELECTRONETICS — *Switching • Controls • Computers • Components*
TELECOM — *Public Telephone Exchanges • Associated Equipment*

NORTH

**NORTH ELECTRIC**
**POWER EQUIPMENT DIVISION • GALION, OHIO**

CIRCLE 30 ON READER CARD



tained from H. L. Tallman, Physical Sciences Extension, Room 6532, Engineering Building, University of California, Los Angeles 24, Calif.

● Aluminum Company of Canada Ltd., Montreal, has received an ITT 7300 ADX Automatic Data Exchange System, the first of its type to be put into operation in North America. The system will be used to handle all administration messages, including sales orders, inventory control, materials purchasing, and scheduling shipping and delivery from ALCAN's aluminum plants and offices in Canada, the U.S. and London.

● An IBM 7070-1401 system has been put into operation by the American Cyanamid Co., Wayne, N. J. The system is being used for accounting, payroll, cost accounting and distribution statistics, as well as scientific research, product development and for the preparation of operation statistics.

● Three S-C 1900 Direct View Display consoles have been sold by General Dynamics/Electronics to Lockheed Missiles and Space Co., Sunnyvale, Calif. In operation, information from an Agena space vehicle will be transmitted to computers on the ground. Input from the computers will be displayed on the 1090 for visual interpretation. The consoles carried a reported total sales price tag of $85,000.

CIRCLE 108 ON READER CARD

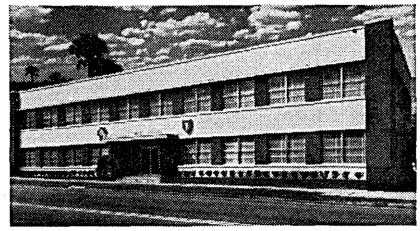
● Datatrol Corp., Silver Springs, Md., has introduced its Operating and Control System (OPCON) programming package, designed for the IBM 1401. Six subsystems make up the package: generalized program library, library maintenance, single step processing, procedure compiling, production, and procedure maintenance.

CIRCLE 109 ON READER CARD

● Gross sales recorded by Operations Research Incorporated for the first seven months of fiscal 1961-1962 totaled $2,700,000, an increase of 104% over the corresponding figure of a year ago of $1,323,000.

● A Joint Task Group has been formed by the European Computer Manufacturers Association and the Business Equipment Manufacturers Association of America to study the various MICR codes. The study will result in a recommendation for a

BLUE CROSS OF FLORIDA, INC. ■ FORMSCARD ■ BLUE SHIELD OF FLORIDA, INC.



# FORMSCARD® HELPS FLORIDA BLUE CROSS & BLUE SHIELD GAIN MAXIMUM OPERATING EFFICIENCY

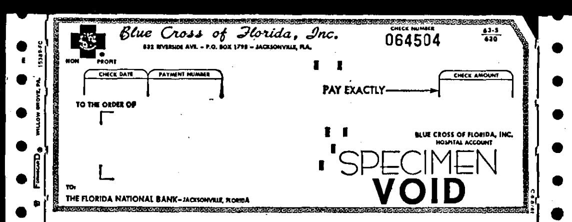
This famous non-profit medical service is another well-satisfied FORMSCARD user.

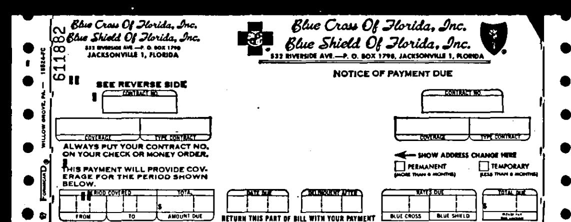
Blue Cross and Blue Shield of Florida, Inc., depend on FORMSCARDS **pre-punched** voucher-checks and payment notices to efficiently process upwards of 9,000 claims a week and to bill company and individual subscribers each month. These forms are operated over the IBM 1403 printer.

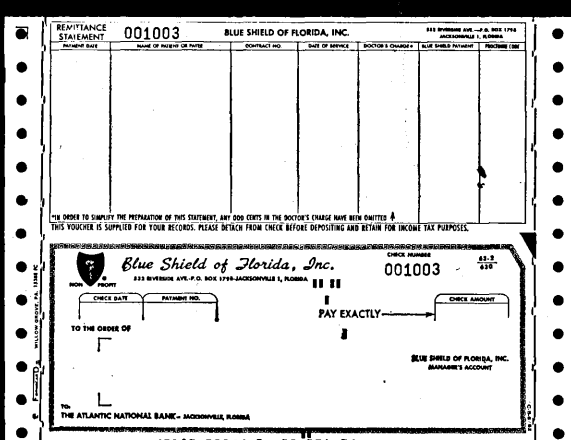
Not only are FORMSCARDS available PRE-PUNCHED, they are also made with vouchers of any length, side, top or bottom. They separate easily, manually or on any burster, and can be part of a multiple part set. Send for your samples today.

# FORMSCARD®

still the only continuous tab card with NO medial waste strips




Call
write or wire

**Forms inc**
WILLOW GROVE, PA.
**Phone: OLdfield 9-4000/Area Code 215**

Manufacturers of line-hole continuous business forms. Samples on request—sales representatives in principal cities.

CIRCLE 31 ON READER CARD

common language code or codes for the benefit of users and manufacturers.

● City Bank and Trust Co., Jackson, Michigan, will install a GE 225. The system, scheduled for delivery in July, 1963, will be used for handling 21,000 demand deposit accounts at the bank's seven offices.

● A device called "Madaline I" by its inventors at the Stanford University Electronics Laboratories has been programmed to balance a cardboard broom and read letters of the alphabet. Larger Madaline systems might be taught to drive a car or fly a plane. The machine, which uses 102 memistors — electrochemical resistors with memory—is tied in with an IBM 1620. A 1500-memistor Madaline is in the planning stage, which is expected to provide the 1620 with a greater adaptive potential than the IBM 7090, at one-tenth the cost.

● The IBM Supplies Division has established a mag tape testing center at Poughkeepsie, N. Y., which will examine tape for wear, noise, density and magnetic characteristics. If defects are found that cannot be removed, the entire reel will be rejected. The center will offer IBM customers in the eastern third of the country a retesting service for used IBM tape.

● A general purpose computer made by the Control Systems Division of Daystrom, Inc., will be used for automatic control of a coal-fired steam plant in Goldsboro, N. C. Designed specifically for use as the central unit of an industrial process control system, the computer has 12K words of random access, magnetic core memory and 86K word drum memory. Command structure is single address with word time of 37 microseconds; command repertoire is 136 instructions.

● An MIT neurologist is using an owl and a GE-225 to study pupillary systems, aimed at developing a clearer understanding of human nerve and brain disorders such as Parkinson's disease. In the experiments being conducted, the computer is programmed to direct a light beam of a certain intensity on the owl's pupil. After the light passes through the pupil, it stimulates the photosensitive cells of the retina, sending electric nerve impulses to the brain. The brain analyzes the disturbance and sends a signal back to the pupil, causing it to contract. The time required for this circuit to be completed and the degrees for dilation or constriction are recorded and analyzed by the computer.

● The National Geographic Society, Washington, D. C., has ordered a UNIVAC III. The society will use the computer to handle circulation records for National Geographic Magazine subscribers, process members' accounts, and for information retrieval tasks. Delivery is scheduled for 1963.

● A program for tabulating data used for market research studies has been made available by Computers for Industry and Business, New York City. The program provides for arithmetic calculation code conversion on input data, complex cross tabulation, calculations on the tabulated data, and formatting of tables. Written for the IBM 7090, the program is said to be general enough to handle most tabulation requirements.

CIRCLE 110 ON READER CARD

● An October 30th deadline has been set for papers to be presented at the March, 1963, conference on "Faster Information Processing," sponsored by the West Coast Subcommittee, Computing Devices Committee, American Institute of Electrical Engineers. The conference will be held March 15-16, 1963, at Cal Tech, Pasadena, Calif. Papers should emphasize improvements in working devices and techniques, and significant new concepts within this field. Four copies of a 500-word abstract should be sent to S. Nissim, Program Chairman, Thompson Ramo Wooldridge, 8433 Fallbrook, Canoga Park, Calif.

● First delivery of a UNIVAC SS II has been made to the Guide Lamp Division of General Motors. The computer replaces a UNIVAC SS I Card System, and will be used for production control, critical path schedule for model change-over planning, payroll, and general accounting. UNIVAC reports 30 SS II systems are on order.

● A $50,000 contract for research on a prototype "woven screen" computer memory plane for possible use in the Navy's Navigational Satellite System has been awarded to the RW Division of Thompson Ramo Wooldridge, Inc., by the Applied Physics Laboratory of Johns Hopkins University. The memory is made by weaving strands of bare copper wire into a screen mesh, creating the equivalent of a magnetic core at each mesh point. The screen is plated with a highly remanent magnetic material, and read and write leads are brought to each mesh point to control or sense the state of magnetization.

● Tulane University and IBM are undertaking a joint effort to study methods and systems for processing large volumes of medical information. Objectives include the use of data processing equipment to analyze complex physiological data from hospital patients, and the investigation of a method of simplifying medical records-keeping by attempting to design a standard medical record form that would be processable by machine.

● Computer Dynamics Corp., Silver Springs, Md., has received a contract to operate the data processing facility for NASA's Launch Operations Center, Cocoa Beach, Florida. The firm will be responsible for the design, programming and computer application implementation, plus the development of procedures, operation and coordination of maintenance for data processing equipment. The contract was reported to be for a quarter of a million dollars.

● The U. S. Atomic Energy Commission has awarded a contract to C-E-I-R, Inc., for counselling and assistance in establishing administrative, accounting and security functions centering around an IBM 1401 at AEC headquarters in Germantown, Md. The contract covers the "breaking-in" period and calls for system analysis, programming, and conversion of existing documents into computer-compatible form, including key-punching of cards as required.

● A stock replenishment system has been developed by the J. C. Penney Co. which provides the firm with control over a flow of some 20 million stock items a week through its 1,700 stores in 48 states. Called SASC (Semi-Automated Stock Control), the system employs an NCR 315. Prepunched tags from merchandise are removed when the items are sold, and sent to Penney's data processing center in New York twice weekly. This information is key punched and read into the computer. Every two weeks the data is processed for each store to determine what orders are to be written.


P. S. Computape doesn't really talk, of course. But in a computer, Computape *reliability* will deliver its own message. New COMPUTAPE, the premium quality computer and instrumentation tape, is the product of the only company devoted exclusively to the manufacture of quality tapes for data processing and instrumentation. *Investigate new Computape today. Better still, immediately.*

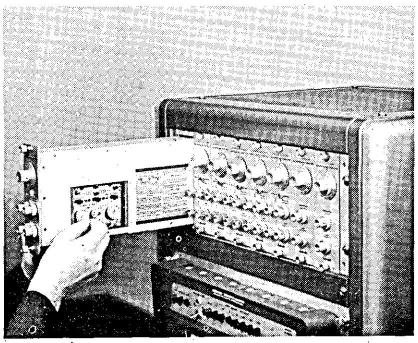
**COMPUTRON INC.**
122 Calvary Street, Waltham, Massachusetts

Send today for your free magnetic tape booklet.

# NEW PRODUCTS

### direct-writing system

This eight-channel, model RF 1783 70, has a basic system sensitivity of 50 mv/chart division and can be used with any combination of plug-in pre-amplifiers from the RD 4200 series. The system includes a vertical writing



oscillograph which has 12 chart speeds from 0.05 mm/sec/to 200 mm/sec. Approximate price, including preamplifiers, is $13,000. BRUSH INSTRUMENTS, 37th and Perkins, Cleveland 14, Ohio. For information:

CIRCLE 200 ON READER CARD

### datanet-15

This 15-channel data transmission controller allows the GE 225 gp computer to automatically receive and transmit information over available telephone facilities, either in-plant or over nation-wide communication networks. Datanet-15 accepts information at up tc 3,000 words per minute. Price range is from $30,000 to $38,000. GENERAL ELECTRIC CO., COMPUTER DEPT., Deer Valley Park, Phoenix, Arizona. For information:

CIRCLE 201 ON READER CARD

### scanning system

The QSS-1 scanning system is able to transmit data bits over a single dc circuit or a single frequency shift channel. The system, which features modular components which may be expanded to handle 40 points, is compatible with existing data transmission and peripheral equipment. QUINDAR ELECTRONICS, INC., 5 Lawrence St., Bloomfield, N.J. For information:

CIRCLE 202 ON READER CARD

### control system

This anticipator-computer control system is able to anticipate the possibility of variations in the quality of process output through pre-sensors, and can compute the conditions which will maintain stability and also acts to create those conditions. The control system eliminates the time lag between corrections which were started by variations in output quality and corrective action at the source by predicting the effects of anticipated changes by means of the computer section. BAILEY METER CO., 1050 Ivanhoe Rd., Cleveland 10, Ohio. For information:

CIRCLE 203 ON READER CARD

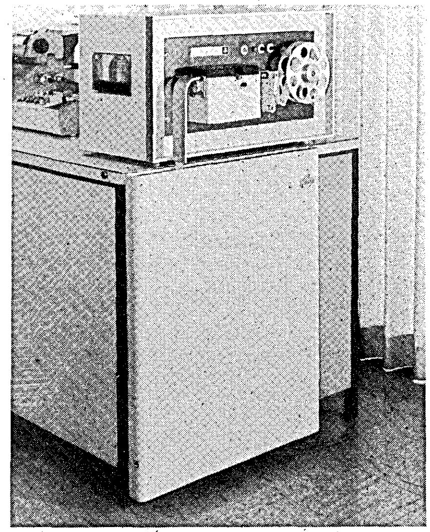
### digital to analog converter

The CG-13-1 is of modular construction and designed for driving devices including plotters, cathode ray displays, meters and oscillographs and for providing inputs to analog computers. GULTON INDUSTRIES, INC., 212 Durham Ave., Metuchen, N.J. For information:

CIRCLE 204 ON READER CARD

### paper tape reader

The Facitape 510 is able to read five through eight channel tape at a basic



speed of 600 characters per second, or approximately 300 characters per second when stopping on a character with the RECOMP III. The 510 includes such features as printed circuitry, simplified load and unload

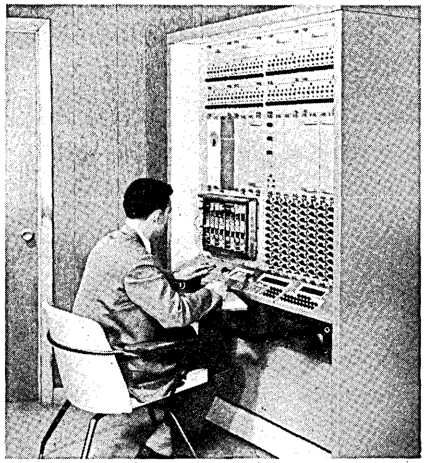
## NEW PRODUCTS . . .

characteristics and transistorized electronics. Price is $4,750. AUTONETICS INDUSTRIAL PRODUCTS, 3400 E. 70th St., Long Beach 5, Calif. For information:
CIRCLE 205 ON READER CARD

### memory system
This word organized memory system is capable of a read/restore cycle in one usec and has an access time of .55 usec for full cycle operation; .50 usec for half cycle operation. The memory system is available in all standard modes of access. INDIANA GENERAL CORP., ELECTRONICS DIV., Keasbey, N.J. For information:
CIRCLE 206 ON READER CARD

### network design
Each master slice circuit bar contains about 30 resistors, capacitors, diodes, and transistors which can be interconnected in hundreds of different ways to obtain many customized circuits for semiconductor networks. TEXAS INSTRUMENTS, INC., P.O. Box 5012, Dallas 22, Tex. For information:
CIRCLE 207 ON READER CARD

### analog computer
The AD-2-64PBC features central pushbutton control and monitoring and offers four control modules includ-



ing the computer control module, voltmeter module, input-output module and address selector module. Prices begin at $15,000. APPLIED DYNAMICS, INC., 2275 Platt Rd., Ann Arbor, Mich. For information:
CIRCLE 208 ON READER CARD

### card keypunch
Model 80 EM punch has been designed to handle the standard 80-column IBM card, but adapters for any card under 80 columns are also available. Special models can be purchased for cards longer than 80 columns. The EM punch is provided with a standard numeric key set to punch numeric or, with multiple key depressions, alphabetic and special characters. ELECTROMECHANICS CORP., 502 Sherbrook Dr., High Point, N.C. For information:
CIRCLE 209 ON READER CARD

### multiplexer series
This solid state multiplexer series features low and high level multiplex analog input with high speed random channel selection. The models are available in three different series and the control signals, size compatibility, power requirements and switching signals of all models are the same. REDCOR CORP., P.O. Box 1031, Canoga Park, Calif. For information:
CIRCLE 210 ON READER CARD

### pj-2 printer
This monitoring instrument prints elapsed time and functional time of all edp and eam systems on a tab card and also provides total functional time and total elapsed time on a nonresettable totalizer. STANDARD INSTRUMENT CORP., 657 Broadway, New York 12, N.Y. For information:
CIRCLE 211 ON READER CARD

### clock oscillator
The DBCO1171, which was designed primarily as a master clock for digital systems, includes a crystal-controlled

Butler Oscillator with associated buffer amplifier. Nominal frequency stability is ±.01% over a temperature range of —40°C. to +100°C. SOLID STATE ELECTRONICS CORP., 15321 Rayen St., Sepulveda, Calif. For information:

CIRCLE 212 ON READER CARD

### memory units

Model LSM memory system has a read/restore or clear/write cycle time of one usec and can perform the function of buffering as a sequential interlace buffer or block buffer at rates to 1.5 megacycles. Model CCM is a coincident current memory system with speeds to six usec as a random access memory or four usec as a buffer. DAYSTROM, INC., 8703 La Tijera Blvd., Los Angeles 45, Calif. For information:

CIRCLE 213 ON READER CARD

### differential amplifier

This 10 kc amplifier, model 461, has an input impedance of 1000 megohms, low full bandwidth noise of four $\mu$ v referred input, and a common mode rejection of 120 db from zero to 60 cps for all gain settings from one to 1,000. The 461 has a gain accuracy of 0.1% and provides fixed gain settings of 1, 3, 10, 30, 100, 300 and 1,000, with a precision vernier for fine adjustments. PACKARD BELL COMPUTER, 1905 Armacost Ave., Los Angeles 25, Calif. For information:

CIRCLE 214 ON READER CARD

### mag tape delay

Model 102-DR is a solid state unit which is able to provide a continuously variable time lag of 0.3 to 10.0 seconds wherever on-line signal delays are needed. The system has a linearity of 0.2 per cent of full scale and a frequency response of 0-400 cps at 7½ ips and 0-100 cps at 1⅞ ips. Price is $1,965. MNEMOTRON CORP., Pearl River, N.Y. For information:


CIRCLE 215 ON READER CARD

### pinfeed labels

These Macomatic pinfeed pressure sensitive labels have been designed for use on various data processing and other electronic business machines for compiling mailing lists, coding, sales, and inventory records. They are available in three standard sizes in both fanfold and roll form. MAY TAG AND LABEL CORP., 111 West 19th St., New York 11, N.Y. For information:

CIRCLE 216 ON READER CARD

### operational amplifiers

Model 2020 operational amplifier functions with either negative or positive outputs and includes open loop gain, greater than 100,000; gain-bandwidth product, 200 KC; common mode rejection ratio—power supply, 200:1; maximum lead at output swing of ±10 volts, 10 K. The 2021 chopper amplifier is a low drift, low noise, solid state companion plug-in unit to extend the range of the 2020. The 2022 power booster is used where greater output current is required for meeting the driver requirements of various process and machine control applications. WESTON INSTRUMENTS DIV., DAYSTROM, INC., 614 Frelinghuysen Ave., Newark 14, N.J. For information:

CIRCLE 217 ON READER CARD

CIRCLE 98 ON READER CARD

# people IN DATAMATION

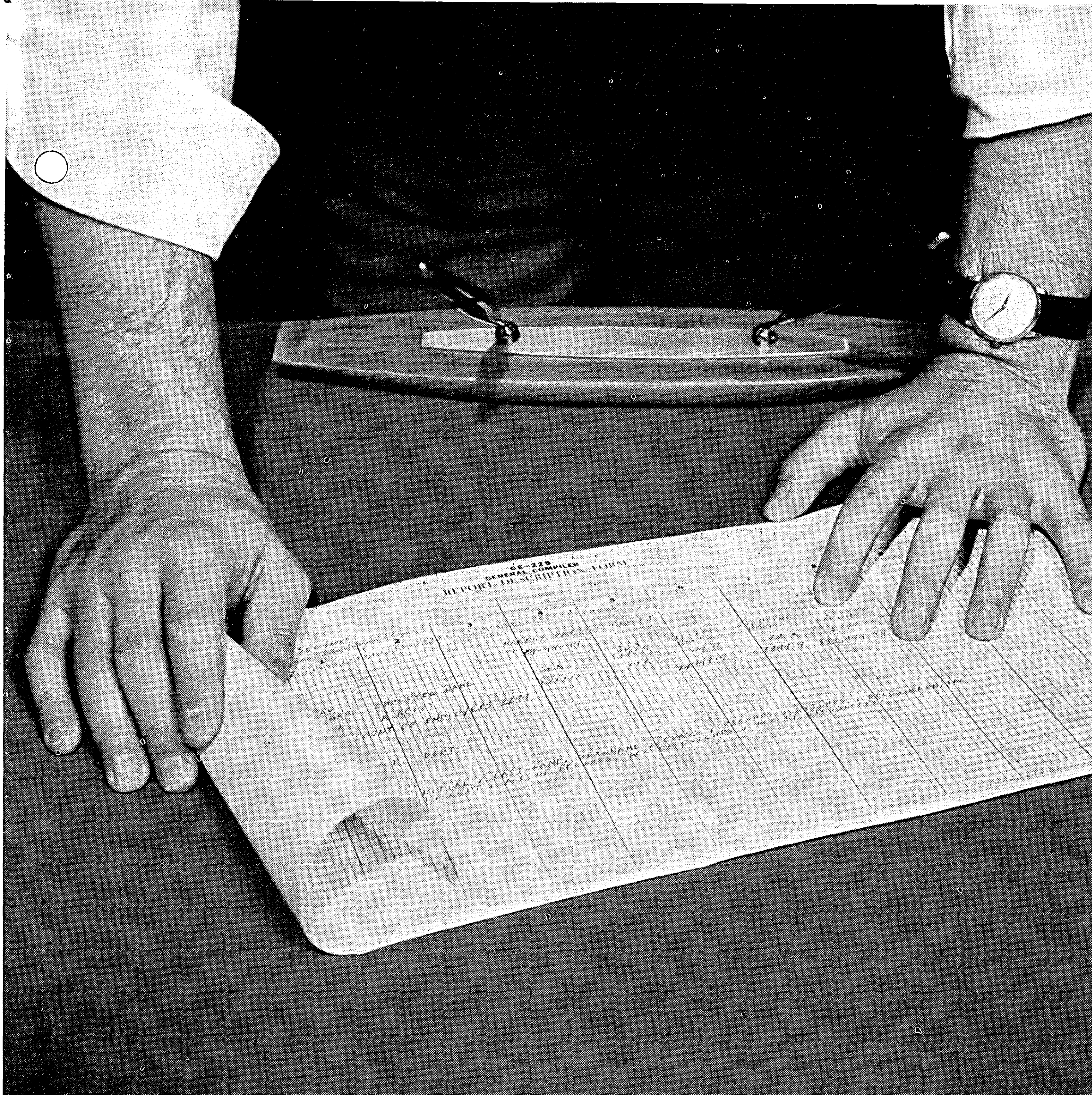
■ Leonard F. Longo, formerly with General Electric, Evendale, Ohio, is now with Douglas Aircraft Co., Long Beach, Calif., where he will assume the position of commercial systems coordinator. He will continue as SHARE's SURGE Project Manager.

■ Henry L. Pope is now manager of Development Planning at Thompson Ramo Wooldridge's RW Division, where he will be responsible for long-range market and sales plans. He was formerly chief, Development Planning, at Nortronics Division's Electronics Systems.

■ Two new appointments have been announced by RCA's Data Systems Division, Van Nuys, Calif. Henry M. Watts has been appointed manager of systems engineering, where he will direct the division's systems engineering activities, including advanced systems planning, operations and requirements analysis, systems synthesis and optimization. Charles A. Wolf was named manager, operations, of the division. He will direct marketing, operations control, the aviation equipment department and program coordination for the division.

■ Jackson W. Granholm has been elected president of Mellonics A-V, Inc., Tucson, Arizona, firm engaged in the concept, preparation and production of various audio-visual aids used in aerospace, computing, communication, and instrumentation. Granholm succeeds the late William D. Bell. Granholm, a computer consultant, was formerly editor and publisher of Computing News, and is a frequent contributor to DATAMATION.

■ Peter M. Kelly has been appointed associate director of the Philco Scientific Laboratory, Blue Bell, Pa. He was formerly associated with the Aeronutronic Division of Ford Motor Co., and has served on the staff of Hughes Aircraft Co. and Douglas Aircraft Co. While with Aeronutronic, he contributed to the development of a self-organized electro-optic instrument capable of recognizing and classifying simple three-dimensional objects.



# This program speaks for itself

It's the form used with General Electric's new report writer for the GE-225. Your management can read it, understand it, OK it. Just like that. And it's easy to write. Takes only a fraction of the time you'd spend on hand coding. That's time and work saved all the way around.

This new method of report planning takes the tedium out of special reports. You can document, debug, make changes—all at the source level. Just fill out the printed form in English. Lay it out just as you want the report to appear. It's ready to show management for a final check. Then hand it to a key-punch operator. No flow charts or hand coding necessary for formatting, data organization, or printout.

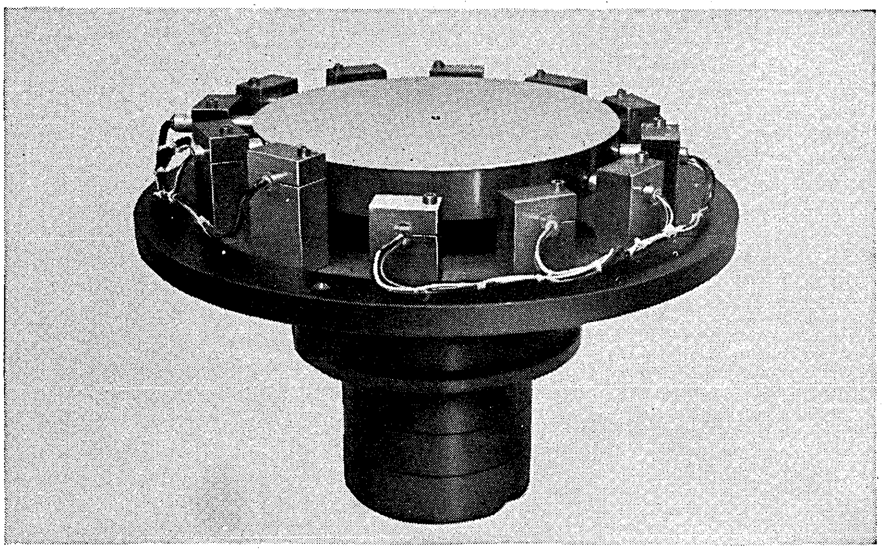
The General Electric report writer system is another capability of GECOM—the most powerful and versatile computer programming aid. Our brochure CPB-224 will tell you more about it. Write General Electric Computer Department, Section J10, Phoenix, Arizona.

*Progress Is Our Most Important Product*

**GENERAL ELECTRIC**

CIRCLE 39 ON READER CARD

# TWO LOW-COST WAYS TO SOLVE YOUR DATA STORAGE PROBLEMS



**FOR UNDER $500***

**Bryant's C-675 drum offers . . .**

- 12 tracks
- 2450-bit track density
- 1800 rpm
- 73.5 kc pulse repetition frequency
- 20 mv minimum playback signal
- 10-year bearing life
- 12 pounds total weight
- 6-inch drum diameter
- ¾-inch drum length

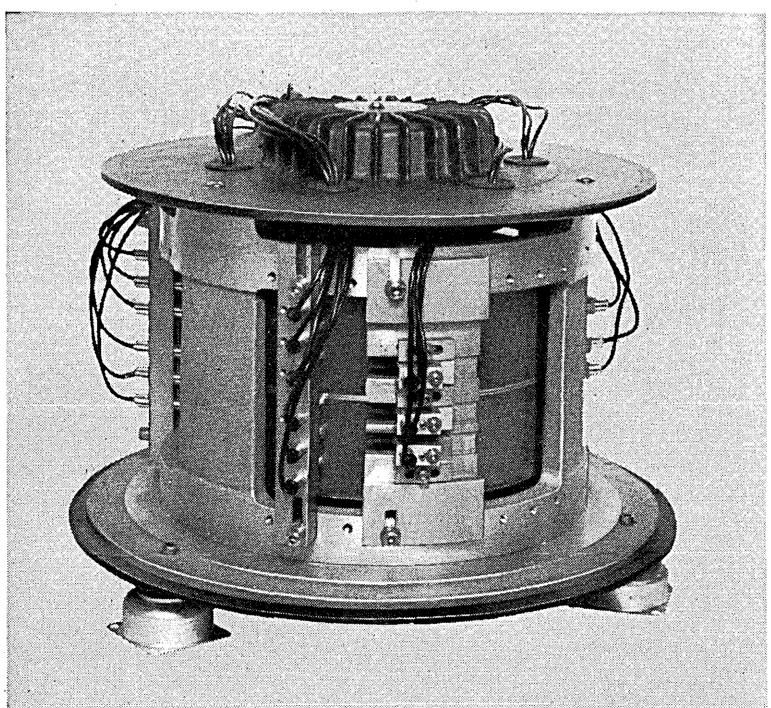
*drum alone—$298
drum with 12 heads—$478

**FOR UNDER $2000***

**Bryant's C-105 drum offers . . .**

- 54 tracks
- 3000-bit track density
- 1800 or 3600 rpm
- 90 kc pulse repetition frequency (1800 rpm)
  180 kc pulse repetition frequency (3600 rpm)
- 30 mv minimum playback signal (1800 rpm)
  50 mv minimum playback signal (3600 rpm)
- No less than 14 db minimum signal to maximum noise ratio
- 10-year bearing life
- 55 pounds total weight
- 10-inch drum diameter
- 5-inch drum length

*drum alone—$780
drum with 50 general storage heads—$1878



62-44CP

BRYANT **COMPUTER PRODUCTS**

852 Ladd Road • Walled Lake, Michigan • MArket 4-4571

*A DIVISION OF EX-CELL-O CORPORATION* XLO

CIRCLE 43 ON READER CARD

# NEW LITERATURE

**ELECTRONIC MODULES:** This 16-page booklet discusses the manufacture of high-density welded electronic packages and various types of welded modules. Also included are assembly steps, weld-schedule development, process control and quality assurance procedures. ENGINEERED ELECTRONICS CO., 1441 E. Chestnut Ave., Santa Ana, Calif. For copy:
**CIRCLE 130 ON READER CARD**

**CRAM:** Features and operating principles of the Card Random Access Memory, which is available with the 315 computer series, are offered in this 28-page brochure. NATIONAL CASH REGISTER CO., DATA PROCESSING SYSTEMS, South Main and K Sts., Dayton 9, Ohio. For copy:
**CIRCLE 131 ON READER CARD**

**EDP COMPUTER UTILIZATION REPORTING:** This eight-page booklet details the functions of the Data-Graph III, the Data-Register LL, the Data-Printer L, and the Data-Meter I. Included is the layout of a typical 1401 system and how it can be monitored by these various instruments. STANDARD INSTRUMENT CORP., 657 Broadway, New York 12, N.Y. For copy:
**CIRCLE 132 ON READER CARD**

**TAPE PRIMER:** "Facts You Should Know About Magnetic Tape" is a simplified, 10-page booklet which offers general information about magnetic tape for computer use and also clears up a number of erroneous ideas on tape manufacture. COMPUTRON INC., 122 Calvary St., Waltham 54, Mass. For copy:
**CIRCLE 133 ON READER CARD**

**DATANET-15:** This data-transmission controller for use with the 225 gp computer is presented in an illustrated pamphlet which explains key operating features of multiple-channel, two-wire communication equipment and computer-to-computer automatic data exchange. GENERAL ELECTRIC CO., COMPUTER DEPT., Deer Valley Park, Phoenix, Ariz. For copy:
**CIRCLE 134 ON READER CARD**

**INSTANT VERSALOGIC:** A four-page technical bulletin presents the Versalogic system concept and includes circuit descriptions and application notes with diagrams. System design data are covered in chart form and condensed specifications are given. DECISIONAL CONTROL ASSOC., INC., 644 Terminal Way, Costa Mesa, Calif. For copy:
**CIRCLE 136 ON READER CARD**

**PRINTER SYSTEM:** The operating characteristics, features and suggested usage for the 580 are included in this illustrated bulletin. ANELEX CORP., 150 Causeway St., Boston 14, Mass. For copy:
**CIRCLE 137 ON READER CARD**

# NEW FIRMS & mergers in DP

Mark Systems, Inc., is a new firm located in Los Altos, California, and will engage in the development of electro-photo-optical products for application in the areas of graphic data acquisition, reduction, analysis, and display. Bernard P. Marcus was named president of the company, which will function as a subsidiary of Allied Research Associates, Inc., Boston.

CIRCLE 100 ON READER CARD

CIRAD (Corporation for Information Systems Research and Development), has been formed in Philadelphia. Principal officers of the firm are Paul Colen, president and director of systems planning, John J. Ogle, VP and director of systems operations, and Robert S. Barton and Jackson W. Granholm, consultants.

CIRCLE 101 ON READER CARD

A new firm, Inforonics, Inc., has been formed in Maynard, Mass., and will engage in consulting, design, and research in mechanized information systems. Heading up the organization are Lawrence F. Buckland, president, and William R. Nugent, vice president. Both were formerly with the Information Sciences Division of Itek Laboratories.

CIRCLE 102 ON READER CARD

Computer Equipment Corp., Los Angeles manufacturer of electronic cable and quantizer instruments, has been acquired by Technical Systems, Inc. Elected president of CEC was Walter E. Peterson.

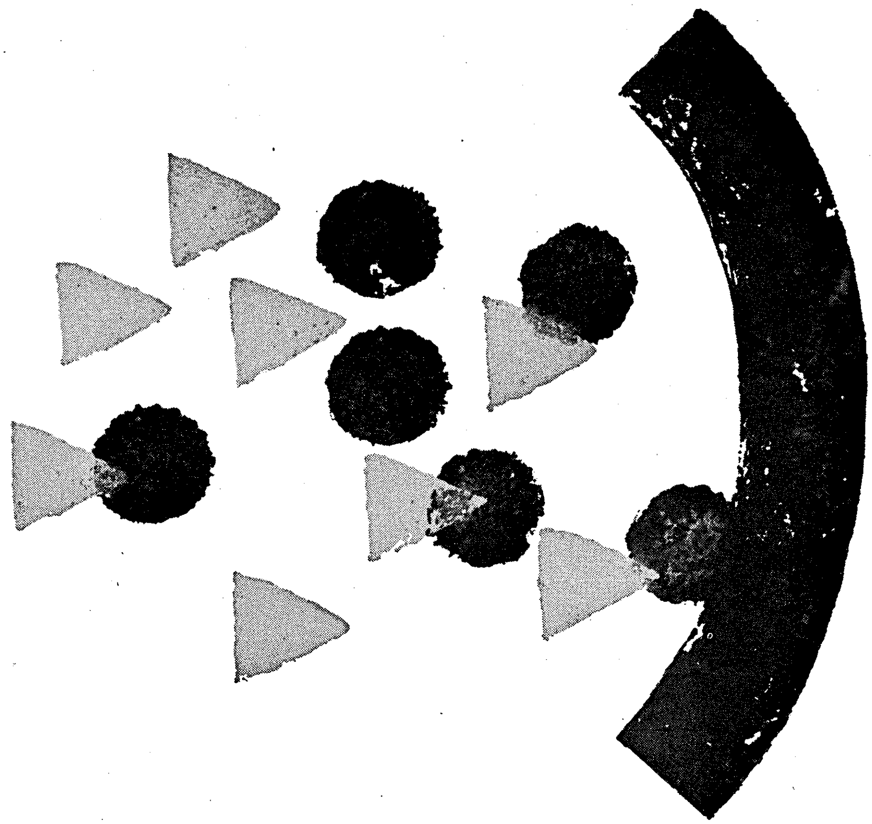
CIRCLE 103 ON READER CARD

Alan L. Stewart, formerly a consultant to GE's Computer Department, has organized his own firm, A. L. Stewart and Associates, Phoenix, Arizona. The firm will be available for consultation in the fields of PERT/CPM, data processing, and operations research.

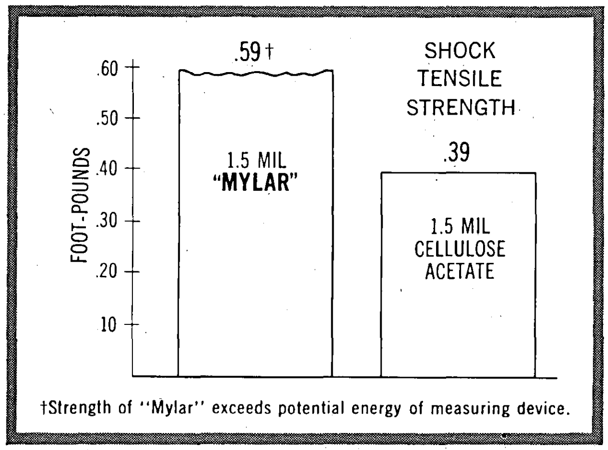
CIRCLE 104 ON READER CARD

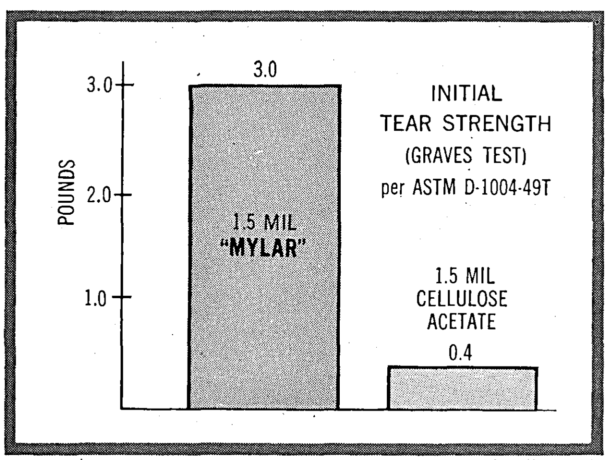
Techniserv Corporation has been established in Los Angeles to provide field maintenance services for manufacturers and users of data processing systems. Walter Brown, a former vice president of Benson-Lehner Corp., is founder and president of the firm. Offices in five cities throughout the country are planned.

CIRCLE 105 ON READER CARD



# GUARD AGAINST DOWNTIME WITH RELIABLE TAPES OF MYLAR®



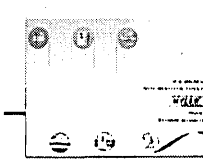




Every minute of downtime costs you dollars. That's why the reliability of your magnetic tape is so important. Tapes of Mylar*, resist breaking from sudden starts and stops since they have high shock-tensile strength . . . and they have 7 times the initial tear strength of acetate tapes.

"Mylar" adds to the reliability of playback. It resists the cupping, swelling and shrinking which can reduce intimate head contact and cause read/write errors. Age, storage conditions and repeated playbacks won't affect "Mylar" either.

The cost of the data . . . the cost of the equipment . . . the cost of time *all* demand reliability. Get it with tapes of "Mylar". Send coupon for free booklet of comparative test data and judge for yourself. Du Pont Company, Film Department, Wilmington 98, Delaware.

*Du Pont's registered trademark for its polyester film.



DU PONT
REG. U. S. PAT. OFF.
Better Things for Better Living . . . through Chemistry

DU PONT
MYLAR®
POLYESTER FILM

E. I. du Pont de Nemours & Co. (Inc.)
Film Department, Room #12, Wilmington 98, Delaware
Please send free, 12-page booklet of comparative test data to help me evaluate magnetic-tape reliability.

Name__________ Position__________

Company__________

Address__________

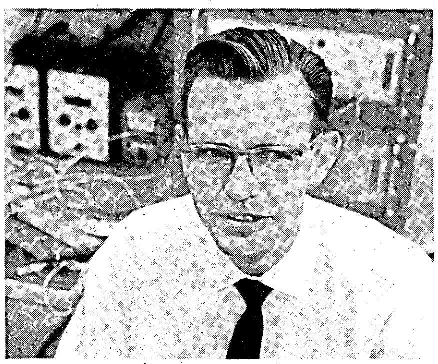
City__________ Zone_____ State__________

CIRCLE 48 ON READER CARD

*(Advertisement)*

# Mean Time to Failure: 2,000 Computer Hours

According to Robert M. Beck, V.P.—Engineering for Scientific Data Systems, Inc., "it is now logical to expect a full year of 40 to 50 operating hours per week from a digital computer, without failure." Beck's opinion, based on the reliability program for Scientific Data's 900 Series computers, represents "a balanced philosophy going far beyond a component count numbers game."



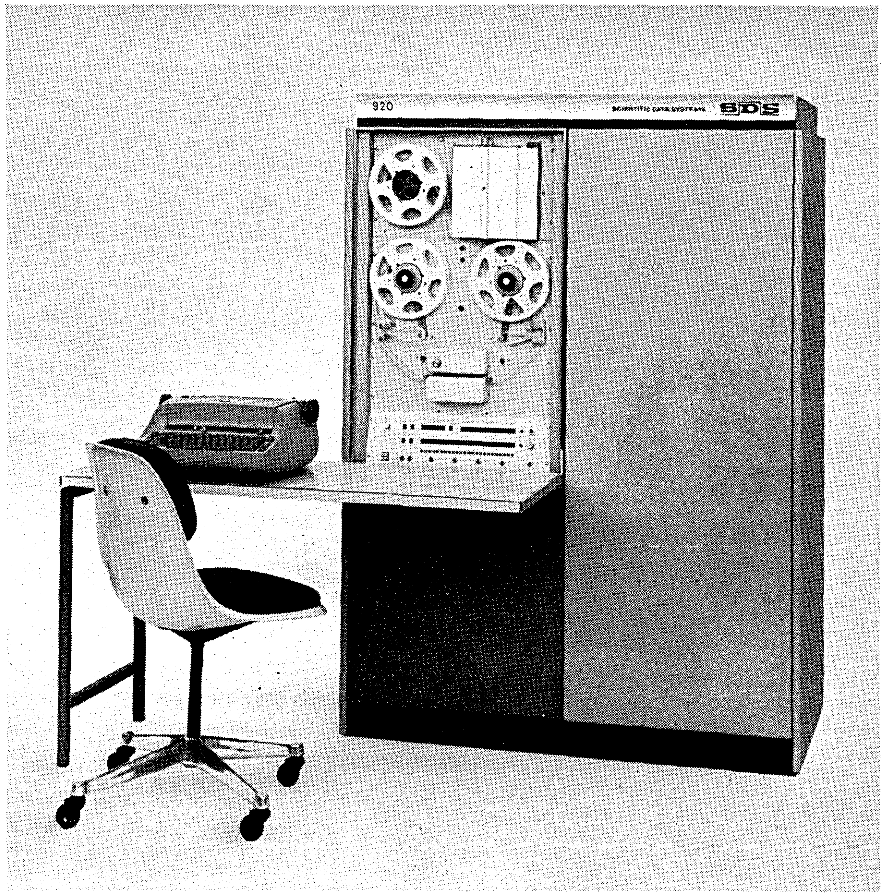
BECK

**Circuit design is critical.** According to Beck, "we consider circuit noise as the paramount design consideration with respect to computing reliability and therefore use noise rejection networks at both the input and output of all active circuit elements. Further, our circuits have wide built-in tolerances — for example, voltages may vary by as much as 36% without affecting computer operation. Silicon transistors allow for operation to 130°F. to eliminate the need for inherently unreliable air conditioners."

**Reduced component count.** Beck continued, "once it is established that circuits are reliable and tolerances are wide, reduction of component count becomes a major factor. At SDS, we use new techniques that reduce component count by a factor of from 3 to 5 with a corresponding increase in reliability."

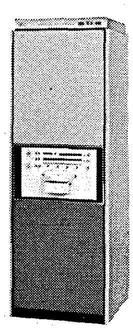
**Input/Output reliability.** Beck further stated: "Digital I/O equipment is unreliable by its very nature (i.e. the failure ratio between a paper tape punch and a computer is 1,000,000 to 1). Therefore, the computer, with its great inherent reliability, is the best available entity for detecting and compensating for I/O error. Our 900 Series input/output schemes are designed so that the computer operates in closed loop synchronization with its I/O equipment, automatically monitoring every input signal, rejecting errors and initiating recycling of correct data. In addition, we have made design allowances for input signals with the worst possible noise, amplitude, chatter and timing."

"We are certain that our new 900 Series Computers place a new standard on predictable computing reliability."




# cost carver

**The new SDS 920 General Purpose, Solid State Digital Computer** cuts both the original cost and the operating costs of scientific/engineering computation and systems integration. It has all the speed (16 μsec. add — 32 μsec. multiply) and operating features found only in much more expensive equipment. This single address, core memory computer has five distinct built-in input/output systems, including a high speed buffer. A comprehensive software package, including FORTRAN II with magnetic tape statements is available. The 920 is priced at only $89,000.



**The SDS 910, a smaller computer,** costs only $41,000 (half the price of comparable machines) yet shares the principal features of the 920. The 910 is designed for on-line control and real time systems work, as well as general purpose computing.

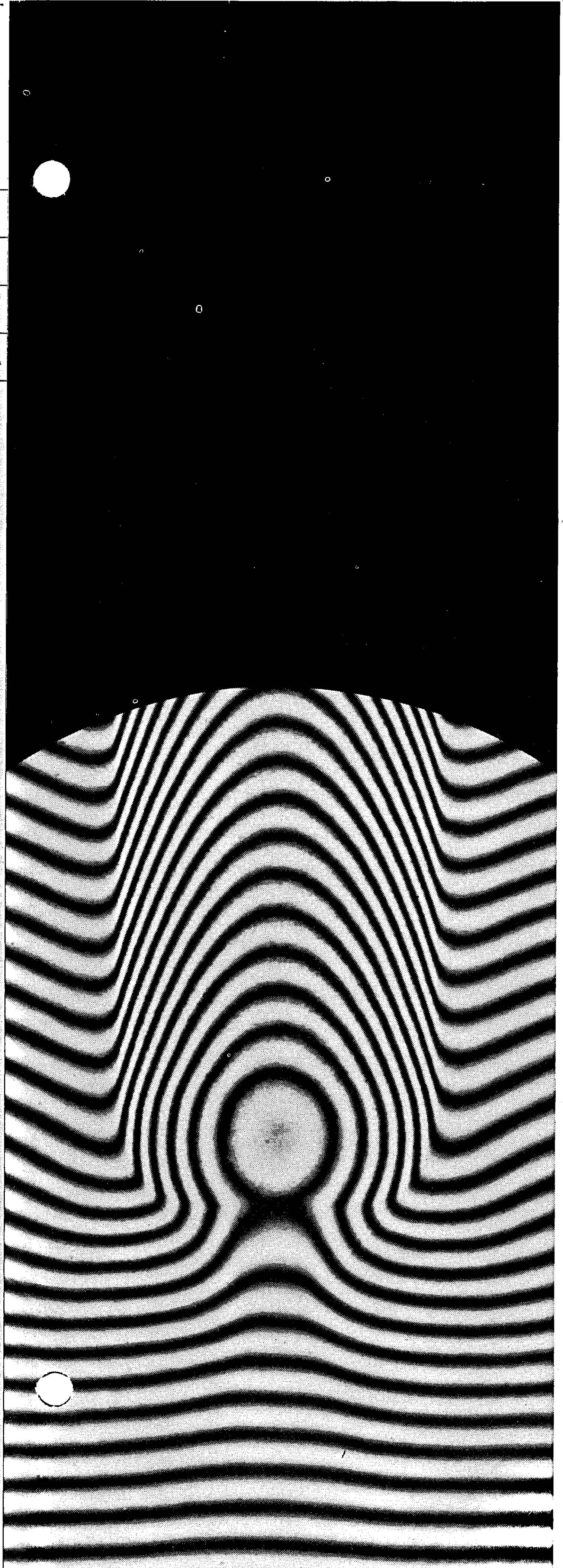
**The SDS 910 and 920 computers and their optional peripheral equipment are described in the SDS 900 Series brochure. Request your copy today.**

SCIENTIFIC DATA SYSTEMS

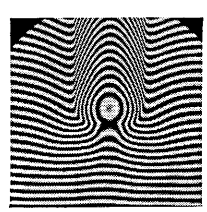
SDS

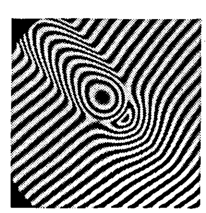
1542 FIFTEENTH STREET • SANTA MONICA, CALIFORNIA

CIRCLE 53 ON READER CARD









Ball track across copper crystal is narrower, harder and has higher friction in a cube diagonal direction (right interferogram) than in a cube edge direction.

WILEY BOOKS

*The first book about ALGOL . . .*

# A GUIDE TO ALGOL PROGRAMMING

*By* DANIEL D. MCCRACKEN, McCracken Associates Inc. This is the first book written specifically about Algol Programming—an algorithmic language that is an advanced or refined version of Fortran. Algol, which is already widely used in Europe, may soon become the international, *scientifically oriented* programming language. Written for and completely understandable to the beginner who does not know Fortran, algorithms, or the strong and weak points of a computer, this book is a slow, easy introduction and exposition to Algol, and it also tells the student what scientific computing *in general* is about. The approach is also good for a person who knows computers in general and wants to learn Algol.

As a textbook, *A Guide to Algol Programming* can be used for courses in programming for beginning students (at any level in science or engineering) who will be using one of the large standard computers (650-1401). It can also be used for numerical analysis courses which have a practical or programming slant. The only prerequisite is some familiarity with mathematical notation. 1962. 112 pages. $3.95.

IVERSON: **A PROGRAMMING LANGUAGE**

This is a unification, rationalization, and extension of existing notations into a single language for the precise and concise description of algorithms. It is the central thesis of this book that the descriptive and analytic power of an adequate programming language amply repays the considerable effort required for its mastery. This thesis is developed by first presenting the entire language and then applying it in later chapters to several major topics. Although the language is new, it relies heavily on existent mathematical notations. Its development centers about the welding of notations from diverse fields (such as symbolic logic, number theory, matrix algebra, set theory, and graph theory) into a consistent whole. The entire language is summarized for reference in a ten-page appendix. 1962. 286 pages. $8.95.

KENT: **TEXTBOOK ON MECHANIZED INFORMATION RETRIEVAL**

The first systematic introduction to information retrieval by means of machines, this book defines the basic principles and techniques of applying computers to library science. An Interscience Book. 1962. 268 pages. $9.50.

TOMOVIC-KARPLUS: **HIGH SPEED ANALOG COMPUTERS**

An advanced treatment of high speed or repetitive computation dealing with theoretical questions. About 1/3 of the book is on theory. 1962. 255 pages. $9.95.

HUNT: **CONCEPT LEARNING: An Information Processing Problem**

Concept learning examined from a problem-oriented point of view. 1962. Approx. 296 pages. Prob. $6.95.

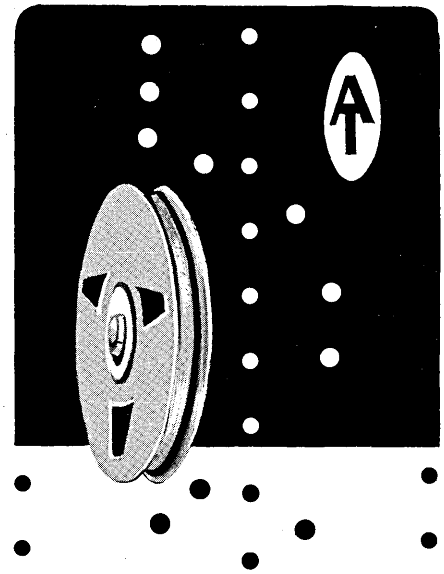
MONROE: **DIGITAL PROCESSES FOR SAMPLED DATA SYSTEMS**

Deals with the use of a digital computer as a data processing device in a system which may be partially analog in nature. 1962. 490 pages. $12.50.

*Send now for on-approval copies*

**JOHN WILEY & SONS, Inc.**
*440 PARK AVENUE SOUTH, NEW YORK 16, N.Y.*

CIRCLE 50 ON READER CARD



***SPECIALISTS IN THE COMPUTER FIELD . . .***

Our clients, with nationwide facilities have career positions for qualified professionals with experience in the following fields . . .

. . . INFORMATION RETRIEVAL
. . . OPERATIONS RESEARCH
. . . SCIENTIFIC PROGRAMMING
. . . DATA REDUCTION
. . . COMPUTER APPLICATIONS
. . . INFORMATION PROCESSING
. . . LOGIC DESIGN
. . . RELIABILITY ENGINEERING
. . . SYSTEMS DESIGN
. . . FIELD SERVICE
. . . SYSTEMS TEST TECHNICIANS
AND NUMEROUS OTHERS

Inquiries and resumes may de directed in confidence to:

Computer Division
ANDERSON TAYLOR & ASSOCIATES
1406 CHESTNUT STREET
Philadelphia 2, Pa. • LOcust 8-8190

CIRCLE 92 ON READER CARD

# BOOK capsules

**COMPUTER LANGUAGE—AN AUTOINSTRUCTIONAL INTRODUCTION TO FORTRAN,** by Harry L. Colman and Clarence Smallwood, 1962, McGraw-Hill Book Co., Inc., 330 W. 42nd St., New York 36, N. Y., 196 pp., $3.95 (text edition), $5.95 (trade edition).

The autoinstructional teaching technique is used in this text, based on the Skinner reinforcement theory of learning. The authors depart from traditional applications of the theory by including what they term a no-response mode, a situation when overt response and verification are omitted. The no-response sections are supplemented by exercises and program-writing tasks.

**A GUIDE TO ALGOL PROGRAMMING,** by Daniel D. McCracken, 1962, John Wiley & Sons, Inc., 440 Park Avenue South, New York 16, N. Y., 106 pp.

Examples and case studies are presented to aid the reader in transforming a problem statement into a problem-solving procedure, and includes illustrations of some of the basic ideas and techniques of numerical analysis. The book may be used for both self-study and formal courses. About 20 percent of the text is devoted to exercises.

**MACHINE-INDEPENDENT COMPUTER PROGRAMMING,** by Maurice H. Halstead, 1962, Spartan Books, 6411 Chillum Place N.W., Washington 12, D. C., 270 pp., $6.50.

A course taught at the University of California, "NELIAC, a dialect of ALGOL," is the basis of this book. Contents range from introduction to machine-independent computer programming, how to read NELIAC or ALGOL, how to write in the language, load programs, process noun lists, generators, and compiling compilers and compiler systems.

**INTRODUCTION TO ELECTRONIC DATA PROCESSING EQUIPMENT,** by Robert V. Oakford, 1962, McGraw-Hill Book Co., Inc., 330 W. 42nd St., New York 36, N. Y., 340 pp. $10.

Using the IBM 650 as an example, this book emphasizes man-machine communications, and discusses computer organization and machine language programming. Subjects covered include basic compter components; machine language; symbolic and algorithmic programming; and program testing.

**A GUIDE TO IBM 1401 PROGRAMMING,** by Daniel D. McCracken, 1962, John Wiley & Sons, Inc., 440 Park Avenue South, New York 16, N. Y., 200 pp., $5.75.

This text is aimed at those who desire a rapid grasp of the use of IBM 1400 Series equipment in business data processing. Opening chapters provide a basic understanding of data processing methods and equipment before proceeding to a detailed discussion of programming concepts and techniques. Included are a number of case studies which illustrate programmnig principles and indicate typical equipment applications.

**PLANNING A COMPUTER SYSTEM—PROJECT STRETCH,** edited by Werner Buchholz, 1962, McGraw-Hill Book Co., Inc., 330 W. 42nd St., New York 36, New York, 322 pp., $9.75.

This book describes the selection of an instruction set and related functional characteristics in the design of the IBM 7030. Some of the ideas and concepts described by chapter authors, who were directly concerned with the Stretch project, include: interrupt system, variable field and byte size operation, indexable addressing to the bit level, instruction look-ahead, and the generalized input-output approach.

## NEXT MONTH IN DATAMATION

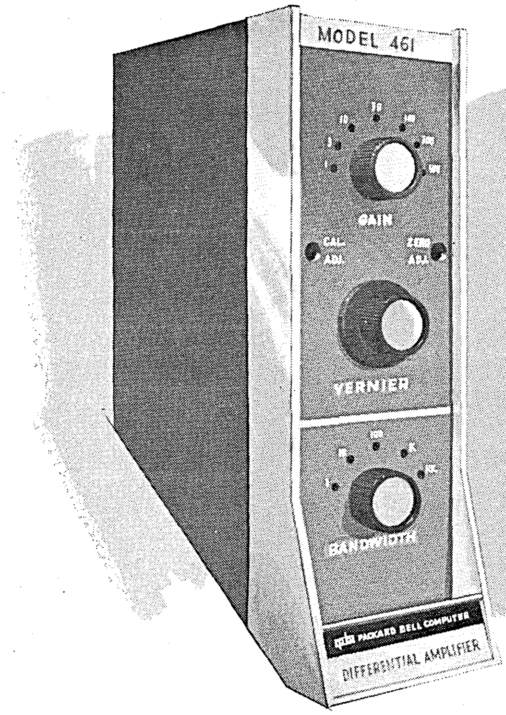
Billed as the industry's largest, single computer conference, this year's Fall Joint is scheduled for Philadelphia's Sheraton Hotel, Dec. 4-6, and is the principal subject of a special section in DATAMATION's November issue. A complete conference preview will include abstracts of the technical papers, a guide to an exhibit of more than 70 firms, a special events calender, field trip sites, ladies program and maps of the conference area.

Also on tap is part two of The RAND symposium on programming languages; an updated Adams' Guide to Computer Characteristics, now an annual event in DATAMATION, and an unusual cost survey of time buying throughout the country.

In addition, Kludge fans are reassured that they will not be forgotten in November and for the gals, a DATAMATION staff report on the Sociology of the Woman Programmer.

# ADVERTISERS' INDEX

# NEW VERIDICAL DIFFERENTIAL AMPLIFIER




**Temperature Range 0° to 75°C □ Bandwidth dc to 10kc □ Drift 0.3$\mu$v per °C □ Noise 4$\mu$v at 10kc Bandwidth □ Linearity 0.025% □ Recovery from 500% Overload to 0.05% in 200$\mu$sec □ Maximum Common Mode Voltage 250 volts**

Packard Bell Computer's all new silicon component *Model 461 differential amplifier* answers the need for high reliability amplification of low level transducer signals from ground support equipment, test stands, automatic checkout systems, range instrumentation, and industrial data acquisition systems. Combining high performance and low price with long term reliability, *Model 461* is also ideal for general purpose laboratory applications. Other features include adjustable 5-position selectable output filter; gain from 1 to 1000 with vernier; high input impedance 1000 megohms; and line driving capability unaffected by high line capacity. For complete specifications and demonstration, write:

pb Packard Bell Computer

*A subsidiary of Packard Bell Electronics*

1931 ARMACOST AVENUE • LOS ANGELES 25 • CALIFORNIA

CIRCLE 2 ON READER CARD

CIRCLE 3 ON READER CARD