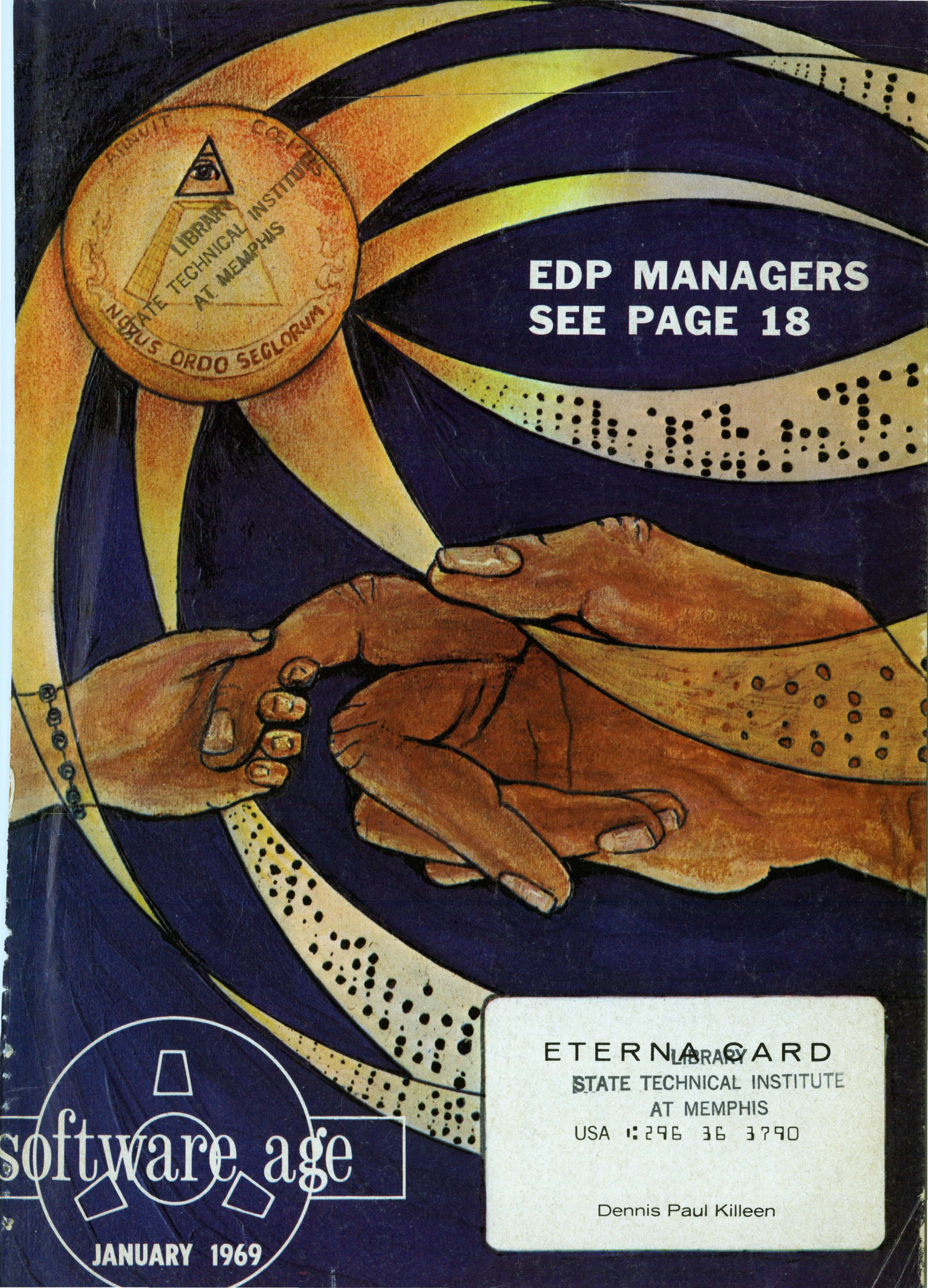
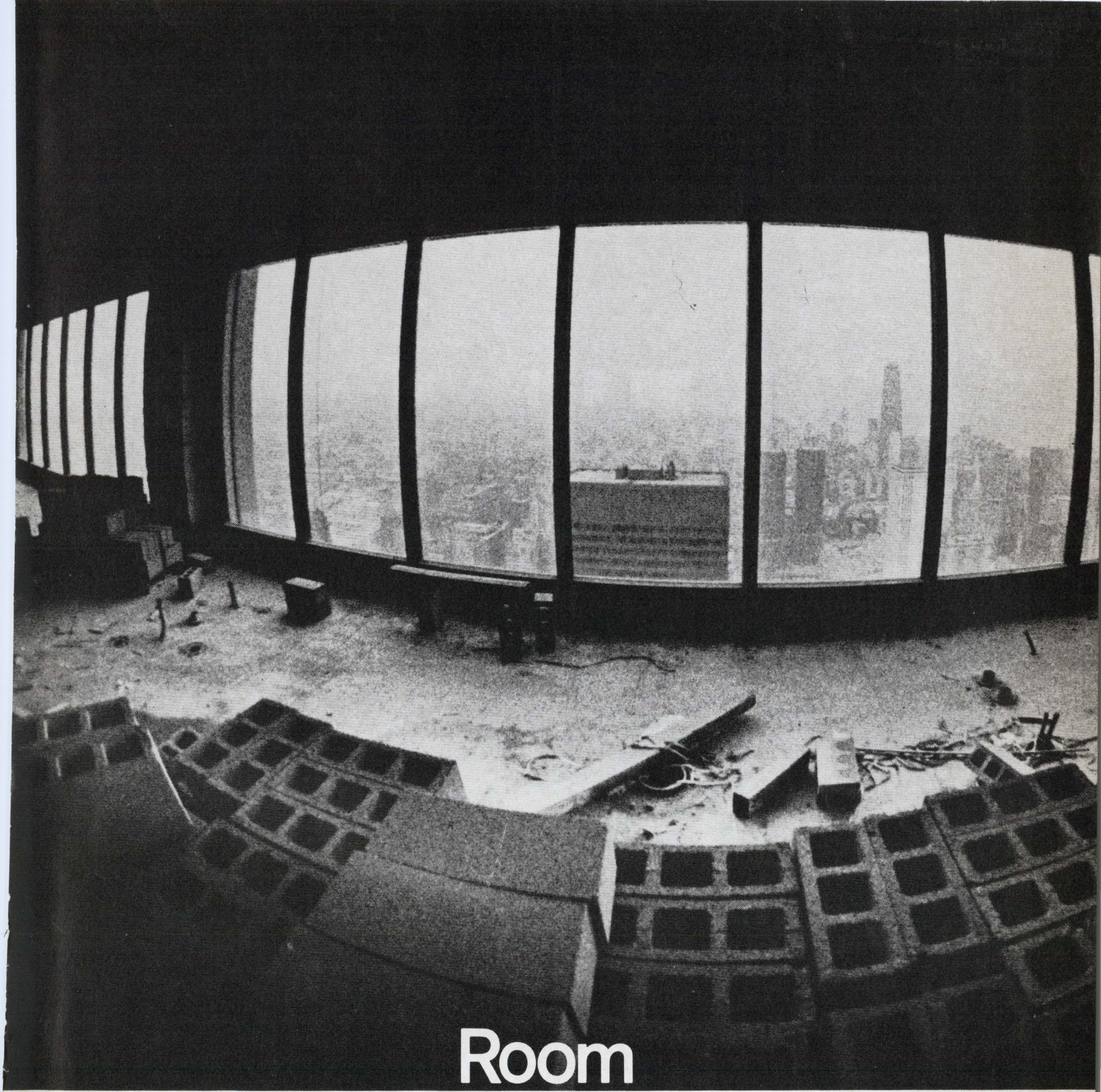
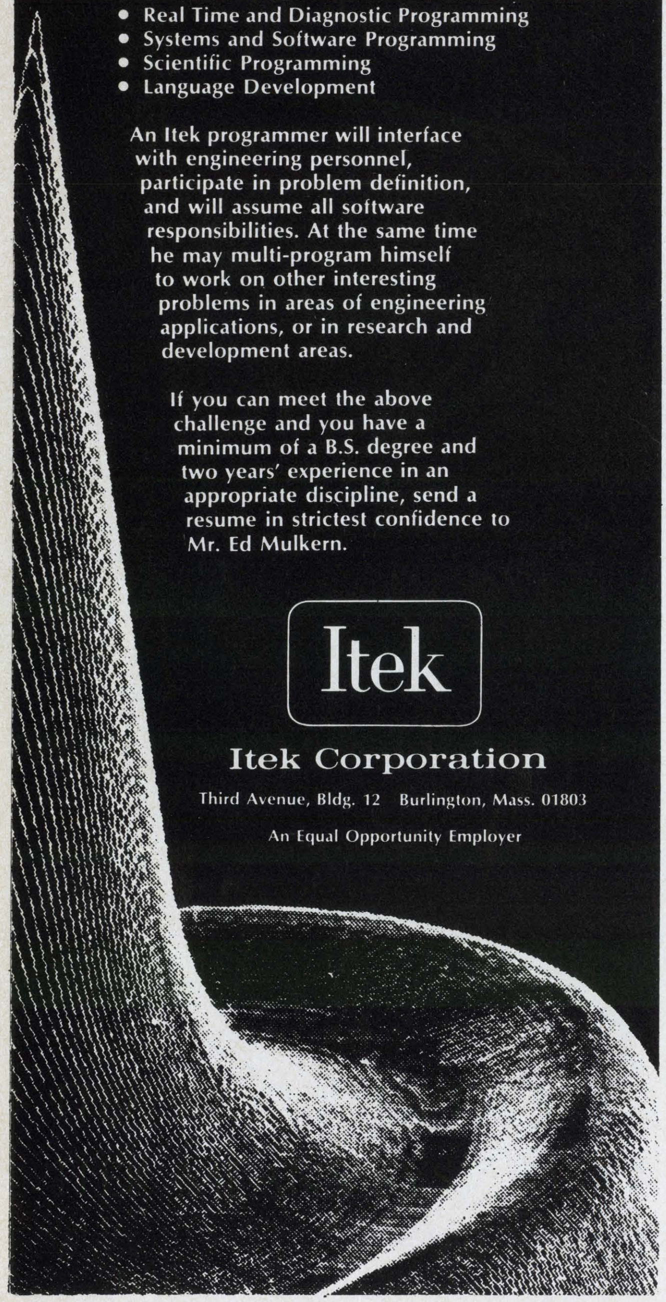


EDP MANAGERS
SEE PAGE 18

software age

JANUARY 1969





software age

JANUARY, 1969 Vol. 3–No. 1

CONTENTS **PAGE**

Publisher .... H. L. Rothra
Associate Publisher .... David W. French
Advertising Manager .... Norman Jacobs
Production Manager .... Betsy Pavkovich
Director of Circulation .... Howard Rogers
Circulation Manager .... Judith Arnopolin

Circulation of this issue more than **110,000**

SOFTWARE AGE is published monthly by PRESS–TECH, Inc.
1020 Church Street, Evanston, Illinois 60201
(312) 869–1244

Subscription free to qualified readers. Others, $10/yr. Individual copies, $1. Foreign subscriptions, $15/yr.

Main Sales Office: Norman Jacobs (Adv. Mgr.) or Norman Brodsky, 1020 Church Street, *Evanston, Illinois* 60201—Telephone (312) 869–1244.

Gerald Green, 60 E. 42nd, *New York, N.Y.*—Telephone (212) 697–5356.

Richard Faust, 9800 S. Sepulveda Blvd., *Los Angeles, California* 90045—Telephone (213) 776–0100.

Boston, Massachusetts 02021—Telephone (617) 542–1466.

Richard D. Clemmer, 27 Acoma Lane, *Collegeville, Pennsylvania* 19426—Telephone (215) 489–9141.

Richard Faust, *Palo Alto, California*—Telephone (415) 327–8340.

Which Way to the Cashless Society?

(Not *THAT* Way!)

> . . . he causeth all, both small and great, rich and poor, free and bond, to receive a mark . . . and that no man might buy or sell save that he had . . . the number of his name . . .
>
> —*Revelation 13:17,18*

■ In its issue of Dec. 2, 1968, *Advertising Age,* trade journal of the advertising business, reported:

> NEW YORK, Nov. 27—First National City Bank of New York will exchange the current Everything Card for a "new, bigger and better Everything Card" called Master Charge soon after the turn of the year.
>
> The new card can be used not only in New York but in 40 different states, Mexico, Japan and Europe for "almost every conceivable kind of retail purchase or service." In the lower right corner of each card will be the symbol "I" which signifies membership in the Interbank Card Assn. and means Master Charge holders can charge at more than 200,000 Interbank participants around the world.

Admittedly, the development described in that article hardly fulfills the prophesy of St. John—or the numerous prophesies of a "cashless society" made in more recent times by ardent proponents of the computer.

On the other hand, Apollo doesn't take us into Buck Rogers' world of the 25th Century either; but it takes us close enough to that new era to insure that we won't have to wait five centuries for it to materialize. When man blasts off the surface of the moon and returns safely to earth, he will have developed most of the technology to visit Mars or the other near planets. The outer planets then become only a few orders of magnitude (in propulsion and life-support technology) away.

The disturbing feature about the "new bigger and better Everything Card" is that—in proposed scope—it represents a long journey down the road toward the cashless society *without the first logical step having been taken!*

In the Apollo program, we find management deciding on alternative approaches by such logic as:

"For this particular mission, Approach 'A' would suffice; but we will use Approach 'B' because that's the way we're going to have to do it sooner or later as the missions become more complex."

Yet we see dozens of big credit operations growing bigger and bigger—closer and closer to that cashless society concept—without having taken the first logical step of assigning members and participating businesses a *truly universal identification.*

So long as only a Gulf card is good at my neighborhood service station, only American Express is honored at my favorite restaurant in Amsterdam and (for the future) only a Carte Blanche card will trip the cigarette vending machine in the Omaha airport and only a Bank of America card is honored at the crap table in Las Vegas, we still haven't taken that first step which will permit of the eventual cashless society.

We have seen some stumbling efforts toward unification of credit operations: airlines, with their own credit plan, now honor some of the leading credit cards as do oil companies having their own plans; major motel chains honor major oil company cards. But we shall begin to approach the cashless society when I hold a single card (with the credit organization of my own choice) and anything I charge, anywhere, can be automatically entered against my account with that one organization.

Obviously, First National City Bank or American Express cannot always approach its problems in exactly the same way NASA can. After all, NASA reports to its "stockholders" in rather general terms. Bank of America's stockholders (correctly) may feel that the bank didn't go into the credit field in order to make life easier for Shell's or American Airlines' credit operations. If the bank showed a major expenditure for the development of a universal credit identification system, you can bet some "little old lady from Pasadena" would be on her feet at the next stockholders' meeting wanting to know why!

But, curiously enough, the major task—that of assigning every American and every American business a unique identifying number—has already been done via Social Security and Employer Identification numbers. The industry would merely need to agree on identifying digits to indicate the credit organization through which I am to be billed and the one through which the businessman I bought from is to be paid.

When my bank account is identified by the same number, we can approach the cashless society, for my credit organization will be able to notify my bank of the amount due and that amount will be deducted from my account. A mailed notice to me of the amount of the debit may be the only "hard copy" required.

It would seem obvious that the credit organization which is first to move in this logical direction will make the move most economically; for the *first* firm to take the step can convert in the manner and the format which best interfaces with its old system. Those which follow will have to do so in the established format regardless of the difficulties the conversion from their particular system may impose.

Whoever makes that first conversion should consider some special identifying symbol to insure that mortician's bills go to the executor rather than the customer. ■

LIGHTWEIGHT HARDWARE FOR

Program Evaluation and Revie Technique (PERT) Analysis is b one of many applications devel oped for the IBM 1130 computin system at the Kingston plant Known schedule changes can b easily updated to quickly show th effect that these changes may hav on an overall project. Probabilit data, project slack and critica paths are some of the informatio that can be made availabl through the use of IBM 113 PERT programs.

HEAVY-DUTY

James W. Schlotzhauer

■ Freeing today's highly-skilled professional employees from routine and repetitive calculations is an area of concern in many industries—and is a problem being faced by many industries whose demands for quality are high and whose technology becomes obsolete in short periods of time.

In an effort to give their professionals greater flexibility to perform professional tasks, IBM's Systems Manufacturing Division plant in Kingston, New York, has provided a series of quick-response computer centers.

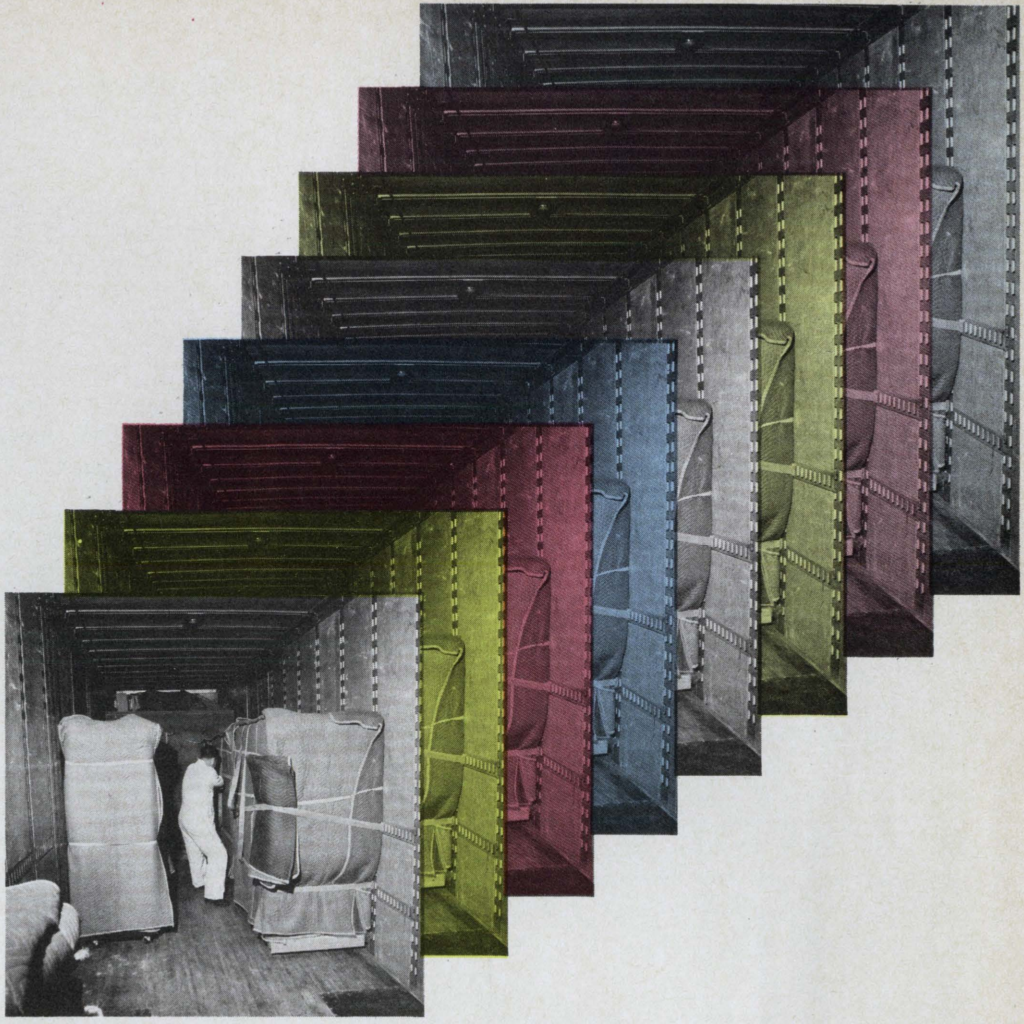
The Professional Computer Service (PCS) was developed by the Information Systems Department and has addressed this need. It not only performs routine calculations, placing more time at the disposal of the professional for analysis, but also allows for more efficiency in the job performance and more meaningful job results.

Recent surveys at the Kingston facility indicated that time, as well as money, could be saved if an easy access, computer-based approach to solving manufacturing problems was available to the professional.

It was also foreseen that the mid-1960's methods of planning and controlling would not be suitable for future work programs.

IBM Kingston already possessed the computing power to solve the large-scale, massive volume problems, but what of the countless "smaller" problems that needed answers? How could they continue to maintain a high standard of quality on little jobs while meeting voluminous schedules? How also could they continue to make short term improvements in product quality while producing the required volume more efficiently? It was determined that the answers to these problems lie in the installation of small-scale computing centers.

The Professional Computer Service presently available at IBM Kingston is centered around the relatively small and inexpensive IBM 1130 computing system. This system was selected on the basis of its ability to meet the criteria established as necessary to support the professional within his working environment at Kingston. Considerations which resulted in installation of this service ranged from determining the users requirements to the methods to be used for Post Installation Evaluation. (See Chart I) The criteria which ultimately resulted in the decision to install these centers included:



ırough the use of the IBM 1130 utomated Van Loading program, eveloped by industrial engineers IBM Kingston, trucks can be uickly loaded with elements of odern data processing systems. ıe planned loading of these units as brought about a decrease in nroute damage to these elec- onic units.

PROFESSIONALS

- The size of the system. The 1130 is a small, yet powerful, computer which is easy to use by personnel who are not computer oriented.
- The availability of the computer to the user area. This called for hardware which required only a small amount of floor space and did not need a controlled environment. (The 1130 requires a very small amount of non-controlled environment space (169 sq. ft.) and is run on a 115 volt, 1 phase, 3 wire, 60 cycle, nonlock plug.)
- The need to maintain budgets at a reasonable level. (The low cost of the 1130 computing system fulfills this need.)
- The necessity for larger (greater than 100) amounts of card input and the need for more printed output than a typewriter could provide. (More than adequately met within the 1130 system capability.)
- The ability for the professional user to easily convert his problems into machine readable language. (This is available through the 1130's FORTRAN capability.)

The 1130 configuration supporting the Professional Computer Service at IBM Kingston consists of the following:

- A very compact central processing unit, housing the main core storage, logic circuits, input/output attachment controls and a single disk storage drive.
- A console printer and keyboard, as well as operating switches and display lights for easy operator-machine communication.
- A 2315 disk cartridge, which, when mounted in its single disk storage drive, provides 512,000 words of back-up storage. The interchangeability of the 2315's permits an unlimited off-line disk storage capacity.
- Input/output equipment, consisting of an IBM 1442 card read punch capable of reading 400 cpm and punching 91 cpm, and an IBM 1132 printer, which operates at 80 lpm alphanumeric or 110 lpm numeric.

A supporting IBM 029 keypunch and storage cabinets complete the hardware of a typical IBM 1130 computer center at IBM Kingston.

Four of these IBM 1130 computing systems were installed at key locations throughout the facility.

The systems are utilized for general purpose computing and are particularly oriented toward

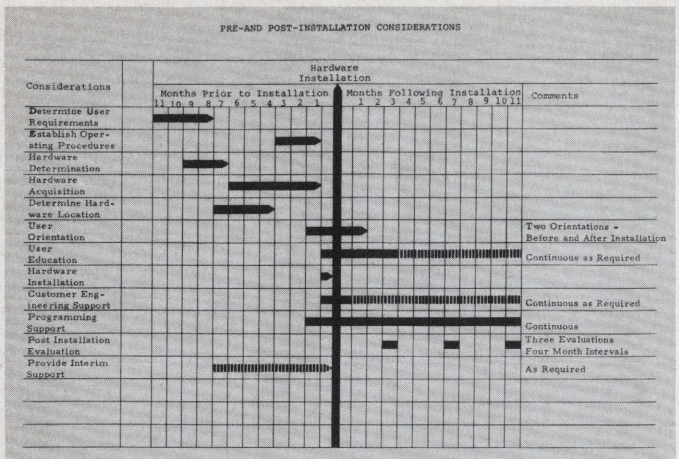


CHART I

operation by the individual requiring the problem solution on a compute-study-revise-compute cycle. The convenience of the 1130 system to his working environment also permits the professional to utilize it as a mathematical problem solver.

The centers were in addition to the already powerful computing strength available at IBM Kingston. Many Quality Assurance areas, for example, are utilizing the service to simulate conditions existing throughout the plant. Changes in any number of conditions could create detrimental effects on the quality of Kingston products. By recognizing a situation through simulation by the computer, Quality Assurance people can spot the trouble areas early and make immediate recommendations for improvement.

Industrial Engineering personnel have also found many uses for the computing centers. In their case they have proved to be a handy tool in devising programs for a more efficient use of the resources available at IBM Kingston. The 1130 system has been especially helpful when it comes to preparing a cost projection simulation. This program accepts requirement projections, inventory, commitments, and related costs and reflects projections of the actual cost to be applied.

The use of regression analysis for correlating work accountability activities with manning factors to evaluate indirect manpower is also another benefit of the Professional Computing Center used by Industrial Engineering.

One industrial engineer at IBM Kingston has, on the other hand, used the PCS to address the problem of loading the maximum amount of material aboard a van without creating hazards to either the material being shipped or the vehicle. An IBM 1130 program called "Automated Van Loading," was written and can automatically generate the following information:

- the number of manhours required to package each load;

Enjoy being treated as "special"?

Join Miles, in our Management Science department.

Our remarkable growth as a leading pharmaceutical and chemical company has long been based on utilizing the *individual* skills and talents of scientists, physicians, engineers, and specialists of many kinds.

It's second nature for us to treat each one as "special". And the same will apply to you in our Management Science department. You'll enjoy an atmosphere tailor-made to foster your best efforts, and personal growth. And the rewards add up, solidly.

That's why so many able people have moved to Miles . . . and then confined their moving to moving ahead.

SOFTWARE SPECIALIST

Take responsibility for coordination of software efforts of multi-divisional corporation.

Duties include generation, extension, and implementation of manufacturers' and other proprietary software including OS/360 multi-programming and tele-processing; and feasibility studies on proposed hardware and software. Experience with IBM system 360 (MFT or MVT) and a good knowledge of two or more programming languages is required.

PROGRAMMERS, PROJECT LEADER, SENIOR SYSTEMS AND PROGRAMMING ANALYSTS

Expansion creates positions at several levels, permitting full use of your developed skills in Management applications. A working knowledge of COBOL and/or BAL is needed, and familiarity with FORTRAN or PL 1 will be useful.

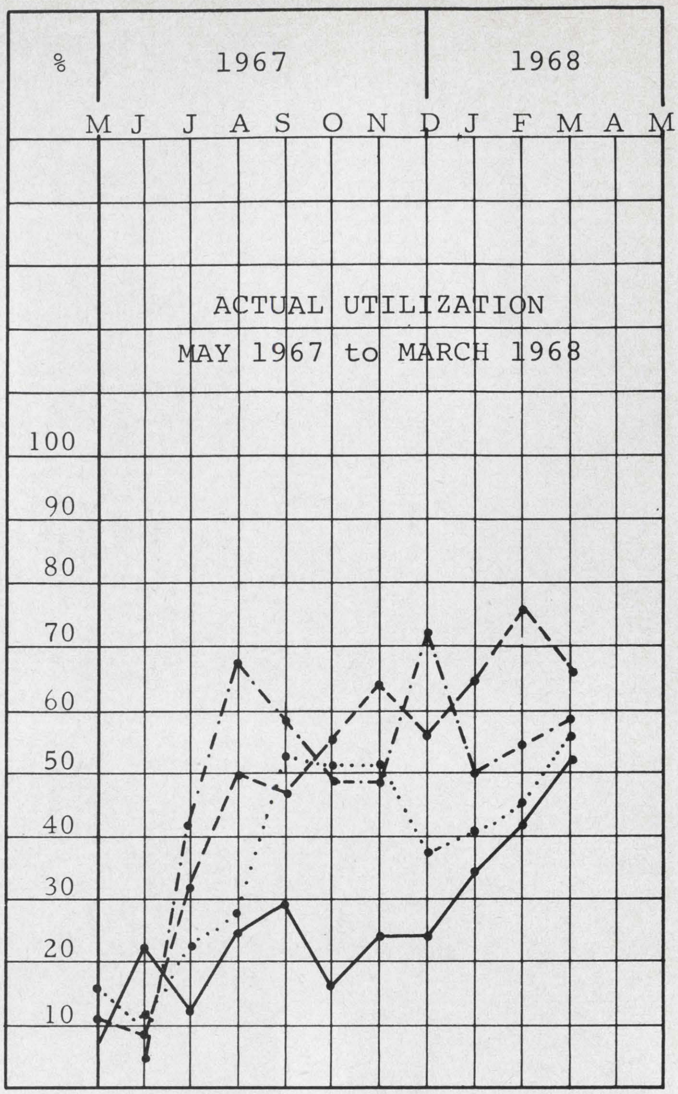
Attractive starting salaries, plus benefits that include profit sharing. Elkhart, an exceptionally problem-free, prosperous city, offers excellent living and a fortunate location.

Please send resume in full confidence, include salary history, to:
John W. Schulz, Placement Representative, or call collect (219) 264–8863

MILES LABORATORIES, INC.

ELKHART, INDIANA 46514

An Equal Opportunity Employer



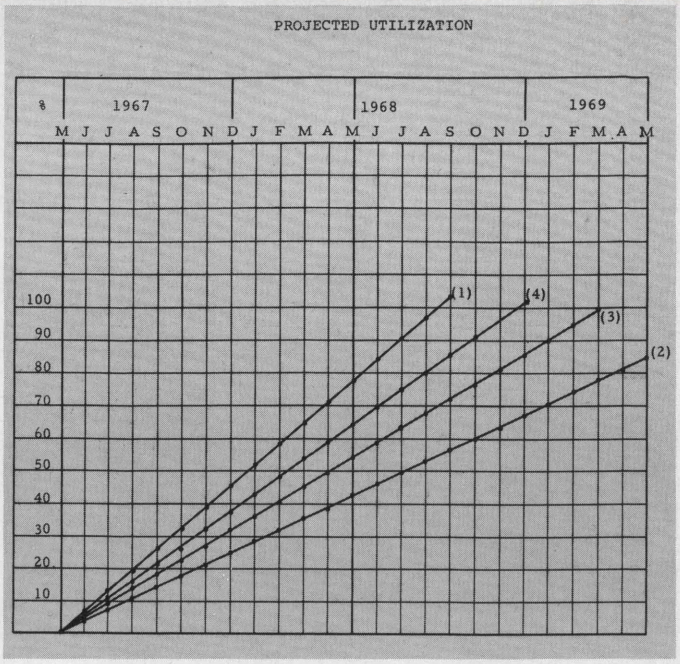
--- PCS Center 1, ——— PCS Center 2,
. . . PCS Center 3, —.— PCS Center 4

- the specific in-freight requirements;
- a van loading diagram; and
- incidental data, such as customer number, computer model, ship date, system number, and the date the van loading diagram was prepared.

This computer-oriented program, written in FORTRAN, first matches a shipping order to a file of the physical characteristics of each component. It then identifies the location within the van which will provide the maximum protection in terms of ride, padding, the elimination of box shifting, and accessible tie-downs. This location is determined by built-in parameters which express the characteristics of each type of van.

Implementation of this program has provided for more accurate ordering of air and motor equipment, faster loading, and has helped to reduce damage to customer shipments.

One of the most active centers is located in the Manufacturing Engineering area. This center uses many of the standard programs obtained through IBM's Data Processing Division.



1–PCS Center 1, 2–PCS Center 2, 3–PCS Center 3, 4–PCS Center 4.

These programs range from PERT Analysis and Project Control System packages to Linear Programming and General Purpose Simulator packages.

An area of concern to the Production Control organization was the creation and utilization of systems test schedule charts. These charts serve as Production Control roadmaps for scheduling the use of multiple test areas and have traditionally been prepared by hand. Often out of date before they were released to the Manufacturing managers, they were both costly and time consuming to prepare.

The PCS offered a possible solution to this problem. Following evaluation of the existing situation, programs were prepared in FORTRAN IV, which quickly and economically generate a systems test schedule chart. Other by-products of the programs include:

- a weekly or monthly summary for each system type scheduled for test. The summary lists the number of each system type scheduled to start system test, and the number of systems scheduled to be shipped.
- a list of schedule changes which supersedes specific information carried on the previous chart; and
- a legend to explain the symbols used on the test schedule chart; e.g., model type, advance planning, reserve areas, etc.

The system takes the bulky flow charts, easel pads, and multi-colored pens out of the hands of the Production Control scheduler and replaces them with a deck of IBM cards with which he is able to handle all the required data about the system and test areas.

Efforts expended prior to the actual installation of the 1130 computing systems facilitated the acceptance of the PCS by the professionals

who were previously not oriented toward the use of computers. The pre-installation orientation, education, and demonstrations created a feeling of anticipation to acquire the "hands-on" use of a computer. This, in turn, resulted in more rapid use of the centers, once the systems were installed.

The use of each of the installed centers and their projected growth trends can be seen in Charts II and III.

The individual using the 1130 is first of all an engineer or other professional, and secondly a programmer. He, therefore, must rely on data processing professionals when he encounters problem areas which are unique and not readily recognized by individuals who are not computer-oriented.

There are, however, computer education programs available at IBM Kingston—some are voluntary, while others are of the on-the-job training type. These programs make it possible for anyone desiring to utilize the computing center to acquire the skills necessary for operation of the system. Programming courses in the FORTRAN and assembler languages, operating techniques for the IBM 1130 system, and advanced system courses in computer systems programming, are available to the employees within the Kingston facility. In addition, the engineering areas have included Professional Computing Service training in their orientation of new employees.

To date, there have been more than 250 professional employees who have taken advantage of the Computer Education Programs, available to them at IBM Kingston on a voluntary basis. It has been the consensus of individuals involved and their management that a majority of these individuals, and those receiving equivalent training in the new employee orientation programs, have displayed a new and progressive outlook toward their jobs.

The benefits derived from this system are many—and varied. They include:

- reduction in the overtime previously required to produce quick response answers to complex questions.
- reduction of the time previously taken to go from the problem to a final result. This reduction has been a direct result of the amount of data capable of being used in arriving at the initial result of a problem. The increased number of possible solutions tested has also reduced the problem-to-solution cycle;
- the need for increased manpower to perform a particular function has been greatly reduced due to the increased productivity provided by the computing service to those already performing that function;
- providing the professional employee with more time to perform his job by eliminating excessive clerical activities;
- helping to produce more meaningful and accurate solutions to a problem;
- allowing for sounder decisions to be made due to the decisions being based on more extensive and accurate data; and
- acquiring top professional talent from colleges and universities by displaying the availability of such a service to assist them in performing their jobs.

The role of the professional in the propagation of industrial progress is being realized and recognized by the Information Systems community. The transition from a multi-program batch mode manner of providing computer services to that of generalized on-line remote processing is in process. The concepts that a Professional Computer Service presents are in keeping with the evolutionary process and will make the conversion more meaningful and productive. ■

Mr. Schlotzhauer is a manager in Information Systems at IBM's Systems Manufacturing Division plant in Kingston, New York.

He joined IBM Kingston in June of 1962 as a programmer trainee in Information Systems Development. He was promoted to junior programmer in that area in December of that year. In June of 1963, he was promoted to the post of pre-professional programmer, and in June of 1964 was promoted to associate programmer.

From January of 1966 to January of last year, Mr. Schlotzhauer was a group coordinator in Information Systems Development. Following that he was promoted to manager of the Information Systems Production Planning and Industrial Engineering Services Department, a post he held until December of 1967, when he was appointed manager of the Manufacturing Information System Installation Department at Kingston.

executive routine

Jerome T. Murray

■ Scheduling is one of the problems that proves troublesome to a good many DP managers—particularly to those relatively new to the responsibility.

With this problem, as with many others, "it ain't what you do, it's the way that you do it" which counts.

In a small installation capable of running only one job at a time, it's fairly easy: "The job we're running now can't be interrupted and it'll take four more hours to finish it. Therefore we can (or can't) handle this other job today with (or without) 30 min. overtime." It can all be taken care of with flat statements.

But consider: your installation is large and unusual. Each of six departments of the company has a computer dedicated to its needs. The six computers share a common I/O device pool facilitated by switching units. The Operations Research Dept. has just sent you a deck of punched cards which they want put on mag tape using a special program which they have sent with the input cards. The O-R Dept. says the run will take 2 hrs. and 30 min. It's now 1:30 PM and your department closes for the day at 5:15, and the "no overtime" order is in effect. O-R wants the job out today. Can you do it?

Well, you have a lot of information available and it *can* be put into a series of flat statements:

1. O-R's computer is idle.
2. Manufacturing's computer is using the CRT output and will be tied up for 4 hrs., 30 min.
3. Sales' computer is running a file update with input from a remote terminal.
4. Credit's computer is running a card-to-disk sort.
5. Personnel's computer is running a classification program with paper-tape input and mag-tape output.
6. Credit's computer will need 30 min. to complete the sort.
7. Personnel's classification run will be finished in an hour.

Etc., Etc., Etc.

When you have put down these, and all the other statements you can be sure of, what do you have? It looks like one of these brain teasers: "Smith was born on the same day of the week as Jones. Brown has a seven-stroke handicap. The one wearing the plaid shirt has two children. Which one was late for work this morning?"

You can make your scheduling problems a whole lot easier (and simplify solution of those brain teasers, incidentally) if you'll put down all you know in tabular fashion rather than as a series of flat statements.

When you see the information in that form, it becomes a whole lot easier to cut through the fog of un-

Comp.	1	2	3	4	5	6
Dept.	Pers.	Budget	Mfg.	Sales	Credit	O-R
Prog.	Classif.	Comput.	Payroll	Update	Sort	-----
Output	Mag Tape	Prtr.	CRT	Cd. Pnch.	Disk	-----
Timing	1 hr.	3 hrs.	4½ hrs.	4 hrs.	30 min.	-----
Input	Paper Tp.	MICR	OCR	Remote	Card	-----

Perhaps you could suggest a name for this column. WE have been looking for some pat, broadly-accepted term to connote the top man in the EDP department . . . something akin to the infantry company's "top kick" or the newsroom's "slot man."

Unfortunately, any specialty as new as EDP or software tends to be highly specific in its terminology. We questioned a long-time EDP manager of our acquaintance as to a broad term for his post and he said the only two appellations with which he was familiar were "Hey, Chief," (from below) and "Hey, You!" (from above).

If there ISN'T any good general term for the top man in the EDP department, there surely SHOULD BE. How about suggesting one?

Systems Programmers/Analysts:

Give IBM a hand and we'll try to make it a free one.

If predictable thinking were what we wanted, we'd just program it. But what we're looking for are imagination and ingenuity. If you have these qualities, and you qualify for a job with IBM, you can be sure you'll get to use them.

Immediate openings.

There are openings for Programmers/Analysts at IBM Endicott, N.Y.

You'd work on the designing, programming, testing, and installation of application programs. These would aid manufacturing operations, as well as Special Production Planning and Control.

Your work will relate to real-time teleprocessing applications, plus direct access and sequential-storage devices.

A Bachelor's degree and at least a year's experience are required. You should have a knowledge of basic assembler language and the detailed design aspects of manufacturing.

Grow with IBM.

Today's major growth industry is information handling and control. And IBM is a leader in that field. Our job is to help our customers solve their information-handling problems, which range from diagnosing diseases to urban redevelopment.

If you're a problem-solver who wants a personal sense of achievement for your hard work in a growth company, consider IBM.

Call or write.

Jerry Hinkley is the man to talk to. Call him collect at (607) 755-2855. Or write to him at IBM Corporation, Dept. CA1012, 1701 North Street, Endicott, N.Y. 13760.

An Equal Opportunity Employer

IBM®

IBM programmers and systems analysts work in small teams on an IBM System/360 computer.

important facts and zero in on those that are pertinent. To fill in the blanks under Computer 6, we need Mag-Tape output, Card input and 2½ hrs. timing (before 5:15 PM.) The table makes it obvious that the card input will be available in 30 min. (at 2 PM according to our problem.) The mag-tape needed for output will be available in 1 hr. (at 2:30.)

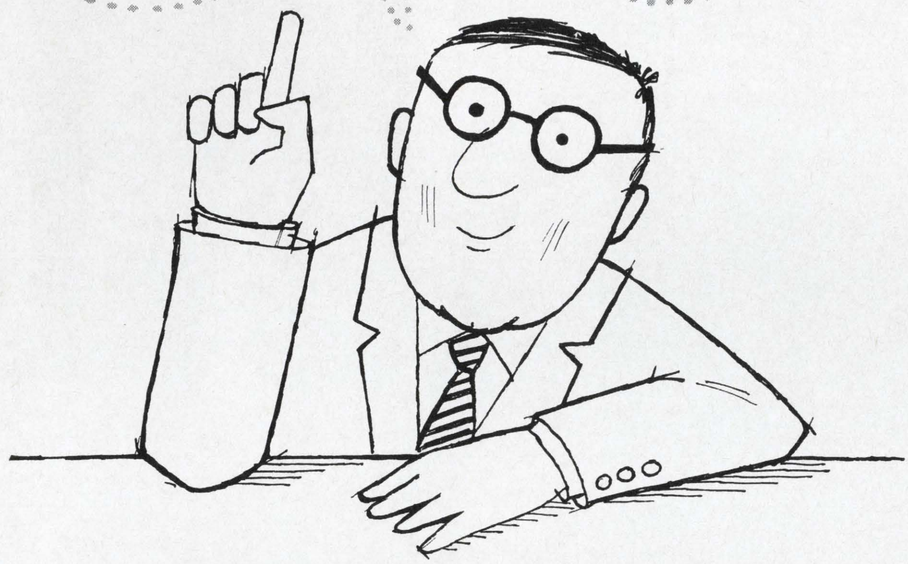
Obviously, if O-R's time estimate is correct—or even within 15 min. of being correct—you can finish their job today without running into the forbidden overtime.

Now, granted, you may never be called upon to schedule for six computers in one department; but even if your present computer doesn't provide for it, the next one your department gets will probably be capable of running three or more programs simultaneously and it will be your job to be able to say (and know what you're talking about!):

"We'll have the cpu capacity for this job available in 10 min.; but *this* job can't go on then. It requires punched-card output and that job on the front section has the punch tied up until 3 o'clock."

Shared I/O devices are fairly common already and they are going to become a lot more common. And, how expertly their time is scheduled will pretty much determine whether your company gets its money's worth out of its equipment. ■

ASSIGNMENT: DESIGN OR PROGRAM A COMPLETE ORDER ENTRY SYSTEM ON LINE TO A 360/65



How does that grab you!?

If that assignment brought a glint to your eye, read on. There are more. *Like designing or programming a data base for over 300,000 parts . . . or mathematical and simulation models of VTOL aircraft . . . or software in support of a real-time data acquisition and reduction system, airborne computers or a major retrieval system for technical data.* We could go on, but that will give you the idea.

If this is the kind of career excitement you relish, maybe you should get involved with us. Get involved with far-out programs that will hone your abilities to a fine edge. You'd be joining an innovative group dedicated to producing the most advanced VTOL aircraft—high-speed TurboTrains—and high-performance marine vehicles. *And the computer systems to support them.*

Your tools will be bright and shiny. Our inventory includes UNIVAC 1108's and IBM 360's with graphics and teleprocessing.

If you're ready for new maturity, we want to talk careers with you. We have exceptional assignments at all levels of experience for: **Programmers and Analysts in both Commercial and Scientific fields.**

Send your resume in confidence, stating salary requirements, to:

MR. LEO J. SHALVOY, PROFESSIONAL EMPLOYMENT.

Sikorsky Aircraft

DIVISION OF UNITED AIRCRAFT CORPORATION

STRATFORD, CONNECTICUT

An Equal Opportunity Employer

TROUBLE-TRAN PRESENTS XTRAN'S ADVENTURES IN FORTRAN

By GEORGE N. VASSILAKIS

Send your ANSWER to the problems posed here in each issue to:

TROUBLE-TRAN EDITOR

software age

1020 Church St., Evanston, Illinois 60201

You can also profit by submitting PROBLEMS for this feature. If your problem is FORTRAN programming is selected for use in this feature, you will receive $25.00

TROUBLE-TRAN's Objectives:

1. To have fun.
2. To promote USA Standard FORTRAN by pointing out differences and inconsistencies of existing FORTRAN Compilers.
3. To alert programmers to the physical limitations of hardware.

Contest Rules:

1. The best answer (best explanation) bearing the earliest postmark wins .. $25.00
2. The second best answer with the earliest postmark will net the reader submitting it $15.00

PROBLEM OF THE MONTH

John and Tom are two of our young engineers who are doing more and more of their own programming now that they have access to remote terminals. Yesterday, John was looking in his Math Library for a table of factorials when Tom walked in and suggested that it would be quicker to write a program and run it on the remote terminal down the hall.

The following program was written and executed:

```
      F = 1.
      DO 10 J = 1,100
      X = J
      F = F*X
      WRITE (1,100) J,F
  100 FORMAT (I10, E15.8)
   10 CONTINUE
      STOP
      END
```

As they were logging out, XTRAN came by and with a quick glance at their answers realized something was wrong and offered to help them.

What was wrong? How would you have programmed it?

ANSWER TO LAST MONTH'S PROBLEM

The purpose of this problem was to illustrate some of the uses and misuses of the non-USA Standard FORTRAN statement "ENTRY"; and to promote the use of USA Standard FORTRAN by emphasizing what NOT to do.

The answer to our problem was 18 or 15 depending on the compiler used. For the purpose of awarding the prizes, it is better to talk about the best answer and not the correct answer. The best answer of course is based on the best explanation of why the answer is 18 or 15.

1. CDC-6000 FORTRAN

The statement ENTRY FPRIME(A) would not compile, because of the dummy argument A. Would someone please explain why the compiler was designed to require the calling program to use arguments but not allowing the appearance of corresponding dummy arguments in the ENTRY statement?

> "In the calling program, the reference to the entry name is made just as if reference were being made to the function or subroutine in which the ENTRY is imbedded."

2. IBM System/360 FORTRAN

The G-level FORTRAN accepted the statement and the value of Z in the calling program was 15.

3. IBM IBSYS

IBM 7094 FORTRAN Version what? Oh, yes, don't waste your time with Version 12; it won't work. Try Version 13, but keep in mind there is more than one Version 13 depending on which update of Version 13 your installation is using.

The Version 13 I used has not been updated for two years and gave me 15 as the value of Z. It seems to me I should have a diagnostic instead, since I violated the following rule:

> "3. In a FUNCTION subprogram, only the FUNCTION name may be used as the variable to carry a result back to the calling program. The ENTRY name may not be used for this purpose."

Of course, it is rather difficult (if not impossible) for any compiler to diagnose such a violation.

P.S. Wanted:

Volunteers who would be willing to forfeit their rights of winning, by accepting TROUBLE-TRAN problems and testing them before publication on computers which are not readily available to me. I would also appreciate receiving a copy of a FORTRAN manual on each of the following systems:

1. B5500
2. GE-635
3. H-8200
4. Philco 2000
5. Spectra 70/55
6. Sigma 7
7. SCC-6700
8. IC-6000
9. UNIVAC 1108
10. You name it

XTRAN

TROUBLE-TRAN WINNERS—AUGUST

1st—Edward K. Strom, West Allis, Wis. 2nd—Daniel J. Kaminski, State College, Pa.

EDP SALESMEN & APPLICATIONS SPECIALISTS

Not everyone will like the Washington, D.C. area . . .

That's why we have 49 other locations.

Information Systems

Time-Sharing Service

But they're all the same from one viewpoint. Opportunity. It's there. As much as you can handle.

Of course, that's what everybody says. But not everybody is the leader in time sharing. Only GE. And we're extending that lead. With a broadened library of user programs that's still growing. More powerful systems software. Third-generation hardware. And pioneering—in a big way—of new applications like time-shared N/C programming.

Right now, you're probably wondering what this means to you. Besides a more marketable product, that is. For one thing, it means diversity: sales careers in all shapes and sizes. Including customer liaison, influencing market planning . . . even suggesting, educating and exploiting whole new markets. Plus whatever else your ingenuity can add to the spectrum.

There's another facet to a GE job you won't want to neglect, either. It's the kind of company we are. In terms of recognition. Promotion. Stability. And success.

But we'd rather tell you about all the other pluses in person. If you've EDP experience (preferably in the customer applications area) and know your FORTRAN as well as your goals, you'll want to look into the opportunities at any of our locations throughout the country. For specific information, write to the Regional Manager of whatever location interests you.

But if you like the Washington, D.C. area, then write to F. W. Gibbins, Manager, Personnel Practices, GE Co., Information Service Department, Sect. 164-A, 7735 Old Georgetown Road, Bethesda, Md. 20014.

Information Service Department

GENERAL ELECTRIC

An Equal Opportunity Employer m/f

Where do you want to work?
There are GE installations in all these cities. You are invited to forward your resume to the Regional Manager at the address shown *above* the city of your choice.

Write to Mr. A. F. Jones, Manager, Central Region, Sect. 164-A, 20800 Center Ridge Road, Suite 400, Cleveland, Ohio 44116 for openings in:

Illinois:
Chicago

Indiana:
Indianapolis

Michigan:
Detroit

Missouri:
St. Louis

Ohio:
Cincinnati
Cleveland
Columbus
Dayton
Toledo

Pennsylvania:
Pittsburgh

Wisconsin:
Milwaukee

Write to Mr. K. G. MacDonald, Manager, Western Region, Sect. 164-A, 2721 N. Central Avenue, Suite 915, Phoenix, Arizona 85004 for openings in:

Arizona:
Phoenix

California:
Long Beach
Los Angeles
Palo Alto
Reseda
San Diego
San Francisco
Santa Barbara
West Covina

Colorado:
Denver

New Mexico:
Albuquerque

Oregon:
Portland

Utah:
Salt Lake City

Washington:
Seattle

Write to Mr. W. D. Robusto, Manager, Northeastern Region, Sect. 164-A, 6563 Ridings Road, Syracuse, New York 13206 for openings in:

Connecticut:
Hamden

Massachusetts:
Boston

New York:
Buffalo
Rochester
Schenectady
Syracuse

Write to Mr. R. A. Else, Manager, Eastern Region, Sect. 164-A, 1490 Queen Ann Road, Teaneck, New Jersey 07666 for openings in:

New Jersey:
Teaneck

New York:
Long Island
New York City

Pennsylvania:
Philadelphia

Write to Mr. D. Stone, Market Development, Southeastern Region, Sect. 164-A, 1485 Northeast Expressway, Atlanta, Georgia 30329 for openings in:

Florida:
Orlando

Georgia:
Atlanta

Louisiana:
New Orleans

Maryland:
Baltimore
Bethesda

Oklahoma:
Tulsa

Texas:
Dallas
Houston

South Carolina:
Greenville

TERP

A METHOD OF SOFTWARE-CONTROLLED ERROR RECOVERY PROCEDURE TESTING FOR THE DOS/TOS SYSTEM/360

Part II Continued from December

R. A. Greene
Systems Development Division
IBM Corp.
Endicott, New York

ABSTRACT: The development of the Testing Error Recovery Procedures (TERP) program was a successful pioneer effort in the area of software-controlled Error Recovery Procedure (ERP) testing. TERP met the need for a standard method of ERP testing and satisfied the requirements of both the error recovery procedure programmer and the systems test programmer.

TERP is an ERP test case generator. Test cases generated by TERP can simulate I/O errors and machine check errors. TERP provides self-checking facilities which aid in the verification of test case results. The writing of ERP test cases is minimized to coding in a macro language because TERP is implemented through the use of assembler macros. The time required for testing ERP is also considerably reduced because there are no prerequisites or special operating procedures for executing TERP test cases.

A standard method of DOS/TOS ERP testing has been established through the use of TERP. Although TERP was developed around a specific operating system, its underlying principles can be applied to testing error recovery procedures of operating systems in general.

3. *TERP Logic Diagrams*
The flowcharts in Figures 1 and 2 describe the TERP ERP test case logic which permits the simulation of I/O errors and machine check errors. Figure 1 describes general TERP test case logic. Figure 2 describes, in detail, the function of the TERP I/O interrupt monitor. A schematic of the TERP I/O error simulation cycle is provided in Figure 3.

III. TERP APPLICATIONS

The following outline defines existing applications of TERP.

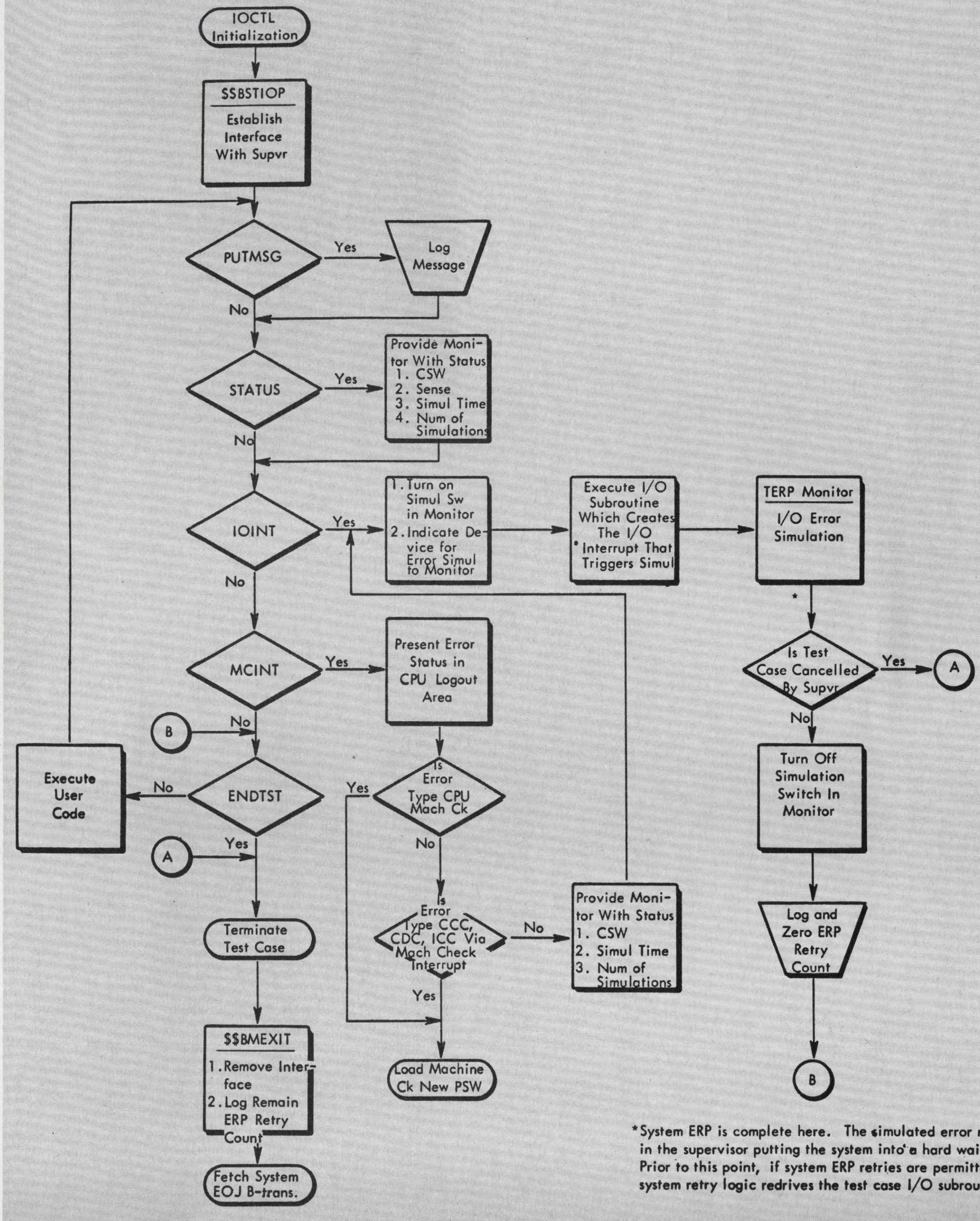
A. DEVICE ERP TESTING
   1. Unit Record
   2. Magnetic Tape
   3. DASD
B. CHANNEL ERP TESTING
   1. Multiplexor
   2. Selector
C. MACHINE CHECK ERP TESTING
   1. CPU System/360 Model 30
   2. CPU System/360 Model 40
   3. CPU System/360 Model 50
D. LIOCS ERROR OPTIONS TESTING

IV. TERP USER CONSIDERATIONS

The following list of information is intended as an aid in generating ERP test cases with TERP. Appendix A defines the format and function of the TERP macros. Appendix B contains a sample test case generated with TERP.

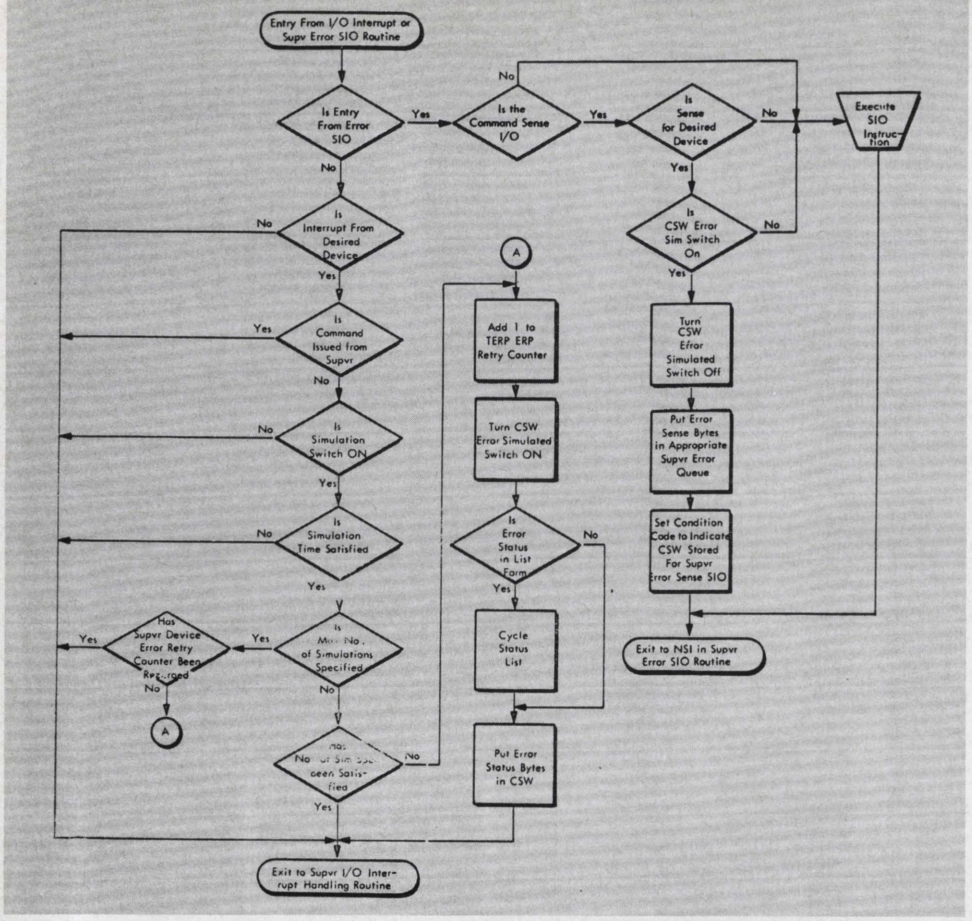
1. TERP generated code is self-relocating and may be executed in any partition in either the batched-only or multiprogramming environment.
2. The method of dynamically establishing an interface between the supervisor being tested and the TERP test case allows the test case to be executed using any supervisor and eliminates the need for special system generations, etc.
3. Test cases generated with TERP must begin with the IOCTL macro and terminate with an assembler END statement. The ERP test case must not contain an assembler START statement or code for establishing or loading program base registers since these functions are performed by the IOCTL generated code. The TERP monitor uses base register 10. Base registers 8, 9, and 10 are reserved for the remainder of the test case.
4. User code may exist anywhere in the ERP test case, provided the first source instruction is the IOCTL macro. This code may contain privileged operations.
5. User I/O subroutines specified in the IOINT macro statement may be written employing PIOCS or LIOCS. If PIOCS is used, the subroutine will normally contain an EXCP and WAIT. If LIOCS is used, the subroutine will normally contain an imperative macro statement, such as GET or PUT. LIOCS error options may be tested in this manner.
6. The IOCTL macro generates four I/O subroutines which may be specified in the IOINT macro statement as routines during which error simulation is requested. Following is a description of these subroutines.

Fig. 1



*System ERP is complete here. The simulated error may result in the supervisor putting the system into a hard wait state. Prior to this point, if system ERP retries are permitted, the system retry logic redrives the test case I/O subroutine.

Fig. 2—TERP I/O Monitor Logic (DOS)



a. *For Existing Devices.* Three subroutines are generated for issuing the DASD commands: SEEK, READ COUNT KEY DATA, and WRITE COUNT KEY DATA. These subroutines are available to the user by specifying the routine name of USEEK, UREAD, or UWRITE in the IOINT macro statement.

A fourth subroutine, UNOP, may also be specified in the IOINT macro statement. Execution of this subroutine causes a SENSE I/O command to be issued to the monitored device. This subroutine should be specified only when the system ERP does not inspect the command code during its processing of the simulated error.

b. *For Simulated Devices.* Execution of the UNOP subroutine enables the user to create an I/O interrupt on a simulated device. Simulation of an I/O device is accomplished by ADDing an existing channel and unit address as the desired device type. For example, to simulate a 2321 data cell using a tape drive with the address 180, the following ADD statement would be required at IPL time: ADD X'180', 2321.
The majority of ERP testing can be performed on the simulated device. Device simulation should not be employed in the following two situations. The first occurs when the system ERP inspects the command code issued by the I/O subroutine. The second occurs when the system ERP issues commands other than SENSE I/O to the simulated device.

c. *Use of TERP I/O Subroutines.* The desired I/O sub-

set your sails...

Systems Engineering Laboratories in Fort Lauderdale has the following career openings:

■ **SYSTEMS PROGRAMMERS:** Development of operating systems, assemblers, compilers, monitors, math libraries, customer liaison, etc. Applicants should have at least two to five years related systems programming experience plus a technical degree or equivalent.

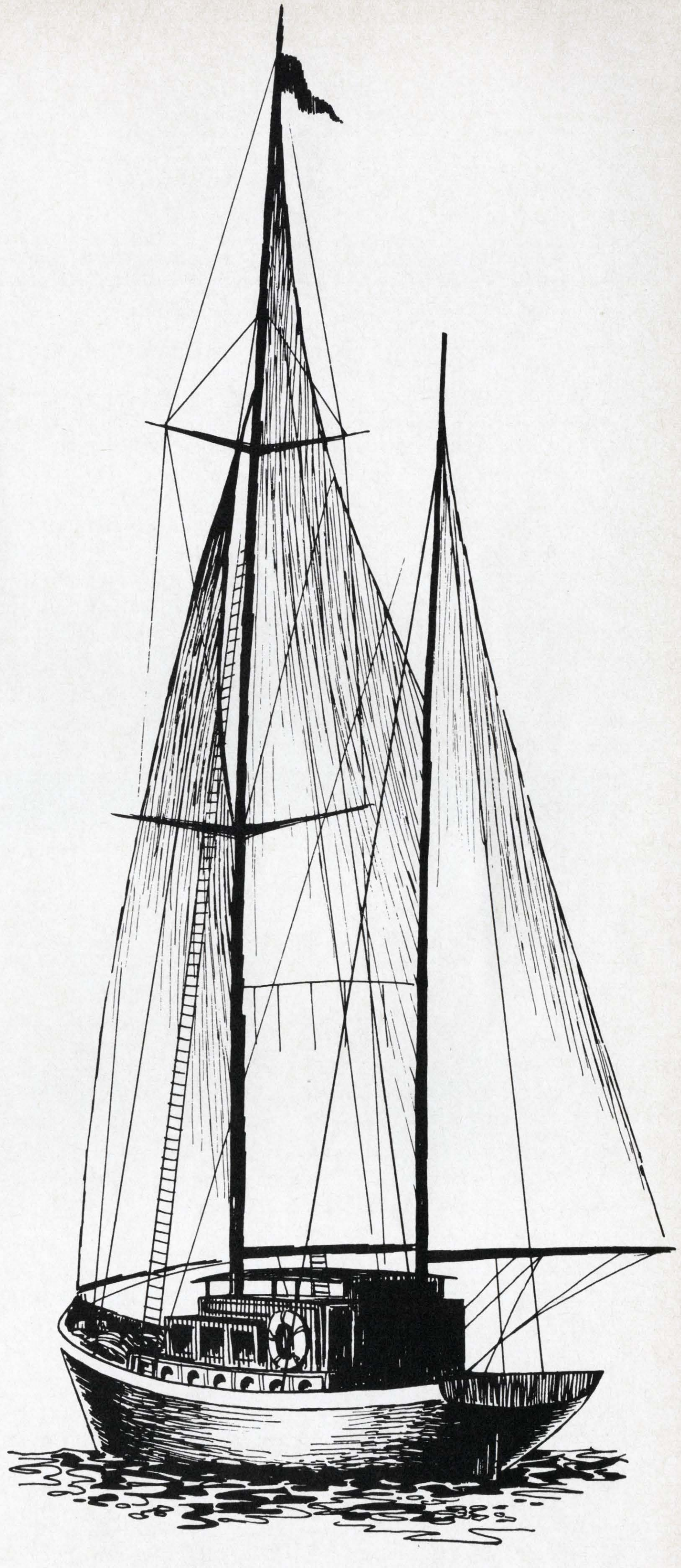
■ **APPLICATION PROGRAMMERS:** Development of assembly language software for data acquisition, process control, automated check-out, communications, etc. Applicants should have at least two years related programming experience plus a technical degree or equivalent.

■ **ELECTRICAL ENGINEERS:** Computer mainframe and peripheral logic design, digital systems design, analog and digital circuit design, and manufacturing project engineering. Applicant must have BSEE Degree plus at least three to five years of related experience.

Systems Engineering Laboratories is a rapidly growing computer manufacturer and offers challenging assignments and real growth opportunties. Interested persons are invited to submit resumes to Mr. L. K. Buck, Dept. SA-I, P. O. Box 9148, Fort Lauderdale, Florida 33310.

An Equal Opportunity Employer

on beautiful ft. lauderdale.



Systems Engineering Laboratories
6901 West Sunrise Blvd., Fort Lauderdale, Florida 33313
Area Code 305 587-2900

routine should be specified by name in the IOINT macro statement. Code generated by IOINT dynamically modifies the logical unit in the CCB associated with the specified TERP I/O subroutine to agree with the DEVICE parameter in the IOINT macro-statement.

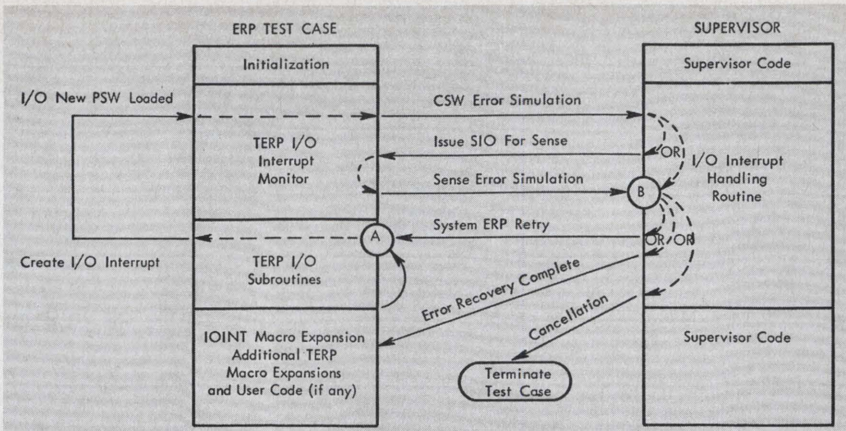
7. TERP provides three self-checking features which aid in the verification of test case results. First, the PUTMSG macro may be employed for logging expected results. Second, I/O trace may be employed for verifying the commands issued by system ERP. Third, the actual number of system ERP retries performed is logged following the error simulation.

V. CONCLUSION

Perhaps the most significant result of the TERP project is the establishment of a procedure that provides considerable savings in man hours and machine hours required to perform systems ERP testing. Man hour savings are derived in test case generation, execution, and evaluation. Test case generation effort is reduced to a minimum due to the macro method of TERP implementation. Test case execution can be performed without programmer "hand on" time because test cases are software-controlled. Test case evaluation is simplified through the use of self-checking features. Machine hour savings are derived through the execution of software-controlled test cases. The awkward characteristics of previous methods for ERP test case execution, such as hardware fixes, core patches, special system generations, external introduction of data, manual intervention, and special operating procedures, have been eliminated through the use of TERP.

Another significant effect of the TERP project has been the adoption of TERP as the standard method of DOS/TOS ERP testing. As an accepted standard, TERP has provided a common ground for communication between various ERP testing groups and a tool for projecting man and machine hours required to test future ERP support. While TERP will change to meet new applications, its basic philosophy should remain constant as a standard for future ERP testing.

It should be understood that as a pioneer effort, TERP is by no means the ultimate in software-controlled ERP testing. The following points can be considered for improvement. First, TERP was implemented around a specific operating system and consequently can only be used for DOS/TOS ERP testing. This does not, however, indicate that the TERP philosophy cannot be applied to operating systems in general. Second, TERP is not completely self-checking, in that visual review of the ERP test case output is required. Perhaps linkage to the system ERP message writer could be established which would allow TERP to internally verify the results of the test case. Third, multiple errors which result in cancellation cannot be combined into a single ERP test case. Perhaps linkage to the system cancel routines could be established that would allow control to be returned to the test case following



(A) A TERP I/O error simulation cycle begins here. Initial entry into the cycle is from the IOINT macro expansion. Additional entries occur if system ERP retries are performed.

(B) A TERP I/O error simulation cycle is complete here.

Fig. 3—TERP I/O Error Simulation Cycle

simulation of an error which normally results in cancellation.

The development of TERP has resulted in a powerful and flexible tool for software-controlled ERP testing. The success of the TERP project is demonstrated by its wide acceptance as the standard method of DOS/TOS ERP testing.

APPENDIX A

FUNCTION OF TERP COMPONENTS

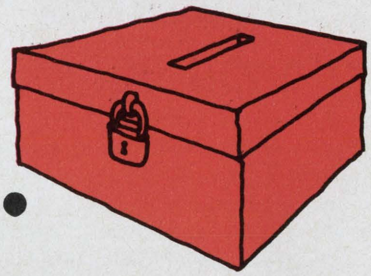
A. TERP B-TRANSIENTS

1. $$BSTIOP ENTRY B-TRANSIENT (core size approxmiately 225 bytes)
   a. Executed only at initial ERP test case case entry.
   b. Fetched by IOCTL initial entry.
   c. Modifies the problem program PSW by setting the storage protect key to 0 and turning off the problem state bit.
   d. Modifies the I/O new PSW address to point to TERP monitor.
   e. Modifies the supervisor SIO for error sense instruction to a LPSW with the PSW address pointing to the TERP monitor.
   f. Replaces the system EOJ B-transient name stored in the supervisor with the TERP exit B-transient name, $$BMEXIT.
   g. Modifies the address portion of certain instructions in the TERP monitor with addresses that cannot be determined until execution time.
   h. Exits by returning control to the ERP test case.
2. $$BMEXIT EXIT B-TRANSIENT
   a. Executed only upon ERP test case completion.
   b. Fetched by the system at ERP test case termination.
   c. Logs the system ERP retry count for test cases which result in cancellation.
   d. Restores the I/O new PSW address to original state.
   e. Restores the supervisor SIO for error sense to original state.
   f. Restores the system EOJ B-transient name in the supervisor.
   g. Exits by fetching the system EOJ B-transient.

B. TERP MACROS

1. IOCTL (maximum generation approximately 2700 bytes)
   IOCTL is the macro name. There are two optional operands: SYSTEM and TRACE. SYSTEM specifies the operating system under which the test case is to be executed. It may equal DOS or TOS, with DOS being the assumed value if the SYSTEM entry is omitted. The TRACE option should be specified when verification of I/O commands issued by system ERP is required. TRACE may equal SYSLST or SYS000–SYS014. TRACE specifies the symbolic unit to which the actual device being used for trace output is assigned. The actual device employed must be either a unit record printer or a magnetic tape de-

this plain, simple lockbox will help you find that better job at the IEEE Convention without fanfare and without cost.



Everyone's coming to the IEEE Convention searching for new ideas. A lot of Engineers will also be searching for new positions.

And Engineering Opportunities will help you. Our cost-free Resume Center will let you advance your career without subjecting you to the perils and problems of job hunting. We'll protect your privacy, let you keep full control and charge you nothing.

1. You hand pick the companies. Fill out the special resume form at the Center, listing your hotel room and local telephone number, or bring your own resumes. Companies in all areas and fields will be represented in the Resume Center. A free copy of Engineering Opportunities will be given you so that you may study the ads of the participating companies.

2. Free resume duplication service. Our Dennison operators will photocopy the special resume or your own one page resume FREE of charge and return ALL copies to you.

3. You hand deposit your resume. Next, you'll drop your resume in the lockbox of the company you're interested in. Only the company interviewer has the key.

4. You make your own appointment, your own arrangements. If the company interviewer is interested in you, he will leave a message at your hotel or at home for you to give him a call. You will set up the interview with him at your convenience. All arrangements are made by you and the company. No one else. There's no middleman, and no cost to anyone. You or the company.

EMPLOYMENT MANAGERS

For information on how your company can participate in the E/O Resume Center at the Americana Hotel in New York, during IEEE convention March 23rd thru March 26th, fill out coupon below.

Att. Mr. Norman Jacobs, Advertising Director
Engineering Opportunities Magazine
1020 Church Street, Evanston, Ill. 60204
Tele: 312-869-1244

Dear Mr. Jacobs: Please send me detailed information on participating in the E/O Resume Center, during IEEE

Name____________________

Title____________________

Address____________________

Company____________________

Phone____________________

City & State____________________

engineering opportunities resume center

Americana Hotel, New York City Sunday March 23rd, 1 PM to 7 PM, Monday 24th, 9 AM to 7 PM, Tuesday March 25th, 9 AM to 7 PM, Wednesday March 26th, 9 AM to 1 PM.

The E/O Resume Center is operated by ENGINEERING OPPORTUNITIES Magazine, 1020 Church St., Evanston, Illinois 60201. Norman Jacobs, Advertising Director

vice. Trace output records are 120 characters in length.

The I/O trace is active only during error simulation. It causes a trace of the I/O interrupts which occur from the monitored device and of the commands issued to the monitored device by the supervisor error SIO routine. The information provided per I/O interrupt trace is the I/O old PSW, CSW, CAW, and CCW chain. The CSW associated with the interrupt which triggers error simulation reflects the simulated error status. The information provided per error SIO trace is the CAW and CCW chain. The simulated error sense status is also provided if the command is SENSE I/O. The error SIO is traced at SIO time; therefore, the I/O old PSW and CSW are not applicable. Both the I/O interrupt trace and error SIO trace contain the current CAW and CCW chain pointed to by the current CAW. If there are more than five CCW's in the chain, on the first four and the last are displayed. TIC commands, except TIC to *–8 or *–16, are traced from the data address portion of the TIC CCW, which points to the next CCW in the chain.

IOCTL provides code for taking control from the supervisor when an I/O interrupt occurs. Initial program entry to IOCTL causes a B-transient, $$BSTIOP, to be fetched. The initialization code generated by IOCTL is only executed at initial program entry. All other entries to IOCTL enter the TERP I/O interrupt processing routine when an I/O interrupt occurs or when the supervisor issues a sense command to the device in error. The TERP monitor modifies the CSW and

sense status on the monitored device with user supplied error status. Control is then returned to the supervisor for processing the simulated error status.
IOCTL generates I/O subroutines for logging console messages, reading count key data, writing count key data, and seeking to DASD devices. Also, an I/O subroutine is provided for issuing a SENSE I/O command to existing or simulated I/O devices.

2. PUTMSG (generation approximately 100 bytes)
PUTMSG is the macro name. There is one operand enclosed in quotes. This macro provides the code for logging the expected test case results. The maximum length of this message is 80 bytes, however, successive PUTMSG macros are permitted.
3. STATUS (maximum generation approximately 70 bytes)
STATUS is the macro name. This macro generates code which provides the TERP monitor with the user supplied CSW and SENSE bytes for I/O error simulation. In addition, STATUS generated code provides the monitor with simulation time and number of simulations to be performed per error. CSW and SENSE status may be provided with the CSW and SENSE operands. The hexadecimal bytes supplied in this manner are used repeatedly during the error simulation, SETERR number of times.
The user may also supply the CSW and SENSE to be used for simulation in list form. If LIST is specified, the first two bytes of each entry in the list must be the CSW status. The remaining bytes for each entry (from 1 to 6) are the SENSE status. Entries must be fixed length and hex format. LIST allows simulation of multiple errors within the system ERP retry loop, SETERR number

of times. If the list is exhausted before SETERR has been satisfied, the monitor will recycle to the beginning of the list.

4. IOINT (generation approximately 110 bytes)
   IOINT is the macro name. DEVICE is the symbolic unit to which the actual device being monitored for error simulated is assigned. DEVICE may equal SYSRDR, SYSIPT, SYSPCH, SYSLST, SYSLOG, SYSLNK, SYSRES, SYSSLB, SYSRLB, or SYS000–SYS014. IOROUT is the name of an executable I/O subroutine during which the error simulation takes place. Linkage to the subroutine is made via register 6, therefore, the subroutine must terminate with a BR 6 instruction. IOROUT may equal one of the ERP I/O subroutines (UREAD, UWRITE, USEEK, UNOP) or the name of a user supplied subroutine. The user may perform I/O operations in his test case without using the IOINT macro. I/O operations requested in this manner execute without interference from TERP.
   When the IOINT generated code is executed and the simulated error results in a noncancellation condition, the TERP CSW COUNT, the field in which the TERP monitor counts attempted system ERP retries, is logged and then zeroed for further simulation requests by the user. The TERP CSW COUNT is always 1 greater than the system ERP retry count because it includes the initial failure.
5. MCINT (maximum generation approximately 575 bytes)
   MCINT is the macro name. DEVICE is the symbolic unit to which the actual device being monitored for CCC, CDC, or ICC error simulation is assigned. DEVICE may equal SYSRDR, SYSIPT, SYSPCH, SYSLST, SYSLOG, SYSLNK, SYSRES, SYSSLB, SYSRLB, or SYS000–SYS014. Dependent upon the parameters furnished, the MCINT generated code provides the appropriate error status in the CPU logout area. CSW error status may also be simulated for CCC, CDC, or ICC errors which are to be received by the specified CPU model via an I/O interrupt. Control is then given to the supervisor for processing the simulated error status.
6. ENDTST (generated approximately 5 bytes)
   ENDTST is the macro name. There are no operands. This macro provides code for terminating the user's program. This code includes an EOJ statement. The user's source program must terminate with an assembler END statement which contains a blank operand. ENDTST also generates a LTORG statement followed by a DS 0H statement. This causes the proper boundary alignment for user code which may follow the ENDTST macro statement.

APPENDIX B

SAMPLE ERP TEST CASE USING TERP

A. SOURCE FOR ERP TEST CASE GENERATION

```
// JOB DE32076
// EXEC ASSEMBLY
        TITLE 'DE32076–2314–SIMULATE OVERRUN'
```

An I/O interrupt from the 2314 causes this program to take control from the supervisor and simulate an "overrun" error by posting bit 38 of the CSW and bit 5 of sense byte 0. Supervisor ERP should retry error 10 times. Program size is approximately 1800 bytes.

EXECUTION

1. The ENDTST, IOCTL, IOINT, PUTMSG, STATUS macros must be cataloged in the source statement library at assembly time. Assembly time is 4 minutes.
2. B-transients $$BSTIOP, $$BMEXIT must be cataloged in the core image library at execution time. Execution time is 2 minutes.
3. Initialized 2314 scratch disk must be assigned to SYS000 at execution time.
4. Reply retry to I/O error message for SYS000.

OUTPUT

Test case is successful if the following message occurs on SYSLOG prior to EOJ—
OP14 "overrun"
CSW count = 00011

```
IOCTL
PUTMSG 'RESULT SHOULD BE 0P14 OVERRUN, EQUIP ERR, CSW COUNT = 00011'
STATUS CSW = 0200, SENSE = 0400, SIMTIME = CD, SETERR = MAX
IOINT DEVICE = SYS000, IOROUT = UREAD
ENDTST
END
```

B. RESULTS OF ERP TEST CASE EXECUTION

1. Console Typewriter Output

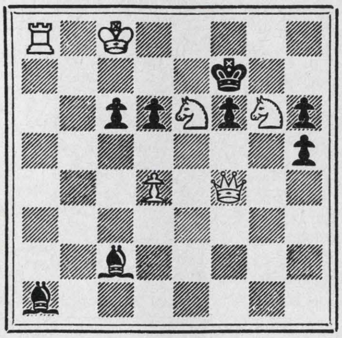
```
// JOB DE32076
// PAUSE ASSGN SYS000 TO 2314 SCRATCH DISK
// ASSGN SYS000, X'130'
// EXEC DE32076
   a) **RESULT SHOULD BE 0P14 OVERRUN, EQUIP ERR EXIT, CSW COUNT = 00011
   b) 0P14D  R  OVERRUN  SYS000 = 130
           CCSW = 1E1000208802000000  SNS = 040000000000
           CCB = 002056  SK = 000000000000
   c) retry
   d) **CSW  COUNT = 00011
   e) CSW  COUNT = 00000
      EOJ DE32076
```

2. Relationship of Console Typewriter Output
   a) Result of TERP PUTMSG macro.
   b) Result of system ERP routines.
   c) Operator response.
   d) Result of non-cancellation I/O error simulation. Actual system ERP retry count is 10. TERP CSW COUNT is 11 including initial failure.
   e) Result of TERP exit B-transient. Residual TERP CSW COUNT at EOJ.

CHECKMATE

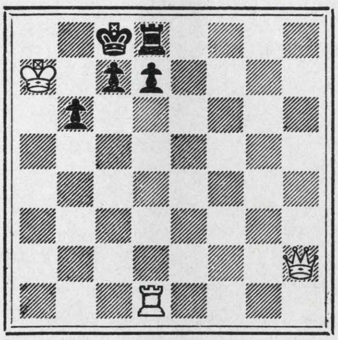
by
GEORGE N. VASSILAKIS
TRW SYSTEMS GROUP

Problem 1



A. A. Alekhine
White to Mate in Three

Problem 2



M. Euwe
White to Mate in Two

$10.00 will be paid to each problem accepted for publication.

Computer vs. Computer

	USSR (White)	USA (Black)
1	P–K4	P–K4
2	N–KB3	N–QB3
3	N–B3	B–B4
4	NxP	NxN
5	P–Q4	B–Q3
6	PxN	BxP
7	P–KB4	BxNch
8	PxB	N–B3
9	P–K5	N–K5
10	Q–Q3	N–B4
11	Q–Q5	N–K3
12	P–B5	N–N4
13	P–KR4	P–KB3
14	PxN	PxNP
15	RxP	R–B1
16	RxP	P–B3
17	Q–Q6	RxP
18	R–N8ch	R–B1
19	QxR mate	

YOUR MOVE

The response to the CHECKMATE announcement in the October issue of this magazine was outstanding.

Here are some excerpts from the mail that has been received:

> "A great idea to prevent the engineering mind from getting stalemated while at home. . . . Through the years, I have enriched my personal chess file with games and problems. I am interested in contributing some items about chess, if you so desire. . . . My fellow workers here at Martin are also awaiting this new feature. . . . How about a contest in which programs written to play a game are pitted against each other. . . . Sock it to us, Botvinnik. . . . Let's get a 'pressure group' going to lobby for government subsidy to promote the Royal Game. . . . Your column would be a fine addition to our activities at lunch time. . . . Suggest items of scientific historical significance, technological connections, etc. . . . As a member of the ROOKS & CROOKS CHESS CLUB here at the Arizona State Prison it was with great interest that I read in your October issue about the possibility of a monthly chess column being published in your fine magazine. . . ."

And so we have decided to start the New Year with the addition of CHECKMATE as a monthly feature. Its success of course will depend on YOU the reader.

To begin with, this author would like to express some thoughts and ideas about this column and ask you to participate by adding new ideas or new versions of old ideas.

- Computerized chess will undoubtedly have high priority since this writer and many readers of this magazine are in the programming business. Any information on what has already been done in this area will be greatly appreciated.
- Quotations from Benjamin Franklin's "The Morals of Chess" will appear from time to time.
- The world champions will be discussed and some of their best games will be published.
- Current news on the activities of Bobby Fisher and other internationally known chess players will be included.
- The "MINIGAME" will make its appearance periodically. This will be a short game played by masters in less than 20 moves.
- The history of chess will be researched thoroughly.

COMPUTERIZED CHESS

On November 20, 1966, a computer at the Institute of Theoretical and Experimental Physics in Moscow made the historic move P–K4 and a computer at Stanford University, California, responded with P–K4. On March 10, 1967, the Russian computer made its 19th move—QxR mate! Several other games were played simultaneously and the USSR computer finally won the tournament.

Does this mean the Russian computers are smarter than the American computers? Of course not. In fact, we can safely say that the IQ of each computer was zero. Why, then, did a slow unsophisticated computer play better chess than its opponent, which was probably the world's fastest and most sophisticated computer at that time? The answer, of course, is the man behind the computer. Each machine was programmed by leading mathematicians and the results merely indicate that the Russian programmers had written a better program.

The motive for the contest was more scientific than sporting. Programming chess is not for the average programmer, and the leading mathematicians can do very little with today's computer technology. Chess falls in the category of problems which have exact solutions but such solutions are beyond the capabilities of present computers. The main problem is the astronomical number of combinations of moves the computer must analyze.

Will computers ever play a good game of chess? Botvinnik believes the time will soon come when a computer will be able to play like a grand master. Professor M. Euwe doesn't think so. He believes computers will be rated as amateurs. A. N. Walker, an astronomer from Manchester University, England, points out that computers are not specifically designed with chess in mind; and a custom-built computer with special circuitry could increase the speed of a chess program 50–60 times.

During the past ten years, computer speeds have increased by a factor of 1,000 or more and present advances in computer technology indicate that a further increase in the order of 1,000 or more might easily be obtained in the next ten years. The trend of micro-miniturization in computer technology indicates that someday we may have computers smaller than a chess clock and a player may be able to enter his company's computer in a chess tournament.

The prices of computers are also going down and we may see the day when a school like MIT or Stanford will be able to dedicate a high speed computer to postal chess. Think of how much analysis such a computer could perform in the 72 hours that a postal chess player is allowed from the time he receives his opponent's move! And further analysis could be performed while waiting for the opponent to answer.

Programming Recursive Functions In Fortran

John Morris
Computer Institute for Social Science Research
Michigan State University

■ In programming a large package of statistical routines (4), I wanted to include functions to compute a variety of probability measures. For many of these, the only formulas that I could find were stated in recursive form, and the rule-books all say that you can't do recursion in Fortran. I thought I'd try it any way. The results show that the rule-books are wrong.

A recursive function is one which, in its definition, includes the function itself as one of its own arguments. You might define the factorial, k!, as follows:

$$k! = k\,(k\text{–}1)! \quad \text{for } k > 0$$
$$k! = 1 \quad \text{for } k = 0$$
$$k! = 0 \quad \text{for } k < 0.$$

This looks elegant and simple, so let's write a Fortran function to compute it:

```
      FUNCTION FACT (K)
      IF (K) 1, 2, 3
    1 FACT = 0
      RETURN
    2 FACT = 1.0
      RETURN
    3 FACT = FACT (K -1)
      RETURN
      END
```

If you fed this function to your compiler, one of three things would happen: (1) It would compile and execute perfectly. If so, you have a *very* sophisticated compiler, and you deserve nothing but admiration and envy. But it's also a non-standard one that may take longer to compile *every* program because of the need for special treatment of recursive function calls.

(2) It would compile, but a call to it with K > 0 would mean serious trouble—possibly a loop within the function.

(3) It wouldn't compile at all, but would end with a diagnostic about your statement 3. You *could* trick the compiler, by setting up two identical functions with different names, and have each of them call the other, but this would put you right back in situation (2), with nowhere to go, and nothing to do but loop until your time is up.

What caused the trouble at (2)? When one routine calls a function or subroutine, the program must store a return address. Otherwise, when the RETURN statement is executed, the computer won't know where to go to get its next instruction. Most compilers, however, provide only one location for storing this return address. This means that if a function calls itself, it clobbers the old return address and inserts a new address where the old one used to be. As a result, the program can't return to the original calling routine. It simply doesn't know where to go.

Some compilers do provide for this situation by using a stack of addresses. Each time that the function calls itself, a new address goes on top of the stack. Then, each time that it returns, it removes the top address from the stack. This means that the return address never gets lost—it's always somewhere, down in the stack, waiting until the time that it's needed.

But the bookkeeping required for stacks like these takes valuable time, and few users think that it's worth it, when only a small fraction of their programs will ever use the facility. For that reason, most compilers omit it.

One solution, which Rice suggests (3), is simply to turn recursions into iterations. Anyone can see that my FACT program could be written with a DO loop, running from 1 to K, that would execute in half the time that my recursive function would require. All of which would be grand, except that, try as I would, the formulas that I had to evaluate simply would not turn into neat iterations. So I tried another way.

Standard recursive techniques require that we store the return address in a stack. Since this would be hard to do in Fortran, I tried storing the arguments of the function itself in one or more stacks. One of the probability functions, which computes Kendall's tau correlation, required the storage of two arguments, and I set aside a 2 x 1500 scratch array for that purpose. The other, the Mann–Whitney statistic, took three stacks, and a 3 x 1000 array was set aside. Ackermann's function, which serves no really useful purpose except to demonstrate what a non-simple recursive function is like, required storage for only one argument.

market place

The Classified Ad Department of SOFTWARE AGE

HELP WANTED

PART TIME PROGRAMMERS/ANALYSTS with COBOL experience in such areas as management information systems, inventory control, accounting, personnel, or general manufacturing or information retrieval applications. Degree not necessary. All work will be corresponded through the mail on high contract rates. Send brief resume to Box SA-101, Software Age., 1020 Church Street, Evanston, Illinois 60201.

MARKETING REPRESENTATIVE
Marketing Representative wanted for a progressive data processing service bureau located in the Chicago area. Applicant should be dynamic, desirous of a large income, willing to work for it and have a working knowledge of programming and/or systems analysis. Excellent benefits and opportunity to move into management. Our employees know of this position opening.
Send resume in confidence to:
Box 7398, Chicago, Illinois 60690

EMPLOYMENT AND SEARCH CO'S

DATA PROCESSING POSITIONS IN THE SOUTHWEST
Contact:
DATA PROCESSING CAREERS
Richard Kemmerly, Suite 1109, Stemmons Towers West, Dallas, Texas 75207. 214/637–6360.

PROFESSIONAL SERVICES

CONSULTANT: Excellent experience designing, programming, debugging and documenting commercial applications. Hourly or job basis. Box 11-H, 272 First Ave., N.Y.C. 10009.

NCR 315/615 PROGRAMMING & CONTROLLED SYSTEMS offers professional programming services. Write Controlled Systems, Box 1, Dabel Branch, Dayton, Ohio 45420.

EQUIPMENT

ADD/SUBTRACT 6-DIGIT HEXADECIMALS in seconds with 100% accuracy. The pocket HEXADDER, $15. Free brochure from HEXCO, INC., Dept. 5, P. O. Box 55588, Houston, Texas 77055.

market place

CLASSIFIED ADS in software age

where you can reach **110,000** programmers, mathematicians, analysts and EDP managers.

CLASSIFIED ADVERTISING

Non-Display Classified: For firms or individuals offering commercial products or services, 75¢ per word (including name and address). Minimum order $7.50. For Blind Ad Service, an additional $10.00. For "Positions Wanted" Ads, 55¢ per word (including name and address). No minimum. **Payment must accompany copy** except when ads are placed by accredited advertising agencies. Frequency discounts: 5% for 6 months; 10% for 12 months paid in advance.

Display Classified: One inch by one column, $70.00. Column width 2¼". Photographs accepted for an additional $20.00. Advertiser to supply all photo, art, cuts, or camera ready copy.

General Information: One inch display Help Wanted and Employment Services ads **will be** accepted in the classified section. Employment ads ⅛ page or more will appear run of book, will be keyed to the resume form in back of publication, and will qualify to free daily resume service.

Closing Date: 1st of preceding month (for example, May issue closes April 1st).

Send order and remittance to: Classified Dept., Software Age, 1020 Church Street, Evanston, Illinois 60201.

software age
1020 CHURCH STREET • EVANSTON, ILLINOIS • 60201

CLASSIFIED ADVERTISING ORDER FORM

Please refer to the above information for complete data concerning terms, frequency discounts, closing dates, etc. Cash with order.

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25
26	27	28	29	30
31	32	33	34	35

________Words { @ 55¢ (Reader Rate) / @ 75¢ (Commercial Rate) } = $________

________Include Photograph @ $20.00 additional (**Display Ads Only**)

☐ Run Ad Blind $10.00 Insert________time(s) Total Enclosed $________

NAME________________

ADDRESS________________

CITY________ STATE________ ZIP________

Signature________________

WORD COUNT: Include name and address. Name of city (Des Moines) or of state (New York) counts as one word each. Zone or Zip Code numbers not counted. (Publisher reserves right to omit Zip Code if space does not permit.) Count each abbreviation, initial, single figure or group of figures or letters as a word. Symbols such as 35mm, COD, PO, AC, etc., count as one word. Hyphenated words count as two words.

SA-1

```
      FUNCTION ACKER (M,N)
      COMMON /SCTCH/ STAK(3000)
      ACKER=O.
      IF (M.LT.O.OR.N.LT.( ) RETURN
      ACK=O.
      XM=M
      XN=N
      INDEX=O
    7 IF (XM.GT..01) GO TO 3
      ACK=XN+1.
      IF (INDEX) GO TO 4
      ACKER=ACK
      RETURN
    3 IF (XN.GT..01) GO TO 5
      XM=XM—1.
      XN=1.
      GO TO 7
    5 IF (INDEX.LT.3000) GO TO 6
      PRINT 400
  400 FORMAT (10X,*STACK CAPACITY
        EXCEEDED*)
      ACKER=O.
      RETURN
    6 INDEX=INDEX+1
      STAK(INDEX)=XM—1.
      XN=XN—1.
      GO TO 7
    4 XM=STAK(INDEX)
      XN=ACK
      INDEX=INDEX—1
      GO TO 7
      END
```

FIGURE 1

The technique illustrated here can easily be extended to more complex functions by providing additional storage stacks. For example, a stack might be used to hold values for use in a computed GO TO statement, which would then permit selection of a number of alternative routines.

Conceptually, the functions take the form of tree structures. At each branch-point of the tree, we generate one or more sets of new arguments for the function. To conserve space in the stack, we test these new arguments before we store them, to find out whether they can

FIGURE 2

```
C*****DETERMINES NUMBER OF I-PART
          PARTITIONS OF K, NO PART
          GREATER THAN J
      COMMON/SCTCH/ISTAK(3,1000)
      PARX=O.
      SUM=O.
C     INITIALIZE FOR FIRST ELEMENT IN
          STACK
      N1=I
      N2=J
      N3=K
      INDEX=O
C     PLACE LARGER N IN SECOND
          POSITION
   11 IF (N1.LE.N2) GO TO 12
      NT=N1
      N1=N2
      N2=NT
C     CHECK FOR AVAILABLE SPACE AND
          ERROR RETURN
   12 IF (INDEX.LT.1000) GO TO 15
      PARX=O.
      PRINT 400
  400 FORMAT (10X,*CAPACITY EX-
          CEEDED*)
      RETURN
C     LOAD FIRST ELEMENT IN STACK
   15 IF ((N3—N2).LT.0) GO TO 16
      INDEX=INDEX+1
      ISTAK(1,INDEX)=N1—1
      ISTAK(2,INDEX)=N2
      ISTAK(3,INDEX)=N3—N2
C     EVALUATE SECOND ELEMENT
   16 N2=N2—1
   10 IF (N3.LT.O) GO TO 13
      IF (N1.AND.N2) GO TO 11
      IF (N3) GO TO 13
      SUM=SUM+1.
      PARX=SUM
C     UNSTACK TOP ELEMENT AND
          CHECK FOR END OF STACK
   13 IF (.NOT.INDEX) RETURN
      N1=ISTAK(1,INDEX)
      N2=ISTAK(2,INDEX)
      N3=ISTAK(3,INDEX)
      INDEX=INDEX—1
      GO TO 10
      END
```

be evaluated. When all the needed information is stored in the proper stack, we remove the top element from the stacks and begin work on it. This may require storing more elements in the stacks. This process is repeated until all the values in the stack have been checked. When the stacks are empty, the process is complete.

We begin at the trunk of the tree and climb upward, always turning left at each branch-point, until we come to the left-most twig. At this point, we can evaluate the function, and we save the results. Then we back up to the last branch-point, and try the next twig, and so on, until every twig has been visited. Then we are through, and our results are the value of the function.

There is a secret here, which is that we never try to store more information in our stack than we have to. By storing only the route by which we arrived at the last twig, we save storage space that could quickly get eaten up by some of the more traditional recursive methods.

The method makes the most sense when you have examples, three of which are given here. Notice, incidentally, that these are not toy problems. The functions to compute the section and third formulas are being used in a production system, where there is no other way (that I know of) to compute them.

FIGURE 3

```
      FUNCTION PTAU (TAU,N)                                          PTA00010
C     EXACT PROBABILITIES FOR KENDALL TAU—RANK CORRELATION
      DIMENSION ISTAK (2,1500)
C     CONVERT TAU TO S VALUE                                         TO 00030
      XN=MX=(N*(N—1))/2                                              TO 00040
      IS=TAU*XN+.01                                                  TO 00050
C     JUMP TO NORMAL APPROX FOR LARGE N                              TO 00060
      IF (N.GT.7) GO TO 2                                            TO 00070
      PTAU=O.                                                        TO 00080
C     COMPUTE FOR ALL VALUES GREATER THAN GIVEN S, SUM THESE         TO 00090
      DO 100 I=IS,MX                                                 TO 00100
C     INITIALIZE STACK                                               TO 00110
      INDEX=1                                                        TO 00120
      ISTAK(1,1)=N                                                   TO 00130
      ISTAK(2,1)=I                                                   TO 00140
      SUM=O.                                                         TO 00150
    4 IN=ISTAK(1,INDEX)                                              TO 00160
C     TEST TO DETERMINE WHETHER THESE VALUES CAN BE EVALUATED        TO 00170
      NN=(IN*(IN—1))/2                                               TO 00180
      KS=ISTAK(2,INDEX)                                              TO 00190
C     IF SO, JUMP TO 0 OR 1 VALUE FOR THIS PAIR                      TO 00200
      IF (XABSF(KS)—NN) 5,6,100                                      TO 00210
    5 MAX=IN—1                                                       TO 00220
      MIN=—MAX                                                       TO 00230
      NN=(MAX*(MAX—1))/2                                             TO 00240
C     DECOMPOSE STACKED VALUES INTO EQUIVALENT VALUES AND
        RESTACK THESE                                                TO 00250
      DO 101 J=MIN,MAX,2                                             TO 00260
      KJ=KS+J                                                        TO 00270
      IF (XABSF(KJ)—NN) 9,10,101                                     TO 00280
    9 ISTAK(1,INDEX)=MAX                                             TO 00290
      ISTAK(2,INDEX)=KJ                                              TO 00300
C     CHECK STACK FOR OVERFLOW                                       TO 00310
      IF (INDEX.LT.1500) GO TO 11                                    TO 00320
C     ERROR RETURN                                                   TO 00330
      PRINT 401                                                      TO 00340
  401 FORMAT (10X,*PTAU CAPACITY EXCEEDED*)                          TO 00350
      PTAU=O.                                                        TO 00360
      RETURN                                                         TO 00370
   11 INDEX=INDEX+1                                                  TO 00380
      GO TO 101                                                      TO 00390
   10 SUM=SUM+1.                                                     TO 00400
  101 CONTINUE                                                       TO 00410
C     RETURN TO CHECK PRECEDING VALUE IN STACK                       TO 00420
      INDEX=INDEX—1                                                  TO 00430
      IF (INDEX) 4,100                                               TO 00440
    6 SUM=SUM+1.                                                     TO 00450
  100 PTAU=PTAU+SUM                                                  TO 00460
C     EVALUATE PROBABILITY AND RETURN                                TO 00470
      PTAU=PTAU/FACT(N)                                              TO 00480
      RETURN                                                         TO 00490
    2 S=IS—1
      OBS=N
      Z=S/XN*SQRT(9.*XN/(2.*OBS+5.))
      PTAU=PNORM(Z)                                                  TO 00510
      END                                                            TO 00520
```

Example 1. Ackermann's function has no interest except for the fact that it has repeatedly been used as an illustration of a recursive function which is not simply recursive. Rice (3) has given one form of the definition of the function:

If $m = 0$, ACKER $(0, n) = n + 1$

If $m \neq 0$, and $n = 0$, ACKER $(m, 0) =$ ACKER $(m-1, 1)$

If $m \neq 0$ and $n \neq 0$, ACKER $(m, n) =$ ACKER $(m-1,$ ACKER $(m, n-1))$.

The program given in figure 1 is recursive, not simply iterative, storing the values of $m - 1$ in STAK while evaluating the right-hand argument in the last form of the function, then returning to the stack to retrieve this value for further evaluation. In principle, the program could evaluate any pair of arguments within the floating-point capability of the machine. Actually, the importance of the function did not seem to warrant extending computation time beyond a few seconds.

Example 2. Given the values 0, 1, 2, . . . m, in how many ways can n of these be combined (disregarding order, with repetitions permitted) to form the sum U? Or, stated another way, in how many ways can a set of U indistinguishable members be partitioned into m or few subsets, with maximum subset size n?

Mann and Whitney (2) give a recursive method of calculating this combinatorial problem, which forms the numerator of the probability associated with their statistic U:

Given m, n, U, $p_{mn}(U) = p_{mn}(U) / (m_m + n)$

where $p_{mn}(U) = p_{mn}(U) = p_{n-1m}(U-m) + p_{nm-1}(U)$,

with the following termination criteria:

For $U < 0$ $\quad p_{ij}(U) = 0$

For $i = 0$ or $j = 0$, and $U = 0$ $\quad p_{ij}(U) = 1$

For $i = 0$ or $j = 0$, and $U > 0$ $\quad p_{ij}(U) = 0$.

These termination criteria tell us when we have come to the end of a twig and can evaluate the function. The initial definition requires that the branch be split in two. In terms of a tree structure, each branch-point therefore has two branches coming from it. The check the first of these branches to determine whether it can be evaluated by one of the termination criteria; if it can, we add 1 to the SUM when it is appropriate and proceed to the other branch. If it cannot, we store the three values associated with m, n, and U in the three stacks, and proceed to evaluate the other branch. When both branches of a given branch-point have been evaluated, we return to the stacks to retrieve the last stored value. When the stacks are empty, the function has been evaluated. The program is shown in figure 2.

Example 3. Kendall's rank correlation coefficient *tau* (1) has a probability associated with it which can be expressed as follows:

Given n and $S = \frac{1}{2}\tau\, n\,(n - 1)$, the significance level for τ is given by

$$P(\tau) = 1/n! \sum_{i = S}^{\frac{1}{2}n(n-1)} U(n, i)$$

$$\text{where} \quad U(n, i) = \sum_{j = -(n-1)/2}^{(n-1)/2} U(n-1, 2j),$$

with the following termination criteria:

Let $nn = n(n-1)/2$
Then if $|i| > nn$, $U(n, i) = 0$
and if $|i| = nn$, $U(n, i) = 1$.

The program shown in figure 3 stacks values for n and S, decomposes these into the required sum, checking each pair of values before stacking them to determine whether it can be evaluated. Although the program is a good deal more complex than the first two programs, the method is exactly the same.

Three external functions are called by the functions in the second and third examples. These are FACT, which evaluates n!, COMB, which evaluates $\binom{h}{k}$, and PNORM, which calculates the area lying under the normal curve to the right of the given ordinate. ■

REFERENCES

1. Maurice G. Kendall, **Rank Correlation Methods**, London: Charles Griffon & Co., 1955, p. 67.
2. H. B. Mann and D. R. Whitney, "On a Test of Whether One of Two Random Variables is Stochastically Larger than the Other," **Annals of Mathematical Statistics**, 18 (1947), 50–60.
3. H. Gordon Rice, "Recursion and Iteration," **Communications of the ACM**, 8 (1965), 114–15.
4. John Morris, "Nonparametric Statistics," Technical Report No. 40, East Lansing: Computer Institute for Social Science Research, Michigan State University, 1967.

new products

The DEC Symbolic Assembler for the IBM 1130 will **assemble source programs for the DEC family-of-eight computers** and produce a binary object tape suitable for input to the DEC Binary Loader. An optional assembly listing may be obtained showing all source and object code, with source errors clearly flagged.

A source library feature is provided with the facility for storing and cataloging commonly-used routines. These routines may be called for and assembled in-line in a source program. A librarian program provides all the necessary maintenance functions for the library.

Another optional feature provides the automatic generation of link address constants in page zero for out-of-page references in the program.

The assembler offers the advantages of card input, printer output, extensive diagnostics and rapid assemblies.

An outstanding feature of the system is that it is coded almost entirely in FORTRAN, allowing easy expansion and modification.

The system is available on a purchase basis or as a service from Infotec, Inc.

For more information, circle No. 50 on the Reader Service Card

* * *

Tab-Tub data printout file handles not only National data binders but other systems as well. Its one piece, all welded construction features attached cover that slides back into rear of file unit when in use. Large ball casters permit easy maneuverability even in close quarters.

Overall outside dimensions: width 17⅜", depth 28", height 29".

Removable divider permits filing data binders in either burst or unburst form—or in combination—in the same rack. The purchase price includes 6 sets of suspension bars and 10 index tabs.

For more information, cricle No. 51 on the Reader Service Card

* * *

Programmatics has introduced **a new concept in software packages** with the announcement of "PI SORT." This package is externally identical to DOS Sort for the IBM 360, or what the hardware industry refers to as "plug compatible." PI SORT, which sorts fixed length records two to four times faster than DOS Sort, uses the same Job Control Language and Sort control cards. First installations are scheduled for February 1969. A prototype model currently operating sorts files up to twice the size of the largest file which can be sorted by DOS Sort while maintaining the ratio of two to four reduction in time. PI SORT will include all the options of DOS Sort.

For more information, circle No. 52 on the Reader Service Card

* * *

A complete off-line plotting system is now available from Houston Instrument, Division of Bausch & Lomb, Inc. The company recently announced the addition of a magnetic tape reader for use with their CØMPLØT$_{tm}$ Digital Plotter DP-1. The new reader will be known as the CØMPLØT$_{tm}$ MTR-1 Magnetic Tape Reader.

Combining the DP-1 and MTR-1 provides the user with a system capable of generating plots from tapes of various densities. The standard MTR-1 will accept 200 BPI, 7 track ½" tape. 556 BPI, 7 track and 800 BPI 9 track are also optionally available. The unit accepts 8½" reels or smaller. Plotting rate is up to 300 increments/second. Also featured on the MTR-1 is single capstan drive and single edge tape guides to reduce tape wear.

For optimum user convenience, the MTR-1 is supplied in a desk that provides ample room on top for the DP-1 plotter.

For more information, circle No. 53 on the Reader Service Card

* * *

General Electric announced a new shared-access information handling system for use with its 200-line computers.

Called the Remote Access Editing System (RAES), the new software package allows up to 20 simultaneous users to write material into computer storage files, retrieve information from those files, change data, and format the text.

A General Electric spokesman said the system would be especially useful for any organization with a sizable amount of written material that is used and re-used.

RAES can be used with Teletype Models 33, 35, and 37; IBM Model 2741, and Friden 7100. Full ASCII coding provides for extra characters as well as upper and lower case printing.

The RAES technique allows the user to build a file using punched paper tape or magnetic tape, and it can erase errors in individual characters, a string of characters, lines, or the entire file.

RAES is designed to produce manuscripts automatically.

For more information, circle No. 54 on the Reader Service Card

* * *

Softpak, Inc. has designed a complete **computer mailing system** which allows anyone with access to an IBM 360 to produce his own personalized computer letters for approximately 2¢ each. In addition, the system permits an organization to maintain and update its own magnetic tape files, store variable length abstract, historical and demographic information on individuals, and prepare mailing labels by zip, name or other characteristics.

The ability to store (and easily modify) variable length data is a key feature of the program. The user must decide what known characteristics should be included and have the capability of storing this information along with the name.

The complete system includes seven separate computer programs, and organizations which have already computerized mailing lists may purchase an individual program for computer letters. This program produces letters in upper and lower case at the rate of 2,000 per hour, one and two pages or two different letters at the same time.

For more information, circle No. 55 on the Reader Service Card

Digitronics Corporation has announced Dial-o-verter Model 5079, a new **data transmission terminal,** which provides two-way communications of punched paper tape. The Model 5079 accommodates input blocks of any size for higher throughput rates. Using standard telephone facilities, the Model 5079 transmits data at rates up to 300 characters/second and receives at 100 characters/second rate.

For more information, circle No. 56 on the Reader Service Card

* * *

DATA TECHnology, Inc., announces the introduction of its new, **IC Bi-Directional Counters** which count and reverse up to rates in excess of 5 MHz. These units are compatible with the OPTECON and PHOTOSYN as well as most other incremental encoders, and feature all TTL integrated circuitry on plug-in PC cards. Typical application areas include numerical and process control systems or any application requiring the net accumulation of bi-directional increments.

The counter's universal input is a unique feature which accommodates quadratured square wave (x1, x2, x4), pulse (up/down), or pulse/level. Single and multi-axis counters provide ± 4 to 6 decades, with in-line (Nixie) numerical display, BCD output, panel-mounted manual reset, overflow indicator, and remote display option. These counters are conveniently packaged in a self-contained, rack-mountable cabinet (½-rack wide per axis), and operate from a 110–115 VAC, 60 Hz supply.

For more information, circle No. 57 on the Reader Service Card

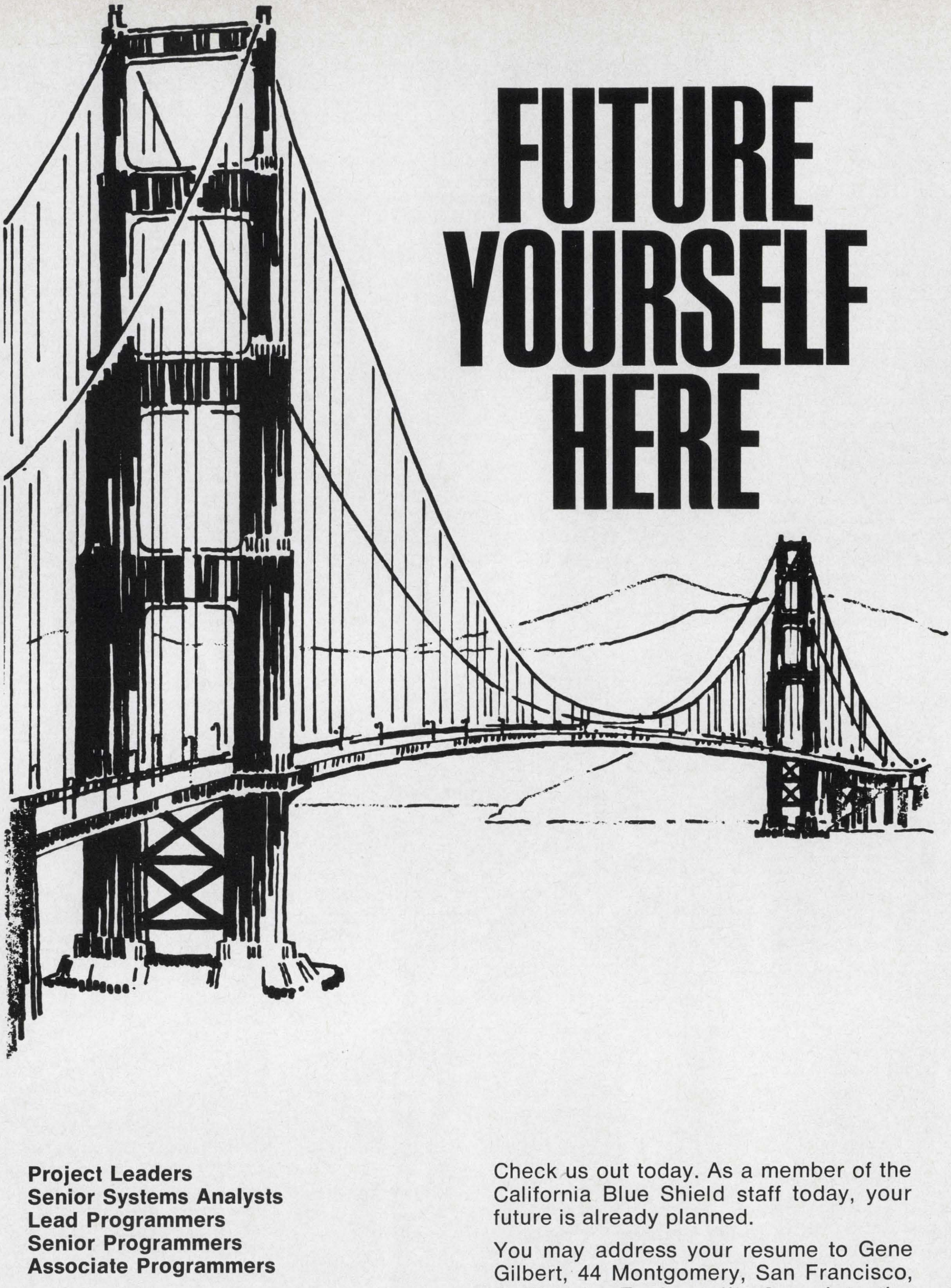
* * *

The M301 **Analog Multiplier** by Intronics, Inc. is a compact encapsulated solid state multiplier achieving low cost through the use of integrated circuits. It employs the pulse height–pulse width modulation principle to give linearity and temperature stability in low bandwidth applications. Frequency response is DC to 1 kHz (—3dB) for both inputs. The unit operates linearly through zero unlike quarter-square type multipliers. Zero offset error is better than 50mV worst case. Additional features include ±volt differential inputs; ±10 volt output at 5mA with short circuit protection; output impedance of 0.1 ohm maximum; and full scale linearity better than 0.2%.

Module size is 3" x 2" x ⅝" and power requirements are ±15 VDC at 60 mA maximum. Applications include power measurement, phase detection, linear voltage controlled attenuation, and carrier modulation.

For more information, circle No. 58 on the Reader Service Card

FUTURE YOURSELF HERE



Project Leaders
Senior Systems Analysts
Lead Programmers
Senior Programmers
Associate Programmers

California Blue Shield has grown to meet the demands of ever-widening computer applications with the most sophisticated third-generation equipment available. This dynamic growth has created new opportunities for energetic professionals.

As a computer professional with California Blue Shield you will enjoy the unmatched benefits offered by the largest organization in its field plus the incomparable charm of San Francisco. Your inquiry will receive an immediate response.

Check us out today. As a member of the California Blue Shield staff today, your future is already planned.

You may address your resume to Gene Gilbert, 44 Montgomery, San Francisco, California. Easier still: Complete the resume form on the facing page and ask Software Age to forward it to us. California Blue Shield is an equal opportunity employer.

CALIFORNIA
BLUE SHIELD
CALIFORNIA PHYSICIANS' SERVICE

software age

CONFIDENTIAL INQUIRY

Your original copy of this form will be retained at the offices of SOFTWARE AGE and will be used for no other purpose than to notify the specific firms which you have checked (on the reverse side) of your interest.

THIS INQUIRY IS IN DIRECT RESPONSE TO YOUR ADVERTISEMENT IN SOFTWARE AGE MAGAZINE

TYPE OR PRINT CLEARLY FOR PHOTO REPRODUCTION

JOB DESIRED: ______________________________

List computer hardware knowledge (names of systems, tape, disk, terminals, etc.): ______________________________

Programming specialties and years of experience (commercial, scientific, theoretical, experimental, analog, etc.): ______________________________

Systems programming on which you have had development experience (compilers, assemblers, executives, monitors, O.S., etc. Indicate for what computer): ______________________________

Programming languages used and extent of experience (COBOL, FORTRAN, etc.): ______________________________

Applications programmed (aerospace, banking, insurance, math subroutines, compilers, etc.): ______________________________

Systems analysis experience (card design, flow charting, operation analysis, etc.): ______________________________

EDP management experience (include years and number of people reporting to you): ______________________________

SALARY: ____________ (current) ____________ (desired) **DATE OF AVAILABILITY:** ____________

EDUCATION: Indicate major as well as degree unless self-explanatory.

Degrees			
Years			
Schools			

EMPLOYMENT: Indicate present employment and previous jobs below.

Employer			
City			
Years	to	to	to
Title or Function			

Name ______________________________ Age ________

Home Address ______________________________ Home Phone ________

______________ (city) ______________ (state) ______________ (ZIP code) U.S. Citizen? ________

Security Clearance ______________ Location Preference ______________

Marital Status ______________

Military Status ______________

BE SURE YOU HAVE CHECKED ON REVERSE SIDE THE COMPANIES YOU WANT TO SEE THIS INQUIRY. PUT FORM IN STAMPED ENVELOPE AND MAIL TO:

software age
MAGAZINE
1020 CHURCH ST., EVANSTON, ILL. 60201

AFTER MARCH 10, THIS FORM WILL NOT BE PROCESSED

WRITE ADVERTISERS DIRECTLY OR WRITE US FOR CURRENT S/A ISSUE

check your interests here

Fill in the confidential inquiry form on the other side of this sheet. This form provides all the information advertisers require to screen applicants. If further information is desired, you will hear from the advertiser direct. Then, check below the boxes of those companies to which you want copies of your form sent. Mail to SOFTWARE AGE, 1020 Church Street, Evanston, Illinois 60201. (Please do not send us your own resume. We will only process this form. A new form must be filled out for each issue in which you are answering ads.)

	Page
☐ 1. Abbott Laboratories	14
☐ 2. Army & Air Force Exchange Service	40
☐ 3. Booz • Allen & Hamilton, Inc.	32
☐ 4. California Blue Shield	44
☐ 5. Collins Radio Co.	20
☐ 6. Encyclopaedia Britannica, Inc.	15
☐ 7. First National Bank of Chicago	3
☐ 8. General Electric Co., Information Service Dept.	23
☐ 9. Honeywell, Inc. Electronic Data Processing Div.	5
☐ 10. Hughes Aircraft Co., Aerospace Div.	13
☐ 11. Hughes Aircraft Co., Fullerton	31
☐ 12. IBM Corp.	19
☐ 13. IIT Research Institute (Chicago)	17
☐ 14. Itek Corp.	4
☐ 15. Lockheed-Georgia Co.	30
☐ 16. Lockheed Missiles & Space Co.	26
☐ 17. Memorex Corp.	35
☐ 18. Miles Laboratories, Inc.	11
☐ 19. New York State Dept. of Civil Service	20
☐ 20. Philco-Ford Appliance Div.	35
☐ 21. Pratt & Whitney Aircraft	28
☐ 22. RCA, Information Systems Div.	7
☐ 23. Sikorsky Aircraft	21
☐ 24. Sylvania	4th Cover
☐ 25. Systems Engineering Laboratories	29
☐ 26. Univac Federal Systems Div.	34

EMPLOYMENT AND SEARCH AGENCIES

	Page
☐ 27. American Computer Personnel, Inc.	40
☐ 28. Callahan Center for Computer Personnel	26
☐ 29. Computer Careers, Inc.	32
☐ 30. Computer Personnel Agency, Inc.	42
☐ 31. Computer Personnel Consultants	14
☐ 32. Computer Personnel, Inc.	16
☐ 33. Data Management Services, Inc.	30
☐ 34. Drew Personnel Placement Center	10
☐ 35. Robert Half Personnel Agencies	34
☐ 36. Input, Inc.	28
☐ 37. Everett Kelley Associates, Inc.	30
☐ 38. LaSalle Associates	42
☐ 39. Lawrence Personnel	20
☐ 40. Longberry Employment Service, Inc.	16
☐ 41. Management Scientists, Inc.	12
☐ 42. Pomeroy Associates	16
☐ 43. RSVP Services	34
☐ 44. Sheridan Associates	16
☐ 45. Source EDP	27
☐ 46. Wells Recruiting Systems, Inc.	41

PRODUCTS AND SERVICES

(Use Reader Service Card)

Anaheim Publishing Co.	2nd Cover
Boorum & Pease	12
Library of Computer & Information Sciences	3rd Cover
Magnetic Aids, Inc.	36
North American Institute of Systems & Procedures	35
Wassell Organization, Inc.	36

☐ I do not now receive S/A. Please enter my FREE subscription.

Name .... Street Address ....

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

Prime Experience in What Industry .... / My Specialty ....

☐ Technical Degree ☐ Non-Technical Degree ☐ No Degree

.... Year Born ☐ I Have Analog/Hybrid Experience

software age

1020 Church Street
Evanston, Illinois 60201