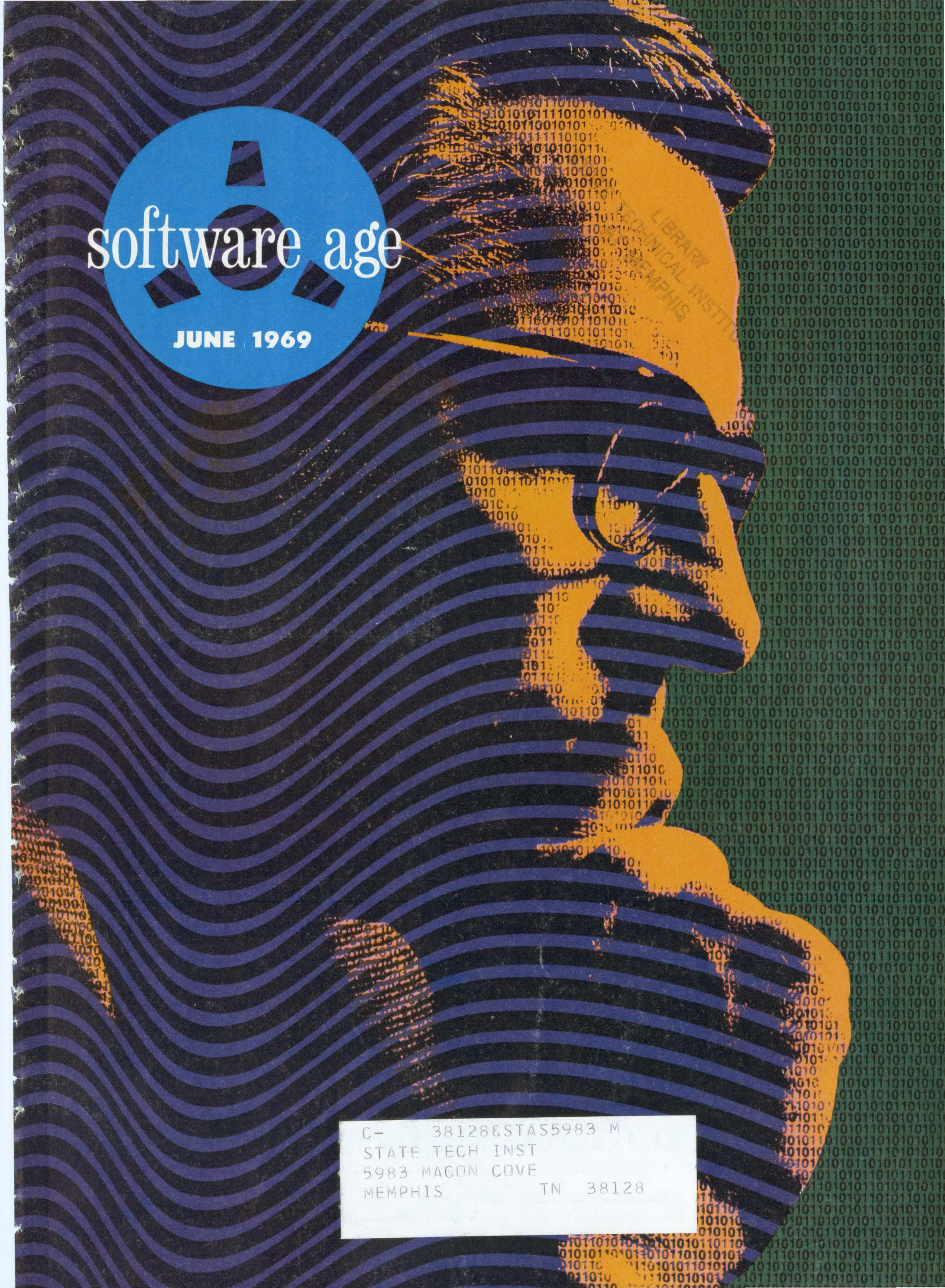
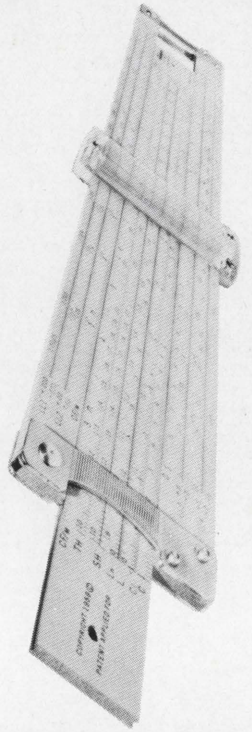

software age
JUNE 1969



software age

JUNE, 1969 **Vol. 3–No. 6**

Publisher .... David W. French
Editor .... John F. Sikora
Managing Editor ...... Jean S. Logan
Editorial Assistant ...... Elspeth Daly
Advertising Manager . . John F. Neiman
Traffic Manager ..... Betsy Pavkovich
Circulation .... Ann Vickers
Business .... Stephenie Jensen
Resume Control .... Anne Wildeman

CIRCULATION OF THIS ISSUE
MORE THAN 128,000

SOFTWARE AGE
2211 FORDEM AVENUE
MADISON, WISCONSIN 53701
PHONE: (608) 249–0128
OR (800) 356–8141

John F. Neiman
New York, N. Y.: (212) 697–5356
Los Angeles, Calif.: Richard Faust,
9800 S. Sepulveda Blvd. (213) 776–0100
San Francisco, Calif.: Ernesto Montano,
595 Buckingham Way, Suite 226 (415) 731–8240
Philadelphia, Pa.: Richard D. Clemmer,
27 Acoma Lane, Collegeville, Pa. (215) 489–9141
Boston, Mass.: (617) 542–1466

SOFTWARE AGE
is published monthly by
press tech inc.
2211 Fordem Avenue,
Madison, Wisconsin 53701

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

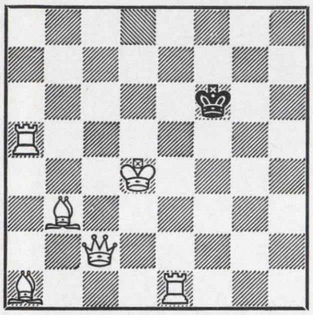
For change of address please return your address label from the magazine.

Controlled circulation postage paid at Madison, Wisconsin.

CHECKMATE

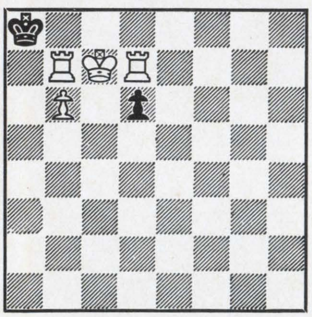
by
GEORGE N. VASSILAKIS
TRW SYSTEMS GROUP

Problem 12
by Tolosa



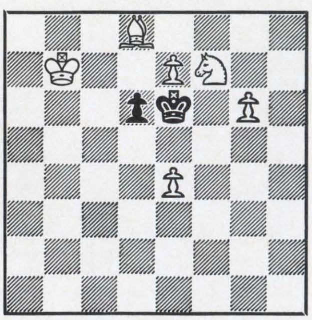
White mates in ½ move!

Problem 13



White mates with Pawn without moving White King.

Problem 14
by Hanneman



White mates in one, two, three or four moves.

THE MINI-MINIGAME

According to Bob Woodworth of Omaha, Nebraska, the following game is the shortest ever played by two masters in an actual tournament.

PARIS (1924)

A. GIBAUD (White)	M. LAZARD (Black)
1. P-Q4	N-KB3
2. N-Q2	P-K4
3. PxP	N-N5
4. P-KR3(?)	N-K6!!
5. Resigns	

THE CHESS COMPANION

"THE CHESS COMPANION" is the title of a new book by Irving Chernev, who has written thirteen other chess books. It sells for $7.50 and is probably the best investment a chess player can make.

The first 147 pages of this book are devoted to short stories about chess and its players. The rest of the book is devoted to games ranging from 1760 to 1960. The problems on this page are a small sample of the kind of problems you will find in this book.

Did you ever hear of mate in ½ move? This is what problem 12 says. Problem 13 is an easy mate in one, but white is asked to mate with the pawn and never move his king. Now, try to get the rook out of the way! In problem 14 you have a choice of mate in one if pawn becomes a queen, or mate in two, three or four, if you choose to play without a queen.

Solution to problem 9:

If White moves across the board to capture the Pawn, Black gets a draw: 1 K-K7, K-B6; 2 K-Q7, K-Q5; 3 K-B7, K-B4; 4 K-N7, K-Q3; 5 KxP, K-B2; 6 K-R1, K-B1 draw.

White wins if he comes down and keeps Black out of the Q5 square: 1 K-K6, K-B6; 2 K-Q5!, K-N5; 3 K-B6, K-B5; 4 K-N7, K-N4; 5 KxP, K-B3; 6 K-N8, etc.

Solution to problem 10:

A LONE Bishop can mate—1 K-B7, P-R6; 2 B-R4, P-R7; 3 K-B6, P-R8 (Q); 4 B-N5 mate.

Solution to problem 11:

1 K-K7!

White must stay on the King file until it forces Black to leave the King file. Then, White attacks on the side of the Black King.

1 . . . K-K7; 2 K-K6, K-K6; 3 K-K5!, K-K7; 4 K-K4!, K-K8; 5 K-K3!, K-B8; 6 K-B4, K-K7; 7 K-N5, K-Q6; 8 KxP etc.

Programmers/Analysts:

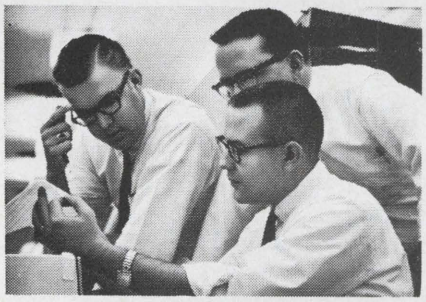
Give IBM a hand and we'll try to make it a free one.

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 design, program, test, and install application programs for our new Manufacturing Information Systems. These systems will aid manufacturing and production planning functions at all IBM facilities.



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

You will be involved with real-time and on-line processing applications, using IBM System/360 programming.

A Bachelor's degree and at least a year's experience are required. You should have a knowledge of basic assembler language and large-scale computer concepts.

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 IBM improve its manufacturing and production planning functions by applying information handling and control techniques.

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 at (607) 755-2855, collect. Or write him at IBM Corporation, Dept. CF1014, 1701 North Street, Endicott, N.Y. 13760.

An Equal Opportunity Employer

IBM®

Controlling Errors

or

"making a remarkable

by Jerome J. Andersen, Donald A. Donaghy, Howard W. Young

In computers—as in anything designed and controlled by human beings—errors are inevitable. Circuits are fallible, programmers make mistakes, and input data can be incorrect. Persons familiar with the limitations of electronic data processing equipment have always recognized this. The parity check bits used in computers stand as testimony to the error.

But, as computers have become faster and applications more complex—particularly in time-sharing situations—more and more attention has necessarily been focused on error detection and correction. In real-time applications, for example, it is impossible by definition to simply rerun a job when errors are suspected. In massive volume jobs, a rerun may mean recommitting hours of expensive computer time. An undetected, uncorrected error in a critical space mission may abort the entire effort. In the modern time-sharing system, scores of users could be inconvenienced by substandard error control.

In the past, once errors were detected in input data, or circuitry (hardware), the whole job was rerun. The optimum approach is to redo (or retry) only the operation which caused the error.

During the early design of the IBM System/360 Time-Sharing System (TSS), a combination of programs was integrated into a support package for the TSS Monitor as a comprehensive approach to the control of failures. Included are programs which automatically record error data, retry failing operations, and assist in isolating faulty equipment. Other programs allow maintenance personnel to perform on-line diagnostic tests and to retrieve information about past failures. The average user is totally unaware of the entire support package unless totally unmanageable errors are encountered. Error Control in TSS/360 is accomplished by this stand-by set of programs.

Jerome J. Andersen is a Staff Engineer at IBM's Systems Development laboratory in Kingston, N. Y., engaged in programming reliability and serviceability for time sharing systems. Previously, he worked for eight years in diagnostic programming.

Donald J. Donaghy is a Project Engineer at IBM's Systems Development laboratory in Kingston, N. Y., specializing in diagnostic programming. He was instrumental in developing the Model 67 time-sharing error control system. Mr. Donaghy earned a Bachelor of Science degree from Drexell Institute of Technology in 1959.

Howard W. Young is a Staff Engineer at IBM's Systems Development laboratory in Kingston, N. Y., specializing in diagnostic programming. He was instrumental in developing the Model 67 time sharing system. Mr. Young attended RCA Institute of Advanced Technology and Columbia University.

Time sharing error control has been geared to two objectives: to minimize or eliminate impact to the operating system from hardware failures; and, to furnish the maintenance engineer with accurate details pertaining to each error incident.

The end result is to provide savings to users and owners because of:

- Higher reliability;
- Better distribution and scheduling of maintenance time;
- Quicker repairs in the event that unscheduled maintenance is necessary;
- Enhanced reputation.

There will be many TSS users in the future who will benefit from the error-control system, unknowingly, as failing operations are successfully retried. Even if an individual user seldom needs the services of this system, the fact that it is there will give him greater confidence.

Error Types—Ever expanding requirements have resulted in an increasing number of relatively large computer systems. At the same time, the larger the system, the more likely that circuit failures will occur.

It is a paradox that the component that fails once in a while is more troublesome than one which having failed, continues to fail until replaced. Such intermittent errors are difficult both to reproduce and to

In Time Sharing

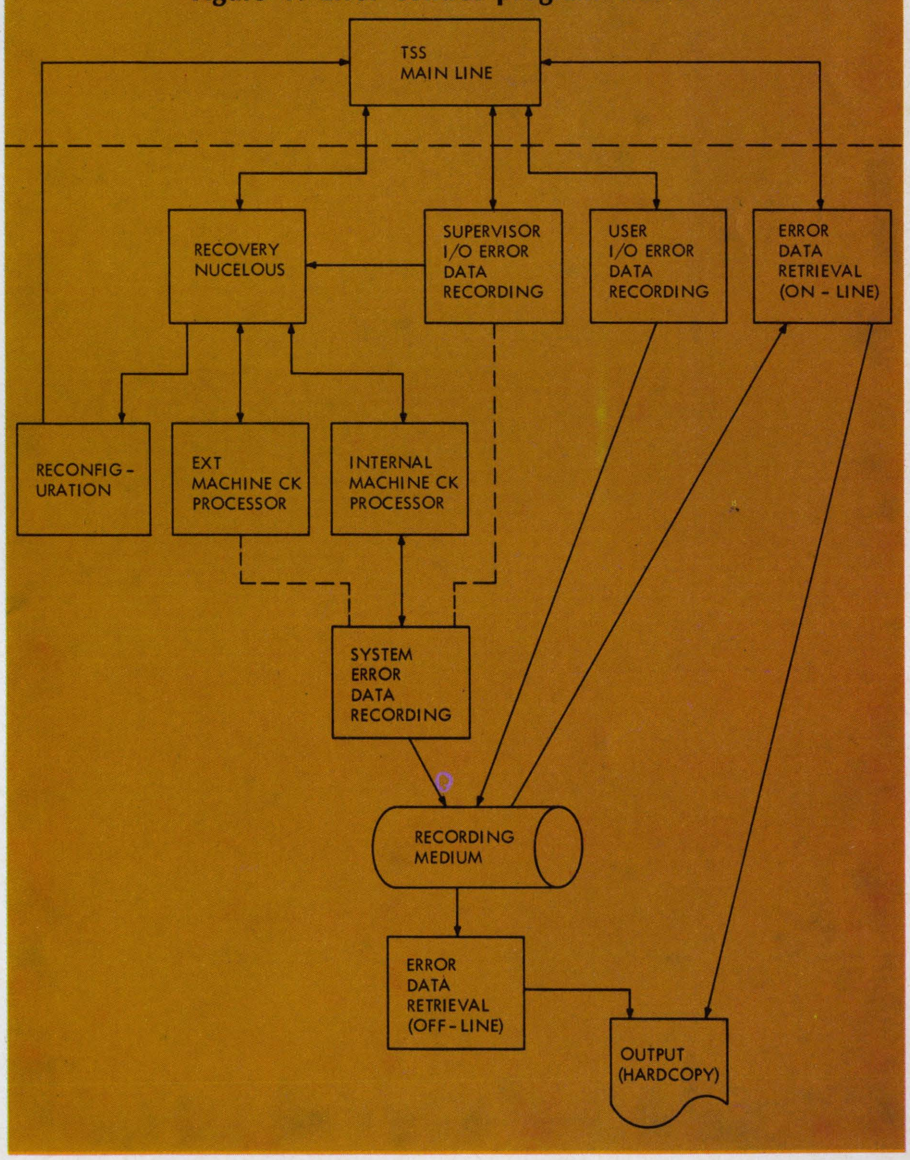
machine more reliable"

find. Their effect on the system is worse than a "solid" failure. With error control programs, however, it is possible to maximize the use of error detection and isolation facilities, minimizing the effect of these intermittent failures.

Errors are classified by the nature of their recovery procedure. The two general classes are those which occur during CPU instruction processing, and those which occur during Input/Output (I/O) sequences. I/O failures can be classified as "soft", for which interruptions and reporting can take place at a convenient time during processing. They are easier to handle than processor errors because a reliable processor is available to diagnose and retry errors. Processor and storage-element failures are classified as "hard"; these cause immediate interrupts and are reported as soon as detected. An interrupt may occur in the middle of an instruction's execution. Recovery from the error may have to be performed by the failing processor itself, requiring sophisticated techniques.

I/O Error Management—A characteristic of The System/360 Model 67 (IBM's Time-Sharing System) which aids error handling is the existence of several paths to a device. These include channel controllers, channels, subchannels, and control units. Multiple paths allow retry over an alternate path, and increase

Figure 1: Error control program elements



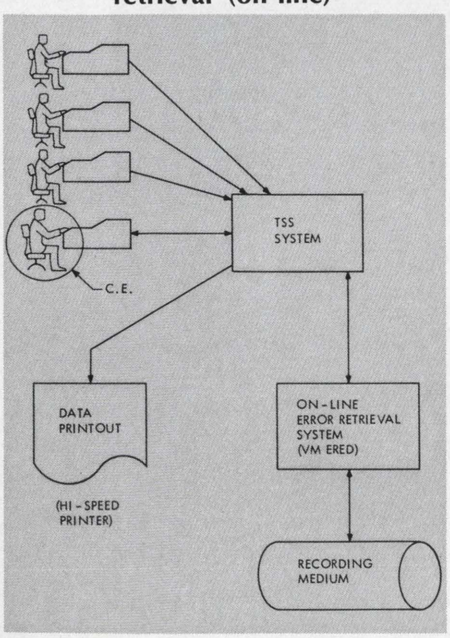
Programmers and systems analysts . . .
Univac:
Where you can help make the world a little bit better
Scientists haven't discovered the exact cause of cancer . . . yet.
But a UNIVAC® computer system now being installed in France, near the world-famous Gustave Roussy Cancer Hospital, will soon be put to work investigating this disease. Which, in turn, could help bring a cure for cancer closer to reality.
Through a combined effort, the Gustave Roussy Institute and Univac programmers and systems analysts are writing the program that will make this application work.
Here at Univac, we understand how important it is for a programmer to work on an important problem like cancer.
So when you join us, we don't put you to work on meaningless, open-ended programs. We put you to work on meaningful, well-defined programs. And you don't just mark time. You make significant contributions. And you're not rewarded just because of your years of service. You're rewarded because of the excellence of your service.
If all this sounds like what you've been looking for in a company, take a hard look at the Univac job opportunities listed on the following page.
SPERRY RAND
UNIVAC
An Equal Opportunity Employer M/F

the likelihood of successful retry. When a specific path fails solidly, it is logically deleted from the system. If all paths fail, the device is presumed unusable, and is removed. If either a device or a path is deleted, it cannot be used again until repaired and restored to the system.

When an I/O operation fails, the control unit retains the conditions surrounding the error and the status of the data processing operation at the time of the failure. After the error is reported to the CPU (via an I/O interrupt), this data can be retrieved by a special I/O (Sense) command to the device. The sense data identifies the exact error type, establishing the procedure necessary to recover from the error. The program then repositions the device, if necessary, and retries the operation The retry will take place over the original path, and the recovery procedure continued until either the operation is successful or a threshold value for that device has been exceeded. If eventually successful, the error is termed intermittent.

As each attempt at recovery is made, error statistics for the device are updated. Counts are kept of the number of times that each type of failure occurs. When a failure count exceeds a predetermined number, an I/O failure incident report is recorded, and the operator notified. Each device has its own unique threshold values, determined by device engineers and product performance analysts. Uniform recovery techniques can therefore be employed for all devices.

Figure 2: Error incident data retrieval (on-line)



If the error threshold is exceeded while retrying, the retry operation is then performed over an alternate path. If successful, operation continues. If all paths fail, the user is notified and his job is terminated. If practical, he can try again with another device. In all cases, pertinent information is recorded.

On channel failures a similar approach is followed. The error-detection circuitry groups failures into three types: data checks, control checks, and interface checks. Channel data checks are retried; control and interface checks are not. All three failure types result in incident recording.

Control and interface checks cause a hardware "channel log-out" with pertinent status signals and channel registers automatically stored in a predefined storage area. This information and the I/O commands in execution at the time of the failure are recorded. If a failure is solid, the channel is logically separated from the system and operation continued unless there are no alternate paths to critical devices served by the channel. In this case, manual intervention is required for physical reconfiguration and repair.

CPU Errors—The retrying of CPU instructions is a combined program–hardware operation using a number of hardware features. These include:

- Error-detection circuitry in the key areas of the CPU and storage elements;
- An immediate interrupt after an error is detected;
- Automatic transfer (log-out) of hardware status into core storage when an error is detected;
- Error isolation indicators available for program inspection;
- Malfunction-Alert interrupt to the other processor after a detected error in a multiprocessing system.

The sequence of events following a CPU error is a malfunction alert, a log-cut, a processor reset, and a CPU-instruction-sequence interruption. The interrupt transfers control to the "machine-check" error-control program, which must determine if one or two processors are in the system. If there is a single CPU, the recovery procedure is immediately initiated. If two, the processor is put into a state of "limbo", capable of responding only to a special signal from the other processor. The second processor will have received the malfunction alert and will itself perform the recovery action.

The error-handling program must first preserve the original machine environment, since the machine-check interrupt system will be used during the failure analysis, destroying the original failure data. The first step in the analysis is to run CPU and storage element checkout programs to determine if an obvious solid failure exists. These programs resemble conventional CPU and memory diagnostics and are designed to provide a reasonably thorough test as quickly as possible. A solid error detected by a checkout program is considered a sufficient reason to rule out an instruction-retry attempt.

The checkout programs are exercised, using all combinations of CPU and core storage, to aid in the isolation of units which have solid failures. Reports are compiled for later evaluation.

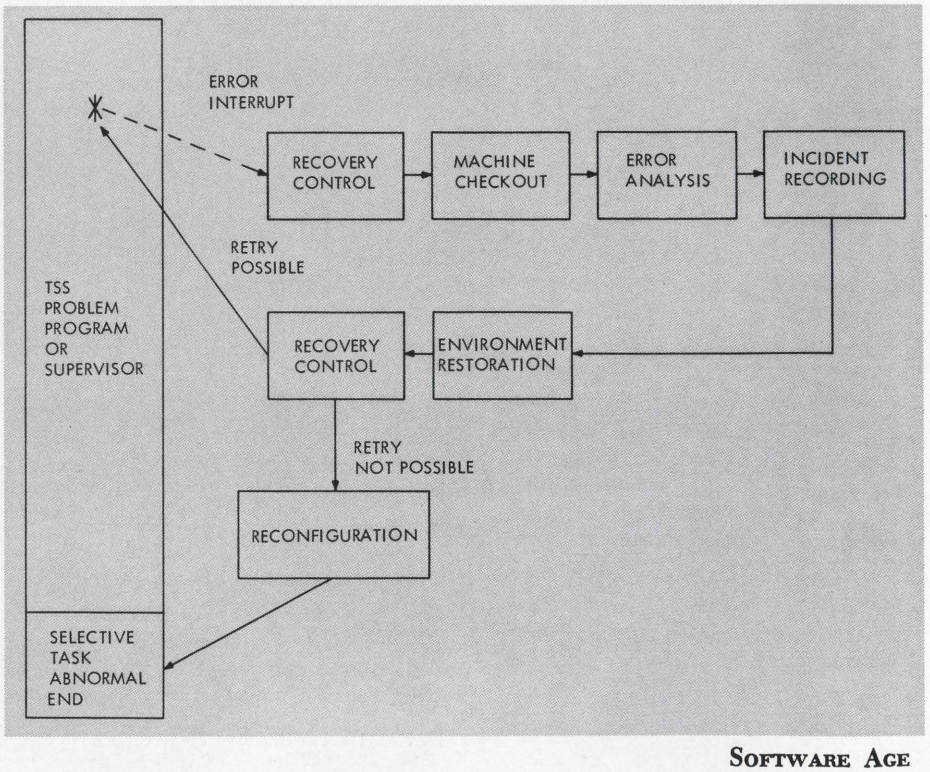
After use of checkout programs, retry-analysis programs are run which locate the instruction being processed at the time of the failure and determine if a retry threshold has been passed. The original contents of a general-purpose register,* for example, may have been modified so the instruction cannot be directly re-executed. In practice, however, the original contents of the register may still be available from the log-out, or there may be a way to reconstruct them. Thus, many instructions with an apparent threshold turn out to have no threshold at all, because original operands can be restored throughout the execution of the instruction.

Once the address of the instruction to be retried is found and it has been determined that no retry thresholds have been passed, the retry programs merely record the failure, restore the environment to its condition just prior to the error, and restart the program that was in progress at the time of the error. The restart is initiated at the instruction that was interrupted.

Reconfiguration—"Retry", while always desirable, is not always possible. The alternatives are selective termination or startover. Selective termination is the process whereby the task in operation at the time of failure is aborted. All other users are unaware that a malfunction has occurred. The affected user is notified that his task was terminated and he can then reinitiate his job. Although one user was impacted, the system continues to operate.

* IBM System/360 general purpose registers are programmable registers used for indexing or computational purposes.

Figure 3: Processor/storage error recovery flow



Some failures that cannot be retried are of such a nature as to preclude even selective termination. In these cases only startover is possible. All tasks are terminated and system operation must be completely reinitiated. At this time, however, it is possible to physically partition failing units from the system.

These units are then available for off-line checkout and repair.

In either case, the objective has been to make the best of an unpleasant situation. This concept of "graceful degradation" allows the users to obtain maximum utilization of the system, even though all units are not available.

System Error Checkout—The error retry and recording approach has been extended beyond hardware-detected failures to include three major categories of software-detected errors: "minor", "major", and "hardware". A "minor" software error is indicated when inconsistent or missing data is detected by a program but the operation can be bypassed by causing at the most one task to be restarted. A "major" error indicates extensive damage to data and recovery is not possible. An automatic system startover results.

The third type, "hardware", is unique in that the supervisor module which invokes it, suspects that an undetected hardware error has occurred. Whenever a program detects this condition, a call is made to a special system-error routine which can utilize the checkout modules in the error recovery package. These CPU and storage-element checkout programs already exist in the recovery program. Thus, an additional capability has been provided to enable the system-error routine to make a cursory check of the CPU and storage elements. This capability provides an on-line test of the CPU and storage.

If an error is detected, a system restart is initiated after the error data is recorded. If no errors are found, the resulting action is determined by the system-error program which chooses a suitable recovery procedure.

Error Incident Recording—Effective error control does more than reduce system down time; it also improves system maintenance. Efficient maintenance can only be accomplished if sufficient information is

available, i.e., the error history of the system. Too often, maintenance is based on second-party information, often inaccurate and usually lacking in detail. In the IBM time-sharing system, an attempt has been made to put hardware maintenance on a par with program maintenance through error incident recording: a concise picture of hardware status preserved *at the time of error.*

Two forms of error recording are used: a preservation recording of the detailed information which can be retrieved later for maintenance, and an immediate summary report at the operator's console. The summary report is given when a significant error which may change the system configuration occurs, or to warn the operator of possible degradation of required system components. For example, all processor errors are considered to be in this category.

Examples of the information contained in a record of a processor error are:

- CPU hardware "logout" data, which includes the status of control latches, contents of internal data registers, and core storage information;
- Operational data collected by the recovery program, including the programmable registers used, mode of operation, time of day, etc.
- Analysis data, which identifies the instruction in progress at the time of failure and its storage location, and operand addresses and contents. Details on the execution of the instruction relative to the failure are also given. If the operation is not retrievable, the reasons are given, and multiple error conditions are assessed for the original cause. CPU-storage checkout test results are also posted.

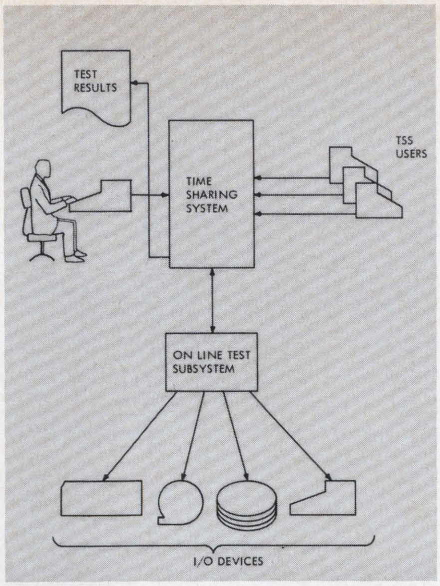


Figure 4: On-line test system

The above information can be provided in approximately 720 bytes. This data, along with other error incident reports such as on I/O, is systematically catalogued in chronological order.

The Error Recording Medium—The medium used for recording errors is of the direct-access type for high-speed. The parallel drum[1] was chosen because it can provide considerable usable storage without cost to the customer. For operational reasons the drum is formatted for 4096-byte records (pages), with a gap of 246 bytes between records. These gaps are used to record error incidents. Approximately 200 error incident records can be recorded per drum without danger of data overflow.

The error recording functions overshadow in importance even the recovery of the CPU, since they make possible the location and correction of errors. When a solid failure occurs, it is possible to detect this and cause the defective element to be partitioned. The element can be repaired and returned to the system.

The intermittent error that cannot be retried because of the instruction retry threshold being exceeded is a more serious problem. This type of error, resulting in startover after startover, can burden a system. The capability to collect and preserve large amounts of error data makes it possible to compile error statistics (statistical element counts). On each machine check call, a count pertaining to the CPU and storage element involved is incremented (available upon demand). Maintenance personnel will suspect a particular element if the count for that element appears extremely high with respect to the other elements. Statistical element counts augment checking facilities in the hardware; they are also valuable because detection of an error in a particular unit does not necessarily incriminate that unit.

[1] IBM 2301.

Retrieval of Error Incident Data—Two versions of a retrieval program are available, with which maintenance personnel can format and print the incident reports from the drum. An off-line version is available for use during maintenance or other periods when the system is inoperative. An on-line version allows data to be retrieved without an impact on other users. Incidents can be recalled by category, such as storage element failures, CPU failures, I/O failures, etc. Specific failure data can be retrieved or the entire error history printed. Whenever possible, the on-line program is used, enabling data to be obtained on the day's incidents before the maintenance period starts. Maintenance personnel can then examine the printout and determine the problems that have occurred and will need attention during the coming scheduled maintenance period. This systematic collection of error data also allows reliability engineers to note failure trends and initiate corrective action.

On-Line Testing—TSS/360 is a system that may contain a large number of I/O devices, such as terminals, disk drives, tape drives and unit record equipment. Repair, adjustment, and verification of failure and repair should, when possible, be done while the system is in use. If all repair and adjustments were to be suspended until time for scheduled preventive maintenance, there would be an unnecessary loss of system resources, taking hours away from the customer, and requiring additional time for system maintenance. An on-line test facility therefore provides the capability to perform this maintenance concurrently with normal use.

An on-line-test control program provides I/O test programs with the interface needed to communicate with the system. The control program also furnishes assurance that system data will not be inadvertently destroyed.

A maintenance man "logs" onto the system as a user, but with a unique class and privilege. He requests the On-line Test System by simply typing "RUN SYSTEST". The control program is activated by the system and proceeds to prompt the Customer Engineer on the testing procedure and the format for entering data, as well as requesting the device and tests he wishes to run. As the Customer Engineer gains experience, he can enter commands directly, without prompting.

The control program then asks the operator to confirm that the device will be used for testing. Following his confirmation, the tests will be run in the desired sequence. Options are available for looping on error and suppressing error messages. More than one device can be tested at a time and more than one Customer Engineer can use the On-line Test System at one time.

Normal I/O access programs automatically retry failing operations, rather than pass failure data back to the user. The On-line Test System, on the other hand, contains an access program designed specifically for the diagnostic programs that will use it.

Error information is returned to the test program from analysis and the program can select the specific path to the device that it wishes to test. This prevents failure data from being lost, allows the test program to control the environment of each test, and provides the flexibility needed for an efficient on-line test system.

In most cases the on-line test programs parallel the standard off-line diagnostic programs, thus reducing the "learning time" of maintenance personnel. Error printouts are also standardized; consequently program outputs are identically formatted messages for all devices. The objective has been to provide programs that would be effective, easily learned and easily used.

It is important to realize that a computer error is very much the exception, rather than the rule. Today's transistorized circuits are reliable to an extent that would have been considered impossible but a few years ago. A computer that runs a day, for example, may have executed 100 billion instructions, without error. Emphasis on error control then, in perspective, is an attempt to make a remarkably reliable machine even more reliable. By coordinating the efforts of design engineer, diagnostic engineer, and systems programmer, it is possible to make error control an efficient and transparent addition to the hardware and software package we call a "computer system". ■

better books

by Dennie Van Tassel

YOUR CAREER IN COMPUTERS by George Fischer. New York: Meredith Press. 1968. 194 pages. $4.95.

George Fischer gives a brief introduction to computers in an easy to read manner and tells how they are used in business, government, the armed forces, science, education, and outer space. Twenty-four different jobs are related to computers, but with a tendency to show the more glamorous side while ignoring the monotonous part of the profession. While the book is really not for people already established in the computer field, it is a good book to have around the house to inform people who want to find out something about the computer field as a possible profession.

CONVERSATIONAL COMPUTERS by William Orr. New York: John Wiley & Sons. 1968. 227 pages. $8.95.

Mankind had several hundred years to get accustomed to the automobile. First there was the sled, then the wagon, later on the wagon was hooked up to animals. During the nineteenth century steam power was added and in the present century the internal-combustion engine brought about the development of the personal passenger car.

In contrast mankind has had only 20 years to become accustomed to personal computers and according to William Orp, personal computers will have an effect on all of us that will be at least as profound as the effect of personal passenger cars.

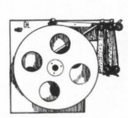
Conversational computers will come as no surprise to the computer profession but it is doubtful if the general public is aware of their potentialities or ready for the cultural shock they are about to experience. This book is an attempt to introduce conversational computers to the intelligent curious non-specialist.

The subjects covered are: Problem-Solving Modes; Instructional Modes; Retrieval and Query Modes; Graphical Conversational Modes; Toward the Computer Utility; and Psychological and Social Implications.

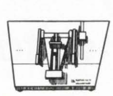
In the past rigid demands of production programming, together with high computer costs and unacceptable time lags, combined to inhibit the free play of imaginative human intelligence when it is tied to a computer system in the conventional manner. Conversational computers, with their advantage of instant response and mass information systems promise to eliminate much of the past restraints and allow man to concentrate on digesting information at his top most intellectual ability. Whether man is ready for this or not, the possibility is at hand. ■

IS PERFORATED TAPE YOUR PAPER TIGER?

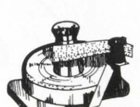
Most perforated tape installations suffer the constant problem of keeping paper tapes organized. Splicing, editing, rewinding, unwinding, repairing, repunching and many other "routine" operations contribute to the growing paper jungle. Relax. Data-link can tame your paper tiger. Now you can efficiently and economically round up loose tapes and keep them in line. Data-link offers Splicer-Gauge-Punch equipment, manual, electric and high-speed winders, center-feed unwinders, complete work stations and an exclusive line of self-adhering patches and correction strips. If your perforated tapes are on the prowl, better write today for the free D-L catalog.



HIGH-SPEED WINDERS



SPLICER-GAUGE-PUNCH



CENTER-FEED UNWINDERS

MANUAL WINDERS

FREE CATALOG!

Describes full line of Data-link Perforated Tape Equipment and Supplies.

DL® **Data-link Corporation**

100 S. Ellsworth Ave.,
San Mateo, California 94402
(415) 347-7001

For more information, circle No. 5 on the Reader Service Card

Can You Bridge the Software Capability Gap?

By
Joseph S. Herbets
Vice President
CONTINENTAL COMPUTER ASSOCIATES, INC.

There is no question as to the integral and vital role of the computer in administration, production, marketing and service in virtually every business and industry today. Each of these areas has been markedly upgraded—with theoretical economy and efficiency—through computerization.

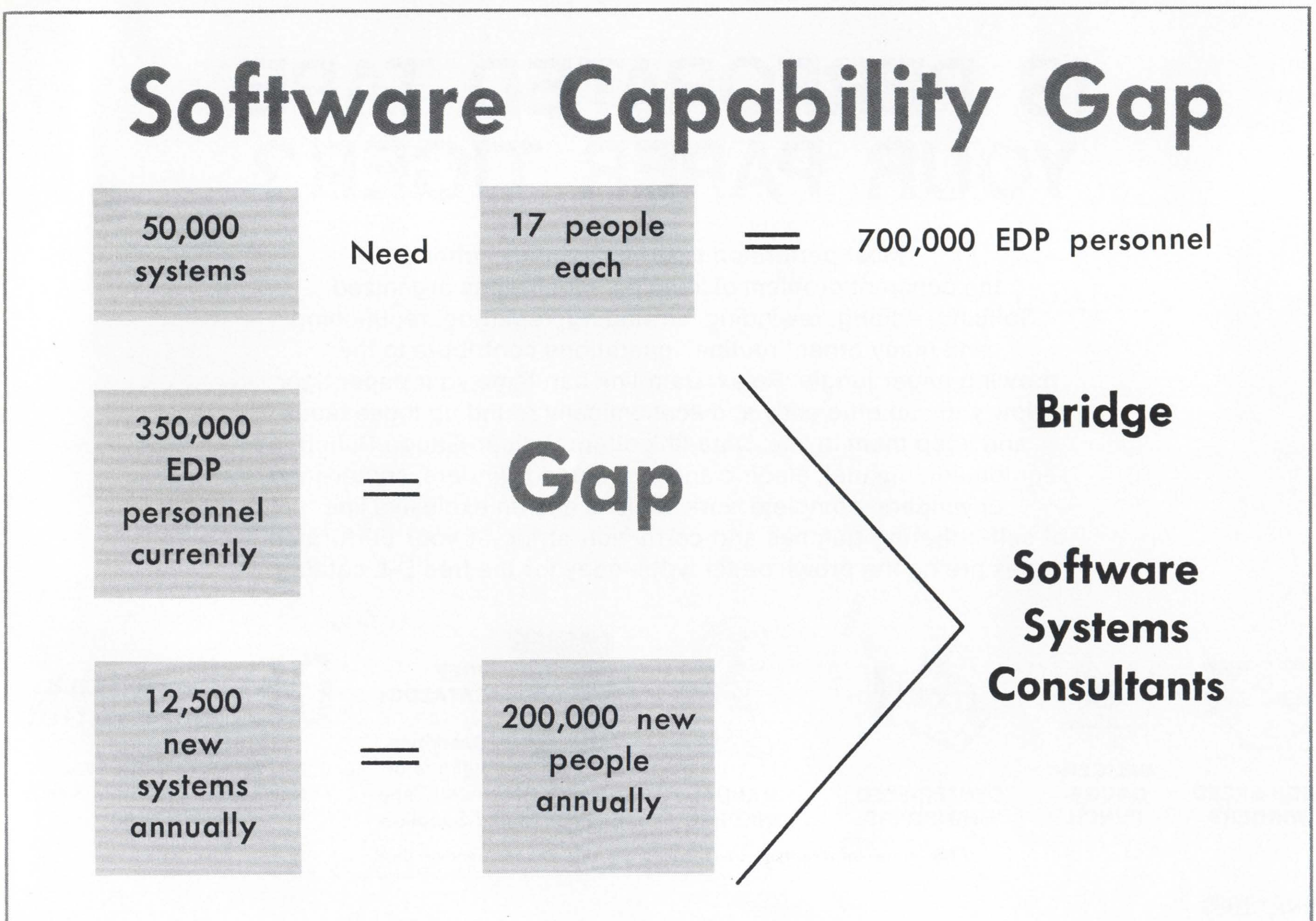
More systems are going on the air daily; the backlog in orders for new systems is mounting; demand for time is mushrooming; the computing power of installed equipment has gone up by an astronomical multiple; services are at a premium; and, most importantly: people are scarce.

Installation of a piece of computer hardware and peripheral equipment is one thing. Turning this inert pile of metal, circuits, and associated gadgetry into a swinging, singing, sagacious source of reports, invoices, statements, and bottom line profits is another. It takes a lot of professionally warm bodies to do that. But where are they? That is the question.

Let's look at the available statistics:

By the latest count, there are conservatively 50,000 computer installations (and estimates range from 50,000 to 60,000) with an aggregate dollar value of about $18 billion. Another $7 billion in unfilled orders is in the offing.

According to one industry source the median number of data processing personnel involved in a given installation is 17. Larger installations involve hundreds! Taking, however for the sake of this discussion, the 17 and multiplying by the total number of systems, we arrive at a figure just short of 700,000 people. This includes everyone from EDP manager to key punch operator.



The best guesstimate to date of the number of people actually employed in EDP falls in the 300,000 to 350,000 range. A far cry from 700,000. Therefore, the average number of available EDP personnel would be more nearly 8 or 9; not 17. Thus, an immediate gap.

Where Does the Gap Lie?

The gap principally involves the higher, more sophisticated EDP personnel. Number one on the hard-to-get list is the systems analyst, with the programmer running a close second.

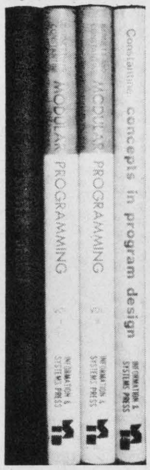
The principal user of these people is the inhouse EDP staff, while a large secondary market for these personnel is one in which their services are utilized by multiple users: computer software houses, service bureaus, hardware manufacturers, and peripheral equipment makers. But even here the demand exceeds the supply. How many more people are needed? Where will they be found? Here, again, the guesstimates run rampant.

But to answer the first question first. It's obvious from the projected figures referred to earlier that there is an immediate need for a dramatic increase in computer personnel just to handle the hardware in place now. If the computer census continues at the current rate of about $5 billion yearly, we can calculate (without the aid of a computer): 50,000 computers worth $18 billion = an average price of $400,000 per system . . . then our current $5 billion annual growth rate is equal to about 12,500 new systems annually. Using whatever average you wish, the answer boils down to one basic fact: mega-people are needed to make these new systems operational; trained people that we do NOT have, but need.

Management Barriers Compound the Gap

Until recently, the capability gap problem was further complicated by a prevalent attitude on the part of senior management to avoid any confrontation with the computer or to accept responsibility for it. The machine was for all practical purposes a useful toy under the care and feeding of the technician—the only person who really understood it. Compounding this attitude, the technicians didn't speak management's language—only machine language. The language alone was sufficient to erect a solid barrier between management and EDP people.

But the barriers are beginning to crack. Management, hopefully, is beginning to assume its proper role in the EDP scheme—making impor-



tant decisions as to what the computer should and should not be doing. Computer professionals, even in this new branch of corporate operations (the first computers went on line about 15 years ago), are finally breaking into first line management and beginning to have their say. After all, the computer is young, a capital investment and, like any other piece of capital equipment, should show a return. But even with recognition by and support of management, the chief stumbling block —*software*—remains, and software includes people.

The fun part of any problem—as anyone in the computer field will agree—is arriving at a solution. Where are these people to be found? Let's look at the major sources.

What Are the Available Resources?

The first and foremost is the existing roster of EDP specialists on the payroll. But they're our base, our foundation; upon this we have to build. Another growing source, EDP schools, is plugging hard and heavy to fill their classrooms, but often leave a lot to be desired in the quality of their graduates. They are, however, a leading supplier in filling this gap, and they're bound to improve their quality.

Moving up from EDP schools, we can look to the nation's colleges and business schools. We see that there is a trend toward degreed EDP personnel. This is especially true in the classifications of Systems Analyst and Programmer. According to a recent study, the percentages of people in the category of Data Processing Manager, with either Systems Analyst and Programmer degree or some college training is 87.3%, 89.3% and 82.4% respectively. The most sought-after college grad is the one with math or science majors ***plus*** EDP training and/or experience.

The last resort—or in many cases the ***first*** resort—of many organizations is to look at the grass on the other side of the fence: the competitor's employees. Pirating the other fellow's help is rampant in many businesses where the demand exceeds the supply. Data processing is in there with the best of them. But it's no solution to the gap, only a reshuffling of the cracks.

Once a likely prospect is hired, training and compensation factors take over. The average annual salary for a beginning ***trainee*** in systems analysis or programming has been quoted at $6,000. With a little experience under his belt, the analyst or programmer can move into the higher income brackets with relative ease. In the more generous markets, incomes for these positions can range from just under $15,000 to over $20,000 annually for the more experienced professional.

Bridging the Capability Gap

We cannot quickly resolve the gap between numbers of men and machines, but we can, at least, temporarily bridge it. How? By means of the computer software service company. The advisory, design, implementation and programming support services rendered to multiple users affords a more economical use of available EDP experts. In essence, it "spreads them around".

To the prospective user, the software organization serves as the Devil's Disciple to management in determining to go or not to go with data processing. And, if affirmative, to what extent.

The existing user, suffering from gap-itis, sees the software services company as a gap filler in the areas of performance and effectiveness analysis, continuing outside advisory assistance, programing and re-programing (2nd to 3rd generation language changeover), systems design, installation and service "policing".

The great advantage the software company can offer the computer user is an instant data processing staff, well trained in the equipment the user has chosen or capable of aiding in the selection of specific hardware, that can get the job done rapidly, and then fade off the users' payroll. When necessary, a software company can provide the complete function of design, development, programing, and documentation, turning over the completed, approved results to the users' operating staff.

The economics of such a situation are obvious. The user has a better feel for the cost of a project, and after design or definition, the software company will probably guarantee the complete job for a fixed fee, including the important documentation. Just as important, it will also guarantee the date of completion. The many computer installations, countrywide, that have spent the last several months, or years, converting from one computer to another, from one programing language to another, or from tape to disk (or vice-versa), have not been able to accurately determine starting dates for new applications, let alone completion dates. Many users have found it rewarding to continue current plans using in-house capability while employing the outside temporary professional to implement new cost saving applications. ■

Partially Predictable from Non-Deterministic

by Dr. Thomas C. Lowe

Introduction—Sequential machines may be partially, rather than completely specified for a number of reasons. When next-state behavior is not completely specified (i.e., "don't care") it is generally assumed that any operations on the machine will terminate when a transition to an unspecified state is encountered. But it is equally possible to make an assumption about the nature of the unspecified transition, and to consider conditions which drive the machine through a sequence of unknown states. When this is done a class of machines evolves which can be sent to unspecified states and then "reset". One practical application of this is in design when behavior after an error is considered.

Machines and Their Descriptions—It is convenient to think of a machine as being a physical device upon which a hypothetical investigator can perform experiments, in the manner of Moore's Gedanken experiments.[1] The machine has some input and some output devices, a finite input alphabet and a finite output alphabet. An experiment occurs in time, conceived as a sequence of distinct instants $t = 0$, $t = 1$, etc. An experiment of length x consists of setting the initial state of the machine[2] (the state at $t = 0$) and applying an input sequence of symbols from the alphabet of the machine, one such symbol at $t = 0$, and one at every instant up to and including $t = x - 1$. The outcome of such an experiment is a sequence of x output symbols, occurring at $t = 0, t = 1, \ldots, t = x - 1$.

A deterministic machine exhibits consistent behavior when experimentally investigated. A non-deterministic machine is similar except that under some circumstances the device may exhibit differing output sequences upon successive applications of the same experiment. The present paper is not concerned with the probabilistic analysis of behavior.

Many machine descriptive schemes have been used. Some schemes are exclusively behavioral in nature, such as predicate calculus formulas, while others are semi-structural, such as state diagrams. Regular-expression-language descriptions seem to lie between the two just mentioned. In this paper modified state diagrams will be used. In these, an arrow (normally representing an edge) not terminating on any vertex (or node) is used to represent an unspecified next state. A similar convention is used to identify unspecified outputs. But what meaning can be assigned to partially specified machine descriptions?

Every semi-structural description containing unspecified entries is sub-

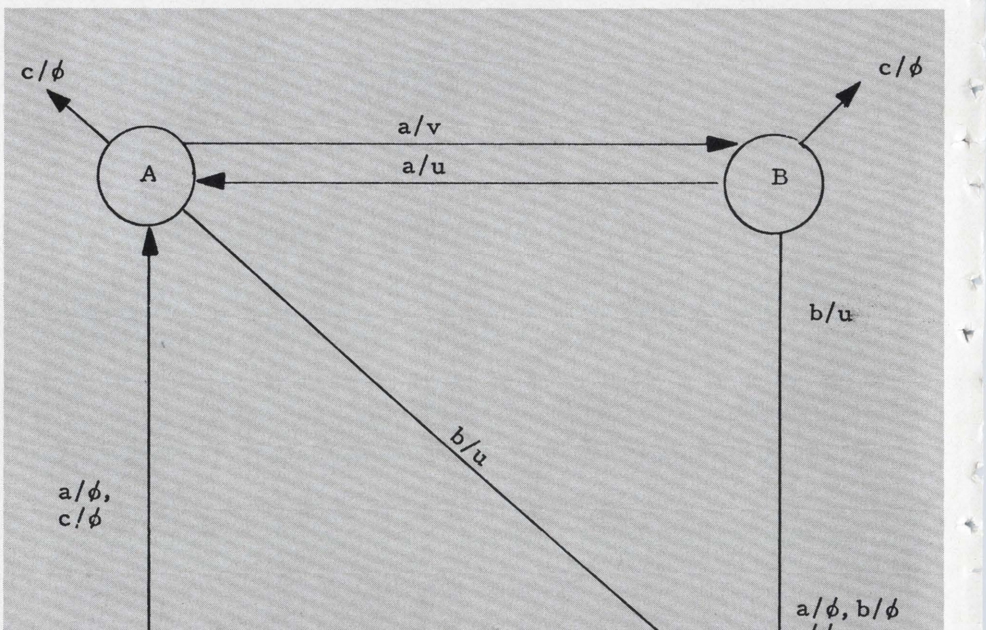


Figure 1. A Non-Deterministic Machine

Results

Machines

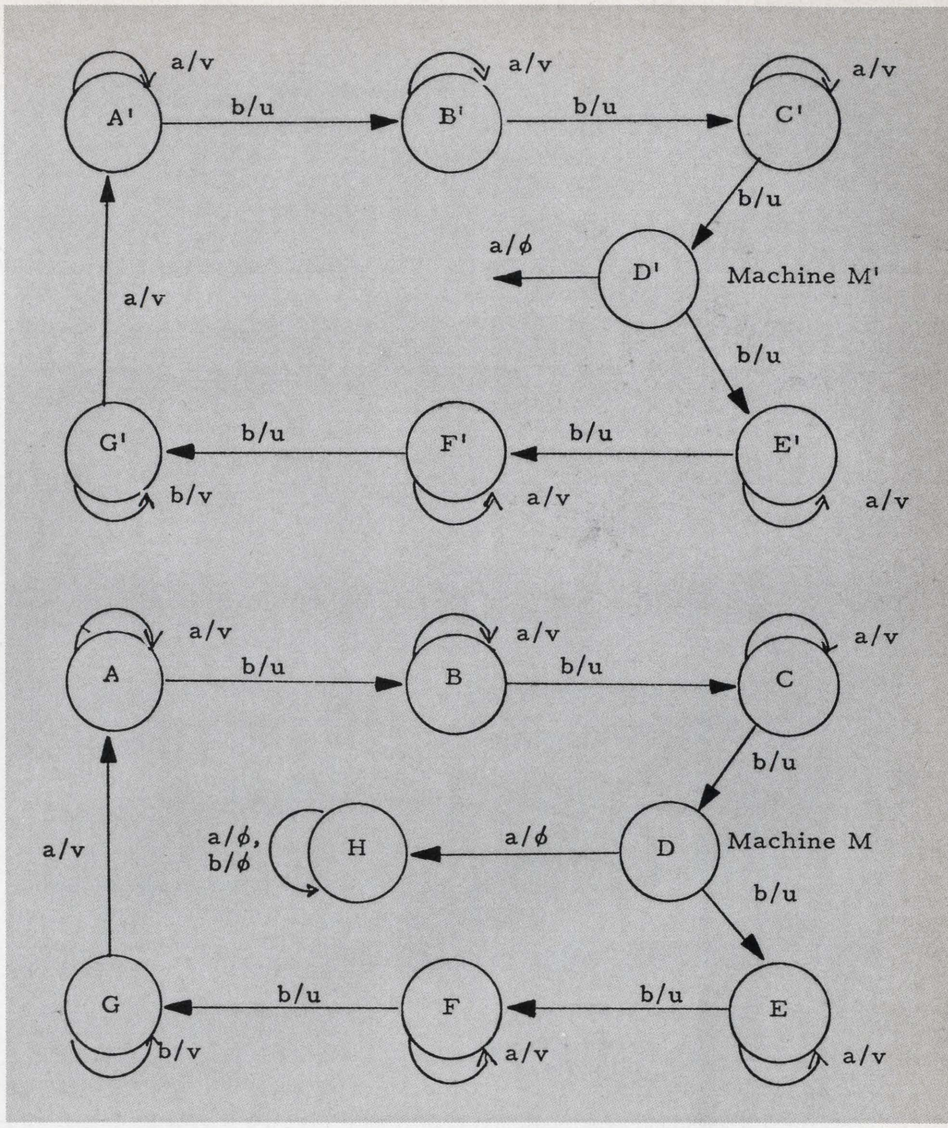


Figure 2. A Machine Exhibiting Partially Predictable Behavior

ject to a variety of interpretations, including an interpretation as a description of a set of deterministic machine, an incomplete description of one deterministic machine, or a description of a non-deterministic machine. Here, a restricted version of the last interpretation will be employed. Since a non-deterministic machine equipped with a clever demon could imitate the behavior of any element of a set of similar deterministic machine, in one sense the results obtained on the former are applicable to the latter.

Deterministic Machines—For use in the present work a conceptually easy method of representing machines is in terms of sets of states, inputs and outputs, and the functions which take inputs and present states into next states and outputs.

A *deterministic machine* is defined as: a set of n states $S = [s_i;\ 1 \leq i \leq n]$; a set of m inputs $I = [I_k;\ 1 \leq k \leq m]$; a set of p outputs $Z = [Z_j;\ 1 \leq j \leq p]$; a next-state function $\Delta\ (S_i, I_k)$; and an output function $\theta\ (S_i, I_k)$. The latter two functions are defined for every combination of S_i and I_k as above, subject to the following restrictions:

1) The next-state function has a defined value for each of its possible argument pairs, and that value is an element of the set S.
2) The output function has a defined value for each of its possible argument pairs, and that value is an element of the set Z.

When the concept of experiments upon the deterministic machine is of importance, the behavior of the machine is thought of as a sequence occurring in a domain of discrete units of time, $t = 0, t = 1, \ldots,$ etc. The experimenter is thought of as some agent who sets the initial state and then *applies* a sequence of inputs, elements of the set I, to the machine, one at time $t = 0$, one at time $t = 1$, etc. At any time the machine *is in* exactly one present state or element of S, *receives* exactly one input from the set I, and *gives* exactly one output from the set Z. This output is defined as the value[3] of the output function $\theta\ (S_i, I_k)$ with arguments that are the present state at time t and the input at time t. The state at time $t = 0$ is set by the experimenter, and is called the *initial state.* For all $t \geq 0$, the state at time $t + 1$ is the value of the next-state function $\Delta\ (S_i, I_k)$ with arguments that are the state at time t and the input at time t.

Non-Deterministic Machines—Before the definition is given, some additional explanation of why such a definition is needed is in order. What would be meant if a machine had for some pair S_i, I_k, no specification for $\Delta\ (S_i, I_k)$ or $\theta\ (S_i, I_k)$? What would the hypothetical experimenter expect if he applied an input to a machine that would take that machine to an unspecified next state or result in an unspecified output? He could either expect that the machine would consistently behave just as if some constant value had been assigned to the unspecified entries, or he might expect that it would be perfectly possible to observe two different behaviors on two different occurrences of the same S_i and I_k. That is, the incomplete specification could either represent some single non-deterministic machine, or a set of deterministic machines. Consider the former.

The work described here was, in part, supported by the Rome Air Development Center under contract AF 30(602)–2382.

A *non-deterministic machine* is defined as: a set of state $S = [S_i; 1 \leqq i \leqq n]$; a set of m inputs $I = [I_k; 1 \leqq k \leqq m]$; a set of p outputs $Z = [Z_j; 1 \leqq j \leqq p]$; a non-specification element ϕ; and an output function $\theta\ (S_i, I_k)$ and a next-state function $\Delta\ (S_i, I_k)$. The latter two are defined for every combination of S_i and I_k above, subject to the following restrictions.

1) The next-state function has a defined value for each of its possible argument pairs, and that value is either an element of the set S or ϕ.
2) The output function has a defined value for each of its possible argument pairs, and that value is either elements of the set Z or ϕ.

When the concept of an experiment upon a non-deterministic machine is of importance, the behavior of the machine is thought of as a sequence occurring in a domain of discrete units of time, $t = 0, t = 1, \ldots$, etc. The experimenter is thought of as some agent who determines the state of the machine at time $t = 0$, (the *initial state*) and applies a sequence of input elements of the set I to the machine, one at time $t = 0$, one at time $t = 1$, etc. At any time t the machine *is in* exactly one present state or element of S, *receives* exactly one input from the set I, and *gives* exactly one output from the set Z. The output is defined as the value of the output function $\theta\ (S_i, I_k)$ with arguments that are the state at time t and the input of time t, if the value of $\theta\ (S_i, I_k)$ is one of the elements of the set Z. If the value of $\theta\ (S_i, I_k)$ is ϕ, then the machine gives as an output any element of the set Z; the machine may, in a repetition of the same conditions on the present state and input at time t, give an output that is either another or the same element of Z. For all $t \geqq 0$, the state at $t + 1$ is the value of the next-state function $\Delta\ (S_i, I_k)$ with arguments that are the state at t and the input at t, if the value of $\Delta\ (S_i, I_k)$ is one of the elements of the set S. If the value of $\Delta\ (S_i, I_k)$ is ϕ, then the machine may, in a repetition of the same conditions on the present state and input and time t, assume a state at time $t + 1$ that is either another or the same element of S as that assumed at $t + 1$. *(This establishes the novel part of the present treatment, restricting the interpretation of ϕ).*

Consequences—Under the definitions presented, it is possible for the outcome of an experiment to consist of predictable and unpredictable subsequences. Consider the result of allowing input sequences which require only that their elements be members of I. For the machines shown in *Figure 2*, an experiment is proposed.

If "?" is used to indicate here an output which cannot be known in advance, then the output sequence of the machine M′, when started in state A and presented with the input sequence bbbabbbbbbbaabbab is uuu???????vvvuuvu. Certainly less can be predicted about the output sequence of machine M, even though in Paull and Unger's terminology[4] M covers M′. The output sequence of machine M is uuu??????????????.

Conclusions—It has been shown that if transitions into unspecified states are restricted in a certain way, so that the unspecified states are elements of the specified set of machine states, the sane automata may exhibit behavior composed of both totally predictable and unpredictable components. The applications of this concept are various. It is possible to define a class of automata as "resetable", in that some sequences exist which (when appended to other specific sequences) will cause the automaton to return to a deterministic mode after leaving it. ■

[1] Moore, E. F., "Gedanken–Experiments on Sequential Machines," *Automata Studies,* Ed. C. E. Shannon and J. McCarthy, No. 34, *Annals of Mathematics Studies,* pp. 129–253. Princeton University Press, 1956.

[2] It is not the case in this work that the initial state is included in the machine specifications. However, there is no reason why such specification, if convenient, cannot be used. The use of such a model requires only trivial modification of the present form.

[3] This wording, which may appear unnecessarily clumsy here, is required for reasons that will become apparent in the next section.

[4] Paull, M. C. and Unger, S. H., "Minimizing the Number of States in Incompletely Specified Switching Functions," *IRE Trans. on Electronic Computers,* V. EC-8, No. 3 pp. 356–367, Sept., 1959.

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
P. O. Box 2076
2211 Fordem Ave., Madison, Wis. 53701

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

Submitted by J. S. Blanchard

```
   DØ 10 I = 2,3
10 IF(I.EQ.4) GØ TØ 20
20 DØ 30 I = 1
30 WRITE(6,90) I
90 FØRMAT(4H1I = ,I4)
   STØP
   END
```

Will it compile? Will it execute? What values of I will be printed?

ANSWER TO LAST MONTH'S PROBLEM

The problem was to guess what values of I would be printed if the following program would be executed?

```
    DØ 100 I = 1,20
    IF (I.NE.1.ØR.I.NE.3.ØR.I.NE.5) GØ TØ 100
    WRITE(6,101) I
101 FORMAT(5H I = , I1)
100 CONTINUE
    WRITE(6,101) I
    STØP
    END
```

The first impression of the unwary programmer is that values 1, 3 and 5 will be printed. Actually, the first WRITE statement is never executed because the IF statement is *always* TRUE, for any value of I. The three logical expressions are connected by the logical operator .ØR. which means that if any one of the logical expressions is true, the whole expression is true. The second WRITE statement will be executed and, even though I = 20 at that point, a zero will be printed because of the I1 format.

April's Problem Revisited

This was the problem with the statement GØ TØ (101,102,101,102,101),K where K= —5 at the time this statement was executed. Congratulations to those of you who guessed correctly that there was a typographical error and the second appearance of statement number 101 should have been 102. Incidentally, the GE-635 got into a loop because the FORTRAN compiler generates coding to test for K = 0 and K greater than 5 but it does not check for negative values of K.

Correction:

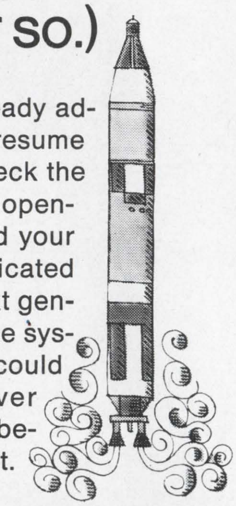
In explaining the IBM/360 compiler, I had erroneously stated that: "the manual clearly states that the compiler assumes K = 1 when K is outside its range." Unfortunately, I had misplaced the manual at the time I was writing this explanation. Several days later I found the manual and realized I was wrong but my material was already in the hands of the printers.

In order to set the record straight, let us quote the manual once more, "If the value of i is outside the allowable range, the next statement is executed."

See page 31 for the Trouble Tran winners.

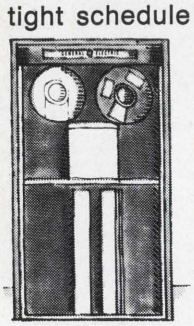
Join us in time for the next generation of space systems software.

(You'll only have to wait a month or so.)



Things happen that fast here. Because we're engaged in a broad spectrum of programs, from small spacecraft to major space vehicle programs. And because, Galileo, Copernicus and Einstein to the contrary, we're at the center of the universe (conceptually at least) when it comes to space exploration.

With today's (and tomorrow's) tight schedule of probes, everything from microcircuits to Mars; from meteorology to manufacturing; from Monte-Carlo theory to matrices; from mission simulation to man-hours gets programmed, analyzed, synthesized, projected and validated on our computers.



It isn't easy. It isn't ordinary. But it is exciting.

We could talk all day about specifics. About tasks ranging from component simulation and Q/C analysis to analysis of complex trajectories and multiple interacting variables. About how you can tackle anything from black-box design evaluation to total spacecraft design optimization. And top off the discussion with an hour-long list of projects and disciplines.

But we won't. If you're creatively and temperamentally ready for the kind of software we create at GE, you're already addressing your resume anyway. So check the list of current openings, then send your resume as indicated below. The next generation of space systems software could be peering over your shoulder before you know it.

Openings for:

Systems Programmers & Analysts □ Test Programmers □ Information Systems Engineers □ Senior Programmers □ System Test & Validation Engineers: Software/Hardware □ Data Processing Engineers □ Display Engineers □ Information Systems Analysts □ Command & Control Analysis Engineers □ Software Systems Engineers

For immediate consideration, please submit resume including salary history, to: Mr. Ronald W. Stutzman, Sect. 154F, General Electric Co., Space Systems, P.O. Box 8555, Philadelphia, Pa. 19101.

GENERAL ELECTRIC

An equal opportunity employer (m/f)

new products

Robins Data Devices, Inc., a subsidiary of Robins Industries Corp., has announced the recent release of new **high-speed perforated-tape winders** and fanfold. According to the data devices sales manager, the motorized winders have take-up speeds of up to 24,000 codes a minute, to mesh with today's fastest tape readers.

Robins' fanfolded tape is available in oiled and unoiled paper, laminated paper, mylar paper, and vulcanized fiber and black photo electric varieties. Fanfold tape is said to be ideal for short run application, with handling efficiency and ease in filing and mailing.

For more information, circle No. 9 on the Reader Service Card.

* * *

Canon U.S.A., Inc. has introduced a new **portable microfilm duplicator** that reproduces 16 mm or 35 mm microfilm on Kalvar roll film in one continuous operation.

The desk-top Canon Roll Duplicator 500 requires no darkroom, chemicals, water or gas and it is capable of copying at speeds of up to 30 feet per minute. From a master roll of microfilm, the "500" is capable of making reversal duplicates (positive-to negative and negative-to-positive)—on Kalvar roll films.

During the operation of the unit, a master film and Kalvar film are threaded through the film guide with their emulsion sides in contact as they pass through the exposure unit. The apertures and tracking speed are set according to the density of the master and type of Kalvar film used. The aperture slit can be adjusted to 15 positions, and the tracking speed can increase up to 30 feet per minute. The film is developed as it passes around a heated drum, and subsequently passes through a fixing unit to set the image. These operations can be done with normal artificial illumination since the high-resolution Kalvar film is sensitive only to ultraviolet light.

For more information, circle No. 10 on the Reader Service Card.

* * *

Golf entered the computer age today as a new device for teaching and practice was unveiled in Fort Worth, Texas, by the golf pro Billy Casper.

The new invention, which is called the Billy Casper Golf Computer, enables a golfer to measure his swing scientifically—for both distance and accuracy—without the use of a golf ball or elaborate trapping net.

The golfer stands atop a 6 by 3-foot platform which is similar to a driving range mat with its grasslike covering of "Astro-Turf". When he hits a golf ball-like object on a pivot the movement generated by the swing is converted electronically into measurements that appear immediately on a control panel. The "distance" of the shot is measured to 350 yards on a dial, while red, green and white lights on directional arrows reflect the amount of push, pull, hook or slice on the simulated drive. Both the distance and the accuracy of the swing are determined by the analog computer in the platform that utilizes a complicated maze of transistors, sensors and other electronic equipment.

The analog computer is designed to operate at both high and low intensities. When programmed for low intensity (for beginners or inexperienced golfers) it will

not be as precise in its measurement of hooks and slices. However, when it is operating at a high intensity it is able to give an excellent picture of the distance and direction of the "shot".

For more information, circle No. 11 on the Reader Service Card.

* * *

Kybe Corporation has announced an **automatic disk pack cleaner,** the DP-10, compatible with all 1316 type disk packs. The DP-10 is said to be as easy to use as a home washing machine.

The DP-10 reduces computer operating costs and decreases the risk of disk drive head damage by performing its cleaning operation off-line.

The heart of the system is a pair of electromechanically operated wiping posts inside the cleaning chamber (which is pressurized with filtered air). The posts are fitted with lint free pads which are impregnated with iso-propyl alcohol. After the cleaning operation, five minutes or

less, the residual cleaning fluid on the disk pack surfaces is evaporated by filtered ambient air.

For more information, circle No. 12 on the Reader Service Card.

* * *

EMR-Computer has announced a 500 nanosecond, 16-bit word central processor, **the ADVANCE 6135.**

The EMR 6135, designed for real-time data acquisition and data reduction applications, is capable of an input/output burst rate in excess of six million words per second in a 32,768-word configuration. It is well-suited for seismic, biomedical, process control, telemetry, and automatic test and other applications requiring high throughput.

In addition to the fast memory cycle time, the medium-scale EMR 6135 system also incorporates total monolithic integrated circuitry, asynchronous bus system of logic organization, three hardware index registers, hardware multiply and divide, and double-precision integer arithmetic.

The EMR 6135 offers a versatile software system of extremely flexible programming packages and a wide variety of peripheral equipments and options for standard and special applications.

For more information, circle No. 13 on the Reader Service Card.

* * *

Honeywell's Computer Control Division in Framingham, Mass., has introduced a **real-time control computer system** for data acquisition, monitoring, supervisory control, automatic testing and production control uses.

The new H1603 control system is made up of the division's H316 mini-computer and a new integrated-circuit unit called the real-time interface. Included in the system are a basic operating package, Op-16, and more than 500 field-proven programs.

While the H1603 costs as little as half the price of previous real-time control systems, according to John W. Hoag, division product manager for industrial control systems, the low-cost system offers users the option of breaking their control problems into several parts, each controlled by its own computer.

For more information, circle No. 14 on the Reader Service Card.

* * *

A software package, WAMPUM, has been developed to solve simultaneous algebraic systems of equations in a new way. In addition to being faster than other methods now available, the new method overcomes the major difficulties that arise in computing solutions to such systems automatically.

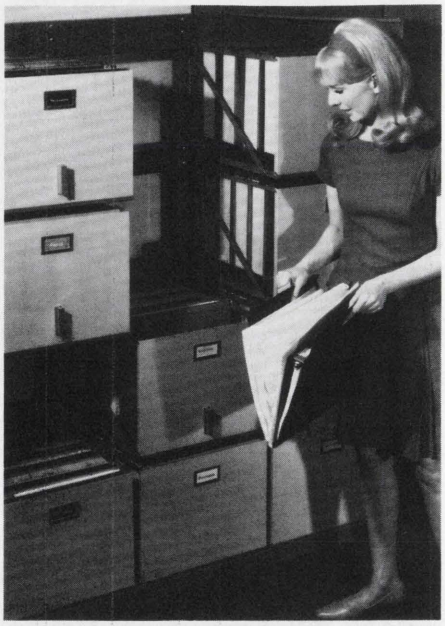
WAMPUM depends on a parameter that makes solutions arbitrarily insensitive to the inaccuracies or meaningless perturbations in a system. In addition, the method is not prone to inaccuracies that arise due to loss of calculating significance, or to round off errors. It handles singular systems in a meaningful way that is easy to interpret, and it may be used in work involving more unknowns than equations, as well as non-singular systems.

The new method is available for use on any digital computer, and should be particularly valuable to users working with large matrics for business or scientific problems, and with numerical techniques for solving partial differential equations.

For more information, circle No. 15 on the Reader Service Card.

The Wilson Jones Company has announced the newest addition to its line of **data processing referral and retrieval equipment,** the "Data-Center".

The "Data-Center" was designed to hold all manner of EDP printout and input-output material in binders, slings, or card trays. Each "Data-Center" has 13 compartments for suspending over 100 flexible Nylon Post Binders, more than 50,000 "tab" sheets, in 12 square feet.



"Data-Racks" are also available for transporting the records to operating offices or departments, minimizing wear and tear to the binders and marginal-punched sheets.

For more information, circle No. 16 on the Reader Service Card.

S/A market place

where you can reach 128,000 programmers, mathematicians, analysts and EDP managers

RATES

Classified—75¢ per word, minimum order $7.50. Blind ads are an additional $10.00. Payment must accompany copy.

Display Classified—One inch by one column, $75.00. Column width is 2¼". Help wanted and employment services ads will be accepted in the classified section. Ads of ⅛ page or more will be listed in the Advertisers' Index, and will qualify for the free resume service.

Closing Date—the 10th of the month preceding (i.e. May issue closes April 10th). Send order and remittance to:

software age
Classified Department
P. O. Box 2076, Madison, Wisconsin 53701

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-601, Software Age, P. O. Box 2076, 2211 Fordem Ave., Madison, Wis. 53701.

WASHINGTON, D. C. & OVERSEAS
PROGRAMMERS, two or more years' experience in COBOL for IBM 360/30, 40 and 50. **SYSTEMS ANALYSTS**, knowledge of FFS or other data management systems. **PROGRAMMER**, IBM 360/30, with heavy management and supervisory experience. Salaries competitive based on education & experience. For application write James J. Redfern, Professional Talent Search A/PM, Agency for International Development, Washington, D.C. 20523. U. S. Gov't is an Equal Opportunity Employer.

EMPLOYMENT AND SEARCH COMPANIES

PROGRAMMERS 360 COBOL
Jrs. $8-10M, Srs. $16-22M, Intermediates $10-15M
Contact R. Lowy who has had 10 yrs. exp. in programming & systems. We specialize in data processing placements. Resume not necessary. Co. pays entire agency fee plus travel expenses.
ERIN EXECUTIVE AGENCY
132 W. 34 St., New York, N. Y. 10001
Phone: (212) 594-8296

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

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

WANTED: Query from a small software company that wishes to become a large software company by marketing copyrighted, fully documented, COBOL program for disk file maintenance and data retrieval. . . . W. Campbell, P. O. Box 688—Edgemont Br., Golden, Colorado 80401

TROUBLE-TRAN WINNERS

$25.00 for submitting March problem: C. A. Argila, TRW Systems, Space Park Drive, Houston, Texas 77058

$25.00 first prize for March: H. T. Gladwin, Box 8855—Skyway Branch, Seattle, Washington 98178

$15.00 second prize for March: C. R. Tupper, Jr., 10810 Triola, Houston, Texas 77072

$25.00 for submitting April problem: L. R. Zar, 1000 Bob-o-Link Rd., Highland Park, Illinois 60035

$25.00 first prize for April: A. E. Hoffman, 2119 N. Wanamaker St., Philadelphia, Penn. 19131

$15.00 second prize for April: J. L. Pearlman, 28 Daniel Drive, Burlington, Mass. 01803

CLASSIFIED ADVERTISING ORDER FORM

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 = $_________

☐ Blind Ad $10.00 Insert_________time(s) Total Enclosed $_________

NAME_________________________

ADDRESS_________________________

CITY_____________ STATE_____________ ZIP_________

Signature_________________________

WORD COUNT: Includes name and address (unless blind ad). The name of the city (Des Moines), state (New York) and zip code are counted as a total of two (2) words. Each abbreviation, initial, single figure or group of figures or letters count as a word. Symbols such as 35mm, COD, AC, etc. count as a single word. Hyphenated words count as two (2) words.

SUBSCRIBER SERVICE

Mail to: SOFTWARE AGE
2211 Fordem Ave., Madison, Wis. 53701

Please fill out the blanks below whether you are changing your address or requesting a new subscription.

☐ Please change my address

☐ Please enter my subscription

____
current occupational specialty

____
highest degree awarded—year

ATTACH LABEL HERE

FOR CHANGE OF ADDRESS

If you're moving please let us know 5 weeks before changing your address. If you do not have your label print your old address below.

Old home address _________________

City, State _________________ (Zip)

Name _________________

New home address _________________

City, State _________________ (Zip code)

software age

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

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.

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 ______________________ Signature ______________________

Home Address ______________________ (city) Home Phone ______________

______________________ (state) ______________ (ZIP code) Age ________ 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
P. O. BOX 2076
2211 FORDEM AVE., MADISON, WIS. 53701

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, P.O. Box 2076, 2211 Fordem Avenue, Madison, Wisconsin 53701. (**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.** Army Air Force Exchange Service	26
☐	**2.** Blue Cross–Blue Shield	4th cover
☐	**3.** Conductron Corp.	4
☐	**4.** Eastern Airlines	17
☐	**5.** General Electric Co., Information Service Dept.	3
☐	**6.** General Electric Co., Space Systems	28
☐	**7.** Honeywell, Inc.	14
☐	**8.** Hughes Aircraft Co., Ground Systems Group	32
☐	**9.** IBM Corp.	7
☐	**10.** Lockheed Electronics Co.	2nd cover
☐	**11.** Lockheed Missiles & Space Co.	22
☐	**12.** Pratt & Whitney Aircraft	13
☐	**13.** RCA, Information Systems Div.	3rd cover
☐	**14.** Sikorsky Aircraft	5
☐	**15.** Univac Corp.	10 & 11

EMPLOYMENT AND SEARCH AGENCIES

		Page
☐	**16.** American Computer Personnel, Inc.	18
☐	**17.** Callahan Center for Computer Personnel	23
☐	**18.** Computer Careers, Inc.	15
☐	**19.** Computer Personnel Agency, Inc.	16
☐	**20.** Robert Half Personnel Agencies	29
☐	**21.** Input, Inc.	16
☐	**22.** Everett Kelley Associates, Inc.	22
☐	**23.** LaSalle Associates, Inc.	12
☐	**24.** Lawrence Personnel	30
☐	**25.** Management Scientists, Inc.	21
☐	**26.** RSVP Services	15
☐	**27.** Sheridan Associates, Inc.	21
☐	**28.** Source EDP	17
☐	**29.** Welles Recruiting Systems, Inc.	12

PRODUCTS AND SERVICES
(Use Reader Service Card)

Data Link Corp.	19
Information & Systems Press	21

☐ 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

P. O. Box 2076
2211 Fordem Avenue
Madison, Wisconsin 53701