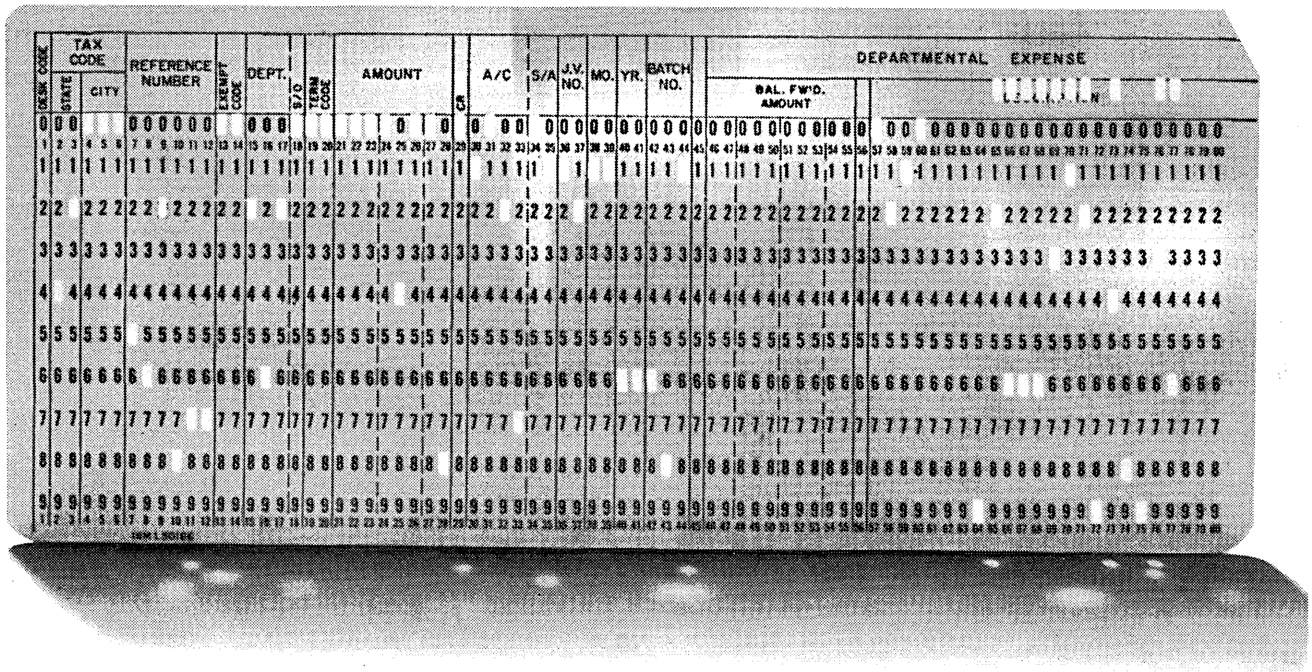
February, 1969

computers and automation

Computer-Programmed Crane



Get it

ou ur

s n.

If keypunching is a major bot
flow of data through your com
system.

That's what the Oilwell Divis
With a little help from Friden.

Before Oilwell's comput
invoice, every branch office
mailed to the home office for cc

The whole routine took a full
automatic writing machine by
now does it in two days! On
chine was installed in each
well's 85 branch offices. No
original order is typed with th

*A TRADEMARK OF FRIDEN, INC.

s. Because these cards con-
ped on the order, 90% of the
atically error free.

pes out a priced order, it pro-
ontaining all invoice informa-
computer reads this tape
re mailed within 48 hours of

will cost to get keypunching
our system. Call your nearest
office. Or write Friden, Inc.,
eandro, California 94577.
and service throughout the

Friden
DIVISION OF SINGER
DIVERSIFIED·WORLDWIDE

gnate No. 23 on Reader Service Card

Letters To The Editor

Handling Small Area Data — Comments

We were pleased and complimented that you reprinted the text of the talk that I made at the Federal Statistics Users Conference ("Handling Small Area Data With Computers," Dec. 1968, p. 16). We would like to purchase several hundred reprints for distribution within the Company and to some of our clients.

It will be interesting to see what reaction you get from the italicized headline that says "It's no trick at all for the computer to generate a million or so statistics for one block in a medium-sized city." To be sure, this is possible, and it would *really* be a job to figure out what to do with them. Actually, what we meant was that we can generate a million or so statistics at the block-total level — and even then the job of what to do with them is tough enough!

Especially including the sensational headline, your people did an excellent job of setting up the text, and we extend our thanks and compliments to all concerned.

R. S. HANEL
Vice Pres. and Mgr.
Urban Statistical Div.
R. L. Polk & Co.
431 Howard St.
Detroit, Mich. 48231

Software

As this letter slides "Across the Editor's Desk," could it possibly be included under "software" in your next issue?

As a tuition-scrambling undergraduate at Lafayette College, I can't finance a full ad, but would appreciate a word in the newsletter section of your exciting publication about the MI FORTRAN program I've developed.

DANIEL BROCKLEBANK
200 High Street
Easton, Pa. 18042

(Ed. Note — *We are pleased to include the announcement of your program in the "Software" section of this issue — and we invite all undergraduate college students to submit new, worthwhile programs of their design to us to be considered for announcement.*)

Paid vs. Controlled Circulation

Computers and Automation has offered informative and challenging articles in the field of automation, and I am much interested in continuing my subscription on a professional controlled circulation basis. With your competitors offering this service, I cannot justify the expenditure for renewal of a subscription to your magazine.

W. S. RINGLER
United States Steel Corp.
Homestead Works
Homestead, Pa. 15120

For the past two years, your magazine has been the only one in the data field for which we have had to pay for a subscription. We are on the qualified list of every other publication in the field.

We feel that we belong on the qualified list for your magazine too, and ask that delivery of the magazine be continued on that basis.

IRA M. KAY
Southern Simulation Service, Inc.
P.O. Box 1155
Tampa, Fla. 33601

(Ed. Note — *Unlike most other magazines in the computer field,* Computers and Automation *is a paid circulation magazine. We have only two complimentary lists, an exchange list and a list for copies to bona fide prospective or actual advertisers. Our theory is that to keep our magazine as good as possible, and to maintain a maximum amount of editorial freedom, it should be paid for by its readers, not by its advertisers. "He who pays the piper calls the tune." Our number of subscribers has grown from about 10,000 in January, 1968, to about 16,000 now — and all of them are paid.*)

Numbles

I'm enclosing my solution to Numble 6812 ("Use soft words and hard arguments."). The inclusion of these puzzles in C&A has made the magazine even more entertaining and enjoyable. Congratulations on a fine publication!

RICHARD T. LYNCH
City of Madison Data Processing Center
210 Monona Ave.
Madison, Wisc. 53709

computers and automation

Vol. 18, No. 2 — February, 1969

Editor	**Edmund C. Berkeley**
Associate Editor	**Sharry Langdale**
Assistant Editors	**Moses M. Berlin**
	Linda Ladd Lovett
	Neil D. Macdonald
Market Research Director	**I. Prakash**
Art Directors	**Ray W. Hass**
	Daniel T. Langdale
Contributing Editors	**John Bennett**
	Andrew D. Booth
	Dick H. Brandon
	John W. Carr III
	Ned Chapin
	Alston S. Householder
	Peter Kugel
	Leslie Mezei
	Rod E. Packer
	Ted Schoeters
Advisory Committee	**T. E. Cheatham, Jr.**
	James J. Cryan
	Richard W. Hamming
	Alston S. Householder
	Victor Paschkis
Fulfillment Manager	**William J. McMillan**

Advertising Representatives

NEW YORK 10018, Bernard Lane
37 West 39 St., 212-279-7281

CHICAGO 60611, Cole, Mason, and Deming
221 N. LaSalle St., Room 856, 312-641-1253

PASADENA, CALIF. 91105, Douglas C. Lance
562 Bellefontaine St., 213-682-1464

SAN FRANCISCO 94123, Richard C. Alcorn
2152 Union St., 415-922-3006

ELSEWHERE, The Publisher
Berkeley Enterprises, Inc.
815 Washington St., 617-332-5453
Newtonville, Mass. 02160

Editorial Offices
BERKELEY ENTERPRISES, INC.
815 WASHINGTON STREET,
NEWTONVILLE, MASS. 02160
CIRCULATION AUDITED BY
AUDIT BUREAU OF CIRCULATIONS

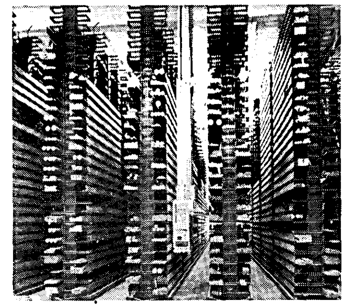
815 Washington St., Newtonville, Mass. 02160, by Berkeley Enterprises, Inc. Printed in U.S.A. Subscription rates (effective March 1, 1969): United States, $18.50 for 1 year, $36.00 for 2 years, including annual directory issue — $9.50 for 1 year, $18.00 for two years without annual directory; Canada, add 50¢ a year for postage; Foreign, add $3.50 a year for postage. Address all editorial and subscription mail to Berkeley Enterprises, Inc., 815 Washington St., Newtonville, Mass. 02160. Second Class Postage paid at Boston, Mass.
Postmaster: Please send all forms 3579 to Berkeley Enterprises, Inc., 815 Washington St., Newtonville, Mass. 02160.
Change of address: If your address changes, please send us both your new address and your old address (as it appears on the magazine address imprint), and allow three weeks for the change to be made.

computers and automation

Vol. 18, No. 2, February, 1969 *The magazine of the design, applications, and implications of information processing systems.*

Special Feature:
Software and Programming

Regular Features



In the center of the front cover picture, a computer-programmed crane is shown. It travels at 400 feet-per-minute in aisles of a steel bar and tubing warehouse at Joseph T. Ryerson & Son, Inc., Chicago. The computer-controlled crane automatically locates required items from among thousands in the 33,000-ton inventory, picks up a load, takes it to a processing station, and returns another load to the racks — all in about 3½ minutes per trip. For more information, see page 53.

NOTICE

C&A SUBSCRIPTION RATES CHANGE MARCH 1. YOU MAY RENEW AT OLD RATES BEFORE FEB. 28. SEE ANNOUNCEMENT ON PAGE 17.

Departments

Machine Language, and Learning It

In my experience, it seems to be true that almost all extremely able computer programmers prefer machine language (assembly language) to any other programming language. They apparently believe that they can solve all the unusual (and therefore really interesting) programming problems faster and more efficiently by using machine language than by using any higher-level programming language such as COBOL, FORTRAN, LISP, etc.

So how should you teach machine language?

— Use the machine language of a very common computer? Then you do not give students the practice on other kinds of machines from other manufacturers.

— Use machine language on an invented or hypothetical computer? This was valid enough perhaps half a dozen years ago — but it is a silly waste of time nowadays, for none of the learning on the invented machine is usable, verifiable, or *real*.

The novel, exciting, and powerful answer worked out by a group of Upper New York State computer people was this:

> Invent a reasonable machine language for a hypothetical computer (MOHAC) AND implement it on any number of different computers.

This has all the earmarks of a splendid answer to the problem.

The answer is now expressed in (1) a fascinating and well-written book called "The Elements of Digital Computer Programming" by Edwin D. Reilly, Jr., and Francis D. Federighi,[1] and (2) the organization of a users' group (the MOHAC Users' Group) to extend and amplify the implementations.

I can do no better now than to quote a number of passages from the book, as preparation for a final recommendation stated at the end of this editorial.

(Start of Quotation)

This book is concerned primarily with machine language programming. Let us state at the outset that it is not our intent to propagate archaic and quaint programming techniques at the expense of newer methods. There is no question in our minds that applications programming should now be confined to higher-level languages such as ALGOL, COBOL, FORTRAN or PL/1, except for unusual instances where the ultimate in machine efficiency is necessary. . . .

What then is the purpose of this book? Merely this: We want to teach the fundamental principles of digital computer programming to college and university students who have had no prior exposure to the subject and who do not necessarily have any mathematical preparation beyond high school algebra. . . . We hold that training in the higher-level languages is best taught after students are given at least a rudimentary knowledge of a typical digital computer organization plus some feeling of what it's like to program a loop "the hard way." . . .

But why a *new* book? . . . Each one we examined had one or more of the following shortcomings:

(1) It required a mathematical preparation inconsistent with our desire to train university students from *any* discipline.

(2) It was based on an actual computer which would not be available to Albany students over a reasonably long time frame.

(3) It was based on a hypothetical computer which could support only a theoretical course devoid of actual programming practice.

To avoid the first pitfall, we merely exercised care in the selection of example problems and student exercises. To straddle the last two together, we chose to use a hypothetical computer, the MOHAC, and to make it real through simulation. . . .

MOHAC contains a REPEAT instruction which lays the groundwork for . . .

MOHAC has both conventional arithmetic registers and a pushdown stack . . .

MOHAC has an extensive range of software, ranging from its own operating system to a complete assembly routine. In between are loaders, dumps, disassemblers, and floating-point routines for a number of the common mathematical functions. . . .

It is our earnest hope that no one who is attracted to this material will be dissuaded by the lack of an available MOHAC. To protect against this, the MOHAC simulator is itself written in FORTRAN and has been compiled successfully on a wide variety of contemporary computers. . . .

We wish to acknowledge that the MOHAC described in this book is an expanded version of a machine designed by members of the Education Committee of the Hudson-Mohawk chapter of the Association for Computing Machinery during a period when one of us (EDR) was committee chairman. . . .

We are grateful to the several hundred students at Albany State who have worked through this text problem by problem . . . primarily because they helped us recapture the spirit and challenge of that first encounter with the fascinating machines that insist on doing what we told them rather than what we *thought* we told them. . . .

MUG (MOHAC Users' Group) is an organization of MOHAC users who are interested in sharing programs and programming techniques in the same spirit as any of several organizations centered about particular digital computers. Available from MUG are MOHAC simulators for a wide variety of computers. . . . The address of MUG is Box 2675, Schenectady, N.Y. 12309.

(End of Quotation)

Recommendation by this enthusiastic editor: To all persons who desire to teach machine language programming to students in a significant and general way: Send in your order for this book and send in your application to join MUG. (P.S. — I have done so.)

Edmund C. Berkeley
Editor

1. Published by Holden Day Inc., 500 Sansome St., San Francisco, Calif., 1968, 222 pages.

NEW PATENTS

Raymond R. Skolnick
Patent Manager
Ford Instrument Co.
Div. of Sperry Rand Corp.
Long Island City, N.Y. 11101

The following is a compilation of patents pertaining to computers and associated equipment from the "Official Gazette of the U. S. Patent Office," dates of issue as indicated. Each entry consists of: patent number / inventor(s) / assignee / invention. Printed copies of patents may be obtained from the U.S. Commissioner of Patents, Washington, D.C. 20231, at a cost of 50 cents each.

November 26, 1968

3,413,613 / David L. Bahrs, Liverpool, N. Y., and John F. Couleur, Richard L. Ruth, and William A. Shelly, Phoenix, Ariz. / General Electric Company, a corporation of New York / Reconfigurable data processing system.

3,413,614 / Susumu Seki, Kokubunji-shi, Japan / Hitachi, Ltd., Tokyo, Japan / Semipermanent memory device.

3,413,615 / John L. Botjer, Hyde Park, Edward O. Donner, Poughkeepsie, and Harold E. Frye and Howard S. Keeler, Wappingers Falls, N. Y. / International Business Machines Corporation, Armonk, N. Y., a corporation of New York / Delay line buffer storage circuit.

3,413,616 / Arwin B. Lindquist, Poughkeepsie, N. Y. / International Business Machines Corporation, New York, N. Y., a corporation of New York / Persistent supercurrent associative memory system.

3,413,617 / James L. Smith, Bedminister, N. J. / Bell Telephone Laboratories, Inc., New York, N. Y., a corporation of New York / Waffle-iron magnetic memory access switches.

3,413,618 / Joseph P. Shuba, Joliet, Ill. / Automatic Electric Laboratories, Inc., Northlake, Ill., a corporation of Delaware / Memory apparatus employing a plurality of digit registers.

3,413,619 / Bruce E. Briley, La Grange Park, Ill. / Automatic Electric Laboratories, Inc., Northlake, Ill., a corporation of Delaware / Magnetic memory systems employing myriaperture devices.

3,413,620 / Joseph M. Bernstein, Omaha, Nebr., George J. David, Addison, Ill. / Automatic Electric Laboratories, Inc., Northlake, Ill., a corporation of Delaware / Memory core matrix with printed windings.

3,413,621 / Hisao Maeda, Ota-ku, Tokyo-to, Japan; Hisaaki Maeda, heir of said Hisao Maeda, deceased / Magnetic storage element having constant flux distribution.

3,413,628 / Ward W. Beman, Glendale, Calif. / Whittaker Corporation, a corporation of California / Random access data storage apparatus.

AS WE GO TO PRESS

THE JUSTICE DEPARTMENT FILED AN ANTITRUST SUIT AGAINST INTERNATIONAL BUSINESS MACHINES CORP. ON JAN. 17. The suit charged that IBM, since 1961, has pursued four lines of action to illegally prevent other manufacturers of general-purpose digital computers from competing effectively.

First, it charged that IBM has maintained a policy of quoting a single price for computer hardware, software, and support which has limited the development of an independent software and support industry.

Second, the suit charged that IBM has used its accumulated software and related support to preclude other computer makers from competing effectively for customers.

Third, IBM was charged with introducing "selected computers, with unusually low profit expectations, in those segments of the market where competitors had unusual competitive success" in order to restrain competitors from entering or remaining in the general-purpose computer market. Further, the suit alleged that IBM announced future production of new models for competitive markets "when it knew that it was unlikely to be able to complete production within the announced time."

And fourth, the suit charged that IBM dominated the educational market for computers by granting "exceptional discriminatory allowances in favor of universities and other educational institutions." The educational market was important, the suit asserted, because it was large and because it influenced computer purchasing decisions by commercial computer users.

The suit requested "such divorcement, divestiture, and reorganization" of IBM's "business and properties" as may be necessary "to restore competitive conditions to the general-purpose computer industry."

IBM responded to the suit by saying it will "defend itself forcefully against this action, which it believes is unwarranted and without foundation." The IBM statement went on to say: "One of the key issues in these discussions has been whether there is sufficient competition in the data processing industry or whether IBM has such monopolistic power that fully effective competition does not exist.

"Evidence of the open and strongly competitive nature of the computer business is abundant. Virtually nonexistent 20 years ago, it has grown into a multi-billion-dollar industry that has attracted more than 60 manufacturers of computer systems and some 4,000 companies dealing in related equipment, support, and services.

"Despite this evidence of strong and increasing competition, the Government has decided to bring this suit."

The Justice Department suit follows a civil antitrust suit filed by Data Processing Financial & General Corp. on Jan. 3 which accuses IBM of numerous violations of the anti-trust laws in its manufacture and distribution of computers, software, and related services and products, and the civil suit filed by Control Data Corp. against IBM on Dec. 11, 1968, which was reported last month in this column.

THE PURCHASE BY WESTERN UNION OF AT&T'S TWX SERVICE HAS BEEN APPROVED BY THE BOARDS OF BOTH COMPANIES. The estimated price of the service and the associated equipment Western Union will acquire from the Bell System is about $80 million. The Bell System's TWX and Western Union's competing Telex are telegraph exchange services. Subscribers to each of the services dial connections with other subscribers of that service, and use directly connected machines to exchange typewritten messages and data. In addition, subscribers to Telex can dial a Western Union computer to obtain special automatic services which add to the flexibility of the basic exchange service.

The Bell System currently has a total of approximately 40,000 stations in customer TWX service; Western Union's Telex, which has been competing with TWX for ten years, has a total of more than 26,000.

The agreements are subject to approval by the boards of the Associated Bell telephone companies, and other preliminary conditions, including negotiation of arrangements with other carriers and regulatory approvals.

THE NATIONAL HEART INSTITUTE HAS AWARDED A $2 MILLION GRANT TO DR. LYSLE PETERSON AND HIS ASSOCIATES AT THE UNIV. OF PENNSYLVANIA to continue a project in which the human circulatory system will be simulated by computer. The computer model of the circulatory system will include the cardiovascular system and the aspects of the nervous system, the kidney system, and the endocrine system that control and regulate the cardiovascular system.

The model should be a major contribution to the development of an artificial heart that will react to changes in the environment in the same way the real heart reacts, Dr. Peterson said. He feels that eventually an artificial heart may come equipped with a miniature computer that will collect all the information (about the constant changes in the environment and in the rest of the body that affect the heart) and adjust the heart as efficiently as a natural heart adjusts.

A 60-man team of physiologists, programmers, mathematicians, biologists, and clinical experts has been working on the project for six years, supported by grants totalling approximately $3 million.

SHOULD YOUR ORGANIZATION BE IN "THE COMPUTER DIRECTORY AND BUYERS' GUIDE, 1969"?

The Fifteenth Annual Edition of the "Computer Directory and Buyers' Guide" (a special issue of C&A to be published additionally in June 1969) is now being prepared.

If your organization has recently entered the field of computers and data processing, or if you are not sure that we have your organization's name on our mailing list to receive entry forms in February for this Directory, please write us at once and ask us for your free entry forms:

Directory Editor
Computers and Automation
815 Washington Street
Newtonville, Mass. 02160

*7.4731

(Hello. Is this the computer?)

11111

(Yes it is. Go ahead.)

#8.3000.77.73842

(Sold 3,000 units item #77 to Allu Corp.)

*1.1.0

(Availability confirmed. Account current.)

#4.12.965

(What quantity of item #12 is available for shipment to the Duluth area?)

*9.22000.12.965

(You can have 22,000 pieces in Duluth on Friday.)

11.8

(Thank you.)

10.8

(Don't mention it.)

A new concept in data transmission systems. Now you can put your computer or data bank to work anywhere there's a phone. And Hi-G's remote computer terminal system gives you so many advantages at such a low cost, you'll find it hard to believe.

Like hard copy readout, for example. At once a verification of the message you're sending or receiving, and a permanent printed record of every transmission, both ways.

Like input and output—the simple ability to feed accurate, verified information into your computer and take needed information out. (Keeps a computer honest and up-to-the-minute.)

Like compatability with any computer and information system. Which means you can plug in at no additional cost — get immediate maximum usefulness out of whatever system you're using.

And at under 15 pounds, this unit is as mobile and flexible as you are.

For about $20 a month, Hi-G's RCT 203 computer terminal opens up a smart new variety of uses for companies large and small. Salesmen can expedite availabilities and orders from the field. Service users can instantly verify and update accounts, charge cards, plates. Plant managers can check — or order — supplies, at a moment's notice. If you've got a central payroll you can feed it data with a phone call. *The only limit to how you can use this unique instrument for data transmission is imagination.*

Let our experienced system specialists supply some of that imagination. Just fill in the coupon or give us a call.

Hi G Incorporated, CA 1
Spring St. & Route 75
Windsor Locks, Conn. 06096
Gentlemen:

I like the idea of getting maximum computer usefulness with your new low-cost remote computer terminal phone system.

☐ Please send me complete data by mail.

☐ I can't wait for the mails. Please have your representative contact me for an appointment.

NAME__________ TITLE______

COMPANY__________

ADDRESS__________

CITY______ STATE_____ ZIP____

Designate No. 24 on Reader Service Card

MULTI-ACCESS FORUM

MARTIN LUTHER KING MEMORIAL PRIZE CONTEST

Computers and Automation has received an anonymous gift and announces the annual Martin Luther King Memorial Prize, of $300, to be awarded each year for the best article on an important subject in the general field of:

> The application of information sciences and engineering to the problems of improvement in human society.

The judges in 1969 will be:

> Dr. Franz L. Alt of the American Institute of Physics; Prof. John W. Carr III of the Univ. of Pennsylvania; Dr. William H. Churchill of Howard Univ.; and Edmund C. Berkeley, Editor of *Computers and Automation.*

The closing date for the receipt of manuscripts this year is April 30, 1969, in the office of *Computers and Automation,* 815 Washington St., Newtonville, Mass. 02160.

The winning article, if any, will be published in the June issue of *Computers and Automation.* The decision of the judges will be conclusive. The prize will not be awarded if, in the opinion of the judges, no sufficiently good article is received.

Following are the details: The article should be approximately 2500 to 3500 words in length. The article should be factual, useful, and understandable. The subject chosen should be treated practically and realistically with examples and evidence — but also with imagination, and broad vision of possible future developments, not necessarily restricted to one nation or culture. The writings of Martin Luther King should be included among the references used by the author, but it is not necessary that any quotations be included in the article.

Articles should be typed with double line spacing and should meet reasonable standards for publication. Four copies should be submitted. All entries will become the property of *Computers and Automation.* The article should bear a title and a date, but not the name of the author. The author's name and address, and four or five sentences of biographical information about him, should be included in an accompanying letter — which also specifies the title of the article and the date.

WORKSHOP ATTEMPTS "OBJECTIVE" EVALUATION OF TIME SHARING SERVICES — VENDORS CHALLENGE THE TESTS

Dr. Ernest Schubert
National Information Research Institute
P.O. Box 3358
Santa Monica, Calif. 90401

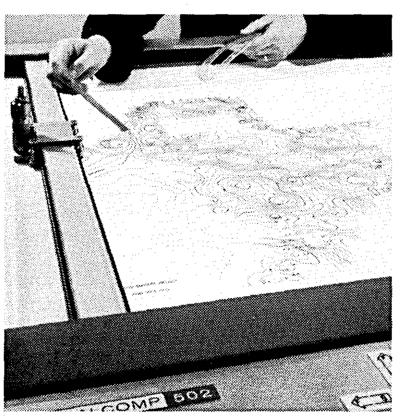
The National Information Research Institute (NIRI) held a workshop on "Comparative Test & Evaluation of Time-Sharing and Remote EDP Services" in Los Angeles last fall. The announced purpose of the workshop was to give the prospective time-sharing service customer some tools he could use to evaluate competitive offerings. But the workshop attracted about six times as many sellers as buyers, and turned into three days of lively discussion between NIRI representatives and the systems designers and marketers who comprised the bulk of the attendance.

The sharpest challenge to NIRI came on the ten benchmark programs which NIRI had programmed on the nine services being evaluated. Ground rules for the tests were: (1) NIRI purchased all services at regular rates, in order to put the Institute squarely in a user's shoes as far as rates, services, and customer support are concerned; the purchase of services was also intended to assure objectivity of the testing and credibility of results; (2) All systems were subjected to the same tests under uniform conditions; and (3) Tests were conducted in public, and in the presence of a designated technical representative of the seller of the time sharing service being tested.

Critics complained that the benchmark programs were simplistic, and did not test each system to the full limits of its own specs. NIRI countered by saying that big enough differences in actual on-line performance showed up even on simple tests to be interesting — if not eyebrow-raising — to any prospective customer.

At a deeper level, the underlying philosphy of the NIRI tests was challenged on the grounds that they tested generic system capabilities (such as memory size, arithmetic speed, file management, execution modes, transmission error rates), and not real-life problems or applications (such as a binomial distribution or payroll management and check-writing). We at NIRI maintain that the high development cost and the proprietary sensitivity of applications programs

3 NEW SOFTWARE PACKAGES FOR SALE?

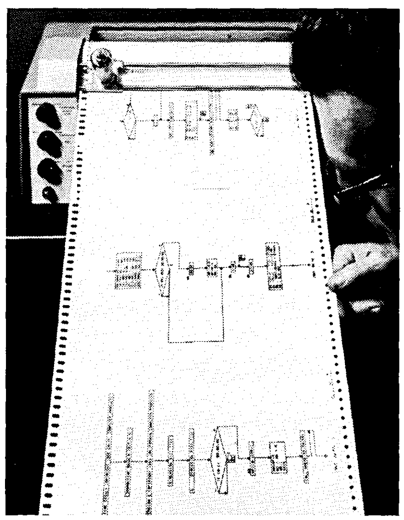


CALCOMP GPCP

(General Purpose Contouring Program)

This program automatically plots functions of two independent variables in the form of contour diagrams or maps. Written in FORTRAN IV, it is easy to use, extremely flexible, accurate, economical. It is about 30 times faster than manual and does jobs impossible to do by hand. Used with any CalComp plotter system, GPCP can be applied to such fields as geophysics, meteorology, engineering, biology and medicine.

Now available for a one-time lease charge of.....................$10,000

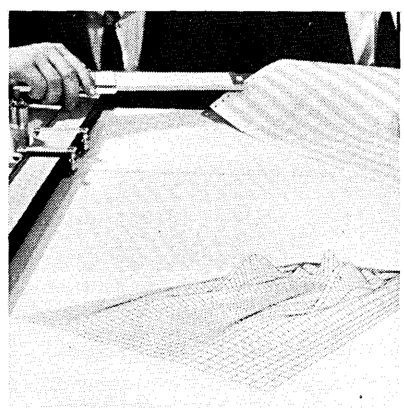


CALCOMP FLOWGEN/F

(Flowchart software package)

This program allows any computer programmer to automatically produce flowcharts of his program on any CalComp plotting system. An extremely useful tool in documentation of checked-out programs, it is even more valuable during the check-out phase of a new program or a new computer. FLOWGEN/F is fast, time-saving, accurate.

Now available for a one-time lease charge of......................$4,500



CALCOMP THREE-D

(Perspective drawing software package)

This program is a set of FORTRAN subroutines for use with any CalComp digital plotting system to produce perspective drawings of surfaces. It can also generate stereoscopic views of surfaces, and, with CalComp Model 835 microfilm plotter, can produce animated films. Easy to use, flexible and economical, THREE-D can be applied to such fields as marketing, engineering, toolmaking and designing.

Now available for a one-time lease charge of......................$3,000

CALCOMP

Leader in Computer Graphics

California Computer Products, Inc., Dept.A-2 305 N. Muller St., Anaheim, California 92803 • (714) 774-9141

Designate No. 18 on Reader Service Card

precludes their use and acceptance as across-the-board benchmark tests.

In addition, we find that although everybody gives lip service to objective, impartial testing — just as they do to God, Mother, and BASIC — the truth is that the seller really wants to create a special mystique about his product. He tries to get his intended customer to think that his product is the only one in the universe. When pressed, the seller quite understandably wants to expose only his product's strongest points to comparison. So from a seller's viewpoint, if there are going to be any tests at all, they should be designed by his sales department.

The most sensible approach to benchmark testing is to build the tests around elemental functions. In batch processing, IBM has heavily promoted the concept of a "statistical mix" of functions to characterize business, scientific, real-time control, and other kinds of applications. At NIRI, we think this approach makes sense for time sharing too.

Our lively controversy over benchmark programs apparently changed few minds, however, and sellers left muttering about the inconclusiveness of the tests. One attendee, however, admitted to having second thoughts about a check-writing application program he was working on, in the light of transmission error rates of 8 per 1000 which turned up in the NIRI benchmark tests.

In addition to the benchmark tests, the workshop also considered comparative detail data describing other time-sharing system parameters of interest to prospective users such as hardware features, executive control, edit capabilities, file management capabilities; language capabilities, programming aids, and rate structures of services.

The tests were performed on the following systems: Computer Time Sharing Inc. (CDC 3300); Remote Computing Corp. (Burroughs B5500); Computer Center Corp. (PDP-10); General Electric Mark I (GE 265); General Electric Mark II (GE 635); Tymshare, Inc. (SDS 940); Allen Babcock Co. (IBM 360/50); International Business Machines BASIC (IBM 360/50); and QUICKTRAN (IBM 7044).

NIRI is a militant partisan of the user. We feel that neither user organizations, professional organizations, nor tax-supported non-profit organizations have assumed responsibility for applying their money and expertise to the user's problem of comparative system evaluation, so we are doing it. It is our feeling that many of the basic attitudes toward customers that have prevailed during the computer industry's emergent phase will have to be cast aside if the industry wants to succeed in a mass market, and the workshop seems to confirm that theory.

At the request of workshop attendees, the ten copyrighted benchmark programs will be included in the Workshop Proceedings, which will also include the printouts, comparative cost analyses, and six tables of detail data on comparative system attributes. Information about obtaining copies of the Proceedings may be obtained from the address above. NIRI has announced a follow-up workshop in Philadelphia in May 1969 at which time-sharing and remote EDP systems — CPU's, peripherals, and associated software — will be tested under similar ground rules. □

LEGAL PROTECTION FOR COMPUTER PROGRAMS

Association of Independent Software Companies
P.O. Box 4548
Washington, D.C. 20017

The Association of Independent Software Companies was formed in May, 1968, in Washington, D.C. The following is a Position Paper favoring legal protection for computer programs which was adopted by the Association at their annual meeting on November 21, 1968.

Introduction

There is no question that computer programs represent an extremely valuable body of industrial property. The adequate protection of this property is vital to the business interests of the members of the Association of Independent Software Companies. The Association should recognize that the Patent Office is discriminating against inventors who chose a program as the preferred embodiment in favor of a Hardware embodiment for the same inventive concepts. The creation of programs is the major function of most of the membership of this Association.

It is recommended that the Association take a strong position in favor of patent protection for computer programs.

Background

For many years there has been controversy regarding legal protection for computer programs and in particular patent protection. The computer manufacturers have been outspoken in their opposition to patent protection. However, in August of 1966, guidelines for patents on computer programs were published by the Patent Office. Commissioner Brenner said, "These guidelines were prepared after careful consideration by the persons in the Patent Office who were most skilled in the computer field, including members of the Board of Appeals. They were not intended to establish new law or practice, and were not based on considerations of expediency. Rather they represented an attempt to state a practice which should be uniformly followed in the Patent Office."

In November of 1966, a report of the President's Commission on the patent system was transmitted to the President. It in essence said that the patent system was wonderful for all areas of technology except computer programs. The commission made the following remarks concerning patents in general, "The patent has in the past performed well its *Constitutional mandate to promote the progress of useful arts.* The members of the commission unanimously agreed that a patent system today is capable of continuing to provide an incentive to research, development and innovation. They have discovered no practical substitute for the unique service it renders.

"First, a patent system provides an incentive to invent by offering the possibility of reward to the inventor and to those who support him. This prospect encourages the expenditure of time and *private* risk capital in research and development areas.

"Second, and complimentary to the first, a patent system stimulates the investment of additional capital needed for the further development and marketing of the invention. In return the patent owner is given the right, for a limited period, to exclude others from making, using or selling the invented product or process.

"Third, by affording protection, a patent system encourages early public disclosure of technological information, some of which might otherwise be kept secret. Early disclosure reduces the likelihood of duplication of effort by others and provides a basis for further advances in the technology involved."

About March of 1967, the administration proposed its patent reform bill which included a section that eliminated computer programs from patentable subject matter.

In April of 1967, a panel invited to the President's Office of Science and Technology opposed the elimination of computer programs from patentable subject matter.

In January of 1968, Commissioner Brenner said, in his testimony before the Senate, that it may be premature to enact legislature at the present time and recommended that a section excluding computer programs from patentable subject matter not be included in any patent reform legislation.

In April of 1968, a revised version of the Senate's Bill for patent reform was published which contained no reference to computer programs. The elimination of patents for programs has been deleted from all subsequent revisions to the Senate Bill.

On October 22 of 1968, Commissioner Brenner said, "In view of the controversial nature of the subject, and of the recommendation of the President's Commission that computer programs be excluded from patentable subject matter, it was considered advisable to delay further action on the proposed guidelines until it became clear what position would be taken as to legislation. By July 1968, however, it had become apparent that no legislation affecting computer programs would be passed by the 90th Congress, and it therefore appeared to be highly desirable that guidelines should be adopted. Accordingly, further intensive consideration has been given to the guidelines originally proposed in the light of all the comments received both orally at the hearing and in writing, and such revisions have been made as appeared to be in order. You will be interested to learn that the revised guidelines covering the examination of patent applications relating to computer programs will be officially published shortly in the *Official Gazette*.

"In effect, the guidelines provide that a computer programming process does not constitute proper subject matter for patent protection under the statutes if its ultimate result is merely a numerical or informational statement, as distinguished from the alteration of the form, composition or condition of a physical substance or object." However, he later said, "Patents are intended to protect novel and unobvious inventions regardless of the form or appearance in which they may be manifested."

Could it be that the administration is dictating what it couldn't legislate?

The Problem

The problem is basically an economic one. Should Software Manufacturers have the right to compete with Hardware Manufacturers? Does Software Invention (Investment) deserve protection as adequate as Hardware Invention (Investment)?

Let's take a hypothetical example: Software Manufacturer "A" decides he can build a COBOL compiler for the 360 that will be 25% more efficient. "A" proceeds to build this product which employs some inventive concepts and expends one million dollars. The product performs as originally estimated. "A" advertises in the trade publications that it offers a COBOL compiler which is 25% more efficient than IBM's for only $5,000.00. "A" distributes literature and user manuals to prospective clients and runs demonstrations. He even installs a few systems. However, as word gets out about this new system, the users put pressure on IBM to produce an improved system. IBM "reluctantly" agrees to do this. In building the new version some of "A"'s principles are employed which were disclosed in the promotion of "A"'s product. Since patent protection was not available to "A", he has no legal position to protect his investment in the inventive concepts that he developed. "A" lost a lot of money.

You may say that can't happen but a very close parallel to the situation described above happened with Digitec and a FORTRAN Compiler. It happened to ADR with a system called ESI. It will continue to happen when the economics of the situation or the customer pressure require it. Software Companies cannot base the existence of their business on the philanthropy of the computer manufacturers. This will become even more important if separate pricing of Hardware and Software comes to pass.

In the COBOL Compiler example used above, copyrights and trade-secrets did not protect "A" but a patent would.

Proposed Plan of Action

1. The Association and its members adopt a position favoring patents for computer programs and make this position known to the Patent Office, the Justice Department, and as many members of Congress as practical.
2. That a special effort be undertaken to acquaint the new administration with the need for Patent Protection in the hope that legislation favoring patents for programs could be added to the proposed patent reforms.
3. Engage counsel and file an Amicus Curiae brief with the U.S. Court of Customs and Patent Appeals in support of the pending appeal of Mobil Oil for a patent on a computer program.

Resolution

We move that the members present and voting in this Association adopt the Proposed Plan of Action as outlined in IV above.

This position was adopted on Thursday, November 21, 1968. □

MORE EFFORT SHOULD BE CHANNELED INTO NEW APPLICATIONS AND IMPROVED COMMUNICATION WITH THE COMPUTER

Isaac L. Auerbach, Pres.
Auerbach Corp.
121 N. Broad St.
Philadelphia, Pa. 19107

The computer industry can expect a continuing lull in major hardware breakthroughs during 1969. There is a growing realization that technology for technology's sake alone is not sufficient. In a high-cost, capital equipment-type industry such as the computer industry, the introduction of major change has impact only if such change has significant economic value.

Assume, for example, that a dramatic reduction in the cost of transistorized circuitry within a computer were possible. It would not have any real economic significance, since the central computer circuitry accounts for less than ten percent of the total cost of the computer installation to the customer. To spend hundreds of thousands of dollars, or even millions, for developing the technology and for tooling would not result in a dramatic economic saving. More effort should, instead, be channeled into the development of new applications utilizing existing equipment and know-how, and into improved means of communicating with the computer — that is, the man-machine interface.

There are two main areas in the computer industry where solid growth is likely to take place in the immediate future. One is in the field of computer-assisted instruction, where we shall see the continuing evolution of techniques that improve the ability to learn and the ability to provide instruction supplemental to that offered by the conventional teacher. The industry is faced with a challenge to develop new material for use in computer-aided instruction. While this whole concept is still experimental, in time the field of education will wholeheartedly accept and adapt to these new devices and techniques, thus opening a market that will burgeon in the future.

The second area of anticipated growth is the commercial process-control field. I believe we will see a rapid rise in applications that enable us to accomplish functions previously impossible in this field.

In addition, there will continue to be a lot of activity in the areas of software and time sharing. The proliferation of new software companies throughout the country during 1968 can be attributed to the low capital requirements and ease of entry into the software field. However, a large number of these firms will disappear during 1969 for all of the standard reasons that most new companies disappear — except that the death rate will be even higher than normal. I do not believe that 20 percent of these new firms will celebrate their third birthdays as separate entities. Time sharing services will again expand dramatically during 1969. Of course what is actually being performed under the label of time sharing is quite different from what was originally connoted by the term. But if we include any interaction from a remote terminal through a computer as time sharing, time sharing services can anticipate a doubling in the gross volume of business over that of 1968.

The international field, particularly for computer consultants and software firms, will expand at an increasingly rapid rate. Potential users of these services abroad, as well as users of computers, continue to trail in number behind the United States at present. But the gap is closing, and growth should continue strong during 1969.

The challenge to research organizations or universities who want to make a significant contribution to a major breakthrough in the computer industry lies in the development of languages for communications of intelligence between man and the computer, and the computer and man. This remains one of the most important areas for study in the information sciences. □

PROPOSAL FOR PARTNERSHIPS BETWEEN COMPUTERS AND SOCIETY TO HELP SOLVE INTERNATIONAL AND URBAN PROBLEMS

Gerald A. Finkle
Computer Management Consultants, Inc.
5214 W. Main St.
Skokie, Ill. 60076

Communication is a vital problem for all levels of society today. This is true, as we all know, on a people to people basis. But how much more critical is it on a government to government basis?

Every modern nation now has a significant investment in computers, but unfortunately, they're merely working on routine tasks, mundane tabulating, record-keeping, file and retrieval systems — reducing the paper blizzard of nation-keeping. This is a contribution, to be sure, but think of the potential problem-solving capacity of the computers if a common international computer language were developed, and national data banks regularly communicated with their counterparts in other nations. Might not machines do a better job than diplomats?

It's virtually inconceivable that in this era, with all the variables which present themselves, that any one man, a group of men, or even a series of international bodies, can grasp all of the facts relating to various problems. A computer, with its infallible memory, its tenacity for untiring tasks, and its ability to continually think logically, is really the appropriate tool for the world to turn to for major solutions.

Of course man's decision-making power should not be discounted. Final control of the computer will always rest with the intellectual community, for the quality of the machines' output will always depend upon the quality of thinking and the respect for life which the machines reflect. The mathematics of logic, the language which the computer understands, provides a real incentive for man to think.

Another area where computers should be used as problem solvers is in the area of urban affairs. I'm disturbed when I read that a former cabinet member says: "the first commandment of highway transportation states that there is *no way* to eliminate congestion on the highways and in cities during rush hours. The highway system is not adequate now for rush hours and it never will be."

This is a defeatist attitude which should never be tolerated. Of course there are too many people choking our roads, crowding the parks, ruining the precious green belts which lie so inconveniently on the perimeter of our major metropolitan areas. But let's turn the coin over — and ask, why are they clogging the roads, where are they going?

Urban planners have long recognized that if the uncoordinated subdivision-type of development is undesirable, then the highway-oriented transportation systems, which we have been bent on building for so many years, should be revised so such developments would not be encouraged. It is now possible, using linear programming methods of the computer, to examine all of the alternative, comprehensive metropolitan land use and transportation plans available to a community, and then to forecast accurately such urban factors as population growth, land use, transportation requirements, and community services such as schools, medical facilities and law enforcement.

The time has come when we must take some positive national action in the area of computer utilization. First, a national task force composed of sociologists, urbanologists, city planners, architects, engineers, and information specialists should be organized. This group would work closely with the nation's cities and provide the opportunity for every major community in every state to put some creative thought into some of urban life's most complex and cumbersome problems such as: planning and development, recreation and park use, public housing, school and library planning, air and water pollution control and analysis, public health statistics, rapid transportation systems, and voter registration and election procedures.

Second, the results of the information and thought accumulated by this group should be developed into a Federal data bank through which state and local governments could feed into a national computer-based information system which could develop a profile of each government unit in the nation.

The success of such an effort depends, however, upon good accurate data. The system should permit a partnership between the urban planner and the information scientist designed to focus attention at solving the total problems facing urban areas — not just some of the more obvious isolated ones. It is very important, however, that in the interest of achieving accuracy and alternatives, computer programmers do not invade privacy, nor produce something that is politically unfeasible, nor morally unacceptable. I cannot over-emphasize the importance of these considerations. □

"SUBCOMMITTEE ON PRIVACY AND CONFIDENTIALITY" SEEKS IDEAS FOR DEALING WITH THE PRIVACY PROBLEM POSED BY COMPUTERS

Paul J. Anderson, Chrmn.
Intergovernmental Board on EDP
State of California
915 Capitol Mall
Office Bldg. No. 1
Sacramento, Calif. 95814

An Intergovernmental Board on Electronic Data Processing was set up in the state of California by the California Legislature in 1968. The Board consists of representatives of state, county, and city government and the education system within California. Its function is to establish policy with respect to the operation of intergovernmental data processing systems within California.

One of the first activities of the Intergovernmental Board was to establish a "Subcommittee on Privacy and Confidentiality" in order to cope with the privacy problem posed by computers. Following is a list of that Subcommittee's objectives:

1. Recommend policy to maintain the privacy of information on individuals and to insure the security of personal and organizational records as defined by law where such records are in the custody of electronic data processing facilities of governmental units, in the files and in transmission.
2. Recommend guidelines to the Board for all levels of government on the management of confidential information in electronic data processing facilities through the use of administrative techniques, available legal sanctions, technical methods and the promotion of ethical standards.
3. Recommend means for achieving liaison with:
   a. Computer equipment manufacturers to stimulate the incorporation of machine-security provisions in equipment currently being planned and developed.
   b. The data processing community in all levels of government to achieve coordination of activities to safeguard information.
   c. Other organizations and individuals having similar objectives, e.g., educational institutions, data processing organizations, computer scientists.
4. Recommend legislation on privacy and confidentiality matters as these pertain to electronic data processing facilities.
5. In the vigorous pursuit of these objectives, maintain a responsible concern for the information requirements of government.
6. At least annually, review these objectives and report on accomplishments.

These objectives are presented here in hopes of stimulating an exchange of ideas with other states and, perhaps, federal agencies. Interested parties are invited to contact the Board chairman at the address above. □

INFORMATION INDUSTRY ASSOCIATION IS FORMED

Eugène Garfield
Institute for Scientific Information
325 Chestnut Street
Philadelphia, Pa. 19106

A new trade group, the Information Industry Association (IIA), was formed at a meeting in Philadelphia last fall. William T. Knox, a vice president of McGraw-Hill, was elected as president of the new group. The purpose of the IIA is to provide "an industry identity for those commercial firms that are involved in applying advanced data processing and data communications technology to information services to the business, industrial, government, and academic communities." In addition, the association will "work to foster the growth of private enterprise in the information services field, and will help provide guidelines for industry growth by developing and promoting product and service standards."

Membership in the association is open to all commercial, for-profit organizations and individuals engaged in the creation, supply, or distribution of information products and services. Permanent headquarters for IIA will be established in Washington, D.C. Additional information concerning membership is available from William T. Knox, McGraw-Hill, 330 West 42nd St., New York, N.Y. 10036. □

1969 SYSTEMS SCIENCE AND CYBERNETICS CONFERENCE — CALL FOR PAPERS

Hugh A. Raymond
General Electric Co.
P.O. Box 8555
Philadelphia, Pa. 19101

The 1969 Systems Science and Cybernetics Conference will be held Oct. 22-24, 1969, in Philadelphia, Pa. The theme of the conference will be understanding and controlling natural (non-engineered) systems through the application of systems science and cybernetics, and the use of natural system models in the design of engineered systems.

Papers dealing with this theme are invited. Suggested topics might include:

- the modelling of a particular ecology such as animal, insect and plant life in fresh water rivers and lakes, and the effect of man-introduced pollutants on that ecology;
- the design of urban or regional systems;
- the modelling and control of psychological, sociological, biological, political, or economic systems;
- related engineered systems applications.

Abstracts of approximately 1000 words in length should be submitted by April 15, 1969 to the address above. □

Announcement

Effective March 31, 1969, Edmund C. Berkeley, the editor of Computers and Automation, will no longer be the Editorial Director of the Library of Computer and Information Sciences (of Professional and Technical Programs, Inc., 866 Third Ave., New York, N. Y.); and Isaac L. Auerbach, Jack Belzer, A. D. Booth, Ned Chapin, and Alston S. Householder will no longer constitute the Editorial Board of the Library of Computer and Information Sciences.

JOVIAL PROGRAMMING LANGUAGE COMMITTEE BEING FORMED

A. R. Sorkowitz
NAVCOSSACT, Code 21
Bldg. 196, Washington Navy Yard
Washington, D.C. 20390

The United States of America Standards Institute (USASI) is forming a working group to investigate the desirability of standardizing the JOVIAL programming language. Individuals interested in participating in such a group are invited to contact the chairman of the group at the above address. The first meeting will be held on March 7, 1969, at BEMA, 235 E. 42 St., New York, N.Y. 10017. All interested individuals are invited to attend. □

WHO'S WHO IN THE COMPUTER FIELD, 1968-1969 — ENTRIES

Who's Who in the Computer Field 1968-1969 (the Fifth Edition of our Who's Who), will be published by Computers and Automation during 1969. The Fourth Edition, 253 pages, with about 5000 capsule biographies was published in 1963. The Third Edition, 199 pages, was published in 1957.

In the Fifth Edition we hope to include upwards of 10,000 capsule biographies including as many persons as possible who have distinguished themselves in the field of computers and data processing.

If you wish to be considered for inclusion in the Who's Who, please complete the following form or provide us with the equivalent information. (If you have already sent us a form some time during the past eight months, it is not necessary to send us another one unless there is a change in information.)

WHO'S WHO ENTRY FORM
(may be copied on any piece of paper)

1. Name? (Please print)____________
2. Home Address (with Zip)?____________
3. Organization?____________
4. Its Address (with Zip)?____________
5. Your Title?____________
6. Your Main Interests?

Applications	( )	Mathematics	( )
Business	( )	Programming	( )
Construction	( )	Sales	( )
Design	( )	Systems	( )
Logic	( )	Other	( )
Management	( )	(Please specify)	

7. Year of Birth?____________
8. Education and Degrees?____________
9. Year Entered Computer Field?____________
10. Occupation?____________
11. Publications, Honors, Memberships, and other Distinctions?____________

(attach paper if needed)

12. Do you have access to a computer? ( )Yes ( )No
    a. If yes, what kind of computer?
    Manufacturer?____________
    Model____________
    b. Where is it installed:
    Manufacturer?____________
    Address?____________
    c. Is your access: Batch? ( ) Time-shared? ( )
    Other? ( ) Please explain:____________
    d. Any remarks?____________
13. Associates or friends who should be sent Who's Who entry forms?

Name and Address

(attach paper if needed)

When completed, please send to:

Who's Who Editor, Computers and Automation,
815 Washington St., Newtonville, Mass. 02160

CORRECTION IN "REMOTE BATCH PROCESSING AND OTHER GOOD NEW IDEAS IN THE COMPUTER FIELD," OCTOBER, 1968

I. From Charles S. Hooper, Jr.
Vice President
The First National Bank of Maryland
Baltimore, Md. 21203

I read your interesting editorial ("Remote Batch Processing and Other Good New Ideas in the Computer Field") in the October 1968 issue of *Computers and Automation*. I was surprised to see in your last paragraph that there are 5000 vendors of time-sharing services currently.

In Mr. Hammersmith's article in the same issue, he estimates "5000 companies using time-sharing services currently."

Would your 5000 figure refer to users rather than vendors?

II. From the Editor

Yes, the number "5000" in the last line of my editorial (on page 43 of October, 1968) was in error and you put your finger accurately on my source of error — misreading the sentence in question. Thank you for calling my attention to the error.

I shall try to find a much better estimate of the number of time-sharing vendors, counting local branches, etc., and publish a correction.

(Please turn to page 48)

ANNOUNCEMENT

As of January 1, 1969, Computers and Automation is being published 13 times a year instead of 12 times. The new June issue will have the same kind of editorial content as the other monthly issues; and "The Annual Computer Directory and Buyers' Guide" issue will become a special 13th issue published additionally in June.

The annual subscription rate of Computers and Automation WITH the "Computer Directory and Buyers' Guide" will become $18.50, and WITHOUT the Directory issue the annual rate will become $9.50, for United States subscriptions. The effective date for this rate increase has been extended from February 1, 1969 to March 1, 1969.

ALL OUR SUBSCRIBERS ARE INVITED TO RENEW THEIR PRESENT SUBSCRIPTIONS ON OR BEFORE FEBRUARY 28, 1969, and thereby receive the additional issue at the old rates. This same invitation is extended to all readers who are not yet subscribers.

Since 1960 our subscription rates have been unchanged. But because of the additional issue we will publish, and because of continually increasing costs of producing and publishing the magazine, this price increase has become necessary.

DP FOCUS

INDUSTRY AND COMPANY REPORTS
FOR EXECUTIVES CONCERNED WITH
THE DATA PROCESSING MARKETPLACE

61 Helen Drive
Marlboro, Mass. 01752 U.S.A.
Telephone: 617-485-7510

SPECIAL OFFER — EXPIRES MARCH 15, 1969

DO YOU KNOW ABOUT THE HIDDEN OPPORTUNITIES IN THE COMPUTER FIELD?

HAVE YOU ANY QUESTIONS CONCERNING THE INDUSTRY?

HOW ARE YOU FIXED FOR ANSWERS?

DP FOCUS is an industry and company report published 18 times a year for executives concerned with the Data Processing Marketplace. This unique publication provides you with a digest of information on this dynamic industry. Recent issues have included:

- MARKET RESEARCH
- COMPANY SURVEYS
- NEW SECURITY ISSUES
- NEW PRODUCT LINES
- COMPUTER LEASING
- PERIPHERALS
- TIME SHARING
- EARNINGS FORECAST
- DP DATA COMPUTER INDUSTRY CENSUS
- MARKET TRENDS
- IBM'S COMPUTERS
- SOFTWARE TRENDS
- SUPPLIES
- INTERNATIONAL MARKETS

As a subscriber to DP FOCUS, you will be among the first to receive news and analyses from special correspondents, covering all aspects of the Data Processing field. You will receive reports on DP Organizations, Applications, Investment Opportunities, New Firms, Management Surveys, etc.

We are gratified by the enthusiasm with which our publication has been received by subscribers in a wide variety of occupations. In fact, as we approach our first anniversary, we find that we have become a standard source of information in the field.

We believe that we may have something important to offer to you. Try our SPECIAL OFFER — 3 issues for $6.00. Mail the coupon today and we will send you, as a bonus, a copy of the MANAGEMENT SURVEY ISSUE.

DP FOCUS, 61 Helen Drive, Marlboro, Mass. 01752

Please send the MANAGEMENT SURVEY ISSUE free and enter my subscription as indicated by the enclosed check.

( ) Three issues - $6 ( ) Six Months - $37 ( ) One Year - $63

Name________________ Title________________

Company________________

Street________________

City________ State________ Zip Code________

PURCHASING PACKAGED SOFTWARE— A CUSTOMER'S POINT OF VIEW

Duane Burke, Research Engineer and
Robert Gillespie, Chief, Research and Development
Computing Department
The Boeing Company
P.O. Box 3707
Seattle, Wash. 98124

"It is reasonable to expect that the additional tasks that the customer must perform to get a software package successfully implemented after *the vendor considers it to be 'installed' will cost many times more, in terms of manhours and computer time, than what it cost to get the package in the door."*

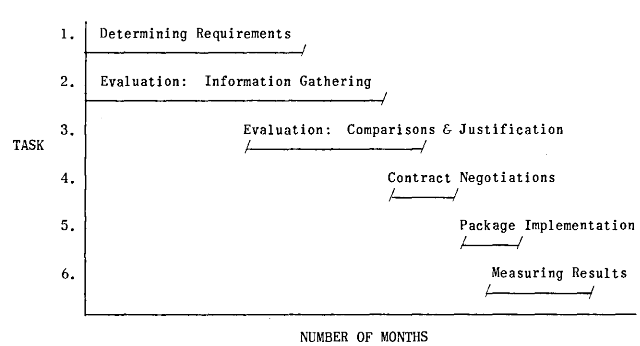
Companies can save both time and money through the use of purchased or leased "packaged programs." And from a customer's point of view, the opportunities for such savings are growing rapidly, as established software companies bring packaged programs of good quality onto the market. These savings occur since a software house is able to distribute the development cost for a package over many customers, and so the package *can* be sold or leased for less than a potential customer would pay for a similar in-house development. In addition, benefits which would accrue from having the package are lost during the time that the package is being developed in-house.

In this article the term "packaged programs" refers to such programs as generalized file handlers, decision table processors, flowcharting software, and computer evaluation software, which are being marketed separate from computers and, in most cases, by companies which specialize in software development and/or marketing.

As with any new area of business, no common body of knowledge yet exists to aid a potential customer in the acquisition of packaged programs. It is, therefore, the object of this article to pass on some useful points and experience formulated from the point of view of one customer of packaged programs. Figure 1 illustrates the phases of acquisition that will be discussed and their relative timing.

Note: The Boeing Company assumes no responsibility in connection with the use of this paper and shall not be liable for any loss or damages arising from such use. There shall be no warranty, express or implied, by The Boeing Company, as to accuracy, function, or otherwise in connection with this paper.

Figure 1. Phases of Acquisition



Determining Requirements

The requirement for a particular software package may not appear to exist until the package is available on the market. This phenomenon is not a myopic condition suffered by computer people, but is simply a matter of economics and priorities. Companies whose primary business is other than computing products or services will often find it difficult to seriously consider internal development of the kinds of generalized software which are currently being marketed. Hence, the requirement often does not get much attention if any.

Once the potential need for an existing package is identified, however, then other requirements associated with its use, if acquired, should be examined. Will the package become an integral part of an application system? or a key tool, which once adopted is almost indispensable in the development and maintenance of application systems? Packages for processing decision tables, for handling files, or for providing shorthand coding capabilities will commonly be in this category. Such packages may be termed "critical" to the functioning of the systems on which they are used.

On the other hand, a software package for producing flowcharts, although desirable from a customer's point of view, may not be critical to the functioning of any application systems, since it is used essentially for documentation and program debugging.

Some other important factors to be considered are:

(1) Hardware
(2) Operating System
(3) Other software which must co-exist and interface with the packaged program
(4) Probable use.

Evaluation: Information Gathering

The information needed for the evaluation will come from three main sources:

(1) The software vendor
(2) Other users of the software being evaluated
(3) Tests of the software especially formulated for evaluating it.

The Software Vendor

The vendor for a package is, of course, the source of most of the information needed in the evaluation. He will usually be able to give a written explanation of the following items for the package being marketed, especially if he is threatened with loss of a sale:

a. Its capabilities and features.
b. Its intended uses.
c. How to use it.
d. Its known limitations. (Usually the vendor treats this very lightly.)
e. The names of other users.
f. The date it could be installed.
g. The steps and time necessary to get it installed.
h. The results of previous tests.
i. The initial and follow-on costs.
j. The maintenance support available.

This information must be supplied in written form to facilitate its review, and to help insure mutual understanding between the customer and vendor on these details, and to express the commitment on the part of the vendor.

The "users manual" which a customer would be expected to rely on, if the package were acquired, must also be obtained. Besides giving the details necessary to prepare tests of the package, the manual may also provide a rough measure of the quality of the software.

The known limitations and errors in a package will commonly be omitted. Detailed (and emotional) discussions with the vendor will bring some of these to light. Other users, however, will provide the best source of information on limitations, errors, and other problems associated with using a package.

The availability date may be a key limiting factor in the choice of packages which can be considered for use.

Typically, the maintenance support provided by the vendor will consist of sending corrections or corrected versions, as necessary, and responding to questions regarding the package.

With regard to maintenance of a software package, this might be a good place to suggest that customers keep their software packages as near the vendor's 'off-the-shelf' version as possible in order to take advantage of the maintenance that the vendor provides and also to improve the chances of getting prompt assistance when problems with the package arise.

Other Users

The experience of other users of a package is a valuable source of information on the limitations, errors, and problems associated with a software package. Although the probability of a customer admitting he made a mistake in acquiring a package is rather remote, most customers will frankly discuss problems they may be having in the use of one. Beside the obvious advantage of helping to insure a correct choice of software, information thus obtained may help insure that the new customer's software package is more trouble-free than earlier editions.

Tests of the Software

There is no substitute for preparing and running your own tests of a software package. And the most satisfactory test site is your own. However, due to the vendor's costs and other considerations, the likelihood of getting much on-site testing during the evaluation phase is not high. This is always a serious limitation since often usage problems, program errors, and shortcomings in documentation, will come to light as a result of running only a few practical cases.

The degree and thoroughness of testing required by the customer depends on how much the package has been used by other customers and on the vendor's responsiveness in correcting errors in his package. The first half-dozen customers for a package should expect to do much more testing, than say, the twentieth customer.

It is relatively easy to identify the features and situations for which tests are desired. Somewhat more difficult and expensive is the job of preparing test data. In some cases the vendor will have a test deck which can be used to test the basic capabilities of a package. In any event tests should be prepared by customer personnel having a variety of backgrounds in order to call forth various points of view on the ease of use, usefulness, and applications for the package.

Evaluation: Comparisons Package vs. Requirements

The process of comparing software packages is an iterative one, with each iteration taking in some different facet of the software or its use. In comparing packages to requirements, the basic questions to be answered are:

(1) Will it satisfactorily accomplish the jobs for which it is being considered?
(2) Will it be available when needed?
(3) Is it cheap enough to justify acquiring it for the jobs for which it is being considered?

One of the best ways to determine if the package will be satisfactory is to arrange that the expected users prepare the tests. Thus realistic tests can be prepared which test the "use procedures" as well as the software package.

Package vs. Package

Comparing packages with each other may be fairly easy if sufficient information has been gathered and sufficient comparisons made with the requirements. When comparing packages on the basis of a list of factors, often one package will clearly emerge as the best choice. The following is a list of factors upon which a choice between packages will depend:

(1) *Cost* — primarily the acquisition and use costs

(2) *Features of the packages*

(3) *Processing speeds*

(4) *Ease of use*

(5) *Status of Package* — The package when marketed may be in a specification stage, testing stage, or in production use. Status of the package is particularly important if the package is being justified based on its use on a particular project (which presumably has schedule commitments to be met).

(6) *Other customers* — The more customers already in existence the more likely it is that the package is well checked out.

(7) *Modification Constraints* — Many software packages are being marketed today in such a way that the customer cannot readily modify the package. At the very least, this can be inconvenient to the customer. In the case of "critical" packages,* this constraint may significantly limit the usefulness of the package, depending on the package and the customers intended use of it.

(8) *Usage constraints* — Some software packages are being marketed for use at only *one installation site*. Packages for additional sites may be obtained at additional cost. Packages for additional company sites can usually be obtained at a reduced rate.

(9) *Service Response on Problems* — One measure of the probable response to servicing problems can be obtained from noting the support provided by the vendor during the evaluation. Service response is particularly important on "critical" packages.

(10) *Financial Arrangement* — There will sometimes be a choice between leasing or buying packages.

(11) *Form of the Delivered Package* — Some packages are being marketed in object module form only. For others, the source deck can be obtained. Having the source deck gives the customer an important and in some cases a necessary degree of flexibility in using and maintaining the package.

(12) *Maintenance* — Some period of maintenance support by the vendor will customarily be included in the price of the package. More comprehensive or longer term maintenance support can usually be obtained at additional cost.

(13) *Implementation Support* — The more time the vendor can remain on site to assist with the implementation of the package and to handle initial training in its use, the smoother the implementation is likely to be. Even small problems become a significant nuisance until some in-house expertness on the package has been acquired.

(14) *User Groups* — The availability of some formal means for inter-user communications can be an asset.

Evaluation: Justification

To determine if the package is economically justified for its intended uses, it is necessary to include all of the potential costs and savings. The major sources of savings will usually be easy to identify even if not so easy to compute. Pinning down all of the costs, however, may take a little more thought. Listed below are some costs which will usually be present:

(1) Purchase or lease price.
(2) Implementation, maintenance, and training costs, including manpower and machine time.
(3) Documentation costs, which include preparation, reproduction, and distribution of supplementary documentation and vendor-supplied documentation.
(4) The day-to-day machine time costs which are incurred as the software package is used for production purposes.

Cost estimates are likely to be far from accurate. The chances of getting good estimates are enhanced, however, by running tests of the package prior to trying to justify it.

Contract Negotiations

Some of the conditions and terms which should be considered in negotiating the contract are as follows:

(1) Avoid restrictions on the use of the package.
(2) Avoid clauses which would require special handling of the reference manuals for the package.
(3) Be prepared to include a liability clause relative to the distribution *not authorized by the vendor* of the software package, especially if delivered in source deck form.
(4) Provide for an acceptance period of at least 30 and preferably 60 days.
(5) Provide for return of the package at no cost if not acceptable.
(6) Make payments depend on fulfillment of contract.
(7) Get statements on responsiveness to correcting errors.
(8) Specify, in detail, any agreed-to modifications to the "off-the-shelf" package.
(9) Avoid modification constraints.
(10) Arrange for reduced price, special considerations, or a chargeback clause if you are one of the first one or two customers. The chance is remote that a package will be completely debugged until after it has been in the field awhile.

Package Implementation

The steps and time necessary to get the package installed, as described by the vendor, will not, in general, be all that the customer must do in order to successfully implement a package. A vendor may consider a package installed after the following things have been accomplished:

(1) Package is running on the customer's specified computers.
(2) Any training course provided for in the contract has been given.
(3) Documentation describing the use of the package has been provided.
(4) An explanation of what to do when a new version of the package is sent to the customer has been supplied.

Beyond these things, however, the customer will commonly find it necessary to do the following:

(1) If the customer has more than one computer on which the package can be run, it will probably be necessary to get it running on these other computers also.

(2) Due to the short time during which the vendor is on site to assist with implementation, it may not be possible to get the package processing at optimum speed. Therefore additional experimentation with I/O devices and channel assignments for the package may be necessary.

(3) Training, beyond that provided for in the contract, will usually be required for persons who could not be included in the first training sessions given by the vendor. Additional training may be purchased from the vendor, or

*As previously defined in the section on Determining Requirements.

c.a

PROBLEM CORNER

Walter Penney, CDP
Problem Editor
Computers and Automation

PROBLEM 692: A STRAIN ON THE BRAINWARE

"Maybe it's economical so far as storage is concerned," Al said, shaking his head, "but it's putting a strain on the brainware."

Bob looked at the sheet Al was working on. "You mean that method you were telling me about — of storing triangular arrays as linear sequences? Where does the strain on the brainware come in?"

"Well, we have to be able to tell where every item is. In the triangular set-up the first row would go from 1 to N, the next row from N + 1 to 2 N − 1, with the N + 1 under the 2, the N + 2 under the 3, and so on."

Bob was beginning to get interested. "Then all the rows were lined up on the right?"

"Yes, right justified as the boss likes to call it."

"Why not just work out the numerical values of the starting points for each row and use those?"

"That would be all right for this array, but the program must be flexible enough to work for any N — actually this program is being written for N up to 100, that is, as many as 5050 entries."

"In other words you're going to have up to 5050 consecutive locations and you want to know where the item that was in I row, J column is?"

"Right! Of course J will always be equal to or greater than I," Al added.

What is the location of the item that was in I row, J column?

Solution to Problem 691: Who Is Missing?

The first column is off by 6; this means a missing F, P or Z. Likewise the second column implies C, M or W is missing, the third, E, O or Y, the fourth, I or S and the fifth, G or Q. The only plausible name that can be read through this block is ZWEIG, probably on the last card, which somehow didn't get included. □

Readers are invited to submit problems (and their solutions) for publication in this column to: Problem Editor, Computers and Automation, 815 Washington St., Newtonville, Mass. 02160.

---

else provision must be made for an in-house training course.

(4) The user's documentation, as provided by the vendor, may adequately describe the details of how to use the product. But it will not usually contain all of the information needed by the customer for its effective use. For example, the job control statements will commonly not be complete, due to variations in equipment configurations that the package can run on.

Some additional points regarding implementation are worth mentioning.

(1) The vendor should remain on-site or near at hand during as much of the test period as possible, in order to speed correction and explanation of problems and results.
(2) It is best to start in a small way the usage of the package, especially if it has not been previously field tested.
(3) Provide adequate local documentation and consultation support for the package, so that those for whom the package is intended will be able to make effective use of it.
(4) Establish a "consultation log" on the use of the package so that data on problems and uses can be accumulated for measuring the results of implementing the package.
(5) Maintenance support must be provided in order to insert revisions which the vendor sends out or which the customer may originate (if he has a source deck).

It is reasonable to expect that these additional tasks which the customer must do, to get the package successfully implemented, will cost many times more, in terms of manhours and computer time, than what it took to get the package in the door.

Measuring Results

Measuring the results is an important, though imprecise, aspect of implementing a packaged program. *Tangible* results, based on "before and after" information of the type listed below, can usually be obtained, but not without some cost and difficulty:

(1) Manhours, machine time, and flow time expenditures.
(2) Characteristics of applications of the packaged program.
(3) Training and documentation requirements.

Ideally, this information will be obtained with little effort from those (programmers or others) who are expected to use the new program.

Some of the required information can be obtained (at some cost in machine time) by executing routines for gathering statistics each time the software package is executed. The statistics may be printed as part of the results which are returned to the programmer, or better, may be accumulated on a file for later analysis by another program.

The remainder of the required information can usually be obtained from personal interviews with management, programming, and other staff personnel.

Although large savings are possible through the use of packaged programs, considerable effort and work are required in order to realize the savings. Furthermore, until the job of measuring the savings is complete, no accurate judgement of the real worth of the package can be made.

References

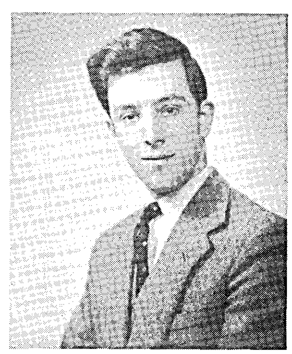
1. *Datamation,* April 1968, "The Problems of Packaged Programs: The American Management Association Conference Report."
2. *Data Processing Digest,* April 14, 1968, Volume 14, Number 4, "Effective Program Development: The Choices, The Packaged Program," by Robert V. Head, Software Resources Corporation. □

SEGMENTED-LEVEL PROGRAMMING

"Because program modules are genuinely independent, production of a single program can be planned and controlled; because the interfaces are rigorously standardised, modules are interchangeable; because program design is a meaningful activity, the quality and performance of programs is better; because of the particular techniques employed, generalised procedures can be used at any level in the program structure."

Michael Jackson
Technical Director
Hoskyns Systems Research Ltd.
Boundary House
Furnival St.
London EC4, England

Anthony B. Swanwick
Vice President, Marketing
Hoskyns Systems Research Inc.
61 Broadway
New York, N.Y. 10006



Michael Jackson (left) is Technical Director of Hoskyns Systems Research Ltd., the software company of the Hoskyns management consultancy group. Before joining Hoskyns in 1964 he was computer adviser to the Maxwell Stamp organisation. He majored in classics at Oxford and mathematics at Cambridge, England.

Anthony B. Swanwick (right) is Vice-Pres., Marketing, of Hoskyns Systems Research Inc., the U.S. subsidiary of the Hoskyns group. His previous experience includes the marketing of an on-line business data processing service and systems analysis with a computer manufacturer. He is an honours graduate in Law.

Segmented-level programming is a particular technique of modular programming, supported by a software package. The software has been implemented for a number of machines and languages, including IBM System/360 Assembler Language. In this article we shall be concerned with establishing the basic approach which led to the development of segmented level programming (SLP), and to describe in outline the SLP techniques and concepts.

Why Modular Programming?

Modular programming is not a new idea; nor are the problems new which it sets out to solve. Programming for business data processing is absurdly difficult and expensive in every way, and the resulting programs are of a dismally low quality. Estimates of project cost and completion date are often little better than guesses; measurement of progress on a single program is almost meaningless; the cost of reallocating partly completed work is intolerable; program testing is a casual affair, in which an arbitrary few cases are tested out of the millions of possible cases; production programs continue to display new errors month after month; programmers continue to spend their own time and their employers' money on solving over and over again the same problems that are being solved in a thousand other installations across the country. Modular programming is generally intended and supposed to offer a solution to these problems.

And it can offer a solution, provided that we recognize that it is not in itself the solution; it is simply a fresh problem. A programmer struggling with all the usual difficulties of writing a large non-modular program is not helped by the suggestion that he ought to break it down into a number of modules. That would only introduce new difficulties of designing interfaces, coordinating the module design, writing special testing programs, and fitting the modules together to make the final product; and his present difficulties are pretty bad, and there doesn't seem any reason to add to them. So why modular programming?

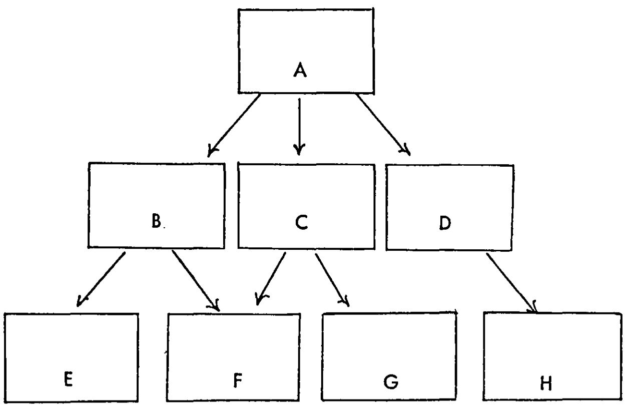


Figure 1. Program Structure

Because the problems modular programming poses are technical problems which can be solved at a general level, once and for all; and when we have solved them we can use the resulting techniques and facilities to solve the more primitive difficulties of programming. We need to develop a detailed technology for modular programming which will allow us to obtain the benefits that modular programming is supposed to give.

An Engineering Analogy

An engineering analogy is instructive. A modern computer can only be constructed on a modular principle; no other method would allow so complex a device to be built successfully. But it is not enough to decide on modularity; the precise details of the modularity must be settled also. The logic will be built on circuit cards of precisely this pattern, each with precisely this number of terminal pins, arranged in racks of precisely these dimensions. Only by developing this technology at the outset can the design of the computer itself proceed unhampered. In exactly the same way, the technology for modular programming is a prerequisite to the practice.

SLP provides a detailed, fully worked-out technology. SLP programs are composed of closed subroutines arranged in a hierarchical structure; the interface between subroutines is standardised, and the subroutines themselves are in a standard format.

An SLP subroutine is here called a *procedure*. A procedure would normally contain between ten and one hundred executable machine instructions. The size of any particular procedure is determined by the requirement that a procedure should contain the coding corresponding to a single coherent sub-problem of the program specification. That coding will in general contain calls to procedures at a lower level, which in turn contain the coding corresponding to sub-sub-problems, and so on.

The interface between procedures is standardised; in an Assembler Language implementation it is incorporated in a set of macros which handle all problems of register usage. The standard interface allows up to five parameters to be specified in a call. There are two versions of the standard interface, differing in the coding generated in the called procedure: the high-speed version does not store and restore the registers for the calling procedure; a procedure equipped with this high-speed interface cannot therefore call other procedures in its turn, and is known, for obvious reasons, as an inner procedure.

A simple example of an SLP program structure is shown in Figure 1. Procedure A, at the top level, is called by the *job control* part of the operating system, and returns to job control when it has completed execution. Execution of procedure A includes both ordinary instructions and calls to procedures B, C, and D; execution of procedure B includes both ordinary instructions and calls to procedures E and F; and so on. A *procedure call* is closely analogous to an ordinary instruction, and the analogy is important. The name of the called procedure corresponds to the operation code, and the parameters of the call correspond to the operands; execution of a procedure is like execution of a complicated machine instruction.

Storage Allocation

In general, procedures are strictly 'read-only'; that is, no part of the core storage occupied by the procedure is modified during program execution. Except for a special case concerned with file definitions, all working storage is defined in the top-level procedure of a program, and is acquired dynamically by procedures at lower levels. The working storage is organised in a *push-down stack*. The stack is set up in its initial state when the top-level procedure is loaded, and this procedure may then make use of storage at the top of the stack. The amount of storage that may be used is limited only by the total size of the stack. When a procedure at the next level is called, the stack is notionally "pushed-down" to make unused storage available to the called procedure. The distance the stack is pushed down depends upon the needs of the calling procedure. The programmer will have described its working storage by SLP macro-instructions, and will have specified that some parts are to be retained while others are temporary; the generated coding ensures that the stack will be pushed down far enough to save the parts to be retained, while the temporary storage will be overwritten. Every time a procedure is called the stack is pushed down in this way; every time control returns to the called procedure, the stack is "popped up." The process is illustrated in Figure 2, using the program structure shown in Figure 1.

There are several advantages to the push-down stack scheme of storage allocation. For the batch-processing installation with limited core space, the most important advantage is that core requirement for working storage is minimised. In the program shown in Figure 1, suppose that Procedure A uses 1,000 bytes, that B, C, D, E and H each use 500 bytes, and F and G each use 250 bytes. Then the stack size required is 2,000 bytes (the maximum reached during the execution of E and H) instead of a total of 4,000 (the sum of all the individual requirements). This economy is achieved without the careful planning that would be needed for a conventional scheme of common working storage.

Test-bed Program

SLP procedures are tested on a *test-bed program*. This program is equipped with the calling side of the standard interface, and provides facilities for symbolic definition of test data. It controls execution of the procedure under test, captures program-check interrupts, and prints results selectively according to the test data accepted. In a single execution of the test-bed program, several procedures may be tested either together or separately, and several tests may be executed on each.

Principles of Program Testing

Design of this software is based on certain explicit principles of program testing. The first and most important of these is that testing at the program level is never adequate. A typical commercial program may handle data cases in which there are fifty significant binary conditions; there are then 2^{50} potentially significant data cases. It is entirely impractical to test more than a few of these; the normal technique is to rely on the program design to ensure that most of the conditions are not significant in combination, and to test some of those combinations that remain.

However, experience shows that this technique does not work: programs may run for months, or even years, and still be capable of producing surprising errors. It is accordingly necessary to test the program design at every level; in the context of an SLP program, this requirement means that each procedure must be individually tested. Once a procedure at the bottom level has been tested, a call of that procedure assumes the character of a machine instruction. In testing a procedure containing that call, the bottom-level

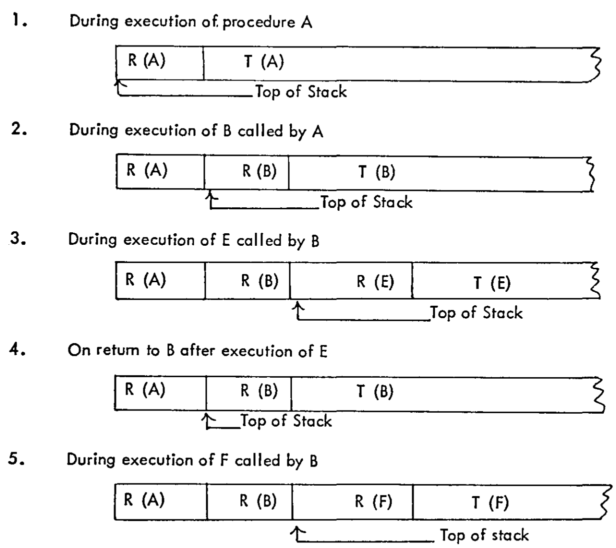


Figure 2. Allocation of Working Storage
R(X) = Retained storage for procedure X
T(X) = Temporary storage for procedure X

The program structure is shown in Figure 1.

conditions are no longer significant; the operation of the called procedure now becomes reliable, and it is enough to test that the call itself is correctly coded, and that the results are correctly handled by the calling procedure.

The full facilities of a regular SLP implementation are much more comprehensive than the skeleton described above. But the fundamental advantages remain the same. Because the programming environment is technically disciplined, programmers are freed from the distractions and problems of inventing their own basic techniques, and can focus their efforts on the problems they are required to solve.

Program Design

In particular, the work of program design springs into prominence. A programmer using a monolithic, non-modular technique is unable to design his programs: if a program has no parts, it can have no structure; if there is no structure there is nothing to design. A programmer using an ill-defined and undisciplined modular technique is only slightly better served: he is aware of the need to dissect his problem, but there is too little constraint on the shape and size of the resulting parts; he has to think out for himself the elementary principles which can tell him whether his dissection is the right one.

But a programmer using SLP is faced with a well-defined task in program design. He knows what a valid procedure must look like; he knows in detail how it must fit into the program structure and how it must interface with other procedures; he knows how the working storage is to be allocated, and the precise effects of allocating a particular part of it in one procedure rather than another; he knows how he must test his procedures, and he knows what will make that testing easy. Against this background, the work of program design is the expression of a problem solution in well-defined terms. The concepts and the constraints of SLP provide a language in which the designer can think and can express the results of his thinking.

Consider, for example, the relationships between the problem, the data, and the program. Program design is largely a matter of getting these relationships right. In an SLP program, the workstack mechanism constrains the possible relationships between a procedure and the data it operates on. The data may be compiled as part of the procedure coding: then it must be a local constant. It may be compiled as part of another procedure: then it must be a local constant of that other procedure and an input argument of this one. It may be generated at run time in the workspace of a higher-level procedure, and its address passed to this procedure as a parameter: then it must be an input argument or an output result of this procedure. It may be generated at run time in the retained workspace of this procedure: then it is an intermediate result of this procedure. And so on. The point is that the programmer doesn't have to think out these categories for himself every time he sits down to design a program: he can think about the data/procedure relationship in terms of workstack allocation; the soundness of the technology guarantees that this shorter and surer way of thinking is itself sound.

The Advantages of Standardized Techniques

Against this background it becomes possible to develop articulate principles of programming, and to educate trainee programmers to a standard that was previously thought attainable only by long experience or unusual natural ability. Ultimately, this is the most important of the advantages of SLP. Without a developed technology of programming, each new generation of programmers starts from where its predecessors started: the operating systems and the compilers are more elaborate (and sometimes more reliable and effective), but basic understanding of program structure starts again from scratch. What principles can the trainee be taught that will guide him in program design? What criteria can he be given that will distinguish a good program from a bad? None. Without a developed technology the framework is missing; there is no language in which lessons learnt can be readily communicated. SLP technology provides the framework and the language.

Often, programmers have rebelled against standardised techniques, seeing in them nothing beyond a mindless regimentation imposed from above: any benefits were to be reaped by management: the disadvantages and discomforts were the programmers' lot. Too often, this rejection has been justified: standarised techniques are not an end in themselves; they must represent and embody a technology that is better than the programmer could devise for himself; they must amplify his skills, not cripple them. Our own experience in Hoskyns Systems Research has shown us that programmers, whether trainees or skilled veterans, welcome the opportunity to develop their capabilities by using a valid modular technology; far from rebelling against the disciplines it imposes, they are constantly seeking to extend its scope.

Two practical examples of the use of SLP are given below. The first example (Figure 3) shows the hierarchical structure of a complete program. A simple program has here been selected for reasons of clarity. The actual block diagram in practice would be laid out in a more formal style.

The program in Figure 3 is a simple update. Customer records consist of customer number, name and address, fixed status codes and some transaction trailer items. New records may be inserted and existing records amended. The top level procedure, L, contains the coding for reading master and detail files, comparing records on customer number, writing new and unchanged records to the new master file, and for calling procedures M, N, P and Q appropriately. Procedure W is called by procedures S and P. Its function is to convert the customer number, name and address from their fixed input formats variable-length file format — its parameters would include core addresses of the fields in input format and of the areas (probably in output buffers) to receive the converted fields. Procedure T is called by procedure P several times for one customer if there are several trailer items to be amended for that customer.

Note in particular how the function of any part of the program is easily understood at any level. If the program requires amendment the appropriate part is immediately identifiable and the effect of the change on related modules is obvious; the risk of the amendment causing failures elsewhere in the program is eliminated.

The second example is taken from a software package for IBM System/360 called *Formatter*. Formatter is designed to enable programmers to specify and amend print layouts outside their problem programs. It also provides facilities for writing records for several different layouts on to a single file from one program.

Formatter consists of three programs; the first, FORMALIB, creates a small library on the disk for holding format specifications; the second, FORMSPEC, handles insertions to the library; the third, FORMATTR, reads the print file and operates on it in accordance with the appropriate layout specified on the library. The three diagrams in Figure 4 demonstrate the use of identical procedures in different programs.

The modules shown in Figure 4 are:

MAKO10: This initialises the disk library by performing a number of consecutive "write" operations; because of other considerations, a direct-access technique must here be used.

MAKO15: This reads or writes a library record, according to the setting of a parameter switch. The module is also supplied with the record number and the start address of the library.

FDA18F and FDA18G: These modules handle IOCS (the input/output control system). FDA18F reads data from a disk record or writes new data to a disk record. FDA18G creates a new record. Both modules were developed to improve on IBM IOCS facilities.

MAMO39: This is a general-purpose module for adding a number of tracks to an IBM 2311 disk address to form a new address. MAMO39 is also used in other programs, and a version exists for performing the same function for an IBM 2314.

The problem examined is that of creating, updating and reading a direct access library. This file consists of fixed-length records held on one "extent" of an IBM 2311 disk; each track contains two records, and each record contains a print layout specification in coded form for a given format number.

FORMALIB initialises the disk library by writing a predetermined record at every location in the library.

FORMSPEC updates the disk library from cards; any records may be updated in any sequence, according to the record numbers punched into the cards.

FORMATTR prints a sequential file according to the print layout specification contained in one of the library records; a field in the print file determines which record is required.

Note that FDA18F is used in FORMSPEC to write records, and in FORMATTR to read them, depending on a parameter passed to it by the calling procedure, MAKO15. This calling procedure has been instructed accordingly by its calling procedures.

Recently the package was amended to read print files from IBM 2314 disks as well as IBM 2311 disks and 2400 tapes, and to create the format library on 2314 disk as well, at three records per track (instead of two for the 2311 disk). This proved to be an easy task as the relevant modules could be unplugged and new ones plugged in.

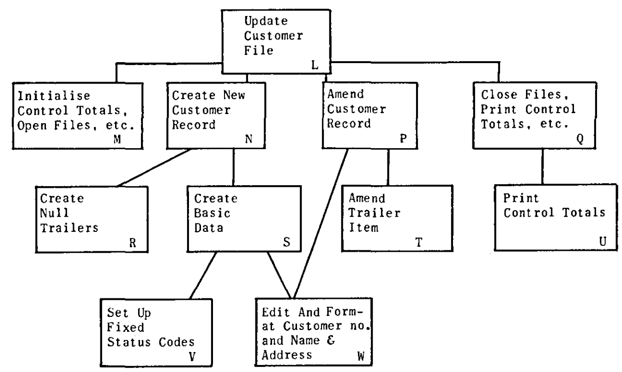


Figure 3. A Simple Commercial Program

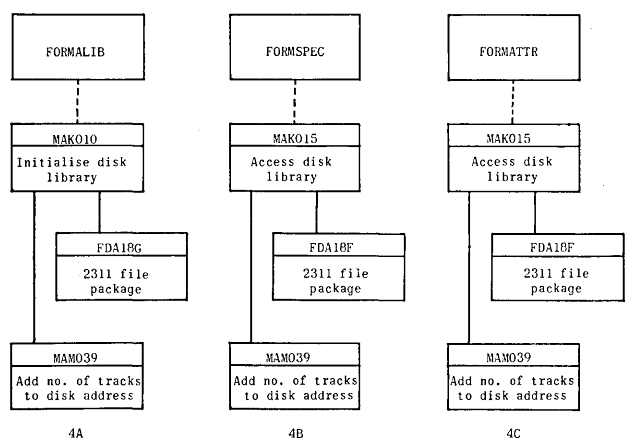


Figure 4. An example of reusing procedures in different parts of a software package.

Summary

The overall benefits of SLP technology are large. Because program modules are genuinely independent, production of a single program can be planned and controlled; because the interfaces are rigorously standardised, modules are interchangeable; because program design is a meaningful activity, the quality and performance of programs is better; because of the particular techniques employed, generalised procedures can be used at any level in the program structure. All of these benefits depend on the detailed technology. Modular programming makes the promises; SLP delivers. ☐

FR-80 is for certain birds

People close to computers think they see the graphic output problems very well. To render an engineering drawing from a digital tape, they might recommend a Stromberg 4060. For charts and graphs from tape, a Calcomp 835 is an excellent choice. 3M's EBR is suggested for forms generation. An E-K KOM-90 qualifies for outputs of personnel records.

But a man somewhat above the action can see that all of these things might be done by one machine. Instead of four trained operators, he'd need only one. Instead of four maintenance contracts, one. Instead of four rooms and four sets of supplies, one.

Instead of four partly idle, special purpose systems, a busy all-purpose one.

If you are that man, congratulations. Here's how you sell your department managers on a centralized film-recording system.

Your chief engineer: Tell him his drawings will be of graphic arts quality — sharper than any method short of re-drawing, sharp enough to stand up through multiple reproductions at E-size.

Your marketing manager: Tell him his charts, graphs and tables can be in any format, with any symbols, any line weights, and of a quality perfectly suitable for use in printed proposals and reports.

Your personnel manager: Tell him he will be able to do selective, computer-processed listings on the same pass with the film recording. Much faster than two-step processing and sharp enough for multiple reproduction.

Your production manager: Tell him his parts lists will be keyed to any desired scheme, placed in any format of your choosing, selectively ordered, rapidly generated, and just as sharp as the engineering drawings.

FR-80 is the first system to combine a fast computer and the latest developments in precision CRT imaging on 16 or 35 mm film. No other electro-optical system can touch its resolving power (80 line-pairs per millimeter) and none is so versatile.

FR-80 will accept any tape format and generate any graphic output.

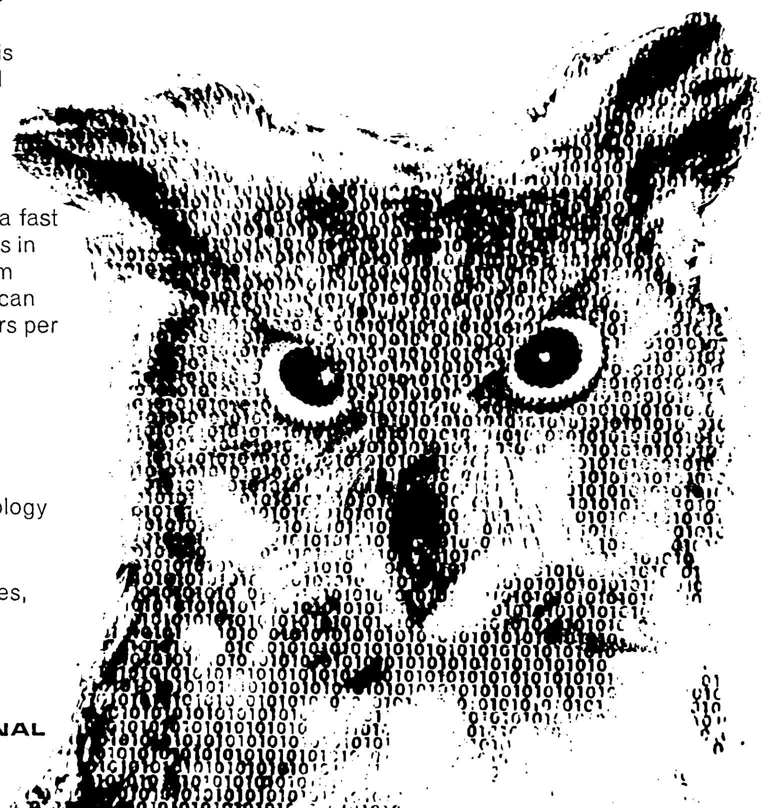
Write. Even a wise man needs more information.

Information International, 545 Technology Square, Cambridge, Mass. 02139, (617) 868-9810;

1161 West Pico Boulevard, Los Angeles, Calif. 90064, (213) 478-2571.

INFORMATION INTERNATIONAL

Designate No. 25 on Reader Service Card



THE DESIGN OF COMPUTER LANGUAGES AND SOFTWARE SYSTEMS:

A Basic Approach

Richard K. Bennett, E.E.
Computer Horizons Inc.
Conant Road
Lincoln, Mass. 01773

The Current State of Computer Languages and Software

An issue of the *Communications of the ACM* some years ago sported on its cover the drawing of a huge stone tower under construction. Although obviously only a portion of the tower had been finished, it nevertheless already reached into the clouds. The building blocks of this tower were labeled with names such as Fortran, Cobol, Sap, Mad, Mystic, Goofus, and Mishap. At the base of this tower appeared the inscription "BABEL 1960." That was nine years ago — and how the tower has grown since!

The Tower of Babel problem in computer languages refers to the confusion arising from differences and incompatibilities among the hundreds of computer languages which have been created to meet many diverse needs in many areas of computer applications. The language differences have made the interchange of programs among the different languages at best difficult, and often almost impossible. At the same time, the differences have required a multiplicity of different compilers and assemblers; and this has made software systems expensive, cumbersome, incomplete and sometimes ineffective. The total cost of unnecessary computer language differences may well be on the order of hundreds of millions of dollars annually.

Is there no solution to the problem of the ever-growing Tower of Babel? If no solution has yet appeared, it is not the case that it has not been sought. Many millions of dollars and thousands of man-years have been spent towards solving this problem. Some newcomers to the computer field, coming from more disciplined sciences, wonder if the problem can really be all that difficult. But the design of computer languages and their compilers is often so shrouded in mystery that only a few "initiated fraternity members" are thought to be qualified for computer language design.

Research in computer languages has often been concentrated on linguistic sophistications, theoretical systems, automata theory, etc., which are of academic interest rather than of practical value. Meanwhile, practical work in the development of computer languages, their translators, and software systems in general, has often suffered from the lack of theoretical basis, or even an orderly approach. Part of this problem has been discussed by L. Constantine.[1]

I believe that there is a solution to the problem of the Tower of Babel growth in the complexity and multiplicity of computer languages and software. I think this solution lies in the direction of simplification and unification. The validity of this approach has been suspected by many and glimpsed by some. In particular, Mark Halpern has proposed a unifying approach to the design of computer languages by the use of macro processors.[2,3]

Specifically, I propose that we must:

1. Simplify the syntax of our languages.
2. Integrate the design of languages with the design of their translators.
3. Establish a systematic procedure for defining computer languages.

The implications of this proposal will now be elaborated.

Language Elements

A careful examination of many computer languages reveals that the important features of existing languages are concentrated in a small set of simple elements. To simplify the syntax of languages and to unify their design, we must first isolate these basic elements.

Characters

First, all computer languages use *characters* as their most basic unit. Let us call the string of characters which the user generates the *message string*. Thus, a computer program would consist of a message string; and the communication from a user to the machine on an interactive console would consist of a sequence of message strings, interspersed with messages from the computer.

Words

Second, elementary groupings of these characters occur. We may call these *words*, because these groupings take on the attributes of words in English: that is, these groupings represent entities which have meanings independent of their constituent characters.

We must be careful not to prejudice the meanings of our words at the outset, by classifying them according to the type of characters they contain. All of the following are words (in one or more computer languages):

"The total cost of unnecessary computer language differences may well be on the order of hundreds of millions of dollars annually. Isn't it possible that by isolating a set of basic elements of computer language and software systems, and using these elements to create a general language base and a general software base, this waste can be eliminated?"

READ	(
WRITE	,
+	ADD
**	)
FOR	IF

Note that some of these words consist of only a single character; others have two or more. Some are composed of alphabetic characters; others, of punctuation or special characters.

To retain the basic nature of words, let's keep separate the process of word delimitation from the process of assigning word meanings. Then we will not have prejudiced their use. For example, in one language we might want English word operators, like PLUS, while in another, we might want to use a symbol such as +, for the same purpose.

Word Groupings

Third, all languages frequently group words together as larger entities. The method of their grouping varies from purpose to purpose, and from language to language. All of the following are word-groupings in one or more computer languages:

- a single argument (parameter)
- an argument list (a parameter list)
- a quotation (for printing a comment)
- a statement
- a phrase
- a part of a program
- a whole program

We should add word-grouping as the third in our list of elements.

We might include a method of indicating the word grouping in the message string. Parenthesis marks have the advantage of providing a set of matched open and closed marks, and are familiar to Fortran and Algol programmers. To satisfy Cobol devotees, we may need later to devise a more implicit method of establishing word-grouping.

Lists

Fourth, there is a special case of word-grouping that needs special consideration and separate handling: this is the *list*. Lists are found in one form or another in most computer languages; and there are languages which are totally built on lists.

Statements

Fifth, we find, as an element, in all computer languages the *statement* (or command) form. For example, in assembly languages we find:

- instruction statements, such as "ADD X",
- pseudo-instruction statements, such as "BSS 100" (which leaves room for a block of storage words), and
- data statements, such as "DEC 3.1415926" (which puts in decimal data).

In higher-level languages, we find:

- input/output statements, such as "READ (5, 7)";
- type declaration statements, such as "INTEGER X";
- conditional statements, such as "IF (A GR B) GOTO Q", and
- algebraic assignment statements, such as "A = B + C * D".

Thus, we see that statements appear in many forms — in fact, too many forms. Let's isolate the basic forms.

First, we find what has been called the *prefix* form, where the action word of the statement *precedes* the parameters which make that action specific. This form covers almost all statement types. The exception, we find, is in the so-called assignment statement of Fortran or Algol, where the statement appears in the form of an algebraic equation. In this case, the action word (or *operator,* as it may be called) does not appear first. As a matter of fact, if we look at a typical Fortran equation we find many operators. To clarify the situation, we need to look at the operators one at a time. For example, take the statement: A = B + C * D. The principal operator here is the equal sign (=), which assigns the variable to its left as the value of the expression to its right. The operator here is *in-between* its parameters (or operands, as they are often called in this situation), and therefore the form is called the *in-fix* form. The other operators in this example, plus (+) and multiply (*), are also of in-fix form.

For completeness, it should be mentioned that a third form exists, the *suffix* form, where the parameters appear first and the operator (or action word) appears last.

In the example "A = B + C * D", where many operators are combined in one statement, we see an illustration of what has been called *recursion,* that is, a condition where a parameter may be replaced by a whole statement (or operation).

This type of replacement can be repeated within the statement (operation) which replaced the parameter in the first place, and the replacement can be repeated indefinitely down to any desired level. Thus, recursion looks essential — at least for some purposes and let's include it in our outline.

Five Basic Elements of Languages

Thus, we have isolated five basic elements of languages:

1. Characters
2. Words
3. Word-groupings
4. Lists
5. Statements
   a. Prefix form
   b. In-fix form

In addition we have noticed that recursion (recursive nesting of statements) is a basic capability required in at least some languages.

With these language elements we now have the basis for building computer languages of all sorts and purposes. We have the start of what might be called a *language base.*

Procedures for Describing a Language

We now need a set of specific procedures for describing a language in terms of these elements. The development of a set of such procedures was the essence of a research project supported by the Air Force Office of Scientific Research. The final report of this project[4] contains a detailed description of such a set of procedures constituting a proposed language base.

A Proposed Language Base

An outline of that proposed language base is presented here.

1. Characters

The first step in defining a computer language is the establishment of the usable character set. The proposed language base contains procedures for introducing a character set, adding new characters, deleting characters, and changing the information about characters. Each character in the set is assigned one or more character classes, by the person defining the language, for the purposes of word delimitation. The principle is that words will be formed of the same class or related classes. Characters of classes different from those of the characters forming a word will act as delimiters for that word. To retain generality, all characters are treated symmetrically, rather than being divided into "delimiters" and "non-delimiters."

There is, however, one special class of characters for which characters always result in single-character words and never combine with other characters or themselves to form multi-character words. For example, the comma (,) would generally be assigned this class. In addition, there is a class for characters which do not form words, but which delimit any word being formed. The space (or blank) character would normally be assigned to this class. In addition other characters might be assigned this class, such as the tab (if available), the carriage return (or end of line), and the line feed.

Introducing New Characters

The proposed language base contains a procedure for introducing a new character, or for changing the classification of an existing character. To illustrate for our purposes here, let us name an operator CC (for "Classify Character"). Then for example, to introduce the character " ¢ ", and to assign it to the class SPL (for "Special," a class assumed to be already available), the language definer would state:

CC (¢, SPL)

The proposed language base also contains a method for adding a new character class to the set of existing character classes. Here we name the operator ADCC (Add Character Class). Then, if the language designer wished to add a new character class called PUNCT and then add the character " ; " with the class assignment of this new class, he would state:

ADCC (PUNCT)
CC (; , PUNCT)

In this way, the language definer could indicate the characters which would be legal in his language and how they would behave in the formation of words.

2. Words

In the establishment of the character set for a language, the character class assignments will have already determined the rules for the formation of words in a new language. However, the meanings of the various words to be used remain to be established. To retain generality and to provide for extendibility, the proposed language base makes no fundamental distinction between words the *language designer* defines and the words which the *ultimate user* defines (when he is using the language in, say, writing a program). Of course, the language designer can restrict the user in the types of words the user may define (by removing the pertinent definitional operators before delivering the language to the user). But at least in principle the word-defining process is treated uniformly.

The words in computer languages are employed for many different purposes, such as names of variables, subroutines, operators, macros, quote marks, etc. Accordingly, the proposed language base provides for first classifying a word by its *type.* Thus, in defining a new word, its type must first be given (or implied). Fortran programmers are familiar with this concept, for they "declare" "INTEGER" or "REAL" to specify the type of variable they are dealing with, and then they give a list of variable names, which are to be assigned that type.

An important generality should be noted here: the definer, whether he is the language designer or the user, has the freedom to assign any *type* to a new word, *independent* of the *class* of its constituent characters.

3. Action Operators

One type of word is of particular interest because it serves to unify a number of similar functions which are handled differently in different languages. Let us name this type *action operator,* because a word of this type causes an immediate action to take place when it appears in the message string. All of the verbs (statement types) in higher-level languages are action operators. The pseudo-operations (commands to the assembler) of assembly languages are also action operators.

Perhaps more than any other word type, the action operator serves as a unifying concept between the design of computer languages and the design of translators. On the one hand, the action operator represents the action intended by the user. On the other hand, the action operator represents the action taken by the translator (compiler) in translating the statement into the action desired by the user. For example, the IF statement of Algol represents the action of checking a conditional expression and the execution of the following statement if the expression is "true" and skipping it, if the expression is "false." To the translator, IF is a word in its Dictionary, which is labelled of type action operator. The Dictionary also contains the location of a subroutine which handles the translation of the user's IF statement.

Thus, the language designer declares the type of the word IF to be "action operator" and he specifies the algorithm for translating an IF statement. This algorithm becomes the subroutine which later translates IF statements occurring in the user's program.

4. Word Groups

The grouping of words for various purposes is a process which is akin to quotation. Therefore, such groups of words will here be named *quotes*. The proposed language base provides several methods for marking and handling quotes. One of the simplest and most general methods is to use the open and closed parentheses as matched quote marks. This method permits unlimited recursive nesting of statement forms. In a more restricted environment, where recursive nesting is not required, alternate rules can be established, which do not employ the parentheses or other such quoting marks.

5. Lists

Lists play a vital role in most languages and provide a natural and powerful method for describing the properties of languages themselves. Lists are used in specifying the parameters of a given action operator, in specifying the variables to be printed, in giving the formal names of arguments in defining a subroutine, etc.

The proposed language base provides for lists in a variety of forms, with properties adequate for the description of languages, as well as for inclusion in the object languages. The proposed language base provides for six forms of lists as they appear in the message string, including fixed-length lists and variable-length lists. In addition, certain features have been included which are valuable for parameter lists, such as the ability to refer to elements of lists by name.

6. Statement Forms

The proposed language-base also provides for prefix and in-fix statement forms. Other forms may be added by the language designer, if he wishes.

The prefix form normally utilizes the parenthesis quote form for the parameter lists. Thus the action word would normally be followed by the parameter list, which would be enclosed in parentheses. This notation is the same as the usual mathematical notation for a function and its variables, and the notation has some useful properties. The parameter list is a quote form.

Properties of the Resulting Languages

Languages built on a language base of the sort just described would possess some very valuable properties.

First, the languages would possess a unifying structure, so that they would present a unified appearance to the user. The languages would differ primarily in their vocabulary and their functions, rather than in style and syntactic differences. Where style and syntactic differences did occur, these differences would be handled systematically. For example, different assignments of character class would result in different rules for word delimitation. However, a user who was familiar with one language would understand the general rules of word delimitation and would therefore need to know only the differences of character class assignment from that of the language which he knew. Ordinarily, these differences would only affect the special characters and would represent a relatively minor change from one language to another.

Thus, the user would find himself at home in any language, needing only to learn the new functions, their names, and their calling procedures. This learning process then would be comparable to that required when a Fortran programmer, say, obtains a new subroutine package from a library or sharing group. In that case, he must learn the functions of the subroutines of the package, their names and calling procedures. In contrast, today's languages require learning a great deal of additional detail about the syntax of the various statement types and about the special rules of word delimitation, etc.

Second, the user would be able to interchange languages and parts of languages. For example, he could use the list facilities of one language with the algebraic statement capabilities of another.

Third, languages would be easily extendible. The extension of a language would be merely the continuation of the definition process. The extension could be special for a single program, or it could apply to a large group of programs. Although it would normally be best for a systems programmer to define language extensions, many extensions would be within the capability of a good programmer. The proposed language base includes a method for adding new variable types and new operators, for example. This procedure could be described in a way which could be comprehended by a good Fortran programmer.

Fourth, all languages built on the language base could be translated by one simple, general processor. Once such a processor is available on all computers, then programmers would be able to write in any (such) language for any computer. Furthermore, any special language extensions or modifications would be accepted by any computer. (Of course, if the extensions include machine-language coding for a specific machine, then the machine-language portions would have to be recoded for each machine of interest.)

A Software Base

In our investigation of computer languages, we determined a small set of basic elements upon which languages are built. If we now look at the translators of these languages, that is, the assemblers and compilers, we find that they too, are constructed from a small set of basic elements. Of course, just as in computer languages, the basic elements are often obscured by more complex structures. But if we look carefully, we can see that there is really only a small set of essentially different mechanisms and facilities.

It is not important here to identify all of the mechanisms, but it is worth mentioning that they include the following:

- character-string manipulation,
- partial-word manipulation,
- symbol manipulation,
- dictionary accessing,
- the capability to call subroutines recursively,
- evaluation of expressions,
- list processing.

By carefully providing these basic mechanisms and facilities, we in effect establish a general software base. This base contains all of the basic mechanisms that we identify, and machine instructions as well — and new mechanisms can be added as they are discovered. Thus, all necessary software may be built on this software base. In particular, we may use this software base to construct a general processor which will serve as the translator for languages built on the proposed language base.

This general processor would be equipped to handle the definitional operators of the language base. In the process of defining a language, the vocabulary of the language, together with the algorithms for its translation, would be supplied to the processor, in much the same manner as a program is now constructed by writing subroutines. The processor would add the vocabulary to its dictionary and translate the algorithms into subroutines. These subroutines would be executed when the corresponding statement names (IF, DO, etc.) appeared in the message string, thus effecting the translation of users' programs.

The Translator

Thus, the general processor would itself be the translator for all of the languages built on the language base. The translation of a language would be controlled by a set of subroutines corresponding to the verbs (or functions) of the language. There subroutines could be stored in a library.

In this approach, the translator is in effect a by-product of the process of language definition. This process is quite different from specifying a language for a syntax-directed compiler. The reason is that in our system we have essentially eliminated syntactic questions and problems, in much the same way — and incidentally with much the same justification and motivation — as is done with subroutine calls. (When a Fortran user writes a set of subroutines, he does not have to worry about the syntax of their calls, because the syntax has already been established by the Fortran system. Since the names of his subroutines express the actions they perform, these names constitute the vocabulary of what may be considered as a new language.)

Run Time and Translate Time

The language designer using this approach performs the same functions in the design of a language as the Fortran user performs in writing subroutines. The language designer names the actions in his language and provides the subroutines for their translation. The principal difference between the two is the *time* at which these actions take place. In the case of the Fortran programmer, the actions take place at *run time,* that is, at the time his program is in execution on the computer. In the case of the language designer, the actions take place at *translate time,* that is, at the time a program written in the new language is being translated.

The difference between translate-time actions and run-time actions may appear to require different syntactic structures. However, this appearance is due to the fact that existing user-languages generally have many complex syntactic structures for translate-time actions, but a single simple syntax for run-time actions (i.e., subroutine calls). We have become used to this arrangement and have thus come to assume that the complex syntax of the translate-time actions is natural and necessary for adequately rich languages for human beings. Very few researchers have even questioned this assumption. As a matter of fact the assumption is usually taken for granted in articles on the subject and not even discussed.

Syntactic Complexities

Actually, the syntactic complexities in existing languages are in my opinion not necessary to human beings and in many instances they are not even helpful. Furthermore, these complexities form the basis of the Tower of Babel problem and at the same time produce an obscuring element which misdirects our efforts towards dealing with the problem.

The complexities referred to include the syntactic differences between statements within a given language and between languages. These complexities were introduced apparently because neither their value to the user nor their cost to the economy were adequately understood. Today's languages are therefore poorly designed from this point of view. This poor design requires the multiplicity of compilers to translate them, on the one hand, and the large effort to develop syntax-directed compilers, compiler-compilers, etc., on the other. Both are actually unnecessary.

Similarity of Subroutine Calls and Action Operator Calls

The fact that action operators are represented in the translator by subroutines leads us to realize that action operator calls and subroutine calls must possess a similarity. Furthermore, work in recent years[5] has shown that the calls can be handled identically from a syntactic point of view. Here in this article, we use the mathematical function notation for both purposes. This notation, which employs parentheses, is in line with our selection of parentheses for our quote form, since the parameter list is one of the word groups for which the quote form was created. Some other forms are also reasonable. The important point here is that we should employ *regular* forms, rather than a conglomeration of syntactic structures.

Meta-Language Considerations

The procedures and operators of the language base, together with the parameters they take, constitute a *meta-language,* that is, a language which can be used to describe the computer languages to be defined. This meta-language however, is structured in the same way as other computer languages and therefore it can be treated in the same way. This similarity is the result of the fact that meta-languages have the same elements as ordinary computer languages. Thus, our meta-language can operate on itself. It can produce other meta-languages, as well as computer languages — and as a matter of fact, no distinction is made between the two. All or part of the initial meta-language, or any later extension made to it, can be included in the computer language being defined. Thus, languages designed with this approach are naturally extendible.

The property of the proposed language base, that its languages can refer to themselves or to other languages, is an important and desirable property for a language. Incidentally, the natural languages of human beings have this property: English, for example, can be used to talk about English.

Practicality

The approach we have been considering is inherently practical, due principally to its simplicity, and has many advantages. It does not suffer from the complications of syntax-oriented systems. It provides the optimum translation speed; at the same time it provides a universal and comprehensive system. Of course, if the proposed language base is employed to translate a syntactically-complex language, as it could be, then the translate-time will increase accordingly. But the pure system is simple, basic, and fast.

Implementations

The approach described here has evolved over a period of about ten years. The earliest implementation was the DECAL assembler/compiler for the DEC PDP-1 computer, in 1960. The second general implementation was the SET Self-Extending Translator for the IBM 7094, in 1964.

These implementations demonstrated practically that the basic concepts presented here were sound. The later one, SET, demonstrated the capability for handling the tasks of language definition and translation.

Value

Once the proposed language base is generally available and in use, it promises to reduce the cost of software systems and of large-scale applications programs by upwards of an order of magnitude. The predicted savings in software costs is based on several factors. First, a far more powerful tool (the general processor) is available for software production. Second, the general processor will translate all languages, and this processor would be no more complex than a single one of today's compilers. Third, the extension of the modularity concept to the whole of the software system can result in a similar reduction in the cost of other components of the system.

The reduction of applications programming costs is based on the availability of much more powerful languages for large-scale systems. In addition, savings in small-scale programs can also be realized because user languages can be tailored to their needs and will be compatible with one another, so that they will all be available on every computer.

Summary

By way of summary, we have isolated a set of basic elements of computer language and software systems and have utilized these elements to create a general language base and a general software base. Languages are then defined on the language base by specifying their verbs and the parameters they take. These definitions are accepted by the software base. The vocabulary is added to the dictionary of the general processor of the software base. The algorithms of translation are added as subroutines to a central library.

This approach has simplified the syntax of languages to the point where it is no longer a problem to either the language designer or the user. The removal of syntactic complexities, which abound in today's languages, produces unification of computer languages and simplifies software to the point where only one translator is required. This translator, though no more complex than an ordinary compiler, has essentially the power of all existing compilers and assemblers.

The language designer finds this system supplies him with a simple method of defining languages. Since the languages are compatible, he is not adding to the Tower of Babel problem. Since the languages do not require separate compilers, he is not adding to the software system problem.

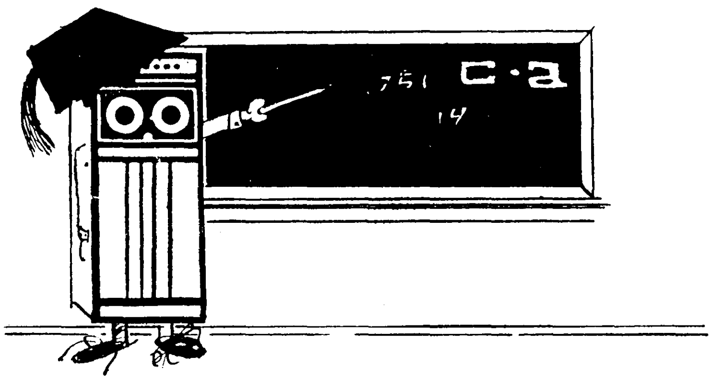
The user should find the languages easy to learn and easy to use. Furthermore, he can go from one language to another easily and he can use pieces from different languages together in one program.

Finally, the systems programmer finds part of his job done, in that he does not have to cope with a multiplicity of compilers. For the rest of his job, he will find the proposed general processor to be the ultimate tool, in that it contains all of the facilities he might want. He may employ machine language where he wants to; or he may use higher-level language facilities where he wants to; and he can carefully tailor these to produce exactly the object code he desires.

Thus, the proposed language base serves to unify the design of computer languages; and the proposed software base eliminates the need for separate assemblers and compilers, yet at the same time provides a base for software systems and a tool for the software programmer.

References

1. Constantine, L. L. "The Programming Profession, Programming Theory, and Programming Education." Computers and Automation, Feb., 1968.
2. Halpern, M. "XPOP: a Metalanguage without Metaphysics." *Proc. AFIPS 1964.* FJCC, Col. 26, pp. 57-68.
3. Halpern, M. "Toward a General Processor for Programming Languages." Comm. *ACM* 11, (Jan. 1968).
4. Bennett, R. K. "BUILD, A Base for Uniform Language Definition: A User's Manual for Computer Language Designers and Systems Programmers." June 1968. (Will be available from DDC, Cameron Station, Alexandria, Va. Mention Air Force Contract F44620-68-C-0007 when ordering.)
5. Bennett, R. K., and Kvilekval, A. H. "SET, Self-Extending Translator," published by DATA PROCESSING, INC., Waltham, Mass. Memo TM-2. March 3, 1964. □



Course C12:

COMPUTING, PROGRAMMING, AND SYSTEMS FUNDAMENTALS FOR SUPERVISORY MANAGEMENT — WITH 'HANDS-ON-THE-COMPUTER' ORIENTATION AND EXPERIENCE

When you, sitting at a computer,

- have the entire machine at your command;

  can look into any register you choose (to see what is really there);

- can experiment first with simple programs, then with more complicated ones;

- can experiment with several different programming languages; and

- have someone at your elbow to answer questions when you are perplexed or curious

— then you are participating in one of the most exciting, interesting, instructive, and important experiences of the computer age.

This kind of experience is, we think, part of the essential background of supervisory management. With such experience, supervisors of data processing departments and divisions are better able to:

- make reality-based appraisals of computing and data processing;

- form sensible judgments that are relatively independent of what the computer professionals in their groups may tell them;

- avoid commitment to unworkable proposals and costly errors.

HERE IS AN OPPORTUNITY FOR YOU TO GAIN THIS EXCITING AND ESSENTIAL BACKGROUND EXPERIENCE —

We (Computers and Automation's Computer-Assisted Instruction Center) have a powerful, modern, small general-purpose computer, a Digital Equipment Corporation PDP-9 with 8000 registers of core memory; it can perform 500,000 additions per second; etc. It is especially suited for investigation, experiment, research, and instruction.

Using this computer, and our experience since 1939 in the computer field, we are teaching, each month:

Course C12: COMPUTING, PROGRAMMING, AND SYSTEMS FUNDAMENTALS FOR SUPERVISORY MANAGEMENT — WITH 'HANDS-ON-THE-COMPUTER' ORIENTATION AND EXPERIENCE

This course will be offered Wednesday through Friday in the third week of each month March through November, at our instruction center where the computer is installed. Dates: March 19-21, April 16-18, May 21-23, ...

The enrollment in the course is limited to 12. Computer time for course enrollees is included in the course fee and is available Wednesday through Sunday, during the week when the course is offered. Each day begins with a lecture 9 a.m. to 11:45. After the lecture, the course divides into study groups of three or four persons who will have access together to the computer for three hours at a time; while one person runs his program, the others will work out or correct their programs. The instructor is available for guidance at all times. The fee for the course is $235.

WHO SHOULD TAKE COURSE C12?

In a recent article in Computers and Automation, Swen Larsen, now President, Computer Age Industries, said:

> In many companies, the top operating executive — the one who makes the key decisions — came into his position of responsibility before the computer revolution. Of all the men in an organization, he is probably the one in the greatest need of knowledge of the computer. Two computer experts describe the manager's plight in this way:
>
> > The executive is likely to be baffled, or confused, or snowed. He has confidence in his firm's EDP manager, but he doesn't understand the jargon that he hears, nor does he comprehend what can be effected from the tools he controls.

Course C12 is directed squarely towards these people and this problem.

WHAT TOPICS ARE INCLUDED IN COURSE C12?

- Fundamentals of Computing, and Orientation in Computers and Programming, with "hands-on-the-computer" experience in:
  - how to compute;
  - how to program;
  - how to edit a program;
  - how to assemble a program;
  - how to debug a program; etc.

- Some Powerful Concepts in Programming
  - modular programming;
  - multiprocessing;
  - multiprogramming;
  - remote batch processing; etc.

- Introduction to Programming Languages (including hands-on operation)
  - machine language;
  - assembly language;
  - special-purpose programming languages;
  - problem-oriented programming languages;
  - DDT (for an operating system);
  - FORTRAN (for numerical calculation);
  - LISP (for operations on lists);
  - EXPL (for vocabulary analysis); etc.

- Basic Principles of Systems in Computer Applications
  - observation of systems and networks;
  - dimensions and behavior of systems;
  - feasibility studies for computer applications;
  - benchmark problems;
  - principles of systems synthesis;
  - external constraints; etc.

- Applications and Nonapplications of Computers
  - standard applications
  - rare applications
  - frontier applications
  - man vs. computer, advantages vs. disadvantages

- The Natural History of Mistakes, and How to Avoid Them
  - colossal historical mistakes: Maginot Line, 1929; Mont Pélée, 1902; etc.
  - colossal industry mistakes such as the Edsel; distortion; bias; fallacies; etc.
  - ten principles for preventing mistakes in computer systems and applications.

WHO IS THE INSTRUCTOR?

The instructor for this course is Edmund C. Berkeley, editor and publisher of Computers and Automation since 1951, and president of Berkeley Enterprises, Inc. since 1954. He has been in the computer field since 1939. He took part in building and operating the first automatic computers, the Mark I and II, at Harvard University in 1944-45; he is now implementing the programming language LISP for the DEC PDP-9 computer. Mr. Berkeley was a founder of the Association for Computing Machinery, and its secretary from 1947-53. He is: the the author of eleven books on computers and related subjects; a Fellow of the Society of Actuaries; a member of the Operations Research Society of America; etc. He has been invited to lecture on computers in the United States, Canada, England, Japan, the Soviet Union, and Australia. He is a graduate of Harvard College, 1930, A. B. summa cum laude; his major was mathematics.

SOME COMMENTS FROM STUDENTS

I liked particularly the opportunity to write and run a program, and the informal seminar fashion in which the course was presented. I enjoyed the course very much and think it will make a good base from which to expand my knowledge of EDP operations by means of additional reading.
— Actuary, a large mutual life insurance company

The lectures and discussions were most interesting and informative.
— Assistant to the Manager of Technical Services, a large manufacturing corporation

The course fulfilled my expectations in giving me an understanding of the potential value of the computer to my State.
— Training Technician, Bureau of Personnel, a New England State

I found the small group interaction to be most effective. Small groups of this type are definitely the answer to securing the maximum information in the minimum time permitted. It was most gratifying to have so many hourly segments of "hands-on" time available on the computer, and I must admit DDT is the first enjoyable machine language I've worked with.
— Director of Computer Research, a commercial college

I wish to express my satisfaction with the generous time allocated to the operation of the PDP-9 as a "verification" tool. I feel that this is valuable to both the experienced and novice alike. .... I found that the small class was extremely beneficial as it gave ample time for covering the many individual questions raised.
— Assistant Manager, Patent Licensing, a large electrical manufacturing company

(Names will be supplied on request)

PLEASE REGISTER ME FOR Course C12: Computing, Programming, and Systems Fundamentals for Supervisory Management

( ) March 19-21 ( ) April 16-18 ( ) May 21-23

Name:________________________

Title:________________________

Firm:________________________

Street:____________ Phone:____________

City:__________ State:________ Zip:______

Additional Students:

Name:____________ Title:____________

Name:____________ Title:____________

( ) Bill me ( ) Bill my company ( ) Check enclosed

Mail to: computers and automation
COMPUTER-ASSISTED INSTRUCTION CENTER
815 Washington Street
Newtonville, Mass. 02160 (617) 332-5453

WORLDWIDE

REPORT FROM GREAT BRITAIN

Looking back on 1968, the two EDP events of outstanding importance so far as British-owned industry is concerned are the appearance of International Computers Limited (ICL) as *THE* business and scientific computer maker, and the announcement by the General Electric Company of Britain concerning the way in which this group, Europe's biggest electrical/electronic complex, proposes to run its automation business.

ICL, now fighting IBM, Honeywell, and Burroughs on every front, needs a period to settle down and solve its internal problems. These stem mainly from the sequence of mergers which, in the end, constituted the company. The last was between International Computers and Tabulators and the Computer branch of English Electric (minus process control). It is the one from which most problems will stem.

ICL Copes With System-4

The ICL management — as part of the package deal with Government — is required to continue making, developing and supporting System-4, which is English for Spectra-70. While sales of the bigger machines have been good — doing quite a lot of harm to the top of the 1900 series — the smaller ones have not been popular. At the same time, production facilities have been laid down for System-4 which were totally unjustified by the order book. They are lavish. The numbers of machines coming off the line are not.

So ICL, instead of being able to call it a day, still has to solve all the hardware and software problems of a series designed ostensibly to "compete with IBM 360, be compatible with it and outperform it." In theory this is a laudable objective. In practice it can and does involve the British company in some agonising decisions, particularly on peripherals, if it wants to take business away from IBM with equipment better suited to real-time uses than the 1900 series — though the 1904A and 1905A should meet this requirement better than the previous range. These have, however, not yet been announced or described in detail.

In fact, the whole armoury of ICL is in some disarray. There is the extremely popular 1901A and the 1902A and 3A for which the peripheral back-up is so far a good way from being comprehensive. Then come the 1904/S E and F machines which are not microcircuit-based (T^2L) like the A series, but are the enhancements of the first 1904/S brought out to meet the encroachment of System-4. Much higher up the scale comes the 1906A (E^2CL) which has a starting price tag of about $3 million and will not be seen till 1970 and then finally the 1908A which is the UK answer to the super-scale equipment from America.

On the System-4 side are three machines. These are the "30" which is a Spectra 70/3S, the "50" and the "70." There is a special time-sharing version of the latter called the "75" which made its first appearance one year late on the 80-terminal project for Edinburgh University. Some observers believe ICL may cut its losses and drop this version, waiting till the early 1970's and a design of machine compatible with both 1900 and System-4 to provide true time-sharing capability up the scale.

Still to be solved also is the burning question of peripherals, especially discs, since ICL has yet to announce a multi-spindle disc system of its own.

GEC Expands Process Control and Military Systems

On the process-control, defence and weapons and radar display front, General Electric Company acted immediately once the legal formalities of its English Electric takeover had gone through. This company, with a market valuation of $2,100 million, has set up two large groups. GEC-Elliott Automation will cover the whole industrial automation front from processes requiring only simple sequential controllers to a full-scale turnkey contract to supply an automated drive for a big steel rolling mill, plus unique ability on traffic control.

GEC-Marconi Electronics, with a yearly turnover of $300 million, is probably the world leader in radar. But it will also englobe a microcircuit production probably eighth in line so far as size goes, on a world scale. To this must be added an enormous military systems capability in nav-attack equipment and submarine devices. This company will be responsible for making the computers that it and the automation group will require. It has thousands of orders in the pipeline for tiny airborne computers for engine and navigation equipment control.

Burroughs and NCR Become More "British"

Coming at a time when both Burroughs and National Cash Register Company have raised money on the UK market and attained a quotation on the London Stock Exchange — thus taking on a more "British" identity than ever before — these moves are of particular significance. They mean that in business and scientific computing as well as in process control and other applications of automation techniques there will be just one British company, facing competitors who are trying very hard to attain Britishness, that is all except IBM. It could be that in the latter's case there is an element of shrewd political calculation in that the company expects that neither the Gaullist opposition to Britain's entry into the Common Market nor the present Socialist Government in Britain will outlast the end of the present decade. ☐

Ted Schoeters
Stanmore, Middlesex
England

IBM's Future Prospects

I. and U. Prakash, Editors
DP Focus
DP DATA Corporation,
61 Helen Drive,
Marlboro, Mass. 01752

During recent months and weeks, much attention has been focussed on the future of the IBM Company. Many comments have appeared in various publications. We have attempted to obtain the facts on some of these issues, and a summary of the facts is presented here, with an evaluation.

Opinions

Commentaries on IBM have recently stated:

1. *Orders.* Orders at IBM are slowing down. (Possibly this refers to IBM's U.S. business; however, the comment was not explicit.)

2. *Additions to Systems.* IBM is having difficulties and its salesmen are finding it difficult to sell so called "additions to systems" rather than the systems themselves. Some people predict that IBM's salesmen need more training to do their jobs. Others use this reasoning to deduce lower operating returns for the company.

3. *Time Sharing.* IBM is not living up to certain provisions of the 1956 consent decree; specifically, it has re-entered the service bureau business not directly but through the rear door, via the time-sharing services offered by its data processing division.

4. *IBM's Growth Rate.* IBM will have serious difficulties in maintaining its rate of growth from year to year. The primary reason for this statement is the large increases in IBM's reported income for 1967 and 1968 due to purchases of its equipment by independent leasing companies and the possibility that such purchases will show a sharp decline in 1969. With a decline in leasing company purchases of IBM machines, it is argued that its reported income in 1969 and 1970 years will decline from 1967 and 1968.

In conjunction with the computer market research activities of DP Data Corp. information about IBM's orders has been obtained from a number of sources. Specific information from very many IBM customers has been summarized and tested against a market profile model. We believe that the figures presented in Table 1 are as accurate and realistic as possible, without the benefit of official statements from IBM.

IBM Orders

In Table 1, the gross amount on-order is the total amount of orders placed by IBM's customers. However, when a customer places an order for a new machine which will replace an existing IBM machine on rent, the latter is then placed by IBM in a category indicating that it is available for renting elsewhere to another customer who desires that particular system. The net amount on-order is the gross amount on-order LESS the amount of the equipment IBM will receive back from existing installations. Thus, the net amount on-order shows the increase in IBM customer requirements for equipment over existing installed equipment. It is the equipment on order that will result in increased revenues for IBM after the equipment on order has been installed at the customer's premises.

Calculated from the figures presented in Table 1, the ratio of Net Amount On-Order to Gross Amount On-Order on the entire line of current IBM products (excluding unit record equipment) is almost 63.5%. To us, this is an excellent ratio, taking into account both the past IBM record and the past and current record of its major competitors.

Table 1 excludes other IBM products which are available from the company's data processing division. Also excluded are products ordered from other IBM divisions (like Office Products) which can sometimes be used with IBM data processing equipment.

The figures in Table 1 for order backlog at IBM shows that IBM has a healthy business at this time. Although these figures may be below those of some previous period, we believe they are ahead of the figures for other recent periods which have been considered more realistic in terms of a standard of "business on order". The total in Table 1 is also realistic from the point of view of the installations possible in the coming year.

If the figures in Table 1 are added to those of IBM World Trade Corp., IBM's operating subsidiary handling its business in parts of the world outside the U.S., then the total backlog for IBM is in a very healthy state. IBM World Trade is no longer an insignificant part of IBM's total activities; its business is expanding faster than the U.S. business, and so IBM's total business on order is actually considerably above the figures of previous periods.

Table 1

IBM COMPUTER EQUIPMENT ON ORDER IN THE UNITED STATES

Estimates of Gross and Net amounts as of September 1968

Data arranged by type of computer system in order of age since initial customer installation

TYPE OF MAJOR COMPUTER SYSTEM	AMOUNT OF TIME SINCE INITIAL CUSTOMER INSTALLATION Years-Months	GROSS AMOUNT ON-ORDER IN MONTHLY RENTAL DOLLAR VALUE (000) (Rounded to 10,000)	NET AMOUNT ON-ORDER IN MONTHLY RENTAL DOLLAR VALUE (000) (Rounded to 10,000)
7090	8-11	-	(70)
7070	8-7	-	(140)
7074	8-7	60	(440)
1620	8-1	100	(300)
1401	8-1	750	(4,050)
7080	7-2	60	(120)
1410	6-11	130	(1,270)
7094-I	6-1	60	(180)
1440	5-5	120	(2,850)
7040	5-4	20	(580)
7044	5-4	-	(510)
1460	5-0	50	(950)
7010	5-0	50	(910)
7094-II	4-5	-	(80)
3640	3-6	21,840	17,730
3630	3-5	20,710	11,390
3650	3-2	16,930	12,430
3665	2-11	20,020	17,810
3620	2-10	6,800	4,600
1800	2-9	550	460
1130	2-8	630	380
3675	2-8	1,650	850
3644	2-3	410	320
3667	2-2	2,420	1,810
3690	0-11	-	-
3625	None Installed	6,100	6,100
3685	None Installed	4,950	4,950
TOTAL		104,410	66,330

Additions to IBM Systems

Using our files, we have studied orders showing "growth of accounts based on add-on equipment excluding a central processor" and those showing "growth due to the addition of complete systems, including central processors".

This distinction may of course be of little practical importance to those interested in evaluating IBM's total business. IBM's equipment is not priced on a basis which makes any such differentiation between different segments of a data processing system. IBM's main emphasis is on servicing its customer needs. When these needs are better served with more equipment added to the initial system, this is what IBM salesmen propose. However, if a customer's best interests are served by upgrading his system through substitution of a new processor (even of a kind currently in use but having more memory capacity), IBM salesmen usually suggest this. But this distinction is important when studying the total rental value of additional IBM equipment installed.

We estimate that the rental value of additional equipment installed in recent time periods is higher than a year ago. U.S. orders show a slowing down of new central processing units, but IBM World Trade had more orders for central processors than in previous periods. In a nutshell, the changing mix of equipment ordered and installed by users has not had an adverse effect on the future of the IBM Company. Accordingly, based on the records available dealing with "add-on" sales to specific customers collected by our research activities and supporting information from other reliable sources close to the marketplace, there appears to be no evidence that IBM salesmen are experiencing any difficulties in recommending equipment to customers. IBM customers have responded to these recommendations as in the past: accepting some of them, modifying others, and deciding that some do not satisfy their needs. To us the situation shows nothing new or that "IBM salesmen need more training". Whatever market research led to this remark relating to IBM salesmen is clearly not confirmed.

Time Sharing

IBM has operated in the time-sharing business for several years. These operations have not been a significant part of the company's data processing division's business. We believe that claims by other companies, particularly General Electric, about their great expansion in time sharing, led IBM to defend its "image" rather than its pocketbook. However, IBM apparently concluded that this "image" would cause problems in the marketing of data processing systems at some date in the future. Thus, the Information Marketing Department (IMD) was set up. Apparently, this was an expensive endeavor since over 200 salesmen and other technical support personnel were utilized for a revenue running in the tens of millions of dollars. The equipment used was competitive at the inception of these services. It appears to us that IBM acted reasonably in supplying all of its latest System 360 equipment to its regular customers rather than diverting some of it for time-sharing services. The time-sharing services did not suffer any major competitive disadvantage from the use of earlier equipment, particularly since the systems were powerful enough to handle all the available business. In addition, proven software was available to these time-sharing customers.

General Electric's time-sharing and other computing services set up in recent years did not take the same route. General Electric, lacking success in the data processing marketplace for lease and sale of its systems, apparently utilized the equipment produced by its factories in the time-sharing service area. Here customers were persuaded to "try GE equipment" on a time-sharing basis, with the hope that they would become regular customers for GE installed equipment at a later date. Also, in this niche of the marketplace, GE would be able to claim some superiority in offering services, a claim not of success in use of its services by clients (or the profitability of such services to GE), but of stressing the availability of time-sharing facilities and hardware.

It should be noted that up to this time no one claimed that IBM's time-sharing services and activities were violating its consent decree provisions in any way.

During 1968, with the partial catch-up in demand for its System 360 equipment, IBM began to replace its earlier equipment with System 360 hardware and to move into the time-sharing business in a big way. To do this made common sense to IBM. In addition, IBM started to use its newly developed software, some of it specially designed for the System 360 and with time-sharing services.

From fragmentary reports available at this time, users of IBM's new time-sharing services were pleased with what they got for their money. When customer reactions became so favorable that some of the other competitive time-sharing businesses started to have greater difficulty attracting customers (and in some cases lost existing customers), the bogey of anti-trust violations was raised. (These losers in competition were aided and abetted by some reporters and editors who seek to present headlines, no matter how distorted they may be. Here is a way of making people read their publications, the same idea used in many newspapers around the world who in this way stress all that is unfortunate and undesirable in human life.) Our research staff surveyed a large number of publications but found only a few who were interested in the facts. It became fashionable to speak "for these small businesses" (is General Electric one of them?) and against the newly arrived competitor, IBM. Even the users did not speak up loudly.

Service Bureau Corporation

To us it appeared IBM would quietly seek to settle the issue. IBM did this by transferring its time-sharing services to its wholly owned subsidiary, the Service Bureau Corporation (SBC). The 1956 consent decree permitted IBM to conduct a service bureau business through SBC. However, how much is the user to pay for this apparently illogical application of the anti-trust laws, to keep companies in business when they are unable to compete fully? We believe IBM will push this issue which affects its "image" more than its pocketbook. We hope they do so with vigor, for the benefit of any and all users and a free market. To us it appears that GE and others, who claim a headstart in computer time-sharing, have now met full competition.

Based on information collected by our research department, we believe that SBC, which has been in some difficulty in meeting its financial growth objectives in recent months, will be happy to get the new business and expand it. SBC has several strengths: adequate amounts of the latest data processing equipment; a national sales organization and coverage; a large systems and programming staff; and an established rapport with many customers wishing to use outside services. The time-sharing business will bring SBC new products which dovetail with existing SBC services. A factor of no small importance, the attention of IBM's top management, will be devoted to enable SBC to render a top quality service to its time-sharing customers. We think it likely that IBM would probably have made time-sharing a separate independent entity within IBM if the anti-trust bogey had not been raised. However, from a business standpoint, the move to shift time-sharing to SBC is an even better arrangement. SBC's growth in both batch processing and the terminal inquiry market (TIM) seems assured, as well as intensifying competition among the existing suppliers in the marketplace.

The Growth Rate of IBM Income

Leasing company purchases of IBM equipment started several years ago. Only in the second half of 1967 did these purchases become large, in total dollar value of equipment purchased, as compared with the normal increase in IBM's total business (and total increase in lease income from equipment on rent). Since IBM income from equipment purchased by the leasing companies expanded at a high rate, IBM total income also experienced a rise faster than in previous periods. The increase in the rate of total income increase was not caused by an acceleration in IBM's business but by an increased sales to lease ratio. Since equipment sold by IBM brings in fifty dollars compared to only six to twelve rental dollars, depending on whether it is new equipment installed on an even rate during the year or previously installed equipment, the distortion in usual IBM total income during 1967 was great.

Purchases of IBM equipment by leasing companies has continued during 1968. We believe that IBM 1968 financial results will fully reflect this tremendous increase. However, in 1969, if leasing company purchases in the U.S. drop off, IBM's U.S. income from the purchase of its equipment is likely to show a sharp decline. Such decline cannot be overcome by the slowly increasing rental income, assuming no other changes are made by IBM. Since IBM continues to install equipment with users at a rate consistent with previous periods, lease income cannot expand at a rate fast enough to compensate for the drop in income from purchases of its equipment. (In this review, we have assumed no change in IBM policies, particularly those relating to depreciation of its equipment.)

It is quite possible that leasing company purchases of IBM equipment outside the U.S. will counterbalance the drop in the U.S. income. Current reports indicate that most of the large leasing companies plan to expand into Europe in late 1968 and during 1969. If they do, and if the 1969 purchases by leasing companies in Europe expands appreciably, it is quite possible that 1969 will not witness any drop in total IBM income. However, how long can the day of a drop in IBM's total income be put off? The postponement is now, to a certain extent, beyond IBM's control in the short run, assuming the present practices, procedures, and restraints upon IBM. Since the 1956 consent decree required that IBM offer all its equipment for sale if desired by a customer, IBM's total income necessarily depends on the actions of many other parties in their decisions to rent, lease, or purchase its equipment. In the short run, independent leasing companies largely control the answer to the above question. In the long run, IBM has various avenues open to bring its yearly income more under its own control. This would swing the situation back to what it has been over the greater part of the past fifty-year history of the company.

ESTIMATING THE TIMING OF WORKLOAD ON ADP SYSTEMS: An Evaluation of Methods Used

Fletcher J. Buckley, Lt. Col., U.S. Army
Automatic Data Field Systems Command
Fort Belvoir, Va. 22060

"The advantages of benchmark techniques are that the actual operating system, the actual compiler, and the actual machines proposed by the vendor are all tested."

As part of a procurement of equipment for automatic data processing (ADP), specifications are developed and released to industry. After study, vendors reply to these specifications with bids, each stating how he proposes his equipment to perform the stated job. As a portion of the evaluation of these proposals, it is necessary to validate the vendors' timing data to insure that the machines that they propose can, in fact, perform the total workload in the required time.[1] At present, a number of substantially different methods are used to estimate the amount of time required for a specified computer system to perform a given workload. The purpose of this paper is to provide an understanding and an evaluation of methods for timing ADP systems.

System Elements

The elements that comprise an automatic data processing system are shown in simplified fashion in Figure 1. Each of

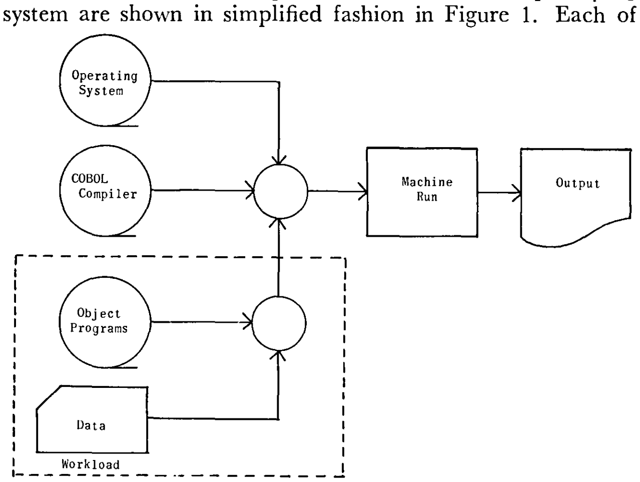


Figure 1. ADP System Elements

these elements can vary significantly from manufacturer to manufacturer. Each of the variations can significantly affect the ability of the system to process a specified workload.

The operating system is normally supplied by the vendor and can use a substantial percentage of the actual machine time available for productive runs. This "overhead" time is not necessarily a constant with the operating system on a specified machine configuration, but may vary with the type and size of the object programs, the data, and the organization of the queue.

Compilers also can vary significantly in several ways; for example, time to compile, size of machine language program, and time required to execute the compiled machine language program.[2] The effects of each of these factors depends upon the usage of the ADP system; for example, if the programs which are compiled are unique and run only once, then the compilation time becomes more significant than if the program is compiled once and run many times.

The object programs and the associated data comprise the workload. The execution of the workload is the purpose of the entire configuration. However, unless the operating system does a pre-sort for maximum effective use of the hardware, the time required for the machine run can be a significant function of the order in which the object programs and the data are provided to the machine. Thus, the workload queue must be fixed for accurate workload timing if the operating systems being considered are not able to pre-sort.

The ADP machines on which the programs are projected to be run have widely varying characteristics. These include not only those easily identified; e.g., memory cycle time and type of addressing (single, double, triple, etc.), but also characteristics which, while more subtle, can be far more significant; e.g., instruction repertoire. As an example of the latter type, consider the MOVE-EDIT instruction on the IBM 360 series. There is nothing comparable to this instruction on a 7090. As a result, an operation which may take 2000 executed instructions on the 7090 may require substantially fewer on a 360.

System Timing Methods

A number of means may be used to estimate the ability of a specific ADP system to perform a specified workload.

First, mathematical calculations may be performed either by hand or by machine. They are normally based on a workload description which includes a detailed layout of the files and generalized program descriptions. The detail of the program descriptions used for these calculations is usually restricted to a number of bulk estimated figures reporting the amount of core storage required to store the program and the number of instructions executed as this program goes to completion.

From the workload descriptions, two types of data can be derived. The usage of a central processor by a given program in a specific machine is provided by multiplying the figure for the number of computer instructions executed in that program by the execution time for an average instruction (previously computed for that particular machine). Input/output (I/O) processing time is computed by examining data and programs which must be passed to and from I/O devices, and using appropriate machine constants, such as, print time, device access time, etc.

The validity of this technique should be questioned for four reasons.

(1) The effects of the operating system are not determined.

(2) The efficiency of the compiler is not tested.

(3) The figure used as a number of instructions executed may be highly subjective. (It should be noted that there is no significant relationship between the number of instructions in the program, and the number of instructions executed as the program goes to completion.)

(4) The average instruction execution time is based on standard mixes of instructions, such as the so-called Gibson mix, which is also questionable as applied to various programs. The process of obtaining an execution time for a standard mix for a specific machine can be a further subjective action.[3]

Simulation

Second, simulation may be performed. Simulation of various classes of general purpose computers has been common for some time. This technique for actual workload timing for the spectrum of today's sophisticated computers has continued to be applied.[4] This technique uses the basic methodology described above, but couples this with attempts to yield better representations by:

(1) Using more complete object program descriptions. This can take the form of a "compute" function embedded with the I/O actions. This can be implemented to several different levels of detail.

(2) Interrelating the characteristics of the particular operating system to the object programs.

(3) Providing effects of random occurrences by incorporating probability tables.

Deficiencies of present efforts at simulation in general flow directly from some of the previously described deficiencies of mathematical techniques. These include: the use of average instruction execution time and an average instruction mix; specification of the total number of executed instructions; inability to relate the effects of the compiler to the total effort; and the substantial effort required to provide more sophisticated input requirements to the simulation.

The work required to make simulators valuable in comparative evaluations includes developing:

(1) Objective means of measuring the basic parameters of computers. These parameters include: time to compile; size of compiled program; and time to execute. This would have to be a relative measurement compared to other vendors' compilers. The simulation program would then be modified to permit this as input.

(2) Objective means of testing the basic parameters of operating systems.[5]

(3) Better methods of workload description (perhaps, using the number of compiler source statements). This then could be combined with: dynamic trace programs to achieve a better instruction mix; and an effort to arrive at an objective determination of execution time for the average instruction.

Additional work along these lines seems impractical. This work appears to be on the extreme fringe of the state-of-the-art. The general validity of its applicability to workload timing estimation appears to be questionable. Efforts to extend simulation into these areas are envisioned as a long-term effort with only uncertain success.

Benchmarks

Third, benchmark techniques seek to represent the total ADP workload by representative programs (see Figure 2).

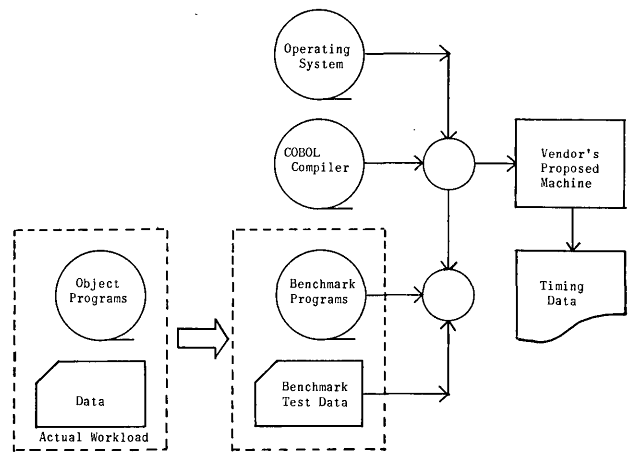


Figure 2. Benchmark Process

These programs are then run on the actual machine using the operating systems and compilers proposed by the individual vendors, to provide the desired data.[6]

The advantages of this system are that the actual operating system, the actual compiler, and the actual machines proposed by the vendor are tested. Abstraction of various machine characteristics, and further representation of the operating system, and compiler, are not necessary.

Significant objections associated with this technique are:

(1) Uncertainty that the benchmark programs and test data do actually represent the prescribed workload.

(2) The cost of providing benchmark programs, including the time actually used to program the benchmark and to acquire the test data.

(3) The amount of effort required to convert the benchmark program into a program for an individual vendor's machine.

(4) Cost of running the benchmark programs.

(5) Difficulty in obtaining the precise configuration of the vendor's proposed system on which to run the benchmark, and the apparent impracticability of attempting to completely benchmark all aspects of a workload, for example, 30 remote users in a real time situation.

Partial solutions to some of the current problems in the benchmark approach are, however, available.

Most of the workload should be grouped into significant classes, and each class should be represented by a particular program. The significance of a class should be determined by: does it occupy a major portion of the total workload in terms of running time? or does it place unique requirements on the ADP system?

The benchmark programs can be derived from a number of sources. Actual programs from the present ADP installation being replaced should be used. If this is not practical,

IDEAS: SPOTLIGHT

Computer-Managed Instruction

. . . A fifth-grade teacher with 30 or more students in four major subject areas, each consisting of some 100 teacher-learning units, is immediately presented with some 10,000 combination-patterns of students and teaching-learning units. To shape these patterns and keep track of each student's set of over 1000 behavioral objectives exceeds the teacher's memory and capacity to correlate, but is a trivial job for a third-generation computer.

The computer monitors and operates an informational system to support the teacher, who can then spend much more time with the students.

The computer in an individualized instruction system is a highly desirable but not essential ingredient. The system can run without it, but not as well and less economically since it is necessary to replace the computer by a staff of clerical personnel to take data, score tests, and correlate information. In computer-managed instruction, computers handle these clerical tasks. . . .

All of the instructors who participated in the [experimental program] were surprised at how well they knew their students after just a few months in the program. The bright students covered and learned more than twice as much material as usual and, more important, the slow and average students also successfully completed their objectives. . . .

— From "Computer Managed Instruction," p. 974, by Harvey J. Brudner, Vice President and Director of Research and Development for the Westinghouse Learning Corporation, 100 Park Ave., New York, N.Y., in *Science,* for 29 November 1968, pp. 970-976, published by the American Association for the Advancement of Science, 1515 Mass. Ave., N.W., Washington, D.C. 20005. □

---

then programs from other sources can be used. Portions of these programs should then be mixed to provide a tailor-made benchmark.[7] In similar fashion, test data can also be obtained.

It is sometimes considered appropriate to use two sets of data. The first set is used by the vendor to ensure that the benchmark program has been suitably modified to run on his machine. The second set is run under customer supervision to obtain actual running times.

Costs of Benchmark Programs

Reprogramming of benchmark programs to enable them to run on any of several individual vendor's machine is an important problem. Present Department of Defense policy is that COBOL is the required business programming language. There are differences, of course, in each vendor's COBOL compilers. Recognizing that reprogramming is a substantial cost for an ADP installation, the adjustments required to enable a COBOL program to run on different machines represent a significant element to be entered into the total cost estimate.[8]

It appears that two types of adjustments should be allowed. The first would be the minimum adjustments required to enable the benchmark program to run. The second would be any other change proposed by the vendor to enable the program to better utilize the specific capabilities of that vendor's system. Both of these adjustments should be separately documented and used as a part of the total evaluation process.

The actual costs of physically running a benchmark program are sometimes raised as a potential objection. But it should be recognized that the benchmark program does not have to run a significant amount of time compared to the actual workload. That is, a day's workload, a month's workload, or a year's workload does not have to be actually run. The only thing the benchmark has to do is to enable a representative workload to be demonstrated.

Consider the cost of running a benchmark in machine time. A class of machine with the characteristics of the IBM 360/50, with a normal complement of peripheral equipment, has a current commercial cost of approximately $300 an hour. Running a two-hour benchmark is not a substantial cost, compared to the remainder of the effort normally devoted to producing a vendor proposal.

For benchmark purposes, a configuration similar to that which is proposed should be available to the vendor. Of course some limited analysis and extension may be required to extend the timing data provided by the benchmark programs themselves. The amount required will require determination on an individual case-by-case basis and hopefully would be as small as possible.[9]

In conclusion, although the benchmark approach is not a panacea, yet with detailed analysis it can produce good results.

References

1. It should be noted that the means used by the vendor to make timing estimates are directly dictated to him by the specification format.
2. Efficiency of compilers can vary significantly even inside the same model computer; e.g., consider the case of attempting to use a COBOL compiler on the CDC 3300 without the BCD package. For a detailed comparison, see Joslin, E. O. and Aiken, J. J., "The Validity of Basing Computer Selections on Benchmark Results," *Computers and Automation,* January 1967, pages 22-23.
3. For further discussion of the use of a standard mix of computer instructions for computer system evaluation, see Arbuckle, R. A., "Computer Analysis and Thruput Evaluation," *Computers and Automation,* January 1966, pages 12-15, 19 and Calingaert, B., "System Performance Evaluation: Survey and Appraisal," *Comm ACM,* 10, 1, January 1967, pages 12-19. For a specific example of computation of average instruction times based on an average mix, see Raichelson, E., and Collins, G., "A Method for Comparing the Internal Operating Speeds of Computers," *Comm ACM,* 7, 5, May 1964, pages 309-310.
4. There is at least one commercially-available product in this line. See Herman, D. J., and Ihrer, F. C., "The Use of a Computer to Evaluate Computers," *Proc. AFIPS 1964 Spring Joint Computer Conference,* Vol 25, April 1964, pages 383-395.
5. It is important to note that although detailed representations of operating systems can be constructed based on what the manuals state the operating systems are, how the actual operating systems perform may unfortunately be something else.
6. Implicit in this benchmark approach is that the hardware and the software are "in being" at the time of vendor proposal.
7. The point here is not that general evaluations cannot be made using the benchmark approach. Such evaluations are in use today. For example, see Hillegass, J. R., "Standard Benchmark Problems Measure Computer Performance," *Computers and Automation,* January 1966, pages 16-19. The point is that evaluations of individual configurations for specific tasks must be job oriented.
8. For a detailed discussion of cost evaluations, see Joslin, E. O., and Mullin, M. F., "Cost-Value Technique for Evaluation of Computer System Proposals," *Proc. SJCC,* 1964, pages 367-381.
9. For a specific example of use of the benchmark technique, see Williams, Perrott, Weitzman, Murray and Shober, "A Methodology for Computer Selection Studies," *Computers and Automation,* May 1963. □

In Retrospect: The FIRST Business Feasibility Study in the Computer Field — Part 2

Henry W. Schrimpf, Jr., Supervising Actuarial Clerk
Clyde W. Compton, Methods Analyst
Prudential Insurance Co. of America
Newark, N.J. — July, 1947

"The main purpose of this report was to convince many unconvinced persons that an 'automatic large-scale sequence-controlled calculator' could actually perform the required business work. The report was successful; it produced the conviction."

INTRODUCTION (continued) by Edmund C. Berkeley

Part 1 of this article, published in the January 1969 issue, explained some of the surrounding conditions that led to this report, the first feasibility study that ever occurred in regard to the application of computers (then called "large scale calculating machines") to business problems.

Part 2 printed here gives a partial picture of the series of Exhibits included in that report. This part shows the thoroughness of the test designed and performed in 1947. This test consisted of producing by automatic computer 46 premium notices; they were produced correctly and responded to a wide variety of different conditions for making appropriate premium notices for ordinary life insurance policies.

Exhibit 2

POLICY SAMPLE - ORIGINAL INFORMATION FOR BILLING

[This exhibit consisted of photostatic copies of the front and back of the 46 renewal premium payment history cards — each card partly typed, partly handwritten — for the 46 policies.]

Exhibit 3

POLICY SAMPLE - SCHEME FOR TRANSLATING INTO MACHINE LANGUAGE

Item or Alphabetical Character	Numeric Substitute Assigned

Name and Address

(Note: Mark 1 has no alphabetic printing controllable by the machine. For this reason, the following scheme was adopted.)

A space	0
Any letter of first name	1
Any letter of middle name	2
Any letter of last name	3
Any letter of designation (Jr., Sr., etc.)	4
Any letter of name of street	5
Any letter of street classification (Street, Ave., etc.)	6
Any letter of name of city	7
Any letter of name of state	8
Any letter of title (Mr., Mrs., etc.)	9
Any number	The same number

Month

January to December	1 to 12, respectively

Mode of Premium Payment

	At Start	After Calculation was Completed
Annual	3	4
Semi-annual	1	2
Quarter-annual	0	1

Dividend Option

Reduction	1
Cash	2
Accumulation	3
Paid-up addition	4

Rate Group

Old dividend rate group	1
New dividend rate group	2

Dividend Block Code

[This consists of 67 "dividend block codes" from 1 to 67, each associated with a category of policy for dividend purposes. One example follows:]

Dividend Block:	18
Year of Issue:	1943
Kind of Policy:	20 payment life
Dividend Rate Group	Old
Age Group:	15-19

Dividend Rates per $1000 for the Ages in the Age Group:
3.74, 3.79, 3.83, 3.84, 3.85

Exhibit 4

POLICY SAMPLE —
EQUIVALENT NUMERIC INFORMATION PUNCHED IN CARDS

[Sample information is shown here for one policy only out of 46 policies in the complete exhibit.]

Card #1

Policy Number:	4761744
Premium Anniversary Month:	7
Mode of Premium Payment Code:	0
Dividend Block Code:	5
Age:	32
Paid-to-Date (Year, Month, and Day):	47-7-7

Card #2

Policy Number:	4761744
Rate Group Code:	1

Dividend Option Code:	1
Amount of Insurance:	5000

Card #3

Policy Number:	4761744
Premium:	3430
Name of Applicant or Assignee:	—
Name of Insured:	10203333333000000000

Card #4

Policy Number:	4761744
Number and Street:	37055506600000000
City and State:	777777777708080

Exhibit 5

POLICY SAMPLE - RULES FOR BILLING PREMIUMS
— PRELIMINARY MEMORANDUM FOR REFERENCE

Rules

July 1947 Billing;
Premium Notice and Receipt Forms;
Source of data : Renewal Cards

General For every premium notice required, type in the boxes provided:

Policy number
Date due
Mode of premium payment
Premium

1. If premiums are payable annually and the premium anniversary date falls in July, type a premium notice and receipt form.

2. If premiums are payable semi-annually and the premium anniversary date falls in January or July, type a premium notice and receipt form.

3. If premiums are payable quarter-annually and the premium anniversary date falls in January, April, July, or October, type a premium notice and receipt form.

4. If rules 1, 2, or 3 do not apply, no premium notice or receipt form is required.

5. The name and address of the insured is to be typed in the address box if "Add to" is not shown in the top left hand corner of the renewal card. (The "Add to" would indicate a juvenile policy.) Type the full name of the insured if the notation "full name" appears on the renewal card or if no middle initial of the insured is shown. Otherwise type the initials of the first two given names.

6. If "Add to" is shown at the top left hand corner of the renewal card (juvenile case), the name of the insured is to be typed under the policy number and the name and address of the applicant shown under the insured's name on the renewal card is to be typed in the address box.

7. If "Dup notice" is shown at the top left hand corner of the renewal card (Assignment or rights case), an additional receipt and notice form is to be typed for each assignee or holder of rights shown on the renewal card. These additional forms should have receipts typed in red. The name of the insured is to be shown under the policy number and the name and address of the assignee or holder of rights typed in the address box. The red typing on the receipts will serve as an indication that the receipts are to be destroyed.

8. If the dividend option in effect is the "reduction dividend" option, if the premium anniversary date falls in July, and if the policy is old enough to receive a dividend, the amount of the dividend and the amount due, which have been entered in the current premium posting box, should be typed in the "dividend" and "amount due" boxes.

9. The "dividend" and "amount received" ("amount due" box on notice) boxes are to be left blank if:

a. The dividend option in effect is not "reduction"
b. No dividend is payable because the premium anniversary date does not fall in July
c. The policy is not old enough to receive a dividend

Any one of the above conditions is indicated when the dividend and net premium figures are not shown in the current premium posting box of the renewal card.

Interest Notice Forms
Source of data - Loan Cards

General For every interest notice required, type in the boxes provided:

District
Date due
Policy number
Loan principal
Interest due

10. If a policy has a loan against it, and if the premium anniversary date falls in July, an interest notice is to be typed.

11. The name of the insured is to be shown in the address box if "over" is not shown in the memo box of the loan card.

12. If "over" is shown in the memo box of the loan card and a memo is on the back of the card to indicate that loan was granted to the applicant, the name of the insured followed by "__ins" is to be placed below the policy number on the interest notice and the name of the applicant is to be typed in the address box.

13. The address of the insured is to be shown in the address box if "over" is not shown in the memo box of the loan card and if the address is shown on the loan card.

14. If "over" is shown in the memo box of the loan card and a memo is on the back of the loan card to indicate that loan was granted to the applicant, the address of the applicant is to be shown in the address box if the address of the applicant is on the loan card.

15. If "Dup not over" is shown in the memo box of the loan card, an additional interest notice is to be typed for each assignee or holder of rts shown on the back of the loan card (or on the renewal card). The name of the insured followed by "__ins" is to be typed below the policy number and the name and address of the assignee or holder of rights is to be typed in the address box.

16. If rule 10 does not apply because the premium anniversary date does not fall in July, no interest notices are required.

17. If rule 1,2, or 3 applies and the insured requests that no premium notices are to be sent to him, the notice should be destroyed after the notice and receipt forms are typed.

18. If the policy has no loan against it, interest notices are not required.

19. If rule 1, 2, or 3 applies and premiums have been paid in advance, no premium notice and receipt is required.

Calculation of Interest

Determine for each amount of loan the number of days during which that loan was outstanding in the period between anniversary premium dates.

For initial loans effective interest is charged.

For reductions, simple interest is charged for the old and reduced loans.

For increases effective interest is charged for the old loan and added to the increased loan. Then effective interest is charged on the increased loan.

The rate of interest charged depends on the date of issue:

1-1-16 to 11-30-38	6% due end of year
12-1-38 and later	5% due end of year

The loan principal shown on the interest notice is the amount of loan outstanding on the interest due date at which time interest is due.

Calculation of Dividends

To calculate a dividend, the amount of dividend per unit amount of insurance for a policy depending on:

The year of issue
The rate group
The kind of policy
The age at issue

is found. This is multiplied by the number of unit amounts of insurance contained in the policy.

H. W. Schrimpf, Jr.
Methods Division

Z. I. Mosesson
Assistant Mathematician
Methods Division

Exhibit 6

POLICY SAMPLE - LIST OF RULES THAT APPLIED TO EACH POLICY

Policy Number	Rules to Follow to Produce Premium Notices and Interest Notices								
2708863	3	5	7	9b	16				
2942602	4	18							
3836291	2	5	9b	17	18				
4008972	3	5	9a	18					
4761744	3	5	7	7	8	10	11	15	15

[The rest of the exhibit shows the numbers of the rules that applied to each one of the remaining 41 policies in the policy sample.]

Exhibit 7

INSTRUCTIONS TO MACHINE - SETTING OF CONSTANT SWITCHES

Switch No.	Switch Code	Content of Switch	Machine Columns from Right to Left
1	741	1	1
2	742	7	4
3	7421	11	3,4
4	743	29	3,4
5	7431	60	3,4
6	7432	470732	3 to 8

[etc., for five more switches]

Exhibit 8

INSTRUCTIONS TO MACHINE - ASSIGNMENT OF STORAGE COUNTERS

Notes:

(1) For the meaning of the abbreviations, see the glossary at the end of this exhibit.

(2) In the following, under each counter, the left hand column shows the content of the counter, and the right hand column shows the line (coding line or instruction line) when the content becomes as shown.

Counter 1

Content	Line
XXX	12
Month	13
Month - 7	14

Counter 2

Content	Line
XXX	18
Month	19
Month + mode	27
Month + mode - 11	28
Month + mode - 11 - 29	32
Month + mode - 11 - 29 - 60	35

Counter 3

Content	Line
XXX	21
Mode	24

Counter 31

Content	Line
XXX	47
Pol #	48

Counter 32

Content	Line
XXX	53
Pol #	56

Counter 321

Content	Line
XXX	1
Pol #	101
Pol # & div op	115
Pol #, div op, & div	146

Counter 4

Content	Line
XXX	6
55 - n	7
XXX	11
Month - 7	15
2 Month - 7	16
2 Month - 7 - 1	17
XXX	29
Month + mode - 11	30
Month + mode - 1 - 11	31
XXX	36
-1	38
(Month + mode - 40 x Month + mode - 100) - 1	40
XXX	50
470732	51
470732 - pd to date	52
XXX	75
-1	78
Applicant - 1	79
XXX	83
-1	84
Div - 1	85
XXX	106
Unit digit of (2 x age)	108
Unit digit of (2 x age) -1	117
Unit digit of (2 x age) -3	120
Unit digit of (2 x age) -5	123
Unit digit of (2 x age) -7	126
Unit digit of (2 x age) -9	129
Unit digit of (2 x age) -M + rate group	133
XXX	139
-1	140
Div op - 1	143
XXX	189
-1	190
Applicant - 1	191

Counter 41

Content	Line
XXX	91
Div #	92

Counter 42

Content	Line
XXX	96
Age	97
2 x age	100

etc.

[The rest of the exhibit shows the content at every line (in similar style) of 20 more counters, from counter 43 to counter 74321, and also the LIO, SIO, Multiplicand, and Multiplier counters.]

Exhibit 8, Continued

Glossary of Abbreviations

Abbreviation	Meaning of Abbreviation
XXX	counter cleared
addressee	name of addressee
age	age of insured or effective age of insureds
amt	amount
ann	annually
applicant	name of applicant or assignee
carriage	typewriter carriage
ctr	counter
div	amount of dividend
div #	dividend block code number assigned to policy of Exhibit 3
div op	dividend option
div rate	dividend rate per $1000
ins	amount of insurance
insured	name of insured
LIO	logarithm-in-out counter
M	equals 1,3,5,7, or 9 depending on when counter 4 goes negative, on lines 117, 120, 123, 126, or 129
mode	mode of premium payment
month	premium anniversary month
pd-to date	paid-to date
pol #	policy number
prem	amount of premium
SIO	sine-in-out counter
SS	sub-sequence

Exhibit 9

INSTRUCTIONS TO MACHINE - MAIN SEQUENCE PROGRAMMING*

Main Tape

Purpose and Operation	Result on "In" Counter	Line	Out	In	Misc.
To Start Operation (Lines M1-M5):					
Clear ctr 5 and ctr LIO	0	M1	5	5	763
Clear ctr 531 and ctr SIO	0	M2	531	531	7321
Clear ctr 61	0	M3	61	61	7
Add 55 into ctr 61	55	M4	751	61	7
Call SS 1, line 1		M5		861	

To print totals (Lines M6-M11):

Add total divs billed into print ctr	Total of divs billed	M6	5	74321	
Print total divs billed on record sheet		M7		7521	651
Add total gross prems billed into print ctr	Total of gross prems billed	M8	531	74321	
Print total of gross prems billed on record sheet		M9		7521	651
Add total divs not reductions into print ctr	Total of divs not billed	M10	831	74321	
Print total of divs not reductions		M11		7521	651

*For meaning of abbreviations, see glossary at end of Exhibit 8.

Exhibit 10

INSTRUCTIONS TO MACHINE - SUB-SEQUENCE PROGRAMMING*

Sub-Sequence 1

Purpose and Operation	Result on "In" Counter	Line	Out	In	Misc.
1. To start operation (Lines 1-4):					
Clear ctr 321	0	1	321	321	7
Clear ctr 72	0	2	72	72	7
Clear ctr 721	0	3	721	721	7
Clear ctr 7321	0	4	7321	7321	7
2. Stop billing at end of run and print totals (Lines 5-9):					
Subtract 1 from sum in ctr 61	55-n	5	741	61	732
Clear ctr 4	0	6	4	4	7
(55-n) into ctr 4	55-n	7	61	4	7
Clear ctr 53	0	8	53	53	7
Conditional call, calling in main tape, line M6, if end of run, i.e. if ctr 4 goes negative	-	9	842	764	7
3. Put policy data from first card into ctrs and start to determine if month is July (Lines 10-22):					
Feed first of 4 policy cards	*Note 1	10	8652	7321	7632
Clear ctr 4	0	11	4	4	7
Clear ctr 1	0	12	1	1	7
Month into ctr 1	Month	13	8531	1	7
Subtract 7 from ctr 1	Month - 7	14	742	1	732
Positive absolute value from ctr 1 into ctr 4	Positive absolute value of (month - 7)	15	1	4	72
Double amt in ctr 4	Twice above value	16	4	4	743
Subtract 1 from sum in ctr 4	Above value -1	17	741	4	732
Clear ctr 21	0	18	21	21	7

*Note 1:
Div # and age in ctr 72, pol # and pd to date in ctr 721, month and mode in ctr 7321

*Note 2:
For the meaning of the abbreviations, see the glossary at the end of Exhibit 8.

[The exhibit continues with 14 more pages of description of the operations, covering: the rest of Sub-sequence 1 and all of Sub-sequences 2 through 10; lines of program from 19 through 217; and 27 more "purposes".]

Exhibit 11

INSTRUCTIONS TO MACHINE - BOARD PLUGGING

[This exhibit consists of three pages showing the plugboard wiring for two boards; the MP-DIV Plugboard; and the FUNCTIONAL Plugboard.]

Exhibit 12

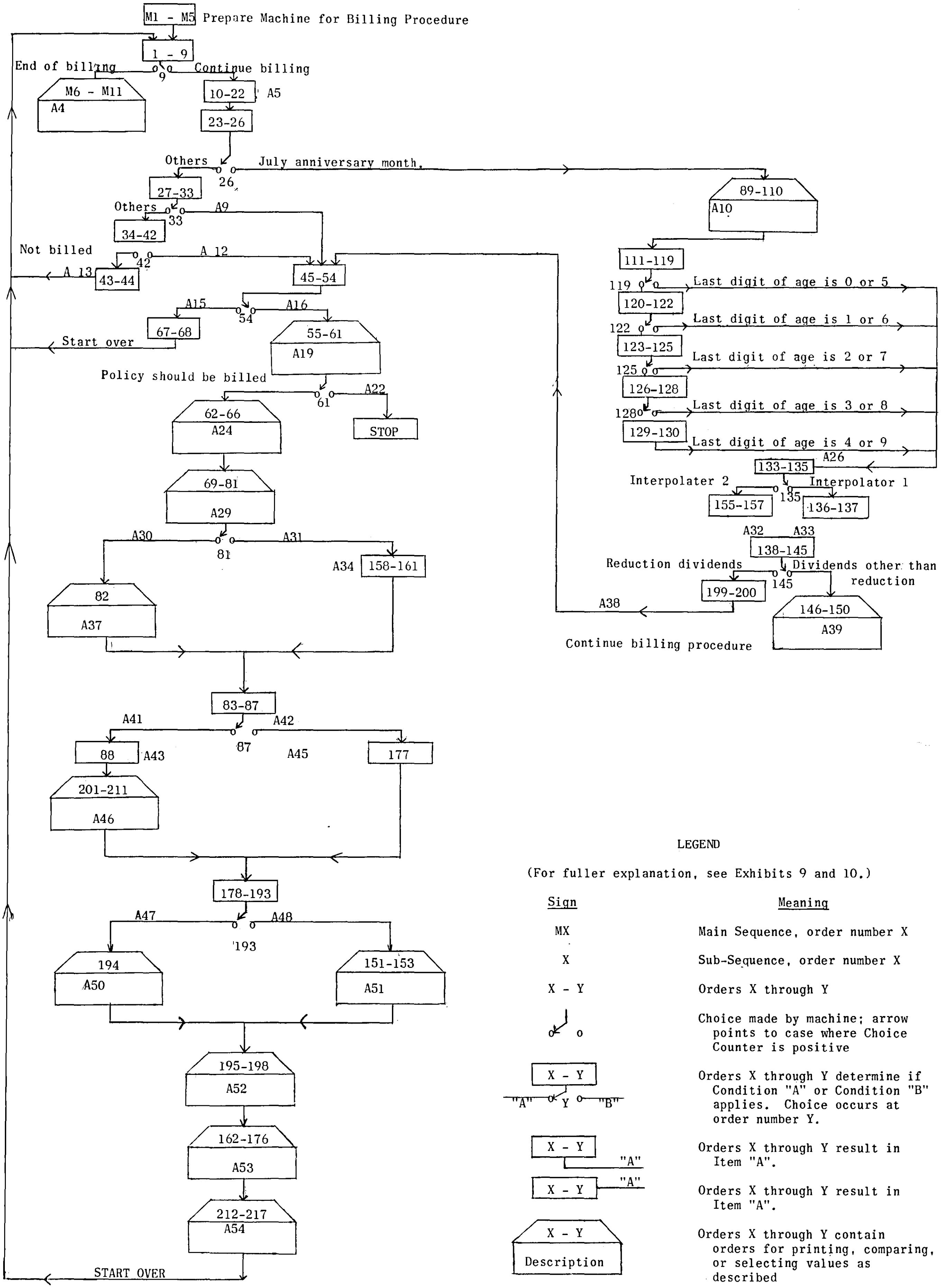
SCHEME OF PROGRAMMING FOLLOWED BY THE MACHINE

This consists of a flow chart containing 38 blocks and a legend explaining them. Each block states (1) the numbers of the instructions used to carry out the operation, and (2) the operation performed. Figure 1 shows the flow chart, and the legend.

The original Exhibit 12 contained many written explanations throughout. These explanations, if not copied in the flow chart shown here, are designated A1 to A55 and the translations are shown below.

- A 4. Print on dividend record: total of dividends not shown on bills, total gross prems. billed and total dividends billed.
- A 5. Part one of selection of policies having July anniversary month.
- A 9. January anniversary month payable semi or quarter annually.
- A10. Select group of dividend rates.
- A12. April or October anniversary month payable quarter annually.
- A13. Start over.
- A15. Premiums paid in advance not billed.
- A16. Premiums not paid in advance.
- A19. Comparison: in case policy going to be billed should not be, stop machine.
- A22. Policy should not be billed.
- A24. Print on bill: policy number and day due.
- A26. Dividend rates.
- A29. Print on bill: mode of premium payment and gross premium.
- A30. Bill requiring name of insured under policy number.
- A31. Bill not requiring name of insured under policy number.
- A32. Dividend rate.
- A33. Dividend rate.
- A34. 81 Tab typewriter to dividend box since name of insured is not to be printed.
- A37. Print on bill: name of insured under policy number.
- A38. Continue billing procedure.
- A39. Print on dividend record: policy number, option and amount of dividend.
- A41. Premium to be reduced by dividend.
- A42. Policy receiving no reduction dividend.
- A43. Skip to printing orders for dividend and net amount due.
- A45. Position typewriter to third line of billing form.
- A46. Print on bill: dividend and net amount due.
- A47. Bill to be sent to applicant or assignee.
- A48. Bill to be sent to insured.
- A50. Get ready to print: name of applicant or assignee.
- A51. Get ready to print: name of insured.
- A52. Print on bill: name of addressee.
- A53. Print on bill: number and street, city and state of address.
- A54. Position typewriter to next billing form.

Exhibit 12, Figure 1: Flowchart



LEGEND

(For fuller explanation, see Exhibits 9 and 10.)

Sign	Meaning
MX	Main Sequence, order number X
X	Sub-Sequence, order number X
X - Y	Orders X through Y
o	Choice made by machine; arrow points to case where Choice Counter is positive
X - Y "A" Y "B"	Orders X through Y determine if Condition "A" or Condition "B" applies. Choice occurs at order number Y.
X - Y "A"	Orders X through Y result in Item "A".
X - Y "A"	Orders X through Y result in Item "A".
X - Y Description	Orders X through Y contain orders for printing, comparing, or selecting values as described

Exhibit 13

OUTPUT FROM MACHINE - DIVIDENDS NOT DEDUCTED FROM PREMIUMS, AND TOTALS OF DIVIDENDS AND PREMIUMS

1. List of Dividends not Deducted from Premiums [four dividends]

	Sample
Policy Number:	12075205
Dividend Option:	3
Amount of Dividend:	0014.50

2. Totals

Dividends Billed:	64.17
Gross Premiums Billed:	830.84
Dividends Not on Bills:	23.34

Exhibit 14

OUTPUT FROM MACHINE — BILLS

[This exhibit consists of 34 bills, produced by the machine, of which the following is a partial sample of one bill:]

Policy Number:	4761744
Date Due:	7
Payable:	1
Premium:	34.30
Dividend:	14.90
Amount Due:	19.40
Addressed To:	10203333333 370555066 777777777788

Exhibit 15

COMPARISON - BILLS AS TYPED IN COMPANY

[This exhibit consists of 34 bills, of which the following is a partial sample:]

Policy Number:	4761744
Date Due:	7 July 1947 ["July 1947" is imprinted, not typed]
Payable:	Q
Premium:	34.30
Dividend:	14.90
Amount Due:	19.40
Addressed To:	R. G. Tessler 87 Bay St. Manchester, N.H.

Multi-Access Forum

(Continued from page 17)

UNIVAC VETERANS PLAN SOCIAL GATHERING

An informal gathering of individuals associated with UNIVAC I and II Systems is being planned to coincide with the dates of the 1969 Spring Joint Computer Conference to be held in Boston, May 14-16. The purpose of the social gathering is to bring together as alumni the people who worked with these systems as employees of UNIVAC, or as users, or as commercial firms contracting with either of these in an era when electronic data processing was in its infancy.

Persons interested in helping to organize the event may contact: Lawrence Dorf, 408 Camino Verde, S. Pasadena, Calif. 91030; Jack Tauber, 152 Iroquois Ave., Oakland, N.J. 07436; or Noel Zakin, 110 Barnwell Dr., White Plains, N.Y. 10607. □

C.a

NUMBLES

Number Puzzles for Nimble Minds — and Computers

Neil Macdonald
Assistant Editor

A "numble" is an arithmetical problem in which: digits have been replaced by capital letters; and there are two messages, one which can be read right away and a second one in the digit cipher. The problem is to solve for the digits.

Each capital letter in the arithmetical problem stands for just one digit 0 to 9. A digit may be represented by more than one letter. The second message, which is expressed in numerical digits, is to be translated (using the same key) into letters so that it may be read; but the spelling uses puns or is otherwise irregular, to discourage cryptanalytic methods of deciphering.

We invite our readers to send us solutions, together with human programs or computer programs which will produce the solutions.

Numble 692:

```
        C A R E
  + B E H E A D S
                          C > D
  = B E A C H S E         U > L > B

     1842  355
```

Solution to Numble 691

In Numble 691 in our January issue, the digits 0 through 9 are represented by letters as follows:

A = 0	L = 5
E = 1	C = 6
I = 2	T = 7
H = 3	U = 8
M, N = 4	K = 9

The rest of the message — with disguise — in numerical form (which was omitted from the January issue in error) was: 73249 52277 551. The full message is: Talk much, think little.

Our thanks to the following individuals for submitting the correct solution to Numble 6812: A. Sanford Brown, Dallas, Tex.; T. Paul Finn, Indianapolis, Ind.; Daniel J. Glanz, King of Prussia, Pa.; Claude Grenier, Quebec, Can.; Robert C. Jensen, Endicott, N.Y.; R. L. Johnston, Sharon, Pa.; Richard T. Lynch, Madison, Wisc.; John McCormick, Milwaukee, Wisc.; William Phillips, Washington, D.C.; and D. F. Stevens, Berkeley, Calif. □

CALENDAR OF COMING EVENTS

Feb. 13-14, 1969: Assoc. of Data Processing Service Organizations Management Conference, Hotel Frontier, Las Vegas, Nev.; contact Jerome L. Dreyer, Assoc. of Data Processing Service Organizations, Inc., 420 Lexington Ave., New York, N.Y. 10017.

March 20-21, 1969: 7th Annual Atlantic Systems Conference, Americana Hotel, New York, N.Y.; contact Atlantic Systems Conference, P.O. Box 461, Pleasantville, N.Y. 10470

March 24-26, 1969: 10th VIM meeting, (users group of Control Data 6000 computer series), Florida State University Union, Tallahassee, Fla.; contact Carol J. Richardson, Control Data Corp., 8100 34th Ave. So., Minneapolis, Minn. 55440

March 24-27, 1969: IEEE International Convention & Exhibition, Coliseum and N.Y. Hilton Hotel, New York, N.Y.; contact IEEE Headquarters, 345 East 47th St., New York, N.Y. 10017

March 26-28, 1969: Seventh Annual Symposium on Biomathematics and Computer Science in the Life Sciences, Univ. of Texas Graduate School, Houston, Tex.; contact Office of the Dean, Univ. of Texas Graduate School of Biomedical Sciences, Div. of Continuing Education, P.O. Box 20367, Houston, Tex. 77025

March 26-29, 1969: 16th International Meeting of The Institute of Management Sciences, Hotel Commodore, New York, N.Y.; contact Granville R. Garguilo, Arthur Anderson & Co., 80 Pine St., New York, N.Y. 10005

April 1-3, 1969: Numerical Control Society's Sixth Annual Meeting & Technical Conference, Stouffer's Cincinnati Inn, Cincinnati, Ohio; contact Peter Senkiw, Advanced Computer Systems, Inc., 2185 South Dixie Ave., Dayton, Ohio 45409

April 15-18, 1969: The Institution of Electrical Engineers and the Institution of Electronic and Radio Engineers Computer Aided Design Conference, Southampton University, So 9, 5 NH., Hampshire, England; contact Conference Dept., IEE, Savoy Place, London, W.C.2

April 23-25, 1969: 21st Annual Southwestern IEEE Conference and Exhibition, San Antonio Convention and Exhibition Center, San Antonio, Texas; contact William E. Cory, Southwest Research Institute, Box 2296, San Antonio, Texas 78206

May 5-6, 1969: Association For Computing Machinery (ACM) Symposium on Theory of Computing, Marina del Rey Hotel, Marina del Rey, Calif.; contact Prof. Michael A. Harrison, Dept. of Computer Science, Univ. of California, Berkeley, Calif. 94720

May 6-9, 1969: The Association of Educational Data Systems (AEDS) Annual Convention, Portland Hilton Hotel, Portland, Ore.; contact Wayne J. Smith, Convention Contractor, 201 Massachusetts Ave., N.E., Washington, D.C. 20002

May 7-9, 1969: International Joint Conference on Artificial Intelligence, Statler-Hilton Hotel, Washington, D.C.; contact Dr. Donald E. Walker, IJCAI Program Chairman, The MITRE Corp., Bedford, Mass. 01730

May 14-16, 1969: Spring Joint Computer Conference, War Memorial Auditorium, Boston, Mass.; contact American Federation for Information Processing (AFIPS), 210 Summit Ave., Montvale, N.J. 07645

May 18-21, 1969: Power Industry Computer Application Conference, Brown Palace Hotel, Denver, Colorado; contact W. D. Trudgen, General Electric Co., 2255 W. Desert Cove Rd., P.O. Box 2918, Phoenix, Ariz. 85002

June 8-12, 1969: Sixth Annual Design Automation Workshop, Hotel Carillon, Miami Beach, Fla.; contact Dr. H. Freitag, IBM Watson Research Ctre., P.O. Box 218, Yorktown Heights, N.Y. 10598

June 9-11, 1969: IEEE International Communications Conference, University of Colorado, Boulder, Colo.; Dr. Martin Nesenbergs, Environmental Science Services Administration, Institute for Telecommunication Sciences, R614, Boulder, Colo. 80302

June 16-19, 1969: Data Processing Management Association (DPMA) 1969 Internat'l Data Processing Conference and Business Exposition, Montreal, Quebec, Canada; contact Mrs. Margaret Rafferty, DPMA, 505 Busse Hwy., Park Ridge, Ill. 60068

June 16-21, 1969: Fourth Congress of the International Federation of Automatic Control (IFAC), Warsaw, Poland; contact Organizing Comm. of the 4th IFAC Congress, P.O. Box 903, Czackiego 3/5, Warsaw 1, Poland.

June 17-19, 1969: IEEE Computer Group Conference, Leamington Hotel, Minneapolis, Minn.; contact Scott Foster, The Sheffield Group, Inc., 1104 Currie Ave., Minneapolis, Minn. 55403

June 19-20, 1969: Assoc. of Data Processing Service Organizations Management Conference, Sheraton Ritz Hotel, Minneapolis, Minn.; contact Jerome L. Dreyer, Assoc. of Data Processing Service Organizations, Inc., 420 Lexington Ave., New York, N.Y. 10017.

June 19-20, 1969: Seventh Annual Conference of the Special Interest Group, Computer Personnel Research of the Association of Computing Machinery, Univ. of Chicago, Chicago, Ill.; contact Dr. Charles D. Lothridge, General Electric Co., 570 Lexington Ave., New York, N.Y. 10022

June 21-28, 1969: Second Conference on Management Science for Transportation, Transportation Center at Northwestern University, 1818 Hinman Ave., Evanston, Ill. 60204; contact Page Townsley, Asst. Dir., Management Programs, 1818 Hinman Ave., Evanston, Ill.

June 30-July 3, 1969: Institution of Electrical Engineers Conference on Computer Science and Technology, Univ. of Manchester Institute of Science and Technology, Manchester, England; contact Conference Secretariat, Institution of Electrical Engineers, Savoy Place, London, W.C.2, England.

Aug. 6-8, 1969: Joint Automatic Control Conference, Univ. of Colorado, Boulder, Colorado; contact unknown at this time.

Aug. 11-15, 1969: Australian Computer Society, Fourth Australian Computer Conference, Adelaide Univ., Adelaide, South Australia; contact Dr. G. W. Hill, Prog. Comm. Chrmn., A.C.C.69, C/-C.S.I.R.O., Computing Science Bldg., Univ. of Adelaide, Adelaide, S. Australia 5000.

Aug. 25-29, 1969: Datafair 69 Symposium, Manchester, England; contact the British Computer Society, 23 Dorset Sq., London, N.W. 1, England

Sept. 8-12, 1969: International Symposium on Man-Machine Systems, St. John's College, Cambridge, England; contact Robert C. McLane, G-MMS Meetings Chairman, Honeywell Inc., 2345 Walnut St., St. Paul, Minn. 55113

Sept. 28-Oct. 1, 1969: Association for Systems Management International (formerly Systems and Procedures Association) International Systems Meeting, New York Hilton Hotel, New York City, N.Y.; contact Richard L. Irwin, Association for Systems Management, 24587 Bagley Rd., Cleveland, Ohio 44138.

Oct. 1-5, 1969: American Society for Information Science, 32nd Annual Meeting, San Francisco Hilton Hotel, San Francisco, Calif.; contact Charles P. Bourne, Programming Services, Inc., 999 Commercial St., Palo Alto, Calif. 94303.

Oct. 6-10, 1969: Second International Congress on Project Planning by Network Analysis, INTERNET 1969, International Congress Centre RAI, Amsterdam, the Netherlands; contact Local Secretariat, c/o Holland Organizing Centre, 16 Lange Voorhout, The Hague, the Netherlands

Oct. 22-24, 1969: IEEE 1969 Systems Science and Cybernetics Conference, Philadelphia, Pa.; contact C. Nelson Dorny, Moore School of Electrical Engineering, Univ. of Pa., Philadelphia, Pa. 19104.

Oct. 27-31, 1969: Business Equipment Manufacturers Assoc. (BEMA) Annual Business Equipment Exposition and Management Conference, New York Coliseum, Columbus Circle, New York, N.Y. 10023; contact Laurance C. Messick, Business Equipment Manufacturers Assoc., 235 East 42nd St., New York, N.Y. 10017

Oct. 30-31, 1969: Assoc. of Data Processing Service Organizations Management Conference, Regency Hyatt Hotel, Atlanta, Ga.; contact Jerome L. Dreyer, Assoc. of Data Processing Service Organizations, Inc., 420 Lexington Ave., New York, N.Y. 10017.

Nov. 18-20, 1969: Fall Joint Computer Conference, Convention Hall, Las Vegas, Nev.; contact American Federation for Information Processing (AFIPS), 345 E. 47th St., New York, N.Y. 10017 □

What Kind of Jobs? — Part 1

Business Equipment Manufacturers Assoc.
235 East 42 St.
New York, N.Y. 10017

Computer hardware and its accessories are made by more than one hundred companies. These companies employ many types of engineers, scientists, sales representatives, customer service engineers and technicians.

Computer programs or software are written by analysts, programmers, scientists, business experts and many other types of specialists working for computer manufacturers, computer users, and consulting firms.

But any attempt to pigeonhole an information-processing professional into one or the other of these two groups won't always work. Software people continually work in hardware areas, helping the engineer make the computer easier to use; and hardware engineers have to keep software requirements in mind when designing computer systems. Furthermore, when you get to the computer users, there's a whole group of people who belong in neither hardware nor software categories, but who are indispensable in the actual processing of the information: the computer operators.

In fact, versatility is the hallmark of the computer professional, and it is one of the reasons so many people find a career in information processing attractive.

Literally dozens of job titles exist for the various categories of work, since there is very little standardization in the industry. On the other hand, we can classify the types of work into general categories of activity and comment upon the level of education and training usually required.

Designing and Manufacturing

First there is *designing* and *manufacturing* the computer. This is not just developing improved circuits and bolting panels together. Computer manufacturers support a great deal of basic research in physics, chemistry, mathematics, and related sciences. They also employ scientists and engineers who carry on applied research in order to develop better techniques, materials, and equipment for computers. From this applied research come the ideas which development engineers work into specific products for use with or in the computer. *Research* and *Development* includes such areas as component and circuit design, communications, equipment design, semi-conductor and micro-electronic research. *Production Engineering* and *Quality Assurance* include developing production techniques, the design of production equipment, test equipment and test procedures, and also packaging design (an increasingly challenging problem in our age of progressive miniaturization).

Engineering jobs generally require a college degree. Some companies will hire high-school graduates with technical school training as technicians, engineering aides, or draftsmen. Technicians help engineers in such tasks as testing, building experimental circuits, and collecting data. In general, they handle much of the detail work for engineers. If they learn quickly and take part-time college courses, they may have the opportunity of advancing into more professional or supervisory jobs.

Based on a booklet entitled "Computer Careers," published by the Business Equipment Manufacturers Association in cooperation with the American Federation of Information Processing Societies. The booklet provides detailed information regarding careers in the designing, production, marketing and application of computers. Single copies are available without charge; 2 to 99 copies are 40¢ each, etc. Requests for additional price information and orders should be sent to BEMA at the address above.

Manufacturing computers generally requires higher education and skills than the manufacturing of other types of equipment. A U.S. Department of Labor researcher recently studied the skill requirements of computer manufacturing and found that white-collar workers make up half of plant employment. A college degree, or equivalent experience, is required for about 25 percent of those employed. The researcher concluded that although the data were not precise, it was clear that electronic computer manufacture increases the demand for high-level skills. Performance of research and development was the primary function of about 60 percent of the scientists and engineers. Managers and officials made up 3 percent of total employment; sales workers totaled 2 percent. One worker in seven was in the clerical group. Blue-collar jobs made up half of the total employment.

Marketing and Sales

Developing and manufacturing a computer, like making a better mouse trap, is just part of the job. Another major part is *selling* or *marketing*. The computer salesman is called a sales or marketing representative, and he is well trained in both marketing and computer applications. He may work with a potential customer for months or even years before he sells him a computer or an auxiliary piece of equipment. He may develop a computer system for a commercial business that requires a complete reorganization of the paperwork system used by the company. Then again, he may help set up a network of computers to tie together the scattered operations of a huge manufacturing complex or government agency.

The sales representative who specializes in business systems must be familiar with all phases of company organization and intricate details of modern management. This is a large area to cover; in fact, many sales representatives specialize in a single type of business such as manufacturing, distribution, or communications. They generally learn this intensive knowledge of their business specialty through courses and through working closely with customers in that particular business.

Other marketing representatives specialize in engineering or scientific applications. Such work requires a technical background that can enable the salesman to understand the information-processing needs of a customer and to recommend the precise configuration of equipment that will satisfy those needs.

Careers in computer sales are open to college graduates and to those who already have experience in sales in allied technical or management fields. Degrees may be in business administration, in any of the engineering or scientific disciplines, or in liberal arts. Most computer manufacturers give intensive sales training courses to their beginning salesmen — often 40 to 50 weeks of paid training, and sometimes even more. Such courses are followed by on-the-job training under a senior salesman. Training for those who sell auxiliary equipment is generally less intensive.

Earnings of marketing representatives are generally based on a salary plus a commission for each sale. Those who enter the computer marketing field after several years in business or technical positions generally start with higher c.a

PROOF GOOFS

Neil Macdonald
Assistant Editor

We print here actual proofreading errors in context as found in actual books; we print them concealed, as puzzles or problems. The correction that we think should have been made will be published in our next issue.

If you wish, send us a postcard stating what you think the correction should be.

We invite our readers to send in actual proofreading errors they find in books (not newspapers or magazines). Please send us: (1) the context for at least twenty lines before the error, then the error itself, then the context for at least twenty lines after the error; (2) the full citation of the book including edition and page of the error (for verification); and (3) on a separate sheet the correction that you propose.

We also invite discussion from our readers of how catching of proofreading errors could be practically programmed on a computer.

For more comment on this subject, see the editorial in the September 1968 issue of *Computers and Automation.*

Proof Goof 692

(Find one or more proofreading errors.)

The peasant has always been the key to pacification. Previous programs had overlooked the peasant's role and underestimated the importance of his indifference to pacification efforts, and thus the peasant continued to support the Vietcong. The Vietcong had, by and large, not offered material rewards to the peasant but nevertheless were able to engender both tacit and overt support. This support was not achieved by terrorizing the bulk of the peasant population but rather sprang from the empathy the peasant had with the Vietcong.

Since this was the fundamental obstacle to overcome in initiating a pacification program, it was necessary to determine an alternative to the ideological appeal of the National Liberation Front, and in order to do so certain fundamental elements of the NLF ideology had to be understood.

The ideological approach of the Vietcong has been based on the peasant's hamlet-centered world. It is imperative to differentiate clearly between the hamlet and the village. The village is a geographical distinction much like the area of a county in the United States. Within its arbitrary boundaries there may be two to ten separate hamlets. Where the village is geographic the hamlet is an ethnic and cultural entity. *Parochialism in Vietnam originates at the hamlet level and this has been the focal point of NLF propaganda.* The NLF strives constantly to plant a picture in the minds of the peasants that the guerrillas are "their" boys who have been born or brought up in their hamlet or in one similar to it. Another point NLF propaganda makes is that the aims of the guerrillas must be good because they are fighting for the people against such traditional peasant enemies as the rich city people, the absentee landlords, corrupt government officials, and now finally the strange-talking American. The Americans receive the lion's share of the propaganda attacks because so much destruction is associated with their presence in the war against the Vietcong.

The Vietnamese peasants are aware that "their" boys are acting tough. They are aware of all the things we stress in our propaganda — the VC impress youngsters into service, the VC taxes are high, VC presence in a village heightens the changes of U.S. military action in the area. However, in spite of all this, the peasants continue to maintain empathy with "their" boys.

The Vietnamese peasant has a strong desire to survive and more often than not hopes the Vietcong will win because he imagines a Vietcong victory will eradicate the conditions he currently faces. Our experience showed that the Vietnamese peasant will help the Vietcong when there is not too much risk of doing so and that in the great majority of cases the peasant considers it unthinkable to betray the Vietcong to the enemy. The peasant generally does not see himself as a coward, and he rationalizes away all such considerations by his belief in the fact that he is a Vietnamese, the Vietcong are Vietnamese, and that therefore the Vietcong are "good." Fortunately hamlet-nationalism, because of the passive personality of the peasant, has not yet resulted in a violent xenophobic reaction against the U.S. troops. Normal resentment has been evident but not the all-consuming features of virulent xenophobia.

We also noted that in addition to his nationalism, the Vietnamese peasant was even more strongly motivated by hunger. The gnawing fear of a bad harvest, the destruction of a rice crop by crop-destroying chemicals or U.S. military operations, and the absentee landlords all combined to make the Vietcong propaganda credible to the peasant. *In many areas pacification had not only brought American troops, but also meant the return of the absentee landlords,* who made use of the peasants' gullibility and fear of authority to equate the power of the U.S. forces with the authority to enforce collection of illegal land rents. In most cases the U.S. forces were totally unaware of the exhorbitant land-rent practices in the areas they were controlling.

— From *The Betrayal* by William R. Corson, pp. 160-162, published by W. W. Norton & Co., Inc., 55 Fifth Ave., New York, N.Y. 10003, 1968, 317 pp.

Solution to Proof Goof 691:

Paragraph 3, line 10: Replace "day" with "way." ☐

---

salaries. Many successful computer marketing representatives earn incomes comparable to top industry executives at the level of vice president and president.

Customer or Field Engineering

Once a computer is sold, it must be installed and then serviced periodically (often daily) in order to keep it operating satisfactorily. This is the job of the maintenance engineer, or, as he is often called, the *customer or "field" engineer.* Being out in the "field" means working at the customer's computer installation. The ability to work well with customer personnel is essential. A college degree in engineering is very helpful, but usually is not mandatory. Factory training is usually provided by the employer, but most employers require previous experience or technical training in electronics. The maintenance or field engineer usually works under considerable pressure, since each hour that a computer is out of service delays vital information and is very costly for the computer user. Salary increases are frequent within the first two or three years following a training period of three to five months, but salaries tend to stabilize after a few years. College graduates are usually selected for supervisory and managerial positions. ☐

Part 2 of "What Kinds of Jobs?" will appear in the next issue.

ACROSS THE EDITOR'S DESK

Computing and Data Processing Newsletter

Table of Contents

APPLICATIONS

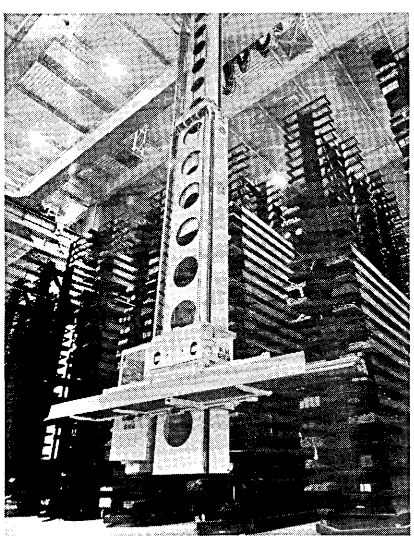
Cover Story

COMPUTERIZED STACKER CRANES MOVE 5400 TONS OF STEEL PER DAY

Two high-speed unmanned cranes are moving bars and tubes of metal in and out of inventory at the automated steel bar and tubing service center recently opened at the Chicago plant of Joseph T. Ryerson & Son, Inc. The center's computer-

ized materials handling system controls the cranes by feeding punched cards into a central computer console as shown in the picture above.

Material is stacked in racks 40½ feet high which contain 6700 pockets with a total capacity in each rack of 5 tons. Material in the stack racks is stored in metal containers or "pans" which can accommodate lengths up to 24 feet. Each stacker crane can be programmed to bring a pan out (see picture below) to any one of ten stations in the process-



ing and order assembly area. Included in each area are a central order-picking station with five-pan capacity for sequential order processing, and two cold saw stations, each with three-pan capacity.

The precision of the automated control permits the cranes to operate safely at an average cycle time of 3.6 minutes. This includes going to the proper pocket, bringing the material to the cutting and loading area, and taking the contents of another pocket back to inventory. When the crane is programmed to work only one side of one aisle, cycle time is cut to an average of only 2.9 minutes. The unmanned cranes will save at least 160 man hours per three-shift day compared to the operation of a conventional plant of the same capacity.

The crane bridge travels at 400 f.p.m., the trolley at 200 f.p.m. and the fork hoist at 60 f.p.m. Optimum performance in a 14-hour order-processing day would total 540 cycles for the two cranes. Figuring a 5-ton load in and out, this would result in the movement of 5400 tons of material per day. For the balance of the day the cranes are used for inventory receiving and for order loading operations with their auxiliary hoists.

"EARTHQUAKES" PRODUCED ON A COMPUTER TEST STRUCTURAL DESIGN OF NEW BUILDINGS

Engineers at the California Institute of Technology, Pasadena, Calif., are producing earthquakes on a computer to test the ability of newly designed structures to withstand strong earth tremors in real life. In their computer-simulated earthquakes, the Caltech engineers — P. C. Jennings, G. W. Housuer and N. C. Tsai — record the frequency, direction and intensity of each tremor in a record known as an accelogram. These records are just the same as those recorded for real earthquakes. The programmed accelograms from the computer make it possible to analyze mathematically the effect that the repeated shaking, resulting from a quake of a particular strength, has on different building designs.

The engineers have modeled four types of earthquakes which they believe to be most significant for studies on structural design. The first represents a catastrophic quake such as the Alaskan one in 1964. It models the shaking of the ground near to a major fracture in the earth's crust. The second model is shaking close to the fault during a severe quake of the type that struck Puget Sound in Washington in 1949 and 1965. This type of earthquake can also cause great damage to property.

The other two types of quake simulated are much less severe in action, but are likely to occur often during the lifetime of a building. Such tremors, although they may not cause immediate damage to buildings, could produce cumulative effects which would be expensive to repair. Their computer simulation allows designers to run their structures through a number of these tremors in quick time to discover the damage that in real life would only become evident after a number of years.

THE UNIVERSITY OF MONTREAL "WEIGHS" COURSES BY COMPUTER

Recognizing that all courses taken by a student do not have equal value to him, the University of Montreal (Canada) "weighs" its courses — on a Honeywell Model 200 computer. Students sign up for courses and then the faculty determines a "weighting factor" for each, based upon its value to the student. For example, chemistry probably is more important to a medical student than is English literature and so is given a larger weighting factor.

All marks from tests and examinations are fed to the computer system and then multiplied by the weighting factor for each course. The computer uses the "weighed" grades to determine an over-all mean average for each student, and the student wins his degree by attaining a passing mean average of 60.

COMPUTER HELPS ANALYZE DREAMS TO AID IN PSYCHIATRIC CARE

Dreams of patients under psychiatric care are being studied with the aid of a computer at the C. J. Jung Institute in San Francisco, Calif. Psychiatrists, psychologists and volunteer patients are using the computer for research in conjunction with analytical psychotherapy.

Written reports or transcribed tape recordings, describing a patient's dream, are typed into a CALL/360 DATATEXT terminal at the C. J. Jung Institute, which is connected by telephone lines to the distant computer. As the program develops, it is anticipated that the computer will be able in a matter of seconds to classify the dream description, helping the psychiatrist discern a pattern that may lead to a better understanding of the patient. The institute also is using the computer system to establish an archive of psychological information, including dreams, fantasies, test data and therapy reports.

COMPUTERIZED SYSTEM FOR CAREER INFORMATION AIDS STUDENTS AND COUNSELORS

Willowbrook High School in Villa Park, Ill., has created an electronic "library" of career information to help students explore vocational opportunities and to provide them with facts upon which to base mature vocational decisions. An IBM system provides students with data on the training and academic preparation required for more than 400 vocations. It also stores, for use by authorized personnel, information on grades and the results of interest and achievement tests for the 1,700 sophomores, juniors and seniors at the coeducational suburban Chicago school.

The program, called Computerized Vocational Information System (CVIS), was developed by the school's guidance staff over a period of two years. CVIS gives students access to the career data through IBM 2260 visual display stations such as the one shown in the picture. The

terminals are linked by telephone lines to an IBM System/360 Model 30 computer at the nearby College of DuPage which is cooperating with Willowbrook in the project. Students in the upper three grades can use CVIS during free periods.

In addition to the primary goal of presenting to students an organized system for vocational decision-making, a secondary goal is to make information about occupations and students more readily available to the counseling staff. Counselors will have visual terminals in their offices, so they can retrieve the records of students they are advising as well as the occupational briefs in the system. Later, they will have access to information about colleges, technical schools, apprenticeships and local job opportunities.

EDUCATION NEWS

"CONVERSATIONS" WITH COMPUTERS FOUND TO STRENGTHEN LANGUAGE SKILLS

A large-scale experiment in Computer-Assisted Instruction (CAI) at the Stony Brook campus of the State University of New York was the basis of a report presented at a recent meeting of the American Association for the Advancement of Science by Dr. E. N. Adams of International Business Machines Corp. In the experiment, a group of first-year German students supplemented their regular classroom instruction with "conversations" with a computer. Test scores indicate that these students performed far better in reading and writing German than a control group of students taught by conventional methods. Moreover, the improvement was greatest among students in the bottom half of the class.

All 250 Stony Brook students who registered for first-year German in the 1967-68 class were involved in the experiment. Of the total, 141 students were assigned to the regular college course. The other 109 students supplemented regular German instruction with "conversations" with a computer. After two full semesters ending June 1968, the performances of the two groups were compared by their scores on a standard achievement test: The Modern Language Association Cooperative Foreign Language Tests.

The tests showed that the two groups were similar in speaking and understanding spoken German. However, the CAI students were markedly superior in reading and writing the language with 85% of the CAI students scoring as high or higher than the average student in the regular class. Furthermore, the weaker students were helped the most, confirming a trend observed in earlier, smaller-scale CAI experiments.

BOSTON UNIVERSITY'S SCHOOL OF LAW IS PRESENTING A COURSE ON INTERACTION BETWEEN COMPUTERS AND THE LAW

Boston University's Law School (Boston, Mass.) is presenting what is believed to be the first course on the interaction between computers and the law. The course is being taught by Roy Freed, a Yale Law School graduate and senior consultant at Harbridge House, Inc., an international management consulting firm in Boston. Mr. Freed pioneered the investigation of special legal questions raised by the use of computers for record-keeping, factory operation, clinical medical procedures and the myriad other applications in business, industry, the professions, and government. His more than 50 articles on computers and law make him perhaps the most prolific author in that area.

The course will cover such topics as how novel types of computer records will fare in the courts, how reliance upon computers instead of people affects legal liability, what problems confront banks that turn to computers (and their depositors, as well), where computers fit into the many facets of securities marketing, how to avoid pitfalls when buying or leasing computers, whether existing criminal laws cover misdeeds involving computers, and how computerization can streamline court operations.

TRAINING IN BASIC SYSTEMS FOR PROGRAMMERS

The Basic Systems Course for Programmers, offered by Systemation, Inc., Colorado Springs, Colo., is a twenty-five week correspondence course designed to qualify a programmer to do administrative systems work in the data processing department. The course includes 17 basic areas of administrative systems work which are covered in 125 study units. Modern methods of programmed teaching are used to tie in the man's learning experience with immediate on-the-job application of systems.

The course was written by Leslie H. Matthies, Executive Director of The Foundation for Administrative Research and W. Norman Wood, Assistant Director.
(For more information, designate #41 on the Reader Service Card.)

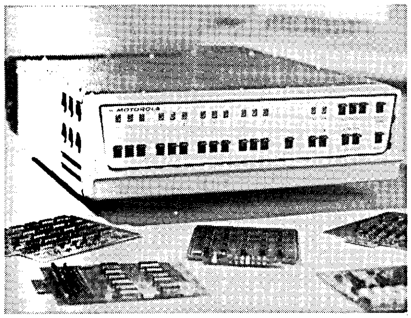
NEW PRODUCTS

Digital

MDP-1000 COMPUTER / Motorola Instrumentation and Control Inc.

A new, small-scale digital computer selling for about $8,000 is now available from Motorola Instrumentation and Control Inc., subsidiary of Motorola Inc., Phoenix, Ariz. The new computer, the MDP-1000, is designed primarily for systems applications.

Among its features are a wide variety of input/output options; a 4,096-word random-access memory (expandable to 16K) with a cycle time of 2.16 usec; six programmable



12-bit registers; parallel adder; two accumulators; and a priority interrupt system. A line of interface modules also is available with the new machine.

Software includes routines for assembly, debugging, utility tasks, and a math program.
(For more information, designate #42 on the Reader Service Card.)

PDP-12 COMPUTER / Digital Equipment Corp.

The PDP-12, designed for laboratory applications, is Digital Equipment Corporation's (Maynard, Mass.) newest computer system. While the new system was designed with the laboratory in mind, Edward Kramer, PDP-12 Marketing Manager, stressed that it is fully general purpose. Potential application areas include bio-medicine, oceanography, chemistry, physics, education and a variety of industrial environments.

The PDP-12 (PDP for programmed digital processor) in its computer (LINC) configuration as a laboratory instrument, includes a central processor with a 4,096-word memory expandable to 32,768 words, two magnetic tape storage units, a seven-by-nine inch cathode ray tube display, a 16-channel analog-to-digital converter and multiplexer, a data terminal, a teletypewriter and paper tape reader and punch.

Unlike the LINC-8, the PDP-12 does not have two central processors. The LINC-8 was like two computers, so designed to give operators more flexibility. Despite one central processor, the PDP-12 will run the programs available for the LINC-8 and DEC's PDP-8 series of computers. This includes a wide range of applications packages, as well as those for such things as plotting and manipulating data, signal averaging and frequency.
(For more information, designate #43 on the Reader Service Card.)

GE-105 COMPUTER / General Electric Company

A new small-scale information processing system has been announced by General Electric Company's Small Computer Marketing Operation, Bridgeport, Conn. The new system, designated the GE-105, will be marketed to small businesses "on the grow." The GE-105 joins the GE-115 and 130 to expand the GE-100 line.

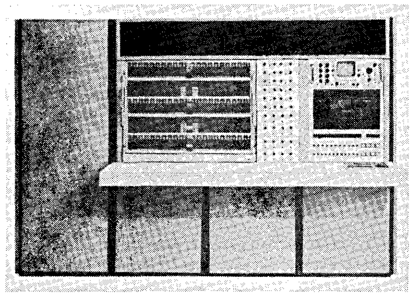
Two models are available: GE-105 Model A has a 7.5 microsecond memory cycle; 4,096 or 8,192 store locations; 350 card-per-minute card reader; 250 line-per-minute printer; and 60 to 200 card-per-minute card punch. The GE-105 Model B has a 7.5 microsecond memory cycle; 8,192 store locations; 300 line-per-minute printer; and a 300 card-per-minute card reader/punch.

Systems software available for the GE-105 will include an assembly programming system, an extended report program generator called LOGEL, utility programs such as List and Summarize and Reproduce and Gang-punch, and many general routines and subroutines. The new system will be used in such applications as invoicing, inventory control, payroll, purchasing, general accounting, production control, public utility services, etc.
(For more information, designate #44 on the Reader Service Card.)

Analog

COMCOR-550 COMPUTER / Astrodata Inc.

A new, medium-sized analog computer, Comcor-550, has been developed by Astrodata, Inc., Anaheim, Calif. The new computer is a solid-state, 100-volt machine, specifically designed for expansion to a hybrid configuration. It is compatible with such digital equipment as CDC, IBM, SDS, and Honeywell computers.



The new 550 (shown above) uses only one patchpanel. Analog and digital patching facilities are combined in different areas of the same board. The Comcor "integrated patchpanel", provides easy storage, quick programming, and reduced costs.

For hybrid configurations, both the Comcor Intracom interface unit and Astrodata software developed initially for the C-5000, are fully compatible with the 550. In addition, Astrodata programmers have developed and successfully tested a series of comprehensive hybrid software packages.
(For more information, designate #45 on the Reader Service Card.)

Special Purpose Systems

REAL-TIME PROCESS CONTROL SYSTEM / Machine Control Co.

Machine Control Co., Bedford, Mass., has introduced an economical computer-directed system for real-time process automation — Computa-Trol. Computa-Trol can be installed and on-line with little additional engineering required, since the basic package is interfaced and multi-plexed for almost immediate applications.

The hardware of Computa-Trol handles addressing, interrogation, multi-plexing, and maintenance of general housekeeping functions relating system inputs and outputs. The package contains a 4K core computer (expandable to 8K capacity) with 1.6 microsecond instruction time, a real-time clock, data logging capability via teletype unit ASR-33, one millisecond A/D ten-channel input multi-plexor accepting ±10 V analog signals and multiplexed stepper motor outputs. Standard outputs direct from one to twenty-four stepper motors, clockwise or counter clockwise, via time-shared, real-time control.

A basic Computa-Trol package includes the initial software that inter-relates inputs and outputs under computer control. For specific process needs, some further programming is generally required, at additional cost.
(For more information, designate #50 on the Reader Service Card.)

AUTOMATED SYSTEM FOR GAS ANALYSIS / Varian Associates

A new automated system for the continuous on-stream monitoring of up to 26 gases with simultaneous computerized output of qualitative and quantitative data has been developed by Varian Associates, Palo Alto, Calif. The automated gas analysis system will resolve the composition of complex atmospheres with one part per million accuracy according to J. W. Fowler, Varian Associates' Vacuum Division Product Manager.

The new gas analysis systems are built in modules allowing their acquisition on a building block basis in regard to automated control and data processing and in regard to monitoring at multiple locations. The five key elements in the gas analysis system are: (1) the gas sampling and vacuum system; (2) the quadrupole residual gas analyzer; (3) an automatic data acquisition system; (4) a compact, low-cost digital computer (Varian's 620/i); and (5) special computer programs.

Possible applications include: monitoring air and water pollution both ambient and at the source; in observing and controlling critical industrial processes; in governing combustion chambers and furnaces; and in medical applications, technical manufacturing, and research.
(For more information, designate #60 on the Reader Service Card.)

Memories

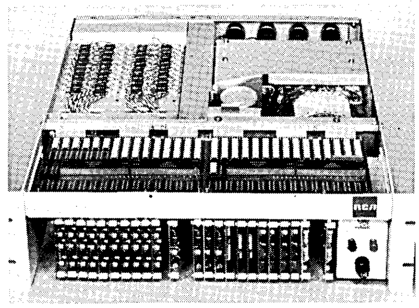
DRUM MEMORY SYSTEM / Bryant - Computer Products

Bryant Computer Products of Walled Lake, Mich., has announced a new compact, low-cost, ten-inch rotating drum memory system, called the CLC-1. The device can store up to 1.2 million bits of data and has an average access time of 8.5 milliseconds. The CLC-1 has four multiple pole-piece head assemblies, each containing eight write/read heads on 32 data tracks, operating at 3600 rpms. It is sold complete with a digital interface making it suitable for a wide variety of data processing systems and applications.
(For more information, designate #46 on the Reader Service Card.)

750-NANOSECOND CORE MEMORY / RCA

The RCA MS3370 is the newest member of the RCA line of "off-the-shelf" three-wire, 3-D, coincident-current memory systems. Manufactured by RCA/Electronic Components, Needham Heights, Mass., the system has a full-cycle time of 750 nanoseconds and access times as fast as 290 nanoseconds.

System capacity includes 4,096 words x 4 bits to 16,384 words x 40 bits using a standard "building block" approach. Capacities up to 131,072 words are provided by paralleling 16,384 word modules.



The system is modular in concept with a single memory plane design covering the entire line. Integrated circuits are used in the memory address register, memory information register and control and timing circuits. The memory core stack is a pluggable module and the complete power supply can be removed as a single unit.
(For more information, designate #47 on the Reader Service Card.)

DISC MEMORIES / Data Disc, Inc.

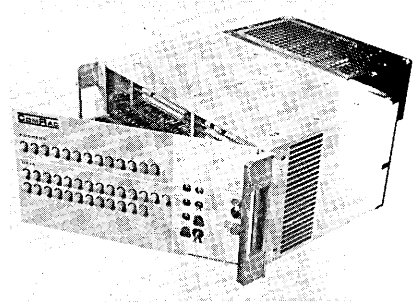
Data Disc, Inc., Palo Alto, Calif., has developed a new series of head-per-track disc memories that interface directly with all types of commercial digital logic: TTL, DTL and RTL. The memories use a proprietary "in-contact" recording technique to store up to 6.4 million bits on a single disc surface, double the capacity possible with conventional "floating head" techniques. The greater storage density permits savings up to 35% in equipment costs.

Average access time is 16.7 ms, and data transfer rates can be as high as 3 million bits per second. A complete memory system occupies only 8-3/4" of rack space. A separate power supply is available as an option. Currently available are: Model F6 (6.4 million bits); Model F3 (3.2 million bits), Model F1.5 (1.6 million bits), and Model F.75 (0.8 million bits).
(For more information, designate #48 on the Reader Service Card.)

RANDOM ACCESS CORE MEMORY SYSTEM / Information Control Corp.

Information Control Corporation, El Segundo, Calif., has added a new Commercial Random Access Core Memory System to their product line. The ComRac 200 memory has a 1 usec. full cycle time with bit capacities up to 8K x 36 or 4K x 72 in an 8-3/4" high, 19" rack mount chassis.

The memory has front panel access to the plug-in assemblies. The ComRac memory system incorporates



3D selection, fast switching, 20 mil lithium core and integrated circuits.

Operating temperature range is 0° to 50°C. Standard memory options are available.
(For more information, designate #49 on the Reader Service Card.)

Software

ANALYZE / Computer Resources Corp., McLean, Va. / Developed for general use by programmers, DP managers, and computer facilities to interface between the users program and the IBM-360 DOS/TOS supervisor. The program is designed to stop premature cancellation of computer programs because of defects in data, or programming errors. The ANALYZE Routine automatically intercepts program checks or operator interrupts and transfers control to the operator, who may select any of several actions to correct the error and resume processing if desired. The package includes a program listing, load deck, and instructions for usage.
(For more information, designate #51 on the Reader Service Card.)

COGO-10 / Applied Logic Corporation, Princeton, N.J. / A time-sharing version of COGO-90 (the civil engineering problem-oriented language), the language can be used in office and on-the-job locations to find immediate answers for geometric problems in control, land and right-of-way surveys; highway and interchange design; construction layout and bridge geometry. No previous computer programming knowledge is required. Format control permits output to be easily read and associated with corresponding input data. COGO-10 is available on a time-shared basis through the AL/COM time-sharing network.
(For more information, designate #52 on the Reader Service Card.)

CROSSTABS / Boston Computer Software Corp., Boston, Mass. / A research tool for marketing, media and operations analysis, CROSSTABS handles studies and surveys of any size or operates on existing data bases to produce easily usable statistical tabulations. The system is designed for use with the IBM System/360 and can be utilized by clerical assistants with only a few hours training. Price of CROSSTABS is $9500 and includes installation and training in its use.
(For more information, designate #53 on the Reader Service Card.)

LPS/360 and LPS/1130 / IBM Corporation, White Plains, N.Y. / Two new linear programming packages, based on the Linear Programming-Mathematical Optimization Subroutine System (LP-MOSS) originally developed for the 1130, for users of System/360 Models 25, 30 and 40 and the 1130 computing system. The new packages will have input data compatibility with LP-MOSS and each other. Linear programming is a technique used to determine the optimum allocation and blending of resources such as capital, raw materials, manpower and plant facilities.
(For more information, designate #54 on the Reader Service Card.)

MAID (Merger Acquisition Improved Decision) / Economatics, Pasadena, Calif. / The new proprietary software program MAID analyzes countless questions that must be considered when contemplating a merger. MAID answers specific questions for short-term rather than long-term effects. Written in FORTRAN IV, it can operate on any second or third generation computer. The price of $12,500 covers the FORTRAN deck, a users manual, and a demonstration on the customer's computer or at a computing facility.
(For more information, designate #55 on the Reader Service Card.)

MI FORTRAN for the IBM 1130 / Daniel Brocklebank, Easton, Pa. / MI (Multiple Integer) Fortran is a new programming language for the IBM 1130 designed for disk-oriented systems; it simplifies the coding of commercial problems using the familiar Fortran syntax (eliminating the need for "commercial subroutines"). Some of the features are: 15 digit decimal arithmetic using standard Fortran statements; automatic I/O overlap; 110 lpm numeric printing on the 1132; I/O error detection statements; and money editing formats. The price of $600 includes documentation, instructional materials, and support.
(For more information, designate #56 on the Reader Service Card.)

PHOTOSTET / Honeywell Electronic Data Processing, Wellesley Hills, Mass. / This generalized computer application provides high-performance hyphenization, justification and file maintenance for photocomposition typesetting mcahines. The system may be used to set straight matter or intricate retail display advertising copy for newspapers and for high production straight matter work by book, magazine and catalog publishers. Minimum computer configuration required to use the Photostet system is a small-scale Honeywell Model 120 central processor with 28,672 characters of main memory for drum storage or 32,768 characters for disk storage; a Type 270A random access drum or Type 258/259 disk pack drive; three half-inch magnetic tape drives; paper tape reader and paper tape punch; punched card reader and high-speed line printer with 132 print positions.
(For more information, designate #57 on the Reader Service Card.)

SCERT 50 / Comress, Inc., Washington D.C. / An advanced version of the SCERT (Systems and Computers Evaluation and Review Technique) proprietary simulation program, SCERT 50 has increased speed and capacity, as well as expanded input validation and diagnostics. It requires 131K core memory and is designed for native-mode use on UNIVAC 1108, IBM Series 360, Models 40 and larger, as well as for the RCA Spectra 70 Systems, Models 45 and larger. SCERT 50 is compatible with, but not dependent on, SCERT V and earlier versions of the simulation program, allowing users to phase to the more powerful simulator as the need arises.
(For more information, designate #58 on the Reader Service Card.)

WHOLESALE GOODS INVENTORY CONTROL SYSTEM / Delta Data Systems, College Park, Md. / Designed to provide wholesalers with a comprehensive tool that will aid management in the control of high inventory costs, the system is capable of processing twenty-one types of inventory transactions. A complete Purchase Order Subsystem is included. The system produces 19 reports which aid in controlling inventory levels and purchases and in making management decisions. The package, containing 20 COBOL programs, has been written for operation on an IBM 360 disk oriented computer system.
(For more information, designate #59 on the Reader Service Card.)

Peripheral Equipment

THERMAL PAGE PRINTER / The National Cash Register Company

A thermal page printer which is silent and virtually maintenance-free is being introduced for commercial applications by the National Cash Register Company, Dayton, Ohio. The system is an adaptation of thermal printer models which NCR has produced in quantity for high-performance military applications.

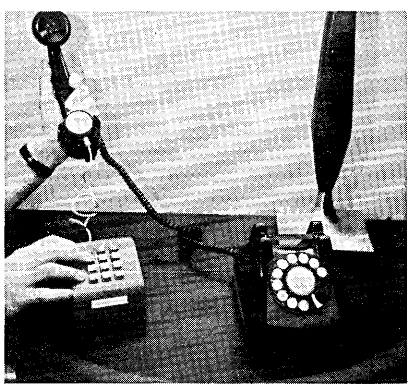
The printing action requires only heat, which is applied by a matrix-type print head to NCR-developed thermal paper, thereby forming numerals, alphabetic characters or other symbols. Up to 300 words a minute can be printed by the unit, which weighs only 11 pounds and measures 3-1/2 x 11-1/2 x 9-3/4 inches. There is no ribbon to change, no type to clean, and no print deterioration. The only moving part is the paper-advance mechanism. All machine functions are controlled by solid-state circuitry. Integrated circuits are used for buffering and decoding.

The new printer can be adapted to a variety of applications including mobile communications and as a low-cost page printer for general small-systems use.
(For more information, designate #61 on the Reader Service Card.)

TELEPHONE COMPUTER TERMINALS: PORTABLE and ACOUSTIC-COUPLED / Metroprocessing Corporation of America

The first product in its FONE-TONE line of portable acoustic-coupled telephone computer terminals has been announced by Metroprocessing Corporation of America, White Plains, N.Y. The terminal, called FT-1200 "SPARTAN" is a battery-operated device with the standard twelve-button dial for sending letters and numbers, as used in the modern Touch-Tone telephones. It includes an amplifier for hearing the voice or tone responses sent back over the telephone by the computer. "SPARTAN" can quickly be slipped on to any ordinary telephone in a few seconds, at the office or at home — or even in a phone booth.

With the "SPARTAN" terminal, every ordinary telephone can at once be used to transmit data to a computer system. The inputs and outputs from the portable unit are identical to those from the regular twelve-button phones installed by the telephone company. There is no need for the traveling man to use different procedures or codes than used by the employees in the home office.



— FONE-TONE "SPARTAN" — This minimum configuration shows the Dial Unit and Sound Unit (which slips onto the mouthpiece of an old-fashioned 1940 telephone) forming a complete low-cost terminal.

Ordinary transistor radio batteries power the "SPARTAN" terminal. It is never necessary to plug the terminal into an electric outlet, or to have a direct connection to the telephone wires. Complete with carrying case, the "SPARTAN" weighs only five pounds. It sells for $210 to $235, depending on the quantities ordered.
(For more information, designate #62 on the Reader Service Card.)

COMMUNICATIONS TERMINAL / Scientific Control Corp.

The versatile DCT-132 remote communications terminal, developed by Scientific Control Corp., Dallas, Texas, is designed to provide a more effective method of transmitting data from remote locations to large scale computer systems for instantaneous processing and return. With the new terminal, data can be transmitted over common carrier or dedicated lines at 2000 to 4800 bits per second. The DCT-132 may be connected with a printer, card reader, card punch or teletype.

The new DCT-132 can accept data for transmission to a computer from card, paper tape or keyboard. It is capable of receiving data from a computer and recording it as high-speed printer hard copy, keyboard copy, paper tape or punched cards. The device can transmit and receive data simultaneously if a full duplex line is used.

The modular design of the DCT-132 allows the peripheral controllers and peripheral devices to be added only when the peripherals are required.
(For more information, designate #63 on the Reader Service Card.)

PORTABLE TELETYPEWRITER / General Electric Company

A new, portable teletypewriter, the DATANET-730®, for use in communicating with a time-sharing computer over normal telephone lines, has been introduced by General Electric Co., Phoenix, Ariz. The DATANET-730 essentially is a tabletop keyboard set that looks like a large portable typewriter. It comes as a keyboard manual send/receive unit, or as an automatic send/receive unit, equipped with a punched paper tape option for transmission of messages.

The complete terminal comprises the portable teletypewriter; an acoustic coupler which serves the telephone handset; and a conventional telephone for immediate access to a time-sharing computer. It employs a standard Model 33 teletypewriter character set and keyboard layout.

Communication is initiated by plugging the DATANET-730 into any 115-volt a-c outlet. The computer is then dialed by telephone. Then the receiver is cradled in the acoustic coupler, and tone signals feed data back through the teletypewriter at 100 words a minute. Three models of the new terminal are available.
(For more information, designate #64 on the Reader Service Card.)

MICROFILM RECORDER / Information International

Information International of Los Angeles, Calif., and Cambridge, Mass., has introduced a very fast microfilm recorder, offering four times the highest resolution now available. The recorder, called the FR-80, generates graphic information on microfilm directly from digital output. The FR-80 competes as off-line computer output with line printers and digital plotters.

It has a resolution of 16,384 by 16,384 programmable points, better than or equal to that of plotters up to and including large four-foot flat bed models. The FR-80 produces 20,000 lines per minute — or about 10 times that of a typical IBM System 360/30 using three magnetic tape units and two line printers. The FR-80 accepts formatted tapes for the Stromberg Datagraphic 4020, 4060, and 4400, for California Computer Products plotters, and for IBM 360/1401 impact printers.

The recorder has a variety of character styles, and a choice of 64 programmable character sizes, eight programmable intensity levels, five programmable character rotations, and eight programmable spot sizes. Speed is 10,000 characters per second, with an optional 100,000 characters per second. The FR-80 has a comprehensive, system-oriented software package with on-line editing, debugging and paging.

The system records on both 16 and 35 mm sprocketed and unsprocketed film. First deliveries are scheduled for the spring of 1969.
(For more information, designate #65 on the Reader Service Card.)

DATA ENTRY SYSTEM: DISC STORAGE FILE / Logic Corporation

Logic Corporation, Haddonfield, N.J., has introduced a data input system which utilizes a direct keyboard-to-magnetic disc storage file. The LC-720-Disc Data Entry System consists of standard keyboard input terminals, a central processor, which will accept data from as many as 120 keyboards simultaneously, and an IBM 2311 disc pack drive.

The basic LC-720 Data Entry System introduced a few months ago (see Computers and Automation, November, 1969, p. 54) uses a magnetic tape recorder as the storage medium. The addition of the magnetic disc to the system gives it the advantages of bulk storage and high speed access of data for computer entry. Information transfer rate of the disc is 156,000 characters per second — twice as fast as the transfer rate for magnetic tape.

The LC-720-Disc System also offers simultaneous entry and verification of data by two different operators. Record size is from 1 to 240 characters long. A large library of up to 30 programs may be stored in the system and each is available simultaneously to all operators.
(For more information, designate #66 on the Reader Service Card.)

DIGITAL PLOTTER FOR COMPUTER GRAPHICS / Houston Instrument

The latest in the CØMPLØT® series of plotters, the DP-5, now is available from Houston Instrument, Division of Bausch & Lomb, Bellaire, Texas. (The CØMPLØT series of plotters includes the 300 increments/second DP-1 and the 22 inch wide DP-3.)

The new DP-5, having 1200 increments/second continuous speed on both axes, is specifically designed for computer graphics. It will operate on-line, off-line, or with a time sharing terminal.

Programming for the DP-5 remains the same as conventional 300 increments/second systems. This is possible because the pen up/down rate also has been increased by a factor of four (25 ms vs 100 ms).
(For more information, designate #67 on the Reader Service Card.)

Data Processing Accessories

COMPUTER USAGE TIME RECORDER / Datachron Corp.

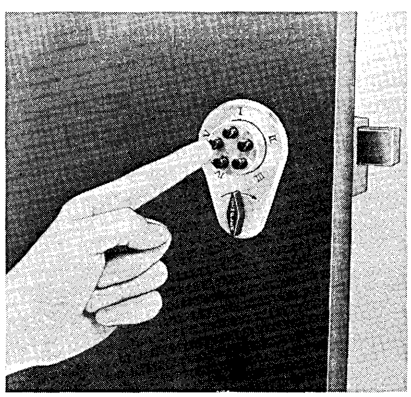
A new company in the computer accessory field, Datachron Corp., New York, N.Y., has announced its first product, a computer usage time recorder, the DATAPRINTER 101. The DATAPRINTER 101 can be interfaced with any computer and produces a printed record of when and for how long the system is tied up for each program. It records not only total job time and actual processing time, but it also breaks down non-productive time into setup time, downtime during the run, and idle time between jobs.

Printout is on a standard tab card, which is then keypunched and processed with further information to give a complete report on each job. (A reminder system signals the operator if a job is started without a card in the machine.) This report can be the basis for allocating computer costs to the various user departments or divisions or for customer billing in a service center operation.
(For more information, designate #68 on the Reader Service Card.)

COMPUTER ROOM LOCK / Simplex Lock Corp.

According to recent reports by the Wall Street Journal, and Harvard Business Review, easy access to computer rooms has resulted in large financial losses to many companies. Cases cited include theft of valuable mailing lists, computer programs and operating procedures. Erasure of vast files by disgruntled employees was another example given of damage occurring through loose security in computer centers.

A new lock, called Simplex, manufactured by Simplex Lock Corp., New York, N.Y., combines the security and protection features of a conventional combination lock with the ease, speed and convenience of push button operation. Operating on a patented push button combination principle, Simplex requires no keys. The device is completely mechanical and requires no electrical power source. Five buttons in a circle, numbered 1 to 5, can be used to create thousands of combination possibilities.



As an example, 1, 3 and 5 can be pushed in sequence to open a door. Or 1 and 3 may be pushed at the same time, and then 5, to effect an opening. In the latter case, pushing 1 and 3 in sequence will not open the lock; they must be pushed at the same time. From one to five buttons can be used in a combination. Combination changes can be made by any authorized individual — in less than a minute.

Using a template supplied with each lock, installation may be accomplished in a matter of minutes, following simple instructions. The only tools needed are a screwdriver, electric drill and a 70¢ hole saw set. The Simplex lists at $30.
(For more information, designate #72 on the Reader Service Card.)

MYLAR REINFORCED TAPES FOR N/C / Arvey Corporation

Arvey Corporation, Chicago, Ill., has added three new Mylar reinforced combinations to its present line of perforator tapes. The tapes — identified as R-V-CA-64, .003", Mylar/foil/Mylar; R-V-CP-115, .004", paper/Mylar/paper; and R-V-CT-52D, .0028", metalized Mylar — are reported to offer improved performance for numerical control applications.
(For more information, designate #69 on the Reader Service Card.)

PORTABLE FIRE PROTECTION SYSTEM / Fenwal Inc.

A compact, fully automatic self-contained fire protection system is now available from Fenwal Inc., Ashland, Mass. The system, called FIREPAC 360, is about the size of a portable television set. It uses a new suppressant, Du Pont's "Freon" FE 1301, which will not harm electronic equipment, papers, furnishings or personnel. It is electrically non-conductive and leaves no residue.

Each FIREPAC 360 has its own sensitive fire detectors capable of sensing heat, flame or smoke, an emergency power supply and a supply of "Freon" FE 1301. A single FIREPAC 360 can protect 1500 cubic feet of space. An unlimited number of units can be connected in series for greater area protection. A detection signal is flashed through a control unit to the suppressor. The FE 1301 is then released from the suppressant reservoir before the fire can endanger the protected object. The system may be fired manually either from the units or from a remote point.

Among the many applications for FIREPAC units are protecting high value areas such as book storage, record file rooms, and electronic equipment rooms.
(For more information, designate #70 on the Reader Service Card.)

MATTE FINISH MYLAR TAPE / Numeridex Tape Systems, Inc.

A new mylar tape which will eliminate tape misreads and the resulting costly down time on N/C machine tools has been developed by Numeridex Tape Systems, Inc., Chicago, Ill. This tape, NUMERIDEX Series #5000 is a Mylar/Aluminum Foil/Mylar, with a matte or low lustre finish which has been developed for reflective photo-electric readers. It can also be used on all other types of photo-electric and mechanical readers and is priced competitively with all other mylar tapes.

The Series #5000 tape is 100% opaque. The tape is available in either .0037" thickness, Pink over Gold, or .0032" Blue over Silver. Test samples of the new tape are available.
(For more information, designate #71 on the Reader Service Card.)

TIME-SHARING SERVICES

ITT'S NEW CHICAGO TIME-SHARING CENTER WILL SERVE MIDWEST AREA

ITT Data Services' new computer service center, recently opened in Chicago, Ill., will offer time-sharing, programming and conventional "batch" data processing services. The new center is equipped with an IBM System/360 Model 50 computer and supported by a large staff of programming, systems and data preparation experts.

Robert A. Leonard, president of this division of International Telephone and Telegraph Corporation, said the center will be the hub of a regional data processing network serving Colorado, Illinois, Indiana, Iowa, Michigan, Missouri, Ohio and Pennsylvania. It will operate around-the-clock, seven days a week.

UNIVAC INFORMATION SERVICES DIVISION ANNOUNCES REMOTE BATCH PROCESSING SERVICE

A new low-cost remote batch processing service allowing customers to use large-scale UNIVAC 1108 Systems from their own offices has been announced by Sperry Rand Corporation's Univac Information Services Division. The service, called RPS — Remote Processing Service — includes use of batch processing terminals designed for economical tie-in with Univac's national network of data processing information service centers.

By means of the leased terminals, customers can share use of an 1108 computer in an Information Services Division Computer Center, and communicate with any point in the national network.

COMPUTER RELATED SERVICES

PUBLISHING INDUSTRY OFFERED NEW SERVICE BY UNIVERSITY COMPUTING COMPANY

A proprietary computer service for magazine subscription fulfillment is now available from Data Link Division of University Computing Company. The system, known as MAGIC (from MAGazine Industry Circulation System), is designed to automate the tedious clerical chore of maintenance and updating of subscription lists.

As a by-product the MAGIC system will prepare audit reports in conformity with Audit Bureau of Circulation (ABC) and Business Publications Audit (BPA) requirements as well as generating many special marketing reports on magazine subscribers. Monthly mailing labels are also produced on a high speed computer printer. Additionally, the system is geared to generate, as required, renewal notices, billings, verifications and other periodic reports.

Service on the MAGIC system is available through Data Link centers in several U.S. cities.
(For more information, designate #73 on the Reader Service Card.)

NATIONAL REGISTRY DEVELOPS JOB-MATCHING SYSTEM USING COMPUTERS

The National Registry, an affiliate of Computer Applications, Inc., New York, N.Y., has developed a system matching man-with-job through the use of computers.

In explaining the system, John E. Sullivan, president of The National Registry, said that executive professional and technical people submit a simple pre-coded form outlining their qualifications for storage in the computer's "memory bank." At the same time, corporations feed the computer detailed reports — "Opportunity Bulletins" — on job openings. The computer then matches man to job opportunity and a copy of the corporation's report on the opening is mailed to each pre-qualified subscriber. Subscribers remain completely anonymous until they elect to respond to an "Opportunity Bulletin" offer.

Mr. Sullivan predicts that within five years, 30% of all employment placements will result from computerized job-matching. He said that The National Registry is currently operating exclusively in the computer recruitment advertising field but that the system can be applied to a broad range of communications problems.
(For more information, designate #74 on the Reader Service Card.)

NEW LITERATURE

"SOFTWARE CENTRAL", A MONTHLY PUBLICATION FROM COMPOSITION INFORMATION SERVICES, INC.

Composition Information Services, Inc. (CIS), Los Angeles, Calif., in a move to minimize the problem of duplication in software development, has announced Software Central — a periodic forum designed to promote cooperation among non-competitive users and provide a meaningful interface between printing and publishing industry computer practitioners and software suppliers. CIS is a management organization concerned with the application of electronics and computer technology within the printing and publishing industry.

Software Central publicizes related computer systems and program availability and requirements. It is aimed at reducing programming cost and surmounting the shortage of programming talent. The new CIS service was also established to assist computerized typesetting users extend their general-purpose systems into a wide range of business-oriented applications. Interested firms may obtain a free sample edition of Software Central.
(For more information, designate #77 on the Reader Service Card.)

NASA ISSUES PUBLICATION ON SYSTEMS ANALYSIS

A survey of potential uses of aerospace management methods in other large-scale undertakings, private and public, is available in a new publication by the National Aeronautics and Space Administration. "Applications of Systems Analysis Models" (written under contract for NASA by ABT Associates, Cambridge, Mass.) has been issued by the NASA office of Technology Utilization to enable industry and other Government agencies to benefit from NASA's experience and findings.

The publication discusses the adaptation and application of NASA know-how and methods to problems that continually confront management, such as long-range planning, cost effectiveness and control, and market development. It also deals with use of these techniques in urban and regional planning and points out their potential helpfulness to city officials and administrators of public health, educational and other programs.

The survey is available as NASA SP-5048 for 50 cents from: Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.

AUERBACH RELEASES REPORT ON COMMERCIAL TIME-SHARING

Richard A. Gorton, Manager of Product and Marketing Planning for AUERBACH Associates, Philadelphia, Pa., has announced the firm's release of a comprehensive two-volume report, "A study of Commercial Time-Sharing Services". The report, undertaken to determine the present status of time-sharing and to assess its probable future growth and development, concentrates primarily on commercial services in the U.S. However, in order to provide suitable perspective, consideration also has been given to certain special-purpose services and to non-conversational, remote batch processing services.

The AUERBACH Associates staff based its findings and conclusions upon interviews with managers, marketing staffs and technical personnel of 19 commercial time-sharing services; field interviews with users involved in a broad range of applications and industries; interviews with equipment manufacturers and industry spokesmen; and research into the texts, journals, reports, conference papers and other literature pertinent to the time-sharing field.
(For more information, designate #75 on the Reader Service Card.)

"COMPUTER AIDED DESIGN", A NEW QUARTERLY PUBLICATION

"Computer Aided Design", a new quarterly publication, specializes in the expanding field of computer-aided design applied to a wide range of industrial processes. The first issue dealt specifically with Control Systems, Graphics, Civil Engineering, Electronic Equipment Design, Railways, Chemicals and Mathematics. This issue is intended to be an introduction to the subject, as an assistance to the engineer encountering the subject for the first time.

Contents of the second issue, published in January, 1969, include such information as: Water Turbine design by computer; further articles on specific railway engineering design problems; a computer manufacturer's graphics philosophy; and Computer-aided design in the USA.

The new quarterly is published by Heywood-Temple Industrial Publications Ltd., London, England. Mr. Ed. Patterson, B.Sc., editor of "Design Electronics", is editor of the new publication, "Computer Aided Design."
(For more information, designate #76 on the Reader Service Card.)

ORGANIZATION NEWS

NEW MARKETING CONCEPT FOR SOFTWARE

A major change in marketing policy by Factsystem Inc., a Chicago-based (Illinois) software firm, is based on "industry-wide skepticism" encountered in test marketing of a proprietary commercial software package currently offered. Factsystem president, Dr. Leslie J. Jacob, announced that the package, an integrated total management information system, termed Factsystem, (of which portions remain in development) will be leased on a short-term, piecemeal basis "to counteract a reluctance on the marketplace toward making a permanent commitment prior to the system's completion." Dr. Jacob stressed in his announcement that the term lease and piece-by-piece implementation arrangement is made practical by means of Factsystem's modular design.

The new marketing arrangement permits installation of any or all of the completed portions of the system — general ledger, accounts receivable, payroll, etc. — on a term basis with payment beginning only as they are installed. Other sections may be added as they are completed. Provision is made for application of a major portion of the term lease payments to permanent licensing of the system.

The company has published a revised price schedule for all elements of Factsystem. The revised schedule also includes new anticipated release dates for those elements of the system which are not completed. Factsystem currently operates on IBM System/360 equipment under DOS, requiring 65K core memory.

ADAPSO ESTABLISHES A TIME-SHARING SECTION

The Association of Data Processing Service Organizations, Inc. (ADAPSO), New York, N.Y., has announced the establishment of a Time Sharing Section within the framework of the association. At a recent meeting held in San Francisco, twenty-one time sharing companies elected to affiliate with the Association.

"The Section," stated Executive Vice President of ADAPSO, J. L. Dreyer, will be made up of Time Sharing companies with a chairman, vice chairman and secretary elected from its membership. Each individual firm will be a member of ADAPSO in addition to its membership in the section."

Organizations interested in more information should write to the Association of Data Processing Service Organizations, Inc., 420 Lexington Ave., New York, N.Y. 11017

ITT DIVISION SIGNS SOFTWARE MARKETING AGREEMENT WITH SCIENTIFIC RESOURCES CORP.

A long-term agreement for the promotion and marketing of FORCE-III, has been consummated by Scientific Resources Corp., Philadelphia, Pa. and ITT Data Services of Paramus, N.J. FORCE-III is a proprietary time-sharing software package for use with IBM System/360 computers and was developed by Honig Time Sharing Associates, Inc. (Hartsdale, N.Y.) a Scientific Resources subsidiary. ITT Data Services has been given exclusive worldwide rights to promote, license, lease, and/or sell FORCE-III. In turn, Scientific Resources will provide maintenance support and system updating, and will help train ITT's licensees and customers in system use.

ARIES CORP. SIGNS ITS FIRST FOREIGN FRANCHISE AGREEMENT

Roger C. Dickinson, Director of Marketing for the Products Division of ARIES Corp., McLean, Va., has announced the recent signing of the firm's first foreign franchise agreement. Beginning in January, 1969, according to the terms of the agreement, International Data-Processing Systems of Geneva, Switzerland, would begin marketing AutoDiagrammer II in France, Italy, and Switzerland. (AutoDiagrammer II is a computer software package that automatically analyzes and documents computer programs.) ARIES Corp. plans other such agreements to facilitate the marketing of its proprietary software packages on a worldwide basis.

NEW CONTRACTS

TO	FROM	FOR	AMOUNT
Bull-General Electric	Burroughs Corp., Detroit, Mich.	Keypunch equipment, which will be manufactured to Burroughs specifications for worldwide distribution with Burroughs electronic data processing systems	$10+ million
The National Cash Register Company, Dayton, Ohio	Litton Industries	Design and production of electronic keyboards and high-speed, non-impact thermal printers for use in Litton's TACFIRE (Tactical Fire Direction System, Artillery AN/GSG-10) program for the U.S. Army	$9 million
TISCO, College Park, Md.	National Aeronautics and Space Administration (NASA)	Operating and upgrading the services of NASA's Scientific and Technical Information Facility in Maryland; renewable for two more years	$4.3 million per/yr.
Computer Sciences Corp., Los Angeles, Calif.	Defense Communications Agency	Continuation of the computer-based services CSC provides to the Agency's National Military Command System Support Center	$4.1 million
Interactive Sciences Corp., Braintree, Mass.	Digital Equipment Corp., Maynard, Mass.	Five PDP-10 computer/communications systems; delivery is scheduled over 18 mos. period	$4+ million
Univac Federal Systems Division	National Aeronautics and Space Administration (NASA)	Operational engineering support for UNIVAC 494 computers in the Communications Command and Telemetry System (CCATS) a key part of the Apollo lunar landing mission	$2.7 million
Recognition Equipment Ltd., a subsidiary of Recognition Equipment Incorporated	British General Post Office	Optical reading and high-speed sorting equipment	$2.6 million
Scientific Control Corp., Dallas, Texas	Information Industries, Inc., Los Angeles, Calif.	A Model 6700 time-share computer	$2+ million
Radiation Inc., Melbourne, Fla.	Louisiana Power & Light Company	A supervisory control system consisting of three computer equipped master stations and 25 remote stations	$1.7 million
RCA, New York, N.Y.	U.S. Air Force	Two contracts for expanding computing facilities that help military officials maintain the watch against a surprise missile attack on the United States	$1.7 million
EPSCO, Inc., Westwood, Mass.	U.S. Navy	Data conversion apparatus for Navy's P-3C Orion anti-submarine warfare patrol aircraft	$1+ million
Burroughs Defense, Space and Special Systems Group, Paoli, Pa.	U.S. Post Office Department	Development of an Automatic Mail Directory System; system is designed to speed delivery of incorrectly addressed mail to soldiers in Vietnam	$900,000
The Bunker-Ramo Corp., Stamford, Conn.	First Federal Savings of Detroit, Mich.	A system of electronic teller machines (63) for its 25 offices that will respond to any teller's entry within one second	$800,000
Ampex Corp., Redwood City, Calif.	Varian Data Machines, a subsidiary of Varian Associates, Inc.	Magnetic core memory stacks to be used in various model core memories and computers produced at Varian's Irvine, Calif. facility	$500,000
Burroughs Defense, Space and Special Systems Group, Paoli, Pa.	U.S. Air Force	Development of a low-cost memory system for use with aerospace computers	$283,000
EPSCO, Inc., Westwood, Mass.	U.S. Army Missile Command	Mobile Target Tracking Systems (MITS) apparatus; MITS track, control and plot the position of target missiles from a compact computer-centered portable ground station for troop training in the field	$208,000
Logic Corp., Haddonfield, N.J.	University of Pennsylvania	An LC-720 Data Entry System; an additional 14 keyboard input terminals expected to be purchased through August, 1969, bring total sale to $150,525	$115,528
Technology Incorporated, Dayton, Ohio	Office of Naval Research, Department of the Navy	Development of a system that will automatically eliminate flutter in data reproduced on magnetic tape recorders	$84,384
Computer Sciences Corp., Los Angeles, Calif.	Florida Computer Systems Co., a subsidiary of Florida Gas Co.	Implementation of a computer-based customer information system; will serve over 300 utility companies throughout Florida	—
Planning Research Corp., Los Angeles, Calif.	The Downtowner Corporation, Memphis, Tenn.	A five-year contract subscribing to the new automated international reservation system of Planning Research Corp.	—
Cyber-Tronics, Inc., New York, N.Y.	U.S. Social Security Administration	Rental of some 700 units of data processing equipment for installation at the Social Security Central Office in Baltimore, Md., as well as East Coast and Mid-West Payment Centers	—
Datatronic Rental Corp., Chicago, Ill.	U.S. Social Security Administration	A data processing equipment lease and service contract involving 197 IBM data processing machines for two SSA payment centers	—
Atkins & Merrill, Inc., Sudbury, Mass.	Servicos Aereos Cruzeiro do Sul, S.A., Rio de Janeiro, Brazil	The construction of an electronic flight and navigation simulator for its fleet of twin-engined turbo-prop YS-11's	—

NEW INSTALLATIONS

OF	AT	FOR
Burroughs B500 system	Murray B. Marsh Co., Los Angeles, Calif.	Accounting and inventory control for firm's 11 warehouses serving customers in 13 western states
Burroughs B-3500 system	Interactive Learning Systems, Inc., Boston, Mass.	Educational programs that ILS is developing
Control Data 1700 system	Federal Office of Civil Aviation, Vienna, Austria	Increasing speed and accuracy of weather communications throughout Europe; will serve as a message switching center for all communications networks in the European weather communications systems
Control Data 3300 system	Erlangen University, Erlangen, West Germany	A variety of general research projects in the university departments; also as an educational tool for students of data processing techniques
Control Data 6500 system	FIDES, a subsidiary company of the Swiss Credit Bank	Use as a service center; it is connected to ETH, the Swiss Federal Institute of Technology, which uses the 6500 for scientific, educational and administrative work
	Michigan State University, East Lansing, Mich.	Classroom instructional use, research studies and, when possible, computer time for local industry and other state colleges
Control Data 6600 system	French Atomic Energy Commission, Saclay Nuclear Research Ctr., France	Use in conjunction with fundamental research in nuclear physics, isotope biology and radiation control
Digital Equipment PDP-8	Herald Publishing Company, Rock Hill, So. Car.	Operating firm's new Photon 713-10 and Photon 560 photo composition systems; also handles wire stripping and classified ad storage
	Mobil Research and Development Corporation, Princeton, N.J.	Gas chromatographic analysis
Digital Equipment PDP-8/I	Virginia Polytechnic Institute, Chemistry Dept., Blacksburg, Va.	Use as a data acquisition and analysis system to perform in conjunction with their nuclear magnetic resonance, electron spin resonance, and Massbauer spectrometry instrumentation
Digital Equipment PDP-8/S	Alfred Hospital, Melbourne, Australia	Processing signals obtained from the analysis of organic fluid samples
	Christiania Spigerverk Iron and Steel, Oslo, Norway	Logging data from a steel-melting arc furnace, to achieve information for optimal melting conditions
Digital Equipment PDP-10	Graphic Controls Corporation, Buffalo, N.Y.	Providing customers with additional remote-access computing services
General Electric GE-115 system	Auburn Memorial Hospital, Auburn, N.Y.	Patient accounting, accounts payable and inventory control (system valued at $70,000)
General Electric GE-225 system	Kendall College, Evanston, Ill.	A variety of instructional and administrative purposes; also used by nearby colleges and high schools
General Electric GE-415 system	Arizona State University, Phoenix, Ariz.	Used by more than 500 students training in programming, computer sciences and research
Hewlett-Packard Model 2000A system	Time Share Corp., Hanover, N.H.	Serving firm's clients in education, engineering and business; first of five ordered by Time Share
Honeywell Model 200 system	Higher Education Assistance Corp., Delmar, N.Y.	Handling loan application processing and student payments in New York (state and federal loans)
	St. John's River Junior College, Palatka, Fla.	Instruction in basic programming and assembly computer languages in its computer courses that lead to an associate's degree; also administrative purposes for the county's board of public instruction
Honeywell Model 1250 system	The Smithsonian Institution, Washington, D.C.	Cataloging the more than 50 million specimens (fossils, flowers and fish) kept in the Institution's Museum of Natural History
IBM System/360 Model 40	State of Mississippi, Dept. of Public Safety, Jackson, Miss.	A computer-based road safety system and law enforcement program
IBM System/360 Model 91	Lockheed-California Co., Burbank, Calif.	Aiding in design and development of Lockheed's next generation of aircraft
NCR Century 100 system	San Jose City College, San José, Calif.	Instructional purposes; also library cataloging, student scheduling and general administrative work
NCR 315 system	American National Bank, Muncie, Ind.	Demand deposit processing and installment loans
NCR Series 500 system	St. Mary's General Hospital, Lewiston, Maine	Processing accounts receivable; later automatic cycled patient billing
RCA Spectra 70/46 system	University of Dayton, Dayton, Ohio	Educational and research projects; remote terminals on campus will simultaneously link students and faculty with Spectra 70/46; system also will be available to 10 Dayton-Miami Valley Area schools
UNIVAC 1108 system	Computel Systems Ltd., Toronto Center, Toronto, Ont., Canada	Services covering a wide range of computing tasks (system valued at $2.5 million)
UNIVAC 9200 system	Eberhard Mfg. Co., Cleveland, Ohio	Payroll processing, accounts receivable, inventory and production control
	Electronic Computer Programming Institute (ECPI), Mason City, Iowa	Teaching the fundamentals of computer programming to students
	Earle M. Jorgensen Steel Co., Seattle, Wash.	Production control and job scheduling as well as inventory control and payroll processing
	Truck Rentals Inc., Denver, Colo.	Providing cost records on a more timely basis; also general accounting procedures and miscellaneous tasks concerned with truck operation and maintenance
UNIVAC 9300 system	Stratton Warren Hardware Inc., Memphis, Tenn.	Instantaneous updating of inventory on over 48,000 items in its warehouses, and general accounting
	William Timpson, Ltd., Manchester, England	Warehouse stock control and automatic stock replenishment for nearly 300 retail stores in England, Scotland, and Wales

COMPUTER CENSUS

I. and U. Prakash
DP Data Corp.
61 Helen Drive
Marlboro, Mass. 01752

Introduction:

DP DATA CORPORATION collects information through year-round continuing market surveys. Its market research program includes world-wide files on computer, unit record, and other data processing equipment and software. These files identify by each user installation sites and information regarding digital computers, other data processing equipment, and software.

Part 1, information on leading manufacturers in the U.S. market, is presented in this issue. Part 2, information on other manufacturers in the U.S. market and Part 3, information on all manufacturers in the non-U.S. market will be presented in later issues. As always, we invite and welcome comments, additions, or corrections from informed readers.

COMPUTERS AND AUTOMATION has published a monthly Computer Census for several years. Starting with this issue, an improved Computer Census will be published. Essentially, it will be based on more accurate information and will be in a new format. We have made an arrangement with DP DATA CORPORATION, a leading market research firm specializing in the information processing industry, to provide data from its data base and to supplement the information provided to us by other sources.

PART 1 - THE TOP TEN MANUFACTURERS

Data as of September 30, 1968 (except as noted)

* Data not available at press time

MANUFACTURER	COMPUTER TYPE	DATE OF INITIAL CUSTOMER INSTALLATION	AVERAGE MONTHLY RENTAL $ (000)	NUMBER OF COMPUTERS INSTALLED IN U. S. LEASED & PURCHASED
Burroughs	205	1/54	4.6	25
	220	10/58	14.0	28
	B100	8/64	2.8	90
	B200	11/61	5.4	372
	B300	7/65	9.0	183
	B500	-	3.8	None
	B2500	2/67	5.0	52
	B3500	5/67	14.0	44
	B5500	3/63	25.0	65
	B6500	-	33.0	None
	B7500	-	44.0	None
	B8500	-	200.0	None
Control Data	G-15	7/55	1.6	*
	G-20	4/61	15.5	*
	LGP-21	12/62	0.7	*
	LGP-30	9/56	1.3	*
	RPC-4000	1/61	1.9	*
	160/160A/160C	5/60;7/61;3/64	2.1;5.0;12.0	*
	924/924A	8/61	11.0	*
	1604/1604A	1/60	45.0	*
	1700	5/66	3.8	65
	3100/3150	12/64	10.0	68
	3200	5/64	14.0	46
	3300	9/65	20.0	38
	3400	11/64	18.0	12
	3500	8/68	25.0	1
	3600	6/63	52.0	30
	3800	2/66	53.0	8
	6400/6500	5/66	56.0	23
	6600	8/64	115.0	32
	7600	-	235.0	None
Digital Equipment (Data as of Dec. 31, 1968)	PDP-1	11/60	3.4	48
	PDP-4	8/62	1.7	32
	PDP-5	9/63	0.9	100
	PDP-6	10/64	10.0	21
	PDP-7	11/64	1.3	99
	PDP-8	4/65	0.5	1381
	PDP-9	12/66	1.1	309
	PDP-10	10/67	9.0	36

PART 1 - THE TOP TEN MANUFACTURERS

Data as of September 30, 1968 (except as noted)

* Data not available at press time

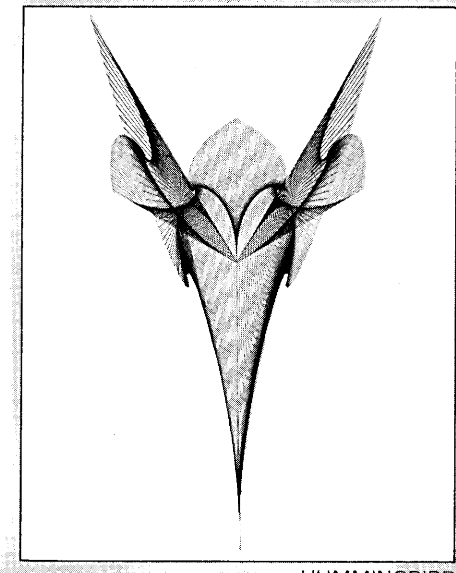
MANUFACTURER	COMPUTER TYPE	DATE OF INITIAL CUSTOMER INSTALLATION	AVERAGE MONTHLY RENTAL $ (000)	NUMBER OF COMPUTERS INSTALLED IN U. S. LEASED & PURCHASED
General Electric	115	12/65	2.2	198
	130	-	4.5	None
	205	6/64	2.9	11
	210	7/59	16.0	35
	215	9/63	6.0	15
	225	4/61	8.3	143
	235	4/64	12.5	54
	405	2/68	6.0	8
	415	5/64	11.0	170
	425	6/64	18.0	47
	435	9/65	25.0	6
	625	4/65	50.0	23
	635	5/65	60.0	19
	645	7/66	90.0	4
	4020	2/67	5.0	13
	4040	8/64	3.0	18
	4050	12/66	7.0	9
	4060	6/65	8.5	8
Honeywell	110	-	2.5	None
	120	1/66	4.0	260
	125	12/67	5.0	16
	200	3/64	8.5	448
	400	12/61	8.5	32
	800	12/60	28.0	42
	1200	2/66	10.0	62
	1250	-	12.0	None
	1400	1/64	14.0	6
	1800	1/64	50.0	8
	2200	1/66	16.0	32
	4200	-	21.0	None
	8200	-	50.0	None
IBM	1130	2/66	1.7	2580
	1401	9/60	5.4	2210
	1410	11/61	17.0	156
	1440	4/63	4.1	1690
	1460	10/63	9.6	194
	1620	9/60	4.1	285
	1800	1/66	5.0	415
	7010	10/63	26.0	67
	7040	6/63	25.0	35
	7044	6/63	36.5	28
	7070	3/60	27.0	10
	7074	3/60	35.0	44
	7080	8/61	60.0	13
	7090	11/59	63.5	4
	7094-I	9/62	75.0	10
	7094-II	4/64	83.0	6
	3620	12/65	2.8	4690
	3625	-	5.3	None
	3630	5/65	9.3	5076
	3640	4/65	19.0	1261
	3644	7/66	15.0	66
	3650	8/65	33.0	478
	3665	11/65	70.0	175
	3667	6/66	138.0	9
	3675	1/66	81.5	14
	3685	-	115.0	None
	3690	11/67	150.0	5
NCR	304	1/60	14.0	5
	310	5/61	2.5	8
	315	5/62	8.7	265
	315 RMC	9/65	12.0	57
	390	5/61	1.9	81
	500	10/65	1.5	620
	Century 100	9/68	2.7	8
	Century 200	-	7.5	None
RCA	301	2/61	7.0	137
	3301	7/64	18.0	24
	501	6/59	14.0	22
	601	11/62	35.0	2

PART 1 - THE TOP TEN MANUFACTURERS
Data as of September 30, 1968 (except as noted)
* Data not available at press time

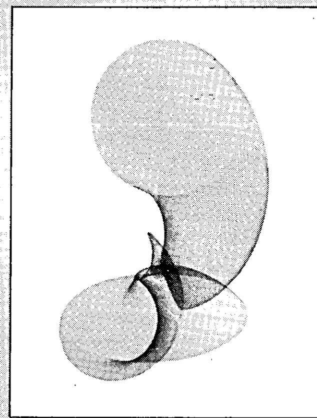
MANUFACTURER	COMPUTER TYPE	DATE OF INITIAL CUSTOMER INSTALLATION	AVERAGE MONTHLY RENTAL $ (000)	NUMBER OF COMPUTERS INSTALLED IN U. S. LEASED & PURCHASED
RCA,	70/15	9/65	4.1	90
	70/25	9/65	6.7	68
	70/35	1/67	8.0	65
	70/45	11/65	23.0	84
	70/46	-	33.5	None
	70/55	11/66	33.5	11
Scientific Data	92	4/65	15.0	7
	910	8/62	2.0	153
	920	9/62	2.9	93
	925	12/64	3.0	20
	930	6/64	3.4	159
	940	4/66	14.0	28
	9300	11/64	8.5	21
	Sigma 2	12/66	1.8	56
	Sigma 5	8/67	6.0	15
	Sigma 7	12/66	12.0	24
Sperry Rand	III	8/62	21.0	25
	418	6/63	11.0	76
	490 Series	12/61	30.0	75
	1004	2/63	1.9	1502
	1005	4/66	2.4	637
	1050	9/63	8.5	138
	1107	10/62	57.0	8
	1108	9/65	68.0	38
	9200	6/67	1.5	127
	9300	9/67	3.4	106
	9400	-	7.0	None

ADVERTISING INDEX

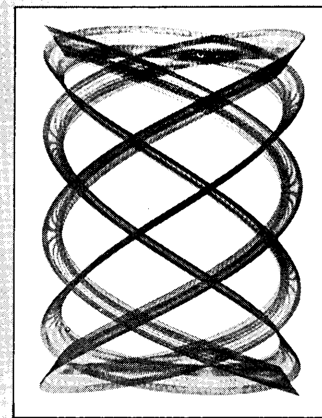
Following is the index of advertisements. Each item contains: Name and address of the advertiser / page number where the advertisement appears / name of agency if any.



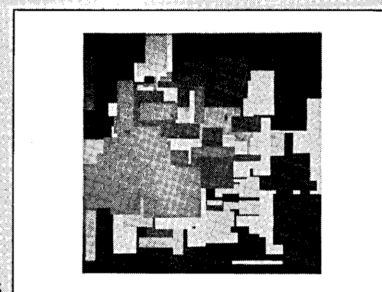
HUMMINGBIRD



INSPIRALATION



PLEXUS

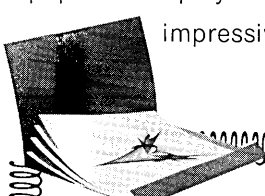


IDEALIZED BRUSH STROKES

SPECIAL OFFER!

full color reproductions of computer-generated artwork

Here is a unique opportunity to own a distinctive and unusual set of four prints from the 1968 Computer Art contest. Each print is 12" x 16" and is reproduced in magnificent full color on heavy weight quality paper suitable for framing. Symbolic of the computer industry, these attractive and interesting prints are ideal for decorating your office, den or home. Packaged in a handsome folio, they make distinctive gifts for friends or business associates. Each is imprinted on the back with a description of the programming technique, computer and plotting equipment employed to produce the art. Everybody in the computer field will want a set of these beautiful and impressive prints. Send for yours now. Supply is limited.



ORDER FORM

Gentlemen: please send your folio of computer art prints.
Enclosed is my ☐ check ☐ money order for $10.00.

NAME______________________________

COMPANY______________________________

ADDRESS______________________________

CITY________________STATE__________ZIP_______

For orders of more than 10 sets, write for quantity price list

Compro

1060 KINGS HIGHWAY NORTH
CHERRY HILL, N.J. 08034
609-667-4709

Weather has no respect for national boundaries. Perhaps that's why international cooperation in forecasting has for a long time been a constant in an area fraught with inconsistency.

Cooperation is important, but it's not the only answer to more accurate and longer range forecasting. Computer technology is a key factor.

This is fortunate for the United States, because no other nation is further advanced in this area.

And no other nation's weather research establishment has two UNIVAC® 1108 Systems at its disposal.

Scientists at the Environmental Science Services Administration's Geophysical Fluid Dynamics Laboratory use them to simulate the earth's atmosphere by means of theoretical models.

Their goal is to develop a satisfactory model comprised of elements which closely approximate those of the real atmosphere.

By changing just one variable in this mathematical model—sea surface temperature, snow cover, atmospheric pollutants, for instance—scientists can in effect experiment with the atmosphere and determine how that one alteration influences the entire model.

These simulations are so complex, the elements so numerous that a 24-hour forecast for the northern hemisphere can take 10 hours of computing.

Despite this scientists are optimistic. In fact, current results indicate that reliable 14-day forecasts are a distinct future possibility. Also, the observational network must be expanded to the whole of the earth.

It will take some time, therefore, before two week forecasts become standard. Scientists estimate five to ten years.

But if it weren't for computers, it might take forever.

UNIVAC

Univac is saving a lot of people a lot of time.

SPERRY RAND

Designate No. 14 on Reader Service Card

The Ultimate Weapon in the Wet War.

No one can stop the rain, but we can help you be ready for it.