Proceedings of the Eastern Joint Computer Conference

Held by the
JOINT COMPUTER COMMITTEE
of the
ASSOCIATION FOR COMPUTING MACHINERY
AMERICAN INSTITUTE OF ELECTRICAL ENGINEERS
INSTITUTE OF RADIO ENGINEERS, INC.

Philadelphia, Pa., December 8–10, 1954

Published by
THE AMERICAN INSTITUTE OF ELECTRICAL ENGINEERS
33 West 39th Street, New York 18, N. Y.
for the
JOINT COMPUTER COMMITTEE

Printed in the United States of America

Price $3.00

ADDITIONAL COPIES

Additional copies may be purchased from the following sponsoring societies at $3.00 per copy. Checks should be made payable to any one of the following societies:

ASSOCIATION FOR COMPUTING MACHINERY
2 East 63d Street, New York 21, N. Y.

AMERICAN INSTITUTE OF ELECTRICAL ENGINEERS
33 West 39th Street, New York 18, N. Y.

INSTITUTE OF RADIO ENGINEERS
1 East 79th Street, New York 21, N. Y.

**Library of Congress Catalog
Card Number: 55-7431**

PROCEEDINGS OF THE
EASTERN JOINT COMPUTER CONFERENCE

Theme: Design and Application of Small Digital Computers

PAPERS AND DISCUSSIONS PRESENTED AT
THE JOINT ACM-AIEE-IRE COMPUTER CONFERENCE,
PHILADELPHIA, PA., DECEMBER 8–10, 1954

Sponsors

THE ASSOCIATION FOR COMPUTING MACHINERY
THE AMERICAN INSTITUTE OF ELECTRICAL ENGINEERS
Committee on Computing Devices
THE INSTITUTE OF RADIO ENGINEERS
Professional Group on Electronic Computers

Published by the
American Institute of Electrical Engineers
33 West 39th Street, New York 18, N. Y.
for the
Joint Computer Committee

JOINT ACM-AIEE-IRE COMPUTER COMMITTEE*

Chairman

For the East
C. W. ADAMS
Massachusetts Institute of Technology

For the West
W. L. MARTIN
Telecomputing Corporation

ACM Representatives

East

C. W. ADAMS
Massachusetts Institute of Technology

F. L. ALT
National Bureau of Standards

West

D. H. LEHMER
University of California, Berkeley

J. L. BARNES
University of California, Los Angeles

Ex-Officio—A. S. HOUSEHOLDER
Oak Ridge National Laboratory
President, ACM

Headquarters Representative—E. BROMBERG
New York University Institute of Mathematical Sciences
Secretary, ACM

AIEE Representatives

East

G. G. HOBERG
Boroughs Research Activity

E. L. HARDER
Westinghouse Electric Corporation

West

R. R. BENNETT
Ramo-Wooldridge Corporation

G. D. McCANN
California Institute of Technology
Computer English Association

Ex-Officio—F. J. MAGINNISS
General Electric Company
Chairman, AIEE Committee on Computing Devices

Headquarters Representative—R. S. GARDNER
Assistant Secretary AIEE

IRE Representatives

East

M. M. ASTRAHAN
IBM Engineering Laboratory

W. D. LEWIS
Bell Telephone Laboratory

West

W. L. MARTIN
Telecomputing Corporation

O. WHITBY
Stanford Research Institute

Ex-Officio—H. T. LARSON
Ramo-Wooldridge Corporation
Chairman, IRE Professional Group on Electronic Computers

Headquarters Representative—L. G. CUMMING
Technical Secretary, IRE

Joint Secretary—Treasurer
W. D. LEWIS
Bell Telephone Laboratory

Technical Program Committee
H. R. J. GROSCH, Chairman
General Electric Company

* The "Organization of the Joint Computer Committee" is printed at the end of this publication.

PUBLICATIONS COMMITTEE

All of Westinghouse Electric Corporation

P. A. ZAPHYR
J. T. CARLETON
W. D. ROWE
R. E. WENDT, JR.

B. H. MOUNT
C. P. SAALBACH
R. W. LONG
M. MIDDLETON, JR.

LOCAL ARRANGEMENTS COMMITTEE*

General Chairman

J. M. BROOMALL....Minneapolis-Honeywell Regulator Company

Committee Secretary

D. P. BURKART....Radio Corporation of America

Conference Secretary

D. ELAINE WHITELAM....Minneapolis-Honeywell Regulator Company

Finance

JOHN McCARTIN, Chairman....Remington Rand Inc.
J. R. WEINER, Vice-Chairman....Remington Rand Inc.
JERRY BARBEN....Remington Rand Inc.
PHYLLIS BLYMIRE....Remington Rand Inc.
G. S. LEAR....Remington Rand Inc.
CONSTANCE M. McLAUGHLIN....Remington Rand Inc.
ALICYA STASINSKI....Remington Rand Inc.

Registration

NORMAN HARDY, Chairman....Hardy and Dixon
R. M. GOODMAN, Vice-Chairman....American Electronic Laboratories
JEAN CARTER....Technitrol Engineering Company
MARILYN HARDY....Hardy and Dixon
SIDNEY KAPLAN....Radio Corporation of America
JANE LOVE....Radio Corporation of America
HELEN MAGEE....Radio Corporation of America
JOSEPH MAHLER....Radio Corporation of America
ARLENE MELCHIORRE....Burroughs Corporation
BERYL O'DONIVAN....Radio Corporation of America
A. A. PEREZ....Radio Corporation of America
ROSE PONTIFINO....Radio Corporation of America
JEAN RIESLE....Remington Rand Inc.
DORA ROFI....Burroughs Corporation
CASS STANTON....Remington Rand Inc.
J. H. WAITE, JR....Radio Corporation of America
DORIS ZAMAL....Radio Corporation of America

Hotels

M. H. ROSEN, Chairman....Radio Corporation of America
S. E. GLUCK, Vice-Chairman....Burroughs Corporation
H. K. FORELL....Minneapolis-Honeywell Regulator Company
E. J. NOSSEN....Minneapolis-Honeywell Regulator Company
R. B. THORNLEY....Minneapolis-Honeywell Regulator Company

Exhibits

GORDON PALMER, JR., Chairman....Technitrol Engineering Company
S. B. DISSON, Vice-Chairman....Burroughs Corporation
JOSEPH DEUTSCH....Burroughs Corporation
GEORGE HAND....Technitrol Engineering Company

Inspection Trip

R. R. WAGSTAFF, Chairman....United Engineers and Constructors, Inc.
JOSEPH WYLEN, Vice-Chairman....Burroughs Corporation
M. L. AITEL....Radio Corporation of America
W. J. BARTIK....Remington Rand Inc.
MARTHA BOLE....Burroughs Corporation
DENNIS BOONE....Westinghouse Electric Corporation
ELAINE BORGSTEDE....Burroughs Corporation
ANGELA DIVITIS....Burroughs Corporation
MARGARET DRISCOLL....Burroughs Corporation
SUSAN JACOBS....International Business Machines Corporation
N. R. KORNFIELD....Radio Corporation of America
ROBERT LANE....Radio Corporation of America
HELEN MARCHESANI....Burroughs Corporation
DOLORES McFADDEN....International Business Machines Corporation
E. H. PETRIE....Minneapolis-Honeywell Regulator Company
PATRICIA POWERS....Burroughs Corporation

Publicity

R. G. LEX, JR., Chairman....Leeds and Northrup Company
MORRIS RUBINOFF, Vice-Chairman....Moore School of Electrical Engineering
H. P. GUERBER....Radio Corporation of America
W. M. RINCK....Radio Corporation of America

Printing Committee

J. D. CHAPLINE, JR., Chairman....Remington Rand Inc.
E. J. LEVINSON, Vice-Chairman....Remington Rand Inc.
F. J. LEARY....Remington Rand Inc.

* The Local Arrangements Committee wishes to thank the following organizations for their assistance during the conference:

The Bellevue Stratford Hotel
United Booth Co.
RCA Victor for Closed Circuit TV System
Burroughs Corp. for cash registers and Ticketeer
IBM for typewriters and lecture podium
Veeder-Root for Veri-tally

E. L. HARDER, Chairman
DORIS JEANNE CHEVIT, Secretary

TABLE OF CONTENTS

Small Computers in a Large World

C. W. ADAMS

THREE years ago, 877 of us gathered in Philadelphia at the first Joint Computer Conference, a meeting organized by a committee representing three professional societies which were then, as now, active in the design and application of electronic computing devices. These are, of course, the AIEE, the Institute of Radio Engineers, and the Association for Computing Machinery (ACM). The theme of that first conference was a review of the state of the art up to that time. In the following two Decembers, in New York, N.Y., and in Washington, D.C., attention was directed first toward the problem of input-output and then toward the problem of reliability. Meanwhile, Western Joint Computer Conferences were held in Los Angeles, Calif., in February of 1953 and 1954, with a third scheduled for the first 3 days of March 1955 at the Hotel Statler in Los Angeles, Calif.

The Conference Committee

This year, through what was for me a happy combination of circumstances for which no one person need bear the entire blame, I found myself serving as chairman of the Eastern half of the Joint Computer Committee. Now, from that exalted sinecure, I have the great pleasure of welcoming you all here and hoping that you will find to your liking the talks and discussions, the printed proceedings which the mailman will one day put in your hands, the exhibits and the tours, the social activities, and above all the exchange of ideas and information with your fellow conferees.

From my vantage point in Cambridge, I managed to keep my hands free of any actual work in the arranging of this conference. Dr. H. R. J. Grosch, who planned the program, John Broomall, the local arrangements chairman, and Dr. E. L. Harder with his publications committee were not as lucky, nor were the many other members of the local arrangements committee whose names are printed in the front of the proceedings. To these men, who have contributed and are contributing a great deal of time and effort, and to their various employers who at no small sacrifice have encouraged them to do so, we owe our thanks.

C. W. ADAMS is at the Massachusetts Institute of Technology, Cambridge, Mass.

The Theme

The theme of this conference is "The Design and Application of Small Digital Computers." This title, like any other that might have been picked, gives rise to two questions, namely, "What does it mean?" and "Why was it selected?"

Defining the term "small digital computer" is almost as difficult as defining an abstraction like the words "thought" or "life." First, I suppose, one collects in his mind as many examples as possible of things which clearly are small digital computers and of other things which equally clearly are not. Then one tries to single out a number of features unique to one class or the other. But, as soon as any or all of these characteristics are used actually to define what is meant by small digital computers, the trouble begins. There turns out to be a vast grey area in which there once were, or now are, or someday might be, or at least really ought to be, devices which some people would prefer to think of as small digital computers and which others would not.

Since, unlike the rest of the program, this keynote address is not formally open for discussion, I would rather not rouse any strong semantic objections. Rather than attempt to define the theme, therefore, I will simply let the program speak for itself. In order to leave myself something to talk about, however, I will nonetheless present some of the examples and mention some of the dividing lines which have occurred to me in trying to formulate a definition.

The Excluded Ones

The most obvious example of a device which is not a small digital computer would seem to be an analogue computer, be it large or small. Even here I am probably treading on unsure ground, for while I am aware that digital computers count while analogue computers measure, I am further aware that the distinction is imperfect because of the analogue-digital conversion devices and other odd bits of equipment that combine features of both.

Be that as it may, let me plunge on to another class of devices which are by definition not small digital computers—namely, large digital computers. To name quite a few, these would include Harvard's Marks I to IV, the ENIAC, EDVAC, and EDSAC, the IAS computer and its family of six (the Ordvax, Illiac, Oracle, Avidac, Johnniac, and Maniac), the Whirlwind, Raydac, SWAC, SEAC, MIDAC, and Dyseac, and the commercially available computers: the Ferranti's, the ERA's, International Business Machine (IBM) *701*'s, and the UNIVAC's. On a par with or even larger than these are some business data processing systems now projected: IBM's *705*, the Radio Corporation of America's Bizmac, etc.

But largest by far of all are the Goliaths of science fiction. Some of you have no doubt, probably to your sorrow, struggled through a pocket-sized novel called "Year of Consent," full of overdone parable and underdone science. In it, the author pictures for us an intellectual dinosaur, all bulk and no brains. Here is his description of a large computer of 1990.

> "The giant electronic brain filled up the first ten floors of our building. There were additional memory banks in several subcellars and in another nearby building.... It contained 500,000 electronic tubes and about 860,000 relays. Not counting the extra memory banks, it had 400 registers totalling 6,400 decimal digits of very rapid memory in electronic tubes and about 6,000 registers totalling 120,000 decimal digits of less rapid memory in relays.... Officially the giant brain was the SOCIAC, but simply because we were all a little afraid of its ability we were seldom that formal. To everyone around the office it was known as Herbie."

Perhaps the antithesis of 1990's Herbie is 1950's Curta, one of the very smallest hand-operated calculators. It adds and subtracts, can be made to multiply or divide 6 to 11 decimal digits at a time, costs only $150, uses practically no power, will fit in every elevator and go through every door. But, if I may coin a distinction, it is merely a calculator, an arithmetic element. It has no storage to speak of, no fully automatic sequence control. In short, while the term "small" presumably has no lower limit, the more rudimentary digital calculating devices, such as desk calculators, cash registers, and standard punched-card equipment, are ruled out of our theme if the term "computer" is restricted to devices which have an appreciable storage element.

The Small Ones

Thus far, we have excluded a goodly number of computers, including, for example, the UNIVAC on grounds of being not small, the REAC as being not digital, and IBM's *407* as being not a computer. What is left seems to divide into five different categories, two general-purpose and three special-purpose.

First to appear were the so-called electronic calculators, IBM's *603* in 1946, *604* in 1948, *605* or CPC in 1949, and *607* in late 1952, accompanied at that time by Remington Rand's *409-2*, not to mention the *604*-like calculators of Hollerith, Powers Samas, and Compagnie Bull. Today there are over 3,500 of these general-purpose machines in use, including at least 250 CPC's. If I may leave you contemplating the spectacle of 3,500 calculating punches stretched end to end from here almost to North Philadelphia, I'd like to digress long enough to remind you of a kind of children's cardboard-bound book made up of alternate pages of text and illustration that some years ago served in place of today's comic books. For some reason, these 10-cent books were small but thick, and their uninhibited publishers therefore called them big little books. Following this cue, it seems to me the CPC and its brethren, which, like the big little books, are symbols of a transition, might aptly be described as big little computers. This would imply, rightly I believe, that a CPC is conceptually a small device somewhat "beefed up" to do a bigger job.

Dr. C. C. Hurd described and demonstrated the CPC at the Rutgers University meeting of the ACM in March 1950. Strangely enough, three other categories of small digital computers were also introduced to the ACM at that meeting, all of them utilizing a magnetic drum for storage. Very appropriately, the topic of Professor Aiken's banquet address at that meeting was "Automatic Computing Machinery of Moderate Cost."

As I recall it, the hit of the show was the Magnetic Drum Digital Differential Analyzer, or MADDIDA, built by Northrup Aircraft. This special-purpose digital device utilized a magnetic drum to store a number of integrands, with one step of rectangular integration being performed on each integrand once each revolution of the drum. A dozen or more such devices have been put into service by Northrup, by Computer Research Corporation, and recently by Bendix.

The magnetic drum showed up at Rutgers in another even more promising special-purpose system, one intended purely for information storage, which was described by John L. Hill, whose name also appears in a similar context on this program. Engineering Research Associates, W. S. MacDonald Company, and Teleregister Corporation have all produced useful inventory record maintenance devices known respectively as the Speed Tally, the Magnefile, and the Reservisor. For inventory or any other record keeping which is subject to short-notice interrogations, the magnetic-drum system has distinct advantages that are being increasingly exploited. Witness, for example, Remington Rand's recent renaming of the ERA *DS63* as the UNIVAC File Computer.

At the Rutgers meeting, also, Professor Paul L. Morton discussed the design of a low-cost general-purpose small digital computer, a topic which was then just becoming of widespread interest. In the keynote address at the first Joint Computer Conference, W. H. MacWilliams pointed out four phases through which computer designers seem to pass. Paraphrased, these are the "we're building it" (or talking) phase, the "we're debugging it" (or silent) phase, the "it's working" (or bright-look) phase, and the "getting results" (or talking-about-the-next-one) phase. The 1951 conference marked the coming of the fourth or final stage for the first batch of large digital computers, but the end of 1951 was just the beginning of the second, or silent, phase for the small general-purpose drum-type computers which had entered the first, or talking, stage at Rutgers. I say this in spite of the fact that only a few days after the Joint Computer Conference in December 1951, the Computer Research Corporation made what I believe was the first delivery of a general-purpose drum-type computer, the CADAC, the somewhat premature prototype of the more recent CRC *102A*.

The CADAC and its many competitors are essentially scaled-down versions of the large computers, using slower storage and less elaborate input-output equipment to reduce the cost. In this sense, then, these might be called little big computers. In any event, following the silent phase of 1952 and 1953, deliveries of these machines on a fairly large scale finally commenced in 1954. In recent months, I have heard of the delivery of the 16th CRC *102A*, the fifth Elecom, the fourth ElectroData, the third Monrobot, the second Miniac, and the prototypes of the Circle, Alwac, and Hughes Airborne computers. The first IBM *650* was delivered to John Hancock for field testing a short time ago, but since more than 450 of them are on order, they should rapidly become commonplace.

While practically all of the 30 or more little-big computers now in the field are being used for scientific and engineering computations, it is interesting to note that well over half of the *650*'s and many of the other such machines on order will be used purely for business-data processing.

Without pretending that it will make my list of little-big computers complete, I should also mention some of the small British machines: the Elliott Brothers' NRDC *401* and Nicholas on which Ferranti is basing the design of their FPC, and the various APE(X)C machines built by Booth at Birkbeck College, London, and being engineered by Hollerith to produce the HEC.

Today's Machines

Recapitulating, there seem to be two types of small general-purpose digital computers, the ones built up from punched-card systems which might be called big-little computers, and the ones built up around a magnetic drum which might be thought of as little-big computers. As usual, there are exceptions which seem to straddle any such dichotomy. For example, the Burroughs *E101*, despite its drum, might be thought of as a big-little computer. Such an example makes it obvious that I should not require my two classifications to be mutually exclusive.

I have described also two types of small special-purpose digital devices both involving drums, namely, the digital differential analyzers and the tallying systems. To these, no doubt, should be added the input-output buffer and communications auxiliaries: the high-speed printers, punches, transceivers, etc. Clearly here we are getting further and further from actual computers, but stretching the term a little here and there perhaps will do no harm.

Tomorrow's Machines

I do not have any aptitude whatever toward being a prophet, so I cannot do justice to the interesting question of what the future has in store. It takes no prophet, however, to note the vast potentialities in improved reliability, in decreased power, space, heat, and weight, and very likely in increased capacity and decreased costs promised by the various solid-state devices, the transistors, the magnetic cores, and the ferro-dielectrics.

Three years ago, Dr. J. H. Felker of Bell Telephone Laboratories discussed the "Transistor as a Digital Computer Component;" this year Dr. Grosch prevailed on him to return to the Joint Computer Conference podium and describe the "Performance of the TRADIC Transistor Digital Computer." Then, too, the transistorized *604* recently unveiled by IBM is presumably in operation at this meeting. Parallel magnetic-core and ferroelectric memories will probably show up first in large digital computers, but these together with the use of magnetic cores as computing circuit elements and the development of very-high-density magnetic-drum recording, both of which are being discussed here, are further good omens in the small computer future which are on full view at this meeting. Certainly the best is yet to come, but more important, what is already here is well worth putting to use.

A Small Definition

When I started off to define small digital computers by examples, I said that there should be certain characteristics unique to small computers that might be used as the basis of a rigorous definition of smallness.

The most obvious of these is price. Certainly anything costing less than $150,000 is by present standards small while anything costing more than $750,000 is large. In between, you may either call them medium, or pick your own dividing line.

Secondly, there is the criterion used by the program chairman in planning this meeting. He defined as small anything that consumes less than 20 kw of power.

A third possibility is to take the stand that any device with more than 10,000 binary digits of storage capacity and with a random access time of less than a millisecond is to be considered as a large computer.

Similar to this, but possibly better yet, is to use the storage performance unit defined some years ago by Jay W. Forrester as being the total storage in binary digits divided by the random access time in seconds. In these units, IBM's *650* and the other small computers would show up with somewhat less than 30 megabits per second, while the UNIVAC could claim 150 (and the magnetic-core machines a whopping 3,000 to 6,000) megabits per second. On this basis, 100 megabits per second might be a fair dividing line.

Why Small Computers?

In talking about the small digital computers, I have tried to indicate the extremely high level of interest that exists concerning these devices. To me, this answers the question of why the theme of the meeting is what it is. It raises, however, another question which I hardly can do more than formulate for you.

Briefly, the question is, "Why are small computers so popular?" Depending on the situation, there seem to be a number of possible answers. Many small companies and many divisions of highly decentralized large companies find that their organizations do not have enough computing to occupy a large computer even if they had one. Other companies, large enough to support one or several large computers, feel that perhaps several small ones will be more efficient. They offer at least three very good reasons: easier scheduling; less confusion if a machine breaks down; and less expensive debugging of programs. Actually, the position of the small company is just a scaled-down version of that of the large company; the small company is merely choosing one small computer rather than using part of a large one being run as a central facility by someone else.

Use of a large central facility by a large company or several small ones certainly can have its frustrations, but it can also have advantages for companies both large and small. This can be seen from a fairly obvious empirical relationship (which we might call Grosch's Law) to the effect that the amount of computation a machine can produce is roughly proportional to the square of the cost. Thus a $30-an-hour, 100-multiplications-per-minute small computer is 100 times faster than a $3-an-hour, 1-multiplication-per-minute human computer, but only a hundredth as fast as a $300-an-hour, 10,000-multiplications-per-minute large computer. This difference in price per multiplication can pay for quite a lot of careful scheduling, effective emergency procedures, sophisticated debugging, and even wasted time. More work needs to be, and is being, done in this area, but even now the number of situations in which small computers are economically justifiable may not be as large as many people seem to think.

In contrast to the "we're too small" and to the "we don't put all our eggs in one basket" attitudes just discussed, the third and perhaps most prevalent is the "take it slow and easy" attitude. There are convincing arguments for starting off in a small way, especially in the commercial data-handling area, and working up gradually to the big one.

It is certainly true that mechanizing for a small computer is good practice for mechanizing for a large computer. It may also well be that a small one can be obtained and applied so much sooner than a large one could that the small one will more than pay for itself in the interim. However, it should be emphasized that mechanizing for a small computer may differ in more than detail from mechanizing for a large computer, because the storage and input-output capacities may be so much different that the jobs have to be broken down quite differently in the two cases.

It appears also that some groups are being rushed into ordering the first attractive computer package they can find, long before they know what to do with it, merely to avoid being left behind. In terms of cost, availability, space, and staff required, the small computers are, to say the least, winsome. This causes some tendency for the small computer to become a kind of plaything. It is not really an inexpensive plaything, however; and the idea may backfire (in a small way of course) on those who leap before they look, and indirectly then on the whole computer field.

By mentioning these negative aspects, I do not mean to be overly pessimistic. I am firmly convinced that small digital computers, both general-purpose and special, have very important roles to play. Happily or unhappily, however, the situation at present is in great turmoil and no one can hope fully to analyze his situation and choose the wisest course without perhaps finding himself left behind, and therefore not on the wisest course at all. The choice between large computers, small computers, or none at all is a personal decision for each prospective user to make, but it is an extremely difficult and important one for all.

I hope that what we will see and hear about small digital computers during this meeting will help us make the necessary decisions as wisely as possible, and that we will come away with a much clearer understanding of the place of the small computer in this very large world.

Why Not Try a Plugboard?

REX RICE, JR.

IN recent years, a very large proportion of the man-hours expended in designing and constructing new digital computers has been devoted to machines that are "internally programmed." By "internally programmed" we mean a machine in which all of the instructions and operands are contained interchangeably in storage. In contrast, we may consider an "externally programmed" machine as one in which operands and a bare minimum of instructions for subroutine control are contained in storage and the bulk of the instructions are wired into plugboards. The net result of the emphasis on internally programmed computers has been that many of the computing fraternity seem to be accepting the belief that this is the only kind of machine to use for computing. To illustrate the extent to which this belief has gone, in the "First Glossary of Programming Terminology" issued for the Association for Computing Machinery, the word "plugboard" is not even listed. It is merely mentioned under the heading of "storage" as a device that holds information, but its use as a simple and very powerful means of replacing coded instructions certainly has not been emphasized. This is perhaps true because the only externally programmed machines available are combinations of accounting machines and cannot really be considered as computers. To date, with one exception, no large plugboard machine properly designed from the beginning as a computer has been available.

This paper will show, by the use of examples, how programming and logical control are easily accomplished on a properly designed plugboard machine. The abstractions such as relative coding, symbolic coding, and automatic coding, which are essential for programming ease in an internally programmed machine, have no parallel in plugboard machines since programming is direct and simple.

In the following discussion, let us hypothesize a plugboard computer that meets the fundamental objectives of an economical, well-balanced machine. As we discuss this machine and compare its features with those of an internally programmed machine, let us remember that our goal is not to develop abstraction after abstraction and not to find newer and more intriguing ways of how elaborate we can become, but, rather, our goal is to reduce each addition, subtraction, multiplication, division, logical test, and subroutine control to a minimum of programming effort.

In a paper of this length it is not possible to discuss in detail all of the features that make a plugboard machine easy to use, fast, and versatile. Consequently, the discussion is necessarily limited to only a few essential features to show that the technique will work. A description of the machine and then an example will demonstrate a few of the possibilities as well as illustrate programming a plugboard machine.

Description of the Machine

In the following discussion, reference should be made to the machine block diagram in Fig. 1.

For input, the plugboard machine has two punched card feeds, each of which is independently controlled by the plugboard instructions. Information from each feed goes directly into the main storage but is completely buffered so that computing may be done on data entered from the previous card while the next card is being fed. At this point, it should be emphasized that only operands, arguments, and parameters need be entered into the computer through the card readers. All instructions and logical control are "externally" wired into the plugboard by the operator. Additional input is obtained from an array of 10-position switches known as a "parameter board." This device is attached directly to the main channel and through it the operator may change parameters at any desired point during the computation. Each parameter value may be called for by the plugboard routine and if previously set up by the operator, is instantaneously available.

For output, a buffer of ten words connected directly to the printer will allow computing to continue and the next set of output data to be stored, while the previously computed results are being printed. Each word of storage in both the input and output buffers is a part of the main storage and may be used and addressed in the same manner as any other storage location. This input-output setup is second to none in actual usefulness. As higher speed input and output devices become generally available, they need only be attached to the buffers. It should be noted that tapes are generally considered to be a form of output, however, humans cannot read them and every tape must funnel through a printer somewhere.

A second and very important form of output is the "selectable diagnostic list" function. By the mere flip of a switch on the control panel, each number passing through the storage register is automatically listed on the output printer. The storage register is a central buffer tying the main storage to the arithmetic unit. A second switch may be used so that output occurs only at predetermined "break point" locations. No programming effort is necessary to use this feature. This is perhaps the easiest-to-use routine checking device yet devised.

Computer design form is the next major consideration. To be able to retain low pulse frequencies and yet obtain high computing rates, the machine must be highly parallel. The main channel has ten lines so that the nine decimal digits and sign are transferred in parallel. Additionally, the functions controlled from the plugboard are established so that a maximum number of operations may be accomplished in one plugboard program step. This highly parallel operation, or expressed in the nearest equivalent in internally programmed machine language, "multiple-address system," contributes greatly to effective speed. In internally programmed machines, even though several instructions may be packed in one word, the execution of the instructions is necessarily serial and may require much storage access time.

In some instances it is desirable to provide the computer with a "standard board." This means that a board will be wired with all the necessary standard functions such as add, subtract, sine, cosine, square root, etc., appropriate to a class of problems. With such a standard board this machine becomes an internally programmed computer with zero instruction time within subroutines. For this purpose an "interpretive routine decoder" is attached to the main digit transfer channel to allow us to decode a word in a single program step by placing individual digits directly into selected computer control elements. Thus, stored words are interpreted as instructions which function as connectives between wired subroutines. By the use of this

REX RICE, JR., is with Northrop Aircraft, Inc., Hawthorne, Calif.

technique, one-time problems may be run without requiring any special board planning or wiring. It appears that a combination of standard-board use, together with individual boards for repetitive problems, will optimize programming.

Word length in a computer is a much debated subject; however, for engineering and most scientific computations, operands of nine decimal digits seem to be sufficient. Double precision may be used in the rare case where greater accuracy is required. Additionally, nine digits lend themselves nicely to shift control and function control in the distributors, thus assisting in programming ease.

Storage size is also a debatable subject. However, experience indicates that the main high-speed storage may logically take either of two forms. A machine with 100 words plus the buffers in the main storage, and with a drum or similar device providing an optional larger storage addressable in blocks of ten words, will give a very powerful computer. In such a machine, any location in the main storage may be addressed by codes of two decimal digits. A more expensive and also more useful configuration would be a machine with a capacity of 1,000 words, plus the buffers in the main storage. This would require storage addressing registers of three digits. Both of these systems may seem small by comparison to the large computers. However, the experience at Northrop has demonstrated an important fact that should be emphasized: In a properly designed computer, the plugboard is at least equivalent to 1,000 words of zero-access instruction storage. A rack of plugboards represents all of the subroutine storage normally found in drums, tapes, etc. Additionally, the paralleling of many operations allows a small number of program steps to do the equivalent work of a large number of single-address instructions. These features, coupled with good input-output facilities, make comparison with internally programmed techniques and machine size requirements meaningless.

The control of storage access for this plugboard machine is unique in the field of computing. The nearest approach in internally programmed machines is the *B* register concept. Access to all buffer storage may be controlled either directly on a program step by plugboard wires or from the numbers set up in any one of the "storage address registers." Access to the balance of the main high-speed storage on a given program step is controlled by the numbers in any one of the storage address control registers. As shown in the block diagram, the address registers are contained in a separate little computer, yet may be entered directly from the main computer channel. For example (see Fig. 1), on a previous program step, a number may either have been emitted directly into register *1* or it could have been inserted in *1* by bringing it over the main channel. On the program step now active a wire from a program step exit to the function "storage readout per *1*" would transfer the word corresponding to the number standing in location *1* to the storage register where it becomes available for use.

In addition to the storage address register controls and controlling functions, there are available on the plugboard automatic comparisons of pairs of the registers. These comparisons are set up between registers *1* and *2*, *2* and *3*, and *3* and *4*. By the simple insertion of a single wire, one of the logical tests of the contents of register ($1<2$, $1=2$ or $1>2$) may be used to transfer control. Similar tests are available for comparisons of the other registers. These tests do not have to be programmed; they are automatically available. An example of one use of pairs of address registers is to have the count keeping track of the number of elements in a matrix row or column in register *1* and to count the computing cycles in a loop in register *2*. When these values are equal the $1=2$ impulse may be used to "pickup" a selector which will transfer control to another routine. The usefulness of the storage address registers and the comparison devices may be better appreciated if the basic machine cycle is discussed first.

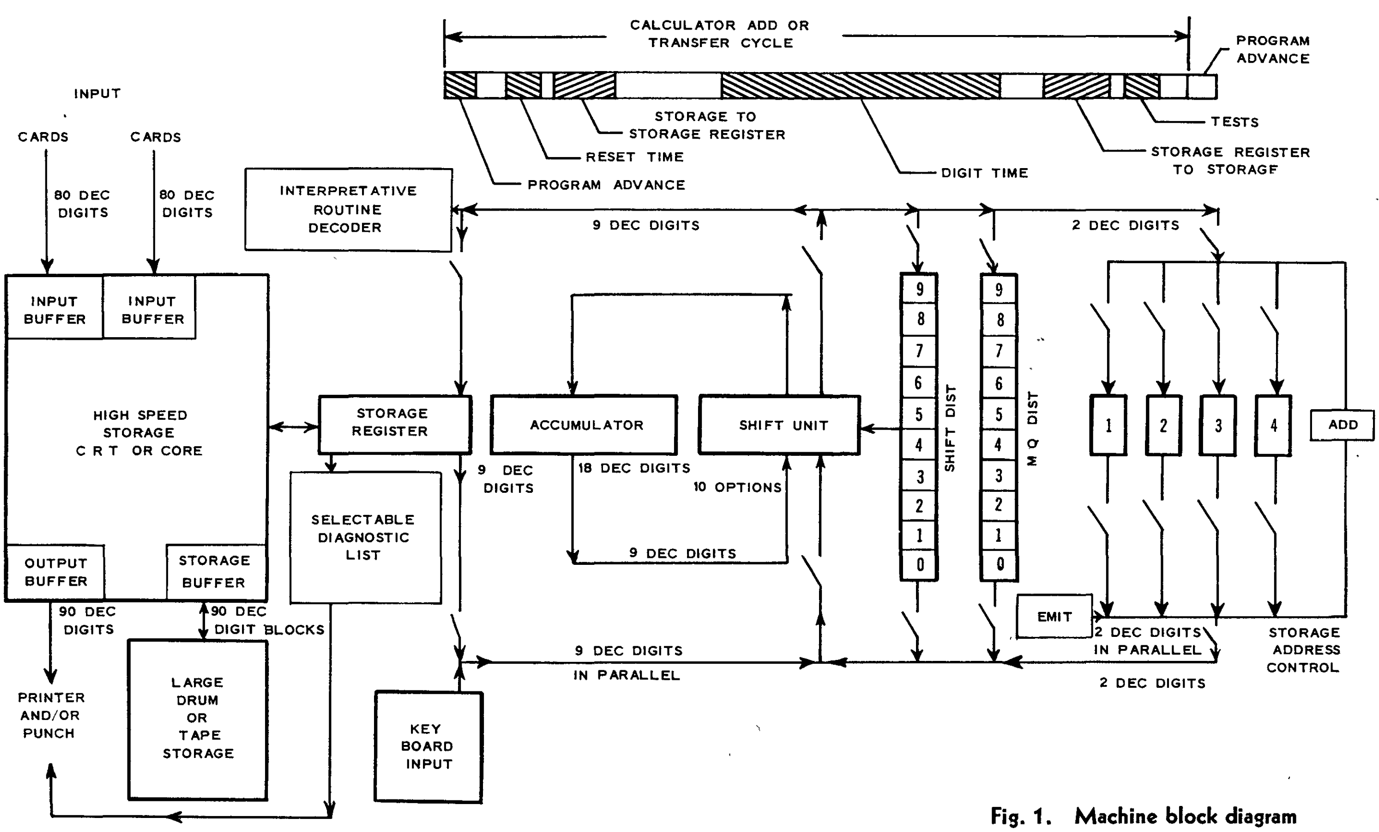


Fig. 1. Machine block diagram

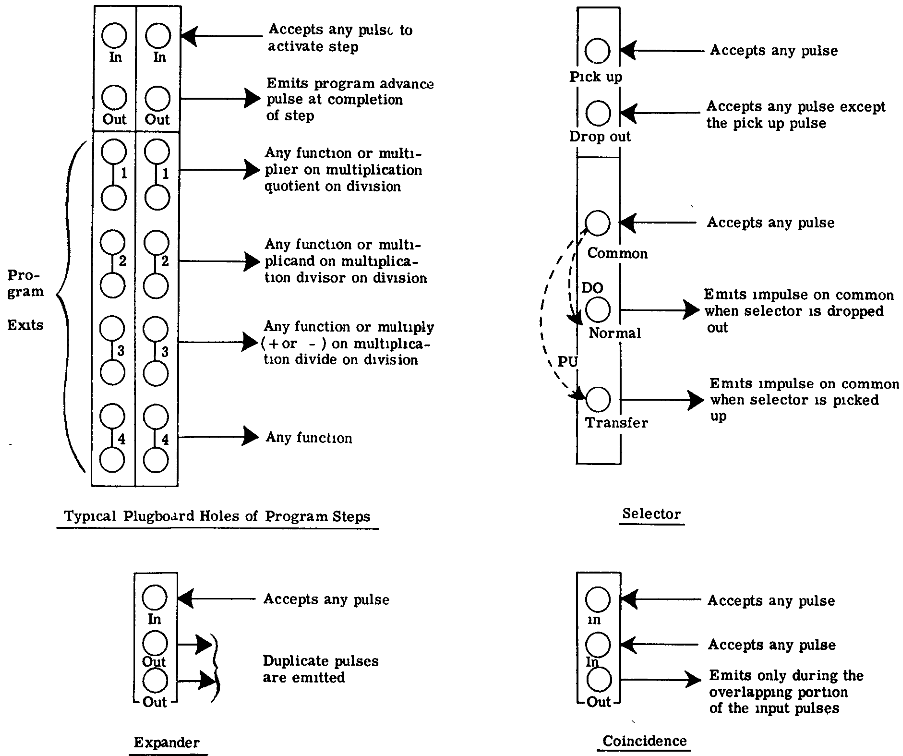


Fig. 2. Plugboard elements

The best way to understand machine operation is to study the timing diagram at the top of Fig. 1 where a simplified primary timer diagram is shown. In machine operation this represents the time of one add or transfer cycle and in the solution of a problem represents the sequence of events during one program step.

On the left is the program advance time. At this time in the cycle a pulse is emitted from the plugboard connection known as the OUT hub of the program step that was previously active. This pulse is available after that step has been completed. It may be used to activate the program step under consideration by putting a wire from the OUT hub to the IN of the step we wish to activate.

The next event is the automatic and/or plugboard-controlled reset of various machine registers. Following reset, any number called for, either by plugboard wire control or from address register control, depending on which is wired on this step, will be transferred from the main storage to the storage register. For example, if on a previous step we had emitted a 50 into address register *1* and on this step we called for "storage read-out per address register *1*," then the operand standing in storage location *50* would be transferred to the storage register.

Next in the primary cycle comes digit time. During this period numbers may be transferred around the machine on the main channel. Examples are as follows: The operand in the storage register may be moved through the shift unit matrix and into the accumulator. Note that shifting is accomplished as numbers are moved along the channel and does not require an extra program cycle. If desired, numbers can be moved from the storage register through the shift unit and into an address register or a digit distributor, whose functions are explained later.

After digit time it is possible to place the number in the main storage from the storage register. For example, "storage read-in per address register *3*" may be called for. At the end of the primary time cycle all automatic tests are performed and the appropriate board hubs are activated. Some of these tests are accumulator +, accumulator −, overflow +, overflow −, channel 0, improper divide, and address register comparison. The use of these hubs in logical operations will be illustrated in the example to follow. By comparison with most internally programmed machines where such tests must be programmed, and where much machine time is required, these automatic tests represent advances in both ease of programming and in reduced computing time.

An additional operation that may be programmed to parallel other operations during digit time is the changing of a number in a storage address register. An example of a parallel operation is perhaps the best way to show this. It is possible on one program step to move a number from the main memory into the storage register per address register *1;* then during digit time, to shift it and add or subtract it into the accumulator. Also, during digit time it is possible to use the digit "emitter" and increase the number standing in *1* by any desired amount. Following this it is possible to read the number still standing in the storage register into some new main storage location by calling for "storage read-in per *1*." In this step we have paralleled the following operations: 1. add a word in memory to the contents of the accumulator; 2. shift for proper decimal alignment; 3. change the number in an address register; and 4. make a transfer of the number originally called out from one main storage location to another.

The digit distributors previously mentioned perform several useful functions. The one on the left in the block diagram in Fig. 1 is called the shift control distributor. It may be used in many ways but in addition to its automatic controlling of shifts during multiplication and division, it is available to the programmer during add or transfer cycles for both shifting control and for logical control. A digit brought into the distributor on a previous program step may be used to control shifting on the active step by putting in a wire which calls for "shift per shift control distributor." Optionally, from outlets on the plugboard, a digit previously stored in the shift control distributor may be used to activate a corresponding hub for picking up or dropping out selectors used in logical control.

The distributor on the right in the block diagram in Fig. 1 is the "multi-quotient distributor." In addition to its automatic use in multiplication and division it is also available for logical control during add or transfer operations.

In a paper of this nature it is not practical to discuss all of the various combinations available to the programmer; consequently, discussion has been limited to a few illustrative uses. The ease of use and flexibility when all of the machine components are available to the programmer cannot be fully appreciated until one has run a problem on the machine.

Plugboard Elements

During the previous discussion frequent reference has been made to the plugboard. At this point an examination of some of the elements on the plugboard is in order. A list of the many machine commands available on the plugboard is too long to be given here. However, complete control of all necessary machine elements and their functioning is available on the plugboard. See Fig. 2.

The program step is the fundamental element on the plugboard used to control machine functions. A balanced computer should have approximately 200 program steps available. Each program step will have hubs available with the functions shown in Fig. 2. The IN hub will accept any pulse to activate the step and start a primary timer cycle. During the cycle the four program exits will become active and wires plugged from them to any machine function will activate the function. As frequently happens in parallel operation, if four exits are not enough, one of them may be plugged to the IN of an expander which will then duplicate the pulse on its OUT hubs.

The coincidences and the selectors are available as board functions for logical control and in a limited sense for storage. The selectors have five hubs: a pickup, a drop-out, a common, a normal, and a transfer. The pickup hub will accept an impulse from a program exit, from the OUT of a coincidence, or from one of the digit outlets of the shift or multi-quotient distributors. It causes an internal connection to be made between the common hub and the transfer hub. This connection remains until the drop-out is impulsed, at which time the selector connects the common to the normal. These selectors are electronic and operate practically instantaneously. In conjunction with the selectors, coincidences allow complete and easy logical control.

A coincidence has two IN hubs that will accept any pulse. During the overlapping portion of these pulses the OUT hub emits a pulse. For example: If on program step 2 we wish to test the accumulator for a negative number and transfer control accordingly, we plug one of the program step 2 exits into an IN of the coincidence. The automatic accumulator negative test is plugged into the other IN. If both of these conditions coincide, then a pulse will be emitted at the OUT. This hub may be connected to the pickup of a selector. Through the selector we may control a jump to any other program step by wiring the program OUT to the common of the selector. The normal and the transfer of the selector may be wired to the appropriate IN hubs of any desired steps that start other routines. By proper wiring, program loops, conditional transfer, unconditional transfers, and many forms of logical control may easily be accomplished. It is interesting to note that a series of logical conditions controlling transfer are exceptionally easy to establish by merely using selectors in series.

Use of the Plugboard Machine

A good way to illustrate programming operations, machine functions, parallel operation, and simplicity of board wiring is to use an illustrative example. See Fig. 3. A square root routine similar to that used on desk calculators is a good example of what a parallel machine can do. The following is a version of this routine as worked out by M. L. Lesser and T. S. Eason.

The basis of the process is the following fact from number theory:

$$n^2-\sum_{j=1}^{n}(2j-1)=0$$

That is, the sum of the n successive odd integers from 1 through $2n-1$ is equal to n^2. This can easily be verified by examining a table of n, n^2, and the first difference of n^2.

Thus, given n^2, n can be determined by simply counting the terms in the summation of the successive odd integers when this summation is built up to equal n^2. This process can be shortened by performing the operation as a subtraction on radicand groups of two digits each to produce one digit of the root. These 2-digit groups of the radicand are formed by measuring in each direction from the decimal point as in the long-hand method of high school algebra. The operation of this shortcut method is best described by example. See Fig. 3.

The square root routine described follows the foregoing procedure with a single exception dictated by machine convenience only. By writing $\sum_{j=1}^{n}(2j-1)$ as $\sum_{j=1}^{n}(j+j-1)$, the subtracted terms count by successive integers, rather than by successive odd integers.

$$n^2-\sum_{j=1}^{n}(2j-1)=n^2-\sum_{j=1}^{n}(j+j-1)=0$$

EXAMPLE:

$$\sqrt{144}=12$$

$$144=\sum_{j=1}^{12}(j+j-1)$$

$$=\sum_{j=1}^{10}(j+j-1)+\sum_{j=11}^{12}(j+j-1)$$

$$=10^2\sum_{j=1}^{1}(j+j-1)+\sum_{j=11}^{12}(j+j-1)$$

ACCUMULATOR	TRIAL ROOT	PROGRAM STEP NUMBER
144		
-000	000	2
144		
-100	100	4
44		
-100	100	2
- 56 OVERDRAW		
+100	100	4
44 RESTORE	10	5
- 10	10	2
34		
- 11	11	4
23		
- 11	11	2
12		
- 12	12	4
0		
- 12	12	2
- 12 OVERDRAW		
+ 12	12	4
0 RESTORE, FINISH	12	5

Fig. 3. Square root routine

Fig. 4 shows a planning sheet which corresponds almost exactly with the "diagram" and "flow chart" combination of the planning for an internally programmed machine. This is the first of three operations needed to get a program into a plugboard. It is the only operation which requires any original thinking; the other two merely are translations of information planned on this diagram.

In Fig. 4 the large rectangles each represent a program step with the functions that occur on that step shown inside. The IN of the step is at the top and OUT is at the bottom. Selectors are shown as branches that might be used to alter either machine functions or to transfer control. Coincidences are shown as smaller rectangles. The connections of their OUT hubs are noted. The following describes step by step the operations necessary to obtain the square root. The program step numbers are shown just above the upper-left-hand corner of the large rectangles.

Step 1

Setup Step

1. "Reset shift control distributor" which will be used to tell what column of the root we are developing.

2. "Reset multi-quotient distributor" which will be used to tell us when we have made nine tries. This is used to prevent automatically a tenth try that we know will always cause an overdraw.

3. "Read-in addressed storage (main storage location) per number in address register *1*." Since no numbers are on the channel this clears the storage where the root is to be developed.

4. "Advance shift control distributor" to *1*. This function adds one to whatever number is standing in the distributor (i.e., zero, since the shift control distributor was also reset on this step). The shift control distributor is used to locate the position in the accumulator from which the subtraction is made.

5. "Block reset" selectors *1* → *10*. This insures that a block containing all the selectors used are in the "dropped-out" condition.

Step 2

Reduction Step—first subtraction. Reduction occurs if all selectors are normal (dropped out) as shown in Fig. 4.

1. "Read-out addressed storage per *1*." This number will be zero on first cycle and will be the trial root on following cycles. Note that the trial root remains in the storage register.

2. "Out per shift control distributor." This sets up the shift unit in the proper fashion for subtraction of the trial root from the radicand in the accumulator.

3. "Accumulator subtract." The function is through selector *1* so that it becomes "add" if the selector is "picked up."

4. "Test accumulator for negative." A coincidence between a negative number at test time and this program step is used.

5. "Accumulator negative" is also wired directly to "selector *1* pickup." This will cause an addition on steps 2 and 4 for restoring after an overdraw.

Note: If an overdraw occurs during this first subtraction, selector *2* is "picked up." This, in turn, "picks up" selector *4* during program step 3, to change the loop. Selector *4* is the logical control and storage for the indication of completion of one series of subtractions which will develop one digit of the trial root. Since the accumulator is negative, selector *1* is also picked up. This causes an addition instead of a subtraction to occur during step 4, bringing the radicand back to the appropriate value for the shift and next trial reduction. Since selector *2* is picked up, an increase in the trial root is prevented so no subsequent subtraction in its tally position in the storage register is necessary.

Step 3

Modification and Storage of Trial Root

1. "Address channel read-out" connects the emitter to the main channel.

2. "Emit *1*." The emitter is activated to tally the subtraction on the previous step. This function is active when selectors *2* and *3* are "dropped out."

3. Shift "out per shift control distributor" places the digit 1 on the channel in the correct shift position.

4. "Read-in addressed storage per *1*." This stores the modified trial root in main storage.

Step 4

Reduction Step—second subtraction

1. "Read-out addressed storage per *1*." This brings the trial root back to the storage register.

2. "Out per shift control distributor" sets up the shift unit so the trial root may be placed in the proper position of the accumulator.

3. "Accumulator subtract" (add if selector *1* has been picked up). This subtracts the trial root a second time or adds if an overdraw was encountered on step 2.

4. "Multi-quotient distributor advance." This function adds a 1 to the multi-quotient distributor only after each double reduction. This distributor is set up so that a 9 in it indicates the third type of possible overdraw.

5. "Test multi-quotient distributor for 9." This function is set up through a coincidence of a digit 9 in the multi-quotient distributor and the accumulator being positive. When these occur simultaneously, selector *4* is picked up. When this condition occurs we know that another subtraction will always cause an overdraw. Since we have not yet subtracted, no reduction cycle is necessary; consequently, we may proceed with the setup for the next position of the trial root.

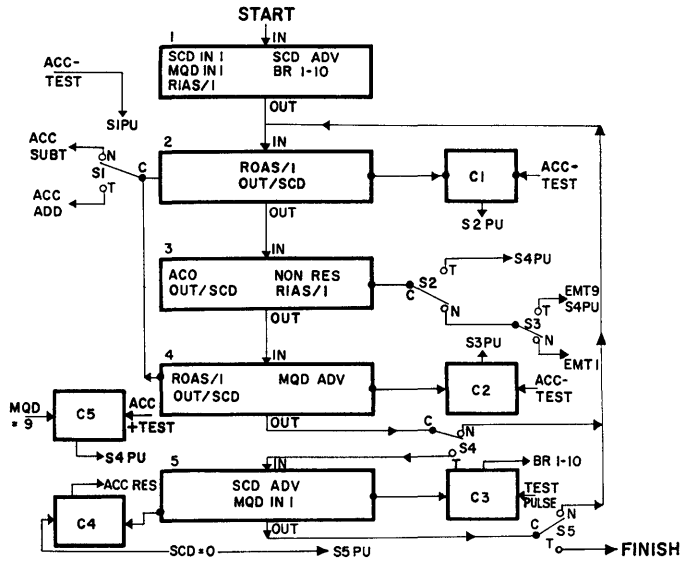


Fig. 4. Diagram and flow chart of square root routine

Setup:
Radicand in accumulator
Decimal at any even period from left address of root in 1
End:
9-digit root in AS/1
Accumulator reset
Negative radicand yields a zero root
ACC = accumulator
S = selector
C = coincidence

The equation in Fig. 3 indicates that for machine convenience we chose to increase the trial root by unity on each subtraction. If overdraw occurred after the first subtraction, it is only necessary to add the trial root back once; however, if overdraw occurs after the second subtraction, it is necessary to add the trial root back twice. This is accomplished by repeating steps 2, 3, and 4 with selectors *1* and *3* picked up.

Selector *1* is transferred automatically any time the accumulator is negative. Selector *3* is picked up by a coincidence of a negative accumulator occurring while program step 4 is active. Note that when selector *3* is transferred, it will cause selector *4* to be picked up on the next cycle through the loop. Also, the transferred function of selector *3* is to "emit 9," instead of "emit 1." Since we are adding into the storage register without carry between digits (using the "non-reset" function) and since the register provides for an end around carry within each digit position, the addition of a 9 has the effect of decreasing the digit contained in the register position by 1. This operation restores the trial root digit under consideration to the proper value if an overdraw occurred on the second subtraction.

Step 5

Second Setup and Shift

1. "Shift control distributor advance." This adds a 1 into the shift control distributor to control the shift so that a new position of trial root is developed on the following cycles. After we have developed a full nine digits of the root, an advance of 1 in this step brings the shift control distributor around to zero. This condition is used to pick selector *5* which signals completion of the routine and also is used through coincidence *4* to reset the accumulator after the last cycle.

2. "Block reset of selectors *1* through *10*." This is accomplished by the coincidence of a test pulse that occurs late in the primary timer cycle and the fact that program step 5 is active. The selectors are reset so the setup for the next trial root cycle starts properly. It should be noted that if "block reset" is impulsed while a pulse is on the pickup hub, that particular selector will remain picked up. This condition occurs with selector *5* after the final reduction cycle.

3. "Multi-quotient distributor in *1*." Since no numbers have been placed on the channel during this program step, this causes the multi-quotient distributor to read in a zero and set it up for the next reduction cycle.

The transferred point of selector *5* may be connected to the IN hub of the next step in the program so that succeeding computations may be started.

The previous discussion on the planning of a routine completes the first operation in the setup of a job. It should be emphasized that the planning of this routine is exactly comparable to planning any subroutine on an internally programmed machine. Once the routine is worked out, it may be used in any portion of a program and placed anywhere on the board by merely changing the numbers associated with the program steps, coincidences, and selectors.

The second operation involves the transfer of this information onto a wiring chart. Fig. 5 shows the chart filled in for the square root routine. On the left we find the hubs for the program steps enumerated. Reading from left to right we find the IN hub, the program exist *1*, *2*, *3*, and *4* and, finally, the OUT hub. The selectors and their hubs follow. On the left we find the pickup hub followed by the drop-out, the common, normal, and, finally, the transfer hubs. Next, the expanders are shown. From left to right we find an IN hub and then the two OUT hubs. Finally, the coincidences are shown. The two IN hubs precede the OUT hub. Above and to the right the common hubs and the outlets for the multi-quotient distributor and the shift control distributor are shown.

In practice this wiring chart is used to establish both ends of each wire required. For example, in program step 1, outlet hub *3* shows the designation *E7*. This is the abbreviated code for expander *7*. By looking in the column for expanders we find that the IN hub of *7* is marked *P1-3*. This tells us that the other end of the wire comes from program step 1, hub *3*. Similar conventions are used elsewhere.

The third operation in establishing a program is to wire a board from the wiring chart. This is usually done by placing a check mark in each box as the wire is inserted. Operations 2 and 3 may be performed by coders or machine operators. Contrary to common belief, the time necessary to wire and check a control panel is insignificant compared to the over-all time spent in the analysis and programming of a problem. Control panels for a machine of this type may be wired in 6 man-hours for problems of a difficult nature such as missile launching trajectories, which include all the various aerodynamic and inertial complexities. This, no doubt, compares favorably with checkout time on an internally programmed machine for such a complex problem. Machine time for board checkout may be as low as 1/2 hour for any person who has wired two or three boards previously.

The discussion to this point has been devoted to a hypothetical plugboard computer; however, the techniques involved and the various machine functions discussed are not pure theoretical speculation. Northrop Aircraft has been using a new type of experimental International Business Machines plugboard computer for over a year. This machine closely approximates the one discussed in this paper. Our experience using this computer has borne out our belief that a properly designed plugboard machine can be easier to program, less expensive to rent or purchase, and, most certainly, be as versatile as the largest available computer. It is interesting to note that on this experimental machine the square root takes a maximum of 25 milliseconds, which is equivalent to two "divide-times."

Advantages

The first consideration in the evaluation of computers should be to obtain the maximum computing per dollar expended. The costs of a computing installation, in general, may be divided into two sections. The first section covers the cost of the machine or its rental, as the case may be, plus installation costs. If we consider that the rental of a large, internally programmed machine, or the equivalent amortized cost of a purchased machine, might approximate $25,000 a month, then its yearly cost will be $300,000. By comparison, a production version of the plugboard machine type described would rent for less than $8,000 per month and two of these machines would have more net output than one large, internally programmed machine. The yearly rental for two of these plugboard machines would be $192,000, as compared to the rental of $300,000 for the large, internally programmed computer. Judgment on machine capabilities must take into consideration the total time spent on a computer and not merely be concerned with compute speed alone. The best information available shows that the time on the larger, internally programmed machines is as follows: 50 per cent of the time the machines are input- or output-limited; 25 per cent of the time they are limited by humans in making setups, check-outs, and in trouble-shooting; 25 per cent of the time they are compute-speed-limited. These figures are based on engineering computing requirements and, most certainly, will vary in other applications and for specific problems; however, they seem to represent a rough average.

The second section in the cost of running a computing installation covers the salaries paid to the programming personnel. As shown by experience, between 30 to 35 people are required to operate a large, internally programmed computer. If it is assumed that their hourly rate, including overhead, approximates $6 per hour, then the total salary paid per year will be $360,000. It should be noted here that the manpower costs exceed the equipment costs. Due to the ease of programming, 20 people could produce the same results with two properly designed plugboard machines. At the same assumed labor rate, the total expenditure for these people would be $240,000 per year.

Summarizing these figures, the total assumed cost of operating a large, internally programmed machine is $660,000 and for the equivalent output in a properly designed plugboard machine, the cost would be $432,000.

A few remarks regarding the type of personnel required to operate a plugboard machine are in order since it has been said that extraordinary people are needed to program a plugboard machine. Experience has shown that this is not

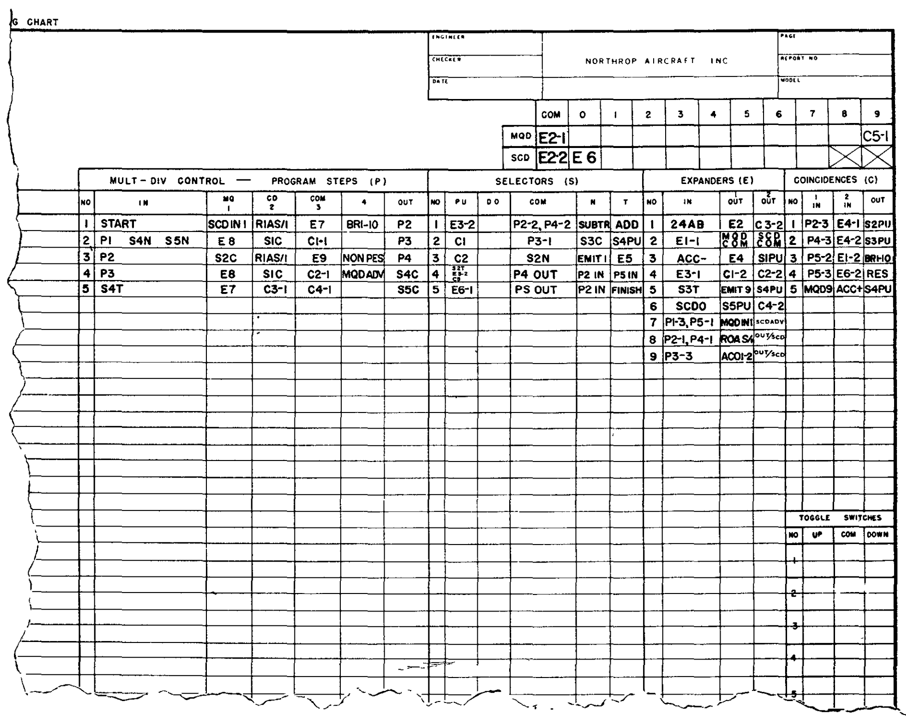


Fig. 5. Wiring chart

true. The people who program the equipment at Northrop have the same background, experience, and abilities, on the average, as the persons programming internally programmed computers in other installations. However, there is one difference in our operation. The programmer is shown that he has available the fundamental building blocks from which he can construct any desired computer. As a result, the programmer is unhampered by the limitations of a built-in operational code. The flexibility and potentialities of a plugboard computer are limited only by the ingenuity of the programmer. In effect, he is told, "Here are the components; design a special-purpose computer which most efficiently does your problem." This approach stimulates original thinking and results in increased interest, initiative, and productivity on the part of the programmer.

Table I. Record of Average Learning Hours for Programmers

Programmers' Background	Time Required to Learn Machine and Program Simple Problem, Hours	Time Required to Learn Sophisticated Use, Hours
IBM *604*, etc.	4	20
Engineers; no computing experience	8	24
Engineers; previous CPC computing experience	2	16
701 programming experience	2	8

Table I shows an actual record of the average learning hours used for people programming the machine now at Northrop.

In the comparison of computer types, the second and final consideration is the ability of the computer to do the problems experienced in a given installation. Table II lists most of the important features for both types of machines.

Table II. Comparative Features of Large Internally Programmed and Plugboard Machines

	Large Internally Programmed Machine	Plugboard Machine
Operand storage	Electrostatic or core, drum, tape, cards	Electrostatic or core, punched cards
Instruction storage	Anywhere in storage	Plugboard wires, machine components, main storage
Operand access	Variable, program for optimum	Automatically optimum
Instruction access	Variable, program for optimum, abstractions increase net access	Effective zero-access
Rapid access storage	Crowded by stored instructions	All available for operands
Logical control	Always programmed, every command serial	1. Tests fully automatic 2. Performed in parallel with computations 3. Series of conditions easy to construct 4. Simple restoration 5. Many types of control more easily available
Operation	Serial	Highly parallel
Computing rate	Should be high because of serial operation	May be order of magnitude lower
Programmer's "feel"	Usually abstract	Direct, physical
Input	Cards, tape, console keys	Two card feeds, parameter board, plugboard
Output	Line printer, cards, tape	Line printer, cards, tape
Physical size of machine	Assume *704* or *1103* as unity	Estimated one third

This paper has described a computer which is small in size, large in ability, and unsurpassed in versatility. After considering the flexibility, the ease of programming, and the reduced operating cost, the obvious question is, "Why not try a plugboard?"

Discussion

H. R. J. Grosch (Chairman): I would like to begin the discussion by asking Mr. Rice if he has had any experience in teaching this type of equipment to someone who started in the stored program area, or something similar.

My criticism is that it seems to me a very difficult machine to learn in comparison with the stored program.

Rex Rice, Jr: One of the questions that has been asked about this machine by people who have viewed it has been, "Well, isn't it difficult to program?"

We have had people with varied backgrounds actually put problems on the machine and we consider that they are ready to try a problem when we have described the machine's functions to them and put a few wires on the board with their help (but essentially with their watching the operation). We then turn a simple problem over to them.

This simple problem is shown in the middle column of Table I. People with IBM *604* experience of the type where they are familiar with a plugboard take about 4 hours to be able to start thinking about this machine. Engineers with no computing experience take about 8 hours. Essentially we have to show them what the program steps, selectors, coincidences, expanders, and things of that nature mean and give them a physical feeling for what is happening in the machine.

Engineers with previous CPC experience are done in about 2 hours. They merely have to find where the holes are in the board. The same applies to people who know how to set up logical operations, who know what routines mean, and things of that nature. It takes us about 2 hours to give, for example, *701* programmers the location on the board of the hubs. This varies with individuals, but I have given you an average figure.

On the right, people with *604* experience take about 20 hours to become sophisticated, meaning that they can code a problem of their own, get it to run without undue difficulty, and they begin to see the potentials of parallel operation.

For engineers with no experience, since they have to learn loops and programming, it takes about 24 hours; for engineers with previous computing experience, about 16 hours; and for internally programmed experience, about 8 hours.

I believe that this record should stand against any internally programmed machines.

V. M. Wolontis (Bell Telephone Laboratories): Opposition to plugboard machines is not universal in the data-processing area. There are many problems—for instance, calculation of means and variances of tabular data—solved more economically by a *604* than by a *650*.

J. Belzer (Battelle Memorial Institute): Are you discussing a specific computer? If so, which one?

Rex Rice, Jr.: The specific computer that I have had available for this experience is the experimental machine in operation at Northrop Aircraft. This machine was produced by IBM.

As far as the hypothetical machine is concerned, it is not as yet announced by any computer company. I hope that my remarks here will prod them in this direction.

H. Robbins (Hughes Aircraft): Have you any features comparable to the machine aids to coding possible with internal storage? That is, a program library, automatic assembly of routines, etc.?

Rex Rice, Jr.: This is a difficult question to answer in a short time. However, let it suffice to say that the library of routines will represent, as we illustrated in the square-root routine, a worked-out wiring chart which some programmer has made and has boiled down to the most efficient subroutine possible. This then gives us a complete library, as they are developed, of subroutines.

The "splatter function" (effectively the interpretive routine decoding device on the channel) was purposely placed there so that we could decode one word in the memory, and this allows us logical control of the transfer between subroutines.

I would like to emphasize that fact, because if you have not worked with a plugboard machine it is not obvious. All of the instructions within the subroutine are contained in zero access storage on the plugboard. The word necessary to transfer control to any other routine desired at the end of that subroutine may or may not, as desired, be contained in the memory. There is nothing in the internally programmed machines that cannot be duplicated easily on this machine. Abstractions become useless because after all of the abstractions are developed on an internally programmed machine you merely approach what we are already doing.

K. F. Powell (Babcock and Wilcox): Despite that fact that this machine is quite attractive, it is not apparent why it should reduce the size of the programming staff.

Rex Rice, Jr.: I will have to base my remarks on experience. The programmers after the first few problems do not have to spend time worrying about the resetting of memory registers which represent the storage for logical control. They begin to get a feel of the fact that a selector is the logical control. They merely insert a selector in the routine, then at any convenient time much later, days later, in their problem or planning if they wish they stick a drop-out wire in one of the hubs of the program step to be sure the selector is dropped out. There is no modification necessary. They do not need subroutine control to make these changes.

This is one example. It is a very difficult thing to define if a person has not actually worked one of these machines. Effectively, it is what we meant in the slide by showing the programmer's feel for his routine. The best answer I can give is that this feel is direct and physical as compared to a relatively abstract feel in the case of an internally programmed machine, particularly when abstractions are used for coding purposes.

R. W. Bemer (Lockheed Missile Systems): Is it true that a certain large user of computers with five IBM *650*'s on order would dearly love to have one of these machines?

Rex Rice, Jr.: I pass.

Characteristics of Currently Available Small Digital Computers

A. J. PERLIS

THE purpose of this paper is to survey a rather well-defined group of computing machines. No attempt will be made to place these machines in order with respect to certain applications. However, a listing of their pertinent characteristics will be given.

The last time the Joint Conference was held in Philadelphia, the proceedings were concerned with "large" general-purpose digital computers, e.g., Whirlwind, Mark III, ERA *1101*, the Princeton series, etc. As a class, they may be roughly characterized by satisfying (*a*) minimum cost in excess of $200,000 and (*b*) a maximum access time to main memory of less than 1 millisecond.

The present conference is concerned with "small" digital computers. This paper will further limit the conference subject by restricting its discussion to the class of "small," general-purpose electronic digital computers, which may roughly be characterized by:

1. A maximum cost less than $150,000.
2. An internal storage of at least 1,000 words.
3. Stored programs.

Thus, such well-known (types of) computers as the IBM CPC, the Remington Rand *409*, the whole class of digital differential analyzers, and other special-purpose computers are not included. Specifically, the following computers, with their respective manufacturers, are to be surveyed.

1. The *650*, International Business Machines Corporation.
2. The *102d*, National Cash Register Corporation.
3. The *30-203*, ElectroData Corporation.
4. The Miniac, Marchant Research Corporation.
5. The Elecom *120*, Underwood Corporation.
6. The Alwac, Logistics Research, Inc.
7. The Circle, Hogan Laboratories.
8. The Monrobot, Monroe Calculating Corporation.
9. The *E 101*, Burroughs, Inc.

All of the computers listed are decimal with the exception of numbers 6 and 7, which are binary. Several of the firms, numbers 2 and 5 in particular, also manufacture binary machines. These machines are not generally intended to be used in on-line control applications but rather as scientific and engineering calculators and as units of data-processing systems in commerce. Hence, it is no accident that almost all use the decimal system.

The extreme pertinence of the conference subject at this time indicates the value placed on these machines. Industry and commerce, the real source of sales in this business, are definitely awakened to the value of these machines. Perhaps this awakening has not always been accompanied by understanding but it exists and is widespread. The questions have changed from, "What are they?" and, "How will it save money and fit the organization?" to, "Should a digital computer be rented or purchased now?" and, "Should it be large or small?"

For many of these computer firms, a day of prosperity seems about to arrive. Considering some of their checkered histories

A. J. PERLIS is with Purdue University, West Lafayette, Ind.

and the severe financial, personnel, and material difficulties they have encountered it as gratifying to see their vision about to be awarded. More about the future possibilities of this type of machine will be mentioned later.

The first *650* has just been delivered to a Boston insurance firm, and there are more than 400 orders for this machine. National Cash Register has delivered 16 of its *102*, a (binary) computer, with orders received for over 30. Four or five of the *30-203*, two Miniacs, six or seven Elecoms, at least one Alwac, one Circle, and one Monrobot computer have been delivered to customers. The first *E 101* should be available soon. Many of the customers are government projects and most have been purchased as scientific and engineering calculators. Nevertheless, the urge to adapt these computers to business problems has proved irresistible, and for good reason.

While it is true that these computers are extremely flexible, i.e., general-purpose calculators, it is not true that they are as efficient in solving problems that have large combinatorial components, such as collating and sorting, as in solving linear systems, nonlinear equations, and certain types of ordinary (partial) linear (nonlinear) differential equations. The first attempt as an equalizer is, of course, a large secondary storage like magnetic tapes. At least one of the manufacturers (Underwood Corporation) has a machine, Elecom *125*, under construction which provides an electronic sorter, internal to the machine, utilizing the drum.

Table I. · Comparative Computer Costs and Power and Cooling Requirements

Computer	Purchase Rental in Units of $1,000	Number of Tubes and Diodes	External Cooling Requirement and Power
650	3.25; 3.75		16 kw. No external cooling required
102d	99.5; 2.4; 4.1	300 tubes; 4,000 diodes	6 kw. No external cooling required
30.203	136; 4.3	1,200 tubes; 3,000 diodes	13 kw. Air conditioning recommended
Miniac	85	700 tubes; 1,500 diodes	3 kw. No external cooling
Elecom *120*	85	350 tubes; 4,000 diodes	5 kw. No external cooling required
Alwac	55		4 kw. No external cooling required
Circle	57; 78	700 tubes in 18 types of chassis; 0 diodes	$3^1/_2$ kw. No external cooling required
Monrobot	75	650 tubes; 200 diodes	No external cooling required
E 101	32.5; 0.850		3 kw. No external cooling required

There are, of course, many types of data-processing systems needed in commerce. They all require, or at least would prefer to enjoy, automatic access to large data-storage files which, at this date, implies the use of magnetic tapes. A difference appears in the extent of printed output required by some organizations. Insurance firms, as an example, may require almost continuous print-out. Others may require periodic (e.g. daily) outputs of relatively short duration upon which are superimposed more extensive printing but which occur less often (e.g. monthly). In addition, many firms may wish to use the equipment to compute strategies and suggest and evaluate alternate policies. These may involve extensive calculations equivalent to solving many linear systems or inverting a matrix. Such is the case if the techniques of linear programming are used. Here a machine with built-in floating point would be desirable.

Consequently, it is important to realize that an evaluation of these computers is dependent upon and determined by the system in which the computer is to be imbedded. The variety of such systems is large but a rough classification might be as follows:

1. The computer is the central computer of a scientific and engineering laboratory. The computations are largely of a mathematical nature involving extensive calculating time relative to input/output time. The types of problems treated
   (*a*). Vary widely from week to week and many are solved only once, or
   (*b*). Are limited to a number of standard problems which are solved repeatedly with only slight changes in problem structure.
2. The computer is the central computer of a unit devoted primarily to data processing. The computations tend to involve a great deal of input and output relative to actual computing time required.
3. The computer is auxiliary to a large computer in a group like item 1 or 2.

The foregoing considerations lead to an evaluation which places emphasis on cost, computation speed, input, output equipment, flexibility of the order code, ease in instruction of personnel, extent of internal storage, number system, checking features, and reliability, and to a lesser extent, size, number of tubes, and cooling required.

One item which is overemphasized is compactness. These machines will not be introduced in mass quantities in any installation and hence their physical dimensions, at least up to an order of magnitude, are not really important. Indeed, compactness is undesirable if it makes access to any part of the machine difficult. It is one thing to design a computer with a neat gray crackle finish so that it looks like a filing cabinet and another to boast that it fits into a desk-size structure and can even be used for one.

Comparison of the Computers

In Table I are listed the cost, both purchase and rental, the approximate number of tubes and diodes, space, and cooling requirements.

Several comments concerning this table are in order:

1. The rental of the *650* is for a 1,000- and a 2,000-word storage, respectively.
2. The rentals quoted for the *102d* are for a 5- and a 1-year lease, respectively, with the latter including a magnetic-tape unit.
3. The rental quoted for the *30-203* is an order of magnitude only, since no definite rental policy has been announced.
4. The prices quoted for the Circle are for a 1,024- and a 4,096-word memory, respectively.
5. The purchase price of the *E 101* includes 1 year of free maintenance by Burroughs engineers.
6. All rentals quoted include free maintenance with the kind of maintenance depending on local conditions.

Table II contains information relevant to the performance of the various computers.

Several comments concerning this table are in order:

1. The standard speed is stated in terms of the evaluation of $A+B=C$, $C+D+E=F$, $G\times H=I$ to compensate for the different number of addresses in various instruction codes.
2. The speeds where given using optimum programming are for optimum spacing of instructions only, which properly speaking is only available on those machines which have two or four address instruction codes. In general, it is more difficult to optimum code for operand location.
3. The relatively high speed of the *30-203* is attained by having both operands and instructions in the high-speed 80-word memory. In normal operation practically all instructions, and nearly all operands, are so situated. This is actually a normal mode of operation.
4. The Alwac word is divided into four syllables and the fast mode is to space the orders in a syllable in each alternate word. This gives the 15 standard operations per second mentioned. Orders are carried out from the 64-word working storage.
5. The instructions in the *E 101* and Monrobot are physically distinguishable from the number storage and represent a departure from the accepted trend in computer design.

Table III contains information relevant to the terminal equipment of the computers.

Some comments concerning this table are as follows:

1. The asterisk refers to auxiliary, usually punched-card, equipment. The adaptor units for the various IBM machines introduce additional cost which may run from $10,000 to $20,000. The rental cost of the IBM equipment is, of course, additional. The *024* and *523* rent for $30.00 and $65.00 per month, respectively.

2. The input to the *E 101* is by a manually operated keyboard for the introduction of data and by a manually (pre-) set pinboard for instructions.

Table IV contains information concerning the availability of magnetic tapes as auxiliary storage for the computers. Only those machines are listed for which the manufacturer provides tape units.

The following comments are pertinent:

1. In all machines the tapes will search for a desired block while the machine is computing. A single order will thus cause a block to read into or from the machine. No computation proceeds during this time.

2. The tape units are included in the purchase price of the *102d* and the Elecom *120*.

3. The Elecom *120* uses a "sprocket" device to ensure the use of only those portions of tape that are free of blemishes. The sprockets are pulses which indicate that a particular slice of the tape is without blemish. The pattern which is engraved on the tape is not apparently changeable.

Table II. Comparative Computer Performance

Computer	Words of Internal Storage	Average Access Time in Msec.	Number of Addresses in Instruction	Number of Digit/Word Number System	Standard Operations: Per Sec. Normal Mode	Standard Operations: Optimum Programming Instructions Only
650	1,000 or 2,000	2 4	1 + 1	10 Decimal	22	33
102d	1,024 +8	12 5 1.5	3	10 Decimal	16	
30-203	4,000 +80	8.5 0 85	1	10 Decimal	50−	
Miniac	4,096	5	1	10 Decimal	12+	
Elecom *120*	1,000 +10	8.5 1.8	2	Binary 8 Decimal	6.5	20
Alwac	4,096 +128	25 4.2	4 syllables 8 digits each	32 Binary	8	15
Circle	1,024	8.5	1	40 Binary	7.5	
Monrobot	100 instructions 100 numbers	16	4	20 Decimal	1	
E 101	128 instructions (pinboard) 100 numbers	9	1	12 Decimal	2	

A factor determining the value of a computer is the flexibility of its order code. The first thing to note is that none of them is flexible enough. It is almost facetious to say that they are all "Turing" machines with a set of instructions that enables the solution of any problem which can be phrased in a very basic system of logic. This is like saying that one has a completely equipped laboratory for the synthesis of all known organic hydrocarbons no matter how complicated they are, if a source of carbon and hydrogen is present.

Some of the most striking differences between these computers appear upon programming them. The (red-tape) bookkeeping and programmed-arithmetic facilitation orders increase in relative value the more they are used.

In general, the efficiency, both programming and machine-wise, of these computers depends upon subroutines. This implies a facility in adapting a routine to many different programs. Consider the specification of addresses in such routines. They are of two types: (*a*) addresses which specify location within the routine; and (*b*) those which specify location outside of, and independent of, the routine. How easy is it to modify these addresses in adapting subroutines to new programs? In practically every computer the modification must be done by arithmetic manipulation of the commands, which manipulation increases the more addresses (per order) there are to modify. Only one of the machines, the *30-203*, possesses the ability to modify the location of input of previously written routines and the variation of instruction addresses without extensive programming. Indeed, because of the *B* register, the *30-203* possesses more of a programming flexibility than many of the larger, more expensive computers.

A component of almost any program is that of counting or indexing. Each cycle counter has its contents increased, usually by one, and the current counter contents are checked against the desired cycle complete number, N. If the cycle is complete, a new segment of the computation is initiated; otherwise, another cycle commences. The computer assays its position, usually by counting down from $N-1$ to 0, by successive subtractions, usually of one. Only at completion is a negative result obtained. One usually brings the counter contents to the accumulator for 1- and 2-address machines, adjusts, and compares. Only three of the machines, *30-203*, Elecom *120*, and *E 101*, possess such an automatic facility with their *B* register, tally command, and stepping switch, respectively. A saving of between two and four commands per cycle is obtained in these machines because of these features.

In many programs a subroutine is successively entered from diverse program locations by a transfer command. Upon completion control is returned to the instruction following the transfer command. The command or control counter, upon transfer, is the register which always contains the desired information. Only three of the computers (*30-203*, Elecom *120*, Circle) possess instructions which transfer information from the command or control counter to the arithmetic unit or, equivalently, storage.

Each of the computers employs a fixed decimal (binary) point in its representation of numbers. Consequently, programs which involve a large number of arithmetic operations require scaling. In certain applications where computation speed is not as important as analysis and programming time, programmed arithmetic, such as floating-point arithmetic, is employed. Here the interval of acceptable numbers is broadened by, for example, 10^{50} and overflows are much less likely to occur. In any event, either scaling or floating point require systematic use of a normalization technique, which shifts a number until the nonsignificant zeroes are removed and the count of number of shifts (power of 10 or 2) is recorded. Half of the computers possess such a normalization or scale factoring command. For the others programmed floating point must be extremely complicated to prepare.

Similarly, in analysis of variance problems double precision may be required. Here, addition into the low-order positions of the accumulator is extremely valuable. Only the *650* possesses this facility.

A single command, which automatically searches a table in memory to extract a specific word, is extremely valuable for many clerical and mathematical applica-

Table III. Terminal Equipment

Computer	Input: Flexowriter	Input: Card Reader	Output in Digits Per Second: Photoelectric Reader	Output in Digits Per Second: Flexowriter	Output in Digits Per Second: Card Punch	
650		200 punched cards/minute 265 digits/sec			100 punched cards/minute 132 digits/sec	
102d	10 digits/sec		200 digits/sec	10 digits/sec	132 digits/sec from IBM *523*	
30-203	10 digits/sec	150 digits/sec from IBM *513, 514, 517, 523* 300 digits/sec from *528*	540 digits/sec	10 digits/sec (typewriter) 14 digits/sec (tape punch)	150 digits/sec from IBM *513, 514, 517, 523, 528*	
Miniac	10 digits/sec		1,000 digits/sec	10 digits/sec		
Elecom *120*	10 digits/sec	20 digits/sec for IBM *024* 150 digits/sec from IBM *523*	200–400 digits/sec	10 digits/sec	20 digits/sec from IBM *024* 150 digits/sec from IBM *523*	100 digits/minute by programmed control of IBM tabulator
Alwac	10 digits/sec			10 digits/sec		
Circle	10 digits/sec			10 digits/sec		
Monrobot	10 digits/sec			10 digits/sec		
E 101		Keyboard controlled		Computer controlled Sensimatic accounting machine only for data Pinboards for instruction		

tions. Only the *650* possesses this facility, although it may easily be programmed in the others.

Simple instruction codes are ideal for the novice and for the casual user. However, extended use of the machine always requires the extension of programming techniques. The more flexible the order code, the more easily this is accomplished. After all, each of these computers possess a subinstruction code which is basic and easy to teach and logically sufficient. It is the icing on this cake, which is important when the machine is used.

Few programs are correctly written on the first try. If the problem load does not make it impossible, the computer should be used to help check and correct a program. What facilities do these computers offer which aid in checking and correcting? The important ones seem to be:

1. The program can be manually stepped from one program order to the next (*650, 102d, 30-203,* Elecom *120, E 101*).

2. The contents of the arithmetic and control registers are visually accessible. In some of the computers all of the register contents are accessible under console manual control only by typing (Elecom *120*). In others (Circle, *E 101*) only the control register contents may be typed. In the *650* all registers can be visually displayed, but only one at a time. In the *30-203* all registers are displayed at all times.

3. The contents of every memory position can be directly typed or displayed on the console by a keyboard manipulation.

4. There is a provision for step or single command operation.

5. A manual transfer of program control can be initiated by the programmer from the console.

There are, of course, other items that are important, which depend on how the machine is to be used.

Of course, all of the aids to the programmer, flexibility of order code, and extreme operation speed are superfluous if the machine makes frequent errors and is difficult to maintain.

The engineers should have access to a control panel, which is a good deal more extensive than the programmer. For example, it should be possible to operate the computer from a test source on variable frequency, to investigate marginal behavior, and to pace every command by the so-called single pulse operation. Every register in the control and arithmetic unit should have a manual input so that arithmetic checking can be divorced from memory and terminal equipment checking.

One of the proved techniques is to vary tube voltages to find marginal electronic components. Most of these computers possess these facilities.

All of these machines use tested components in circuits of conservative design. Loading almost everywhere is below recommended minima for extreme reliability. Reliability is an important factor in the use of these machines. After all, to do a given computation, they must work longer periods without error than do the larger, faster computers. Where speed increases by a factor of ten, say, complexity goes up by a smaller factor and reliability decreases by a still smaller factor. Machines which operate at a speed factor of 100 over these computers are still well within the limits of reliability imposed by electronic complexity.

Many of these computers use automatic checking circuits to facilitate error tracking. The Elecom *120* uses an excess 3 binary code for decimal representation. Thus, the digit patterns 0000, 1111 should never occur. Inputs and outputs from memory are checked for these excluded digit patterns. Their occurrence stops the machine and the cause of the error is somewhat isolated. The *650* and the *30-203* use similar devices. They use the biquinary and straight binary code, respectively. In both, forbidden digit conditions at check points in the computer result in automatic machine stop.

The computers check memory selection by checking the address generator for a coincidence after a complete drum revolution.

Table IV. Availability of Magnetic Tape as Auxiliary Storage

Computer	Words Per Tape 1,000	Cost Per Tape Unit $1,000	Reading of Writing Speed Characters Per Second in 1,000	Average Tape Search Time in Seconds in 1,000	Maximum Number of Tape Units Machine May Control
102d	115	19.5	0.6	0.040	7
30-203	160	14.7	5.0	0.098	10
Miniac	26	5.0	0.915	?	?
Elecom *120*	125	?	0.31	0.7	10

Applications to Data Processing

By and large, these commercial computers will prove to be reliable in operation and increasingly more so as time goes by. Future manufacturing efforts in this field will tend to emphasize this aspect of smaller computer design and construction.

As mentioned earlier, these computers were designed for scientific and engineering calculations. However, so great has been the demand for data-processing equipment in commerce that the designers of these machines have turned to the application of their products to the business field. What are the important characteristics that a small computer should possess in order to be useful in data process application? That question cannot be answered in detail since the variety of applications in business is as large as in scientific and engineering calculations. Certain factors do, however, stand out.

The hallmark of data processing seems to be a processing of large amounts of data. In general, these data are subjected to very little arithmetical manipulation. The main problems seem to be those of sorting data according to preassigned specifications and extracting from large masses of data pertinent subsets of information which may be randomly distributed through a large file. The value of these computers in such problems, at least according to the advertisements, seems to be that they offer a possibility of automatizing these processes from the inception of data into the system to the preparation of reports, checks, and other printed material. If the system is rendered completely automatic, then most of its time will be spent on operations in which it does not excel. Thus, dollar for dollar, these computers do not sort and collate more cheaply than a group of electromechanical sorters and collators. Likewise, the terminal printing equipment on these computers, mostly Flexowriters, is too slow. The objection is not so much that the speed of the typewriter is low but that the computers to which they are attached must supervise the printing and thus be unable to do further calculations. This is not economical considering the relative values and costs of the computer and the typewriter. Here is one positive value of IBM terminal equipment as punch-card output is relatively fast and the typing can be done away from the machine on auxiliary equipment. Furthermore, the equipment, by means of plugboards, can be easily adapted to a variety of formats and alphanumerical printing. The question of storage of large amounts of information, which can be automatically called for by a computer, seems to be supplied by multiple magnetic-tape units.

Some business applications may require almost continuous printed output through the course of a working day. For such systems, the use of any of these machines, unless auxiliary punch-card output is available, seems to be unwarranted. However, there are applications, and quite a few of them where the printing is extensive though sporadic; for example, the preparation and printing of monthly financial statements by a firm. Where there is considerable automatic searching through large amounts of previously stored data for the computing of business statistics, the machines operate more efficiently. In such applications the elements of programming and planning that are natural to the computer's organizations exist.

One of the objections of business people to these machines is the difficulty in making certain that accurate data entered into the system are retained accurately. It is possible, and indeed mandatory, to devise a complete set of interlocking checks on the data at periodic intervals of system operations. This is a difficult programming and accounting problem but, nevertheless, one that can be accomplished and effectively carried out by the computer itself. As a matter of fact, it is probably possible to get more systematic and more frequent checks on the process of the business system with the computer than in a semiautomatic or nonautomatic system. Thus, even though the errors which will occur are the infrequent random kind which cause businessman to see the machine printing out checks for $1,000,000 instead of $7.84, the possibility of over-all checking to catch such errors in a short time is far greater than in any other kind of system. There is a subtle psychological difference in the computer between the way human beings and computers handle sums of digits which are represented by dollars and cents. The computer, unless otherwise instructed, regards all digits equally. The human being, for example, in adding a list of expenditures will be quite careful in checking the first digit, which is the most unimportant one, and the last digit which is the most important one, and most errors occur in between. However, one can always instruct the employees in the mailing room to stop all checks for $1,000,000.

These computers would be made more valuable to the business firm if they included devices as integral parts of the system for carrying out the combinatorial problems of sorting and collating data independently of the arithmetic operations of the computer; and had devices from which one could extract information, properly prepared by the computer, from magnetic tape and transferred to a typewriter independent of the computer. Often, examining some of the routine problems for which the use of these machines is proposed, one finds a serious imbalance between the time the machine is used for computation, that required for input and output, and the sorting of data. Thus, one finds figures of the order of 70 per cent of a day for input of data to the computer, 15 per cent of the day for computation, and 85 to 90 per cent of the day in output. Thus a $3,000 typewriter controls the functioning of a $100,000 machine over 80 per cent of a given day. Even under these circumstances it is possible that one of these computing systems will result in savings, if one averages the cost of the computer over several years. It is especially gratifying to received information that one of these manufacturers (Underwood) is planning the production of a computer at a price of approximately $200,000, which will more nearly satisfy the needs of commerce and industry.

A question which is often raised these

Table V. Programming Aids

	A	B	C	D	E	F	G
650	No	No	Yes	Yes	No	Yes	Yes
102d	No	No	Yes	No	No	Yes	No
30-203	Yes	Yes	Yes	No	Yes	Yes	No
Miniac							
Elecom *120*	Yes	Yes	Yes	No	No	Yes	No
Alwac	No	No	No	No	No	Yes	No
Circle	No	Yes	No	No	No	No	No
Monrobot	No	No	No	No	No	No	No
E 101	Yes	No	No	No	No	No	No

A. Automatic tally orders which are conditional upon completion of a count.
B. Transfer control orders which automatically retain an address indicating the source of the transfer.
C. Automatic normalization of numbers for floating-point and scale factoring.
D. Addition into high and low parts of the accumulator.
E. Automatic modification of addresses by *B* register.
F. Transfer control on +, 0, −.
G. Table look up.

days in business firms is, "Should a computer be purchased at this time?" If so, "What size computer should be obtained?" The answer to this, of course, depends upon the individual circumstances. However, there is a good deal to be said for the purchase of one of these low-priced computers at this time even though it may not be ideal and may even be recognized to have too small a capacity for the problems to be handled. Such a computer, providing it is not treated as a toy, will enable the staff of the firm to introduce the development of automatic process methods in the firm which may eventually find use in a larger system.

This paper has attempted to give a short survey, which must by necessity be conditional and qualitative. It is generally felt that the small computers represent a distinct blessing to research laboratories and businesses. It is because they are so complete and variable as tools of research, that one can complain about their shortcomings in handling any individual type of problem. Their applications will ever increase and, by virtue of their complete generality, become commonplace in industry and commerce. They represent the most outstanding example of an artificial device created by man to give himself a quantitative understanding of his environment.

Techniques for Increasing Storage Density of Magnetic Drum Systems

H. W. FULLER P. A. HUSMAN R. C. KELNER
ASSOCIATE MEMBER AIEE

THE problem of increasing the number of digits per linear inch of track length which can be reliably stored on and recovered from a magnetic medium is considered herein. The techniques to be described evolve from two well-known recording systems: the return-to-zero (RZ) and non-return-to-zero (NRZ) systems. The modifications concern both the nature of the recording and the method of reading back the information. The techniques can be used for tape storage as well as for drum storage.

Effect of Packing Density on Playback Waveform

It may be said in general that as pulse packing density is increased, the limited resolving power of the head and medium makes necessary a more perceptive reading system for successful recovery of the recorded information. It is only possible, however, to indicate accurately the limit of packing density with reference to a particular writing and reading system and head-medium configuration.

Fig. 1*a* represents the waveform of the recording current in the RZ system as a function of time for a particular sequence of digits. (The particular sequence of digits chosen has no significance except that it is a satisfactory one for illustrating the effect of increased packing density.) Fig. 1*b* represents the playback voltage from the reading head when the digits are packed at a relatively low density, say *m* bits per inch.

Figs. 1*c* and 1*d* illustrate the changes in the playback waveform with successive increases in relative packing density, 2*m* and 4*m* bits per inch, respectively. The deterioration of the waveform with increased packing density may be attributed principally to the loss caused by the spacing between the head and the medium and the loss associated with a non-zero gap length. These losses increase with decreasing wavelength.

For the preliminary discussion a frequently used reading system is considered wherein the information waveform is amplified and limited very close to, and on both sides of, the center line. The limited waveform is then sampled by clock pulses derived from a clock track and correctly phased with respect to the information waveform.

Fig. 2*a* shows the sampling clock pulses. Fig. 2*c* is the same waveform as that of Fig. 1*c* after the amplifying and symmetrical limiting process. Here the sampling pulse has occurred during the second half of the digit waveform and it is apparent that the written digit sequence is read correctly.

At the pulse packing density implied by Fig. 2*c*, the first half of the digit waveform can be sampled through the use of the sampling pulses of Fig. 2*b*, obtaining the same information in an inverted form. It may be noted that using both phases of sampling pulse allows the instrumentation of a valuable and inexpensive bit-by-bit checking feature. The first half of the digit waveform gates the first sampling pulse to set a flip-flop. The second half of the digit waveform and the appropriate side of the trigger pair are applied to a double input gate to pass the second sampling pulse as an error indication in the event that the polarity of the digit waveform has not reversed in the time between the two sampling pulses.

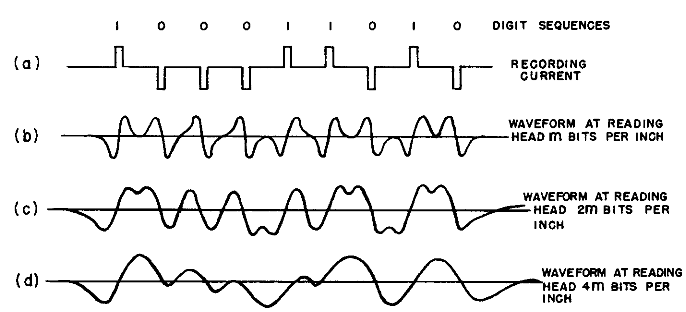


Fig. 1. RZ recording current, and waveforms at reading head for three different relative packing densities

H. W. FULLER, P. A. HUSMAN, and R. C. KELNER are with the Laboratory for Electronics, Inc., Boston, Mass.

Fig. 2*d* represents the waveform of Fig. 1*d* after the amplifying and symmetrical limiting process. It is apparent that at several points the time available for obtaining a correct sample has been decreased. If the sample is taken during the second half of the digit waveform, the samples obtained at points *1* and *3* of Fig. 2*d* may be of the wrong sense. If sampling is performed in the first half of the digit waveform, the samples at point *2* of Fig. 2*d* may be incorrect.

It is clear that in a noisefree system of this type an error is made when the sampled half of the waveform begins with a positive slope and fails to rise to the center line or the sampled half that begins with a negative slope fails to descend to the center line. In the presence of noise, errors may occur before the packing density limit indicated is reached.

Externally Clocked Cancellation Method of Reading

The trouble with the waveform of Fig. 1*d* can be thought of as the insufficient amplitude of the fluctuations of roughly bit frequency in the region of a sequence of "zeroes" and "ones," compared with the amplitude of the fluctuation in a region of a change from "zero" to "one," or "one" to "zero." Superimposed on the small amplitude fluctuations, additionally, is a slow monotonic drift of the waveform between two large amplitude peaks.

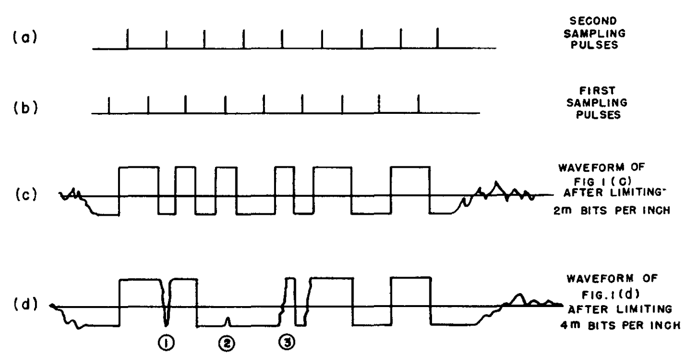


Fig. 2. Sampling pulses, and symmetrically limited reading head waveforms (using RZ recording) at two different relative packing densities

As a result, the waveform is frequently not of the correct polarity to be successfully sampled by a periodic sampling pulse.

One method of enhancing the small amplitude fluctuations and reducing the effect of the slow drift is by differentiating the waveform obtained from the reading head. This operation, however, reduces the signal-to-noise ratio. The "cancellation" method to be described eliminates the two waveform deficiencies mentioned without decreasing the signal-to-noise ratio, and exhibits an advantageous phase characteristic.

Fig. 3. Cancellation method of reading RZ recorded information, at a relative packing density of 4m

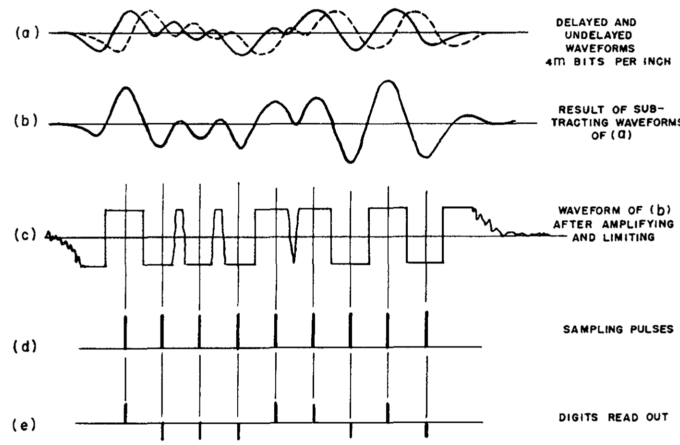


Fig. 3 shows the operation that constitutes the cancellation method. In Fig. 3*a* the solid line is the same waveform that appeared in Fig. 1*d*, at a relative packing density of 4*m*. The dotted line is the same waveform delayed by one half of a bit period. Fig. 3*b* is the resulting cancelled waveform after subtracting the dotted-line waveform of Fig. 3*a* from the solid-line waveform.

Fig. 3*c* shows the cancelled waveform after amplifying and symmetrical limiting. It is seen that the sampling time in each bit cell is at least one half of a bit period, and that a periodic sampling pulse phase can be found, as shown in Fig. 3*d*, that falls very nearly at the center of each sampling time. Fig. 3*e* schematically shows the correctly read pattern; separate "zero" and "one" lines are actually used.

Fig. 4 is a block diagram of the circuit used for the externally clocked cancellation reading system. The voltage e_1 corresponds to the solid line of Fig. 3*a* and the voltage e_2 corresponds to the dotted line of the same figure. The voltage K (e_1-e_2) corresponds to the waveform of Fig. 3*b* increased in amplitude by a gain factor K. The inverse voltage K (e_2-e_1) is obtained from the other output of the differential amplifier. The sampling pulses entering the coincidence gates occur in the time relationship indicated in Fig. 3. The sampling pulses gated through to the "zero" or to the "one" line indicate the digit read.

Fig. 5 is an oscilloscope picture of some of the waveforms that occur in the externally clocked cancellation method. The relative packing density in this case is 6*m*, for comparison with the waveforms drawn in previous figures. The recording and reading was done with the head in contact with the medium, and at a packing density of 880 bits per inch. The first waveform is the signal as it appears at the reading head. The second is the cancelled waveform. The third is the cancelled waveform after amplifying and limiting. The fourth is the output of the "one" line. The blanking pulse in all cases is the sampling pulse, and an idea of the sampling time tolerance can be had by observing the position of the blanking pulse on the third waveform.

It will be noticed in Fig. 5 that the sampling pulses occur at the very peaks of the cancelled waveform. This characteristic is very favorable to the phase tolerance of the sampling pulse, and arises due to the phase characteristic of the cancellation circuit. The phase characteristic is such that the phase shift at the bit frequency is zero, and at one half the bit frequency is $-\pi/4$ radians. The cancellation circuit has an amplitude characteristic that is the shape of a rectified sine wave with the first peak occurring at the bit frequency.

The externally clocked, cancellation method of reading RZ recorded information forces some restrictions to be placed on the organization of the storage system in which it is used. At the high density illustrated in Fig. 5, the effect of the contents of one bit cell on neighboring cells implies that some space must be left be-

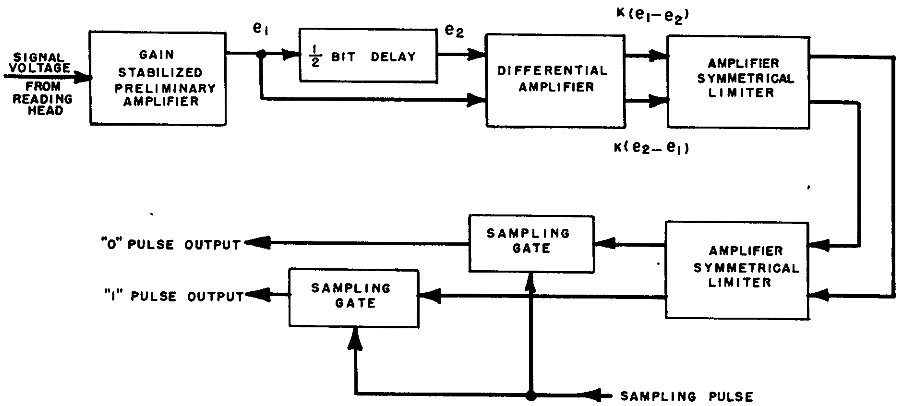


Fig. 4. Block diagram of externally clocked, cancellation reading system

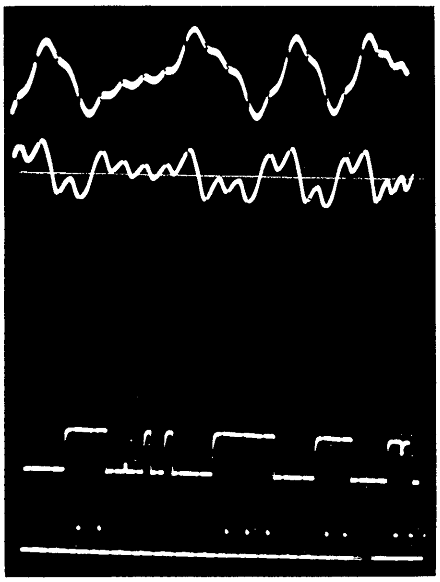


Fig. 5. Waveforms occurring in externally clocked, cancellation reading system. Relative packing density of 6m, absolute packing density of 880 bits per inch with head in contact with medium

tween groups of bits that are to be individually recorded. This further implies that the method is most efficient for serial storage by bits, or at least by characters. The limit of packing density by the use of this method has clearly not been reached in the example of Fig. 5. At still higher densities the cancelled waveform has the property that the fluctuations at bit frequency in a region of successive "ones" or "zeroes" grows progressively smaller in amplitude relative to the amplitude of the fluctuation characterizing an alternate "zero" and "one" region. As long as the maximum number of successive "ones" or "zeroes" is suitably restricted, however, the cancelled waveform continues to display the correct polarity for sampling in all regions, and a full bit period of sampling time in fact results. The limit of packing density, therefore, finally becomes dependent in principle upon the maximum number of successive "ones" or "zeroes" allowed, and the level of noise that accompanies the signal.

Writing Methods

It is seen from the foregoing that RZ recording serves very well the externally clocked cancellation method of reading at high packing densities. Two other writing methods have been developed, for use with other high-density reading methods, that offer the following potential advantages compared with RZ recording. Recording done in one bit cell has less effect upon neighboring bit cells, as judged from the read-out waveform. The amplitude of the read-out signal in a region of all "ones" or "zeroes" is more nearly comparable with the amplitude of the read-out signal in the region of alternate "ones" and "zeroes." These first two characteristics combine to result in at least one center line crossing of the read-out waveform for each bit cell up to higher packing densities than is possible using RZ recording. Finally, both methods result in the erasure of previously written information by the recording of new information, even when the two successive recordings are asynchronously applied.

Fig. 6 shows the recording current waveforms appropriate to the given bit sequence for the basic recording methods, RZ and NRZ, and for the two other methods studied, modified NRZ and double-pulse RZ. The logical rules for the formation of the modified NRZ recording current are seen to be the following: The current is switched from a negative saturating current to a positive saturating current in the center of a bit cell occupied by a "one," and conversely for a "zero." In addition, an appropriate switch in current is made midway between cell centers when this is necessary in order to observe the rules governing the cell centers. It is seen that the method is similar to NRZ in that all regions of the medium are subjected to a saturating flux, but that unlike NRZ, at least one flux change occurs in each bit cell. The former assures erasure by asynchronous re-recording, and the latter makes possible the existence of at least one center-line crossing in each bit cell of the read-out voltage waveform.

The logical rules for the formation of the double-pulse RZ recording current are the following. A "zero" is recorded by a negative current pulse at the center of the cell, followed by a positive pulse one half of a bit time later, and conversely for a "one." It is seen that this current waveform is a rough approximation to the modified NRZ current waveform, and for sufficiently high packing densities, when the response to the frequency components at twice the bit frequency is small, the read-out voltage waveforms are in fact very similar for the two methods. The double-pulse RZ method has the advantage of a reduced-current duty cycle, while the modified NRZ method produces a more thorough erasure upon asynchronous re-recording.

In Fig. 7 an experimental comparison between RZ and double-pulse RZ record-

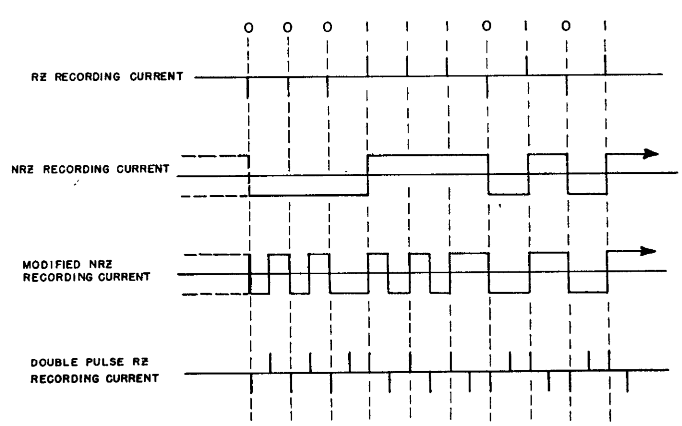


Fig. 6. Head-current waveforms for four methods of magnetic recording

ing is shown. Both recordings were made at 880 bits per inch with the head in contact with the medium; the same peak pulse current was used in each case. The advantages of double-pulse RZ recording as mentioned are apparent, though the bit patterns differ. The cancelled waveforms are also shown, as well as the waveforms resulting from a second cancellation operation. The existence of either one or two center-line crossings in each bit cell of the cancelled waveform for double-pulse RZ recording is seen.

Before passing to a description of a specific self-clocking system for reading the modified NRZ or double-pulse RZ recording, it is noted in general that several noise rejection features can be made to hold for these methods of recording as well as for the case of the externally clocked, cancellation method of reading using RZ recording. Indefinite limiting is applicable and cancellation rather than differentiation can be used. Thus the limited waveform is insensitive to input level, and only the phase-modulating effects of noise normally appear. Since the amplitude of the read-out waveform in the region of a sequence of zeroes or ones is greater in comparison with RZ recording for the modified NRZ and double-pulse RZ methods, however, the amplitude of large noise peaks must be greater (by about 10 db) to affect a normally limiting portion of the read-out waveform. This is important to the self-clocking system to be discussed, since in this system center-line crossings are directly used to extract the information, while before the limited waveform was sampled by a narrow pulse.

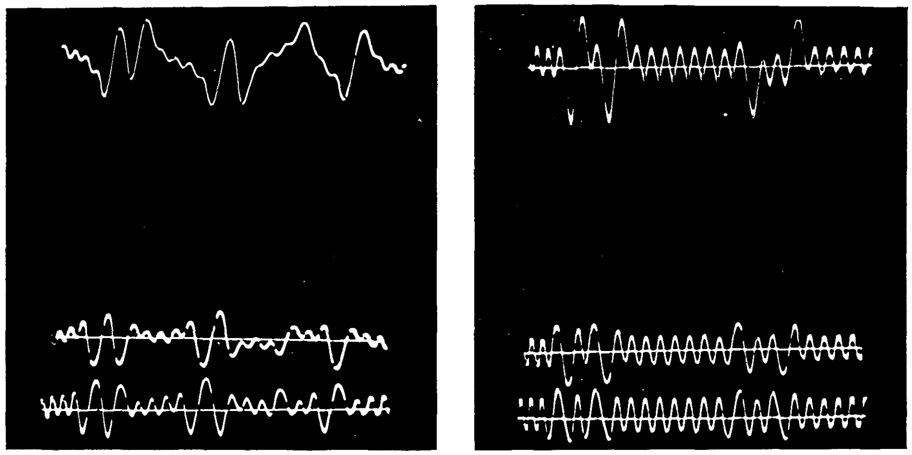


Fig. 7. Comparison between RZ (left) and double-pulse RZ (right) recording methods. Reading head waveforms, cancelled waveforms, and waveforms after a second cancellation operation. Relative packing density of 6m, absolute packing density of 880 bits per inch with head in contact with medium

Self-Clocked Cancellation Method of Reading

If recording and reading are done in contact with the medium or if a very small separation is maintained between head and medium, the numerical packing density that it is possible in principle to attain with the afore-described method becomes so large that another limiting difficulty appears. It becomes difficult to provide sufficient dimensional stability of the storage device to assure that the phase of the external clock track is maintained relative to the information tracks. The phase tolerance is important in the reading operation, and even more important in the RZ recording operation.

A method of recording and reading was developed for high-density storage that relaxes to an arbitrary extent the phase tolerance that can be allowed between information tracks and clock sources of the storage system. Either double-pulse RZ or modified NRZ recording is used for the property that erasure of previously recorded information occurs with asynchronous rerecording. Advantage is also taken in reading of the property that one or two center-line crossings occur, up to high densities, in each bit cell of the read waveform after cancellation.

Fig. 8 is a set of waveforms occurring in the self-clocked method of reading to be described. The recording was accomplished with the double-pulse RZ method; the density is 880 bits per inch with the head in contact with the medium. The first waveform is that occurring at the reading head, and is seen to express the digit sequence "...0010001110000..." The second waveform is the cancelled version of the first, and the third waveform is, in turn, the symmetrically limited version of the second. From the third waveform it is seen that one or two center-line crossovers occur in each bit cell, depending on the local bit pattern. The crossovers are used both for deriving the information that the wave represents, and for clocking the reading operation.

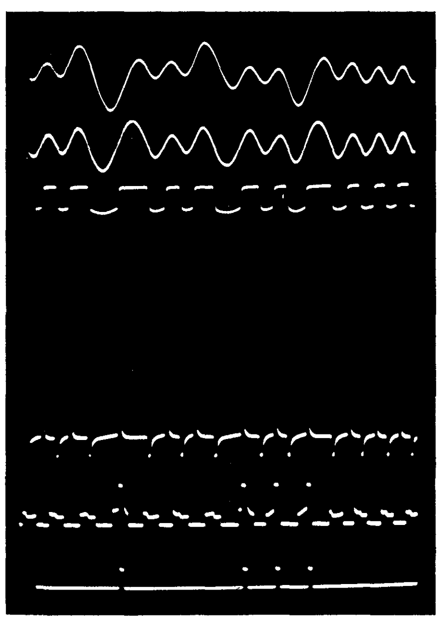


Fig. 8. Waveforms occurring in self-clocked, cancellation method of reading. Relative packing density of 6m, absolute packing density of 880 bits per inch

The mere existence of crossovers does not assure that the information can be extracted, but it will be noted that the intervals between crossovers at roughly the center of bit cells are closely one bit period, while the additional crossovers, when they occur, are closely midway between the others. The third waveform, in fact, resembles closely the modified NRZ recording current waveform that would be associated with this digit pattern.

The crossover occurring at the center of a bit cell will be called, for the convenience of reference, the "significant" crossover, while the additional crossover, when it occurs, will be called the "auxiliary" crossover. A method of reading the digit pattern is immediately apparent: The reading circuit is made to respond to the sense of the significant crossover for extracting the bit it represents (positive going for a "one," negative going for a "zero"), and the reading circuit is prevented from responding at all to auxiliary crossovers. This is accomplished by differentiating the limited waveform and its inverse to form "crossover pulses" on two lines. The crossover pulses occurring on the "one" line are shown as the fourth waveform of Fig. 8. The reading operation is started by allowing a significant

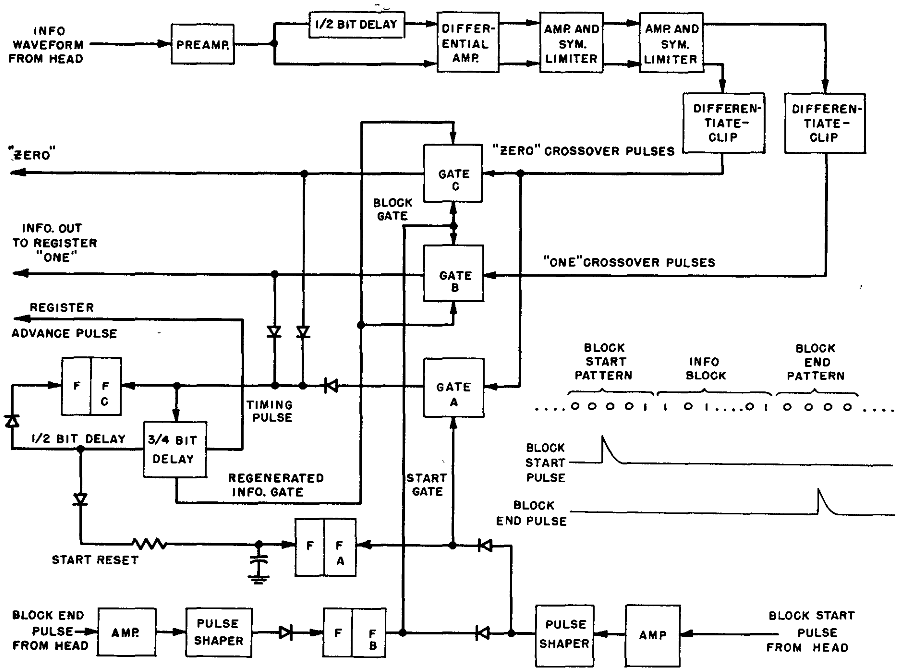


Fig. 9. Block diagram of test circuit for self-clocked cancellation method of reading

crossover pulse to form an "information gate" that is one half of a bit period in length. This gate is delayed by three quarters of a bit period so as to bracket the time of occurrence of the next significant crossover pulse, and to exclude the auxiliary crossover pulse, if it occurs. The gating voltage is applied to two gating circuits, one for the "zero" crossover pulse line and the other for the "one" crossover pulse line. The pulse outputs of these gates express the digit sequence, and the pulse outputs are also buffed together and delivered to the information-gate-forming circuit to perpetuate the information gate.

The fifth waveform of Fig. 8 is the gated output of the "one" crossover pulse line, and the sixth waveform is this same line after a clipping operation. It is seen that successful self-clocked reading by this method depends only on the time interval, with a generous tolerance, between two successive crossovers, and not upon the accurate phase of the information waveform with respect to an external clock.

Fig. 9 is a block diagram of a circuit for testing the self-clocked reading method. Gates B and C are the gate circuits mentioned. A significant crossover pulse passed by either of these gates is seen to appear at the output lines, as well as to operate flip-flop C. The one-half bit period tap on the three-fourths bit period delay line resets flip-flop C to form the one-half bit period information gate. The information gate is conducted from the delay line output to gates B and C to pass the next significant crossover pulse.

The test circuit shows one way in which the reading of a block of information may be started on a significant crossover pulse. A block start pulse is made available from an auxiliary track to indicate the approximate region of the beginning of an information block. An introductory block start pattern consisting of a number of "zeroes" followed by a "one" is recorded along with the information block. The introductory "zeroes" are made to bracket the time of the block start pulse. In reading, therefore, it is known that only crossover pulses occurring on the "zero" line are significant. The block start pulse is accordingly made to operate flip-flop A to form a start gate which is conducted to auxiliary gate A. The first full amplitude crossover pulse that appears on the "zero" line is passed to form the first information gate. Flip-flop A is reset by the first information gate as shown. The integrating circuit in the start reset line assures that flip-flop A is reset only after an information gate has been generated. The block start pulse also operates flip-flop B to form a block gate for gates B and C. Flip-flop B is reset by a block end pulse (which may be the next block start pulse). Not shown in Fig. 9 is a means for sensing the terminating "one" of the block start pattern; occurrence of the terminating "one" informs the reading circuit that the information block will begin with the next bit.

The test circuit is seen to rely on an external timing track for block start pulses. The tolerance in the phase of the two tracks can be made arbitrarily large, however, by increasing the number of bits in the block start pattern. In the event that a block start track is used, however, it is seen that the self-clocked method is most efficient when the information blocks are relatively long, and when completely serial storage is used.

The block start pulse track can be avoided if a block start pattern is used that never occurs within an information block. In this case it is necessary to maintain a small number of blocks spaced at intervals on an information track that are never rewritten unless the entire track is rewritten. These stationary blocks furnish track references that prevent rewritten blocks from eventually overlapping.

The self-clocked method allows the use of a bit-by-bit check of the reading operation. This check is made by providing two additional gates similar to gates B and C of Fig. 9. The additional gates are supplied with an inverted version of the information gate. An auxiliary crossover pulse, if it occurs, is thus passed by these gates, and its presence or absence, as well as the line on which it occurs, can be compared with the two digits read immediately before and after it.

The self-clocked method uses an external oscillator or timing track for forming the recording pulses. This clock source is ordinarily the one used by the rest of the machine. No phase relation-

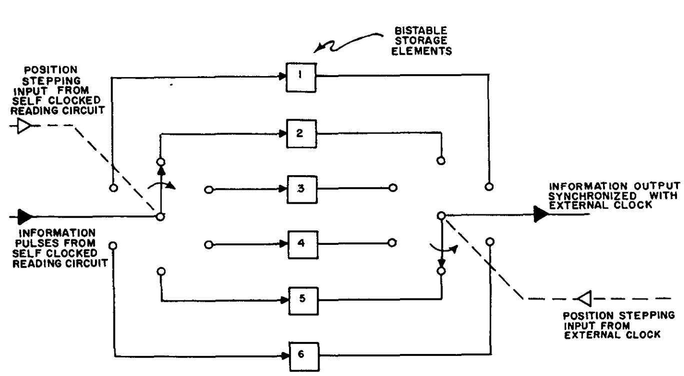


Fig. 10. Block diagram of synchronizer for use with self-clocked cancellation method of reading

ship exists between the external clock pulses and the information tracks, and even the repetition frequency of the clock pulses may fluctuate with respect to the average repetition frequency of the stored information as long as the block start pattern is made appropriately long. The absence of frequency and phase coherence between the read information and the clock requires that the information be retimed for use by other elements of the machine. This can be accomplished by reading the information from storage into a block-long register under control of advancing pulses derived from the self-clocked reading circuit (see Fig. 9), and by reading out the information from the register under control of external clock pulses.

An alternate method of retiming the self-clocked information is by the use of a synchronizer, the block diagram of which is shown in Fig. 10. When information begins emerging from the reading circuit both rotary switches are connected to the bistable storage element labeled *1*. After the first bit of information is inserted in *1*, the left-hand "input" switch is stepped to position *2* by a pulse derived from the delay line of Fig. 9. The second bit of information is inserted in *2*. The input switch finally reaches position *4*, and the first external clock pulse that occurs thereafter reads the information out of element *1* to which the right-hand "output" switch is still connected. An in-between phase of the external clock pulse then advances the output switch to position *2* in preparation for the next external clock pulse. Retimed reading of the information thus continues, with the position of the output switch always lagging the position of the input switch by from one to five steps. The number of bistable storage elements required in the synchronizer, shown as six in the example, is determined by the length of the information block being synchronized, and the difference in repetition frequency that is allowed between the self-clocked information output and the external clock.

Discussion

Kent Enslein (University of Rochester): Is there a mathematical basis for the phase-shift method?

P. A. Husman: I am not sure what you mean by the phase-shift method. Are you referring to what I call the cancellation method? Is that correct?

Kent Enslein: Yes.

P. A. Husman: Well, it is fairly simple and direct mathematically, the results of doing this cancellation, and it is very similar to performing a differentiation to the waveform in its result except that the absolute phase shift is different and that the signal-to-noise ratio is improved because of the frequency-amplitude characteristic. This is by no means the only way of obtaining this, but it seems to be a very convenient way of doing it.

G. H. Myers (Bell Telephone Laboratories): You mentioned the reading head touching the drum. Does this contact limit the rotation speed of the drum?

P. A. Husman: Was the question, "Does it limit the peripheral speed of the drum?"

H. T. Larson (Chairman): The rotation speed of the drum.

P. A. Husman: Yes, it does limit the rotational speed. The in-contact kind of drum is generally used in applications where you require relatively modest access times.

G. C. Randa (International Business Machines Corporation): What is the linear speed of the recording medium being used and must in-contact heads be used?

P. A. Husman: These particular photographs were taken with a medium that was traveling at 50 inches per second and the head which was used was not specifically made for in-contact recording. The head had a half-mil gap, 6-mil laminations.

M. Every (International Business Machines Corporation): What is the linear velocity of the recording medium, and also the write current amplitude and the reading voltage amplitude?

P. A. Husman: I just gave the particular linear velocity at which these pictures were taken, 50 inches per second. We have also done some work at higher and lower speeds, but at 50 inches per second as I recall the playback amplitude was in the order of 10 millivolts peak to peak with an 8-millihenry head being used, and the writing current, to the best of my memory, was about 20 milliamperes.

H. T. Larson: Mr. Every's question continues to ask what number of possible parallel channels for skew less than half the time between adjacent bits can be attained.

P. A. Husman: I am sorry, I didn't hear the question clearly.

H. T. Larson: If you were to organize a system that had a number of parallel channels for encoding one character, for example, the problem of skew begins to be important in counting one digit at the wrong time. He would like to know how many you think you could put alongside of each other for a skew equal to less than half the time between adjacent bits.

P. A. Husman: Well, it is hard to give a general answer to that question because that is one of the things that depends on the dimensional stability of the recording device, so if you can make that stable enough you can eliminate that as a factor. Otherwise, it is the factor which determines how many channels you can put in parallel and work with the skew that exists.

However, in general I do not see that the skew on a drum system is a problem in this case. It certainly would be on a tape system. But the purpose of developing a self-clocking system was to be able to read out individual channels without having to depend on the phase of those channels with respect to the clock channel.

I don't know if that answers your question adequately or not.

M. Every: That's fine.

M. B. Adams (Stanford Research Institute): What core gap is used in the recording or reading head for operation at 800 + pulses per inch, and what bit rate is used?

P. A. Husman: These particular photographs were taken with a half-mil gap. The velocity of the medium was 50 inches per second and the bit rate was 880 pulses per inch, which comes out to something around 40 kc per second. That would be improved somewhat by using a smaller gap on the head. Obviously the losses would be less if you used a shorter gap.

B. Hasbrouck (Westinghouse AGT): Are you aware of the techniques used on the Elecom *120?* Also, what have you done on noncontact heads in terms of bits per inch?

P. A. Husman: No, I am not aware of what is done in the Elecom *120*. In regard to noncontact work, we have done very little of it but we can say roughly that the number of bits per inch which you can pack is roughly again proportionate to the spacing of the head from the medium.

When we speak of "in-contact" work, we are not actually in contact magnetically even though there is a kind of physical contact there, so if we determine what the effective spacing is when we say we are in contact with the medium, then we can determine roughly what kind of packing density we can obtain at various spacing of the head from the medium.

G. I. Williams (Remington Rand, Inc.): In addition to the packing density that you attained in bits per inch for noncontact heads, is any difference in your attainments in RZ and NRZ recording?

P. A. Husman: If you are merely concerned with the difference between the NRZ and the RZ recording, the effect is quite different at low density than it is at high density. At high densities if you use an external clocked cancellation system the densities will be high enough to get very similar results by using NRZ recording current to that which you get with RZ recording current.

In general I see no particular advantage to using the NRZ recording current. It is a little more difficult to instrument this current properly.

H. T. Larson: Can you attach any numbers to the densities you have reached in RZ and NRZ out-of-contact work?

P. A. Husman: No, I cannot give any numbers on those.

L. Casciato (KCS Data Control, Toronto): What are the highest storage densities that have been achieved by these methods?

P. A. Husman: Well, the highest density which we have obtained in contact, and of which we are assured of its reliability, is about 880 pulses an inch. We feel that we can go somewhat higher than this with the self-clocked system, but not a great deal higher.

With the externally clocked cancellation method we could go quite a bit higher until we reached the limit imposed by drum stability, which I mentioned before.

A Self-Checking High-Speed Printer

EARL MASTERSON · ABRAHAM PRESSMAN

As soon as large-scale digital computers made large amounts of data an easy thing to handle, considerable talk arose about the need for high-speed printers. Of the two general classifications of computer problems, that is scientific and business, the former does not usually make anywhere near the demands on printing requirements that the latter does. Moreover, the appearance of the printed matter for scientific purposes does not need to be of nearly so high a quality as required by business problems.

Design Requirements

Generally, the tendency has been to overshoot on the speed requirement at the cost of other important needs. One of the prime requirements of the business field is that of extreme accuracy. One has only to consider that a high-speed printer may print many thousands of checks in a week. A second requirement, very high on the list, is the need for good quality in the appearance of the printing. In many businesses the printed form is the only visible contact between company and consumer. Because of the large volume of printing and consequently the cost of it, a printer, for this purpose, should definitely use standard paper. The cost of even the cheapest paper that is satisfactory for such an application may run into several thousands of dollars a year and it is therefore very doubtful that any system could be considered practical which uses coated or other specially treated papers.

Another factor in the design of a high-speed printer which can help to reduce paper cost is to reduce the type size and consequently the form size. It was believed that type not larger than ten to the inch should be the design aim. The paper-feeding system should be flexible enough to handle all presently used forms; this requirement extends to size, quality of paper, and number of carbons.

Still another important requirement of a high-speed printer, if it is to be used in conjunction with a large-scale electronic computer, is that it should not attempt to compute; it should do only printing. However, a saving in over-all operating cost can be shown if the printer contains a number of automatic editing features. Editing in the computer consumes time and requires rerunning of tapes. Since the computer is a more expensive device, it can be shown that, whenever possible, these features should be included in the design of the printer. The features which help eliminate the editing reruns to the greatest possible extent appear to be achieved best by including a plugboard in the printer in order to permit complete format control of the printed page. With such a plugboard it should be possible to rearrange the order of the information and to leave unplugged information which does not need to be printed. In addition, automatic zero suppression and dual printing saves computer time, and multiline operation saves not only computer time but also tape. In regard to the over-all effective speed of the printer, it can be seen that, no matter what the real speed of the printer is, the effective speed can be increased by providing a flexible control of fast feeding so that the printer quickly skips over areas where no printing is required and stops in the proper position for the next printing to take place.

Specifications

The complete printer is shown in Fig. 1. It is housed in four separate cabinets. The magnetic tape is read by a tape-reading unit (a modified Uniservo) which is housed in one cabinet. The logical control circuits and power supplies are housed in a second cabinet; the memory, plugboards, and actuator driving circuits are in a third cabinet; and the printer mechanism with operator control panel is in the fourth cabinet.

The printer can feed any standard sprocket-hole paper ranging in width from 4 to 27 inches and from 1 to 5 parts. The standard line is 130 columns wide with 10-to-the-inch spacing and there are 51 different characters available in each column.

The paper feed is under the control of a multichannel paper loop which can be programmed to any format up to 22 inches in length. The plugboards give complete format control including zero suppression, dual printing, line spacing, and multiline operation. Self-checking systems are incorporated which make the printing of a wrong character without automatic detection practically impossible.

History

The basic plan of the high-speed printer was adopted after some experimental work on printing mechanisms and after a study of possible memory systems. The printing mechanism finally chosen as most likely to meet the required specifications is the type generally known as printing on the fly. In this method a continuously rotating-type wheel shaft is apposed by individual hammers so placed as to drive the paper against the typewheel at the required instant in a precise and rapid operation.

The choice of memory was a very difficult one. The amount of memory required is too small to justify using basically large-type memories, and on the

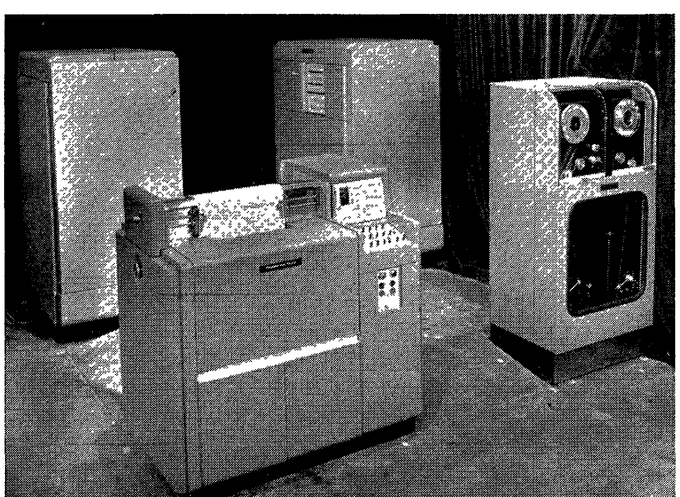


Fig. 1. A general view of the four cabinets which make up the new Remington Rand high-speed printer

EARL MASTERSON is with the Eckert-Mauchly Division, Remington Rand, Inc., Philadelphia, Pa.; ABRAHAM PRESSMAN is with Pressman Associates, Philadelphia, Pa.

other hand it is too large to consider using such straightforward things as flip-flops. After a considerable study of the logical requirements, particularly in regard to the need for complete format control by means of a plugboard, a static-type memory was found far more desirable. Although many memories were in interesting development stages, some experimental tests indicated that perhaps the most conservative design would be to use low-cost cold-cathode gas tubes.

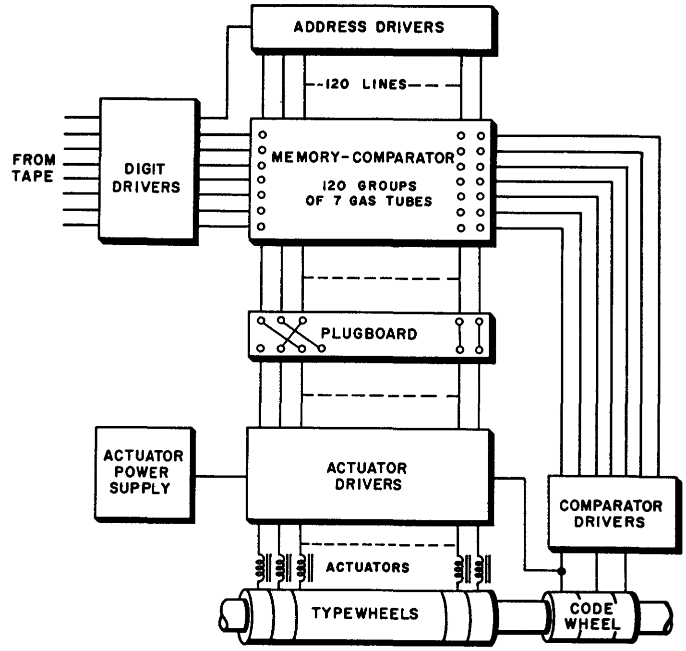
Logical Diagram

Fig. 2 is a greatly simplified block diagram of the high-speed printer. The memory consists of 120 groups of 7 gas tubes each. In operation, the magnetic tape is started and 120 characters are loaded into the memory. The check pulse is stored as well as the information bits. The eighth channel from the tape, which is a sprocket channel, is used to step the address drivers. After the last character is loaded into the memory, the tape is stopped, and the memory is probed by seven lines on which are generated character combinations corresponding to the successive printing positions of the typewheels. Whenever identities occur, a signal is sent to the appropriate actuator driver by means of the plugboard. The timing channel of the code wheel gates with this pulse and produces a controlled actuator operation accurately printing the characters on the paper. The comparison and printing process continues for one complete revolution of the typewheel after which the memory is cleared and the magnetic tape is again started in preparation of printing for the next line. Although it is not shown in the block diagram, the paper is also advanced during the memory clearing and reloading time. The method of operation is completely asynchronous: the printing cycle can start at any character position; likewise, the tape-reading and paper-advance can be of variable length and not waste time waiting for the start of a new machine cycle.

The Memory

Early in the project it was decided to use a type *5823* cold-cathode gas tube for a memory element. Because this particular application was quite removed from the more usual uses of this tube many special tests were set up to study the needed characteristics and to determine a set of special acceptance test limits required for this application. Throughout this activity the manufacturer was highly

Fig. 2. A simplified logical diagram



co-operative, giving us all the advice and help available.

Fig. 3 shows the published breakdown characteristics for the four quadrants of operation of the *5823*-type tube. The bracketed regions of the shaded areas indicate the range of electrode direct voltages required to start a glow discharge between pairs of the various elements. In general, a glow discharge started between starter and cathode in any of the four quadrants immediately causes anode-cathode breakdown if the anode voltage is sufficiently high and if sufficient current is available for the starter-to-cathode glow. In our application we operate the tube across quadrants *I* and *II* but only cause breakdowns in quadrant *I*.

The read-in operation is shown in the simplified schematic of Fig. 4. The starter of each *5823* is driven from a resistance gate which is fed address and digit information. If the voltage on both address and digit lines is up, the starter is up at +132 volts, and at a plate potential of 150 volts the tube fires. If either address or digit line voltage alone is up, the junction of the gating resistors is at +55 volts which must be below the minimum voltage required at this anode voltage to start the cathode-anode glow of the most easily fired tube. If neither the address- nor the digit-line voltage is up, the tube is in the second quadrant at a starter voltage of −55 volts and an anode voltage of +150 volts and the tube does not fire. The tube breakdown data of Fig. 3 are d-c data; however, as is well known, the amplitude of a pulse required to fire a *5823* generally is greater than the d-c value and goes up as its width goes down.

There is a wide range of variation from tube to tube in the minimum required pulse width to fire a tube; the average pulse width varies a good deal from one production run to another. Since such a requirement is not of interest in the application for which the tube was designed, there is no manufacturing control over this characteristic. For any given tube the minimum required pulse width is a function of the light intensity on the cathode and how long since it was last extinguished. Not only must the rated tolerances and the aging characteristics be taken into account, but ambient illumination, which affects photoemission, and worst possible immediate history of a tube must be considered also in determining a firing-pulse width which produces reliable operation. As the tube data bulletins advise, the manufacturer should be consulted for ratings under pulse operating conditions other than 60-cycle alternating current for which ratings are given in data sheets.

In the final design the memory tubes will always enjoy a minimum pulse-width of 30 microseconds (μsec). None of the tubes initially installed in the equipment require more than 17 μsec to write-in under the worst conditions of all the variables. Aging data indicate a life expectancy of over 25,000 hours for the tubes. For either an address-line voltage or digit-line voltage up, the starter is at a maximum of +55 volts with all voltages, resistors, and diode-drops off-center in the worst direction. The most easily fired tubes, with anode voltage and pulse widths off in the direction to make them fire most easily, must stand +60 volts

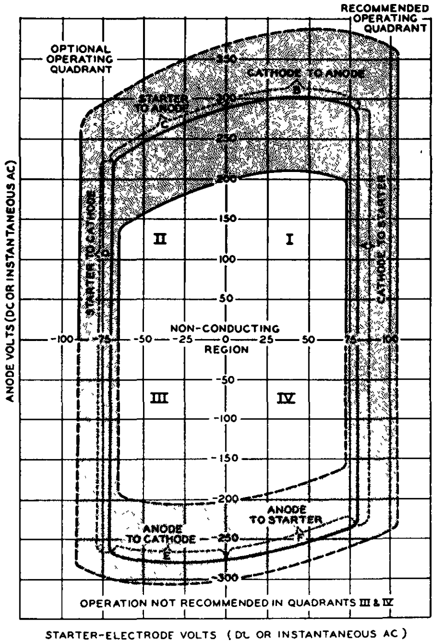


Fig. 3. Breakdown characteristics for type 5823 tube

Starter-electrode series resistance = 200,000 ohms. Ranges shown between inside and outside curves take into account maximum and minimum + and − voltage values for individual tubes and for changes during tube life. The values shown by dashed sections are approximations only

without firing. For both address- and digit-line voltages down, all voltage, resistor, and diode-drop tolerances off-center in the worst direction, the starter is at −55 volts. The most easily fired tube under the worst conditions of anode voltage must not fire at −60 volts.

The digit drivers are cathode followers with a regulated lower clamp on the cathode and a regulated upper clamp on the grids. The address generators are 120 double-coincidence diode-mixing gates. Selenium diodes are used in these gates because of the large inverse voltages to which they are subjected (200 volts) and the low inverse leakage requirements. These mixing gates mix the outputs of two small decoders which decode the first three and last four binary columns of a 7-stage binary counter which is stepped by the sprocket channel pulse from the magnetic tape. The small decoders are conventional germanium-diode matrices putting out 8 and 15 minor addresses, respectively. The amplifiers which drive the selenium decoder have regulated upper and lower clamps.

After the memory is loaded a print-out cycle is entered. A simplified schematic of the print circuits is shown in Fig. 5. As each new character on the typewheel comes into printing position there are 120 simultaneous comparisons of the information in each address with the code coming from the code wheel. The code delivered by the code wheel lines is the complement of the code for the character which is at the moment in the printing position. A printing signal is generated if column for column in each of the seven columns of a character, the voltages at V_A and V_B are complementary. If this is so, V_J is at some level between +85 volts and +107 volts. This leaves V_C at +85 volts and V_D at +71 volts and the primary of the output transformer is energized. If V_A and V_B are noncomplementary, for example, when both voltages are up, in which case V_J and therefore V_K and V_D are driven up, putting inverse voltage on $D1$, then the primary of the output transformer is not energized. If V_A and V_B are noncomplementary and both down, V_J goes down pulling the level of V_C below that of V_D and again the primary of the output trans-

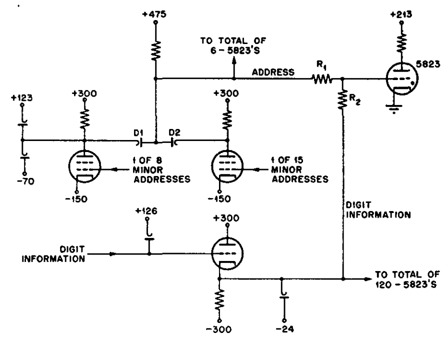


Fig. 4. Simplified schematic of read-in circuits

former is not energized. Points V_J of all the seven tubes of a given address are buffed together into points V_C and V_D. Since they are driving a transformer, the relative d-c level of V_C and V_D must be sampled or chopped at an a-c rate. This is done by keeping V_E at +103 volts so that V_D is always above V_C except during a comparison which is a 300-μsec interval bracketing the exact moment that the *2050*'s should fire. If column-for-column complementary voltages are found, the output transformer delivers a 300-μsec enabling pulse to the no. 1 grid of the *2050* which is then fired by a very accurately timed probing pulse on its no. 2 grid about 250 μsec inside of the enabling pulse. All the diodes shown in the comparator circuits are selenium, again because of the high inverse voltage and because of the low reverse leakage requirements.

Memory Checking System

In order to provide a checking system to ensure that each memory position is read-out, and also to ensure that one and only one pulse is received during a print cycle, an additional primary winding is added to the transformer to allow the memory to be probed to determine which memory locations have been filled with printing (as opposed to nonprinting) combinations. After the memory has been loaded and just before the print cycle starts, a probe pulse is applied to this winding on all 120 transformers. If any one of the group of seven tubes has been ionized during the read-in period, an output pulse is produced which ionizes a check tube which in turn alerts the actuator driver tube. When the true identity pulse is found during the print cycle and the actuator driver tube is fired, the check tube is deionized. If any spurious comparison pulses are received during the same print cycle, the check tube is again ionized. The automatic check is then performed by a detector which determines whether or not all check tubes are out. If any check tubes are on at the end of the print cycle, the printer is stopped and an error light is turned on. At the end of the print cycle the memory is cleared by

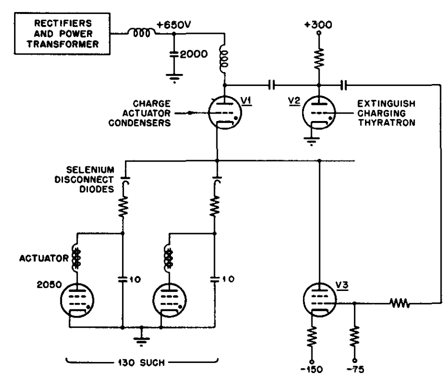


Fig. 5. Simplified schematic of the print circuits

switching the memory anode supply to +50 volts. This, together with holding the starter grids at +55 volts, proved enough to cause the gas tubes to deionize safely in 2 milliseconds.

Code Generation

The typewheels run at constant speed and are attached to a code wheel which controls other operating parts of the equipment including the print hammers according to the positions of the typewheels.

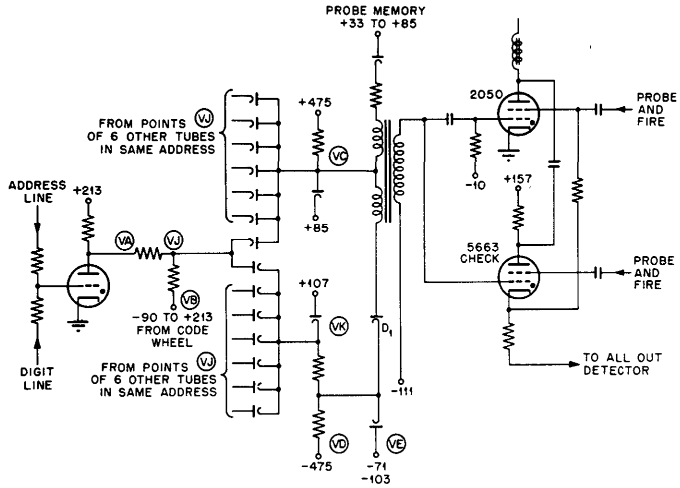


Fig. 6. Simplified schematic of the actuator power supply

Actuator Power Supply

The design of the actuator power supply is of interest because of the large currents handled. (A simplified schematic is shown in Fig. 6.) If every character has been printed in a line, it is then necessary during the memory-reloading time to recharge all 130 1-microfarad (μf) capacitors to the 650-volt operating level. The out-time or memory-loading time is approximately 20 milliseconds. To charge the full bank of capacitors in this time requires that the initial surge of current be some 25 amperes (amp). In order to prevent this surge from appearing back on the power line, a fairly conventional power supply was built with an extremely large bank of filter capacitors. The bank actually consists of 2,000 μf. This permits the bank of capacitors to be charged during the entire printing cycle and the major recharging surge is taken primarily by the bank. Individual current-limiting resistors and disconnect diodes are associated with each of the 130 capacitors. The charging cycle is initiated by triggering the grid of the charging thyratron V_1. The current builds up to 25 amp in approximately one-half millisecond and decays to approximately 1/2 amp in 9 milliseconds at which time all actuator capacitors are fully charged. At the start of the print cycle, V_2 is fired which extinguishes V_1 and disconnects all *2050*'s and their 1-μf capacitors from the $B+$ bus. V_2 is self-extinguishing due to the ringing of the choke and capacitor in its anode circuit. V_3 serves as a sink to absorb back leakage from the selenium disconnect diodes. Thus, capacitors which have not yet discharged leak into V_3 rather than into those which have already discharged where they might cause double firing of a *2050*.

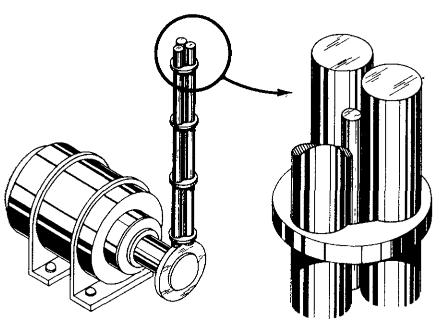


Fig. 7. A test setup employing three brass rods as guides for various weights of hammers dropped against a rotating motor-driven typewheel

Hammer Design

The general principle of printing on the fly is by no means new, however, it is believed that the development and design considerations required to produce a printer to meet the above specifications are new. The two main considerations concerning the quality of the printing in a printer of this type are the length of time that the hammer and the typewheel are in contact and the repeatability of the time required for the hammer to arrive at the typewheel. To establish the initial order of magnitude of the possible contact time between hammer and typewheel, tests were made by attaching a fine wire to a steel ball and dropping the ball on a hardened steel plate. The contact time was observed by using a long persistence tube in an oscilloscope and it was found that contact time is of the order of 25 μsec. It was observed that when dropping the same small steel ball on a piece of unsupported sheet metal, the contact time became many times longer. The investigations proceeded through a more realistic hammer shape and many tests were carried on using the equipment shown in Fig. 7. In this equipment, a hammer guide several feet long was constructed by properly spacing three brass rods so that a small hammer could be dropped from various heights in order to determine not only contact time but also required energy. The three vertical brass rods provided a guide which had very little friction and also prevented the hammer from acting as a piston. The space between the rods provided ample passage for the displaced air. By driving a typewheel with a variable-speed motor and by dropping the hammer a sufficient number of times it was possible to determine the optimum hammer weight and energy requirement to produce a sufficient number of carbon copies and to determine the effective contact time under dynamic conditions. It was generally concluded that 50,000 ergs in a hammer weighing 0.8 gm would produce a blow sufficient for an original and four carbon copies and would produce a contact time which would permit a wheel surface velocity of approximately 200

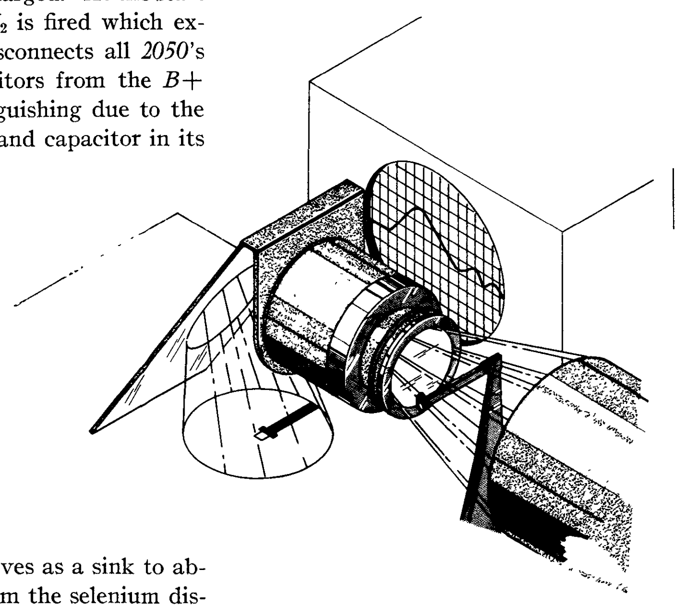


Fig. 8. A method for displaying the hammer or actuator armature displacement on an oscilloscope screen

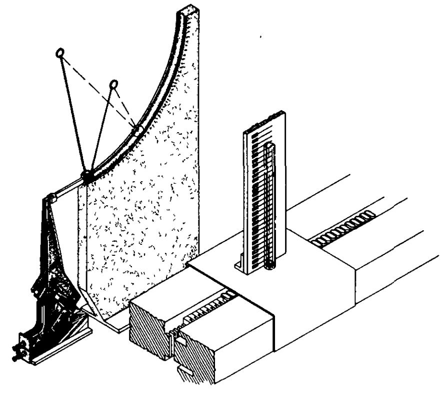


Fig. 9. A ballistic pendulum was used to measure the actuator-hammer efficiency

inches per second. Knowing these figures, it was then possible to start the design of an actuator which would produce the required energy.

Assuming that a hammer-driving system is going to have a certain percentage of jitter or drift in timing, it can be concluded, therefore, that the best design is one in which the over-all time of operation is reduced to a bare minimum. Every effort was made to make the parts light and stiff and to develop an electric system which could build up a magnetic field in a very short time. The coil has 1,100 turns of no. *41* wire and the inductance when the actuator is open is 85 millihenrys. The current can be built up to 1.2 amp in approximately 1/2 millisecond by discharging a 1-μf capacitor which has been charged to 650 volts by means of a *2050* gas thyratron tube.

Through a series of experimental steps, an actuator was designed in which the effective mass of the armature assembly was equal to the mass of the hammer. If the relation of the parts is such that the armature strikes the hammer just before the air gap closes, the energy stored in the armature should be totally transferred to the hammer and the armature should come to a dead stop. This could be called the pool ball system. However, in practice it was found that it was impossible to make the armature assembly stiff enough to represent a single mass. Consequently, only part of the energy was transferred, the remainder being used up in shock waves which presumably travel up and down the armature assembly. It was found that a more efficient transfer of energy could be accomplished by starting the cycle with the hammer and armature assembly in contact. The hammer is then thrown free after the armature air gap closes.

Various actuator designs and methods of operation were studied by various measuring systems. At one point, high-speed movies were used, but two other methods proved more useful because they permitted continuous observation. Fig. 8 shows a method of displaying the hammer or actuator armature displacement on an oscilloscope screen. A projection microscope is used to image a silhouette of the hammer or armature upon a rectangular translucent window. A photoelectric tube, so arranged as to average the transmitted light, will produce an output proportional to the displacement of the device under test. At a repetition rate of some 20 cycles per second (cps), a very good picture is obtained upon the screen.

Another very successful method of studying the operation of the print cycle is to trigger a variable delay flop at the same time the actuator driver tube is triggered. The output of the delay flop is used to operate a stroboscope. If the stroboscope is set up to illuminate the hammer and actuator assembly, and if the delay flop is manually increased in delay time while watching the operation, the complete cycle can be observed as though it were taken in slow motion.

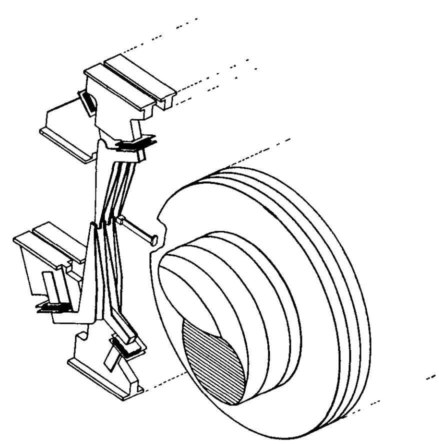


Fig. 10. A cross section of the actuator assembly showing how the actuators are arranged in an 8-way interlace

In the final design of the hammer and actuator, the portion of the cycle which accomplishes the printing is very simple. The hammer and actuator armature accelerate in unison while traveling approximately 50 mils; the air gap is then closed and the hammer flies free another 30 mils and collides with the typewheel. The return to rest, however, is not nearly as simple but it is sufficient to say that the hammer never strikes the typewheel again during the same cycle. There are several minor collisons between the hammer and armature before they both come to rest and are ready for the next cycle. Calibrating the measuring devices made possible accurate measurement of the time of operation for a printing cycle. In the final design, the hammer hits the typewheel approximately 1 millisecond after the triggering signal has been applied to the gas thyratron grid. It requires some 8 milliseconds for the hammer and armature to return to rest. This amount of return time is of no consequence since it requires approximately 20 milliseconds to clear and refill the memory.

Many tests were made to determine the most efficient actuator and hammer design. Fig. 9 shows a ballistic pendulum setup which was used to measure the actuator-hammer efficiency. Some of the first actuators were of a laminated construction which was riveted to a brass base. It was discovered that although various assemblies appeared to be the same mechanically, they gave radically different energy outputs. Even though the assemblies appeared to be riveted tightly, the mechanical energy was being dissipated by minute movements of the laminated assembly in the brass base. When it is realized that the mechanical wavefront of 1 cycle of operation is equivalent to a continuous frequency of 5,000 or 10,000 cps, the parts must be analyzed in a somewhat different light. The parts are actually experiencing frequencies in the upper audio range. Soon after this it was discovered that solid cores were as efficient as laminated and from there it was an easy step to go from a fabricated structure to a 1-piece cast actuator and base.

Fig. 10 is a cross section of the method of interlacing actuators. In order to meet the requirements of ten characters to the inch, it was necessary to make an 8-way interlace of the actuators; there are

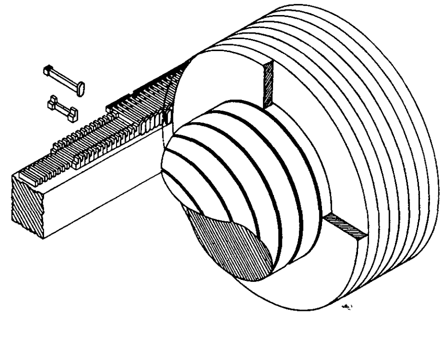


Fig. 11. The shaft in which the typewheels are stacked is grooved so that a snap-ring can hold each pair of typewheels together. Nylon inserts provide bearing surface for the hammers

only four types of actuators, the pattern being repeated both top and bottom. Because of the great number of parts in this assembly, a design was worked out to facilitate servicing which made it possible to remove the entire bank of actuators as a unit from the machine in less than 1 minute.

A system of automatic indexing and supporting insures that the unit will go back into the printer with a position accuracy of less than 0.001 inch. The bench adjustment of the actuators insures that there are at least 2 mils of engagement between the armature and hammer and that the air gap is adjusted for 25 mils. The individual actuator adjustments to correct for slight timing differences are controlled by varying the armature backstop. The lead time of an actuator can be reduced by moving the backstop forward which has a dual effect of shortening the hammer stroke and reducing the air gap.

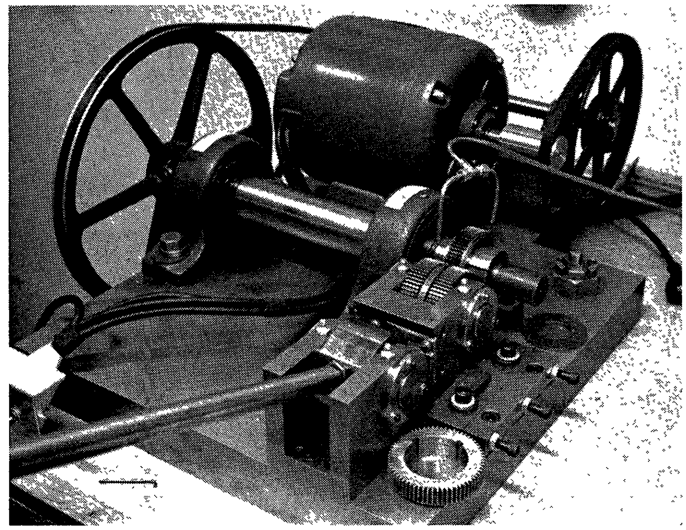


Fig. 12. An early model of a knurling machine for making the typewheels

Typewheel Construction

After a considerable number of tests were made using simple, single-wheel models, a model was built which had eight typewheels. It was found that although very satisfactory 1- or 2-part forms could be printed, a rather serious problem appeared when more carbon copies were required. When a thick pack of paper is struck from behind to print a certain character, it is very difficult to prevent the edges of the adjacent characters both above and below and to the right and to the left of the intended character from showing. Various solutions to the problem were tried but none was found to be satisfactory. As a result, the checkboard typewheel plan was adopted. In this system, the diameter of the typewheel is made twice as large and every other typewheel is staggered one space so that now each character is surrounded by four spaces instead of four other characters. This limits the effective printing speed of the printer but completely solves the adjacent-character printing problem. The principle of operation during the print cycle was then modified so that for each character the search is made in the memory with all odd columns alerted and then one character position later the search is made with all even columns alerted.

Fig. 11 shows the design solution to two problems which could have caused considerable trouble in accumulated tolerances. If the individual typewheels were merely stacked on a shaft, the thickness tolerance would have had to be held to a fraction of a tenth in order to keep the over-all accuracy in some 13 inches from being off more than a few thousandths. Instead, the shaft was accurately grooved, corresponding to the space between every second typewheel. In assembly, a snap-ring is located in a groove, a typewheel is pushed up against the snap-ring, it is followed by a spring washer and a second typewheel. This brings the last surface of the second typewheel in line to accommodate the second snap-ring. A third typewheel is pushed on, followed by another spring washer, and then the fourth typewheel and another snap-ring. This process is repeated until the typewheel shaft is full. The rotational positioning of the typewheel is locked by means of a long key.

A similar method was used to solve the problem of providing nylon guides for the individual hammers. In the original model, continuous steel combs were used to support and locate the hammers. Life tests, however, indicated that the repeated hammer strokes caused serious fretting corrosion. A solution to this problem was found by letting the hammers rest on nylon guides. The dimensional stability of nylon over 13 inches makes it impossible to use the same type of comb construction. The solution to this problem has resulted in locating individual nylon inserts in slots milled in a continuous steel bar.

To justify a printer of this design, it was believed that some effort should be given to cost reduction of the typewheels. When it is considered that a complete font of type is required at each column position, it is somewhat frightening to think that this represents 6,630 type faces in our machine. A typewheel knurling machine was designed which is shown in Fig. 12. In this machine, cold-rolled steel typewheel blanks are rolled at high pressure against a master typewheel. The wheels are case-hardened, plated, and assembled on a common shaft. Because of the general proportions it was decided to make the typewheel two columns wide to prevent warping during the hardening process.

Type Alignment

An investigation of what could be tolerated in the terms of vertical displacement of adjacent characters indicated that for good appearance the displacement should not amount to more than ± 5 mils. Converting this to time at the top wheel speed shows that the error cannot amount to more than ± 25 μsec. Since this tolerance must be split over many elements, it was necessary to consider carefully each in the respect to its contribution to the total. Some of the elements that can contribute to this error are as follows:

1. Angular spacing of the sprocket pulses on the code wheel.
2. Angular spacing of the character positions on the typewheels.
3. Angular keying of the typewheels.
4. Typewheel and code-wheel shaft run-out.
5. Electric-code-wheel pulse-detection and pulse-forming circuits.
6. Triggering of the actuator driver tube and operating and travel time of the actuators.

To keep the angular accuracy of the code wheel and typewheels good, a scheme was worked out in which the control was performed by a set of matched master gears. Six gears were mounted on a common mandrel and were indexed and precision-ground as a unit. After finishing, four of these gears were used to synchronize the rotations of the typewheel master and typewheel blank in the knurling machine which was previously described. The typewheel master which was used in this machine was originally engraved by using another of the master gears as an indexing wheel. The sixth master gear is used in another indexing device which is used to grind and lap the optical flats on the code wheel. With all of these precautions the angular error that is believed to exist between the code wheel and the typewheels is less than ±2 mils. The code-wheel optical system is shown in Fig. 13.

The desire to use not more than two bearings on a single shaft and to have the code wheels close to one of these bearings in order to minimize runout put the code wheels where their diameter had to be somewhat less than the typewheel diameter. This relatively small diameter is not conducive to good angular accuracy. However, it was made up by adopting the optical-arm principle. The effective diameter, because of this principle, is 24 inches. In the design, a projection lens images a lamp filament on a slit which is in front of a photoelectric tube. The light path, however, is first reflected from one of the facets on the code wheel and then again reflected by a common adjustable mirror. Whenever the speed of the machine is changed, it is necessary to change the lead time of the signals to the actuators. This shift is accomplished by tilting the common mirror. The relatively long optical arm produces a rather good rise time in a photoelectric cell and this is further improved by the zero-crossing detection circuits which are relatively insensitive to amplitude variations. It is believed that the electrical detection of the pulse and its formation into a triggering pulse for the actuator driver pulse accounts for another possible ±1 mils of misalignment. The repeatability of the actuator stroke is believed to account for another ±2 mils. Operating the machine at lower speeds will improve alignment in all cases except for angular errors in the typewheel and code wheel. In practice the original goal of ±5 mils is achieved at a machine speed of 400 lines per minute. The misalignment is generally somewhat over this at 600 lines per minute.

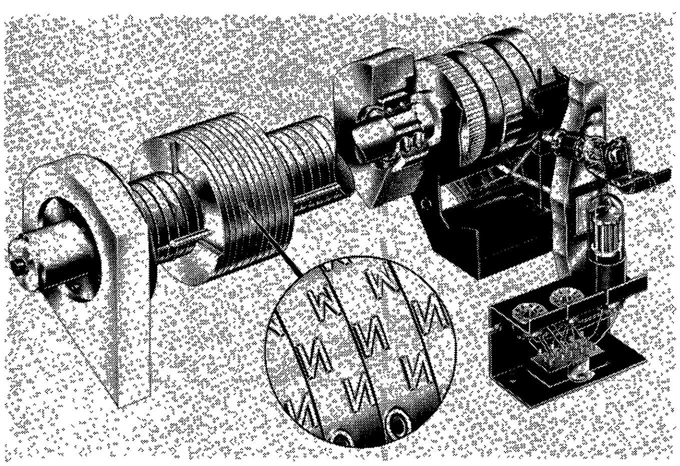


Fig. 13. A view of the code-wheel optical system

Paper Feed

The paper feed requirements are several times as difficult as one might expect by considering only the speed of the machine. It is true that the basic machine speed is some four to six times higher than has been accomplished previously but, in addition, this type of printer requires that the paper advance be confined to a small portion of the over-all machine cycle. In this type of printer, the paper is required to stand still while a complete revolution of the typewheels is performed and then the paper must be advanced very quickly to conserve over-all time. In other words, the requirements are not as easy as if type bars or typewheels were set up and printed a line at a time with a single stroke. The solution to the paper-feeding problem was found by adopting one of the Remington Rand high-speed magnetically operated clutches. This clutch was originally designed as a tape drive for the Uniservo machines.

The clutch output shaft is mechanically geared down four to one to drive a shaft containing a pair of conventional paper-feed tractors. There is no mechanical indexing of any type. The paper can be accelerated to a full speed of 20 inches per second in approximately 5 milliseconds. A complete single-line spacing can be accomplished in approximately 10 milliseconds. This time does not limit the machine speed since it requires approximately 20 milliseconds to clear the memory and read in a new blockette of information. The magnetic clutch, optical system, and paper loop control are shown in Fig. 14.

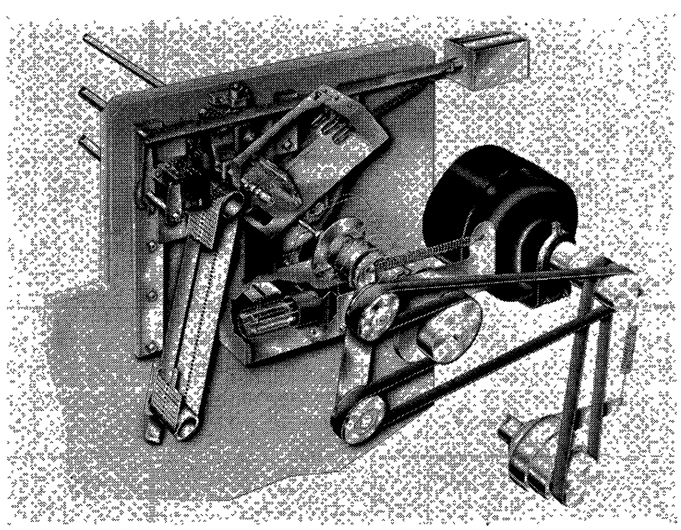


Fig. 14. The paper feed system operates optically and can be controlled by a punched-paper loop

Ribbon Feed

To prevent frequent changes of ribbon as might be required at this speed of operation, a large volume of ribbon is stored within the machine. In slower speed machines, it is satisfactory to transport a narrow ribbon in a direction transverse to the direction of paper feed. In this machine, it was decided to use a ribbon wide enough to span the entire printing area and to move it in the same direction that the paper is fed. The machine is therefore designed to handle a reel of ribbon 6 inches in diameter of any width up to and including the full 13 inches of printing area. The ribbon-drive power is taken directly from the typewheel shaft so that no matter what the machine speed

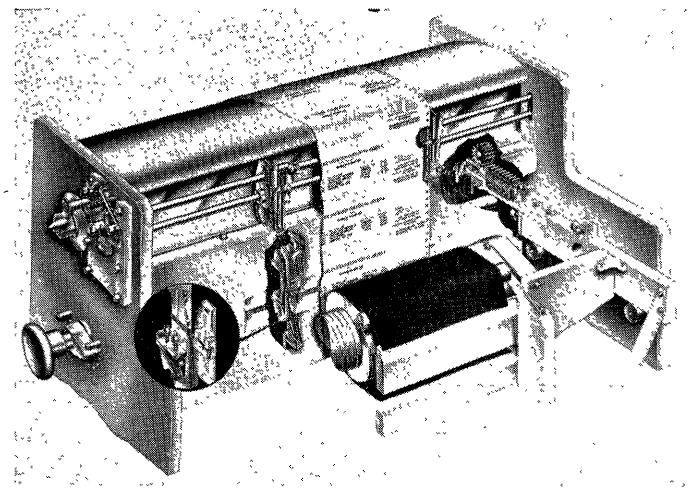
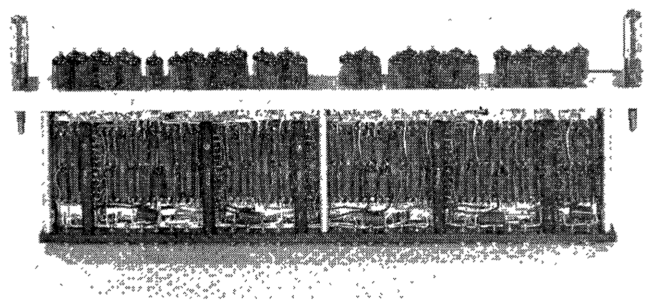


Fig. 15. A cutaway view showing the actuator assembly, ribbon reel, and the gear and rack assembly for opening the paper feed area

Fig. 16 (below). A typical memory chassis



is set for, the ribbon is propelled at the optimum rate. A gear-shift selector makes it possible to feed either a 1-shot carbon-paper with no overprinting or to drive a cloth-base ribbon at a slower speed permitting considerable overprinting. A third gear position permits a rapid winding of the ribbon if it is desired to remove the ribbon from the machine.

A simple clutch is provided to energize the ribbon-driving system only when actual printing is taking place. The take-up roll is driven on its surface which provides a constant ribbon velocity from start to finish of the roll. The back tension on the feed roll is controlled by a diameter sensing idler which in turn controls a friction brake. The ribbon path is designed so that, when the ribbon is reused, it is necessary to turn the roll end for end, which helps distribute the printing wear. The relative positions of the ribbon supply, the ribbon take-up, the typewheel shaft, and the paper feed are shown in Fig. 15. The typewheel shaft along with the ribbon drive assembly is on a movable carriage. The carriage assembly moves approximately 3 inches away from the actuator bank by means of the small drive motor controlled by a switch. The 3-inch opening permits the paper to be threaded easily through the machine and also permits the ribbon to be threaded over the typewheels. The motor-driven carriage returns to the printing position with a position accuracy of less than 1 mil. Combination limit- and safety-switches prevent the possibility of injury to the operator. The spacing between the actuator hammer assembly and the typewheel shaft can be controlled over a range of approximately 40 mils to accommodate various thicknesses of forms.

Physical Construction of Electronic Assemblies

Much of the physical construction of the electronic printer assemblies is merely an adaptation of UNIVAC building-blocks. The circuits are constructed on standard UNIVAC plug-in chassis and are mounted in standard sections which can accommodate 12 chassis; a total of six sections has been used. One of the memory chassis is shown in Fig. 18. There are 24 identical chassis in the memory which fills two sections to make up the entire memory of 840 cold-cathode gas tubes. The diode-comparator stacks can be seen adjacent to the internal board. The actuator-driver and check tubes are mounted in similar chassis and require the space of another section.

The three remaining sections hold chassis to perform the various logical control functions and to operate the various switched power supplies and driver tubes. The main power supplies are mounted on standard UNIVAC trays and they are supported in a suitable framework in one of the electronic units. The printer cabinet contains not only the printer mechanism but also the large actuator capacitor charging power supply as well as the paper-feed circuitry.

The Uniservo has been modified from the standard UNIVAC Uniservo to provide a high-speed magnetic-clutch driving system and has other circuit additions to make it independently operated.

All four units are provided with casters and are interconnected with flexible cables which can be run under a false floor or can be dropped into ductwork if provided. The two electronic cabinets are cooled by a cold-water heat-exchanger system. A blower is so arranged as to draw air down through the center of the cabinet, out through the dual heat-exchanger coils, and up through the electronic sections, returning back again to the center of the machine. This method of cooling, as in the UNIVAC, has the desired feature of providing cooling without introducing dirt from the room air. The entire printer contains some 1,700 tubes, 840 of which are the cold-cathode gas tubes. An adjustable autotransformer makes it possible to take any 60-cycle power from 190 to 250 volts. The power input is approximately 16 kva.

Discussion

J. L. Bower (North American Aviation): Will you briefly describe the operation of the photoelectric checking system? What failure data have you on the *5823* tube memory?

Earl Masterson: The photoelectric system. By that do you mean the typewheel system that generates the code combinations?

J. L. Bower: That is what I had in mind.

Earl Masterson: On the end of the typewheel shaft is mounted a drum which contains three channels. There are flats which are ground and lapped on the drum. There is one flat for each character position on the typewheel shaft. Light is reflected from these flats into a mirror, actually, and then into a photoelectric cell which is behind a small slit. This produces a fairly sharp pulse of light into the photoelectric cell which consequently gives a sharp signal.

There are three channels. There is one which steps a 6-stage counter, another one supplies the seventh bit to make the 7-bit code, and the third has a single flat to clear the 6-stage counter. The first channel has a flat for each character position and it gates with the signal from the memory to trigger accurately the firing of the actuator thyratron.

The mirror that I previously mentioned

can be tilted to accommodate the differing lead times you need when you change the speed of the machine.

Mr. Pressman will discuss the *5823*.

Abraham Pressman: We ran life tests on *5823*'s to determine at what point in life the tube will not ionize when we feed it a 30-microsecond (μsec) pulse of 130 volts. Actually, we put into the machine no tube which will not fire on a 17-μsec pulse of 130 volts. We have run a group of tubes for about 5,000 hours. The required pulse width to fire has not gone up to 30 μsec, which is the minimum time we will have available under all circumstances. None of the tubes moved up to near 30 μsec at 130 volts.

A life test of about 7,000 hours was run at a 20-cycle-per-second pulse rate, and as far as we know there seemed to be no deterioration, and we anticipate approximately 25,000 hours of life. This is in terms of the machine cycle, taking advantage of the fact that each tube is not printed on every line on a 24-hour basis.

M. B. Adams (Stanford Research Institute): What life has been obtained from the *2050* thyratrons? And, what peak current is used for printing?

Abraham Pressman: We use a peak current of 1.2 amperes with a pulse time of about 1 millisecond to print. In the actual machine operation, each tube takes this current roughly once out of every 100 milliseconds at a 600-line-per-minute printing rate.

We have a number of tubes on life test with this peak current and width at this repetition rate which have been going for about 3,000 hours with no deterioration in the peak current outputs or the grid biases required to keep the tubes off when they are supposed to be off, or deterioration in the voltage required to fire the tubes when you want to fire them.

Mr. Larson: What sort of fail-safe interlocks are provided for broken paper?

Earl Masterson: There is a detector immediately below the typewheel which rides against the paper. It is very much like a spring (a pair of springs may be used for more safety), and when the paper passes this point it will stop the print cycle at the end of that form.

There are also other indications when you are getting near the end of the ribbon supply and also the magnetic tape. When you are at the end of the reel a light turns on. In fact, we have lights on the control panel which will indicate anything that stops the machine, so that you don't have to search around to find out what happened. And this is all immediately in front of the operator. There are so many things on a machine of this complexity that it was necessary to put control lights on these and many more things to tell you where to look.

J. Drewe (British Tabulating Machine Company, Ltd.): Have you any means for detecting an error caused, for example, by an armature sticking and striking the printing wheel too late?

Earl Masterson: I don't believe an armature sticking would cause it to hit the wheel late. In fact, we don't know of anything that will cause it to print late. It has never happened, and all I can say is that our checking, of course, does not extend to the point of reading the printed page with a television system and seeing if we printed the right thing. However, in spite of the fact that we don't go that far, we do not know of any errors that the machine has made. As I pointed out, we have five machines in the field and we have not had any complaints on errors.

Most of the things that can happen beyond the last automatic checking stage are things that are permanent failures—or at least that is what we think they will be. They haven't happened yet. In other words, if an actuator should break in two it is a permanent thing which will be caught; if a hammer breaks in two it is permanent. As I say, there has been no trouble so far.

M. B. Adams (Stanford Research Institute): How fast is the tabulating feed feature of the printer in inches per second? What material is used for the format tape?

Earl Masterson: The tabulating feed, in other words, the feed which propels the paper over the areas where no print is required, will do this at 20 inches per second.

It might be of interest to know that every paper-feed operation in the printer is the same. We give a start signal to the clutch, which is engaged through a gearing system to the feed. After the paper has moved the proper distance, a stop signal is given. This is all there is to it no matter whether it is a single line, double line, triple line, or whether you are feeding a foot of paper. It comes up to the same peak speed and thus the start and stop are the same in every case, so it is no more difficult to feed several inches of paper than it is to feed one line, which is a 1/6 inch.

The format tape, which is a 12-channel tape to control this fast feed skipping, is of two kinds: a fiber base paper, and a rag base. There is nothing particularly special about it. It doesn't seem to be much of a problem.

B. Cox (Stanford Research Institute): Would you care to comment on the zero-suppression scheme?

Abraham Pressman: On command from the plugboard to start a zero-elimination operation on any column the machine will look for zeros and print spaces until it finds the first non-zero character, then it will print real characters. It can do a zero-elimination operation in any one of 18 different groups of columns in any one line

---

Application and Performance of Magnetic-Core Circuits in Computing Systems

R. D. KODIS

THE magnetic cores developed prior to 1952 were nonuniform and, therefore, extremely marginal in their operating characteristics. Not only did they present many circuit problems but they were difficult to manufacture as well. Since 1952, however, the picture has changed radically; rapid development of core materials, more advanced processing techniques, as well as new circuits and design techniques have all combined to place the magnetic core in a position equal to, if not superior to, all other digital components. Today this increased tempo in development is becoming evident in many laboratories and the number of new systems and components that employ magnetic cores is increasing rapidly. In fact, it is quite safe to predict that most computers designed in 1955 will use more magnetic cores than any other 2-state component. The promise of unusual reliability and long life of the magnetic core has been the motivating factor in most of the developments. Then, too, the possibility of reduction in cost, weight, and power requirements has its measure of attraction.

The magnetic-core circuits can be found performing the functions of storage, manipulation, control, amplification, regulation, and several minor special

R. D. Kodis is with the Raytheon Manufacturing Company, Waltham, Mass.

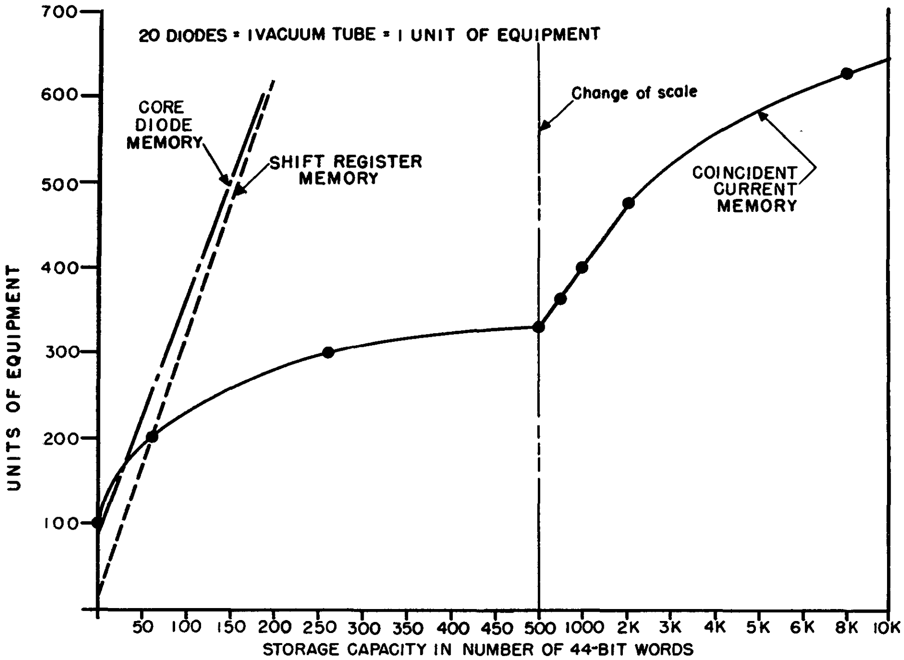


Fig. 1. Amount of equipment versus storage capacity

applications. These will be discussed in functional groups: internal memory; buffer memory or input-output memory; high-speed logic; and low-speed logic.

Internal Memories

The internal memory is usually a major factor in controlling the over-all system characteristics. Thus, the memory specifications are important and should include at least: (1) number of registers and their size; (2) mode of operation—parallel, serial, or any combination; (3) initial and repeated access time; (4) cost; (5) environmental conditions; (6) operating margins.

The unwritten requirements for all the core circuits to be discussed include reliability, a combination of operating margins, life, and transient conditions. The reliability of magnetic-core circuits has been demonstrated to be mainly dependent upon the equipment and components associated with the core rather than the core itself. Thus, it is the high ratio of cores to other components that provides such high reliability of large coincident current memories. Fig. 1 illustrates storage size for 44-bit words versus the amount of equipment for control drive selection and other functions considered to be part of the storage system. (A unit of equipment is defined as one vacuum tube or 20 diodes.) While in these memories, cores are used only in the main storage function, the use of cores in the selection, control, or drive functions would not radically change the comparison shown. In fact, it would improve the reliability of all three memories.[1-4] Fig. 1 presents a persuasive argument for using coincident current core storage in memories of 4,000 bits or more. Consequently, we find the coincident current memory most often used in large, general-purpose storage where its great saving in equipment per unit of storage easily offsets its lack of flexibility. On the other hand, the shift-register memory is found in small or special-purpose systems where it requires the least equipment. It becomes evident, then, that the combination of these two basic types of storage can provide useful circuits for special requirements, such as memory using coincident current to write in and shift-register techniques to read out.

Fig. 2 illustrates the low cost involved in using coincident current techniques in memories of 35 words (1,500 bits) or more. However, a qualification of this low number is usually required when fitting such a small memory into the system. Such qualifications might be a parallel memory in a serial system, certain decoding of address selection, or the flexibility required of a memory cycle. The additional equipment required usually pushes the dividing line closer to 50 words. The operating margins of coincident current memories lie in an area of 1 to 1.3, whereas shift-register memories have margins greater than 1 to 3. Systems with margins in excess of 8 to 1 are available.

The core-diode memory has served useful purposes in several systems. At present, however, it does have the disadvantage of a high overhead cost in equipment, as shown in Fig. 1, and it is relatively poor in operating margins, life, and reliability due to the operating conditions of the diodes.

The shift-register memory has an initial access time of 0.2 microsecond (μsec) and a repeated access of less than 2 μsec. The coincident current memory has an initial access time of 1 μsec and a repeated access time of less than 7 μsec.

Buffer Memories

In buffer memories, additional specifications are usually required, such as the ability to operate over a range of pulse rates, accept either parallel or serial in-

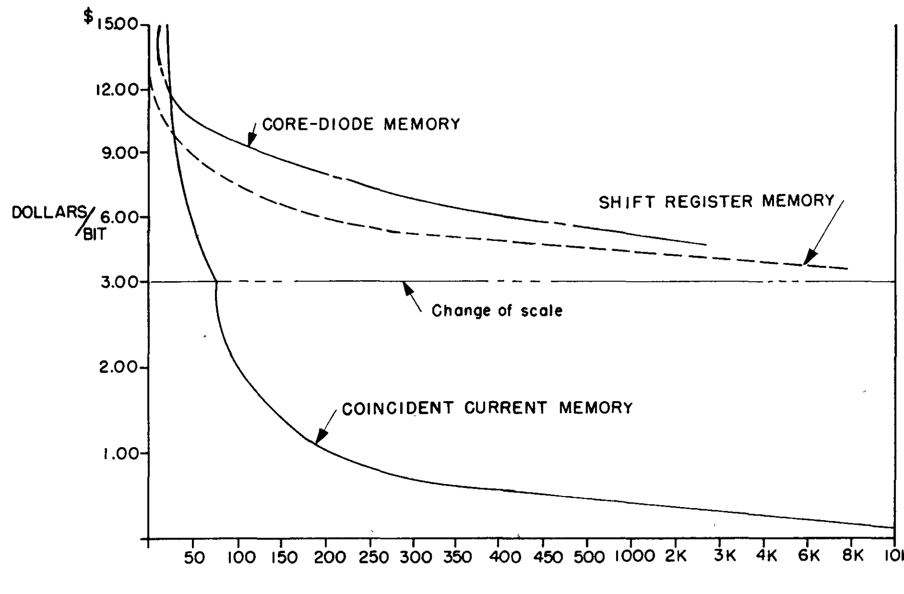


Fig. 2. Production cost of memory versus storage capacity

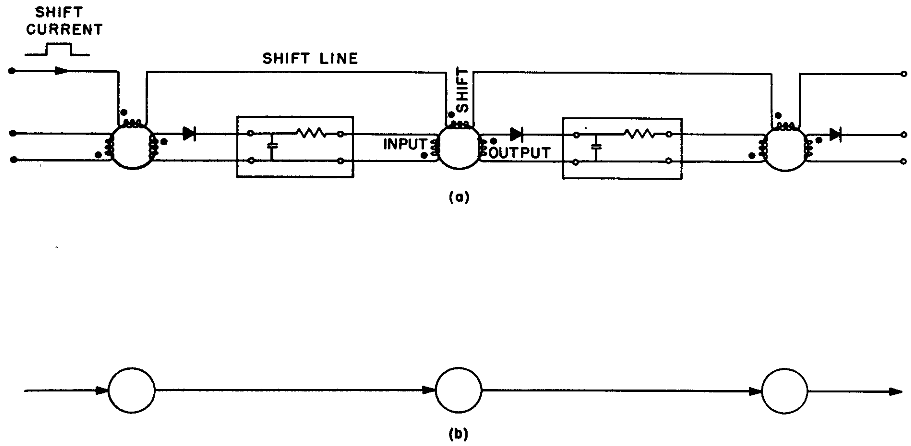


Fig. 3. Single-line shift register

formation, complement, insert, or shift stored information. The magnetic-core shift register is the only component which can accomplish these functions directly. Thus, the shift register is extremely practical for use in buffer memories as large as 10,000 bits. Buffers have also been built using core-diode-type storage. The experience reported from these equipments showed usable operating margins, but relatively poor reliability and life, due again to the diode operating conditions.

Flexibility of controls makes the shift register the only type of buffer which is usable with FM types of magnetic recording systems developed at the author's company. A register is normally used in such manner so that all information is transferred upon command. However, the ability to transfer part of the information any number of times while the remainder is static is extremely advantageous in simplifying certain timing problems.

The 2-core-per-bit or Harvard-type[5] shift register has almost no system application which is not better implemented by the 1-core-per-bit "single line" registers developed at the company. The principle of the "single line" register can best be understood by referring to Fig. 3. As may be seen from the illustration, all the magnetic cores are driven from a common shift line. In this application, the read-out of the n^{th} core is delayed from writing into the $n+1$ core for a period greater than the shift-pulse duration.

Magnetic shift registers using selenium diodes are still doing useful work after 20,000 hours. However, a continual increase in forward resistance of the diode has taken place until the resistance has about doubled at 18,000 hours. The design of long-life registers using selenium diodes must recognize this situation. Magnetic-shift registers employing germanium, gold-bonded diodes have had more than 8,000 hours operation with no measurable changes in operating characteristics or output signals.

High-Speed Logic

The logic required for the control and data manipulation in our hypothetical system can be handled, in many cases, by magnetic-core logical elements. For pulse rates in the megacycle range the core logic must still be considered merely a laboratory device. However, for pulse rates up to 500 kc, the cores should definitely be considered for logical structures. The types of circuits that are being studied for this purpose are too numerous for recounting here.[6–8] Also, since only practical devices are being compared this reduces the discussion to ones now being used in workaday systems.

The "single line" magnetic-core shift register is the basic type of circuit used in the "single-line" magnetic-core logical element. Since these are relatively new techniques a brief description is in order. Fig. 4 illustrates a circuit diagram in which one core feeds two others. In theory, this type of branching may be done for any number of cores. The inhibit function can be accomplished by reversing the polarity of the input winding, as shown in Fig. 5. At present, the magnetic-core logical element should be limited to a combination of three inhibit or input windings. However, more affirmations may be made on one logical stage by buffering the outputs of several elements together on the diode, as shown in Fig. 6. As a matter of fact, the output from one element can be used to drive three other stages.

The combination of these circuits permits the accomplishment of any logical function. For example, the "Exclusive Or" function is easily constructed with three cores, as shown in Fig. 7. The simplicity with which many other functions are implemented suits it for use in moderate- and low-speed systems. For example, a serial arithmetic unit capable of handling 24-digit binary numbers and sign has actually been built. It uses only 150 magnetic cores, 150 diodes, and 8 vacuum tubes to perform all the manipulative functions, controls, and storage and operates at 0.25 mc pulse rates. It requires only 0.25 cubic foot of space, weighs 15 pounds, and consumes a mere 130 watts, including filaments. A unit of this type would have obvious application to air-borne problems where volume, weight, and power requirements are prime considerations. The range of environmental conditions over which this type of component will operate is quite useful even though it is still not all

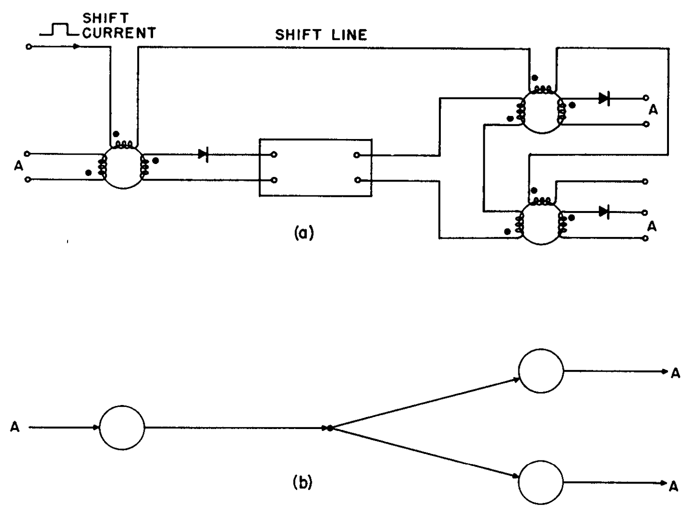


Fig. 4. Amplifier or branch circuit

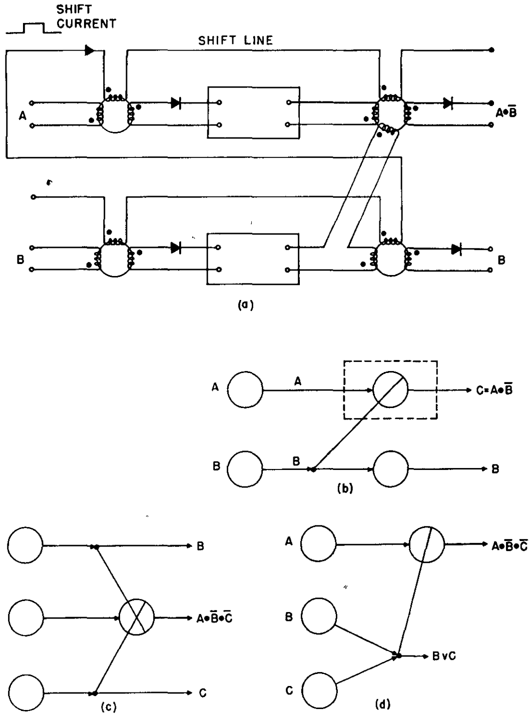


Fig. 5 (left). Inhibit circuits

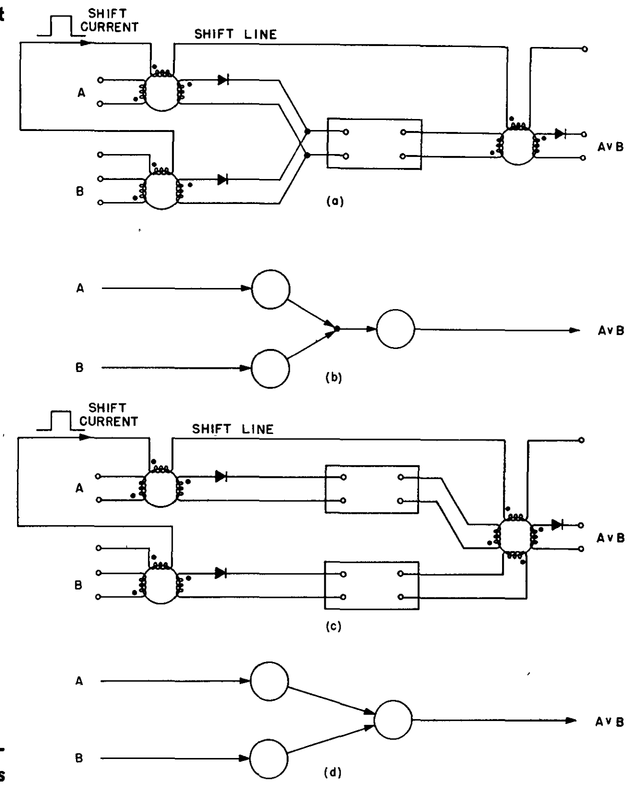


Fig. 6 (right). Mixing or buffer circuits

that is desired. For example, good operation can be held over the ambient temperature range from −80C to +85C.

The magnetic-core logical element is designed to be a low-impedance low-power device. This enhances the core circuits but leaves them at too low a voltage level to satisfy the requirements of vacuum-tube circuits. The limitations on the usage of a magnetic-core logical element still depend upon pulse rate, power level, and choice of diodes. (On the other hand, transistors work well with this type of magnetic core circuit.)

Low-Speed Logic

In the low speed-portion of this system the magnetic-core logical element may be required to control mechanical equipment. If the logical statements require many cores for their implementation, the low-power cores would be used and power amplification provided for the output. However, where the control function is simple, a magnetic core of suitable size may be employed to provide the power directly. The cores themselves can be used for power gain to operate thyratrons or relays. Perhaps of even more significance is the direct read-in and control of magnetic cores from mechanical and electromechanical sources. This elimination of all vacuum tubes and transistors places the reliability problem squarely on the diode. On the other hand, magnetic core-transistor circuits which can perform all of the foregoing logical functions employ no tubes or diodes. Thus, there is a basic choice: diodes with a few tubes versus transistors.

The "single-line" magnetic-core logical element has a reliability close to the "single line" magnetic shift register. The number of elements driven from one tube depends upon the maximum density of positive information, power level of the driving tube, repetition rate of the information, power level of the cores, and the high-voltage supply used. The range of 16 to 65 cores per driving tube is being used. Tube types such as *6AU5*, *5881*, and *6293* are being used as drivers. The driver can be considered as a pulsed power supply, since they do not normally enter into the logical structure.

The magnetic core is also being used as a different type of logical element or gate in selection systems. Several laboratories have developed saturable transformers, biased cores, and time-pulse sequence gates. These techniques are useful as driving source for other magnetic ciruits, magnetic recording, writing, and reading selection systems.[9–10]

The type of equipment required to transfer information from card to tape or tape to printer is especially susceptible to implementation by magnetic core circuits. The shift register storage eases

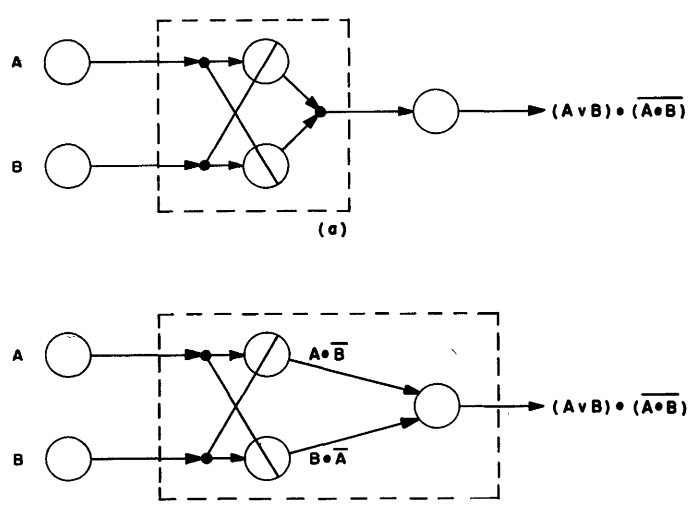


Fig. 7. Exclusive "OR"

the requirements of reading into storage from card reading brushes or other mechanical contacts. This is due to the ability to make the read-in to each core independent, and to combine parallel read-in with serial transfer. These equipments may be required to perform the functions of code conversion, speed conversion, and radix conversion, as well as format controls and checking. These functions, when accomplished by magnetic-core logical elements, are best accompanied by shift-register-type storage.

A simple example is a binary to binary coded decimal converter and binary-coded decimal to binary. This unit uses 100 cores to convert more than 1,000 decimal digits per second.

Summary

There can be little doubt that the magnetic core combines the best elements of reliability and long life to such a degree that its present and future use in digital work is assured.

References

1. Static Magnetic Memory—Its Application to Computers and Controlling Systems, **An Wang.** *Proceedings*, Association for Computing Machinery, New York, N. Y., May 2–3, 1952.

2. A Static Magnetic Memory System for the Eniac, **I. L. Auerbach.** *Ibid.*

3. A Coincident-Current Magnetic Memory Cell for the Storage of Digital Information, **W. N. Papian.** *Proceedings*, Institute of Radio Engineers, New York, N. Y., vol. 40, no. 4, April 1952.

4. A Myriabit Magnetic-Core Matrix Memory, **J. A. Rajchman.** *Ibid.*, vol. 41, no. 10, October 1953.

5. Static Magnetic Storage and Delay Line, **An Wang, W. D. Woo.** *Journal of Applied Physics*, New York, N. Y., vol. 38, June 1950.

6. Testing Magnetic Decision Elements, **J. D. Goodell.** *Electronics*, New York, N. Y., January 1954.

7. Magnetic Cores as Elements of Digital Computing Systems, **M. K. Haynes.** *Technical Report to Office of Naval Research*, Washington, D. C., August 28, 1950.

8. Magnetic Binaries in the Logical Design of Information Handling Machines, **N. B. Saunders.** *Proceedings*, Association for Computing Machinery, New York, N. Y., May 2–3, 1952.

9. Magnetic Switching Circuits, **R. C. Minnick.** *Journal of Applied Physics*, New York, N. Y., April 1954.

10. Saturable Transformers as Gates, **B. Moffat.** *Quarterly Report, Second Series of the Computer Fellowship No. 347*, Mellon Institute of Industrial Research, Pittsburgh, Pa., March 31, 1954.

Discussion

F. T. Andrews (Bell Telephone Laboratories Inc.): How much energy is required to shift a bit of information from one stage to the next in a typical single-line shift register? How does this compare with that in a 2-line shift register which uses simply a diode in the coupling network?

R. D. Kodis: The 500-kc register, which has been used fairly extensively, dissipates or has an input to it of a half-watt average at 500 kc with all ones going through it. The energy required in a single-line register is merely a function of the core and diode available.

In this particular register we wanted an output voltage suitable for driving vacuum tubes, and we wanted a certain impedance level, so that this is a fairly low power register. If you compare this with a double-line register you will find that the energy per shift pulse using the same core and the same diodes might be higher. You will find, though, that you require two shift pulses for the double line. You only require one for the single line, and you find the net result is you are using less power input to the single-line register than you are to the double.

To put this in other numbers, we have built single-line registers working off 100–125-mil shift pulses with a peak voltage drop across the shift winding of 8 volts.

V. L. Newhouse (Radio Corporation of America): How does the 1-core-per-bit circuit compare with the Ramey type of circuits with regard to economy, and how do they both compare to transistor circuits with regard to reliability?

R. D. Kodis: If the Ramey circuit referred to is the so-called 1-cycle magnetic amplifier, then the operation of the core is very similar, but the circuits are certainly different. The Ramey circuit, if this is the one, is strictly an amplifier, while the other circuit is a digital storage device connected so as to be able to be shifted.

As far as economy, I don't see that there is a radical difference between them. As to comparing them both to transistor circuits with regard to reliability, if we used a transistor core-type shift register on which we use no diodes and no vacuum tubes at all, then that particular shift register is as reliable as the transistors, because that certainly is the weak link in the proposition. In fact, it may be better than straight transistor circuits because the transistor is used in an uncritical manner; it is only asked to supply a lump of energy rather than to operate between any particular limits as far as stability is concerned. It merely has to have sufficient gain to supply this lump of energy to the core. The operating margins are so large that this circuit should have longer life.

To compare it with a single-line register driven with tubes, again the tubes are used in a fairly noncritical manner there. I would suspect, not knowing, that if you are referring to transistor shift registers without cores (let's take a 50-digit word or a 40-digit word), there will be with transistors, 80 transistors against 40 cores and 40 diodes and one or two vacuum tubes, so there is no waste power or weight involved and I guess you probably have a toss-up. At this stage, I might rely on the tubes. Maybe a year from now we might think differently.

Lawrence Jones (Radio Corporation of America): What is the ratio of tubes to cores for coincident core circuits, and what are the characteristics of the delay-line shift register with respect to spikes on the leading edge of the waveform?

R. D. Kodis: The first part of the question can be answered by some generalized examples. A 1,000-word coincident current memory will have, for the entire memory system, something over 300 vacuum tubes and envelopes, whereas if you go up to 2,000 words, double the size of the memory, this may be something in the order of 450. If you go to double that again you may go up only by 40 to 50 tubes. A particular design of something over 10,000 words involves around 600 vacuum tubes. This is not using any cores in the driving or selection functions.

In answer to the second part of the question, the single-line register output depends upon the specific design of the delay network between the two cores. In the very simplest of cases it is a fast-rising slope followed by an exponential decay, using up the rest of the period, for the maximum repetition rate for which the equipment was designed.

In more sophisticated cases it appears to have an exponential rise and something looking like half a sinusoid for the rest of the time. In other words, there are essentially no spikes on the leading edge of the waveform.

I might say, this is because what you are essentially doing here is integrating the output; you are not looking at a voltage waveform developed across the core, but rather at the somewhat integrated output of the core, of the flux change.

R. G. Lilly (Bureau of Ships): Do you anticipate or envisage an internal low-energy megacycle memory capable of being operated solely with semiconductors?

R. D. Kodis: I guess there are such memories around. The capacitor diode memory is certainly capable of operating at the speeds mentioned here and it would fit generally into this classification of a semiconductor memory.

There are other types of semiconductor memories around which are apparently not being disclosed publicly at this time, so that we don't have much to comment on. I don't see why we shouldn't expect the type of memories now being built with ferrite cores to be up in this range at some later date.

Teletype High-Speed Tape Equipment and Systems

W. P. BYRNES
ASSOCIATE MEMBER AIEE

COMPUTER input-output is basically a communications process, the link between the human element and the machine element. The very nature of this process creates a close relationship between computer input-output and telegraph communications. This was recognized in the early days of computer development when perforated paper tape, such as has been employed in the telegraph field for many years, was the principal input-output medium used.

It was felt, therefore, that the computer field would be interested in some recent developments in high-speed tape-reading and tape-punching equipment which operates at a speed of 60 characters per second. The speed of standard readers and punches is 6 to 10 characters per second.

It is recognized that even a speed of 60 characters per second does not meet the requirements of some of the large-scale digital computers; however, there are many applications in input-output systems where high-speed readers and punches will enhance the use of paper tape.

1. Paper tape is sometimes used as an auxiliary medium in large-scale systems where magnetic tape handles the bulk of input-output data. Problems with relatively little input-output but with much computation can be handled via the paper tape.

2. The paper-tape equipment can also be used as a high-speed conversion means between paper tape and magnetic tape. By this intermediate step at the input, the maximum pulse packing density of the magnetic tape can be utilized. If the data were put directly on magnetic tape from a random source, like a keyboard, the pulse density would be much less, because the magnetic tape is not stepped at a uniform rate. In the output conversion from magnetic tape to paper tape, the 60-character-per-second punch permits the magnetic tape to be read at higher speeds, which is certainly desirable. Important by-products of these intermediate paper-tape processes are inexpensive permanent records, in coded form, of both input and output.

3. In small-scale digital computers the requirements may be such that equipment operating at 60 characters per second can be used as general input-output.

W. P. Byrnes is with the Teletype Corporation, Chicago, Ill.

A further advantage in the use of paper tape in input-output systems includes the possible integration of computer installations with communications facilities. This is in line with the trend toward a "common language" with which to transfer data mechanically between different types of installations. The 5-hole tape, punched in communications code, has been adopted by the United States Steel Corporation[1] and others as the "common language" medium.

In order to meet the requirements of the computer field and also to provide equipment for experimenting with higher speed telegraph communications, two basic units have been developed: a tape reader and a tape punch, both operating at 60 characters per second. In addition, several electronic units were developed to tie the reader and punch into various systems.

High-Speed Tape Reader

The 60-character-per-second tape reader is a motor-driven device which senses tape mechanically and closes electric contacts in accordance with the code combination sensed. Fig. 1 shows two readers arranged for flip-flop operation. The readers operate alternately; when one reader runs out of tape it stops and turns on the other reader. Fully perforated or chadless tape containing up to eight bits of information per character can be read by this type of reader. The contacts can carry up to 0.075 ampere at 120 volts with a resistive load. Inductive loads naturally require spark-suppression circuits.

The sensing portion of the unit is illustrated in Fig. 2, which shows one of the sensing mechanisms. Associated with each sensing pin is a sensing pin spring, a switch bar, marking and spacing contacts, and a contact spring. The eccentric drive and main bail are common to all sensing positions. A vertical, reciprocating type of motion is transferred to the main bail by means of the connecting rod and eccentric drive. On the upstroke the tape is sensed, and on the downstroke the tape is advanced to bring the next character into the sensing position. The sensing pin extends through a hole in the main bail at point A and rests on the switch bar at point B. At the other end of the switch bar is a contact assembly consisting of marking and spacing contacts. The terms "marking" and "spacing" refer to the two complementary parts of the binary digit. The sensing pin is shown in the down position, having been pulled there by the main bail. In this down position the switch bar makes contact with the spacing contact.

As the eccentric drive starts the upstroke, the main bail is allowed to move up under the tension of the main bail return spring. When the bail moves up it allows the sensing pins to move up under the force of the sensing pin springs. If there is a hole in the tape, the sensing pin continues its upward motion, passing into the hole. The switch bar, which follows the sensing pin, rocks from the spacing contact to the marking contact as the sensing pin enters the hole in the tape. This completes a circuit from the contact spring terminal, through the switch bar, and out the marking contact terminal, putting out

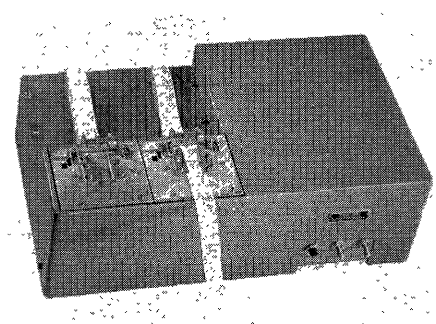


Fig. 1. High-speed tape readers

a marking pulse. The marking contact remains closed for approximately 8 milliseconds out of the total operating period of 16.7 milliseconds. If there had been no hole in the tape, the upward motion of the sensing pin would have been halted by the tape and the switch bar would have remained on the spacing contact. In some cases it is desirable to take an output from both the marking and spacing contacts, in order to provide complementary pulses. This is very useful when it is necessary to recognize specific characters. In addition to the five code switch bars there is a sixth switch bar which puts out a marking pulse with every operation. The sixth pulse is used for blank recognition and control purposes in associated equipment.

On the downstroke, the main bail re-

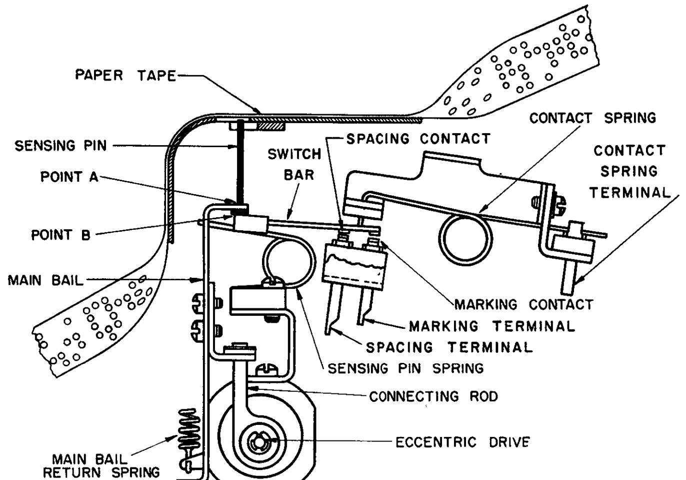


Fig. 2. Tape-sensing mechanism

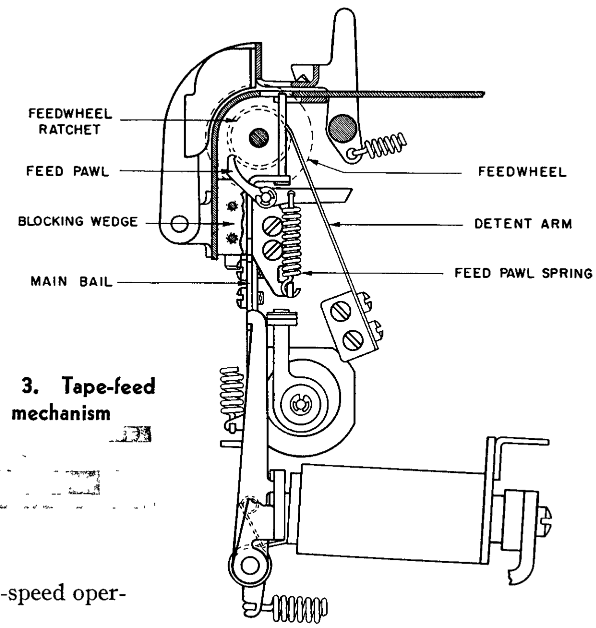


Fig. 3. Tape-feed mechanism

turns all sensing pins, switch bars, and contacts to the spacing position and advances the tape. The tape-feed mechanism, illustrated in Fig. 3, is basically the well-known pawl and ratchet, lightened and refined to stabilize the operation at high speeds. A feed pawl is pivotally attached to a bracket mounted on the main bail and is held in contact with the feedwheel ratchet by means of the feed pawl spring. The mechanism is shown with the main bail in the extreme down position. Near the top of the main bail stroke, the feed pawl engages a tooth in the ratchet and on the downstroke advances the feedwheel ratchet one tooth. This in turn moves the next character in the tape into sensing position. At the bottom of the stroke, the feed pawl wedges between the ratchet teeth and the blocking wedge. This insures smooth deceleration and prevents overriding, because the ratchet rotation is controlled solely by the feed pawl which has simple, harmonic motion. A detent arm prevents the ratchet from backing up as the feed pawl moves up to engage the next tooth.

An interesting feature of the feeding operation is that the sensing pins are still projecting through the holes at the beginning of the tape advance. The sensing pins yield to the forward motion of the tape until they are withdrawn from the holes, at which time they snap back to their normal position ready to sense the next character. Therefore, a larger portion of the cycle is available for tape feed than if it were necessary to wait until the sensing pins were out of the way.

The reader is started and stopped by means of a magnet-controlled latching mechanism. The reader can be stopped within the period of one character when operating at 60 characters per second. There are several reasons why the tape reader is capable of such high-speed operation:

1. Small, lightweight parts reduce required forces and minimize transient effects.
2. The rocking motion of the contact mechanisms eliminates contact bounce because the moving parts are not required to stop abruptly.
3. The tape wraps 90° around the feed sprocket to engage a large number of feed holes, hence substantial pulling power can be transmitted to the tape.
4. The tape can be advanced while the sensing pins are still in the tape, allowing more time for the tape-feed operation.
5. The driving motion is essentially simple harmonic. This is especially advantageous for tape feeding, because the tape is started and stopped near the peaks of the sinusoidal motion where the velocity is low.

High-Speed Tape Punch

The high-speed tape punch is a motor-driven magnet-controlled device which punches 5-, 6-, or 7-hole tape (two standard tapes or one parchment tape) at 60 characters per second. Fig. 4 illustrates a 7-hole punch. Note that there are eight selector magnets, seven for controlling 7-punch pin mechanisms and one for controlling the tape-feed mechanism. All selector magnets must be energized simultaneously, and will pull up in 6 to 8 milliseconds when pulsed with 25 milliamperes at 120 volts.

Fig. 5 shows two punching mechanisms, one behind the other, with their respective selector magnets. Each of the seven punching mechanisms consists of a blocking pawl, long and short toggle arms, a drag link, and a punch pin. The eccentric main shaft generates a continuous 60-cycle reciprocating vertical motion which is transmitted to the punch bail by means of the punch bail drive link. This motion is then transmitted from the bail to all seven toggle arms, drag links, and finally to the punch pins. The total peak to peak displacement of the vertical motion is 0.090 inch.

The eccentric main shaft is shown in the lowermost position. Note that the long toggle arm of the outer mechanism has cleared its blocking pawl, as a result of its selector magnet being energized, and that the long and short toggle arms behave as one stiff member, transferring the complete displacement of the eccentric to the punch pin. This complete displacement is sufficient to drive the punch pin through the paper tape. The selector magnet for the other mechanism is shown de-energized, causing its blocking pawl to block the long toggle arm. This causes the long toggle arm to rotate slightly counterclockwise, throwing the joint between the long and short toggle links to the right as shown. Therefore, since part of the drive displacement is used in collapsing the toggle joint, the punch pin is not driven a sufficient distance to go through the paper tape.

The selection process must be examined from a dynamic standpoint. Picture the eccentric main shaft rotating and the toggle arms moving up and down. The long toggle arm is so disposed with respect to its blocking pawl that it rises above the pawl during the top 0.006 inch of travel (80° of eccentric main shaft rotation). The blocking pawl must be in the selected position, blocking or unblocking, before the long toggle arm reaches the pawl on the downward portion of the stroke.

Obviously this requires that the energization of the selector magnets be properly timed with respect to the punching mechanism. This synchronization is provided by means of a contactor arrangement which is driven from the main shaft and properly phased with respect to the punching mechanism. The contactor determines when the selector magnets are to be energized. Usually the contactor operates in conjunction with an external storage register which delivers the current pulses to the selector magnets.

The tape-feed mechanism is shown in Fig. 6 and is controlled in the same manner as punching mechanism. Attached to the lower portion of the short toggle arm is a feed pawl adjustable link which transmits the reciprocating drive motion to the feed pawl with a 180° phase reversal. The feed pawl is held in contact with the feed wheel ratchet by means of a feed pawl spring. Tape feed is selected by energizing the tape feed-out magnet which allows the toggle joint to remain stiff, thereby transmitting the complete eccentric drive displacement to the feed pawl. At the bottom of the drive stroke, the feed pawl is at the top of its stroke and engages a tooth in the ratchet. As the toggle arms move up, the feed pawl moves down, rotating the feed wheel ratchet counterclockwise one tooth and advancing the paper tape one character space. The feeding cycle is phased with the punch cycle in such a manner that feeding occurs immediately after the punch pins are withdrawn from the tape.

In order to insure smooth operation of the feed wheel, an eccentric stud cams the feed pawl into the ratchet tooth at the bottom of the feed pawl stroke. This prevents any transient overtravel and oscillation. If the tape feed-out magnet is not energized, the joint collapses and the complete eccentric drive displacement does not reach the feed pawl. The feed pawl does not rise high enough to engage the next tooth in the feed wheel ratchet, and the tape is not advanced. Manual tape is provided by the tape feed-out lever which holds the armature in the attracted position.

The features of the punching and feeding mechanisms which enable high-speed operation are summarized as follows:

1. Positive drive in both directions, into and out of the tape. This provides controlled motion throughout the complete cycle and does not depend on spring-loaded mechanisms.
2. Approximate simple harmonic motion. Selection and punching are performed near the peaks of the sinusoidal motion where the velocities are low. The tape is also accelerated and decelerated smoothly.
3. Continuous drive—no clutches.

Systems and Applications

Applications of the high-speed tape equipment just described fall into two natural categories: input-output for digital computers and high-speed data communications systems. Several of the input-output applications were covered earlier in this paper. Since the technical aspects of applying the reader to these systems are somewhat elementary, they will not be dwelt on. It is sufficient to say that the output of the reader contacts can be used for whatever purpose desired, whether it be in a paper-tape-to-magnetic-tape converter or as direct input to the computer.

Application of the punch, however, is more involved since it ordinarily ties into the output storage register of the computer, with the punch contactor releasing one character at a time from storage. Where the output storage register is of the dynamic type, e.g., a magnetic drum, a buffer storage must be used. This is necessary because a character may not always be available when the contactor calls for one.

For these applications the Teletype Corporation developed a buffer storage and control unit which takes up the slack between two dynamic systems. Fig. 7, a functional block diagram of the unit, shows three sections of activity: receiving, storage, and output sections. Characters made up of parallel pulses are fed into the receiving section which consists of a bank of eight thyratrons, seven for code pulses and one for a control pulse. These pulses may vary between 10 microseconds and 15 milliseconds in duration

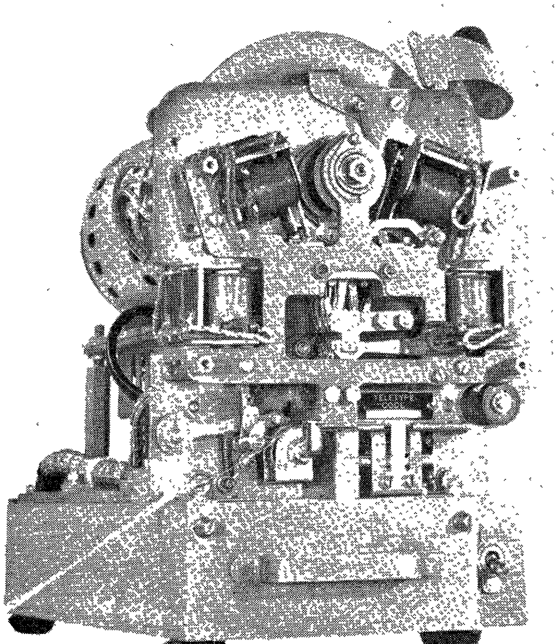


Fig. 4. High-speed tape punch

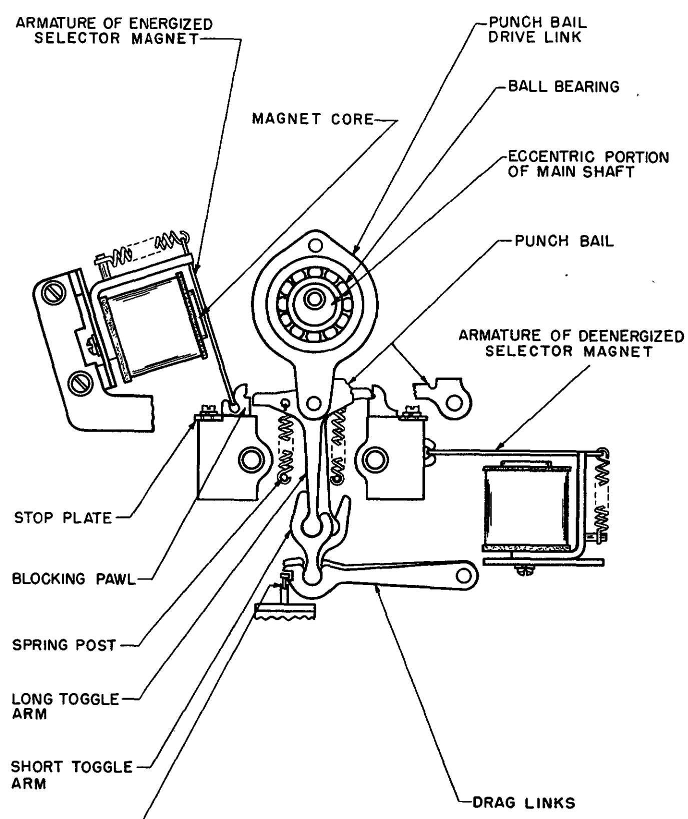


Fig. 5. Punching mechanism

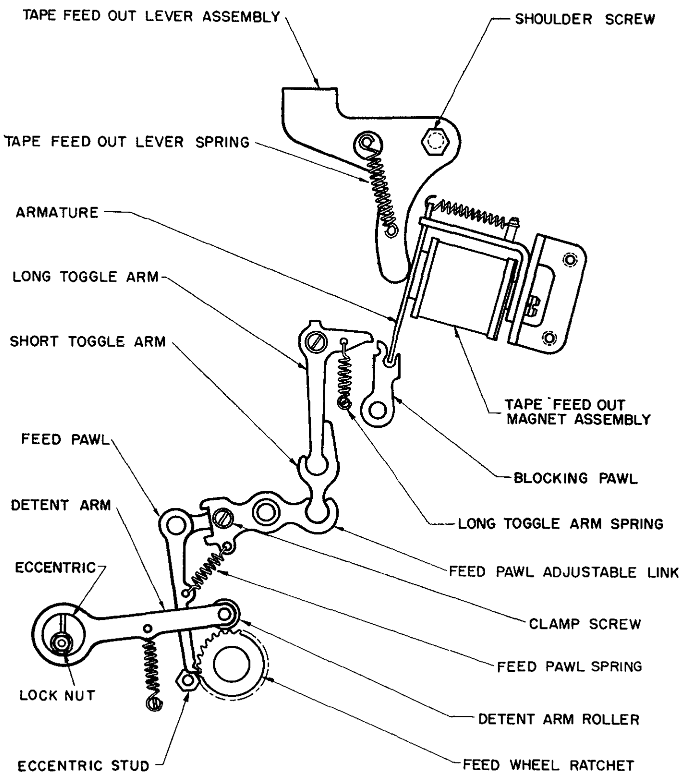


Fig. 6. Tape-feed mechanism

and between 30 and 100 volts in amplitude.

When the complete character has been received it is transferred out of the receiving section and into the storage section, a bank of seven thyratrons. The character remains in the storage section until the contactor pulse from the punch transfers it to the output section. The output section also has eight thyratrons, in series with each of which is a punch selector magnet. When these output thyratrons are fired, the proper selector magnets are energized and the character is punched in the tape. In the meantime, the next character may have been received and is waiting for the next contactor pulse from the punch. Characters may be fed into the buffer storage and control unit at any rate up to the punch speed. Obviously, the rate of input to the storage and control unit must not exceed the speed of the punch or characters will double up in storage. To provide a margin of safety, the input speed should not exceed 98 per cent of the punch speed.

In addition to the applications where the equipment plays a direct part in the input-output system, there are others that supplement the main input-output. One is a tape reproduction system which makes use of one tape reader and several punches. This system could make as many as six simultaneous reproductions at 60 characters per second. The reader contacts can control the punch selector magnets directly. Another application involves two tape readers used as a 60-character-per-second tape verifier. Since the reader contacts can be wires for complementary outputs, it is a simple matter to match the combination set up in each reader. Two tapes can, therefore, be compared. When two characters in the tapes are not identical, a nonverification signal is put out. This signal can be used to stop the readers, ring a bell, and so on.

Applications in the field of data communications are, of course, of interest to the computer field. High-speed readers and punches make possible the development of high-speed communications. Fig. 8 shows a simplified block diagram of a 60-character-per-second start-stop tape communications system which was developed solely for the purpose of exploring the feasibility of high-speed telegraph communications. Start-stop terminology implies that the transmitted signal is a series of information pulses accompanied by start and stop pulses. The standard telegraph code consists of a start pulse, a combination of five information pulses, and a stop pulse. The start and stop pulses are used for synchronization purposes.

The output of the tape reader is a group of six parallel pulses, five information pulses, and one control pulse. These are fed into the electronic transmitting distributor where a parallel to serial conversion is performed, and a start and stop pulse are added to each character. The output of the distributor then is the serial start-stop signal which is to be transmitted over communications facilities to the receiving equipment.

The receiving terminal consists of an electronic receiving distributor, the buffer storage and control unit, and the high-speed punch. The receiving distributor converts the serial start-stop signal back into its original parallel form of five information pulses and a control pulse. This parallel signal is then fed to the buf-

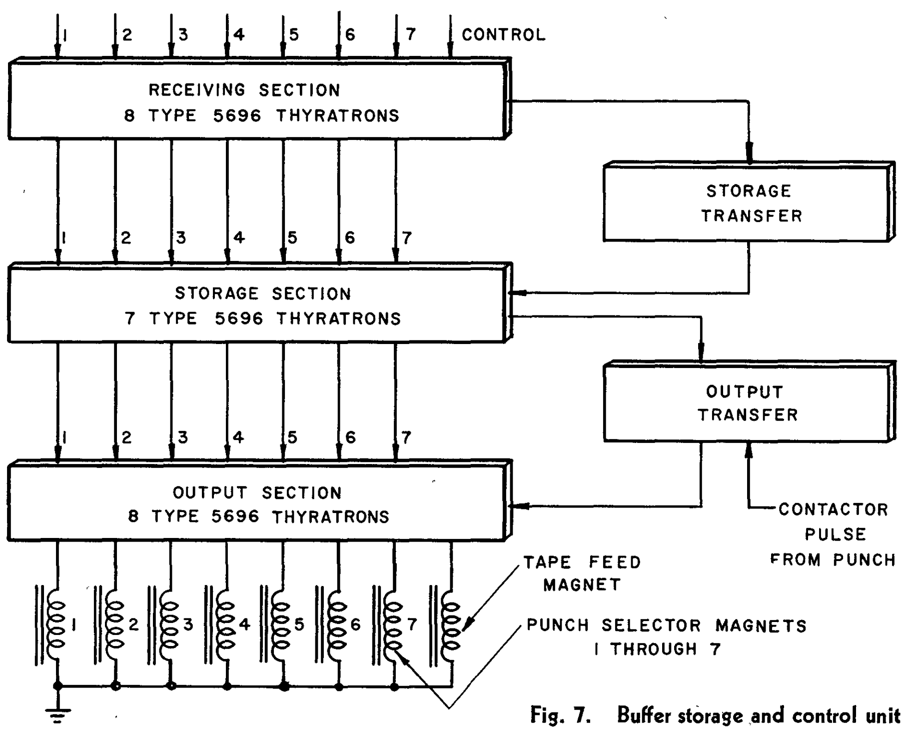


Fig. 7. Buffer storage and control unit

fer storage and control unit and subsequently, to the high-speed punch as previously explained. In this system, the punch is run slightly faster than 60 characters per second in order to insure that it will always be faster than the signal speed. The actual speed of the punch is 63.3 characters per second. Techniques used in the electronic transmitting and receiving distributors were taken from Teletype's electronic time division multiplex.[2]

A field trial of the 60-character-per-second start-stop communication system was conducted during the summer of 1954, extending over a period of 1 month and a distance of some 200 miles. The trial showed that such communication is technically feasible. One channel operating at this speed offers the information-carrying capacity of six to ten conventional-speed channels. This capacity to handle a large number of information pulses per second is gained at the expense of using relatively short pulses and a correspondingly wide frequency band. Furthermore, the short pulses are relatively vulnerable to distortion by impulse noise, so that special precautions are necessary to exclude such noise from the circuit. Whether the compressing of several conventional channels into one high-speed channel offers enough advantage to telegraph customers to justify the special transmission facilities required has not yet been finally determined.

Another type of high-speed tape communication system involves the transmission of the five information pulses in parallel at 60 characters per second without going through the parallel-to-serial and serial-to-parallel conversions. This, of course, requires at least five parallel communication channels. The transmitting terminal might consist of only a tape reader, and the receiving terminal of only the buffer storage and control unit and punch. This system has the advantage from the transmission standpoint that the information pulses are longer than the pulses in the serial start-stop signal. In fact, the pulses in the 60-character-per-second parallel system can be of approximately the same length as the pulses in a standard 10-character-per-second start-stop telegraph signal. This leads to the possibility of putting each of the parallel pulses on a separate standard telegraph channel. Further investigation on this application will also be required.

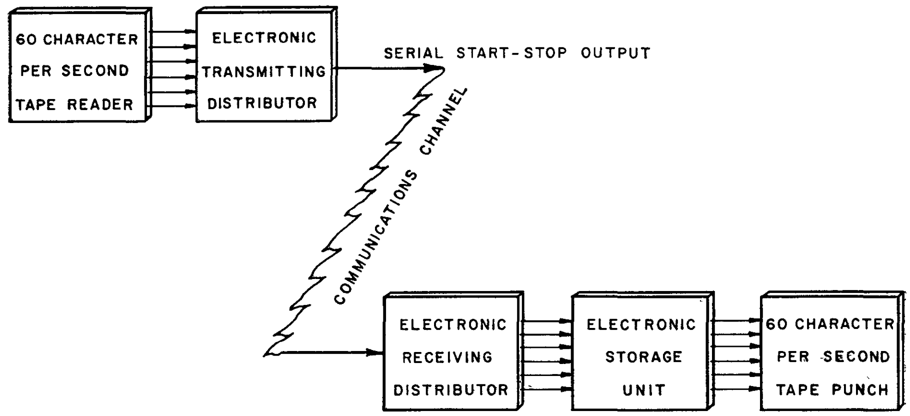


Fig. 8. 60-character-per-second tape communications system

Summary and Conclusion

The equipment which has been described appears to have a variety of applications in computer systems. To summarize briefly, these applications are

1. General input-output for small-scale digital computers.
2. Auxiliary input-output for the higher speed large-scale computers.
3. In data communication for computer installations of any size.

Teletype is interested in the computer applications of its equipment and believes that developments of high-speed telegraph apparatus may solve many of the input-output and communications problems of the computer field.

References

1. A New Approach to Office Mechanization: Integrated Data Processing Through Common Language Machines. Presented by United States Steel Corporation for the American Management Association, February 25–26, 1954.
2. An Electronic Time Division Multiplex Set, T. A. Hansen, R. D. Slayton. *AIFE Transactions*, vol. 70, 1951, pp. 354–61.

Discussion

J. H. Howard (Burroughs Corporation): What bandwidth (including guard bands) is required for 6-, 10-, and 60-word-per-minute transmissions?

W. P. Byrnes: Six, 10, and 60 characters per second, I believe you meant?

Well, for six to ten characters per second the standard telegraph band can be used, which is 170 cycles wide. The exact band required for 60 characters per second has not yet been determined. Investigations are presently going on as to what the optimum channel spacing should be. In the field trial that we conducted we used 1-voice channel for the signal. It was a 2,000-cycle on-off tone signal.

E. I. Organick (United Gas Corporation): What checking systems are available between information received and transmitted? How does this checking system compare with that of IBM's transceiver system?

W. P. Byrnes: At the present time there are no checking systems used in this particular system. As I mentioned, it was developed for the purpose of exploring the field of high-speed telegraphy.

J. L. Hill (ERA Division, Remington Rand, Inc.): What is the duration of reliable contact in the reader in degrees of cycle? How many characters will be read after a "stop reading" signal to the clutch or motor before the tape stops?

W. P. Byrnes: The pulse from the reader is approximately 45 to 50 per cent of the cycle. At 60 characters per second this approximates 8 milliseconds. The unit will stop in one character; on the one intended.

Operating Characteristics of the National Cash Register Company's Decimal Computer, The CRC 102-D

R. M. HAYES

THE *102-D* is a general-purpose computer using a magnetic drum for its internal memory. This drum has a storage capacity of 1,024 words of nonvolatile, permanent storage, each word containing the equivalent of ten decimal digits. Any word of the internal memory can be used for storage of either instructions or data. When used for an instruction, a word is treated as 14 octal digits; when used for data in the arithmetical operations, a word is nine binary-coded decimal digits plus a 6-bit section containing a sign bit and an overflow bit.

Built into the computer is the ability to use magnetic tape for auxiliary storage. When in use, this magnetic tape is stored in a magnetic-tape unit, the *CRC 126*. This magnetic-tape unit has some independent abilities—in particular, the ability to search independently for desired information. A single such magnetic-tape unit, with a 1,200-foot reel of tape, has a storage capacity of well over 1,000,000 decimal digits. These data are stored in blocks of 80 decimal digits, each such block identified by a so-called block address. The searching ability of the tape unit involves the independent search for a specified block address.

There is, of course, further input-output and external control equipment. In particular, the operation of the computer can be monitored and controlled from the control console, a desk with an associated Flexowriter and various switches, buttons, and lights. There is available an auxiliary high-speed photoelectric punched-paper-tape reader operating at 200 characters per second. Similarly for output, a high-speed paper-tape punch operating at 60 characters per second is available. Finally, punched-card input and output is possible involving only the addition of the necessary punched-card equipment.

The most satisfactory means of describing the operating characteristics of the machine is in terms of the command code. Each command will be considered in turn and significant features discussed. First, however, some remarks concerning general command structure should be made. The instructions to the machine are in general three address instructions. In order to use efficiently the storage capacity of a single word, each address in a command is specified by four octal digits. There are 27 distinct and different operations available, and thus the particular command specifies the operation by a 2-octal-digit code.

R. M. HAYES is with the National Cash Register Company, Hawthorne, Calif.

Arithmetic Operations

1. *Add decimally:* This operation will take two 9-decimal-digit numbers, add them, and then store the resulting 9-decimal-digit number in the designated place. Any overflow occurring will be recorded in the result.

2. *Subtract decimally:* Similar to add decimally.

3. *Multiply decimally and round:* This operation will take two 9-digit numbers, multiply them, round the result to nine significant decimal digits, and then store the result in the designated place.

4. *Multiply double length decimally:* This operation will take two 9-decimal-digit numbers, multiply them, and store the resulting 18 decimal digits in the specified two adjacent 9-decimal-digit words. Each half of the result is given the sign of the result itself.

5. *Divide decimally and round:* This operation will take two 9-decimal-digit numbers, divide one into the other, rounding the result to nine decimal places, and store the result in the designated place. The division operates as if the numbers were fractions. Any overflow occurring is indicated in the result.

6. *Divide and save remainder decimally:* This operation will take two 9-decimal-digit numbers, divide one into the other, and consider them as integers while doing so. The result is recorded as a 9-decimal-digit integer quotient and a 9-decimal-digit integer remainder.

7. *Scale factor decimally:* This operation will "normalize" a specified number by shifting it to the left the number of times necessary to bring the most significant digit up to the decimal point. The number of decimal shifts is recorded in a second word.

8. *Add binary:* Take two 36-bit numbers or instructions and add them binarily, storing the resulting 36-bit number in the designated place. Overflows are indicated in the result.

9. *Subtract binary:* Similar to add binary.

10. *Shift logical:* Takes an entire 42-bit word and shifts it, as specified, either to right or left any desired number of places. The result is stored in the designated place.

11. *Shift magnitude:* Similar to shift logical, except that only the magnitude portion, 36 bits, is affected.

Logical and Transfer Operations

1. *Extract:* Extracts designated bits or digits from a specified word and inserts them into some other specified word.

2. *Test magnitude:* Compares the absolute value or the magnitude of two designated words and, on the basis of that comparison, chooses one of two alternatives.

3. *Test algebraically:* Similar to test magnitude, except that the signs of the two designated words are also considered.

4. *Test bit:* Examines a designated word in selected bits. If ones occur in all of those bits, one choice is made; if any of those bits are not one, another choice is made.

5. *Test search:* Determines whether any tape unit is presently searching for data.

6. *Test switch:* Determines whether a specified switch on the control console has been thrown one way or another.

7. *Buffer load:* Transfers, en masse, a designated set of eight words from the main memory to an 8-word buffer register.

8. *Buffer out:* Transfers, en masse, the contents of the 8-word buffer register into a designated set of words in the main memory.

9. *Halt:* Puts the machine in a state ready to receive new instructions from operator.

Magnetic Tape

1. *Block search:* An instruction to a designated tape unit to search for a specified block address. The operation of searching is initiated by the computer and then continues independently.

2. *Read tape:* Reads from the magnetic tape into the main memory any specified number of words.

3. *Write tape:* Similarly reads from the main memory onto magnetic tape any specified number of words.

Input-Output Commands

1. *Fill:* Calls for input, from either the console Flexowriter or the high-speed photoelectric reader, of any designated number of words.

2. *Print:* Calls for output to either the console Flexowriter or the high-speed paper-tape punch of any designated number of words.

3. *Read card:* Calls for input from punched cards.

4. *Punch cards:* Calls for output to punched cards.

Certain aspects of this command listing are of some interest. The machine has been designed so that both decimal and binary operations are possible. Most arithmetic operations will, of course, be on decimal data. However, efficient use of storage space will frequently make it desirable to store decision information, identification data, and the like in a binary form. This machine deals conveniently with both types of information.

The searching operation has been increased in flexibility by allowing for a "back-up" mode. Upon command from the computer, the tape unit will back-up the magnetic tape, a block at a time. The actual operation is merely initiated by the computer and then continues independently.

The reading and writing operations with magnetic tape have been increased in flexibility by allowing any specified number of words to be read or written. This is accomplished by a single command which specifies the tape unit involved, the first word to be written, and the last word to be written; all else is automatic.

The decision operations have been increased in flexibility by use of the "test bit" command. This is extremely useful for determining the status of an item with respect to certain categorization. One particular specialization of this command is as a test for overflow.

The input-output has been increased in flexibility by allowing for high-speed paper-tape input and output. In addition, there is complete alpha-numeric input and output, and the representation used is such that programmed internal sorting of alpha-numeric data is possible.

Discussion

Milton Adams (Stanford Research Institute): Is the magnetic tape erased while backing up or is rewriting done on top of the old data?

R. M. Hayes: No, the information is not erased while the magnetic tape is backing up. The data are rewritten on top of the old data. There is no reason why the data would necessarily be erased. It might well be that you have recorded data, backed up, and want to read those data again to check the accuracy of what you have recorded. This is one way of controlling accuracy of information.

I might mention this, that when the magnetic-tape unit is at searching, the block searching speed is 90 inches per second. The reading and writing speed is 15 inches per second. The backup speed is also 15 inches per second.

Gilbert Clotar (American Optical Company): Would you give us some idea of the speed of operation of the *102-D* in terms of orders per second, such as plus, minus, times?

R. M. Hayes: The magnetic drum rotates at the rate of 40 drum revolutions per second. An add-type operation will normally take about half a drum revolution with relatively minimum access programming. If you have any bad distribution of the numbers, the access time for the data will therefore be added to that.

The multiply-type operation, such as the multiply or divide either round or double length, will take about a drum revolution and a half, or a little less than that.

William Miehle (Burroughs Corporation): Would you explain category search again?

R. M. Hayes: I would be happy to, and I will also explain the test bit commands, because they are very similar in their nature.

To give you an example which is perhaps foreign to many of you (as it was foreign to me), suppose that you are a bank and you have made loans and you want to determine whether a given loan is delinquent or not. When a payment is made you post the fact that that payment has been made. Actually you post it by cancelling out a one that occurs there and putting a zero. So you tell the machine at the end of the delinquency period, when you are going to send out your delinquent notices, "Search the magnetic tape for the first account which is delinquent." In other words, "Look for a one in this position where I should have erased it." If I did not erase it, then the account is delinquent. And so you specify the bit that you are searching for.

This search is actually over the first word in the block—actually the first two words in the block—so it will search down the tape until it finds an item which has a one in that position and then will halt and you are now ready to read that item in. In effect it does your decision operation for you—and independent of the computer itself.

The test bit command works in exactly the same way. You don't have to have a single bit; you can have a combination of bits and it will examine all of them simultaneously and if a one occurs in every position the tape unit will halt, ready to receive a read instruction, or with the test bit command the decision will be made one way or the other.

Mr. Casey (General Electric AGT): What have you found are the most valuable uses of the extract operation? Is it the 3-address instruction?

R. M. Hayes: As far as the uses of the extract operation are concerned, or even the most valuable ones, I am afraid that I have never evaluated the relative uses of the extract. It is certainly extremely useful in actual command flow in determining in an iterative operation, for example, where you ended the operation. It certainly is extremely necessary if you are packing information, as for example packing information on a drum. You will have to extract to obtain the relevant parts and be able to operate with them. It would be extremely hard to evaluate.

Is it a 3-address instruction? Yes. The second address contains the extractor. That is, you specify the locations that you are taking information from. The first word is the extractee, namely, the address from which you are taking information. You have 42 bits and you want to extract the fifth, sixth, and seventh, say, of a given word. You specify that word and you specify the fifth, sixth, and seventh bits. Then these are extracted into the third address into again the fifth, sixth, and seventh bit positions.

J. Belzer (Battelle Memorial Institute) and **R. J. Pfaff** (International Business Machines Corporation): How many *102-D*'s will be produced and, roughly, what would you quote as the cost of a minimum unit?

R. M. Hayes: As far as quoting the cost of the minimum unit is concerned, I will be glad to discuss that with anyone who is interested. I do not think this is the place to do that.

How many *102-D*'s will be produced? Approximately somewhere in the order of 20 will be produced.

The Marchant Computer System

G. B. GREENE

THE Marchant MINIAC system is a completely new approach to the automatic accounting system, or as it has been frequently termed, the automatic office. With no intention of entering into the current debate as to whether the automatic office has arrived, report of the progress of its development is submitted.

When Marchant determined to undertake the development of an automatic office system, it troubled us somewhat that we had the whole distance to go, that we had no equipment now in use which might be adapted to this system. It has turned out, however, that this was a blessing in disguise since it permitted the engineering, from the ground up, of a completely modern system without the need of compromising specifications to encompass such equipments. Fullest use of modern techniques was therefore afforded.

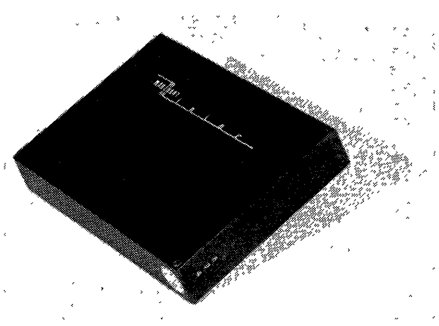


Fig. 1. Magnetic-tape work capsule

Customarily, when an equipment system is being described, one speaks of the principal central equipment, such as a high-speed electronic computer, then the principal peripheral equipment, and finally, the data source equipment in its turn. However, this paper will reverse this procedure and, in fact, go beyond and first discuss a novel data conveyance. Fig. 1 shows what is literally the "little black box," and is one of a number of varieties of magnetic-tape encapsulations (Fig. 2). The storage capacity is in the order of 400,000 characters, or 8 hours of typewriter-out time, all in a volume measuring 5 inches wide by $7^1/_4$ inches long by $1^1/_4$ inches thick, or about the size of an ordinary book, facilitating labeling, handling, storage, and shipping.

It plugs into any of the drive mechanisms which are incorporated in each of the equipments comprising this system. It speaks the same language as all of the equipments in the system, even to and including a uniform cell density. It is a single-track recording device using a self-clocking recording method. While its conveyance is normally transportation by messenger or by U. S. Mails, it can be reproduced by long-distance telephone or leased wire at a speed compatible with the band width of the line. For example, the 2,500-cycle bandwidth of the telephone system will reliably transmit this information at about 200 characters per second. It is literally a piece of the memory of the data-processing system which is sent to data sources for primary records, then returned for processing.

Our own 6-bit code is used, providing 42 alpha-numeric characters, 6 typewriter commands, 6 director digits which are required by various portions of the data-processing system, and 10 prohibited combinations, which constitute ambiguity check against the 10 decimal digits.

The tape package contains the necessary tape-handling equipment, the recording head, end-of-tape sensing means, electric plug-in connections, and a unique package-identification device, whose use will be dealt with in more detail in the following.

In the language of the accountant, chronologically organized lists of transactions of a given type are called "journals"; groups of accounts falling within a given category, such as receivables, payables, etc., are termed "ledgers." It is a salient point with the Marchant MINIAC system that a given ledger or a given journal is a particular tape package. It is equally salient that a package contains all of the data of like kind for a given accounting period, without regard to the portion of its capacity used.

The alpha-numeric input device is based upon a standard electric typewriter, one of several available makes which may be specified by the customer. The Marchant servo unit which converts the typewriter into a magnetic-tape-reading–magnetic-tape-writing typewriter, occupies space below the typewriter, and contains two independent magnetic-tape drives which will accept the tape package described (Fig.3). The obvious advantage of this arrangement is that the most complex part of this equipment is mass-produced for a very large market and is therefore priced most reasonably. It also happens that the high-wear parts of the typewriter system are confined to the inexpensive section. The typewriter may be replaced by a new or factory-rebuilt one when its useful life has been exhausted. It does not have to be returned to our factory for overhaul.

The unit control is achieved through an accessory unit which is of the same physical size as the tape capsule described in the foregoing and plugs into a typewriter tape drive in the same manner. However, as Fig. 4 illustrates, it is a magnetic card-reading–card-writing device, which later will be available for a single-track card, or for a 10-track card, selection of the track to be accomplished by the keyboard shown along the top of the package. The unit-control card is one of two forms (Figs. 5 and 6). A typical arrangement of this equipment in a location like, for example, the sales office, is shown in Fig. 3 in which a standard tape capsule is plugged into one of the two tape drives and the card attachment plugged into the other. In this example, the sales office will have two tub-files. One is an alphabetical listing of all existing customers. If the 10-channel card is used in this instance, the first track will have all information normally found in the heading of billing; another contains terms; another, credit information. Other tracks hold bought-by and shipped-to combinations, etc. A second tub-file will be a numeric arrangement of stock control or catalogue numbers, each card therein containing ten stock-control numbers, descriptions, unit prices, and the like. Coded stop on the card permits manual adjustment, such as insertion of quantity ordered, or the selection of the size or color in the description. Ordinarily an order received by telephone, over the desk, by mail, or by a salesman's

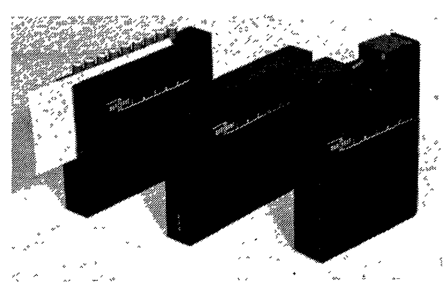


Fig. 2. Three varieties of Marchant MINIAC magnetic-tape encapsulations

G. B. GREENE is president, Marchant Research, Inc., Oakland, Calif.

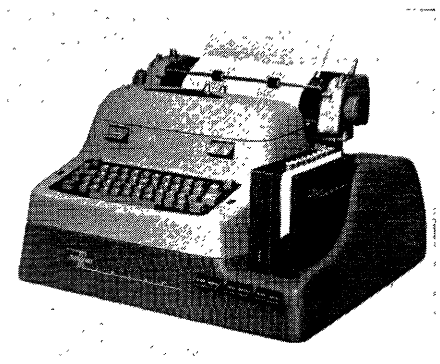


Fig. 3. Marchant servo unit with standard typewriter and two magnetic-tape capsules

notes will call for the selection of the customer card to be transcribed, along with manually inserted date, customer's purchase order number, and sales order number. Successively, items desired to be purchased by this customer are manually dropped into the card-reader slot and the track number chosen by the last digit of the stock control number. Coded stop permits manual quantity adjustment. The hard copy that results from this operation will be the customer's acknowledgment, the sales order, and a sales office copy. The work capsule that was plugged into the typewriter throughout this operation, is the sales journal and contains a record of all strokes of the typewriter, whether they are instituted by the card-reading attachment, or by manual keyboard. It will be seen that except for the insertion of date, customer's purchase order number, and specific quantities of items purchased, there has been no manual operation of the typewriter or any human file or catalogue reference for this transaction entry. In this use, the magnetic card contains format.

Tape Information Processor

Equipment will be provided which will do the sorting, collating, and all other logical operations. It must be borne in mind that this equipment is not a physical sorter of documents, but rather does its sorting by reading from one tape, forms a decision as to which tape to transcribe to, and then carries out that transcription. It is a continuous-flow processing device operating above 1,000 characters per second, every second. It has a flexible register length up to 120 characters. Its speeds for sorting and collating correspond to about 800 80-column cards per minute. It is capable of alphabetic as well as numeric sorting and collating.

The tape information processor may be installed and used as a sorter-collater without any computing facility, or it may be installed in conjunction with the MINIAC computer (Fig. 7). It is for this reason that one generally expects to find several tape information processors in a system to each MINIAC.

The unique identity-keying device incorporated in the tape capsule causes a tape drive associated with the MINIAC or tape information processor to react only to tape commands bearing the address of the tape capsule plugged into that drive. If a given accounting procedure calls for, say, four capsules of certain identities, it is sufficient to have these four capsules plugged into any four drives in the system without regard to which particular drive. Conversely, a tape command referring to a tape capsule which has not been plugged into some transport results in a malfunction and the calling of attention to the fact that a necessary tape capsule is not available to that machine. The operator may prepare for the next accounting procedure while the present procedure is being run, since he can plug the capsules pertaining to this next procedure into any transport with impunity. No time is lost in setting up for a procedure, since immediately upon completion of the last command of the given procedure, the operating plan is established for the next procedure; and when the plan is established, that procedure can commence immediately.

Again in the language of the accountant, at the close of an accounting period a process known as "posting" must be carried out; this requires every item in every journal to be entered as a debit in one ledger, a credit in another. This process is actually carried out by sorting the journal tape into a given order of account number corresponding to the order found in a given ledger, extending prices, then collating with the proper ledger tape. This tape is now re-sorted into another order to correspond with the opposite ledger, whereupon it is collated with that ledger.

It is important in this type of operation to have excellent controls in accordance with the accountant's use of the term. Running-column totals and cross-foot totals are carried out by MINIAC during the time of passage of one word from tape into the information processor and back onto tape again. In this regard, the information processor performs extractions from the long word representing the entire transaction. This extraction is assigned one or more 10-digit words in MINIAC as the running-total, or counter, depending upon the magnitude of the expected column-total. All arithmetic functions are carried out by MINIAC. Malfunction resulting from the failure of cross-foot totals to check is detected by MINIAC's own logics, subject to MINIAC's numeric check for both transfer and arithmetic.

The trial balance, in the language of the accountant, is carried out entirely within MINIAC. It will be seen that all of the recapitulation figures resulting from the posting of all of the ledgers in the accounting system are contained within the drum storage of the MINIAC computer, due to its dual function as computing element adjunct to the magnetic-tape processor during posting. The financial statement, balance sheet, profit and loss statements, etc., are, therefore, only a systematic read-out of the addresses in the internal memory of the MINIAC computer known to correspond with the entities contained in that statement. Format is supplied by a format-tape capsule by collation.

Billing results from a collation of the sorted sales journal, with the sorted shipping clerk's report to determine which of the items ordered by the customer have been shipped, back-ordered, and cancelled. This leads to a corrected sales tape, which contains the customer account number along with all lines of billing. Prices are now extended and summarized. Finally, collation with

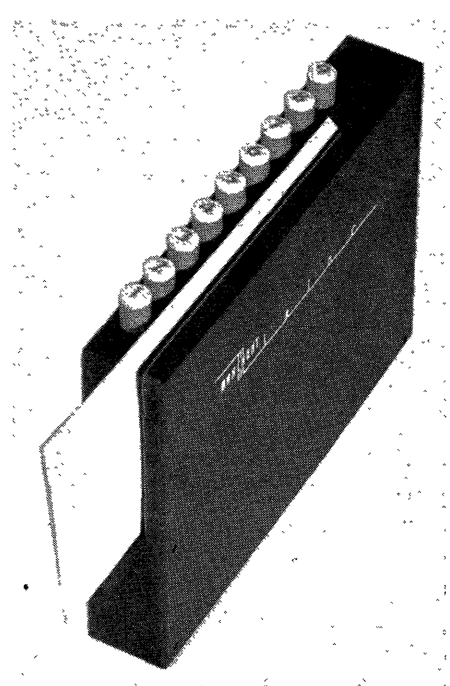


Fig. 4. Magnetic card-reading–card-writing capsule. Reads from or writes on any one of ten selected tracks on magnetic strip associated with card

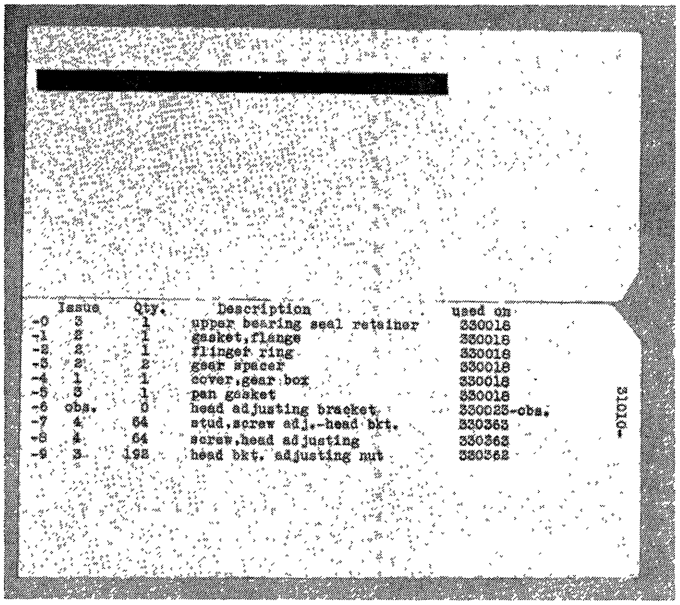


Fig. 5 (left). Card with magnetic strip sealed to back surface

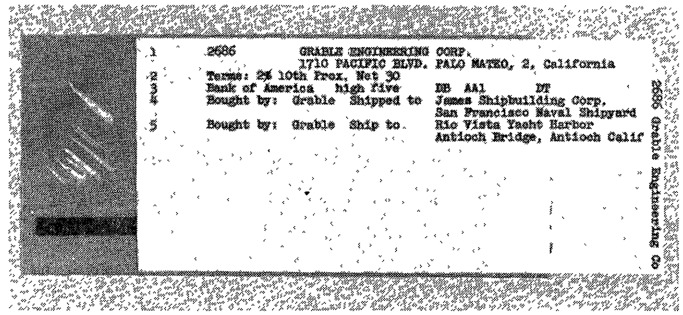


Fig. 6 (above). Card in transparent envelope with magnetic strip affixed to one side

customer's register tape for name and address results in the billing tape, ready to be inserted into the magnetic-tape-reading typewriter so as to produce the hard copy of billing.

Payroll has been avoided as an example in this paper, not because it is a less practicable procedure, but because of the greater complexity of its description. It should be pointed out however, that the automatic payroll procedure requires the following tapes:

1. The timekeeper's authorized pay report.
2. Employees' schedule, listing hourly rate, overtime and premium pay rules, income tax deduction data, etc.
3. Federal Old Age Benefits records.
4. Payroll register, listing payroll account number versus name.
5. Dollar table, listing numerical dollar versus alphabetically expressed dollar, etc.

There results, not one, but a number of tapes, namely:

1. Check printing tape.
2. Federal Old Age Benefits report.
3. State Unemployment Insurance report.
4. Federal Income Tax deduction report.
5. Cashier's report (of other deductions).
6. Possibly company pension plan report.

The column totals may be found in MINIAC drum memory.

We have discussed the use of magnetic-tape capsules as journals, and as ledgers. Other important uses are

1. Intermediate storage devices, such as pockets for sorting.
2. Registers, such as a customer's register, equating account number to customer's name.
3. Purchase-order registers listing all open purchase orders or dues.
4. Work tapes in hybrid systems employing card-to-tape, tape-to-card, or perforated-to-magnetic-tape transformation.
5. Archive storage for data storage required by law for very long periods.

It is easily shown that the real savings a low-labor-cost accounting system can bring about are not fully realized when a hybrid system is used. For example, a system involving a combination of cards and magnetic tape, perforated tape and magnetic tape, or manual and magnetic tape does not afford the ultimate in savings principally because the over-all system is at best only semiautomatic and still contains many labor costs. However, it is a mistake not to recognize that some business firms have excellent reasons for using hybrid systems. This requirement brings about a discussion of a class of equipment contained in the Marchant system. A card-to-tape and a tape-to-card translator operating at 150 cards per minute serves as the control unit for a standard punched-card reader and punch. A perforated-to-magnetic-tape translating device is also offered. This equipment operates at a maximum of 15 characters per second and brings about a full code translation during transfer. A second piece of equipment, a high-speed perforated tape-to-magnetic-tape transcriber is available for use where no translation is required, its speed being 1,000 characters per second. There is no magnetic-to-perforated transcriber for high-speed operation.

Data processing as applied to office precedures involves three phases:

1. *Communications*. Collecting, transporting, disseminating, filing.
2. *Organization*. Arranging, classifying, merging.
3. *Computation*. All arithmetic operations.

The view of many in this industry is that progress in the data-handling field

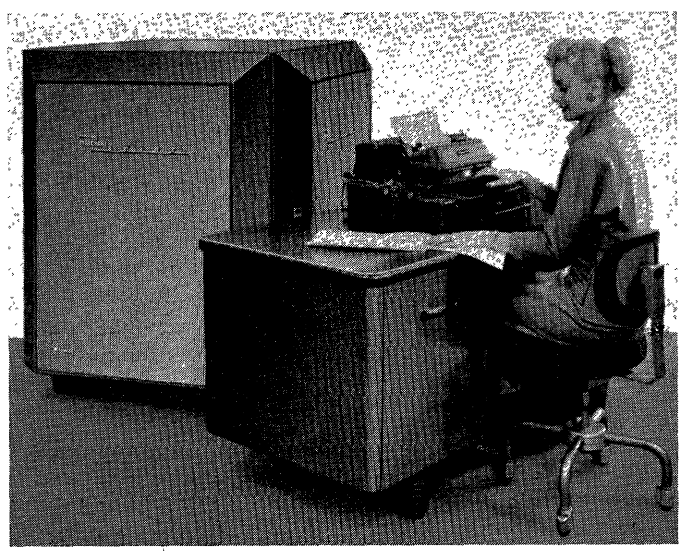


Fig. 7. Standard Marchant MINIAC computer with Flexowriter input-output. The magnetic tape system supplants the Flexowriter

has been retarded by the "giant-brain" school of thought. It emphasizes computation out of all proportion to its importance; it justifies extremely inefficient use of a general-purpose machine for data organization; it accepts the communication bottleneck as a burden to be borne. Indeed, the very inefficiency of data organization forces internal speeds which discourage avoidance of hybrid combination of data media, and imposes unnecessarily grievous programming problems.

Discussion

Salvatore Intagliata (Underwood Corporation): How do you arrange to have the units associated with the operation of the electric typewriter compatible with five different makes of machine?

G. B. Greene: If I understand this question correctly, the way we arrange this is by building a little unit which sets a typewriter, and by customer's specification we can assemble our mechanical matrix unit to provide a given identity to a given order along the typewriter. They all are key devices and all of them are basically able to be operated a key at a time by something pulling one key down as though it were a finger-tripped operating unit. By the same token, keyboard operation will depress this same element, so that it can be, and is, a 2-way street.

R. E. Schoenberg (New York Telephone Company): Did you say your tapes could be reproduced over ordinary talking telephone circuits of 2,500 cycles?

G. B. Greene: Yes.

D. N. Lee (RCA Service Company): What type of checking does the MINIAC employ?

G. B. Greene: The MINIAC has a numeric check system. We have our own code. It is a 6-point, or 6-hole, as some people call it, information system. Roughly it is 42 alpha-numeric characters, six punctuation and the like, and then some control and director digits that are required in other parts of the system, finally bringing us up to a total of 54. Then we have ten other entities that are the ambiguous, or prohibited combinations, of the ten decimal digits. Our people feel that ambiguity checking for alphabetics is not economically justified in most applications.

It is an ordinary ambiguity check system, but there is one small point which might be considered. We use an adder principle which gives us not only the sum digit but the proper ambiguity bit along with it, so the actual error detection is a matter of a transfer check. Each time any digit gets out onto transfer busses it is determined there for once and for all whether it was ambiguous or not.

W. L. Martin (Telecomputing Corportation): What kind of checking and marginal testing are used in MINIAC?

G. B. Greene: We have a very complete marginal checking, marginal manipulation system built into MINIAC, intended, however, to be done on a manual basis in concert with a diagnostic routine.

R. A. Rahenkamp (International Business Machines Corporation): Would you describe the type of card you mentioned? How is MINIAC controlled—by plugboard, tape, or stored program?

G. B. Greene: I overlooked mentioning the card because it can be anything from a calling card to an $8^1/_2$- by 11-inch card. You can think of this as anything that will fit your own system: a piece of paper, a piece of card. Its weight is not important. It has a magnetic track placed on the back of it.

You usually think of the face side of this card as containing the hard copy contained in the magnetic track of the card, and the actual length of this track is something like 4 inches and the capacity of one of these cards with the size of the card that we have always used runs about 120 alpha-numeric characters per track and ten tracks. Obviously there is not room enough for the hard copy of 1,200 characters on a card the size we are all accustomed to using, so this does mean that the card might need to be larger on occasion. We register only to one edge of the card and one end of the card, so its dimensions are not important.

In regard to how MINIAC is controlled, it is a stored program machine. Its program is loaded at 1,000 digits a second through any one of the tape drive devices provided.

L. A. McCabe (Campbell Soup Company); **O. D. Seeley** (Metropolitan Life Insurance Company); **J. W. Pontius** (General Electric Company); **John Mekota** (Raytheon Manufacturing Company); **E. F. Cooley** (Prudential Insurance Company): Please give information as to price and availability and whether or not the Marchant Sales organization has the equipment yet.

G. B. Greene: I don't know where to begin on this question. In the first place, the Marchant Sales organization does not have it yet.

The question of price is quite a problem. As you who are acquainted with systems realize, it is difficult to name a figure, that makes much sense unless speed or volume parameters are employed. In big systems you expect to find multiples of these equipments; it makes no sense to talk about one piece of equipment being worth this much or that it can be rented for that much. It may be of some help, though, to point out that the typewriter without tape capsules is around the $4,000 mark, the tape drive units are around $2,000, etc. It is not expensive equipment. It is intended for large production and is largely electromechanical.

Somebody wanted to know when it will be available. That is one of the most difficult questions yet, since he would mean available in quantity, and big system-wise, no doubt. There will be a few of these equipments built before the end of 1955, but not enough to be particularly significant to the industry. Our real production won't start until the first of 1956.

E. H. Friend (New York Life Insurance Company): In the system described, would you consider the tape information processer, rather than the computer, the real heart of the system?

G. B. Greene: I definitely believe that the tape processer is the real heart. I myself am one of a number of men in this industry who hold that the computer field, the data-processing field, has actually been harmed far more than it has been aided by the so-called "giant brain" school of thought. That school of thought of course glorifies the computation aspects of office procedures out of all proportion to its importance, and I believe that surely in this case the heart of the equipment, that without which you could not operate along this philosophy at all, is a much more rudimentary piece of equipment than a general-purpose computer.

N. C. Jochlin (University of Michigan): Do you have any magnetic-tape to paper-tape conversion units?

G. B. Greene: Yes, we can supply a tape-drive unit and capsules to co-operate with Flexowriter equipment for the conversion.

B. B. Jordan (Western Electric Company): Are your elements to be available separately for automatic production machine control?

G. B. Greene: That would be a policy matter that I would not be able to answer. I am quite certain in my own mind that the answer would be "yes," but I don't believe I am authorized to say so.

V. M. Wolontis (Bell Telephone Laboratories): When a tape capsule is attached to the machine, is the tape brought under the reading head without human intervention? If so, how is this accomplished?

G. B. Greene: Bear in mind you soon lose sight of the fact it is magnetic tape you are working with when all you see is a little black box. The actual tape transport, the machinery for moving tape off one reel around past the head and a capstan for driving it back on to another reel, are all contained in the package rather than in the machine it plugs into, and it is an instantaneous process. You can almost throw these things at their drive and they land in place. No tape can be seen or handled.

The assembly of these capsules requires a fixture; it isn't even recommended that a man try to repair his own.

Performance of TRADIC Transistor Digital Computer

J. H. FELKER

AT the Joint Computer Conference in February 1952,[1] I presented a paper on the transistor as a digital computer component. My talk at that time was based upon some early experience with transistors which led to the TRADIC program. This paper will present a progress report on what has come out of those early experiences. In 1951 we had a high-speed point-contact transistor.[2] We also had an amplifier using that transistor which could regenerate digital data at a pulse rate of 1 mc. At the request of the Air Force,* a program was started to lead to a transistor digital computer for an air-borne application. The application can probably best be characterized by saying that it required a series of extensive computations to be performed periodically upon less than 20 input numbers. The number of outputs required from the machine is even less than its inputs. The computer was to become a part of a military machine; that is, its inputs were not numbers from an electric typewriter or a punched tape but rather were from shafts and special dials. Its outputs similarly were not printed sheets of paper but were shaft positions and operational signals to other machinery. Thus, a large number of problems which we faced had to do with conversions from analogue data to digital data and conversely.†

In January 1954, we completed a breadboard computer which is known as the TRADIC phase-one computer. This machine multiplies or divides in less than 300 microseconds (μ sec) and adds or subtracts in 16 μ sec. It is a serial machine and employs a 16-digit word length. It runs on less than 100 watts of power. The main supply voltage is 8 volts. Auxiliary bias voltages of 2 and 6 volts are also required. The machine uses a group of electric delay lines for its internal memory. It has 18 such electric delay lines each of 1-word capacity. It has provision for 13 16-digit constants which are stored in toggle switches. The program is stored in a plug board.

Although the machine was finished in January of 1954, it was not apparent for several months who had finished whom. The TRADIC phase-one computer is a complex machine. It has 700 point-contact germanium transistors and 11,000 point-contact germanium diodes. There are 6,000 resistors and 4,000 capacitors and more than 1,000 transformers. During the first month of operation, the machine suffered from the equivalent of the usual diseases of childhood and as soon as the measles were past, the mumps set in. Although the symptoms of many diseases were present, it turned out that there was only one ailment, loose and improper connections. On May 1, the machine was put on a life test which is still continuing. During this test, the machine operates 24 hours a day, usually on error-detecting programs. Since we are proud of the machine, we frequently interrupt it to run special display programs and sometimes use it to produce useful output results. We find that it will run over periods of several days duration without errors of any kind. Since the machine is being operated in a building which is under continuous expansion and remodeling, we have been unable to obtain error-free runs of more than 8 days. The most valuable output of the machine and indeed one of the reasons for its existence is the reliability of figures that are emerging for transistors and diodes. The TRADIC computer is the first large-scale application of transistors and for that reason we are watching it very closely to see if we can get any clues concerning the reliability of the new solid-state art.

In the 5,000 hours that the machine has been operating, we have replaced four of the 700 transistors. One of these was replaced because the machine made transient errors. A second one was replaced as the result of a human failure as much understood as regretted. Two were replaced because the voltage margins of the machine were deteriorating. These transistors had not caused the machine to make errors but they were replaced because it had been decided that the machine should have a voltage margin on the 8-volt supply of greater than ± 0.7 volt. The four replacements have established a replacement rate of approximately 1/10 of 1 per cent per 1,000 hours. This means that a computer with 1,000 transistors in TRADIC circuits could be expected to require a transistor replacement about

J. H. FELKER is with Bell Telephone Laboratories, Inc., Whippany Laboratory, Whippany, N J.

* This work was done under Air Force Contract *AF33 (600)-21536*

† These problems were solved in a group led by J C. Lozier. The central computer was developed by a group under J. R. Harris. The work of both groups was substantially aided by S. Darlington and J. G Tryon of H. W. Bode's Mathematical Research Department.

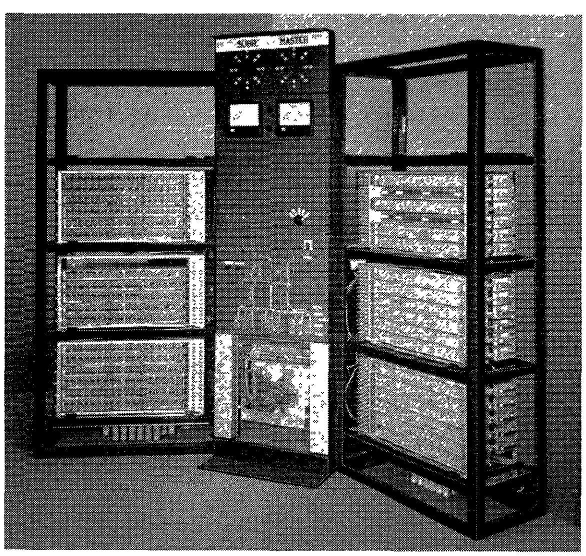


Fig. 1 (left). TRADIC phase-one computer

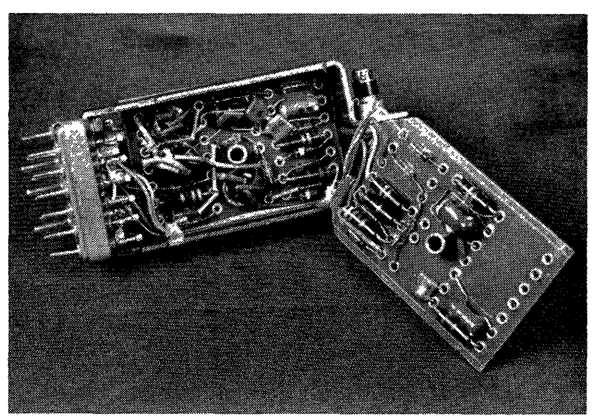


Fig. 2 (right). Package design (an opened view)

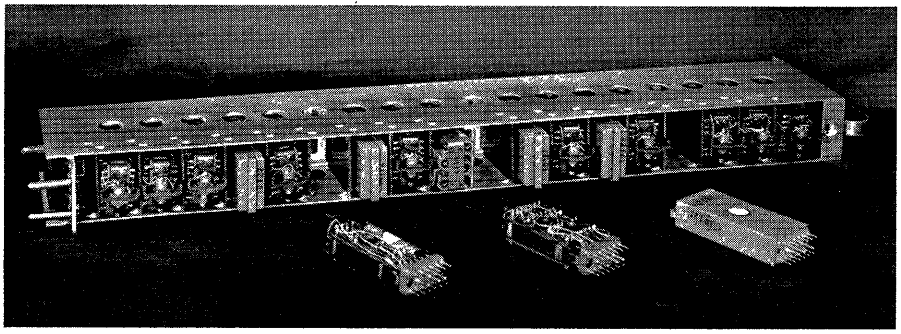


Fig. 3. Packages and mounting strip

once every 1,000 hours, or less frequently than once a month. We are frequently tempted to compute a half-life for transistors based upon this experience. Whenever we think of doing this we see how foolish it would be to make a prediction based upon what has happened to only four out of 700 transistors; yet consider how long we may have to wait before we have enough replacements to make a prediction of half-life with any certainty. It is questionable that the half-life is of practical interest since it appears to be comparable to the life of a human being.

What about the diodes? At the time we started the TRADIC program, we thought that we might use ten times as many diodes as transistors. We have used approximately 15 times as many diodes. We do not regret the use of the diodes. The aid that they give the transistor in performing its functions is no doubt an important contribution to the low replacement rate of the transistor. In the 5,000 hours of life we have had to replace six of our 11,000 diodes. Of these six replacements, two were the results of the same kind of human failure mentioned earlier and four replacements were required in order to keep the machine with good voltage margins. These four diodes were all used in the same spot in our regenerative amplifier circuit. This application makes a very stringent demand upon the reverse recovery time of the diode. Of course, in any future work with these circuits, we will obtain a diode better suited to the application. The diodes are hermetically sealed and the transistors are not. The replacement rate for the diodes is approximately 1/100 of 1 per cent per 1,000 hours.

Based upon newer devices we have seen, it is quite likely that device failure rates much lower than these will be observed. It is comforting and perhaps necessary to the ego of the circuit designer to blame the device when he explains failures. When quality transistors are available in large quantities, the weakest link in the reliability chain is likely to be the lack of thoroughness of the circuit designers. If failure rates of 1/10th of 1 per cent per 1,000 hours are to become common, the time constants of design and testing must be prolonged. A constant failure rate of 1/10th of 1 per cent, as indicated earlier, would predict a half-life of approximately 70 years. This means that years of testing and observation may be required to indicate the effects on reliability of design changes in devices and circuits. In one way we were fortunate when we worked with unreliable elements. We did not have to wait too many weeks before we found out whether or not a change had improved reliability or decreased it. Nature did not evolve the brain in a few million years. If we want to increase the complexity of the machines which we undertake, we should contemplate long programs of research and development.

TRADIC Phase-One Computer

The TRADIC phase-one computer is based upon a high-speed point-contact switching transistor which has been given many numbers in different mechanical versions. The cartridge version is known as the *1734*, the bead version is known as the *1760*, and the hermetically sealed version was known informally as the *1894* but is now officially referred to as the *2N67*. The *2N67* is now in production at the Western Electric Company. The phase-one computer, which used the *1734*, is not a small machine as can be seen from Fig. 1. There just was not time to investigate miniaturization.

Every transistor is used in the same fashion. That is, it is used as a regenerative amplifier. However, we associate the amplifier with a variety of logic and memory circuits. The machine is made out of the eight packages whose functions are listed in the following:

Basic TRADIC Packages
AND—4 terminal
OR—4 terminal
INHIBIT
MEMORY
Passive Delay
Active Short Delay
Integrating AND
Clock Filter

We have a physical embodiment of each of these functions, which can be recognized as conventional logic, memory, and delay functions. Fig. 2 shows the packages. Note the point-contact transistor at the top of the package, a great profusion of diodes, and three transformers. All of these parts are commercially available except the transformers which are a special design based upon miniature ferrite cup cores. However, transformers of the same performance in somewhat larger size could be procured commercially. We associated these packages with one another in strips as is seen on Fig. 3. These strips plug into the frames which make up the computer. To give an example of what size might be realized through miniaturization, Fig. 4 shows an experimental TRADIC package. One can put 400 of these packages and their connecting circuits in a cubic foot.

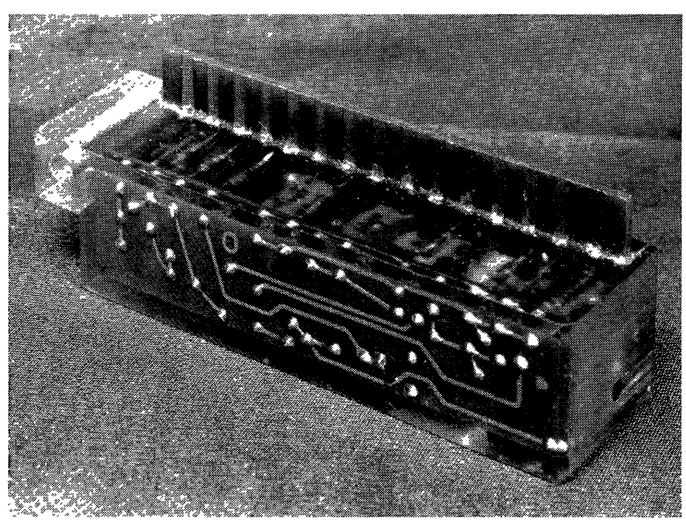


Fig. 4. An experimental package

Three years ago, I made certain predictions which I have reviewed before undertaking any new ones. At that time, I felt that 400 or 500 transistors might be put in a cubic foot, and this density has been achieved. Within the next 3 years it should become possible to put five times that many transistors and associated circuits in a cubic foot. This advance will come about not from using smaller transistors, but by realizing equivalent performance with fewer associated components. The TRADIC phase-one packages are complex circuits. There are approximately 40 parts per transistor. The package density will also be increased, of course, as we improve understanding of what we have sometimes called the microconnection problem. That is, the problem of making, say, 1,000,000 connections in a cubic foot. To do this we must surrender the convenience of 2-dimensional mounting of parts and make full use of all three dimensions. We must do this, of course, in the interconnecting wiring as well as within the packages. A real obstacle is that we do not have a 3-dimensional equivalent of the 2-dimensional printed circuit board.

In so far as operating power is concerned, things have gone about as predicted. Flip-flops have been built with junction devices which operate on less than 100 microwatts of power. Repetition rates have been increased substantially over the past few years. The TRADIC machine running at 1 mc is competitive with the majority of the vacuum-tube computers. Another group at Bell Telephone Laboratories has succeeded in building computer circuits with point contact transistors which run at 3 mc. These circuits have been described in the literature by J. H. Vogelsong.[3] The tetrode[4] as made by R. L. Wallace of the Laboratories has proved to be capable of regenerating digital data at even higher rates. Both Wallace's tetrode and Early's *p-n-i-p* transistor[5] make operation at 10 mc appear realizable in the next few years.

A major result which will soon be achieved is the combination of transistors and magnetic cores. About a year ago we were able to drive a 64-stage shift register with junction transistors at a $62^1/_2$-kc rate. The germanium alloy transistors used developed 200-milliampere pulses. Since that time we have been using transistors to drive small matrices of cores and we expect to use transistors to drive large arrays of cores in the near future. Regarding predictions about the reliability which will be obtained it can be said that replacement rates as low as 1/100th of 1 per cent per 1,000 hours can be achieved with hermetically sealed junction transistors running at rates of about 100 kc. However, when such rates are claimed, it will take many years to establish them.

References

1. The Transistor as a Digital Computer Component, **J. H. Felker.** *Review of Electronic Digital Computers*, Joint AIEE-IRE Computer Conference, New York, N. Y., February 1952.

2. Regenerative Amplifier for Digital Computer Applications, **J. H. Felker.** *Proceedings*, Institute of Radio Engineers, New York, N. Y., vol. 40, no. 11, November 1952, p. 1584.

3. A Transistor Pulse Amplifier Using External Regeneration, **J. H. Vogelsong.** *Ibid.*, vol. 41, no. 10, October 1953, p. 1444.

4. A Junction Transistor Tetrode for High-Frequency Use, **R. L. Wallace, L. G. Schimpf, E. Dickten.** *Ibid.*, vol. 40, November 1952, pp. 1395–1400.

5. P-N-I-P and N-P-I-N Junction Transistor Triodes, **J. M. Early.** *The Bell System Technical Journal*, New York, N Y., vol. 23, no. 3, May 1954, p. 517.

Discussion

E. S. McCollister (ElectroData Corporation): In view of the high component reliability figures you have given, do you think there will be a trend toward more or less checking units in the machine?

J. H. Felker: It is my personal opinion that we will not see a trend toward more checking in a machine; the kind of checking I am speaking of is the checking of the performance of the electronic components. Some of my most intimate associates disagree violently with me on this subject and there is room for difference of opinion, but I think we can build machines reliable enough that they just won't need the checking. If we increase the complexity of our machines by the factor of 100, and talk about 500,000 transistors, I think we will have to build in checking, not to obtain reliability of computation, but as a feature of diagnostic programming in order to find out where the fault lies on the rare occasion when something does go wrong. That is the kind of checking we will need.

David Noble (Remington Rand, Inc.): What about the temperature limits for operating a transistor computer and what precautions have been taken for temperature compensation?

W. D. Rowe (Westinghouse Electric Corporation): What is the range of temperature and humidity over which the computer was operated successfully?

J. H. Felker: This computer has been operating in an air-conditioned laboratory since June 1954. From January to June it was operated in a room that was just an ordinary laboratory room. The temperatures went up and down. I know that if the temperature went to 110 F the machine would begin to tell white lies.

We do not have much temperature margin in the machine. It is possible to design transistor circuits today with temperature margin but as a rule, when this is done, some aspect of performance is sacrificed. You give a little and you take a little there.

We all look forward to having silicon transistors when we won't have this temperature limitation, but it is a very real limitation today. However, I have not seen a computer in many years which didn't have an air-conditioning unit attached to it.

H. Freeman (Sperry Gyroscope Company); **D. F. Albanese** (Federal Telecommunication Laboratories): Does the heat of the encapsulation process affect the transistors and diodes?

J. H. Felker: Well, one thing about temperature is that we were not quite as naive as I may have indicated. Before we put this machine on the air we took all the packages out and cycled them over a temperature range. We wanted to find out if we had any weak sisters in there, and I suppose we eliminated about 5 per cent that way. It was not a very severe temperature cycle, but it did last for 24 hours.

Does the heat of encapsulation process affect the transistors and diodes? It certainly affects some of the transistors. You know the old story about the Indians who threw their babies into the water to teach them to swim, and those who could swim survived. I think the heat of encapsulation has a beneficial aspect like that, too, as long as you don't have any humanitarian ideas about transistors.

It does kill off some of them, but I think it fortunate to be rid of those before you place a machine on the air.

E. Kinnen (Westinghouse Electric Corporation); **R. J. Pfaff** (International Business Machines Corporation); **E. Sard** (Airborne Instruments Laboratory); **R. T. Prince** (Armour Research Foundation): Please compare the junction transistors and point contact transistors.

J. H. Felker: The point contact transistor has been the fastest transistor we have had to work with. It also has been the transistor we had in quantity and has been reliable. The first junction transistors were not very reliable. That situation has improved enormously in the last year.

Junction transistors that are available are not as fast as point contact transistors, but I think it became clear to us about a year ago that the future is with junction transistors rather than point contacts. There were two things that convinced us of this. One was that physicists aren't interested in the point contact. They don't understand and won't work on the point contact transistor so it will never be improved.

The junction device obeys the mathematics that they understand, and this is a very real thing. If the best effort goes on a particular device, that device will get better.

The other thing which is equally significant is that the junction transistor is now becoming faster than the point contact. 10-megacycle regeneration of pulses can be achieved with a few of Wallace's better tetrodes, and we haven't been able to do that with point contacts. The *p-n-i-p* transistors will operate at higher frequencies than point contact transistors.

There are only two things wrong with the tetrode and *p-n-i-p:* We haven't any, or rather we don't have enough to build a computer. We have small samples.

J. L. Hill (ERA Division, Remington Rand, Inc.): Will you identify the 200-milliampere transistor and state your estimate of the date of its availability outside military channels?

J. H. Felker: We have had some internal codes which I would not like to identify since they would cause hard feelings back home; people would be asking for them and not getting them. We get them in small quantities—10, 15, 20—and as soon as we can get 100 or so we will build a core memory with them. But I believe that people haven't really done a good job of looking around at other suppliers. There has been a tendency to wait for the big companies to come out with good transistors. There are a number of small places making germanium-alloy transistors and I think if someone were willing to put four or five engineers to the task of surveying the output of these people they would find available transistors that would switch 200 mils. Maybe the designers don't know it yet, but I think that it is within the capability of a number of alloy transistors which they are putting out now.

Richard Walkling (Remington Rand, Inc.): How do you find the bad components when they do not cause errors? Do you have a marginal check?

J. H. Felker: We have very extensive marginal checking. In Fig. 3 of the paper I showed the strips into which we plug packages, about 20 packages to a strip. We have each one of those strips plugged into a plug in the frame, and associated with the wiring for each strip is a group of switches which permit us to switch that strip from the regulated power supply, which, incidentally, is a transistor-regulated power supply, to unregulated power supplies. We can perform marginal checking on a frame basis on the whole computer, or on individual strips, and we do this frequently. We would not build this many switches into another computer, but this is a research model and we wanted to get as much data as possible.

We are a firm believer in marginal checking. As a matter of fact, I think people ought to marginal check their machines continuously instead of having regulated power supplies. We should have a power supply with a sine-wave variation of the voltage on it, particularly if it is important to know when the machine is going wrong. When I have wished I had this was when I have known that in 3 hours I was going to have to demonstrate the machine. It would be very nice if I had known that the entire week before this machine had been working continuously with a 20-per-cent or 30-per-cent variation in the power supplies.

Marshall Middleton, Jr. (Westinghouse Electric Company): Is the mortality rate of transistors and diodes a function of the number of times the computer is turned on and off?

Frank Wagner (North American Aviation): Why do life tests take so long? How about designing some accelerated service tests?

J. H. Felker: Our computer is on all the time. We are trying to pile up hours of operation and we never turn it off deliberately, although nature sometimes does that for us. We lose power supply with hurricanes and construction, etc., but we have no evidence that turning it off and on should hurt the transistors. I think the low power we have used is significant in reliability.

Why do life tests take so long? I think if you have good devices life tests have to take a long time. It is very good to give accelerated life tests, but I don't know what they prove except that the devices stand the accelerated life test. Nobody has written a theory yet that I know of which connects various acceleration features to the kind of life in which we are really interested.

Our transformer people perform accelerated life tests and it is extremely important that they do so, but it isn't a complete substitute for having the machine run. To me there is no substitute for that.

K. Enslein (University of Rochester): What type of diodes did you use? Who are the manufacturers and what are the type numbers?

J. H. Felker: These diodes were obtained from the Hughes organization. They are excellent diodes, needless to say. We were delighted when they came out. Western does not make a diode that would do our job.

There must be five or six different types used, and I think one reason we have the reliability we have is that the Hughes people were willing and anxious to make diodes to our specifications. I don't know if any of these diodes have Army-Navy designations yet or have commercial numbers, but when we were building the machine we certainly got them in large quantities and I would suspect that someone else could.

Incidentally, we have put out reports which are available through military channels; our Air Force contract is *AF 33(600)-21536*.

D. J. Niehaus (Bendix Aviation Corporation): What voltage tolerances were needed on the diodes?

J. H. Felker: We always tried to use the diode that had the highest back-voltage rating possible and then use it with as low a voltage as possible. Most of our diodes don't stand back voltages of more than a volt or two, except the diode in the clock circuit, which has to withstand 8 volts peak. This is the way to use diodes. There is one exception however. We had the four diodes that failed. They were all in one place. We made an error of judgment there. We needed a diode that would recover very rapidly after a forward transient and we wanted to recover to a back current of less than 100 microamperes in 1/10 microsecond. Hughes did a good job in supplying us with a diode to do it. There must be 700 or 800 of them in the machine, but the four, out of a total of 11,000, that went wrong were all of that same type. The back voltage at which that diode is tested is only 3 volts, so it was rather a marginal thing to do, even though in our circuit it receives a back voltage of only a volt. We might have been smarter to have let the diode have a poorer recovery time in order to get a greater back-voltage margin and take a loss in margin elsewhere in the machine. But those are the things that are difficult to know in advance.

L. M. Schmidt (International Business Machines Corporation): Was an attempt made to utilize unsoldered wrapped connections in TRADIC?

J. H. Felker: There was no attempt to do so. I think that the wrapped connection described in the *Bell System Technical Journal* and probably in Western Electric advertisements is better than the soldered connection, but it was not a technique available for us. We are all interested in printed circuit cables, and I explained about the 2-dimensional limitation. It may be the wrapped wire connector that will give us the microconnection that we are looking for, because wires are remarkably efficient. You can pack them in three dimensions.

Application of the Burroughs E101 Computer

ALEX ORDEN

THE Burroughs *E101* digital computer is designed to fill the gap between standard desk calculators and large-scale digital computers. The machine combines flexibility of operator access and judgment, as in use of desk calculators, with automaticity on repetitive routines, which is the great virtue of the large-scale digital computers. Use of the machine is expected to be generally on a decentralized basis, in individual offices and laboratories, rather than in a centralized computing machine installation. In many situations decentralizations of computations to the source of the data should more than compensate for the greater internal speed characteristic of large-scale computers.

The *E101* can be used effectively on problems of moderate size in the same automatic fashion as a large-scale digital computer. On the other hand, it can be used as a "super desk calculator," that is, in a manner similar to a desk calculator, but with extension of the operations which can be carried out by the machine, from the basic arithmetic operations, to such operations as square root, trigonometric functions, standard deviation, etc. Finally, the combined approach, the handling of computations in a manner which involves a mixture of manual intervention and automatic sequencing, will be presented in this paper as an approach to computation which warrants a great deal of attention.

The sections which follow present:

1. A brief description of the computer in order to highlight its relation to other machines and provide a basis for the discussion of methods of application. (Literature available from the Burroughs Corporation provides detailed information on characteristics, programming, and operation.)
2. Methods of application of the *E101*, primarily with reference to engineering and scientific computation. The paper is not intended as a review of the general scope of applications, but specific topics are discussed to illustrate the general approach.
3. Business applications.
4. Accessories to the basic machine.

Description of the E101

A prototype of the *E101* is shown in Fig. 1. Seen externally, it is the size of an office desk, has a numeric columnar keyboard input like a standard desk calculator, prints up to 12 digits at a time, at a rate of two words per second (that is, a maximum printing rate of 24 digits per second), and has the computation program stored in pinboards, which are located on the desk top to the right of the keyboard. Internally, the machine contains a magnetic drum of 100 registers, a 3-kw tubeless power supply, and electronic circuitry that involves the use of 163 vacuum tubes and 1,500 diodes.

The most novel construction feature is the pinboard programming unit. This involves, as distinct from most large-scale computers, storage of instructions separately from data. Instructions are, however, expressed in a single-address form of the type which is familiar in the large-scale digital computers, where instructions and data are stored in a single internal memory. For example, the instruction to add the contents of magnetic drum register *29* to the contents of the accumulator is written in the form, +29. Such instructions are entered in the pinboard in the manner shown in Fig. 2. The pinboard has 16 lines, each of which holds one single address instruction. The three characters in +29 require the entry of three pins into a line, and these three pins are indicated by black dots in the top line of the illustration. One pin has been entered in the + column, as shown at the top of the pinboard, one pin in the 2-column in the set of columns that are used for the 10's digit, and one pin in the 9-column in the set of columns that are used for the units digit of the address. Eight such pinboards, each of 16-step capacity, can be mounted in the pinboard panel, giving 128 single-address steps. Transfers to subroutines are handled by instructions such as U 4 12, meaning unconditional transfer to pinboard *4* step 12, as illustrated on a lower line in Fig. 2.

Each of the eight pinboards can be removed and reinserted or replaced easily.

The traditional block diagram for digital computers showing boxes for input, output, memory, arithmetic, and control is familiar to anyone who has dealt with large-scale digital computers; it is shown in Fig. 3 for subsequent comparison with the *E101*. The dashed-line block at the top of Fig. 3, "keyboard input for supervisory control," is usually omitted or considered to be covered by the main input block, but is pertinent here in relation to the *E101*.

This diagram serves well as a functional representation of most general-purpose digital computers. If desired, the input, output, and memory blocks can be shown as several blocks in those machines that contain more than one type of input-output or memory, but the basic character of the schematic remains the same.

By comparison, a functional block diagram for the *E101* is as shown in Fig. 4. The keyboard is the main data input and the printer alone is the output. Coded input-output, shown in dashed lines, will be accessories to the basic machine. The instructions, in the form of pinboards, are indicated as an input because of their removability and interchangeability.

Finally, with regard to the general rela-

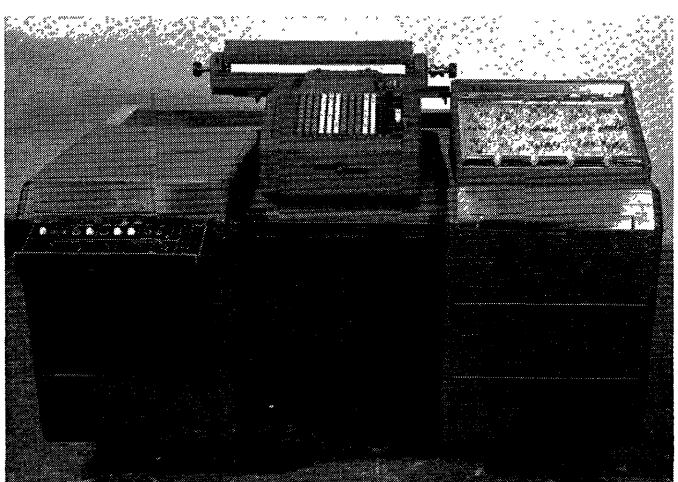


Fig. 1. The E101 digital computer

ALEX ORDEN is with the Burroughs Corporation, Research Center, Paoli, Pa.

tion of the *E101* to other computers, Fig. 5 compares the relative speed and cost of the main existing classes of digital computer. Computers other than the *E101* are divided into four classes: desk calculators; general-purpose punched-card computers such as the International Business Machines Corporation CPC or Remington Rand *409;* magnetic-drum computers; and electronic-memory computers. The majority of the computers under discussion are what is meant here by the magnetic-drum class, while the million-dollar machines with higher-speed memory are what is meant by electronic-memory computers. The computation speed typical of each class and of the *E101*, relative to a desk calculator, is indicated on a logarithmic scale, and above each speed line is a line indicating typical annual operating cost per year including personnel. The need for considering personnel cost, as well as machine cost, is obvious, particularly in the case of the desk calculator where the cost of a machine is minor, but the salary of an operator is, say, $5,000 per year. The indications are that relative to use of desk calculators, the *E101* at a cost increase by a factor of three, can generally increase computation output by a factor of ten. This diagram, no doubt, contains some controversial figures, but they are presented here only as the author's general estimates for the sake of perspective on the *E101*.

Methods of Application

Burroughs literature on the *E101* has stressed the point that the *E101* bridges the gap between large-scale digital computers on the one hand and use of ordinary desk calculators on the other. The main objective of this paper is to explain what is implied here by "bridging the gap."

A first, but rather negative way of explaining what is meant by the phrase has been implicit in some of the foregoing remarks. It was suggested that the general-purpose computers which center in a magnetic drum, and which are called "small computers," are functionally identical with the large computers. It is not common to speak of these magnetic-drum machines as bridging the gap between desk calculators and large digital computers. Of course, they do tend to bridge the gap in terms of cost, but in terms of method of operation and application they are practically the same as the large computers.

The *E101*, as a machine with direct keyboard input and considerable emphasis on

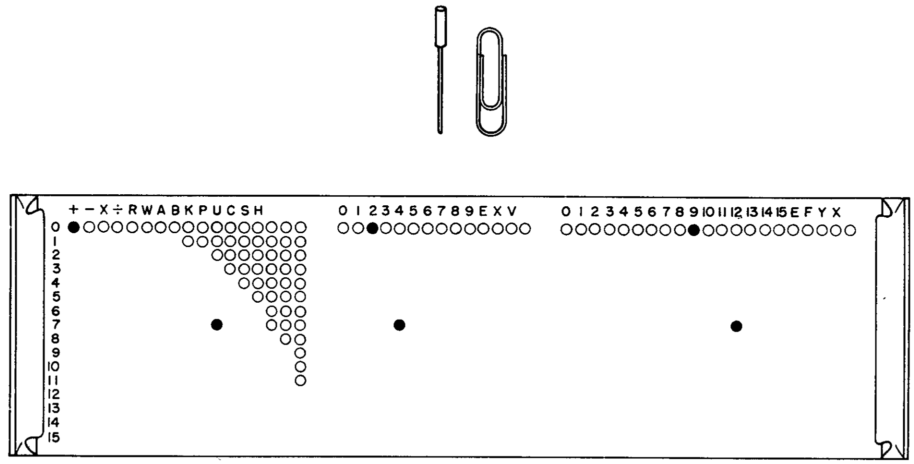


Fig. 2. E101 pinboard

operator participation in the computations does, at least, have more of a gap-bridging character between desk calculators and large computers than the general class of magnetic-drum machines. Incidentally, such remarks are not intended to discount the importance of the general type of magnetic-drum machines. On the contrary, that class of machine is clearly of great importance in the computer field.

The foregoing approach, a negative one, was to suggest that other moderate-speed computers do not fill the gap functionally between desk calculators and large computers but that the *E101* does. For an explanation on the positive side, one can start by raising the question: What would be the proper approach in bridging the gap between automatic and hand computations? An initial answer to that might well be to build a machine that would permit either method of operation. Of course, if such a machine were to overemphasize either side, the automatic or the manual, its use would tend to drift over to that side, and the machine would not bridge the gap.

The *E101* does provide both methods of computation, automatic as in a large-scale computer, and manual as in the use of a desk calculator. The real question will be: Why are both needed in a single machine? But before proceeding to that, the application of the computer, first as an automatic machine, and second as a manual machine, will be described.

As an automatic machine, the *E101* functions in the same way as any of the general-purpose computers, except that in the basic model the data are entered directly through a keyboard rather than by some form of recording. In other words, once the data are entered, the machine proceeds entirely automatically. The size of technical computation problem for which the machine is suitable for automatic processing can be gauged by the following two examples.

1. Solution of systems of differential equations whose complexity can go up to the general level of the equations shown in the

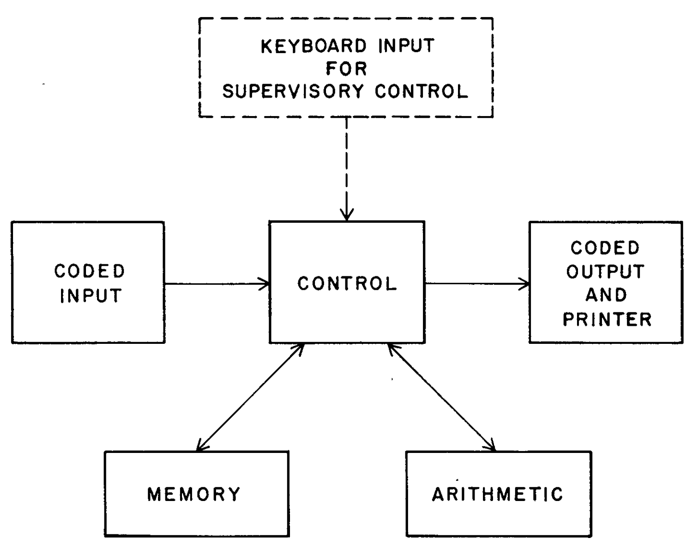


Fig. 3. Traditional digital computer block diagram

following. For this system of equations, the machine would automatically print a set of values of the unknowns and their first derivatives at successive discrete time values at the rate of one set of values every 25 seconds. Differential equations up to this level of complexity are, of course, adequate for many classical aspects of engineering, chemistry, and physics, but are not adequate for the complete solution of guided-missile and aircraft-motion problems. Differential equation suitable for the *E101* are

$$\ddot{x}-2y=x-(1-\mu)\frac{x-x_1}{r_1^3}-\mu\frac{x-x_2}{r_2^3}$$

$$\ddot{y}-2\dot{x}=y-(1-\mu)\frac{y}{r_1^3}-\mu\frac{y}{r_2^3}$$

$$\ddot{z}=-(1-\mu)\frac{z}{r_1^3}-\mu\frac{z}{r_2^3}$$

$$r_1=\sqrt{(x-x_1)^2+y^2+z^2}$$

$$r=\sqrt{(x-x_2)^2+y^2+z^2}$$

x_1, x_2, and μ are constants

2. A typical data reduction problem that the *E101* could handle automatically in the same manner as any other digital computer, except for data insertion, is shown in Fig. 7. This problem is the data reduction involved in wind-tunnel testing of aircraft models. Each set of data would be run on the *E101* in about 2 minutes.

Input Data: $\delta_1 \ldots \delta_6$, α^1, ρ, M, ψ
Printed Results: $(C_Y)_3$, $(C_L)_4$, $(C_L)_5$, α_6, $(C_D)_6$, $(C_M)_6$, $(C_N)_3$

Balance Interaction

$$\overline{NF}_1=a_1\delta_1+a_2\delta_2+a_3\delta_3+a_4\delta_4+a_5\delta_5+a_6\delta_6$$
$$\overline{AF}_1=b_1\delta_1+b_2\delta_2+\ldots \qquad +b_6\delta_6$$
$$Y_1=c_1\delta_1+\ldots \qquad +c_6\delta_6$$
$$L_1=d_1\delta_1+\ldots \qquad +d_6\delta_6$$
$$M_1=e_1\delta_1+\ldots \qquad +e_6\delta_6$$
$$N_1=f_1\delta_1+\ldots \qquad +f_6\delta_6$$

Tares

$$\overline{NF}_2=\overline{NF}_1+(1-\cos\alpha^1)W$$
$$\overline{AF}_2=\overline{AF}_1+W\sin\alpha^1$$
$$M_2=M_1-(1-\cos\alpha^1)Wl_1+\overline{NF}_2l_2$$
$$N_2=N_1+Y_1l_3$$
$$q=(\gamma/2)\rho M_2$$

Coefficients

$$(C_{NF})_3=\overline{NF}_2/qs \qquad (C_L)_3=\frac{L_1}{qsb}$$
$$(C_{AF})_3=\overline{AF}_2/qs \qquad (C_M)_3=M_2/qsc$$
$$(C_Y)_3=Y_1/qs \qquad (C_N)_3=N_2/qsb$$

Body Axes to Wind Axes

$$(C_x)_4=(C_{AF})_3\cos\alpha^1+(C_{NF})_3\sin\alpha^1$$
$$(C_z)_4=(C_{NF})_3\cos\alpha^1-(C_{AF})_3\sin\alpha^1$$
$$(C_L)_4=(C_L)_3\cos\alpha^1+(C_N)_3\sin\alpha^1$$

Wind Axes to Stability Axes

$$(C_L)_5=-(C_z)_4$$
$$(C_D)_5=-(C_x)_4\cos\psi+(C_Y)_3\sin\psi$$

Tunnel Wall Effects

$$\alpha_6=\alpha^1+K_1(C_L)_5$$
$$(C_D)_6=(C_D)_5+K_2(C_L^2)_5+\sin\beta(C_L)_5+K_3$$
$$(C_M)_6=(C_M)_3+K_4(C_L)_5$$

The purpose of these examples is to

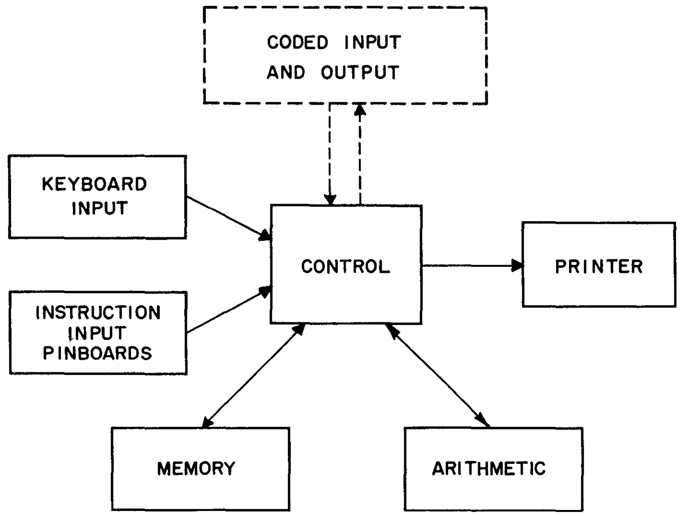


Fig. 4. E101 functional block diagram

indicate up to what level of complexity the *E101* permits automatic solution. The problems indicated are moderately complex, and indicate coverage of an appreciable fraction of the computation problems that arise in engineering. Larger problems are, of course, fairly plentiful, and on the *E101* would require breaking the computations up into several pieces to be handled separately.

Turning to use of the *E101* as a manual computer, it's so designed that it can operate effectively in a manner which might be called the super desk calculator. The pinboards can be set up in general-purpose form for providing such operations as square root, tangent, logarithms, product or quotient of complex numbers, cross product of vectors, and so on. After entry of the numbers in the keyboard, touching one of the so-called "special start buttons" would cause the machine to carry out the desired operations and print the results. Used in this fashion, the machine would be considerably more powerful than a desk calculator, but the main program of computations would be determined entirely by the operator.

Now, why bridge the gap between the automatic and manual approaches? This is the basic question toward which this paper is directed.

There is a large amount of technical computation in which two factors are present:

1. The need for engineering judgment during the computations.

2. A large volume of computational drudgery.

The use of desk calculators allows the operator to use his judgment at every step, but does not take care of the drudgery. The use of large-scale computers, or the small computers modeled after them, eliminates the drudgery but does not provide for the use of judgment. Despite the fact that general-purpose computers are programmed to branch out into a number of alternative computational subroutines, it is not correct to say that any appreciable amount of judgment has been programmed into the computations. The choices made automatically during the computation are essentially routine, and are more accurately described as automatic housekeeping than as judgment. Basically, the program for a given problem on a digital computer is a large mechanized subroutine in a problem in which the main program is not mechanized at all.

The solution of engineering-design problems on large digital computers is likely to involve automatic computation in order to provide performance characteristics of some device or structure many of whose design parameters can be varied. A thoroughly normal situation might be that there are ten design parameters to be studied and that it would be nice to try ten values for each of these parameters. This means 10^{10} cases to be computed. If a large digital computer could run through each case in 5 seconds, it would take about 1,600 years to do the 10^{10} computations, all, of course, without human intervention, except for machine maintenance.

There is a gap, then, between large computers that offer elimination of computation drudgery, but no human intervention for use of judgment, and desk calculators that allow judgment at every step but leave in all the drudgery. The design of the *E101* is based on the opinion that this gap is not a vague no-man's land for the computer field, but rather, that the target is clear-cut: to provide a machine which permits easy access to a variety of routines, while leaving judgment as to the course of the computation in human hands. The routines needed for a specific problem can, of course, be special ones, programmed for that problem, rather than general-purpose operations, such as square root, logarithms, etc., which were discussed earlier. As a first attack on this target, the *E101* offers easy access to routines of relatively small or moderate complexity.

The central objective in the *E101*, then, is to provide maximum simplicity in access to a variety of routines. The main ways in which the design meets this objective are:

1. The first stage of access is getting routines programmed and debugged. As indicated earlier in this paper in the description of pinboard programming, the program is in a very straight-forward language, and is entered on pinboards in a form which is practically as simple as writing instructions down on paper. In debugging routines, the pinboards can be inserted in the machine and run through step by step, with print-out at any step desired. If any error is found in the program, it is usually a matter of changing a few pin positions to correct the trouble The routine is then ready for use.

2. Access to any routine during a computation is as follows: If the pinboard panels have sufficient capacity for all of the routines required, access to any one of them can be obtained by touching a control button. If additional routines are required, pinboards can be replaced easily. Of course, if the character or size of the problem is such that pinboard replacements would have to be very frequent, the procedure becomes awkward and probably unsatisfactory.

3. The data for alternative routines can be entered directly in a keyboard that has the features normally associated with a keyboard calculator.

A reasonably careful presentation of a problem in which the *E101* would be used to good effect along the lines under discussion would require considerable detail, and is consequently not included here. A partial illustration will be given by discussing a standard problem, the solution of sets of simultaneous linear equations.

The *E101* with 100 words of memory in the base model appears at first to be limited to nine equations in nine unknowns, which would involve storage of 81 coefficients in the equations and nine numbers on the right-hand side of the equations. However, even in large computers with 1,000 words or more of fast memory, the solution of large systems of equations is often handled by breaking the set of coefficients into small square blocks, that is, in mathematical terms, by partitioning of matrices. In a well-known UNIVAC code, for example, the coefficients are handled in 10 by 10 blocks, regardless of the number of equations to be solved. The interrelation between blocks of data in a large computer is programmed so that interrelations in the handling of successive blocks is carried out automatically, but this requires a good deal of programming. With the *E101* one can work with 9 by 9 blocks and have the operator guide the combining and sequencing of the blocks. The amount of coding for the *E101* is probably about one-third of that for a large computer, while the basic drudgery of the computation is still essentially taken care of by the machine. This simple example does not illustrate a case in which judgment is required during the computation, but it does indicate how a substantial amount of programming for a large computer can be replaced by a small amount of human guidance of the computations.

Business Applications

The computer-design philosophy discussed applies, in a broad way, to engineering computation. The greatest impact of the *E101* in business computation is likely to be in analytical aspects of cost accounting, and in related business analyses such as those of operations research. In the latter case, the general approach to management problems is often similar to the methods of science and engineering.

Some specific cases of straightforward business applications which have been analyzed for the *E101* involve cost analysis, tax assessment, sales quotas, and parts ordering for manufacturing.

As is well known to those who have been analyzing the use of digital computers in business applications, it is as yet very difficult to make any broad generalizations. Although the *E101* is not expected to apply in all types of business computation, it is expected that it will be very effective in certain areas. The underlying design philosophy on this point is simple: If the shoe fits, wear it. Some characteristics of the basic machine, such as having form facilities that are as good as any standard accounting machine, and some accessories which will be provided, will help make the shoe fit on some of the potential applications.

Accessories to the Basic Machine

The parameters of the basic machine as described and discussed appear to be a sound approach to the basic objective, but in order to encompass a larger realm of application, the *E101* will be offered with certain optional accessories. The first of these accessories to be made available is a punched-tape reader. This reader will have two functions: first, of course, for automatic entry of numeric data; second, as a source of instructions to supplement the pinboards. The basic machine has eight pinboards, each of 16-step capacity. Addition of punched tape, in the form of a closed loop, may be considered as providing a ninth pinboard, which is somewhat slower in operation, but has the virtue of being of unlimited capacity. In adding instructions from tape, the ease of access to computational routines has been preserved. The coding is the same as for the pinboards, the tape can be inserted or replaced easily, and the instructions on the tape are used directly by the machine, that is, they do not first have to be transferred to an internal memory.

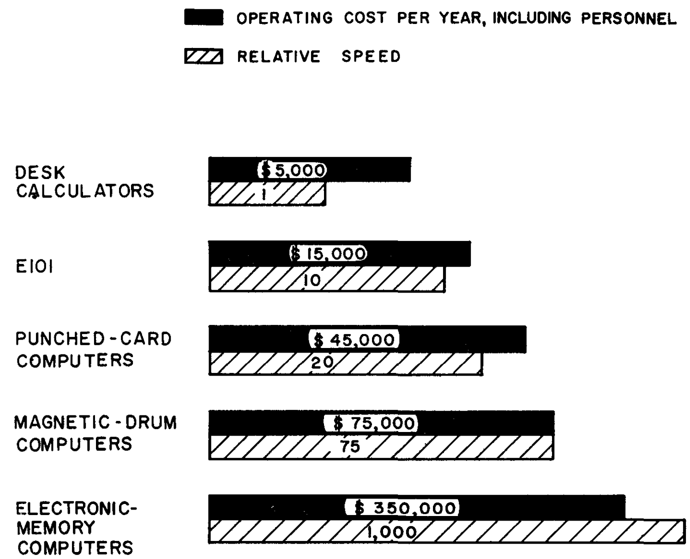


Fig. 5. Classes of digital computers

Discussion

J. H. Howard (Chairman): I might ask one question. I don't think you mentioned how programs could be stored on the template. Do you want to mention that?

Alex Orden: The program is stored in the pinboards which are removable and can be set on a shelf. In addition, as perhaps was clear from the illustration of the pinboard, we provide templates which are simply cards of the same size as the pinboard itself, with all the same numerals on them indicating what the instructions are and what line of coding you are in.

The general mode of programming a problem is to circle holes on a template and then bring the template over to the machine, take a pinboard, and drop pins in where the holes have been circled. Then generally the paper template is left on the pinboard more or less permanently. If the program is going to be used very often, the pinboards are stored on a shelf. If the program is going to be used intermittently, the set of paper templates, say half a dozen of them, can be removed as the pins are dropped out and the templates can be stored in an envelope. Reloading the program onto a set of pinboards at some later date requires some 3 to 5 minutes to load up a set of boards.

C. Gotlieb (Computation Centre, University of Toronto): Could you briefly list the order code available?

Alex Orden: The order code is more or less conventional for a small digital computer, i.e., the same general approach as in a large computer, but a less elaborate repertoire of instructions. The machine has the usual arithmetic instructions, shift instructions, read-in from keyboard, and print-out, as well as conditional and unconditional transfers and a special stepping instruction which allows you to scan through a sequence of addresses in programming loops, which involves the changing of addresses as part of a sequence of instructions.

There are some refinements. For example, absolute value instructions are included. There are also refinements in printing control that enter from the versatility of the accounting-machine-type printing unit, such as four separate types of print command which control format.

Mr. Casey (General Electric Company): Can you give us some figures on the time of various instructions? Has this machine been applied to solution of differential equations by relaxation methods?

Alex Orden: The basic speed was indicated when I said that the machine is generally in the range of 10 to 20 times faster than the use of a desk calculator. The addition-type operations (addition, subtraction, shift, take-absolute-value, and a few others) is $^1/_{20}$ second for each. Multiplication and division are $^1/_4$ second each. Print is half a second for a full word of up to 12 digits and sign.

As far as solution of differential equations by relaxation methods is concerned, I presume you mean the solution of partial differential equations, or at least that is uppermost in my mind.

That happens to be a field in which people speak very highly of the use of judgment and the use of an ordinary desk calculator. One takes a network of points and relaxes by hopping around from point to point, using one's best judgment, or at least using a visual scan which is to a certain extent more efficient than an automatic computer scan to find the largest residual to relax on.

We have been planning to study the use of the *E 101* on this type of problem from the point of view of the "super desk calculator." That is, the machine would be programmed to provide a few specialized routines so that when you have to do something computationally, you enter a number or several numbers and then the machine does the rest, rather than handle the computation fully automatically as you would in a large-scale computer.

My impression is that this will be a good approach in relaxation problems where you have nonrectilinear co-ordinates, that is, in cylindrical or spherical coordinates, where the computational drudgery associated with doing relaxation by hand with a desk calculator gets aggravating. I am doubtful whether it would be worth while to handle rectilinear co-ordinate problems this way.

J. R. Anderson (Bell Telephone Laboratories): Will the pinboard be available as a separate article of commerce?

Alex Orden: The pinboards will be available as an item to be purchased separately. Two full sets of pinboards will be provided with the machine.

D. N. Lee: Can you explain how storage of information is accomplished by the pinboard?

Alex Orden: Each pin makes a connection in a rectangular grid switching matrix. Each line of pin holes is for one single address instruction. Generally, three pins are inserted in a line, one pin to select the type of instruction, one for the tens digit of the address, and one for the units digit. The switching through a succession of pinboard steps is by electromechanical means.

J. Kates (KCS Data Control, Ltd.): Would you break down the $15,000 annual cost? What is the word length and storage capacity?

John Mekota (Raytheon Manufacturing Company): What is the cost of a pinboard? What are the operational speeds?

K. Enslein (University of Rochester): Does this computer also bridge the cost gap? How much for the basic machine?

Alex Orden: Let me begin with the last question, "Does this computer bridge the cost gap and what is the cost of the basic machine?" The machine price is $32,500. The rental price is $850 a month, or about $10,000 a year. We assert that this is a bridging of the gap costwise since this machine can be associated with a small group in a laboratory having one general type of problem and one person would be enough to program and operate this machine.

Of course the time spent on programming and operation might be spread over several men and be the equivalent of one full-time person, so that the rental cost is $10,000 a year, roughly, plus $5,000 a year for an operator. If you are talking about engineers, let us say it is $7,000 or $8,000 for operation. This is about double or triple the cost of having people work at desk calculators where the primary cost is salary. I think that is a bridging of the cost gap.

The price of extra pinboards has not been set yet. The operation speeds were discussed earlier, and I have broken down the $15,000 annual cost. The word length is 12 decimal digits internally and sign.

The average speed is the result of a combination of factors, such as keyboard entry, stepping of instructions, and the electronic mode of internal operation. I like to think of the machine as well balanced internally because if you try to come to grips with ways to speed up the machine then you find that no one way of changing things will alter the speed of solving problems greatly.

H. Freeman (Sperry Gyroscope Company): Could you give an indication of the complexity of the machine, say by giving the number of tubes, diodes, or relays?

Alex Orden. There are about 160 vacuum tubes and about 1,500 diodes. I am not certain of the number of relays. I believe it is something of the order of 20.

W. A. Hosier (Lincoln Laboratory, Massachusetts Institute of Techology): Do you have any reliability figures on the *E 101*, and what type of maintenance is felt necessary?

Alex Orden: We have operated only the model which is here at this conference. A second model is nearly complete with some of the features that will be on the production model, and the first production units will be on test within several months.

The reliability of our present initial model is a little hard to say very much about because, as usual, this is a machine that has gone through engineering tests, engineering modification, training of maintenance people, and all the usual headaches of a first machine.

The machine has been very nice to us in the sense of rising to practically every critical occasion. It has been on the road four times. This is the fourth, and each time it has been moved somewhere, set up, and put into operation. Of course in the framework of a meeting in a hotel whether it has been in operation with 100-per-cent accuracy, I do not know and nobody else knows.

As for type of maintenance, Burroughs will offer the machine by rental, meaning full maintenance, and by purchase, in which case the first year of maintenance will be part of the purchase price.

As far as training of your own maintenance people is concerned, the machine has the complexities that are usually associated with digital computers. On the other hand, it is a smaller machine with fewer tubes and fewer circuits, so that it certainly should be an easier task to train maintenance men for this machine than for the larger machines.

Small Digital Computers and Business Applications

PANEL DISCUSSION

THERE was a group of about 700 in attendance at the panel discussion on small digital computers and business applications. A show of hands was requested to clarify the composition of the audience. An estimated 15 per cent of the audience was interested in the design and manufacture of electronic equipment, while the remaining 85 per cent was interested in application and use of machines, being primarily business people. About 40 per cent of the audience indicated that punch-card equipments are used within their organization and that about 15 per cent now have some kind of electronic equipment in use. About 20 per cent of the audience indicated the intention of procuring electronic systems of some type.

Panel Discussion Method

A number of pertinent and controversial questions in the field of business use of electronic computers were discussed. Each of these questions was initially reviewed and discussed by panel members and then that particular question was thrown open to group discussion. Following this, a new question was then taken up in the same manner.

Questions Discussed

Business people have indicated that one of their most important questions about electronic data processing is: What is the state of the computer art so far as business applications are concerned? Should a company make plans now to acquire a computer system? An associated corollary question is: Has the design of computers stabilized so that a company can be certain that equipment secured now will not soon become obsolete?

Mr. Foster discussed the affirmative side of these questions and indicated that the primary question should be whether or not money could be saved by using electronic equipments. If the answer to this question is "yes," then companies should start to apply and use electronic data-processing systems as soon as possible.

Mr. Murdock presented some of the negative aspects. The electronic computer field did not grow from the needs of business applications. Instead, electronic computers were developed to do technical and scientific computations. However, because many aspects of computation, entering of input data, and of the printing out of results, were common problems in business-data processing, these technical computers could be and were modified to meet the needs of business. Mr. Murdock argued that we still have much to learn about the fundamental requirements of business-data-processing systems.

What should the company interested in electronic data-processing systems look for? Mr. Schutzberger presented information on this subject and enumerated 31 points of comparison in selecting a computing system. He pointed out that not all of these points are of equal weight, and some are interdependent and overlapping. It is difficult to compare various computers, even those in relatively the same price and speed range.

Should equipment be purchased or leased? Mr. Foster defended the principle of leasing equipment, a method of operation that is followed by Prudential Insurance Company with whom he is associated. The advantages of leasing equipment were pointed out as:

1. Equipment can be carried on an expense basis rather than having to purchase capital equipment.
2. There is no concern about obsolescence of equipment under a lease arrangement.
3. Companies supplying equipment on a lease arrangement usually provide maintenance service so that the user does not have to deal with this problem.

Mr. Schutzberger discussed the advantages of buying equipment as follows:

1. Equipment may readily be modified.
2. Equipment may be operated any number of shifts or staggered shifts at no additional charge.
3. There is usually better control of maintenance personnel.
4. Purchasing equipment is much cheaper.
5. The greatest argument against purchase is that of obsolescence, but if a machine can be used 3 or more years, the user will not have lost money as compared to renting, and quite possibly will have actually made money.

What are the practical problems of programming? Mr. Foster reviewed this question, and there was a great deal of audience interest in the topic. It was agreed that the fundamental problem in applying electronic equipment to business problems was that of defining explicitly and clearly the exact problem which the machine is to do. Experience of organizations that are applying electronic equipments to business problems indicates that considerable time must be spent in studying problems, in doing systems analysis work, and in performing the actual programming and check-out on the electronic computer.

Further, it was pointed out that programming is often stated by some people to be very simple to learn and easy to do, and yet by others is claimed to be quite difficult. Actually, both groups are talking about different aspects of the same problem.

The programming of an electronic computer, for example, can be compared to learning to play chess. The command structure and the operations that are built into the computer are the mechanical rules of the game, and these are fairly easily learned, just as it is easy to learn the rules about the moving of pieces in chess. However, to learn the refinements and the special techniques that are available in electronic computers is considerably more difficult and takes a great deal longer to master, just as it takes a long while to learn how to play chess well.

Summary

Conclusions derivable from the panel discussion are

1. There is a great deal of interest on the part of business people in the subject of electronic data processing, and they are eager to learn more about such systems and how to use them.
2. There are many controversial subjects related to using computers for business applications, and much remains to be learned as to the proper equipments to use and the best methods of applying the machines.

PANEL MEMBERS: W. D. Bell (chairman), Mellonics, Los Angeles, Calif.; W. L. Murdock, General Electric Company, New York, N.Y.; H. Schutzberger, Sandia Corporation, Albuquerque, N. Mex.; R. Foster, Prudential Life Insurance Company, Newark, N.J.

Redundancy Checking for Small Digital Computers

PANEL DISCUSSION

The chairman opened the session with some general remarks on errors, giving three kinds:

1. Errors due to machine malfunctioning.
2. Errors due to program mistakes.
3. Errors due to doing the wrong problem or using very inefficient methods.

He went on to say,

"Redundancy—that is, doing more than the minimum amount necessary—is the usual method of attacking these three sources of errors. In practice the form of the redundancy used, as well as the amount used, differs a great deal.

"The first type of error—machine malfunctioning—is the main theme of the speakers. The attack is through using either extra machine equipment that enables the machine to detect its own errors, or extra programming, and consequently machine time, to make the program show up the error. The questions are not only what kind of redundancy, but how much and where to put it.

"In attacking the second type of error—coding mistakes—there have been at least three approaches. One method is to let the coder supply the redundancy by coding more than the minimum. The second approach puts it up to the machine in the form of the magic words 'automatic coding.' In this method the machine does the bulk of the detailed coding and uses its built-in redundancy, if it has any, to see to it that it does not make a coding mistake. A third method has been to build extra equipment separate from, or inside of, the machine to supply either detailed coding and/or checks on impossible combinations of orders.

"These first two sources of errors are well recognized and vigorous attacks are being made on them. In contrast, the third type of error—doing the wrong problem or using inefficient methods—is seldom mentioned publicly. But estimates agree in putting this waste in efficiency above 50 per cent in most computer installations. Perhaps future conferences will discuss this phase of the problem more fully."

The first speaker, Miss Sibyl Rock, reported mainly on a survey she had made among engineers and mathematicians. She found, "that there is somewhat of a divergence in the viewpoint of the man who wants to get the problem done and the man who wants to find and correct the cause of the error. Thus, it is necessary to achieve a compromise between them.

"There is a fair amount of agreement among all interviewed regarding those errors which introduce so much confusion into the information that the program becomes utterly meaningless. Here a built-in check is desired.

"Another area of general agreement is in regard to use of magnetic tapes for storage of files and large amounts of standard information to be used later. These, they agreed, should be automatically checked.

"Where the major disagreement arose was in considering what to do in case the computer slips a digit or in some way introduce an error into the numbers which are being operated on. The service engineer prefers to stop the machine in such a fashion that he will know a little bit about where the error occurred. On the other hand, the man who has the problem to solve prefers to repeat the operation a second or third time, and if possible complete his work, before he abandons the machine. Only then does he want to turn it over to the service man to have the error found. Thus, if a machine were devised with the type of internal circuitry which stops the machine, the mathematician recommends a switch to disable those circuits when his results are otherwise checked.

"Another engineer pointed out a dilemma when it comes to determining the final and optimum checking system. First, the object should be to put any checking circuitry in where the greatest expectation of error is. While this can be predicted to a certain extent, it is not fully known until the machine has been in operation for a while. If the machine is first built without these circuits, and they are later added on the basis of experience, a new model has been produced in which the new circuits may introduce other weaknesses which were not originally anticipated and are not checked. Thus, the determination of the points at which flaw detectors are needed is always out of phase and a cycle behind the building-in of checking industry. Finally, the more conservative of those interviewed indicated that they felt that no internal checking method short of complete duplication of equipment really caught all the possible errors, that the approach of designing conservatively was certainly a must on any computer, and that for any machine the design itself and the experience with it would determine the points at which checks should be placed."

The second speaker, Dr. A. L. Samuel, observed that,

"Most small digital computers are largely used in accounting work. The penalty for an error in accounting work is very great, and one might therefore assume that such computers should use rather more in the way of redundancy for checking than might be the case for a scientific computer. As a matter of fact, the amount and kind of checking used in these machines is surprisingly large and varied. However, the amount of redundancy is frequently dictated by the translation problem which arises through the use of decimal arithmetic and the use of the 1-out-of-12 punched-card code for input and output as much as it is by the desire to use this redundancy in checking. Since the translation requirements introduce redundancy at one point in the system it is foolish not to make full use of this provision in error detection. We may frequently justify the introduction of some additional equipment to bring other parts of the computer up to the same standard.

"There is another aspect of computer design which is too often overlooked by proponents of redundancy codes; this concerns the need for checks relating to the controls within a machine, as contrasted with validity checks which verify data. For example, if data are called for from the store and the address selection has failed in some way and so delivered the wrong data, these data may well be a valid set of symbols which are certified as correct by the redundancy check, but their use will obviously lead to a quite incorrect result.

"One additional comment might be in order, however. It has been our experience in the main that it is better to program for error correction rather than to

PANEL MEMBERS: R. W. Hamming (chairman), Bell Telephone Laboratories, Inc., Murray Hill, N.J.; Sibyl Rock, Consolidated Engineering Corporation, Pasadena, Calif.; A. L. Samuel International Business Machines Corporation, Poughkeepsie, N.Y.; W. C. Carter, Raytheon Manufacturing Company, Waltham, Mass.

build error-correcting equipment into the machine. One of the necessary consequences of any error-correcting scheme is that it interprets many kinds of errors which are beyond its capabilities as being simple errors which it can correct and in attempting to correct an apparent simple error it will compound the real error. By programming, one can interrupt the solution in a way to minimize the damage done to other data in the machine and one can cause the computer to return to a verified point earlier in the solution.

"There is evidence that the reliability of at least one computer is rapidly becoming such that discussions of complicated redundancy checking systems seem almost superfluous. This still does not mean that we will not continue to use redundancy, but it does mean that we will look very carefully at such checking systems and insist that their costs in terms of equipment be as low as the costs in terms of time for programmed checks giving the same over-all reliability."

The third speaker, Dr. W. C. Carter, presented a paper by himself and John E. Makota.

In this paper, which is highly technical, the authors said:

"We propose to summarize the gain obtained by redundancy and will inquire as to whether this gain can best be obtained by using the extra equipment for checking or by using this extra equipment with other operating features. This discussion will be slanted toward small computers used for scientific work, and some conclusions that are reached will not hold under other hypotheses.

"The amount of results obtained for a fixed expenditure from a digital computer depends upon the amount of meaningful information obtained per unit time and the amount of time spent in problem preparation.

"The amount of correct information obtained per unit of scheduled operating time is a function of the reliability of the computer components, the amount and type of periodic testing of the system, and the amount of time necessary to repair the machine.

"Costs can be reduced not only by increasing the machine duty factor, but also by reducing problem preparation costs—especially for scientific problems.

"Thus there are many alternatives to adding checking equipment to increase the machine duty factor. The following additional equipment will provide increased speed or reduced programming costs and so must be considered as possible alternatives to checking equipment:

1. Floating-point operations.
2. Double precision operations.
3. *B*-lines.
4. Fast access lines.
5. Fast multiplication equipment.

These last alternatives may decrease programming time and thus may reduce costs.

6. Compiling routines.
7. Interpretive routines."

After introducing a hypothetical, though perfectly reasonable, machine to use for comparing these various factors, the authors conclude with the following statements:

"These considerations support the conclusion that all possible steps to increase machine speed should be taken first.

"Where programming costs are an important factor, as in varied scientific applications, the addition of programming aids should be made before checking.

"They also support the conclusion that there is a genuine economical and operational advantage in building checking into a machine under any circumstances."

These papers were followed by almost an hour of lively discussion of various pros and cons. No essentially new information seems to have come out of this, but a good many opinions were forcibly expressed. A vote at the request of Dr. V. M. Wolontis (of Bell Telephone Laboratories) on the question of how many people wanted complete self-checking gave about 60 per cent for and 40 per cent against it.

Small Digital Computers to Assist Large Digital Computers

PANEL DISCUSSION

THE participants in the panel agreed that there had been very few examples of installations where both large computers and small computers of the type discussed at the conference were available.

However, the installations at Douglas Aircraft and United Aircraft had contained, or did still contain, small-scale IBM card-programmed calculators (CPC) in addition to the large-scale IBM *701* computers. Mr. Lowe stated that the smaller computers had been abandoned at his installation because of the proved greater efficiency of the larger machine. Mr. Ramshaw stated that the United Aircraft installation still utilizes six CPC's because of their stand-by capacity for data reduction from the large number of test stations there. Use of these smaller (although not stored-program) calculators for these data-reduction problems prevented the necessity for stopping the larger machine to perform data reduction. It was later suggested from the floor by Dr. Grosch of the General Electric Company, that development now underway would allow such remote tests stands to "cut in" automatically on a large-scale computer for small-job computations, without causing more than a momentary delay of the major problem being computed.

It was the general consensus of the speakers that large computers performed more efficiently than small computers. However, Dr. Carr pointed out that, among the possibilities, small computers with the same instruction code as a large computer could be used to "check out" problems on the large machines, that small machines could perhaps be used to handle "exceptions" in a fashion parallel to the main computation performed by a large machine, and that small computers, with enough imagination on the part of the users, might be effectively applied to act as test equipment for larger computers, serve as "buffers" or "inertia storage" for input-output and computer control, and generally serve as accessory devices.

It was pointed out from the floor that

PANEL MEMBERS: J. W. Carr, III (chairman), University of Michigan, Ypsilanti, Mich.; John Lowe, Douglas Aircraft Corporation, Santa Monica, Calif.; Walter Ramshaw, United Aircraft Corporation, East Hartford, Conn.

computers of the Burroughs *E 101* type, with emphasis on low cost and ease of coding, would probably prove extremely useful in many laboratories where large machines were already available, in serving as a local calculator for groups removed physically from the major equipment.

Other discussion from the floor centered around the use of plugboards on small machines to provide competition with large machines, the code-checking possibilities of small computers for large computers, and relative merits of supplementing large machines by small computers at remote locations.

In general, the absence of small computer users, designers, and builders left the discussion mainly to the adherents of larger computers. Nevertheless, the interest engendered by some of the novel proposals introduced indicated that the use of small computers as assistants to the larger machines is not a dead issue.

Numerical Solution of Differential Equations

H. M. GURK

MORRIS RUBINOFF
ASSOCIATE MEMBER AIEE

THE problem of finding solutions to various types of differential equations has intrigued mathematicians from a theoretical point of view for many years. It has also plagued many applied scientists for an equally long period of time. Except for the relatively few equations whose solutions are available in closed form, the best one can do is to obtain approximate solutions.

These approximate or numerical solutions of differential equations are usually obtained in a stepwise fashion. Thus, an approximate solution of the equation

$$\frac{dx}{dt}=f(x, t)$$

is obtained as a sequence of values $x_n, n=0,1,\ldots$ x_n is taken as the numerical solution of the equation at the times $t = nh(n=0,1,\ldots)$ where h is a fixed positive interval called the integration step. If we let $\dot{x}_i=f(x_i,ih)$, then numerical solutions are usually obtained by the use of formulas of the following types:

1. The open formulas

$$x_n=\sum_{j=1}^{M} a_{jo}x_{n-j}+h\sum_{j=1}^{N} b_{jo}\dot{x}_{n-j} \quad M\geq 1,\ N\geq 0$$

where the a_{jo}'s and the b_{jo}'s are real.

Such a formula is denoted as an O_{MN} formula.

2. The closed or repeated closure formulas

$$x_n=\sum_{j=1}^{P} a_{jc}x_{n-j}+h\sum_{j=0}^{Q-1} b_{jc}\dot{x}_{(n-j)} \qquad P,Q\geq 1$$

where for each n this is applied in an iterative procedure with the value of $\dot{x}_n$ at any iterative step being obtained using the approximate available; initially we would have to make an educated guess, perhaps by using an open formula. This formula is denoted rC_{PQ}—the r standing for repeated.

3. The mixed formula $[O_{MN}, C_{PQ}]$ consisting of the open method O_{MN} followed by a single application of the rC_{PQ} formula, which we denote just as C_{PQ}, the result of this being the accepted value of x_n. All the ordinates, that is, x_{n-j}'s, in both the open and closed parts of this mixed formula are the final values computed from the closed formula at previous steps, and all derivatives ($\dot{x}_{n-j}$'s) are computed using the values computed from the open formula.

In the following discussion we will call any open formula O_{MN} with given coefficients, a quadrature formula of type O_{MN}. Likewise, we will talk about quadrature formulas of type rC_{PQ}, and of type $[O_{MN}, C_{PQ}]$.

Classically there has been one method used in choosing the quadrature coefficients which appear in any open or closed formula. This is the so-called polynomial method. For a given type of formula this is equivalent to choosing the coefficients so that the positive integer R is a maximum where the equations $\dot{x}=0$, $\dot{x}=1$, $\dot{x}=t$, $\dot{x}=t^2, \ldots, \dot{x}=t^R$ are solved correctly by the quadrature formula. For these open and closed types of formulas, such sets of coefficients are easily found since the procedure amounts only to the solution of a set of simultaneous linear algebraic equations. The mixed quadrature formulas commonly used are the combinations of classical open and classical closed formulas.

Using the classical open or closed formulas, one can easily compute that the truncation error per step, that is, the error one obtains at the nth step assuming infinite precision and all previous values correct, is of the order of $Ch^{R+2}|f^{(R+1)}(x(\tau),\tau)|$ where τ is some value of t between the greatest and least values used in that step, and C is a constant. Thus, for a given formula, one can make the truncation error per step as small as desired by decreasing h, the size of the step. If the total interval over which the equation is to be solved, and an upper bound for the $(R+2)$th derivative in this interval are known, then the total truncation error can be reduced below any desired bound by making the step length sufficiently small. The classical open or closed formula of a given type is the formula of that type which generally makes the truncation error per step tend to zero fastest as h tends to zero.

It is often the case that the solution of the differential equation is desired over an essentially infinite interval and the knowledge of a bound for the error per step is not of much use. In these cases the asymptotic behavior of the numerical solution may be very important and a close approximation of the true solution's asymptotic behavior is desired. With this in mind, we come to the notion of the

H. M. GURK and MORRIS RUBINOFF are with the Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia, Pa.

The authors wish to thank Dr. H. J. Gray of the Moore School, who did the basic work on stability charts, and Dr. F. J. Murray of Columbia University for their many suggestions regarding the contents of this paper.

This work was performed in connection with contract *Nonr-551(02)* with Special Devices Center, Office of Naval Research, Port Washington, N. Y.

stability chart which was introduced by Gray.[1,2]

Stability Charts

Assume that it is desired to obtain the solution of the linear differential equation.

$$\dot{x} = \lambda x \qquad (1)$$

where λ is complex. λ will be called the true or natural resonant frequency. The true solution of this equation is $x_o e^{\lambda t}$ where x_o is the initial value. Assume that we have decided upon a fixed quadrature formula of a type described above. Let h be the step length and assume that we are given a set of initial conditions for a function satisfying an equation of type 1.

Let $z = \lambda h$. Note that z is a dimensionless natural frequency. Then it is well known[1,3] that there exists a positive integer s, constants c_i $(i = 1, \ldots, s)$ dependent on the initial conditions, and complex valued functions $r_i(z)$ $(i = 1, \ldots, s)$ such that the computed solution of equation 1 is

$$x_n = \sum_{i=1}^{s} c_i [r_i(z)]^n \qquad n \geq 0 \qquad (2)$$

or if $w_i(z)$ is the principal value of log $r_i(z)$ then

$$x_n = \sum_{i=1}^{s} c_i e^{n\, w_i(z)} \qquad (3)$$

The $r_i(z)$ are the roots of a polynomial of degree s the coefficients of which are linear in z and dependent on the quadrature coefficients in the quadrature formula.

Let

$$w_d(z) = w_i(z)$$

where

$$Re(w_i(z)) \geq Re(w_j(z))$$

for $j = 1, \ldots, s$. w_d may be a many-valued function of z. For the moment let us consider only domains where w_d is a single-valued function of z. In this case we call w_d the dimensionless dominant frequency of equation 3.

Now from equation 3, we can see that as n tends to infinity the computed solution x_n will tend to $c_d e^{n w_d(z)}$. Since we know the true solution to be $x_o e^{\lambda t}$, then aside from a common factor the true and computed solution will have the same asymptotic behavior if z is equal to $w_d(z)$.

A stability chart is a picture of the mapping from $z = \lambda h$ to $w_d(\lambda h)$ for a given quadrature formula. Thus, for a given λ and a given h one can look at the stability chart and find the corresponding $w_d(\lambda h)$.

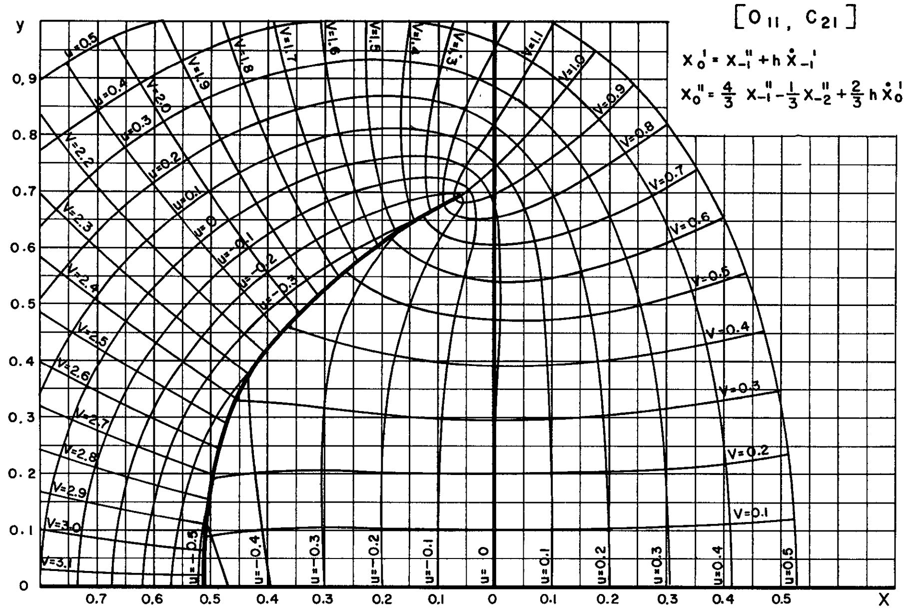


Fig. 1. Classical $[O_{11}, C_{21}]$ stability chart

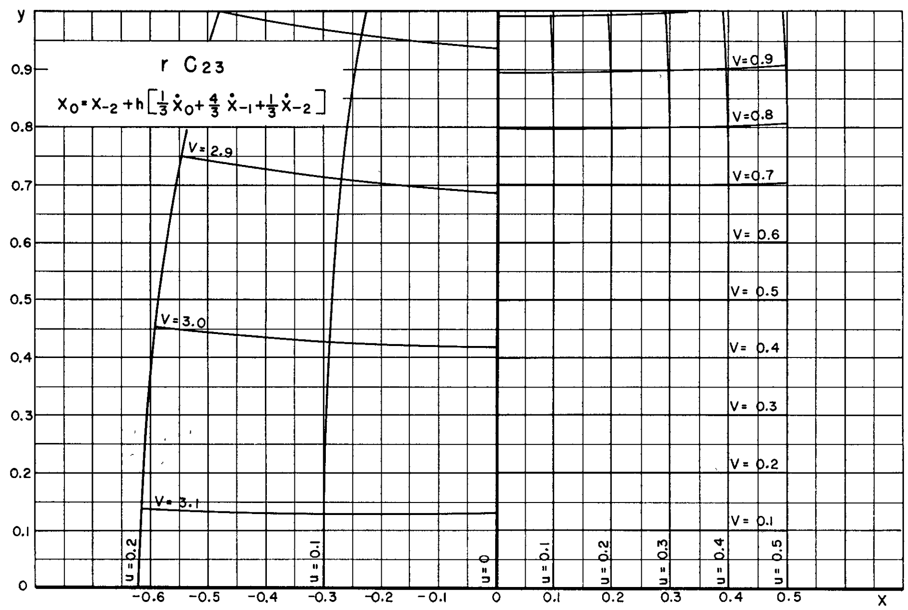


Fig. 2. Classical r C_{23} stability chart

It can then be decided whether or not the computed asymptotic behavior is similar enough to the true behavior for the needs of the problem. (For the balance of the paper the subscript d will be suppressed except where ambiguity may arise.)

As an example let us look at Fig. 1, the stability chart of the classical $[O_{11}, C_{21}]$ formula. On the chart x and y represent respectively, the real and imaginary parts of z, and u and v are the real and imaginary parts of $w(z)$. Thus, if we are solving the equation $\dot{x} = (-3+3i)x$ with this formula and a step $h = 0.1$ second, then $z = -0.3 + 0.3i$ and w would be approximately equal to $-0.3 + 0.285i$. Depending on the needs of the problem one can determine in advance whether or not the asymptotic behavior indicated by this value is good enough.

Notice on the figure to the left of the origin there is a heavy line delineating a locus of discontinuities in w. This line is called the branch contour; it displays values of z to which there correspond two different values of $w_d(z)$, that is, there exist $w_i(z), w_j(z)$ not equal, but both with the same real part u_d, where u_d is greater than the real part of any other $w_k(z)$. For any z on the branch contour, the two values $w_i(z)$ and $w_j(z)$ can be obtained by approaching z from opposite sides of the contour.

The branch contour is more or less a

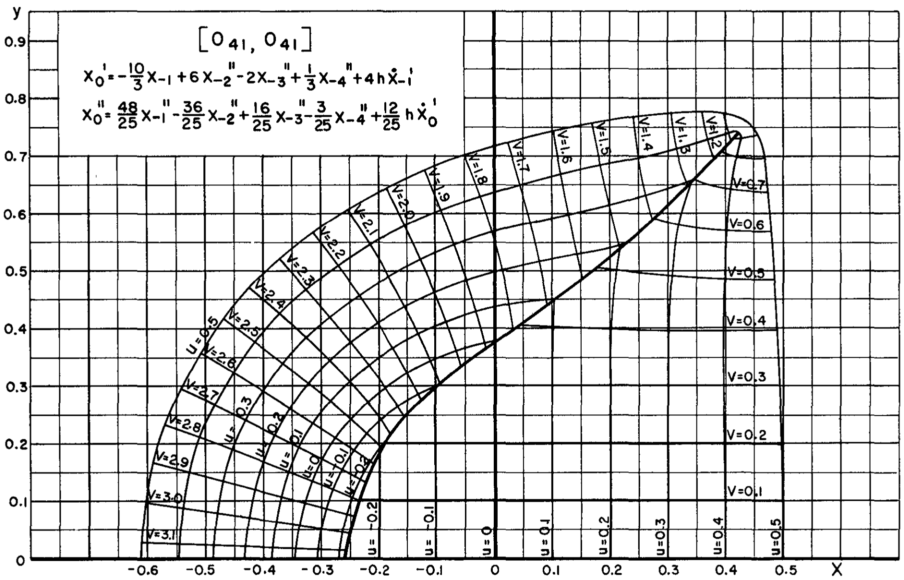


Fig. 3. Classical [O_{41}, O_{41}] stability chart

dividing line between the acceptable and nonacceptable values of w_d. As one goes away from the origin it can be seen that the values of w gradually differ more and more from z. However, as z crosses the branch contour the change becomes more rapid. Indeed the real part of w, that is, u, henceforth decreases as x increases instead of conversely, and the imaginary part v jumps discontinuously to a value considerably different from the true imaginary part y. Thus, the region where $w(z)$ and z, are approximately the same, that is, the region where the $u-v$ lattice is nearly square, is on the origin side of the branch contour.

If we had been solving a system of simultaneous linear equations instead of just a single one, then there would be a number of true frequencies, that is, λ's, involved. Thus, for given h, there would be a w_d corresponding to each λ. We would like all these w_d's to be approximately equal to their corresponding z's. By looking at the λ's we could make certain that our w_d's all fall within an approximately square region just by choosing h small enough. Thus, the stability chart permits an immediate choice of step length.

On the other hand the stability chart can be used as an a priori criterion for evaluating a quadrature method. One need only look at the size of the "square" region. The larger the region, the more desirable the formula. Of course for different problems, different regions are important; thus, analysis should really be made with respect to a given problem.

The study at the Moore School was concerned mainly with equations whose solutions were stable in the sense that they tended to zero or to a constant as t tended to infinity. In the case of linear equations this would mean that the true frequencies all have real parts which are nonpositive. Thus, in analyzing a quadrature formula by means of its stability chart, we would always be on the lookout for a $u-v$ lattice that was approximately coincident with the $x-y$ lattice at the origin and to the left of it. The larger the size of this square part of the $u-v$ lattice and the further the branch contour is away from the origin, the more desirable the quadrature formula.

The question naturally arises as to whether or not quadrature formulas with stability charts with the desired properties could be synthesized. The problem has been attacked by attempting to obtain a mapping which would make a good chart and then finding a quadrature formula to which this would correspond. From our knowledge of the type of functional relationship between z and w, we have been able to develop some general tools which are useful for good synthesis and also several completely new formulas with the desired properties for their stability charts. A particularly interesting fact is that it is often desirable, for a given type of formula, to impose as conditions on the quadrature coefficients some, but not all, of the equations which determine the coefficients for the classical quadrature formulas previously mentioned. In particular the first two equations, which are equivalent to

$$w(0)=0$$

and

$$\left(\frac{dw}{dz}\right)_{z=0}$$

are always imposed. A more complete discussion of this synthesis problem is now being published.[4]

In the case of a frequency with a nonpositive real part, we can even get more information about the asymptotic behavior of the solution from the stability chart. Assume we are solving $\dot{x}=\lambda x$. Then we get a solution of the form

$$x_n=\sum_{i=1}^{s} c_i e^{nw_i(\lambda h)} \qquad (3)$$

Suppose for the purposes of solution we assume initial values $x_j=x_o$ for $j=0, -1, \ldots, -s+1$. Then

$$x_0=\sum_{i=1}^{s} c_i e^{jw_i(\lambda h)} \quad j=0,-1,\ldots,-s+1 \qquad (4)$$

or solving for c_i

$$c_i=x_0 e^{(s-1)w_i(\lambda h)} \frac{\prod_{j\neq i}(1-e^{w_j(\lambda h)})}{\prod_{j\neq i}(e^{w_i(\lambda h)}-e^{w_j(\lambda h)})} \qquad (i=1,\ldots,s) \quad (5)$$

Now consider just $w_d(\lambda h)$. If (a)

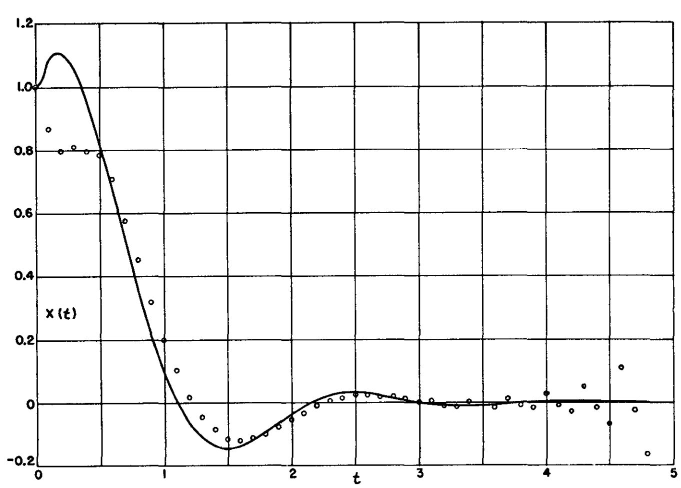


Fig. 4. Numerical solution of equations 6-x

○ ○ ○ h=0.1
——— h=0.02

$Re(w_d(\lambda h))$ is approximately equal to $Re(\lambda h)$ and therefore nonpositive and if (*b*) $Re(w_d(\lambda h)) >> Re(w_j(\lambda h))$ for $w_j \neq w_d$ then from equation 5, the coefficient of w_d in equation 4, which we call c_d, is approximately equal to x_0.

But condition *a* can be checked just by looking at the stability chart and condition *b* is indicated if the branch contour is not near the point in question. Thus, in this case we not only have a comparison of asymptotic behavior of the true and computed solution from their exponential behavior, but also of the constant factors multiplying these exponentials.

So far, we have shown only the use of the stability charts as a criterion for evaluating the worth of a quadrature formula for solving linear differential equations with constant coefficients. Actually, the case of a single equation has been emphasized, but the arguments can be generalized directly to a set of simultaneous equations. What about the cases where there are forcing functions added on, or where the coefficients are varying, or even the general nonlinear case?

In the case of the forcing functions, if they are just transients or if their growth is slower than that of the solution of the homogeneous equation, then asymptotically the differential equation is essentially linear and the previous analysis holds. In the cases of linear equations with varying coefficients or nonlinear equations, if the true solution is of finite exponential order, one could expect to be able to use the stability charts although rigorous proofs of this are still lacking. There is available, of course, the linearization theory of Liaponoff and the work of Bellman on stability of differential equations,[5,6] but the application of these results to obtain a complete, general, rigorous justification in these troublesome cases has not yet been successful.

However, despite the lack of rigorous justification, there has been considerable encouraging empirical justification for the use of the stability charts. The next section discusses a number of specific examples.

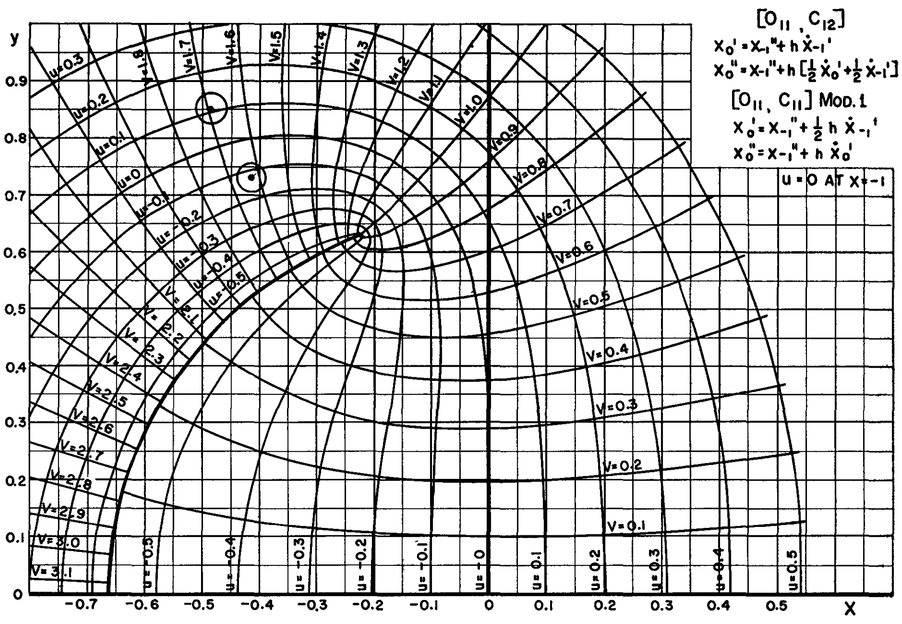


Fig. 5. [O_{11}, C_{11}] mod 1 stability chart

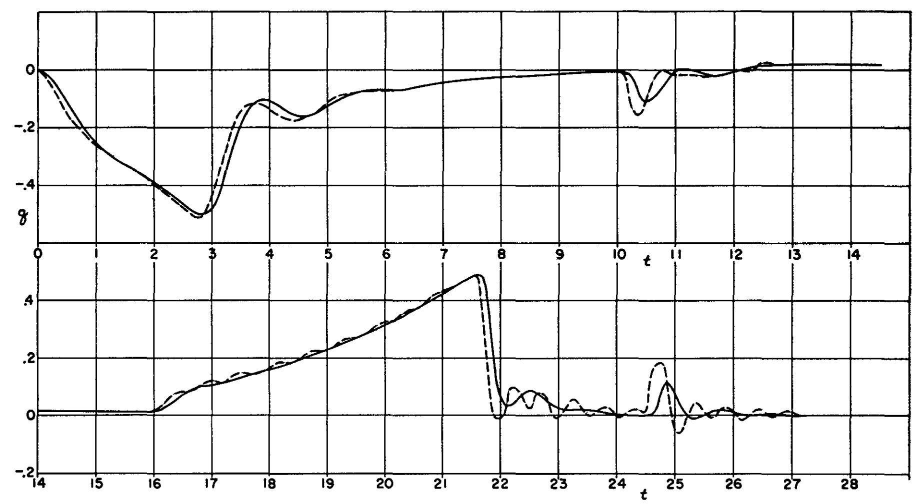


Fig. 6. Dive maneuver—q

——— true solution
– – – [O_{11}, C_{11}] mod 1 solution, h = 0.15

Empirical Results

The project with which this work is connected was concerned with the feasibility of using a digital computer to actuate an operational flight trainer. Speed, of course, was essential since the computations had to be performed in real time. This meant that once the step h was chosen for a quadrature formula, then all the necessary single-step computations had to be performed within that interval. Because of the large number of computations which are necessary for solving flight equations, it was obviously desirable to make h large. On the other hand the solutions were more accurate in general if h was small. And since the trainer had to remain "in flight" for a reasonably long time to be of value, the number of steps was essentially infinite. This asymptotic behavior of the computed solutions was certainly of prime importance.

The stability chart analysis immediately ruled out the use of certain classical, often used quadrature formulas. For example, any formula based on the well-known Simpson's Rule was absolutely wrong for our uses. Fig. 2 shows the repeated closure version of Simpson's Rule, rC_{23}, and indicates a branch contour through the origin and up the y axis. To the right of this axis the chart looks good, but to the left of the axis the mapping shows u is positive where of course x is negative. Since our interests are in frequencies with nonpositive real parts, the afore-said property, which is characteristic of the formulas derived from Simpson's Rule, prohibits the use of these formulas.

A striking verification of the stability chart analysis is given by the solution of

$$\begin{cases} \dot{x} = y \\ \dot{y} = -3y - (6.25 + 2.4t + 0.36t^2)x \end{cases} \qquad (6)$$

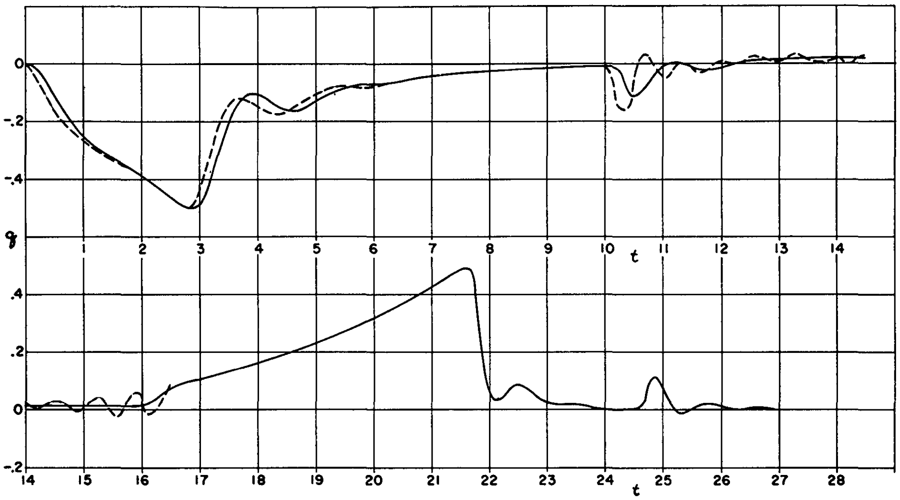


Fig. 7. Dive maneuver—q

—— true solution - - - $[O_{11}, C_{11}]$ mod 1 solution, h=0.175

using the $[O_{41}, C_{41}]$ formula

$$\begin{cases} x'_n = -\frac{10}{3}x''_{n-1} + 6x''_{n-2} - 2x''_{n-3} + \frac{1}{3}x''_{n-4} + 4h\dot{x}'_{n-1} \\ x''_n = \frac{48}{25}x''_{n-1} - \frac{36}{25}x''_{n-2} + \frac{16}{25}x''_{n-3} - \frac{3}{25}x''_{n-4} + \frac{12}{25}h\dot{x}'_n \end{cases} \quad (7)$$

with $h=0.1$ second.

The stability chart for this formula is given in Fig. 3. If we let $a(t)=6.25+2.4t+0.36t^2$, then the frequencies of the system of equation 6 are

$$\lambda = \frac{-3 \pm \sqrt{9-4a(t)}}{2}$$

By looking at the chart we see that the computed solution should begin to oscillate slowly about the true solution when $\lambda h = -0.15+0.25i$, that is, at the branch contour. When $\lambda h = -0.15+0.35i$, that is, as it crosses the $u=0$ line, instability should really begin to show in the computed solution. From the form of $a(t)$ one can compute these two times to be approximately 1 and 3 seconds. Because of the transient effects and the initial conditions, the true solution itself is rather oscillatory at the beginning. Thus, the slow oscillation of the computed solution (represented by circles) about the true solution (solid line) is not clearly evident at $t=1$ in Fig. 4. However, it is evident that at $t=3$ the oscillation is building up and the computed solution is becoming unstable. By $t=4$ this instability is quite evident; the solution gradually getting wilder and wilder. The true solution is represented by the computed solution with $h=0.02$, which is a valid representation at this point in the computation because of the small size of h.

The simulation of an airplane in a dive maneuver offered a good example of the use of the stability charts in choosing the step length. Since this maneuver is two-dimensional, the equations are much simpler than the general flight equations and the instantaneous natural resonant frequencies can be computed easily. The worst oscillations occur when the plane is being pulled out of the dive. Analysis prior to the numerical computations revealed that at this point the frequency was approximately $-2.82 \pm 4.88i$.

The flight equations were "solved" by the formula $[O_{11}, C_{11}]$ mod 1 whose stability chart is given in Fig. 5. Solutions with step length $h=0.15$ and $h=0.175$ were obtained. Computing λh for both h's, where we take λ as the frequency at pull-out, reveals that $w(0.15\lambda)$ is under the $u=0$ line and $w(0.175\lambda)$ is above it (see circled points on Fig. 5). Thus, using $h=0.15$ we have damping as we should, even if at the wrong rate, whereas for $h=0.175$ we have a frequency with a positive real part. Thus, we would expect instability at the pull-out point for the second solution, whereas the first would oscillate around the true solution but gradually tend toward it. Figs. 6 and 7 which represent the values of q, the pitch rate, show just this result, the pull-out being at about 16 seconds. The results for $h=0.175$ were not plotted past 16.5 seconds since they became very wild.

As a final example we offer the results of computation in a very nonlinear case, the Immelman turn maneuver. In this maneuver, the airplane makes half an inside loop followed by a roll through 180°, so that the plane ends right side up. The violence of this simulated maneuver is illustrated by the fact that there is a maximum acceleration of 7 g during the inside loop.

Computations were made first with an accurate, classical, repeated closure method using an interval of 1/40 of a

Fig. 8. Immelman turn—q

—— classical $O_{11}rC_{41}$
⊙ $[O_{30}, C_{31}]$ mod Gurk
× O_{33} mod Gurk

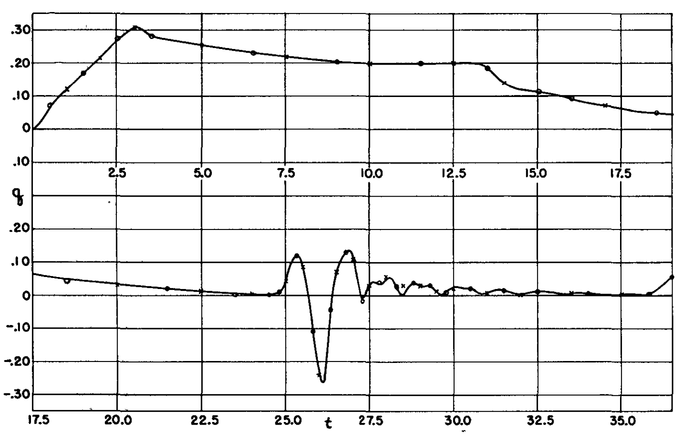
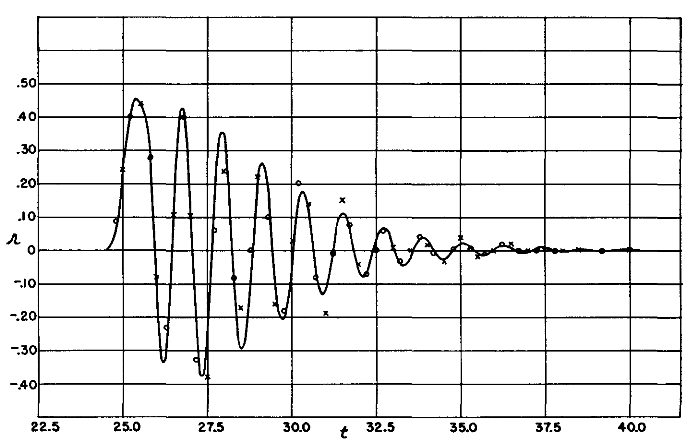
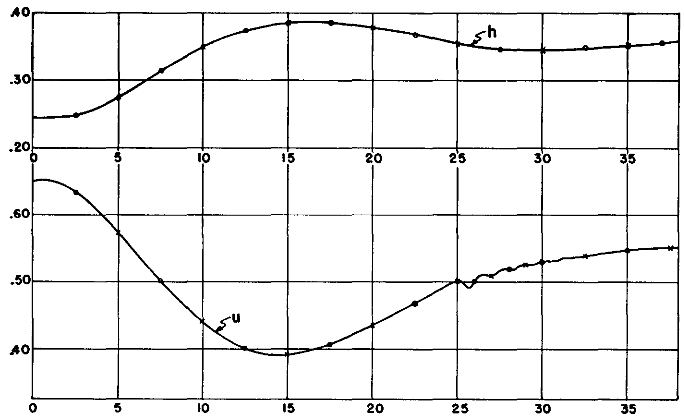


Fig. 9. Immelman turn—r

—— classical O_{11} rC_{41}
⊙ $[O_{30}, C_{31}]$ mod Gurk
× O_{33} mod Gurk





Figs. 10 and 11. Immelman turn—u and h ———

classical O_{11}, rC_{41}
⊙
[O_{30}, C_{31}] mod Gurk
×
O_{33} mod Gurk

second, then with both a nonclassical mixed formula and a nonclassical open formula using double the interval, that is, 1/20 second. The two latter formulas were synthesized by the stability chart method.[4]

Figs. 8, 9, 10, and 11 show some results of these computations. In the legends q represents pitch velocity, r yaw velocity, u forward velocity, and h altitude. The repeated closure method is designated $O_{11}rC_{41}$ while the two newly synthesized formulas are the "mod Gurk" formulas.

It can easily be seen that the errors are very small for all the variables. Only for r are they even noticeable at any time, and even in this case they have died away by 40 seconds. The results make it appear that for equations of this type, the synthesized formulas were essentially as good as the well-accepted, more complicated, repeated closure formula.

Certainly these few examples are not rigorous justification of the stability chart method. They do indicate, however, that even for nonlinear equations, it has much promise.

References

1 Numerical Methods in Digital Real Time Simulation, **H. J. Gray, Jr.** Dissertation in Electrical Engineering, Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia, Pa., June 1953.

2. Numerical Methods in Digital Real Time Simulation, **H. J. Gray, Jr.** *Quarterly of Applied Mathematics*, Brown University, Providence, R. I., vol. 12, no. 2, July 1954.

3. Calculus of Finite Differences (book), **L. M. Milne-Thomson.** Macmillan and Company, London, England, 1933.

4 The Use of Stability Charts in the Synthesis of Numerical Quadrature Methods, **H. M. Gurk.** *Quarterly of Applied Mathematics*, Brown University, Providence, R. I.

5. A Survey of the Theory of the Boundedness, Stability and Asymptotic Behavior of Solutions of Linear and Non-Linear Differential and Difference Equations, **R. Bellman.** Office of Naval Research, Department of the Navy, Washington, D. C., January 1949.

6. Stability Theory of Differential Equations (book), **R. Bellman.** McGraw-Hill Book Company, New York, N. Y., 1953.

7. Numerical Solution of Differential Equations (book), **W. E. Milne.** John Wiley and Sons, Inc., New York, N. Y., 1953.

8. The Use of Exponential Sums in Step by Step Integration, **P. Brock, F. J. Murray.** *Mathematical Tables and Other Aids to Computation* VI, nos 38 and 39, April and July 1952.

Discussion

Sol Gorn (Ballistic Research Laboratories): Has any work been done to include round-off errors in your analysis?

H. M. Gurk: Some work has been done at the Moore School to include round-off errors. For linear equations it was found that the round-off errors essentially satisfied the same equations as given by the quadrature formula.

K. Eisemann (International Business Machines Corporation): Where is there published material available on stability charts?

H. M. Gurk: An article appeared in the summer issue of *The Quarterly of Applied Mathematics*, by Dr. H. J. Gray. There is another article appearing in an issue of *The Quarterly of Applied Mathematics* that should come out within the next few months on the synthesis of quadrature formulas with the stability charts.

We also have much information at the Moore School. We have a book of 30 stability charts which we would be glad to send to anyone who is interested in it.

E. Mehr (New York University): Has this method been applied when some coefficients are periodic?

H. M. Gurk: I believe that some work has been done on this. I might ask my colleague, Dr. Rubinoff, about this. He has been connected with the project a little bit longer than I have.

Morris Rubinoff: Which coefficients are the periodic ones?

E. Mehr: Has this method been applied to the case of a linear equation, nonconstant coefficients, where one of the coefficients is periodic?

Morris Rubinoff: Some work along that line has been done by Dr. Grosswald, who is in the mathematics department, and just where it leads we do not know for sure, but he has treated cases involving such coefficients. The results are not sufficiently conclusive to be able to warrant any direct answer to your question at this time.

L. L. Gotlieb (University of Toronto): Having once found a stable formula, with a large interval of integration, does this become a "best" formula to be used always, or are there limiting conditions?

H. M. Gurk: No, we definitely will not say that a particular formula is the best formula to be used always. As I mentioned in the paper, the analysis should be with respect to the particular region in which we are interested. I might give you a striking illustration of a formula that should be used sometimes but definitely not always.

This is a slide of method rC_{23}, a classical formula. This is the repeated closure formula which corresponds to Simpson's rule. If you look at the chart you can see that to the right of the $x = 0$ line, to the right of the y axis, the u-v lattice is quite square. You would expect that this formula could be used for linear equations with true frequencies with positive real parts with good results. However, the branch contour goes right up the y axis, and to the left of the axis, if we look at values of u, for instance, we can see that they are positive whereas values of x are negative. In this case rC_{23} could be used for true frequencies with positive real parts but would be completely hopeless for true frequencies with negative real parts.

R. G. Tantzen (Holloman Air Development Center): What is the speed of the particular computer that was used for the example given?

H. M. Gurk: The goal of the project on which this work was done was to design a computer fast enough to do this real time simulation. The computer has been designed; it is called a sequential computer.

I am not sure of the exact times of operation, but addition takes on the order of 5 microseconds, multiplication on the order of 15, and division I believe on the order of 105. And we have programmed the simulalation of an airplane with this computer and we have found that the computation could be done in about 1/50 second. And we find that 1/20 second has been quite adequate for accurate computation so far.

Stanley Gill: Have you prepared stability charts for the Rung-Kutta process?

H. M. Gurk: Yes, we have, and as you might expect, the stability chart for the Runge-Kutta process is very good. If you would like a copy of the stability charts for a number of classical formulas, a number of newly synthesized formulas, the Runge-Kutta formula, and one or two others, we would be glad to send you a copy.

A. S. Robinson (Columbia University): Would a correct summary be that the chart requires a knowledge of the true solution and that the solution must be in the form of exponentials?

H. M. Gurk: It is certainly better if we do know in advance the form of the true solution. However, in simulating flight we certainly do not know exactly the form of the true solution. We do know, for instance, that the true solution cannot be unstable and tend toward infinity because the wings of the airplane would probably be pulled off long before it got there. So we do know on this particular project that if we have a set of linear equations we would be interested in frequencies with nonpositive real parts. However, we have used the stability chart analysis, or we have used formulas, let us say, which are synthesized from using the method of the stability chart, on equations about which we really know nothing concerning the solution.

M. A. Hyman (Westinghouse Electric Corporation): If $w = z > 0$, would you say that there is stability? Real parts, that is.

H. M. Gurk: This depends on your definition of the word "stability." We would say so, but if you would like to define stability as the property of a solution tending towards zero to a constant, then the answer is obviously "no." Stability is one of the most overworked terms in mathematics.

M. A. Hyman: May I comment on that? The point of my question was that if the two are equal, the numerical solution will still drift off from the true solution exponentially, and I want to know—and this is, incidentally, unavoidable—if you accept this.

H. M. Gurk: Dr. Rubinoff will answer.

Morris Rubinoff: As Mr. Gurk has said, there is always the headache as to what is meant by stability. Since we were primarily interested in operational flight trainers, or real time simulation if you like, our definition of stability was fitted into two parts. In the case of the true solution it has exponentials with negative real parts so that things should tend toward zero or towards a constant. Then we insisted that the computed solution also have a negative real part, and we would like a negative real part to be as close to the true negative real part as possible. But what we do insist really is that it also tend towards zero or some constant and not go off by itself to a new problem and a new solution and leave the pilot flying backwards. That would be unfortunate.

In the case of positive exponentials we adopted the attitude that when the pilot is foolish enough to have himself in, say, a high-speed stall, to quote Professor C. D. Perkins, of Princeton University, "He has creamed himself by then," and we don't mind if the solution creams itself.

We do, however, feel that stability in that sense, in the case of positive exponentials defined as the ratio of w to z, should be as close to unity as possible; the words "as possible" meaning to get a stability chart that will do it. But our emphasis was on the negative real part because we certainly wanted things like steady, level flight and maneuvers which are inherently stable to remain stable for the pilot.

Mr. Casey (General Electric Company): Please explain again how the stability charts are computed in general.

H. M. Gurk: How they are computed generally? Well, this is a rather long process. This is essentially what makes up the major part of Dr. Gray's thesis, which covered some 100 pages. If you would like to look at this thesis, you are welcome at the Moore School. However, I don't think that I can do it here.

Mr. Edelman (Radio Corporation of America): What about nonlinear equations with Bessel function coefficients?

H. M. Gurk: I assume that you mean have we done anything on such equations. The answer is "no."

Applications of Automatic Coding to Small Calculators

L. D. KRIDER

THERE is a fairly large amount of recent literature on the subject of automatic coding for large calculators. For example there are the various compilers constructed for UNIVAC,[3–7] the International Business Machines (IBM) *701* speed coding system,[2] and the Summer Session Computer of the Whirlwind project at the Massachusetts Institute of Technology.[1]

Little, however, is available on automatic coding techniques for small computers. This is unfortunate because automatic coding has much to offer the small computer, since a small machine may be characterized mainly by limited forms of input and output, relatively small memory, and simple coding systems, such things as automatic address modifiers generally being omitted. By means of automatic coding many desirable features can be synthesized. The use of the given input-output mechanisms can be systematized in the form of standardized subroutines and the human coder provided with short so-called speed codes to utilize them. Automatic coding techniques can also be used to conserve our limited memory, for although a portion of the memory is occupied with the necessary interpreting or translating routines and operational subroutines, a lot of repetition has been eliminated from the problem itself. In the basic machine code division may require several instructions to position the dividend properly before division, to provide for round-off, and possibly to test for overflow. A problem involving many divisions would have to repeat these steps each time division was called for. However if a simple speed code is provided which means "perform all the steps necessary to obtain a rounded, properly positioned quotient," only one memory cell need be occupied by the division instruction. The sequence of operations actually performed by the calculator need be stored but once in the form of a subroutine. This suggests one possible definition of automatic coding as "any coding procedure whereby a sequence of machine instructions is initiated by a single so-called speed-instruction or pseudo-instruction provided by the human coder." Still another need for automatic coding arises in the small organization with a limited number of personnel. Such a group may find a large number of problems bottlenecking at the coding phase of the computation process, while the calculator itself is standing idle for want of a coded problem. It seems logical in these cases to shift as much of the routine labor of coding onto the calculator itself as possible. Briefly, the advantages of automatic coding may be summarized as efficient organization and utilization of limited machine facilities, and time-saving convenience to the human coder.

L. D. KRIDER is with the U. S. Naval Ordnance Laboratory, White Oak, Silver Spring, Md.

A Speed Code for the IBM Type 650

The following describes a few of the essential features of the IBM type *650;* for further details see reference 5. It has a magnetic drum memory of either 1,000 or 2,000 words. Input is by punched cards, and output is also by punched cards. The instruction word is ten decimal digits, the first two being an operation code, the next four normally a data address, and the last four the address of the next instruction to be executed.

The Naval Ordnance Laboratory expects to receive two 2,000-word *650* calculators in the near future. We, in the Applied Mathematics Division of Naval Ordnance, have meantime put some thought and effort into the preparation of automatic coding systems. Our aim, rather different in many respects from that of a business organization, is to make coding as simple and efficient from the standpoint of the human coder as possible. We have many varied problems, some of a rather involved nature, including many where it is difficult, if not impossible, to describe entirely the computational procedure before a great deal of experimental computation has been performed. This means that a continual recoding process is necessary before the problem preparation is complete. We also have many small problems which we would like to mechanize if the coding time is not disproportionately large compared to the computation time. On the other hand, we have a large volume of computing to do and hence do not feel justified in using an automatic coding scheme which would slow computation too much.

Table I

Code	Name	Description
00xxxx	No op.	Perform no operation. Ignore the address and take the next instruction sequentially.
01xxxx	Stop	Stop if a certain control panel switch is set to "stop." Otherwise treat as a no op.
10xxxx	Add	Add the quantity in xxxx to the previous result
11xxxx	Subtract	Subtract the quantity in xxxx from the previous result.
12xxxx	Reverse subtract	Subtract the previous result from the quantity in xxxx.
13xxxx	Multiply	Multiply the previous result by the quantity in xxxx.
14xxxx	Divide	Divide the previous result by the quantity in xxxx.
15xxxx	Reverse divide	Divide the quantity in xxxx by the previous result.
20xxxx	Store zero	Store zero in xxxx.
21xxxx	Store	Store the previous result in xxxx.
22xxxx	Replace address	Replace the address of the speed instruction in xxxx by the previous result.
30000x	Shift right	Shift right 000x places. Truncate without rounding.
35000x	Shift left	Shift left 000x places. Truncate high-order figures, but do not turn on overflow circuits.
40xxxx	Branch on zero	If the previous result is zero, take the next instruction from xxx.
41xxxx	Branch on plus	If the previous result is positive or zero take the next instruction from xxxx.
43xxxx	Branch on tally	(See tally.)
45xxxx	Branch on non-zero	If the previous result is not zero take the next instruction from xxxx.
47xxxx	Branch on overflow	If the previous result produced overflow, take the next instruction from xxxx.
48xxxx	Branch on non-tally	(See tally.)
49xxxx	Transfer	Take the next instruction from xxxx.
50xxxx	Tally	Compare the address of the previous instruction to the address portion of the tally instruction. If the two are not equal add 0001 to the address of the previous instruction and set up conditions so that branch on non-tally will cause transfer. If the address portion of the tally instruction is equal to the address of the previous instruction do not add one, but set up conditions so that branch on tally will cause transfer.
60xxxx	Insert plus	Replace the previous result by the quantity in xxxx.
61xxxx	Insert minus	Replace the previous result by minus the quantity in xxxx.
62xxxx	Extract address	Replace the previous result by the address of the instruction in xxxx.
66xxxx	Insert plus absolute	Replace the previous result by minus the absolute value of the quantity in xxxx.
67xxxx	Insert minus absolute	Replace the previous result by minus the absolute value of the quantity in xxxx.
700000	Read	Read the next card and store its contents in the proper locations as specified by the address field of the card.
7100xx	Punch	Punch the five quantities stored in memory locations 0330 to 0334 on a standard card. The two digits in the address portion of the punch instruction will be punched in the identification field of the card. They convey spacing and positioning information to the printer.
80xxxx	Compute function	Transfer control to the subroutine which begins at xxxx. When the subroutine has been completed, return to the main routine.
84xxxx	Search table	Compare the previous result to the tabulated values which are stored sequentially beginning at xxxx. Replace the previous result by address of the first tabulated value which is greater than or equal to the previous result.

With these considerations in mind we constructed a fixed point speed coding system and are currently working on a floating-point speed code. Investigations of the feasibility of an algebraic, or address-free, coding system are also being undertaken. The following is a brief description of the system already completed.

The fixed-point speed code is a single-address sequence-controlled coding system designed for use of the IBM *650*. Speed codes in this system are in the form of 6-digit words, the first two digits being an operation code and the next four normally specifying the memory location of a datum. This is a reduction of four digits over basic code since we do not in general specify the location of the next instruction. Data are held and computed with, in the form of 10-digit numbers with an algebraic sign. The decimal point is assumed to be located immediately to the right of the first digit. This choice of decimal point location was suggested by Lubkin,[5] and is believed by us to be the most natural and efficient form, and has been used very successfully by us in a general-purpose board for the Card Programmed Calculator.

For use with the speed code data are fed into the machine on a standard card format. Each card has five 10-digit data fields and a 4-digit address field. The five data are stored sequentially, the first going into the location specified in the address field of the card. Provision is made so that less than five quantities may be stored on any one card, if the coder wishes, without disturbing the contents of the storages which would otherwise be affected. Since blank columns are detected by the *650* as an error, we have instructed the machine to treat negative zeros as blanks, that is, nothing is stored from a field that contains −0000000000. If the coder wishes actually to store zero he punches positive zero. Instructions are entered into the calculator similarly to data, except that due to the shorter form of speed code ten speed instructions may be entered per card.

In coding a problem it is always assumed for binary operations, such as addition, subtraction, or division, that one operand is the previous result. For unary operations such as square root the operand is the previous result. Logical operations such as branches make their decisions on the basis of the previous result.

The 32 speed codes are shown in Table I. xxxx denotes some arbitrary drum location.

It will be noticed in the foregoing brief description that there is a good deal of redundancy in the branch codes provided. We have branch on plus and branch on minus, branch on zero and branch on non-zero; similarly, we have branch on tally and branch on non-tally. Obviously one from each of the aforementioned pairs would suffice. It is sufficient, in fact, to have in all only one branch code, say branch on minus. The redundancy actually employed, however, allows the coder greater flexibility in program construction. Thus, for example, if the positiveness of a quantity is to determine whether a given computation is or is not to be completed the coder would find branch on minus the most logical code to use. If, however, it is the negativeness of the quantity which is to determine action, the coder would want to branch on plus. If both branches are not provided the coder may find that extra instructions have to be inserted in the program.

A few highlights of the interpreting routine which is associated with the speed code follow. First, the 6-digit speed codes which the coder feeds into the machine are not stored as such. Actually each speed instruction is replaced by a 10-digit basic machine instruction which is the first instruction in the subroutine which the original speed code represented. This can be seen to lead to a very efficient system. The interpreting routine which must be employed during the course of computation is reduced to a series of four basic machine instructions which keep the computation in its proper sequence. Thus, we have almost gained the computational speed of a compiler without having lost the simplicity and storage-saving properties of an interpreter. This combined compiling-interpreting technique incidently allows the coder to switch back and forth from speed code to basic code at will, for since each speed code is translated into a basic code and will ultimately be used as such, we can insert basic instructions of any desired sort into a list of speed instructions without the use of any special device to indicate the switch. Whether or not this property will prove useful, however, is open to question.

For problems of the nature that are handled at Naval Ordnance, it is estimated that the computation time will be increased 15 to 25 per cent by using speed code rather than basic machine code. This, we feel, is a very favorable comparison in view of the advantages gained. The chief reason the comparison is so favorable is that the standard subroutines and interpreting routines have been minimum access, or optimum coded. That is, instructions and operands have been located on the drum so that they are just under the reading heads when they are needed. This arrangement is made possible by the nature of the basic instruction, which permits arranging the instructions on the drum in any desired arbitrary manner. Construction of a minimum access coded routine, however, is a time-consuming task, one which we would not undertake for every problem we compute. Thus, while speed-coded problems lose time due to the interpreting routine, they gain most of it back because of minimum access coding.

Equipment for Automatic Coding

Finally, consider one of the general problems in automatic code construction. There is a growing feeling in the field that calculators should be designed anticipating that automatic coding techniques will be applied to them. One suggestion for extra hardware designed to aid automatic coding was made by R. D. Rayan.[8] Mr. Rayan suggests a new form of auxiliary permanent memory which consists primarily of a photographic slide and a simple reading mechanism to go with it. Such a memory would be used to hold interpretive routines. It has the advantages of permanentness and relatively low cost of construction. Its separation from the regular memory protects it from erasure due to coding errors. Other pieces of hardware designed to ease the job of the coder are in actual use. For example, the *120* magnetic drum calculator of the Underwood Corporation has auxiliary equipment which enables the coder to count automatically the number of times a given sequence of instructions has been executed and to branch when some prescribed limit has been reached. It might seem a little out of place to propose extra equipment for a small computer but since only limited facilities are provided it is all the more important that we be able to use them as efficiently as possible.

The problem of what auxiliary equipment would be helpful in the design of automatic coding techniques is partly answered if we approach the problem from the viewpoint that automatic coding is the synthesis of one, possibly hypothetical, computer within another.[4] For input and output of our synthesized computer we employ the corresponding parts of the synthesizer; for memory we have that portion of the synthesizer's memory not occupied by interpreting routines. For control, however, we have no really convenient facility to use; we are forced to make the arithmetic unit of the synthesizer serve the double function of arithmetic unit and control. This means that we must intermediately store the results of arithmetic computations, while the arithmetic unit is being used as control. When the function of control has been completed the contents of the arithmetic unit must be stored and the previously stored arithmetic information recalled. This storing and recalling from memory obviously take both time and instructions. It also complicates the problem of designing the interpreting routines in the first place.

As a remedy an extra unit might be used by the interpreting routines to synthesize instructions. It should be separately addressable and the coder should be able to refer to it directly for an instruction without first storing its contents. This proposed unit would not have to be as complete as the arithmetic unit; it need be only one word in length, no provision being necessary for multiplication or division. It should be capable of adding and subtracting positive quantities, and have its own set of branch orders. It should have provision for right and left shifts. No provision need be made for roundoff. If the cost of such an extra unit were considered prohibitive, a reasonable compromise is the feature provided on the *650* where the arithmetic unit itself may be addressed for an instruction.

Assuming the existence of such an extra unit, it is estimated that the interpreting routines for the speed code outlined earlier could be constructed using about 10 or 15 per cent less memory, and that computation time for a typical problem would be speeded about 20 to 30 per cent.

References

1. THE MIT SYSTEM: COMPREHENSIVE, SUMMER SESSION, AND ALGEBRAIC, **C. W. Adams.** *Proceedings*, Symposium on Automatic Programming for a Digital Computer, U. S. Navy Mathematical Computing Advisory Panel, Washington, D. C., May 13–14, 1954.

2. IBM 701 SPEED CODING SYSTEM, **H. W. Backus, H. Herrick.** *Ibid.*

3. NEW YORK UNIVERSITY COMPILER SYSTEM, **R. Goldfinger.** *Ibid.*

4. PROGRAMMED SYNTHESIS OF DIGITAL COMPUTERS WITHIN DIGITAL COMPUTERS, **F. E. Heart.** Memorandum *M-2913*, Massachusetts Institute of Technology, Cambridge, Mass , 1954

5. TYPE 650 OPERATING PRINCIPLES. Form *22-6149-0*, International Business Machines Corporation, New York, N. Y., 1953.

6. DECIMAL POINT LOCATION IN COMPUTING MACHINES, **D. Lubkin.** *Mathematical Tables and Other Aids to Computation*, January 1948, pp. 44–50.

7. COMPILER METHOD OF AUTOMATIC PROGRAMMING, **N. Moser.** *Proceedings*, Symposium on Automatic Programming for Digital Computers, U. S. Navy Mathematical Computing Advisory Panel, Washington, D. C., May 13–14, 1954.

8. A PERMANENT HIGH SPEED STORE FOR USE WITH DIGITAL COMPUTERS, **R. D. Rayan.** *Transactions*, Institute of Radio Engineers, New York, N. Y., September 1954.

Discussion

Sol Gorn (Ballistic Research Laboratories): Has any work been done on automatic coding of the noninterpretive type—preparing the machine code before solving the problem?

L. D. Krider: We considered this and discarded it for a couple of reasons: the memory-saving features of an interpretive-type automatic code; and because of how relatively little time we do lose by using interpreting, due to the minimum access coding.

Sol Gorn: Has the Naval Ordnance Laboratory considered "editing generators" for standard large problems with many applications having many print-out requirements?

L. D. Krider: The answer to that is simple: No. We just haven't considered anything of that type yet.

W. E. Meyer (Convair, Forth Worth): How much storage is required by the speed code? Maximum number of instructions.

L. D. Krider: The speed code itself occupies approximately 350 locations on the drum. This does not include subroutines for computation of sine and cosine or square root, which have not been completed yet. It also does not include a tracing routine which I have outlined but haven't polished off yet, which will occupy another 75 locations.

All in all, when we add the trigonometric functions and tracing routine my guess is 500 to 600 memory locations of the 2,000.

(Essentially the same question was asked by E. L. Harder, Westinghouse Electric Corporation.)

H. Freeman (Sperry Gyroscope Company): On the average, what do you feel is the saving in time to the programmer when using the speed code, and what is the loss in time on the computer due to the longer program?

L. D. Krider: Offhand, with this fixed-point system you have about a half or a third as many instructions to write for what I have been calling a typical problem. I have been doing that a little tongue in cheek. All I did was file through some problems that we did have for CPC and count up the number of multiplications, additions, etc., and estimate the logical operations if it were quoted for *650* and I came out with a standard form of a typical problem. And for such a typical problem I think it will take about one half to one third as many instructions in this speed code system. Another advantage is that only six digits are used for a code instead of ten; although I can see where that might be a disadvantage.

Another advantage is that you get ten instructions per card where ordinarily there is not room for ten digit instructions on a card.

In regard to the loss in time due to the interpreting routine, on the typical problem which I have outlined it runs 25 per cent longer to compute this way. Multiplication and division come out just about the same regardless of which way you do them; that is why this method is so favorable, since our problems involve mostly multiplication. Addition takes about three times as long in the speed code as in the basic code.

Stanley Gill: Has any thought been given to reducing the space occupied by the interpretive routine by excluding parts not used in particular applications?

L. D. Krider: Yes. The trigonometric routines, for example, can be pulled on or taken off or put back on at will. The data I outlined in my paper, though, are at present planned to be on the drum permanently. One reason for this is that in optimum coding routines are so scattered around the drum that if you did pull part of them off it would not be of much use to the coder to know that he has a spot here and a spot there, and would lead, I am sure, to innumerable errors.

Miss B. H. Worsley (University of Toronto): In the fixed-point scheme what automatic safeguards are provided against exceeding capacity?

L. D. Krider: Once I had some safeguards plugged in there. By "exceeding capacity" I guess you mean accidentally addressing the interpreting parts and destroying them?

Miss Worsley: Partly that and partly concerning numerical overflow.

L. D. Krider: I took out all the checks to protect the interpreting routines largely because we are going to have both fixed and floating point. Perhaps it was poor judgment, but we decided that from problem to problem we probably would be changing back and forth between them and have to be loading in the interpreting routines anyway, so we save a few instructions and cut out the protection.

As to overflow, every operation is checked for overflow and if an overflow occurs the machine will stop and the console lights of *650* give you the address of the instruction which caused the overflow, unless you employ the branch-on overflow code. If you do employ branch-on overflow and the overflow was due to multiplication or addition you get the truncated result. If it was due to division you get zero.

J. R. Stock (Swiss Federal Institute of Technology): What work has been done to use the *650* to do its own optimum programming?

L. D. Krider: None as yet, although we have been thinking about it.

J. H. Brun (University of Michigan): Does your speed code use fixed or floating addresses?

L. D. Krider: Fixed addresses.

E. L. Harder: Have you considered the space requirements for floating point?

L. D. Krider: I would estimate that the speed code, including point, would take between 100 and 200 more drum locations. As I say, I am sticking my neck out because I haven't looked at it very closely. But I am pretty sure that is not too far off base.

Automation of Information Retrieval

J. W. PERRY M. M. BERRY F. U. LUEHRS, JR. ALLEN KENT

INFORMATION retrieval is a task that must be accomplished whenever records have to be consulted to provide needed facts. The task may be simple and easy as, for example, in consulting the telephone directory. In contrast, it may require many hours of patient persistence in a well-stocked library to identify and to locate those publications, patents, reports, and similar records that contain information needed in arriving at decisions in various fields of professional activity. This situation frequently arises, for example, when an electrical engineer is confronted by a design problem, when a lawyer must advise a client on a point in law, when a physician must decide on a course of treatment, or when a chemist must develop a process for a needed product.

The role of recorded knowledge in conducting research and development is particularly important to our industrial civilization, as illustrated by Fig. 1. Cost-conscious research management is becoming increasingly aware that successful research is not to be equated with the performance of experiments or the filling of laboratory notebooks with data, however accurate. Rather, research is the application of specialized knowledge to solve well-selected problems.

Information retrieval, considered from the point of view of purpose, enables written documents and similar graphic records to serve as an extension of human memory. To the extent that information retrieval fails to function, libraries and similar accumulations of records become morgues for accumulating embalmed knowledge. The tendency for the accumulation of knowledge to result in its becoming less and less readily available may be counteracted by applying a wide variety of techniques ranging from conventional filing, classifying, and indexing methods through various manual methods, e.g., hand-sorted punched cards, to application of more or less complex automatic devices. Which tools or methods are most appropriate for a given situation must be determined by careful evaluation of such controlling parameters as number of records already accumulated, accession rate of incoming records, rate of obsolescence of accumulated records, complexity of subject matter, breadth of field involved, purposes that the record collection is serving or should serve, present status of organization of the record collection, frequency of use, and urgency for promptness in providing needed information.

It is axiomatic that a system for making information available should be designed so as to provide an optimum margin of advantage in terms of benefits achieved and costs incurred. At the present time, estimation of benefits to be anticipated in a given situation from the automation of information retrieval is no easy matter. As experience continues to accumulate, it is hoped and anticipated that it may prove possible to treat the design of information retrieval systems as an engineering problem in much the same fashion as the design of a solvent recovery system in chemical engineering.[1] However, for the present, the design of digital equipment for automation of information retrieval must be based on rather general considerations.

Acquisition, Recording, Retrieval, and Use of Information

An information retrieval system, by its very nature, is linked, on the one hand, to the acquisition and recording of information and, on the other hand, to the use of information of pertinent interest in a given situation.

As is indicated in Fig. 2, the acquisition of information by observation and experiment either immediately involves or leads directly to a more or less thorough interpretation of observations in terms of existing concepts. In this way, a process of abstraction occurs, to a considerable extent at least, during the acquisition and recording of new factual information. For this reason, any description of an experiment is almost certainly incomplete. Certain observations may not have been made or they may have been left unrecorded as inessential or unimportant. Observations may have been interpreted and recorded in a form that leaves questions in the minds of later workers in the field. For such reasons, a deliberate repetition of an earlier experiment may sometimes be advisable. Even in such cases, however, the repeated experiment can be more expeditiously planned and conducted if the record of previous work is available.

The process of discerning and establishing relationships between phenomena, directly observed, and concepts of a basic, abstract, or general nature continues during subsequent preparation of reports, patents, and similar records and also during their abstracting and indexing. In this way, observational results are interpreted, analyzed, and recorded in terms of concepts which may vary from highly specific, e.g., an individual chemical substance, to highly general, e.g., energy.

The interpretation of original observations may involve considerable mathematical computation. Applications of digital computers to expedite or to facilitate such processing of data are outside the scope of this paper. The realm of activity with which we are concerned is the identification of documents and records that contain information of pertinent interest to a given problem.

Effective use of recorded information that has been identified as pertinent usually requires that it be processed in some fashion or other. Such further processing may be reducible, in some instances at least, to a set of routine operations that could be performed by machine operations. Situations of this type are encountered in banking, merchandising, actuarial, and similar operations.

It seems likely that the future will bring increasing extension of the automatic processing of information identified as pertinent. For example, it may prove possible in the field of tax law to employ automatic equipment to accomplish two purposes. The first would be identification of those statutes, regulations, rules, and court decisions pertinent to a given client's case. The second would be the application of the pertinent law to compute the minimum tax responsibility. Our discussion here is centered on the first type of purpose, namely, the identification of pertinent information. We shall be concerned, in short, with "memory machines" rather than "thinking machines."

Information Retrieval and Machine Capabilities

It is axiomatic that digital equipment can be designed to accomplish any well-defined arithmetic or logical operation or any routine built up from such operations. In theory, it might be possible, perhaps, to design a machine that could scan printed matter or examine other graphic records and then select records of pertinent interest without the need for

J. W. PERRY, M. M. BERRY, F. U. LUEHRS, JR., and ALLEN KENT are with the Battelle Memorial Institute, Columbus, Ohio.

a human expert to conduct a preliminary analysis of the recorded information. A machine for scanning and selecting printed documents would have to be able to perform a series of functions. First, it would have to recognize words, phrases, special nomenclature, and such symbols as the structural formulas of organic chemists and the wiring diagrams of electrical engineers. Recognition of words, phrases, and other symbols by the machine would have to be followed by their interpretation in terms of the information requirements to be met. Such interpretation would require both an extensive memory unit to make the meaning of words, phrases, and symbols available to the machine and also a logical processing unit to test and to detect relationships between the semantic elements that characterize the recorded information being scanned, on the one hand, and the information requirements to be met, on the other hand. A machine able to scan unanalyzed documents and similar graphic records would certainly not be a small digital computer whose modest cost should open up a wide market as an aid to searching and correlating accumulated files of recorded information.

The heart of the problem is how to define tasks to be performed by the digital equipment so that the mechanized searching and selecting system will provide optimum advantages. Particularly careful attention must, of course, be devoted to keeping costs down.

As shown in Fig. 3, equipment cost is not the only item of expense. In an operational system, a considerable investment must be made in analyzing an extensive file of information preparatory to machine searching. Methods for conducting such analysis must be designed so that costs involved in processing documents and similar records do not reach excessive levels. Simultaneously, the design of the equipment must be kept simple in order to avoid excessive construction, maintenance, and operating costs. To meet these two requirements, the same basic principles must apply both to machine design and to the analysis of information.

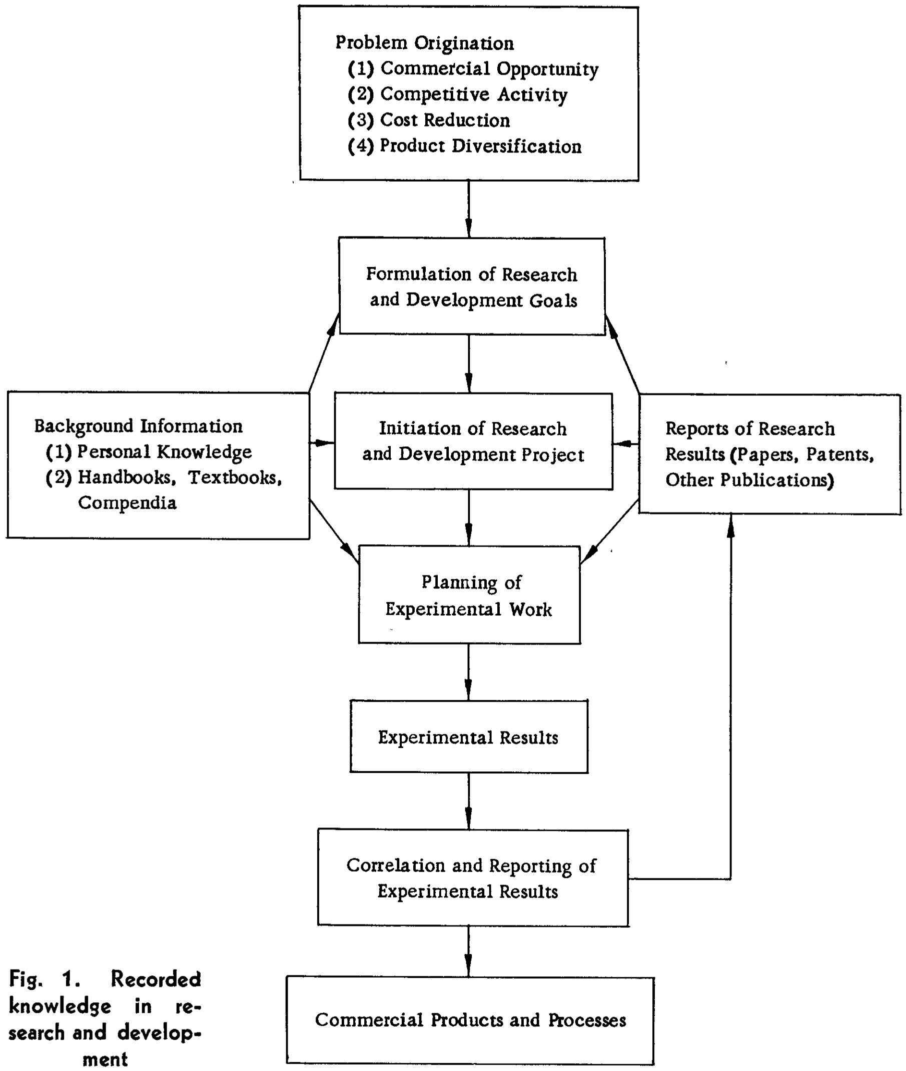


Fig. 1. Recorded knowledge in research and development

Information Retrieval and Class Definition

In seeking a common denominator of basic principles, it is instructive to recall to mind, first of all, that the recording and reporting of factual information in various fields of learning is based on concepts that range from the highly specific, e.g., 2–4–6 trinitrophenol, haematoma, manslaughter, to the broadly generic, e.g., aromatic compound, trauma, crime. Terminology plays a dominating role in recording, abstracting, indexing, and classifying recorded factual information. In seeking needed information, chemists, physicians, lawyers, and other professional men use the same terminology to formulate their information requirements. Such definitions of information requirements will be found on examination, in the majority of instances at least, to involve several terms that are used to characterize the type and scope of information required.

Terminology used to designate various concepts (spatio-temporal entities, abstract concepts, attributes, processes, conditions) may be used to identify one important type of criteria that characterizes both the subject content of records and also the scope of information requirements. It is also important not to overlook a second type of criteria, namely, the relationships between concepts denoted by terminology. Such relationships, in ordinary writing, are denoted by sentence structure or with the aid of prepositions or similar connectives. To make this second type of criteria available as reference points in defining searching operations, it is necessary that important relationships be identified, defined, and explicitly recorded when analyzing information preparatory to machine searching.[2]

The identification and selection of specific records in terms of characteristic criteria may be formulated as the definition of a class whose members are characterized by the specified criteria. The identification of records of pertinent interest may be achieved by matching the characteristics of the subject matter of the records with the characteristics that define the scope of a given information requirement. The matching operations, when formulated on the basis of class definition, provide guidance both for the design of searching machines and also for carrying through the analysis of information.

It should be recalled, in this connection, that the theory of class definition imposes no restriction on the character-

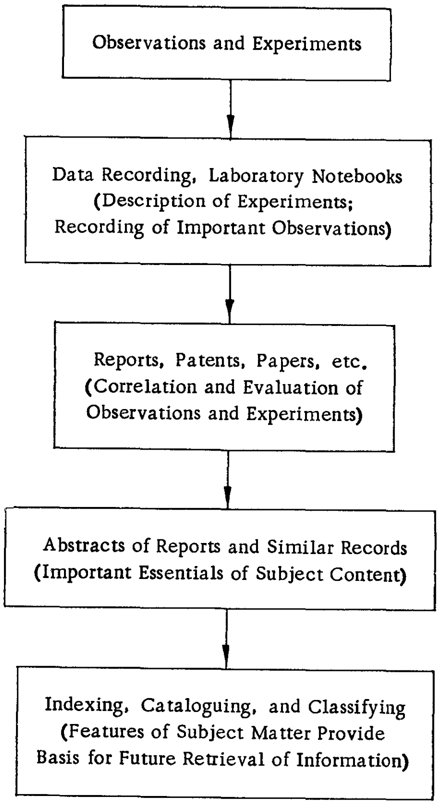


Fig. 2. Steps in acquisition, recording, and processing of information

istics that may be taken into account or, equally important, disregarded when defining a class. The theory of class definition also recognizes the possibility of one class being a member of another class. Furthermore, a class is no less validly a class if it contains only one member or even none at all.

It is instructive to consider how the theory of class definition might be applied to selecting cards on each of which an abstract from "Chemical Abstracts" has been recorded. Perhaps the simplest possibility would be to select those cards that bear an abstract containing some predetermined symbol such as the letter "t" or the numeral "5." Or it might be specified that the abstract shall contain a certain sequence of symbols as exemplified by "nitro," "thermo," "oxidation," or "CH_3." A closely related possibility would be to specify some interrupted sequence of symbols such as "d—tion" or "C_2H_5–H." Such criteria, involving single symbols or sequences of symbols, could be used to detect the presence of certain key words or formulas within an abstract. It is important, of course, to specify that false sequences would not be generated by the interaction of the last few symbols in a word or formula with the first few symbols in the next word or formula scanned by the machine. This means specifically that the machine must be designed so as to detect and respond to certain runs of symbols. One type of run is exemplified by words in ordinary language or by chemical formulas. The next higher order of run would consist of a number of successive words, formulas, or similar symbol sequences to which meaning is attributed. This type of run would correspond to a phrase or clause in ordinary writing. The next higher order of run might be regarded as similar in nature to sentences. Further, still higher orders of runs might be regarded as corresponding to paragraphs, chapters, or complete messages. Symbolically, a run corresponding to a paragraph might be exemplified as follows:

¶SPWxxxWxxxxxWxxPWxxWxxxxxxWxxx
xWxxxPWxxxSPWxxxxWxxxxWPS¶

where

¶ = paragraph start
S = sentence start
P = phrase start
W = word start
x = any individual symbol, e.g., letter or numeral

Note that the start marks designate both the beginning and end of the various orders of runs. As already pointed out, a class of abstracts may be defined by requiring that a given word or formula in the abstract shall contain some symbol or sequence of symbols. Similarly, it may be necessary to identify an abstract as belonging to a class containing information of pertinent interest by specifying that it shall contain at least one higher order run that is characterized in terms of some specified component. For example, in defining the scope of a search we may need to specify that abstracts of interest shall contain at least one sentence characterized by a certain phrase.

It is perhaps obvious that the effective usefulness of a searching machine would be sharply limited, if it were restricted, when scanning recorded messages, to detecting only one criterion of the type corresponding to a single symbol, e.g., "t" or to a sequence of symbols, e.g., "nitro" or "C_2H_5–H." A multiplicity of such criteria may be required to designate the scope of a search. The magnitude of this multiplicity must be held within reasonable bounds, however, as otherwise the cost of providing a large number of detecting units may make the machine excessively expensive. Furthermore, the time and effort involved in setting up the machine to detect an excessively large multiplicity of such search criteria could exceed reasonable limits. These considerations are of importance when deciding how to express the subject content of records in encoded form for machine searching. Before considering this point further, an additional functional requirement of digital equipment for information retrieval should be considered.

Definition of classes becomes much more flexible and effective in literature retrieval if logically defined configurations of criteria can be used to identify abstracts of pertinent interest. To denote such relationships precisely, capital letters, A, B, C, etc., will be used to designate individual symbols or sequence of symbols that characterize words. The three basic logical relationships may then be specified as follows:

1. Logical product
   All of several criteria are required to be present
   Symbolized as $A \cdot B \cdot C \cdot D \ldots$
2. Logical sum
   Any one or any combination of several criteria is required to be present
   Symbolized as $A+B+C+D \ldots$
3. Logical difference
   At least one criterion is required to be absent
   Symbolized as $A-B$

Definition of classes in a form useful for information retrieval may also involve complex logical relationships, as exemplified by:

$$(A+B)\,(C-D)+E$$
$$(A \cdot B \cdot C)+(E \cdot F \cdot G-H)$$
$$A(B+E+F)-(H \cdot K)$$

A fourth relationship specifies that two or more criteria shall be found arrayed in a given order. If we use <> to symbolize such specification of order, then the requirement that the logical product $A \cdot B \cdot C$ is to involve the criteria in that order might be symbolized by $<A \cdot B \cdot C>$. Specification of order may involve more complex logical relationships as exemplified by

$$<(A+B)(C+D)(E-F)>$$
$$(<A-B>+<C-D>)(E+F+G)$$

These logical relationships have been specified and illustrated in terms of criteria designated by capital letters. Such criteria, it will be recalled, are individual symbols or sequences of such symbols. By specifying logically defined configurations of such criteria, classes of words may be defined. Thus, the logical product $A \cdot B \cdot C$ could be used to define words characterized by the presence of the letter "a," the letter sequence "tion," and the interrupted sequence "ox–d." A class of abstracts may then be defined as containing one or more of the words that are characterized by $A \cdot B \cdot C$. This class of words might be designated by A' and other similarly defined classes of words might be designated by B', C', D', ... This opens up the possibility of defining a class of abstracts containing one or more phrases characterized in terms of component words of the classes A', B', C', D', ... As before, such definition of phrases may be based on specification of combinations of words as expressed by logical product $(A' \cdot B' \cdot C' \cdot D' \ldots)$, logical sum $(A'+B'+C'+D'+ \ldots)$, logical difference $(A'-B')$, or more complex logical relationships, as exemplified by $(A'+B')(C'-D')+E'$ or $(A' \cdot B' \cdot C')+(E' \cdot F' \cdot G'-H')$.

Similarly, phrases (symbolized by A'', B'', C'', D''....) may be used to define classes of sentences (symbolized A''', B''', C''', D''' ...) and the latter in turn used to define classes of paragraphs (symbolized A'''', B'''', C'''', D'''' ....).

Application of the basic principles of class definition in this way provides a common basis for the functional requirements of the searching machine and for expressing the subject matter of documents in machine searchable form.[3]

In summarizing machine characteristics from the point of view of required functions, it is perhaps obvious that a machine searching setup will consist of two major units.

One of these two units is an appropriate medium for recording the abstracts or similar summaries that are to be searched. The selection of an appropriate medium is an engineering question. In general, any medium capable of recording digital information might be considered as a possibility. Some of the more obvious possible media are magnetic tape, punched cards, Teletype tape, photographic film (either in continuous or discontinuous form).

The other major unit in a machine searching setup is the device which scans and identifies which abstracts or similar summaries refer to records of pertinent interest. This searching device must be able to perform a number of functions that might be summarized as follows:

1. Conversion of patterns used to record symbols into corresponding patterns of pulses.

2. Detection of pulse patterns used to record individual symbols or sequences of symbols. (This is equivalent to detection of search criteria designated by $A,B,C,D,$— in the preceding discussion.)

3. Detection of the beginning and end of runs of symbols where such runs may be of different order corresponding to words, phrases, sentences, paragraphs, and complete messages in ordinary writing.

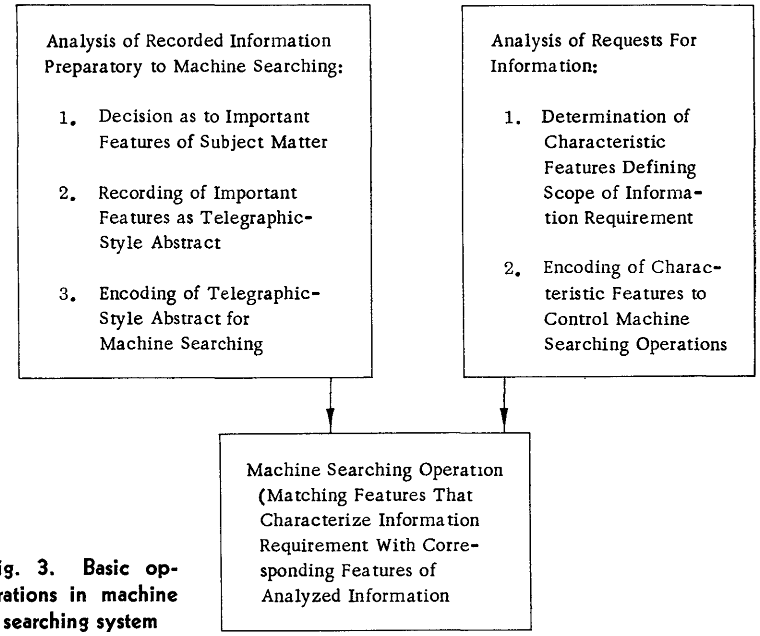


Fig. 3. Basic operations in machine searching system

4. Detection of configurations of logical relationships, either simple or complex, as they may be set up in defining search requirements. Such logical relationships may involve characterizing words in terms of component symbols or sequences of symbols, phrases in terms of component words, sentences in terms of component phrases, paragraphs in terms of component sentences, and abstracts or similar summaries in terms of component paragraphs.

5. Notification of what records have been identified as being of pertinent interest. Such notification may be accomplished, for example, by removing a card from a file. As a recent paper has pointed out, if the "card" is a piece of photographic film, it may bear in microform a readable copy of the abstract or even several pages of printed or other graphic material. Notification of identification may thus be combined with the furnishing of a record of pertinent interest.[4]

These operations can, of course, be performed by a general-purpose computer. A study of the programming required and the effective operational speeds that could be attained supports the conclusion that, for automation of information retrieval on an operational basis, important advantages would be achieved by equipment specially designed and constructed to provide the operational functions summarized.[5]

Analysis and Encoding of Subject Matter

For illustrative purposes, the searching of abstracts as published by "Chemical Abstracts" has been referred to from time to time in previous discussion of class definition as the basis for specifying the functions of digital equipment for searching extensive files of abstracts that relate to complex subject matter. A broad range of flexibility in defining and conducting searches is provided by the afore-specified machine functions. They are, however, obviously inadequate to permit information retrieval to be based in a straightforward and efficient fashion on machine scanning of abstracts or similar summaries written in the English language. The necessity for taking into account the extensive range of synonyms, near-synonyms, and similar alternate terminology available in English and the even wider range of possibilities for alternate phrasing would make it excessively difficult to formulate search requirements so that a high probability of retrieving pertinent information could be assured.

An interpretation of the subject content of documents and records is required to render explicit those aspects of meaning that are to be used as reference points for defining the scope of information requirements and for conducting corresponding machine operations. Such aspects of meaning are of two types. One is ordinarily designated by terminology that denotes spatio-temporal entities, attributes, actions, and processes, conditions, and abstract concepts. When such terminology is used to express the subject content of documents, various important relationships may be involved. For example, it may be important to specify that substance A rather than substance B has a certain property. Such relationships, which are usually designated when writing English by word order, by prepositions, or similar connectives, constitute the second type of important aspects of meaning.

Space limitations prevent a thorough discussion of semantic and systematic considerations involved in rendering explicit, in appropriately coded form, these two types of aspects of meaning. Decisions made in designing a system to meet a given set of requirements must be based on careful consideration of a range of parameters, such as number of documents involved, rate of accession of new material, complexity of subject content, type of information requirement, discriminating ability needed, urgency in meeting information requests, problems involved in distribution of documents identified as pertinent, security regulations, etc.

Searching equipment having functional characteristics as previously discussed provides wide latitude in designing an information retrieval system to meet the exigencies of a given situation. Varying requirements may be taken into account by appropriate decisions as to the following:

1. The degree to which the abstracts, as prepared and encoded for machine searching, provide a more or less detailed analysis of the subject matter of documents or other records. The abstracts may range from brief general annotations to detailed summaries.

2. The extent to which the meaning of words, phrases, and similar terminology is rendered explicit when encoding abstracts. Considerations of cost and convenience make it advisable to record decisions as to the semantic analysis of terminology in a code dictionary. The encoding of abstracts on a routine basis is thereby facilitated. The code dictionary is also helpful in insuring consistency in the analysis of the meaning of new terms and in assigning codes to render their meaning explicit.

3. The type and character of relationships that are taken into account and rendered explicit when encoding abstracts. These relationships may be defined in a broad, general fashion or as specifically and narrowly as may be required by the purposes to be served.

In arriving at decisions, it is necessary to take into account the fact that the discriminating power of a mechanized information retrieval system depends primarily on the extent to which aspects of meaning are rendered explicit by abstracting and encoding. Both generic and specific aspects of meaning can be expected to be important and to require careful consideration. Complexity in a system for analyzing and encoding subject matter inevitably means increased cost, and justification in terms of advantageous benefits must be provided.

Conclusion

Advantageous application of computer-type equipment to automation of information retrieval is determined by ability to analyze the subject content of documents and records in terms of those aspects of meaning that are important in identifying information of pertinent interest. Semantic problems require careful attention in applying automation of information retrieval to meet the requirements of a given situation. In solving these problems in terms of machine operations, the theory of class definition provides the basic framework of reference. Within this framework, there is a wide range of choice as to degree of detail in abstracting and as to rendering aspects of meaning explicit during encoding of abstracts.

Decisions must be made so that automation of information retrieval may provide an optimum margin of advantage as measured in terms of benefits achieved and costs incurred.

References

1. MACHINE LITERATURE SEARCHING. VIII. OPERATIONAL CRITERIA FOR DESIGNING INFORMATION RETRIEVAL SYSTEMS, **Allen Kent, M. M. Berry, F. U. Luehrs, Jr., J. W. Perry.** *American Documentation*, Cleveland, Ohio, vol. 6, no. 2, April 1955.

2. NEW TOOLS FOR THE RESURRECTION OF KNOWLEDGE, Staff Report. *Chemical and Engineering News*, Washington, D. C., vol. 32, 1954, pp. 866–9, 891.

3. MACHINE LITERATURE SEARCHING. VI. CLASS DEFINITION AND CODE CONSTRUCTION, **J. W. Perry, M. M. Berry, Allen Kent.** *American Documentation*, Cleveland, Ohio, vol. 5, 1954, pp. 238–44.

4. THE APPLICATION OF THE KODAK MINICARD SYSTEM TO PROBLEMS OF DOCUMENTATION, **A. W. Tyler, W. L. Myers, J. W. Kuipers.** *Ibid.*, vol. 6, 1955, pp. 18–30.

5. ELECTRONIC DIGITAL MACHINES FOR HIGH-SPEED INFORMATION SEARCHING, **P. R. Bagley.** Master of Science Thesis, Massachusetts Institute of Technology, Cambridge, Mass., 1951.

Discussion

W. D. White (Airborne Instruments Laboratory): Could you comment on "Zator" coding of "Uniterms"?

J. W. Perry: It will not be easy for me to comment on matters involving proprietary interests when, in the past at least, the existence of a large measure of disagreement by the proprietors with me has been stated by them on a number of occasions.

There is some uncertainty in my mind whether "Uniterm" is, in effect, a trademark or the name of a specific system. If the latter, then according to what I have read about it, the Uniterm system appears to be based on the philosophy of word-indexing or word-coding. This philosophy might, perhaps, be summarized by saying that the purposes of analysis of information for subsequent retrieval can be achieved by (1) taking the words or, on occasion, phrases found in documents, and (2) using these directly either as index entries or, more generally, as reference points for conducting searches. "You let your material index itself," as this philosophy is sometimes expressed. This seems virtually certain to open the door to all sorts of difficulties with synonyms, near-synonyms, and with alternate modes of expressing ideas in general. The English language's remarkable flexibility, for which Winston Churchill has expressed so much admiration, can only lead to trouble if indexing is attempted without careful control of the use of words and terminology. This conclusion is supported by extensive experience in using conventional alphabetized indexes, particularly poorly constructed indexes. A careful and precise control of terminology, as achieved, for example, by "Chemical Abstracts," is essential to achieving the full measure of effectiveness that is possible with a conventional alphabetized index.

Achieving the full measure of effectiveness with other information retrieval systems also requires that at least equally careful attention be devoted to the meaning of terminology. It is perhaps obvious that the purpose of using automatic equipment to facilitate information retrieval is to expedite accomplishment of the same job for which alphabetized indexing is so widely used. As we tried to make clear in our paper, the essential step in this job is matching the characteristics of an information requirement with the characteristics of the subject content of the documents that are on file. The necessity of accurately defining such characteristics, particularly when they are to be the basis for defining automatic machine operations, is, I am sure, obvious to all. This means that the meaning of words must be accorded very careful attention indeed when indexing subject matter and this general principle holds true regardless of the system, technique, or device used to accomplish retrieval. I stress this point, as I fear there may have been misunderstanding or even disagreement regarding it between myself and the proprietor of Uniterm.

Now to speak of the "Zator" method. Calvin Mooers, proprietor of Zator, is well aware of the necessity of paying careful heed to the meaning—and indeed to the scope of meaning—of terminology. Calvin will not recommend that you use words in a careless or random fashion. He will tell you, and I agree with him completely on this point, that it is necessary to select your terminology very carefully and to use it in a thoughtful fashion if you are going to set up a satisfactory information retrieval system.

Perhaps what I am trying to say is this: If you apply the Uniterm system, taking care, as does Dr. Charles Zerwekh of Houston, to use terms with close attention to their meaning, then, if the parameters of your information problem are of the right magnitude, the Uniterm system can provide you with advantageous service.

The Zator system is based on the observation that, on the edge of a hand-sorted punched card, or on a similarly edge-notched card, you have a multiplicity of locations. However, this multiplicity is very small by comparison with the number of combinations of locations that can be set up. For example, the number of combinations based on taking 20 locations three at a time is very much larger than 20. Furthermore, it is possible to punch simultaneously on the edge of a card a number of such combinations without having undue trouble with undesirable interactions during sorting operations. "Ghost" combinations resulting from undesirable interactions don't, as experience has shown, become a real nuisance as long as not more than a third of the locations have been punched. Each combination of locations can be ascribed its own meaning corresponding to an index entry or to a characteristic of subject matter of a document. In this way a mechanically performed search can be directed to any one meaningful characteristic or to any combination of the same. But it is perhaps obvious that, here again, your must choose your characteristics carefully and use terminology with precision. If you do this, then I suppose you might consider this an application of Zator coding to the Uniterm system. But, in my opinion, it would be the Zator system straight out.

C. B. Poland (General Electric Company): What study has been made of access times to the data in the computer? In particular, what speed is a probable minimum?

J. W. Perry: Access time and minimum machine speed are variables which depend on other parameters. If the subject matter of your documents is of such nature that they can be divided up into mutually exclusive—or nearly mutually exclusive—categories, then it may be possible to arrange things so that ordinarily an information requirement may be met by searching one or a few categories, in other words, only a fraction of all the documents on file. This can work out in such a way that access time requirements may be met in large measure by simply not having to search a large portion of the file at all. And this, in turn, means that your machine speed can be considerably slower than if you were compelled, for a majority of the information requirements, to search through everything. As may be obvious, the two questions that have just been asked have no simple answers. Rather, the answers to these questions in a given situation must be decided by analyzing requirements to be met, the type of subject matter involved, the number of documents, and similar parameters. Once this analysis has been accomplished, machine requirements can be defined.

The more severe the machine requirements with respect to access time and speed, the higher, in general, you can expect machine costs to be.

P. Bagley of the Massachusetts Institute of Technology (MIT) wrote a master's thesis on this general subject in 1951.[5] If you care to look into these questions in more detail, I'm sure you'll find his thesis interesting and informative.

H. F. DeFrancesco (NSA): Have you performed test scanning on any of the available computers, and if so, please describe the experience and the particular times involved.

J. W. Perry: We were working around to doing this at MIT in connection with the Bagley thesis when, for a variety of reasons, I decided to shift to Battelle. The conclusions in the Bagley thesis were that we would be able to scan, I believe, 5,000,000 documents an hour with a properly designed small computer having the characteristics specified in the paper. The various 5,000,000 documents were considered to have, on an average, 35 index entries, that is, 35 "words" in computer parlance, and it was further assumed that each word would be recorded by not more than 35 digital bits. This is as close as we ever came to an actual trial on an existing computer.

In his thesis, Bagley also investigated the possibility of using the MIT Whirlwind computer to conduct information retrieval along the lines that I outlined in the paper. Programs were worked out, though never actually tested. Even so, there could be little doubt as to what the results would have been. Bagley concluded that effective searching could be accomplished by Whirlwind at about a third the effective rate of an appropriately designed punched-card machine—that is to say, a punched-card searching machine having the operating characteristics outlined in the paper just presented. The reason is that, in a computer such as Whirlwind, you have an arithmetic unit which must make all the decisions. For each decision, data must be fed into the arithmetic unit, the decision made, and the decision placed in storage pending a final over-all decision based on a multiplicity of individual decisions such as establishing the identity of characteristics of information requirements and documents. Before selecting a document as being of pertinent interest a multiplicity of such identifications of characteristics may have to be accomplished, and each identification will almost certainly require a multiplicity of tests for decision by the arithmetic unit. These successive tests constitute a sort of Indian-file parade of information through the arithmetic unit.

This series type of operation turns out to be extremely wasteful in requiring many machine operations to achieve one simple result, e.g., establishing the identity of corresponding characteristics of an information requirement and of the subject content of a document. In particular, many operations, and much time, must be devoted to shuttling information back and forth between the arithmetic unit and the machine's internal storage units. The way out of the dilemma is to design your searching machine—your special-purpose computer—so that identifying operations are performed in parallel rather than in series. This requires establishing a bank of comparator units, each of which may be set to detect some one encoded characteristic of the subject content of the documents being searched. Such identification of characteristics would then be followed by another unit inspecting the decisions made by the comparator units to determine whether the identified characteristics of a document correspond to the logical configuration of characteristics that define the search requirement. The paper just presented reviewed this relationship between identifying operations directed to characteristics and detection of a logical configuration of characteristics. Bagley, in his thesis, pointed out how a plugboard might be used in connection with detecting a logical configuration of characteristics. With this approach, an electronic searching device could be constructed having an effective searching speed several powers of ten greater than attainable with Whirlwind.

Mr. Callen (National Research Council of Canada): What qualifications would you expect in an indexer, and where would you find such people?

J. W. Perry: On talking with various friends in the computer business, they sometimes tell me that the ceiling on the market for computers is determined by the number of available persons having enough mathematical background to do the necessary programming, in particular the interpretation of problems in mathematical terms compatible with computer operation.

In anticipating a similar situation in this field of information searching and correlating, our thinking runs along the following lines. The generation of an encoded abstract for machine searching can be broken up into a series of steps. The first step is someone looking over the document and deciding what is important. This requires a person who has at least a good professional understanding of the subject matter and also a good feeling for the status of the field in question. A person so qualified is in a position to make good decisions as to what is likely to be important, in a given technical paper, to others working in the field to which the paper pertains.

The indexing of the "Chemical Abstracts" is performed by people who have a PhD in chemistry or its equivalent.

Once a person versed in the field to which documents pertain has analyzed them as to important characteristics, the jobs of interpreting these characteristics in appropriate index terms, organizing the index terms into appropriate form, and encoding them, if necessary, should be assigned to persons other than our subject analyst, if costs are to be held to a minimum. It seems virtually certain that problems involving synonyms, near-synonyms, and the meaning of terminology, in general, can be solved by basing the coding of terminology on its semantic analysis. In this way it should prove possible largely to relieve the person analyzing and indexing documents of terminological difficulties. It would be necessary, however, to set up telegraphic-style abstracts to express the subject content of documents. Simple rules for organizing such abstracts are now ready for testing on a pilot-plant scale.

The encoding of properly organized abstracts can be reduced to a clerical routine. For IBM cards, working with recently designed searching equipment, it is convenient to use 3-letter codes to designate basic units of meaning sometimes referred to as semantic factors. Corresponding digital encoding for electronic computer operation would certainly be equally simple to develop.

Your question really goes to the heart of the matter. Permit me to rephrase your question this way: "How do we get large numbers of the right kind of telegraphic-style abstracts prepared?" (Their encoding is, relatively, a much simpler matter.)

Message Storage and Processing With a Magnetic Drum System

A. P. HENDRICKSON G. I. WILLIAMS J. L. HILL

IN large-volume Teletype network operation, such as the periodic collection and dissemination of weather observations or the filing and processing of aircraft flight plans, certain basic clerical operations can be mechanized. Mechanization of this kind greatly improves the effectiveness of a given number of human operators and traffic controllers, since their attention can be concentrated almost completely on the decision-making portion of their work.

An initial exploratory equipment having certain clerical properties was designed by the joint efforts of engineering personnel of the Technical Development and Evaluation Center of the Civil Aeronautics Administration (CAA)* and the Engineering Research Associates Division of Remington Rand, Inc. In September of 1953 a finished equipment was completed and delivered to the Technical Development and Evaluation Center at Indianapolis, Ind. Since that time it has been operated in conjunction with other developmental equipments in an evaluation of a new system of air traffic control data processing. In this service it is operated 8 hours each day receiving and filing messages. These messages contain the latest weather observations at some 100 weather stations, and as each message is received it replaces the previous message for that station. The processing of flight plan messages on a large scale will begin after other suppliers have installed certain auxiliary equipment. This paper will be confined to the specific properties of the physical equipment.

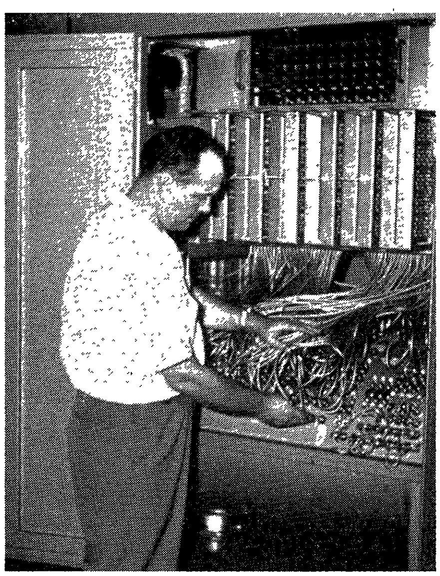


Fig. 1. Storage drum and head-switching circuitry

Basic Properties

The most fundamental property required of this equipment is an ability to perform a clerical filing operation. Messages must be accepted when offered, stored until desired or until relevant to a current situation, and then delivered to a requestor. To be practical, such an equipment should fit into an existing system with a minimum of translating mechanisms, while retaining properties amenable to a future, higher speed system. Therefore, a message assembly and speed converting mechanism is provided between the processing equipment and each of the connected input and output communication lines. This mechanism allows messages received at 75 words per minute to be presented to the processing equipment at 230,000 words per minute. In the case of outgoing messages an inverse speed change occurs.

The processing speed of the equipment will handle saturation traffic on about 60 bidirectional 75-word-per-minute lines. For short-distance communications with an associated set of display mechanisms, the equipment has a set of terminals suitable for a 1,050-word-per-minute message rate. The processing speed of the experimental equipment will provide saturation traffic for about three of these higher speed terminals.

The storage drum has capacity for 2,000 messages, 750 of which can be up to 230 characters long; the other 1,250 messages may contain a maximum of 115 characters. Average message-processing time does not exceed 450 milliseconds, and the processing repertoire includes the following operations:

1. Verify each incoming message for correct format.
2. Accept the verified message for filing (or other action).
3. File the message if no duplicate is currently on file.
4. Find and cancel a specified message.
5. Cancel a specified message and file new copy in its place.
6. Find and read out a specified message.
7. Prepare the proper message of reply to the requestor.

Items 1, 2, and 7 are part of each processing operation. One request message may specify only one action among items 3, 4, 5, or 6. In addition, an internally clock-stimulated action calls for periodic examination of a particular portion of every stored message. A copy of each message examined which has a time entry falling within the next 20 minutes of actual time is automatically prepared for transmission. The address to which each message is transmitted is a part of the stored message. This action takes precedence over all incoming messages.

Processing Sequence

Each incoming message is assembled at its arrival rate in a recirculating magnetic drum revolver having a maximum capacity of 230 characters. When the complete message is at hand, and at the completion of any processing action then occurring, a high-speed transfer to the processing section occurs and the new processing operation is begun.

The first operation in the processing section is performed by comparing specific characters of the incoming message with stored values which occupy corresponding character positions. This verification operation insures that numbers and letters are in acceptable positions in the format, and that certain key characters are present. Failure to pass the verification test stops all further processing of this message and causes it to be typed out at a supervisor's printer with a code symbol indicating the reason for this action.

To minimize problems of initial use of such an equipment in day-to-day message processing, the form of the input and

* The resulting equipment was sponsored by the Air Navigation Development Board (ANDB) which is an agency established by the Departments of Defense and Commerce in 1948 to carry out a unified development program aimed at meeting the stated operational requirements of the common military/civil air navigation and traffic control system. This project (contract *C 13ca-410*), sponsored and financed by the ANDB, is a part of that program. The ANDB is located within the administrative framework of the Civil Aeronautics Administration for housekeeping purposes only. Persons desiring to communicate with ANDB should address the Executive Secretary, Air Navigation Development Board, Civil Aeronautics Administration, W-9, Washington 25, D. C.

A. P. HENDRICKSON, G. I. WILLIAMS, and J. L. HILL are with Remington Rand, Inc., Engineering Research Associates Division, St. Paul, Minn.

output messages was carefully planned to stay as close as possible to convention and precedent. The resulting compromise caused the message format to consist of three distinct sections: a prefix, a criterion, and the balance of the message. The associated communications network uses the first three characters of the prefix to direct each processed message to its intended recipient. On incoming messages, these characters are consumed before reaching the storage and processing equipment. On outgoing messages, a similar prefix is necessary to provide this directing information to the switching mechanisms of the communications system. Therefore, the prefix of each incoming request message supplies three characters to be used in the prefix of the reply message. This feature permits one station to call for an action by the processing equipment and to have the reply sent to any other station in the network. The internally stimulated messages obtain these characters from the stored message.

The prefix of the incoming message also contains characters analogous to the commands used in a computer. These select one of nine different processing procedures available.

The criterion portion of each message is the key by which it is identified, and consists of the 18 character positions following the prefix. The criterion contains three subdivisions: a 3-character place identifier, a 4-numeral time identifier, and a 9-character aircraft or weather report identifier. With the restrictions imposed by the nature of the items identified, about 3×10^{17} different messages can be identified; however, a message is arbitrarily considered a duplicate if the first three and last nine characters are identical with those of another message. This restricts the number of different messages to about 2×10^{14}.

To insure an orderly procedure for message cancellation, it is necessary to reject duplicate messages. As a consequence, each filing operation is preceded by searching the identifiers of all 2,000 messages to insure that the message to be filed has no uncancelled duplicate in the file. The requirement that this search include all 2,000 message storage positions makes filing a message the longest processing operation. During this search, the first suitable vacancy is detected and remembered. If the search fails to discover a duplicate, the equipment control system directs the storage of the submitted message into the remembered vacancy. As soon as storage is effected, a check is made by reading a copy of the message from the storage drum and transmitting it to an outgoing terminal to be sent to the addressee specified by the characters of the incoming message prefix.

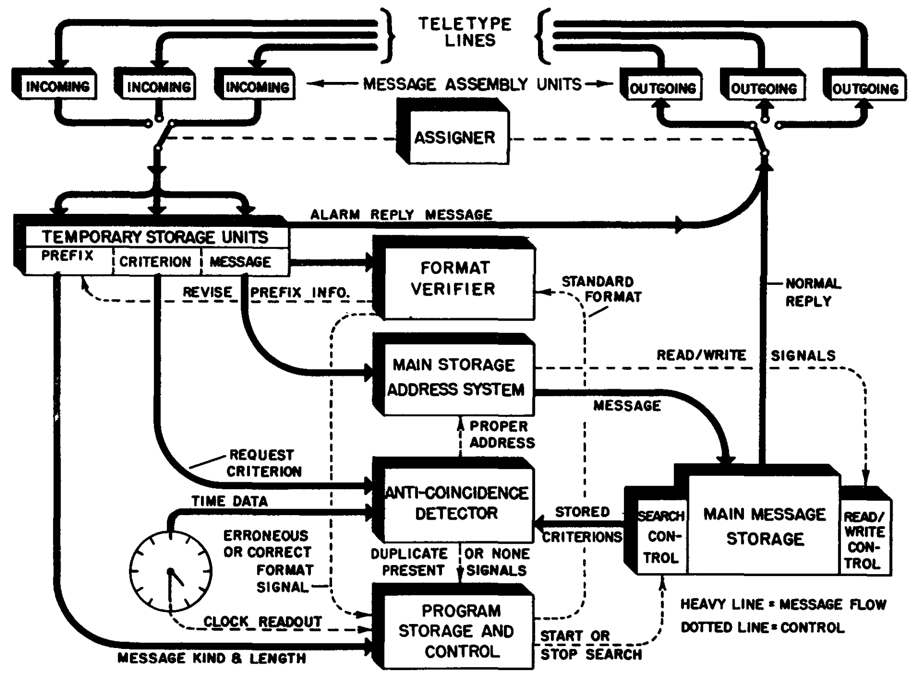


Fig. 2. Condensed block diagram of system

A mechanism similar to the incoming assembling and speed-changing device operates at each outgoing terminal to receive the reply message at high speed and to transmit it at a rate appropriate to the connected communication system. This may be either 75 or 1,050 words per minute.

During the processing attendant to acceptance, filing, and acknowledgment, all operations except the search are performed at 230,000 words per minute, which is the basic equipment data rate. To conserve processing time, the search is conducted on five messages at a time. This results in a 1,140,000-word-per-minute searching rate. The net effect is that the user begins receiving the reply message roughly one-half second after the last character of his request message was received by the processing equipment.

Message Storage

The storage device used to retain the 2,000 messages is a large magnetic drum. This drum, shown in Fig. 1, is 22 inches in diameter and has 275 tracks devoted to message storage. This array of tracks subdivides into 55 bands of five tracks each, each band providing capacity for 5,750 Teletype characters. The drum rotates continuously at 1,190 rpm and produces all timing and control signals used in the message processing. Its surface also provides the mechanism for altering the speed of transmission between the associated input/output lines and the processing section of the equipment.

The various functions of the equipment will be understood best by following several message types through the processing and storage routine. Fig. 2 is a condensed block diagram which shows the major functional units required for message processing.

Assignment and Verification

The message assembly units, at the top of Fig. 2, assemble the incoming messages at the 75-word-per-minute rate and deliver them to the processing equipment on demand at 230,000 words per minute. To avoid having a reply message ready in the processing section and no outgoing message assembly unit available to accept it without delay, the message assembly units are assigned in pairs. The assigner performs this function, and also selects the incoming message assembly unit which shall be serviced next.

When the assigner detects a loaded incoming message assembly unit, and when the processing equipment is next idle, the assigner seizes an unoccupied outgoing message assembly unit and transfers the message to the temporary storage units of the processing section. The prefix is stored in a flip-flop register where two characters are translated to determine the message kind and length. These characters establish the processing program for the remainder of the message.

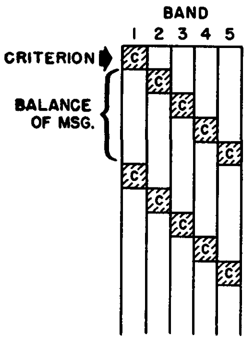


Fig. 3. Information pattern on drum surface

The criterion of the message is stored in a magnetic drum revolver, since these characters must be available in a cyclic manner for comparison with the criterion of each stored message. This revolver provides 115 bits of storage and requires less space and equipment as a magnetic drum revolver than with any other currently available storage means.

The message proper, including the criterion, is stored in another drum revolver holding 230 characters (1,150 bits). The contents of this revolver are compared by the verifier with a fixed format stored on the magnetic drum. The verifier thus ascertains that the form of the request message meets the rules. It is not established whether the text is correct, but a significant check is the examination of certain characters. For example, one character in each message must be selected from the six acceptable message kind-identifying characters used. A length character must occupy a specified character position to designate long or short messages. Two message types are further verified by requiring that specific character positions be figures. Three message types are checked further to assure that none of the first 20 characters are blanks and that the message contains no more than 20 characters.

Without question, the cancellation request is the message of greatest consequence. For this reason, a second kind-identifying character is supplied in place of the length character which is not used with this request. Good protection is thus assured against the possibility of a garbled message kind-identifying character causing an undesired and undetected cancellation.

When the verification results in a rejection, the kind-identifying character is changed to "X." The request message is then sent to a supervisor who determines the reason for rejection, and initiates a corrective action. When the verifier determines that the message format is acceptable, the normal message-processing routine is continued.

Processing a File Request

The next processing operation on a file request is a search to assure that a duplicate is not already on record. The search time is kept to 400 milliseconds (eight drum revolutions) by a special procedure in which no significant time is lost searching through irrelevant data. Fig. 3 illustrates the method used. The criterion information upon which the search is made embraces 18 characters.

Since the search proceeds cyclically, an integral submultiple of both the long and short message spaces must be allocated to contain the criterion information. The appropriate allocation is 23 characters, this being one fifth and one tenth of the length of short and long message spaces. By organizing the drum contents so that these allocations are displaced by 23 characters in adjacent bands, characters from the criterion revolver can be applied continuously to one input terminal of the coincidence detector while another terminal receives characters from the message criteria in successive bands. A switching circuit using germanium junction diodes was developed to switch the reading amplifiers from one band of heads to another. Although four unsearched character periods are available for head switching, the switch used operates in less than 32 microseconds (μsec), or about four fifths of a character period. After four such switching operations, the reading amplifiers are switched back to the first band of information to read the criteria of the following group of messages in that band. In the case of the long messages the switching operation is performed nine times before returning to the original set of heads.

The search for a similar message on the storage drum must result in one of two conditions. If a message with a duplicate criterion is found, the kind character in the prefix is altered and the original file request message is returned to the originator. If no duplicate is located, the file request message is accepted and recorded on the drum in the next revolution. On the revolution following the writing operation, a copy of the stored message is read out and sent to the originator. This procedure assures an exact copy of the recorded message for examination by the originator.

The location on the storage drum in which a message is filed is determined by a concurrent supplementary searching operation. A stored mark, occupying one of the five unused positions of the 23-character criterion allocation of each message storage position, indicates the status of the contained message. For processing purposes, three such status marks are used. One mark designates a newly filed message upon which no additional action has been taken. A second mark indicates an en route message, one that has been the subject of certain additional processing; and the third mark, a cancelled message. The last also designates a storage location which may be used for new filings.

The object of the supplementary search for a location in which to store a new filing, is to locate the first cancelled status mark in that portion of the storage drum corresponding to the indicated message length. The address of this message position is retained by a set of thyratrons and flip-flops (the address captor) during the remainder of the search for a duplicate message. When the search is completed and no duplicate message has been found, the message is both written on the drum at, and read back for checking from, the address held by the address captor.

Processing a Read-Out Request

A read-out request program is included to permit obtaining a copy of any filed information. This processing procedure uses only a brief request message con-

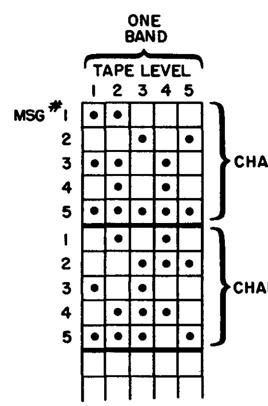


Fig. 4. Character interlace on drum

sisting of the message prefix and criterion. The read-out request message is examined by the verifier and passed on to the coincidence detector for comparison with the message criteria in storage. The coincidence detecting means is actually an anticoincidence detection, since this form of logic assumes the message to be correct unless a disagreement appears, thereby requiring only comparison gates and one memory element per simultaneously examined message.

Characters from five messages are examined during each 43-μsec character period in the criterion revolver. Fig. 4 illustrates the character interlace used on the drum. Note how the first character of the first message is followed by the first character of each of four other messages. Following through the diagram, note that the first character of the fifth message is followed by the second characters of the same five messages. This arrangement equates the several requirements for high drum surface utilization and the circuit economies of the slower character-processing circuits with the 450-millisecond processing time. In many portions of the processing circuitry, the request message is handled in a serial-character/serial-bit mode at 114,000 bits per second, the drum's basic pulse rate.

Upon locating the desired message in a read-out operation, the search control equipment is stopped with the angular location of the message recorded in the address captor. On the next drum revolution the message is transferred to the outgoing message assembly unit along with the prefix from the temporary storage register.

Outgoing Message Handling

To satisfy the requirements of the Teletype transmission system, the outgoing message assembly unit dispenses one character at a time on demand to a Teletype transmitting distributor. When the Teletype distributor has completed each revolution, it requests a new character from the outgoing message assembly unit which is actually another magnetic drum revolver. This revolver contains, in addition to the message text, an index mark which identifies the next character to be transmitted. When the marked character next appears at the terminals of the reading amplifiers, it is transferred to a buffer register and the index mark is moved to the succeeding character. This process repeats at about seven operations per second under the control of the Teletype sending unit. Since an entire 230-character message stored in the revolver passes approximately 13 times for each character sent, ample opportunity exists to keep the buffer register filled.

A high-speed combination incoming-outgoing message assembly unit is also provided. This unit is a high-speed link between the message storage and processing equipment and a special display board located in the air traffic controller's room. The message assembly unit providing this service is almost identical to its lower speed counterparts. In the case of the higher speed unit, a character is received or dispensed each cycle of the revolver, or approximately once each 10 milliseconds. Thus the surface velocity of the magnetic drum establishes the actual character transmission rate of approximately 1,050 words per minute.

Processing a Cancellation Request

In handling message data where real time is involved and the safety of personnel is a consideration, reliability and exactness are of paramount importance. The cancellation request message carries this form of responsibility as it indicates that the aircraft has landed or the flight was cancelled. An uncancelled message means that the aircraft is in the air, overdue, or that the responsible person has neglected to inform the CAA of the completion of a flight.

For this reason a cancellation request has the most stringent verification of all. Every character in the request message must be correct to accomplish a cancellation. Each character not checked by the verifier is compared with the corresponding criterion character of the messages in storage and must agree down to the last bit with the criterion of the selected message to accomplish the cancellation.

Supplementary Functions

To provide a permanent record of messages handled by the message storage and processing equipment, all cancelled messages are sent to a printing tape perforator to be retained as a permanent record. Additionally, every 10 minutes the time from an internal clock is automatically recorded on this tape to provide an approximate indication of the time of message cancellation.

A manually initiated program is also included in the equipment repertoire. This program allows the supervisor to inspect periodically the contents of the storage system by printing out all the messages on the drum in the status classification he selects. For example, all of the en route messages can be selected, thus revealing the flight plans of all aircraft in the air, missing, or those neglecting to cancel their flight plans. Once a flight plan has been printed out, the status mark is so altered that it cannot be printed out again until it takes a new principal status condition; in the case of an en route message, this would occur after cancellation.

Another purpose of this manual program is to permit cancellation of messages stored with garbled identifiers. These cannot be located by the usual means; however, knowing the nature of the garble, the supervisor can construct a suitable cancellation request.

Automatic Processing Actions

A distinguishing equipment feature is the clock-controlled program. Each 10 minutes a clock signal initiates a search of all 2,000 drum storage positions to locate those flight plans whose estimated time of departure is recorded as falling within the next 20-minute interval. This is done by comparing the four time characters in the criterion portion of each message with a Teletype-coded time signal produced by the clock equipment. If an imminent flight plan is located, it is prepared for transmission to the specified departure airport by the outgoing message assembly unit. Its status mark is altered at this time to convert it to an en route message. After each such action the search is resumed until all 2,000 storage positions have been inspected. In this manner, a search is initiated for a 20-minute period each 10 minutes. This overlapping search procedure insures picking up any relevant flight plans filed since the previous clock read-out.

Weather Message Provisions

Two additional programs are included to permit the equipment to be useful for processing weather data. One of these programs controls both the initial filing and the alteration of weather observation reports. The other program controls the read out of filed weather data. The inclusion of these two programs has permitted an effective test of an improved system of weather data collection and dissemination.

Operating Experience

During this year of evaluation, this equipment has seen approximately 3,200 hours of on-line operation. In this time 2.1 per cent of the original 1,900 electron tubes have been replaced and 1.7 per cent of the original germanium diodes have been replaced. Periods of as long as 250 consecutive operating hours have been obtained without equipment failure or parts replacement.

Discussion

W. H. Deering (American Telephone and Telegraph Company): What are the limits as to the number of Teletype lines which may be connected to the drum? What is involved to add additional lines?

G. I. Williams: The number of Teletype lines which can be connected to the equipment and still utilize its speed characteristics are determined by the number of message-assembly units which you connect.

The basic system is capable of handling approximately 60 75-word-per-minute lines at saturation traffic.

B. Glazer (Columbia University): What is the physical form of the drum revolver?

G. I. Williams: The recirculating drum revolver consists essentially of three heads, one a writing head, further down the track a reading head, and beyond that an erasing head.

The output of the reading head is connected to the reading amplifier, which in turn is connected back into the writing circuit. This effectively gives you a drum in this equipment which is one fifth of the circumference of the larger drum and gives you an access time which is five times shorter than the major drum.

G. E. Gourrich (Telecomputing Corporation): Do you have any information on the type and frequencies of garbling in the Teletype messages?

G. I. Williams: This information is not readily available to us since we haven't been concerned with the evaluation of the equipment. However, the Civil Aeronautics Administration maintains fairly high quality lines for their Teletype transmission and I expect that the garble is considerably less than you normally encounter.

Alex Orden (Burroughs Corporation): Is the equipment used to co-ordinate flight plans both before aircraft take-off and during flight?

G. I. Williams: The present equipment as it is designed follows the flight plan through to the estimated time of departure and at that time it is changed to an en route flight plan. The system is designed for 24-hour operation so that a flight plan after 24 hours will look like it is on the following day. It is up to the supervisor to request periodically a read-out of all of the en route messages. Thus, he sees obviously from the flight plan that if the estimated time of departure is up to 24 hours previous to the actual time, something has happened to this aircraft.

A. Robinson (Columbia University): Can you summarize the key causes of failure? And, what read-record system is used and what is packing density in bits per inch?

G. I. Williams: The principal failures in the equipment thus far have simply been tubes and diodes. The distribution of tubes has been quite general across the various types, no specific type causing extreme failure.

The diodes have been failing chiefly because of humidity, because they were the General Electric *1 N 52*, which was not considered hermetically sealed 2 years ago.

The read-record system is the return to zero type and the packing density is 83 bits per inch, 16 tracks per inch axially.

Nicholas Albanes (International Business Machines Corporation): How many drums described are used to store the 2,000 messages and what is the bit density?

G. I. Williams: Only one drum is used for all of this storage, and approximately two thirds of its surface is used for message storage. The rest of it was used for the revolvers, control tracks, etc.

John Fonduk (General Electric Company): Do you ever assemble a complete message in a vacuum-tube register?

G. I. Williams: To assemble a complete message in a vacuum-tube register would not be practical in equipment of this type since the message may contain up to 1,150 bits. If this were done in flip-flops with our present techniques it would require 2,300 vacuum tubes for this alone, and it doesn't provide you any increase in speed since this is a serial machine. Thus, the revolver is the more satisfactory way of storing this.

W. H. Deering (American Telephone and Telegraph Company): What safeguards are used against overloading both as to the message length and message capacity?

G. I. Williams: An overloading occurring from a message which is too long will be rejected by the verifier. No message may contain more than 115 characters if it is indicated to be a short message; no message may contain more than 230 characters if it is indicated to be a long message. The message will be returned to the originator shortened to these two lengths if it exceeds these lengths. However, if the machine has the short message section full and we receive short messages, it will automatically deposit a short message in the long portion of the drum. When the long portion of the drum is full it sounds an alarm, stops the equipment, and notifies the supervisor.

The storage capacity that was chosen for this particular equipment was based on quite an extensive survey of the requirements and we believe that it exceeds the requirements needed in the very near future at least.

F. E. Heart (Massachusetts Institute of Technology): Could you discuss the problem of human error at the transmitting-endtraining?

G. I. Williams: This was one of the principal reasons for constructing this equipment. You will note that we said that this was an evaluation equipment and it is not only evaluating equipment but evaluating this system of data handling. One of the problems that has to be solved, or looked into, is this matter of having human operators who can type messages accurately to a fixed format which the verifier will not reject.

F. E. Heart: What happens if the format is wrong, and what is the percentage of wrong messages?

G. I. Williams: If I understand the question, this is what happens when the format of the incoming message does not agree with the format required by the system. If this is true, the message is rejected and returned to the originator.

To answer the second part, I couldn't give you any information on the percentage of wrong messages because we haven't been evaluating the system.

F. E. Heart: Is the system documented? Is experience with the system documented?

Is the message assembly unit a commercially available device and is it documented?

Has the Teletype equipment itself been satisfactory for this application?

G. I. Williams: Is the system documented? There is a complete instruction book available on the system. Also, we have some printed literature available describing it briefly.

The experience with the system has been only briefly written so far since it has not been possible to evaluate it fully with regard to flight plans.

The message-assembly units can definitely be made a commercially available device. The system used requires a magnetic storage drum and if you have another drum in your system this conveniently could be the same drum with revolver tracks provided. If you do not have the drum in the system you have the problem of synchronization. Perhaps this can be taken care of by a simple buffer register which will take up slack.

The evaluation of the Teletype equipment I wouldn't care to comment on because we are not in the Teletype business.

Analysis of Business Application Problems on IBM 650 Magnetic Drum Data-Processing Machine

J. M. BOERMEESTER

THIS discussion will consider how the John Hancock Mutual Life Insurance Company viewed the problem of how it might use the International Business Machines Corporation (IBM) type *650* machine to fulfill part of a responsibility of considerable magnitude. This responsibility may be stated as follows: Determine each year the amount of dividends, which will be payable in cash or otherwise credited to the policy holders, for each of 4,000,000 ordinary life insurance policies. The type of administrator normally charged with fulfilling this responsibility should be viewed as one who possesses merely a superficial knowledge of electronics and, at most, only a broad general knowledge of the principles concerning the use of modern machine equipment.

When the type *650* machine was announced, the John Hancock Company used a closely integrated system for calculating dividends for a particular class of 1,400,000 policies which were issued under its Monthly Debit Ordinary branch. This system, which is still being used pending further developments, requires extensive manual-checking operations to test the results obtained from the usual array of machines which include sorters, collators, reproducers, tabulators, and type *604* calculators. Checking is required for some 10,000 possible unit-dividend groups which vary with respect to plan of insurance, issue age, and year of issue. Under this system a considerable amount of supervision of the clerical as well as the technical staff is necessary.

In reviewing this problem, the company evolved a system under which dividends for this class of policies may be calculated with a minimum of supervision by making a single pass of the punch-card file through the type *650* machine. Under this system, the policy number order would not be disturbed while the time required with one such machine might be on the order of one fifth of that required under the older system.

For those who are not familiar with the theory of allocating life insurance dividends, the following is a brief description of one of the methods, the so-called Contribution Method, as it is applied by the John Hancock Mutual Company. The three basic dividend elements which arise under the Contribution Method are:

1. A return from excess interest earnings.
2. A return from experiencing a favorably low death rate.
3. A return from experiencing relatively low administrative costs.

The result of this concept may be expressed in a practical dividend formula which in broad terms is as follows:

$$\begin{aligned} {}^{j}{}_{n}F_x{}^{p} &= \text{dividend payable at the end of the } n\text{th policy year} \qquad (1)\\ &= R_i \text{ (reserve of policy)}\\ &\quad + R_m \text{ (amount at risk)}\\ &\quad + \text{(expense margin)} \end{aligned}$$

where

x = age at issue
p superscript denotes a particular plan
j superscript denotes particular scale of benefits
R_i = return rate from interest earnings
R_m = return rate from mortality experience

The terms of equation 1 assume different values for life plans and special plans which have graded benefit scales. When we first approached the challenge to program a dividend-calculation procedure for this class of business, we already had in existence a scale of unit dividends. Accordingly, we decided to see if we could reproduce this pre-existing scale by programming the precise formulas used on desk machines and type *604* calculators. These pre-existing formulas may be expressed generally by the following desk-machine type equation:

$$ {}^{j}{}_{n}F_x{}^{p} = {}^{j}\pi_x{}^{p} \cdot {}_{n}a_x + {}_{n}b_x + {}_{n}e_x{}^{j} - h^{j}{}_{x+n} + \frac{\alpha_1{}^{p} \cdot \overline{N}_w}{\overline{N}_x - \overline{N}_w} \qquad (2)$$

where in terms of elemental functions:

$$ {}^{j}\pi_x{}^{p} = k_1 \frac{\overline{M}_x{}^{j} - \overline{M}_y + \alpha_2{}^{p} \cdot D_y}{\overline{N}_x - \overline{N}_w}$$

$$ n^{a}x = k_2 \cdot \overline{N}_x \cdot {}_2f_{x+n-1} + k_3 \cdot {}_1f_{x+n-1} - j_n$$

$$ n^{b}x = {}_1f_{x+n-1} - k_4 \cdot \overline{M}_x \cdot {}_2f_{x+n-1} + h_x$$

$$ {}_{n}e_x{}^{j} = k_4(\overline{M}_x - \overline{M}_x{}^{j} - \overline{M}_{x+n} + \overline{M}^{j}{}_{x+n}) \cdot {}_2f_{x+n-1}$$

y = age at maturity
w = age when premiums cease
$\alpha_i{}^{p}$ is a variable dependent on plan alone
k_i is a constant

For purposes of programming, equation 2 was expanded to the following 27-term formula:

$$ {}^{j}{}_{n}F_x{}^{p} = k_1 \left[\frac{\overline{M}_x{}^{j} - \overline{M}_y + \alpha_2{}^{p} \cdot D_y}{\overline{N}_x - \overline{N}_w} \right] \times [k_2 \cdot \overline{N}_x \cdot {}_2f_{x+n-1} + k_3 \cdot {}_1f_{x+n-1} - j_n] + k_4[\overline{M}_x - \overline{M}_x{}^{j} - \overline{M}_{x+n} + \overline{M}^{j}{}_{x+n}]_2f_{x+n-1} + \left[{}_1f_{x+n-1} - k_4 \cdot \overline{M}_x \cdot {}_2f_{x+n-1} + h_x + \frac{\alpha_1{}^{p} \cdot \overline{N}_w}{\overline{N}_x - \overline{N}_w} - h^{j}{}_{x+n} \right] \qquad (3)$$

where all terms are elemental functions.

Some 550 elemental functions, apart from orders, were stored initially in the type *650* registers.

We found that the programming for equation 3 could be done quite comfortably for the following range of variables with respect to premium-paying plans except the Family Income Plan.

$0 \leq x \leq 75$ for the range of issue ages
$20 \leq y \leq 100$ for the range of maturity ages
$20 \leq w \leq 100$ for the range of ages when premiums cease
$2 \leq n \leq w - x$ for the range of durations from date of issue to dividend due date
9 values of p, the plan code
6 values of j, the graded death-benefit code
2 values of $\alpha_1{}^{p}$ } variables depend upon plan
2 values $\alpha_2{}^{p}$ } code

After using 400 registers for storing the orders required to program equation 3 alone and discovering registers to spare, we next incorporated a search program which would give the dividends for the Family Income Plan. We also incorporated a program to calculate dividends for paid-up policies.

In the course of programming for equation 3 we decided to incorporate the following steps to edit data:

1. Test for improper plan.
2. Test policy number sequence.
3. Test for improper issue age.
4. Test for improper issue date.
5. Test for improper policy year duration.
6. Test for improper class of business.
7. Test for improper dividend account.
8. Test that amount of dividend lies within a specific range.

Discovering that we still had registers to spare, we decided to incorporate a program to provide for checking results for 168 broad plan-issue age-duration groupings for which confidence limits can

J. M. BOERMEESTER is with the John Hancock Mutual Life Insurance Company, Boston, Mass.

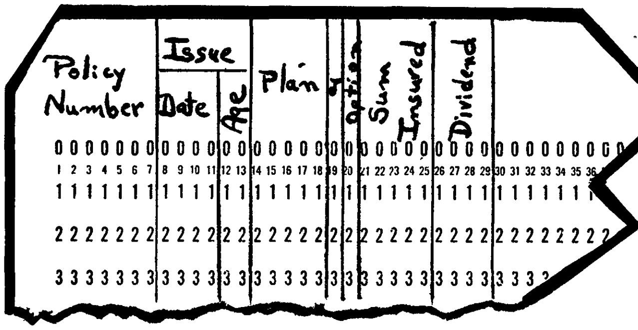


Fig. 1

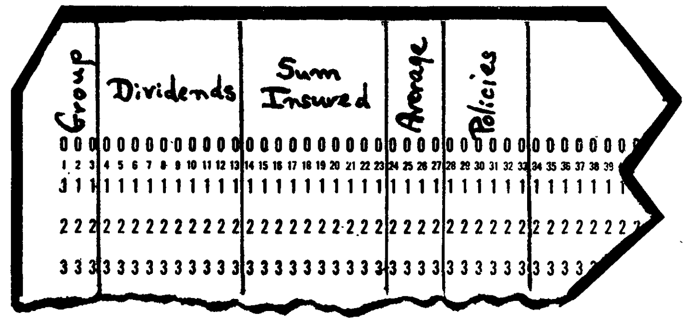


Fig. 2

be established. For each of the 168 check groups, totals were stored for policy count, accumulated sum insured, and accumulated dividend. One of such groups provided for a grand total check. Upon an order to clear these registers, the program provided for obtaining the average dividend per $1,000 sum insured for each group.

In review, the programming concerned the following card types:

1. Answer card for policy data (Fig. 1).
2. Answer card for summary check data (Fig. 2).

The input card for policy data has the same form as card 1, except that dividend field is not required.

The following outline shows the general pattern of the flow of steps which were required:

1. Test type of card.
2. Test for proper dividend option and policy number sequence.
3. Test for proper plan. If Family Income Plan, search for unit dividend and proceed to step 14.
4. Test for proper age range and proper issue date.
5. Store ${}_1f_{x+n-1}$, ${}_2f_{x+n-1}$, w, $\overline{N}_w$. If paid-up plan, compute unit dividend and proceed to step 14.
6. Test for proper duration.
7. Store $\overline{N}_x$, $\overline{M}_x$, $\overline{M}_{x+n}$.
8. Test for graded benefit-type code and store $\overline{M}^j{}_x$, $\overline{M}^j{}_{x+n}$ and $h^j{}_{x+n}$.
9. Store $(\overline{N}_x-\overline{N}_w)$, j_n and h_x.
10. Compute ${}_na_x$, ${}_nb_x$ and ${}_ne_x{}^j$. If endowment plan, proceed to step 12.
11. Compute ${}^j\pi_x{}^p$, $\dfrac{\alpha_1{}^p\overline{N}_w}{\overline{N}_x-\overline{N}_w}$, ${}^j{}_nF_x{}^p$. Proceed to step 14.
12. Store D_w, $\alpha_2{}^p$, $\overline{M}_w$.
13. Compute ${}^j\pi_x{}^p$, ${}^j{}_nF_x{}^p$.
14. Test magnitude of ${}^j{}_nF_x{}^p$.
15. Compute total dividend $=S\cdot{}^j{}_nF_x{}^p$ where S = sum insured.
16. Store results in summary registers. Punch answer cards and return to starting order.

Up to this time, the type *650* machine has not been used to authorize actual payments to policy-holders. Preliminary tests have shown that calculation time is approximately 1 second per policy. A variation of the method being discussed appears to be most promising. Due to a change in requirements recently made in our administrative philosophy, we are reprogramming to obtain a desirable by-product, namely, the policy cash value. Under this variation which we are testing, it appears that dividend calculations for many more plans may be programmed from elemental functions. Among these plans are the Family Income, Retirement Income, Joint Life, and Return Premium plans.

The program under discussion, however, is applicable only with respect to particular issues associated with a specific class of business. Parallel programming has been performed but not completely tested for other issues. Also, we were perhaps somewhat fortunate to be able to express the dividend scale as voted by the board of directors in terms of formulas which do not require extensive tables of empirical values.

It is gratifying to see a machine of this type in anticipation of the availability of more powerful equipment. For these many years the administrator whose activities have centered in part around punch-card systems has found that he could not possibly devote the time required to understand completely the complex skills which are so necessary to integrate all the phases of work. The introduction of this type of machine can change this picture materially.

The programming for this dividend calculation responsibility for the *650* machine was done without benefit of any formal training. The only aids were an instruction manual and help of an International Business Machines representative who clarified relatively few numbers of questions regarding the response of the machine. The broad scope of the operation was first sketched by the administrator who had only a formal 2 weeks' course in programming. The operation was then entrusted to a mathematically inclined staff member who had never had any formal training in programming and to his assistant who tested the program on paper by carrying through specific problems. When the programming was first tested on the machine, it ran without apparent error. The essential quality which appears to be required by people who are to do programming is the possession of reasoning ability, common sense, and enthusiasm rather than technical ability alone.

A curious and inspired administrator can understand the machine. If he understands the machine, he may possibly become a more effective administrator of an appropriate domain by critically reviewing programming which may be presented to him for approval.

Discussion

E. H. Friend (New York Life Insurance Company): Why did the programming group maintain the file in policy number order rather than valuation order? Does not a seriatim handling require considerable extra processing time?

J. M. Boermeester: In answer to this I would like to say that our company prefers to maintain its registers in policy number order so we can make easy reference to them for questions which may arise, and that in the previous operation the policy number order was disturbed. Under this concept which we have employed today the order of the cards is not disturbed, there is no extra handling, and therefore requires not extra time but time which I believe is on the order of one fifth of that now required.

H. O. Rohde (Minneapolis-Honeywell): Can you estimate how many man-hours of programming were required to put the illustrated problem on the machine?

J. M. Boermeester: This programming was done by one person full time and another person part time over a course of, I would say, 3 months. We didn't take any precise time on it. That was our first venture. I would say that we were probably very, very slow. We were feeling our way along and any figure I give to you now would perhaps have no meaning a year from today.

William Miehle (Burroughs Corporation): In the abstract of the paper in the program, it says, "Analysis. . . .illustrates the percentage use of various basic functions of arithmetic and logical decisions." Please explain this.

J. M. Boermeester: Dr. Petrie, maybe you know something about this.

G. W. Petrie, III: Originally it was hoped that in the presentation we would have a very comprehensive system of evaluation as to the percentage breakdowns on the arithmetic instructions and logical decisions and more statistical data. This paper is in the process but has not been completed. Instead of presenting partial results we felt that all of you would be much happier to hear of one actual case in complete detail such as the one that has just been presented to you.

M. Saslow (Airborne Instruments): What are the estimated dollar savings to be gained by this installation of the *650*?

J. M. Boermeester: This is a question on which of course nobody expects a precise answer from me. However, I would say that there are other elements in here, questions of administration which have not been analyzed and the question of speedup in time on which we have not made any precise estimate.

L. Flynn (Curtis Publishing Company): The calculation time of 1 second per dividend, does this mean 1,000,000 seconds for 1,000,000 policyholders?

J. M. Boermeester: When I went to school $1 \times 10^6 = 10^6$, yes. This time is apart from emergency breakdowns; it does not include time for downtime.

Small Digital Computers and Automatic Optical Design

N. A. FINKELSTEIN

THE photograph reproduced in Fig. 1 is an example of good optical imagery. The picture is crisp and considerable detail is resolved all over the area, even at the extreme edges and corners. Fig. 2 is a poor picture. The reasons are obvious. While the central region is still sharp and full of detail, the rest of the area is fuzzy and ill-defined. The quality definitely deteriorates as we move further and further out from the center. These two photographs were taken with the same camera, at the same exposure, under the same subject conditions with two different lenses of the same focal length. Fig. 3 shows the lenses in schematic form, essentially a section through the lens along the axis. The poor picture was taken with the upper lens, a simple biconvex element. The good picture was taken with the lower lens, a very-well-known design called a Tessar, containing four elements, two of which are cemented together. The difference between the two pictures shown obviously is related to the difference between the two lenses which took them. The techniques which lead us from poor picture to good picture, from simple biconvex lens to multielement Tessar, form the province of optical lens design.

The problem of the lens designer is to combine elements of different curvature, thickness, and refractive index in such a manner as to approach perfect imagery of the class of objects to be placed before the lens. In perfect imagery each point in the object is transformed into a corresponding point in the image without distortion or blurring; of course this condition can only be approached.

Probably the most important tool in lens design is geometrical ray tracing, a technique in which the paths of light rays emanating from a point in the object are traced through the several lens elements following the laws of geometrical optics to ascertain the manner in which these rays recombine in the image. In the ideal lens all the rays from each point of the object would recombine at corresponding points in the image as shown in Fig. 4. In general, the rays will not recombine as

N. A. FINKELSTEIN is with Bausch and Lomb Optical Company, Rochester, N. Y.

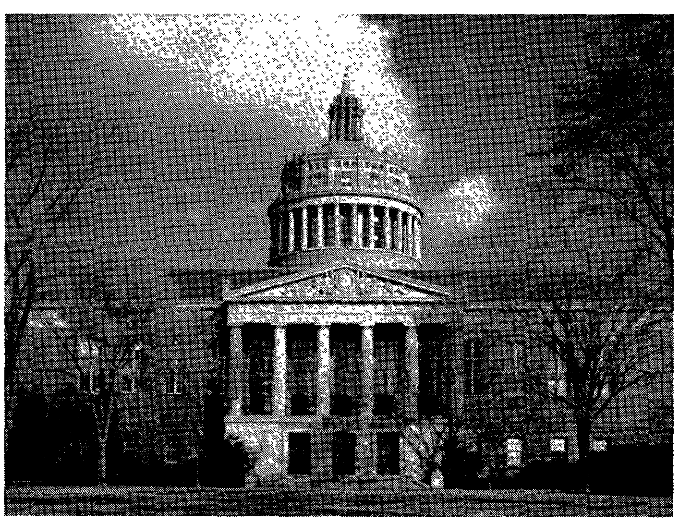


Fig. 1. Photograph taken with Tessar lens of 190-mm focal length at f/4.5

Fig. 2. Photograph taken with simple biconvex lens of 190-mm foca length at f/4.5

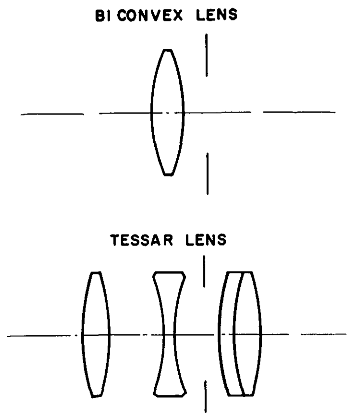


Fig. 3. Optical schematics of simple biconvex lens and Tessar lens

shown in this illustration and the resultant image is then said to contain aberrations. One type of aberration, called spherical aberration, is illustrated in Fig. 5; it is that characteristic of a spherical lens surface which brings rays passing through different annular zones of the surface to focus at different points along the lens axis. The aberrations of a lens are usually grouped into seven categories; five pertain to monochromatic imagery and two must be added when polychromatic radiation is passed through the lens. They are: spherical aberration; coma; astigmatism; curvature of field; distortion; longitudinal color; and lateral color.

There are many approaches to the lens design problem. One of the most powerful is that in which the design is obtained through a series of approximations to the trigonometric-ray-tracing equations. At an interface between any two refracting media the following equation, Snell's Law, holds:

$$n \sin i = n' \sin r$$

where n and n' are constants of the two media, called indexes of refraction, and i and r are, respectively, the angle of incidence and the angle of refraction at the interface as shown in Fig. 6. If we use only the first terms in the series expansion for the sine function, we have for Snell's Law:

$$ni = n'r$$

an approximation which is valid only for very small values of i and r. Using this so-called "first-order" theory, one can determine the power (inverse focal length) and location for each of the several elements required to satisfy the conditions of object and image position and magnification, assuming no aberrations. When this first-order solution is obtained, the

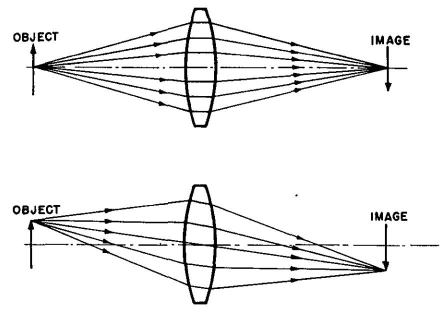


Fig. 4. Ray traces illustrating perfect imagery

next step is to calculate the "third-order" aberrations of the system, aberrations calculated under the approximation:

$$\sin X = X - \frac{X^3}{3!}$$

The usefulness of this approximation lies in the fact that the third-order aberrations (or primary aberrations) can be expressed as explicit quadratic functions of the lens constants; hence, one can solve for these constants as a function of the aberration limitations imposed on the image. It would be ideal, of course, if one could write an analogous set of equations for the exact aberrations, but this is not practical because the lens constants would not appear explicitly, and because the equations would be of very high order and would not be closed. Since the third-order solution uses the first two terms of the sine expansion it yields almost perfect results for a lens of moderate field and aperture and serves as a very useful guide in other cases. Part of the utility of this approach is in its determination of the contributions of each surface of a multi-element lens to each of the lens aberrations, thereby pin-pointing the areas of weakness in the design. The final stage of the design process is accomplished with exact ray tracing through the third-order solution to find the exact aberrations and to make the necessary changes to reduce the higher order aberration effects. This step will vary from a very simple one in a narrow-angle lens, such as a telescope, to a lengthy and complex one in a wide-angle lens, such as the type used in aerial photography.

In understanding the foregoing approach to lens design it is important to

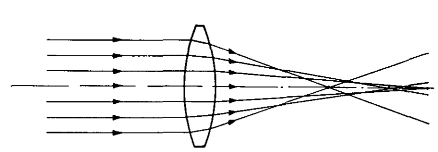


Fig. 5. Ray traces illustrating spherical aberration

note that each step is a necessary, but not a sufficient, condition for the following step. That is, the existence of a first-order solution by no means implies the existence of a third-order solution; and the existence of a third-order solution does not guarantee that there is an exact solution. What is true is the converse; if no third-order solution can be found, then no exact solution exists. This consideration is most important in the first-order design, since only a limited class of the infinite first-order solutions possible will yield a third-order solution. Thus, the designer must keep in mind the primary aberration requirements when laying out his first-order calculation if he is to avoid a mass of extraneous solutions; and he must, in a similar manner, be mindful of the effects of higher order aberrations in his third-order work. It is in this respect that previous experience and knowledge of the field play an important part in determining the efficiency of a designer.

Here we come to the major problem of lens design, that of relating the actual image quality obtained with a lens to the calculated aberrations. While the practice of defining five monochromatic and two chromatic aberrations is mathematic-

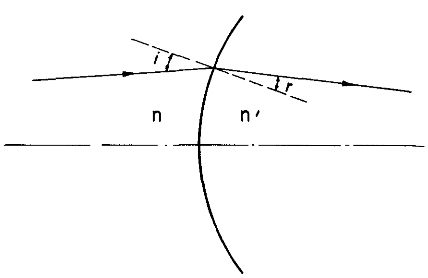


Fig. 6. Snell's law of refraction at a surface

ally appealing as well as convenient to the designer, the final criterion of excellence in a lens is not the amount of spherical aberration, or coma, or astigmatism, but the quality of image resultant from the combination and interaction of these aberrations. The problem is compounded by the experimental fact that a lens which has satisfactory image quality for one class of objects may produce totally unsatisfactory quality for another class. One can show the superiority of lens A to lens B for a test object of black lines on a white background and then completely reverse the judgment for a test object of white lines on a black background. Of recent years there have been many criteria proposed for image evaluation, all having their merits and demerits, none completely accepted by even a majority of

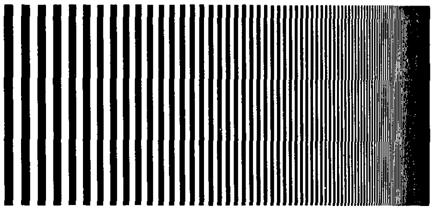


Fig. 7. Test target used in RCR apparatus

optical workers. It is not within the scope of this paper to discuss the many criteria proposed, but some manner of explanation, if not justification, is called for in introducing the criterion of image quality to be used in our proposed system for automatic design, the contrast rendition criterion.

In the recognition contrast rendition apparatus (RCR)[1] a test object is used which is a series of slits separated by opaque regions, the width of successive slits and opaque bands decreasing according to a power series (Fig. 7). This test object moves transverse to the optical axis of a telescope system that images the axial portion of the object onto a slit behind which a photoelectric cell is located (Fig. 8). The telescope system is an aberration synthesizer and can be made to introduce calibrated amounts of each of the primary lens aberrations (Fig. 9). If the output of the photoelectric cell is fed to a strip recorder, then a curve similar to that shown in Fig. 10 is obtained as the test object is moved across the optical axis. As the spacing between slits becomes smaller and smaller, the aberrations introduced by the aberration synthesizer tend to blur together the bright and dark bands in the image until a point is reached at which the bands are no longer resolved and the recorded oscillations cease. The quantity RCR is defined as

$$RCR = \frac{\bar{B}_{\max} - B_{\min}}{\bar{B}_{\max}} \times 100 \text{ per cent}$$

where $\bar{B}_{\max}$ is the average brightness of two consecutive slit images and $B_{\min}$ is the brightness of the image of the dark band between them. Thus, there is an RCR value corresponding to each pair of consecutive slits in the test object. When the peak to trough amplitude of the recorded oscillations becomes zero, $\bar{B}_{\max}$, equals $B_{\min}$ and the RCR is zero. The image quality factor F is measured as the integrated area within the envelope of the recorded oscillatory curve.

It is possible, then, with this apparatus to synthesize any combination of the primary aberrations one wishes to study in a lens and to obtain a single quantity F which is representative of the image quality to be expected from a lens system having these aberrations. If this apparatus is combined with high-speed computing equipment programmed to compute the lens aberrations for a lens whose curves, thicknesses, and indexes are fed into it, then we have the basis for an automatic calculation which proceeds as follows: the designer selects the form of the optical system and specifies nominal values for the curves, thicknesses, and indexes. With these data, the equipment performs an iterative process of successively making a small change in one of the lens parameters, computing the aberrations, setting these aberrations into the synthesizer, and measuring the quality factor F. If for the nth change $F_n > F_{n-1}$ then one makes the next step a larger change in the same direction. If $F_n < F_{n-1}$, then the $(n+1)^{\text{th}}$ must be in the opposite direction. This iteration is continued for variations in each parameter until the image quality is maximized. Fig. 11 shows schematically the equipment used in this scheme. The electronic calculator is an International Business Machines Corporation (IBM) type *607* and the card punch and tabulator are also standard IBM units.

Our present plan is to work with third-order aberrations, because they are easier to calculate than the exact aberrations, and because we know how to synthesize them in the relatively simple arrangement shown in Fig. 9. Assuming that it is possible to design a synthesizer to include the effects of the higher order aberrations, this same approach could be used to reduce the exact total aberrations of a lens. It should be noted that in order to start the iterative scheme described one must have already arrived at the general form of the lens to be used. That is, the scheme is useful only for relatively small changes in a form. While this is far from the goal of completely automatic design it will provide us with a means for eliminating a large portion of the lens designer's work, the tedious and time-consuming task of reducing the aberrations of a lens to the point of acceptability.

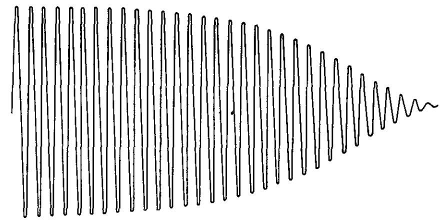


Fig. 10. Typical curve recorded by RCR apparatus

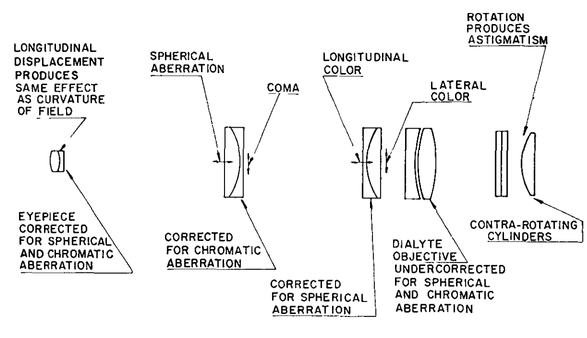


Fig. 9. Schematic of aberration synthesizer

It would appear that the logical extension of this work would be to perform the entire design process automatically. While a scientific worker is seldom on safe ground when he dubs a task impossible, one can certainly say that the prospects for achieving this goal are quite discouraging. One way we might attack the problem is to define a criterion of image quality at each point of the field to be covered by a lens, $F(x,y)$, or $F(\theta, \phi)$ if we chose to work with angular field. The nature of this $F(x,y)$ or $F(\theta, \phi)$ will vary from application to application. For example, the limitation on distortion is much more severe in an aerial mapping lens than in a motion picture camera lens; and spherical aberration must be extremely well corrected in a microscope objective, while a certain amount of this aberration is sometimes desirable in a por-

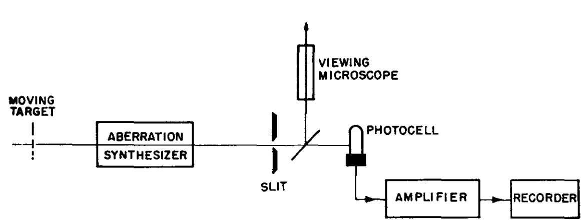
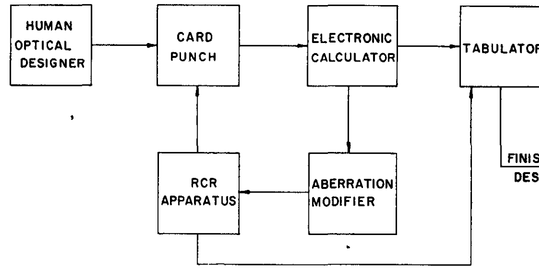


Fig. 8. Schematic of RCR apparatus with aberration synthesizer

trait or landscape lens. Having defined the F function, the problem would be reduced to finding a lens having image quality at least as good as F at each point in the field. Even to begin to handle this problem, we must relate the quantity F to the lens constants of thicknesses, surface curves, and indexes; and this must be done by establishing some system such as the RCR apparatus with its aberration synthesizer working in reverse. That is, we must find a mechanism for going from each F to the aberrations which produce it, and thence to the lens constants. We know of no way to do this, yet; and even if we did, we know of no easy way to get from the exact aberrations to the lens constants, since this would involve solving an open system of higher order equations. What would have to be done instead is to calculate the exact aberrations for all points of the field to be covered and for all possible lenses. Having done this we would have a multiple infinity of lenses from which we would have to "sort out" those which satisfied the restrictions on $F(x,y)$ or $F(\theta, \phi)$. This is not an encouraging approach even for high-speed digital computors, small or large.

The approach that makes more sense to many of us is that of breaking the problem down into two phases: choice of

Fig. 11. Block diagram of RCR combined with computing apparatus



a general lens form and optimization of that form. The first phase involves the decision to use a doublet or a triplet; a Petzval lens or a Tessar. It is now made on the basis of past experience and general knowledge of the designer; but we can well imagine an experience box of some kind into which we fed such information as focal length, total angular field, $f/$ number, and average resolving power, and out of which came recommendations for lens forms worth investigating in detail. The second phase is the one which we would like to attack with our *607* calculator, aberration synthesizer, and RCR apparatus; even the modest attempt we are making to solve the problem for third-order aberrations is fraught with the difficulties of having to make many arbitrary assumptions which can only be partially justified.

Probably the best way to sum up all that has been discussed is to admit that the physics and mathematics of lens design have not been developed to the point where we even could say what we would like a calculating machine to do in order to make completely automatic optical design possible. Until we can solve such basic problems as that of image evaluation we can only continue as we are now doing, making assumptions and attacking pieces of the job with the hope that we may discover how to reformulate the problem so that it may be solved in its entirety.

Reference

1. A METHOD FOR MEASURING THE CONTRAST RENDITION OF OPTICAL SYSTEMS FOR TARGETS HAVING VARIOUS ANGLES OF SUBTENSE, H. S. Coleman, L. V. Foster, D. L. Fridge. *Journal*, Optical Society of America, New York, N. Y., vol 42, 1952, p. 874A.

Discussion

Gordon L. Walker (American Optical): What do you mean by "exact aberrations"?

N. A. Finkelstein: What we mean there is the aberrations calculated by exact ray tracing through the lens system; in other words, by tracing rays through each surface using the full series in the sine expression, the exact ray trace. From that we calculate the exact aberration.

Mr. Casey (General Electric Company): Have you run into any problems of divergence in the iterative machine procedure you described?

N. A. Finkelstein: I would like to point out that we have not yet completed this iterative scheme. What we have now is the RCR apparatus and a method of calculating the third-order aberrations on the IBM *607*.

I think I can answer your question in the light of the work that has been done by others on the CPC. I believe Dr. Hopkins, of the University of Rochester, has used the CPC at Cornell University's calculating Laboratory actually to iterate third-order aberrations. I think he has found that in most cases the iteration does converge and in the cases that it diverges he usually knows what is wrong.

One of the interesting characteristics of lens design I will point out here is that a good lens designer usually knows what is going to happen before it happens. One of the rules to which we like to adhere—and Dr. Hopkins I know feels the same way—is that you should never trace a ray through a lens without knowing where it is going to land before you traced it. You should always have a pretty good idea of what is happening in the lens system because, since they are nonlinear systems, they tend to get out of hand very quickly and unless you keep very close to the right solution, you lose it entirely. Ordinarily, Dr. Hopkins has found that when he does use an iteration approach to solving third-order aberrations, when it does diverge he knows why and it is very easy to remedy. We have not had any personal experience with that phase of the problem.

Franz Edelman (Radio Corporation of America): Could you amplify again briefly your remarks concerning the flow of information through the system? Has anything been done to apply your ideas to problems of continuously varying refractive index (electron optical systems)?

N. A. Finkelstein: I would like to answer the second question first. That is an easy one. Because we haven't been able to solve our own problems we are not going to worry about the electron optical man's problem.

Interestingly enough, we have thought seriously of using some of the techniques the electron optical worker can use because his medium is continuous. We have thought of trying to adapt some of those to our problem, which is something that very often happens. Just as a side remark, I know we had this same experience in filter design. We tried to adapt electric filter design–transmission-line design theory to the design of interference filters for optical work and we discovered that some people in the electric filter design group of one large company were trying to do the same thing in reverse. We have not touched continuous media.

As for the flow of information, what we do is to have the lens designer choose a rough form. In other words, he looks at the requirements of the lens and decides whether it can be solved by a doubler or a tripler, or some other general form. As I said, we could conceive of this being done by some type of experience box, a box into which we feed quantities like f-number and field and which come out with a general class of forms we could use.

Once having chosen this form, the designer makes a first-order solution (in other words, a solution which has the right powers in it), puts it into the *607*, and has the *607* calculate the third-order aberrations. These third-order aberration answers then are fed over to the aberration synthesizer, which was that optical system on which we could put in calibrated amounts of each of the aberrations. He then uses this optical system to image the target I have shown and to calculate a quality factor which is equal to the integrated area under the envelope of the oscillatory curve I have shown. He then goes back to the *607* with this information, goes back automatically, and makes a small change in one of the lens constants and calculates a new quality factor and he uses this iterative scheme to make changes in each constant until he has achieved the optimum quality factor for the over-all scheme.

W. D. White (Airborne Instrument Laboratory): Why not compute the aberration synthesis?

N. A. Finkelstein: I assume you mean by this, why we bother doing this synthesizing optically.

The reason we do is because we are afraid of getting too far away from the problem, because we understand so little about how the quality of the image is related to the aberrations.

As I have said, this idea, this approach of breaking down the defects in an image into spherical and coma and all these other terms, is quite arbitrary, and because of this we are not confident that we know in any sense of the word the relation between these aberrations and the image quality. That is the thing we would like to know, but because we don't we like to be able to synthesize this optically; in other words, not get too far from the problem. We even like to have a microscope there, as you can see, so we can look in and see what the image looks like. We know from past experience what an image that has a lot of coma in it, for example, looks like, and if the *607* calculates a lot of coma and we see a comatic flare we feel a little bit happy. The main reason we don't compute this aberration synthesis is because we really don't know how and we would have to use another rather arbitrary step there, and there are enough arbitrary steps, as you can probably judge, so that we don't want to add any more.

William Kegelman (Philadelphia Electric Company): Are actual sine values loaded into the computer for exact design?

N. A. Finkelstein: In exact ray tracing we try to use an algebraic expression of the trigonometric function. We try to use a formulation which is similar to that of Smith's, if you are familiar with the papers he wrote in the *Proceedings* of the Royal Society in the last 10 years, in which he formulated ray-tracing equations without using sine or cosine function. We try to do that only because the *607* is not a convenient machine with which to use trigonometric functions.

W. A. Malthaner (Bell Telephone Laboratories): Have you considered the analogue model approach to these design problems?

Donn Combelic (Computer Control Company): In your proposed scheme does the designer enter the feedback loop involved in estimating what parameters are to be varied and how much?

N. A. Finkelstein: Have we considered the analogue model approach to these design problems? We have and the difficulty with analogue models is that in lens design calculations we are usually dealing with very small differences between very large numbers and because of that we usually have to work with at least 5-figure accuracy and, preferably, in many wide-angle lenses, with about 7-figure accuracy. We haven't felt that an analogue approach would be of enough accuracy so that we could obtain good results from it.

In answer to the second question, the designer does not enter into the feedback loop in terms of estimating what parameters are to be varied and how much. All he does is to put the general form of the lens in and then go out to lunch and come back and hope it has converged.

Franz Edelman: Judging from the volume of your computations could the work be done economically on a CPC?

N. A. Finkelstein: The answer to that is "yes," but we have found in our work that the *607* is a much more economical instrument than the CPC. In other words, we found that the cost per ray surface for ray tracing, for example, in the *607* under the schemes we have worked out is less than for a CPC. This is probably because we have programmed boards, wired-up boards, and as we do such a large amount of ray tracing this is one of the major jobs of our calculating equipment.

The ElectroData Computer in a Data-Reduction System

K. L. AUSTIN

HIGHER speeds, temperatures, and pressures, greater power, less allowable weight, and other requirements make it increasingly necessary for engineers to employ more accurate design techniques. Data-processing instrumentation has an important role to play in the development and improvement of these techniques.

For example, a government vendor who fails to meet the deadline for submitting performance figures on a weapon he has contracted to build forfeits money and reputation. A manufacturer whose engineers are not provided with enough interpretable performance data during design stages so that required changes can be made soon enough, finds himself falling behind in the competitive race for sales. The validation of theory, the discovery of new techniques, and the proof of satisfactory design in a sufficiently short time are possible today only through the interpretation of data gathered and made meaningful by a data-processing system. Stated simply, the problem is to gather data describing both static and dynamic performance of a product or process and reduce them to comprehensible, meaningful form in whatever interval of time is required. Satisfactory solutions for this data-processing problem have become available only recently because:

1. The need has become critical only within the last few years.
2. The technology necessary for a solution has been, relatively speaking, in its infancy.
3. The attack, has been directed at parts of the problem rather than at system development.

The third reason, perhaps the most important, is now amenable to remedy. The need for data-processing systems will increase and the necessary technological advances will be made to surmount the technical difficulties as they arise.

Plotters, computers, readers, counters, transducers, etc., have been designed to solve parts of the problem but few attempts have been made to link all of the elements necessary for data reduction into one integrated system. For such an attempt the co-operation of several manufacturers is likely to be necessary, and the manufacturers must agree to modify their components wherever modification is necessary to avoid incompatibilities when the system is assembled. The data-reduction system described in this paper is made up of products manufactured or supplied by two corporations modified for integration into the system.

A typical system for data processing contains instrumentation for performing the following functions:

1. Measuring physical phenomena such as

K. L. Austin is with the ElectroData Corporation, Pasadena, Calif.

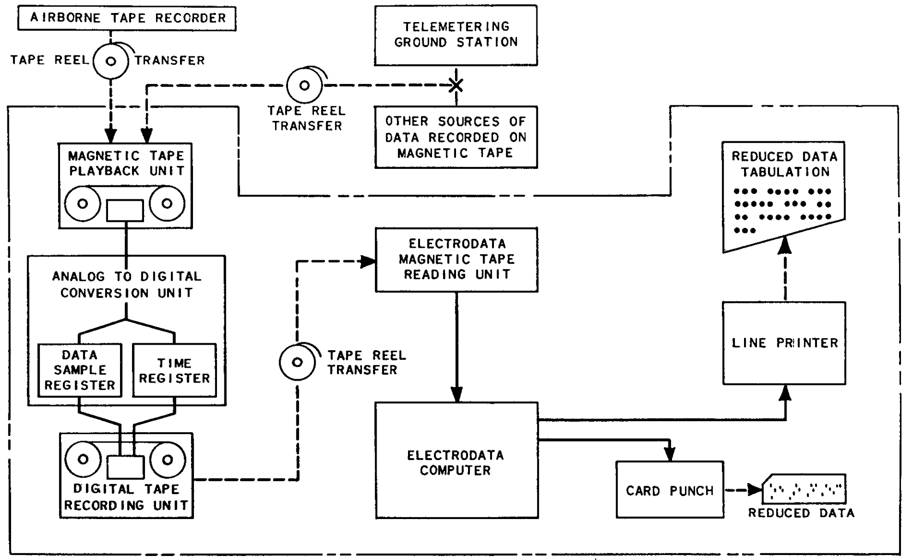


Fig. 1. Basic data-processing system

temperature, pressure, linear and angular displacement, and acceleration. The output of the transducers performing this function is analogue.

2. Standardizing the analogue data from various sources into a form suitable for analogue-digital conversion.

3. Converting analogue data to digital form

4. Interpreting the digitized data and performing the necessary computations on them. The best equipment now available for this is the electronic computer with an internally stored program.

5. Presenting the reduced data for final interpretation by engineers. This presentation may consist of tabulated decimal numbers or a plot of the results.

This paper describes a simplified basic data-processing system capable of performing the five functions listed in the foregoing.

General Description

The principal objectives realized in the design of this system were:

1. To provide adequate identification of data.

2. To reduce over-all data-reduction time as far as possible.

3. To minimize human intervention.

4. To make use and operation as simple as possible.

5. To provide a simple, fast, straightforward system of editing information.

Magnetic tape was chosen as intermediate storage and as the medium of communication between the units of the system because of its compactness and its ability to record or play back information at a wide range of speeds. Input to the system is from magnetic tape. The re-

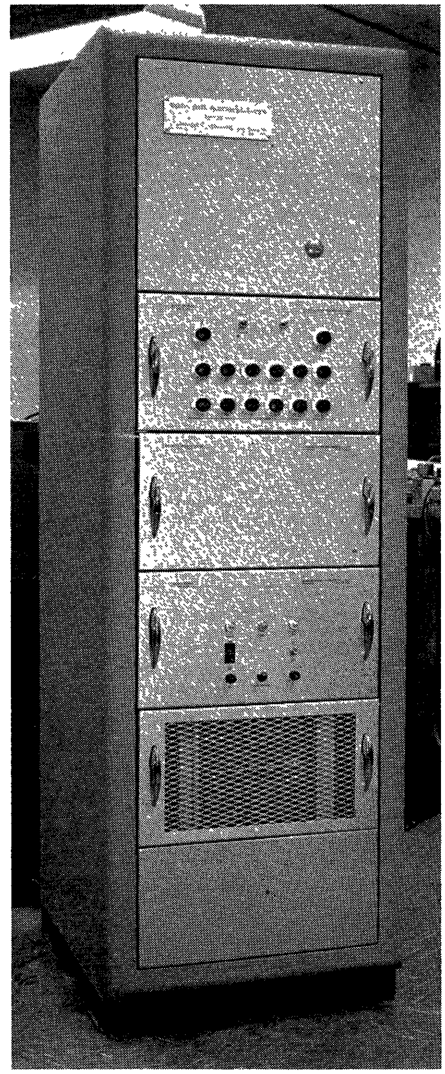


Fig. 3. Tape-to-card conversion unit (storage register)

corded signals may occur in several types of modulation, including pulse trains, pulse width, and pulse frequency modulation. For greater accuracy, compound modulation recording systems can also be used. These methods have been described in various technical publications and will not be covered here.[1,2] The type of recording system most suitable to provide input to the data-processing system is determined by the particular application.

Once the analogue signal is recorded on the tape, the processing is almost automatic. The magnetic-tape playback unit reads the information, including a time channel recorded on the analogue tape, to the analogue-digital conversion unit (Fig. 2), which sequentially records data points and time points in digital form on a digital tape.

The information from the digital tape, through a magnetic-tape reading unit goes into the drum memory of the digital

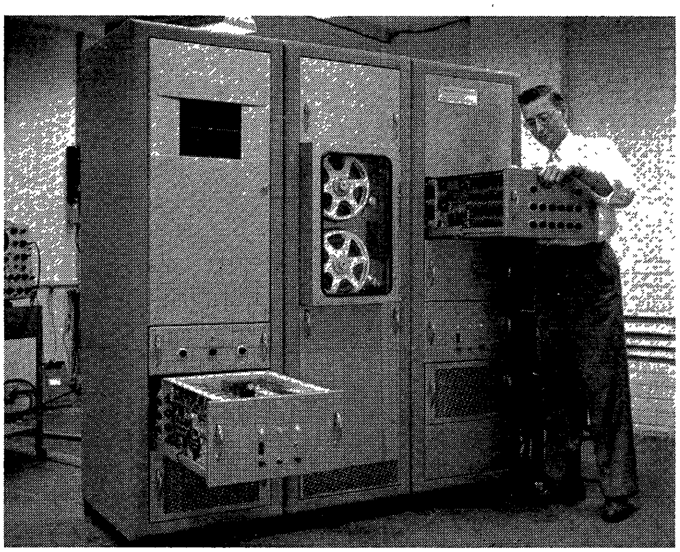
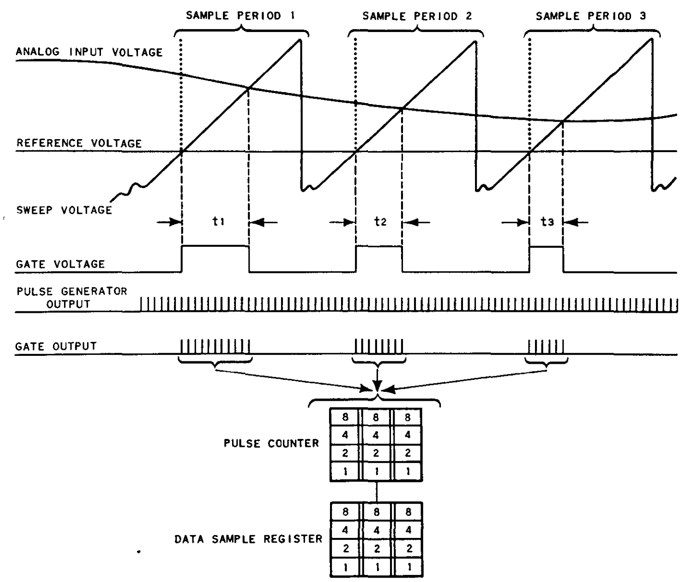


Fig. 2. MilliSADIC analogue - digital converter with tape-to-card conversion unit attached

Fig. 4. Digital conversion technique



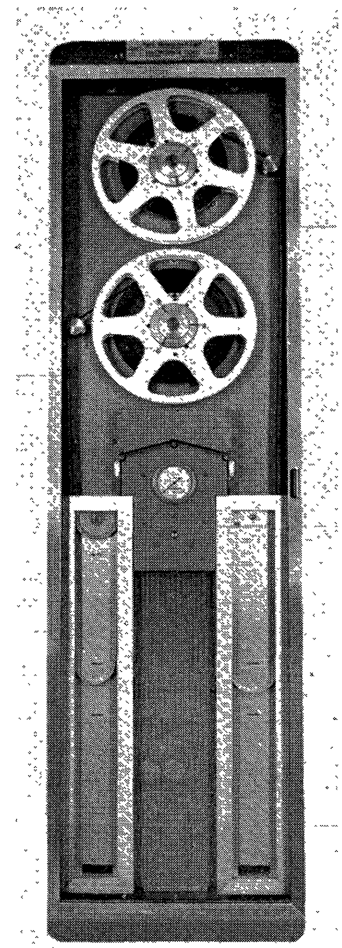


Fig. 5. ElectroData magnetic-tape reading and recording unit

computer, which has been programmed to perform manipulation, adjustment, and computation on the data to yield the desired results.

The reduced data may be printed in final tabulated form on an International Business Machines Corporation (IBM) line printer or may be punched in cards for storage. Other output devices, not shown in Fig. 1, may also be used. If output volume is low, a Commercial Controls Flexowriter attached to the computer can be used. A Commercial Controls paper tape punch can be used for output data to be plotted or stored. If a higher output speed is required than that afforded by the line printer or card punch, a magnetic-tape unit may be used. In this case a subsequent operation using a tape-to-card converter (Fig. 3) and a printer is necessary to put the reduced data into interpretable form.

Magnetic-Tape Playback Unit

The playback unit reproduces analogue signals recorded on multichannel magnetic tape, at a speed servo-synchronized with the original recording speed by means of a clock channel written by the recording system that supplies the original data.

The output of the playback unit to the analogue-digital converter may be a single channel of analogue data, or several channels sampled by a commutator, which supplies data from several sources at approximately the same time interval. Commutation reduces collating and sorting requirements in applications where digital data from several channels are to be involved in computation. Whether one or several channels are read, information from a time channel is part of the output of the magnetic-tape playback unit.

Analogue-Digital Conversion Unit

Three types of analogue information can be accepted by the analogue-digital conversion unit: voltage, pulse width, or pulse train. All other forms must be converted to one of these before they can be digitized by this equipment. Equipment necessary for such conversion, determined by the form of recording used, is supplied as part of the analogue digital converter.

Fig. 4 illustrates the method used to convert analogue input voltage to discrete numerical measurements. Pulses are gated into a counter on coincidence between a sawtooth sweep voltage and a zero reference voltage. The pulse train is turned off when the input voltage and the sweep voltage coincide. The gated pulses drive a 3-decade binary-coded decimal counter. The resulting count is the decimal evaluation of the analogue signal and is referred to as the data sample. At the end of each sample period the counter is reset to zero.

At a rate determined by a manual samples-per-second selector switch, data samples are transferred to a 3-decade data-sample register for subsequent readout to magnetic tape. The setting of the selector switch at particular values between 20 and 1,200 samples per second, is determined by the nature of the analogue data to be converted.

The output of the time channel from the playback unit generates time points in a 6-decade binary-coded decimal counter or in a 6-decade register. The means used to generate time points depends on which of two available methods of recording is to be used. These are

1. Recording an equally spaced series of pulses to represent a specified time increment, say 1/100 of a second. The pulses drive an electronic counter that is cumulative from some zero start-time. This method has the advantage that time can be recorded in fine increments and the disadvantage that any recording or counting error will be propagated through subsequent operations.
2. Recording a pattern of pulses in the time channel to represent an absolute time indication. A time decoding unit transforms the pulse pattern into a binary-coded decimal 6-digit time point. On receipt of each pattern the previous time point is shifted out and the new point is transferred into the time register for subsequent recording on the digital tape. This method has the advantage that recording error in one pattern or transient malfunction of the decoding unit will affect only one pattern, and the disadvantage that time identifications are coarser because of the space required for recording the time pattern.

The output of the analogue-digital conversion unit is data samples and time points. The order in which the binary-coded decimal digits of these two values

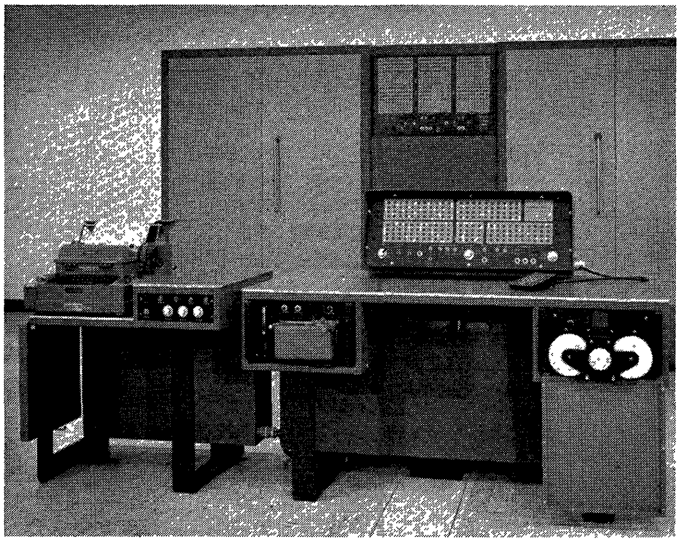


Fig. 6. ElectroData digital computer

will be read out is established by the analogue-digital conversion unit, which controls timing and sequence of the entire operation.

Digital Magnetic-Tape Unit

For an economical recording density, the digital tape unit uses a variable-speed tape drive, the speed of which is determined by the number of samples per second being recorded.

The four bits of a decimal digit are recorded in parallel on four channels, decimal digits occurring serially. This mode of operation is the one used throughout the computer. Two additional tape channels are used to record information needed by the computer system.

Data samples are transferred from the data-sample register at a constant rate. The time available between the transfer of successive data samples is not enough for reading out a 6-digit time point; therefore, time-point digits are interspersed with data samples. This circumstance, together with the computer's requirement of 11-digit words (ten decimal digits and sign) and a digit space between words, produces the following word structure on the tape:

s algebraic sign
H hundreds digit ⎫
T tens digit ⎬ data sample
U units digit ⎭
t time-point digit
O blank
O blank
H hundreds digit ⎫
T tens digit ⎬ data sample
U units digit ⎭
t time-point digit
S space between words

ElectroData Magnetic-Tape Reading and Recording Unit (Fig. 5)

Under control of the computer the tape unit transfers the digitized raw data to the magnetic-drum memory of the computer. This transfer can be made at the rate of 1,000 data samples and associated time points per second.

ElectroData Computer (Fig. 6)

The computer is a medium-speed internally programmed single-address magnetic-drum computer. It contains:

4,000 words (10 decimal digits and sign) of drum storage, termed "main memory" with an average access time of 8.5 milliseconds. 80 words of drum storage, termed "quick-access memory," (four 20-word loops on the drum) with an average access time of 850 microseconds.

A *B* register, used for automatic tallying and command modification.

Decimal notation and arithmetic throughout (binary-coded 1-2-4-8 system) in series-parallel operation.

The relatively high speed (2 milliseconds average for the execution of all commands except multiplication, division, and reference to main memory) is realized by transferring data and commands to the quick-access loops for operation and execution.

The commands which control the operation of the tape reading and recording unit perform the following functions:

1. Read from 1 to 100 blocks into the drum memory from the tape.

2. Write from 1 to 100 blocks on tape from the drum memory.

A block of information contains 20 words. Thus, a maximum of 2,000 words or 4,000 data samples with associated time points may be read in by the execution of one command. If desired, 40 or 80 samples with associated time points may be read directly to the quick-access loops.

The computer is coded to separate, rearrange, and interpret the interspersed data samples and time-point digits after read-in to the memory. The data are then reduced by the use of appropriate coded routines to produce the final result.

The time required for computing final results from raw data will, of course, depend on the type of data and the complexity of the formulas involved. The time required for some typical data reduction programs may, however, give the reader an idea of the time necessary.

1. Total time for reducing flight test data is about 200 milliseconds per data point, including the following operations:

(*a*) Interpretation and checking of all data read into the computer.
(*b*) Zero offset correction.
(*c*) Slope correction.
(*d*) Correction for instrument calibration deviation.
(*e*) Conversion of correct instrument reading to the physical quantity being measured.
(*f*) Computation of local maxima.
(*g*) Computation of local minima.
(*h*) Computation of moving averages for smoothing.
(*i*) Computation of mean square deviation from smoothed curves.
(*j*) Allowance for control and manipulation commands.

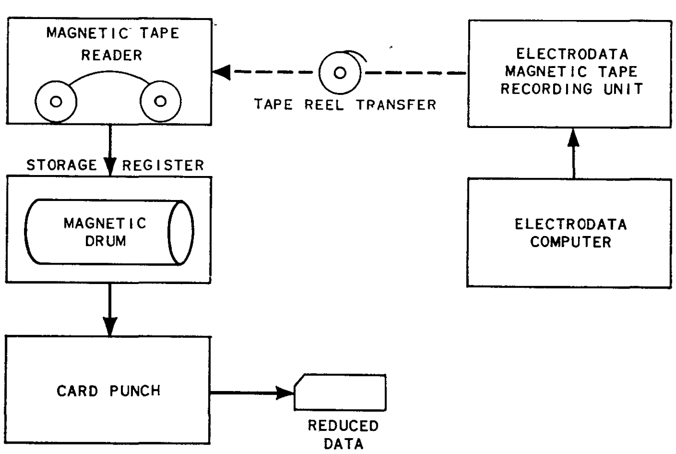


Fig. 7. Tape-to-card conversion

2. Time for reducing 6-component wind tunnel data is about 3.1 seconds per point, on the average.

3. Time for reducing 4-station theodolite data is about 6 seconds per frame.[2]

The Line Printer and Card Punch

The IBM line printer, under control of the computer, can be operated at the rate of 150 lines per minute. From one to eight words per line may be printed. Thus, the maximum output speed of the printer is 1,200 words per minute. It should be noted that output words may contain more than one final result and these results can be separated on the page by proper setup of the printer.

The card punch can be used to record final results at a maximum rate of 800 words per minute.

Alternate Output Units

Alternate output units can be used if the volume of output does not match the capabilities of the line printer and/or card punch.

For smaller volumes a Flexowriter attached to the computer can type a maximum of 60 words per minute. The results may be punched out on a paper tape punch at the rate of 120 words per minute. The resulting paper tape can then be fed into a plotter or Flexowriter for final presentation of the results.

For very large volumes of output the magnetic-tape unit will accept 30,000 words per minute. As has been mentioned, this requires subsequent tape-to-card conversion and printing or plotting.

Fig. 7 is a block diagram of the units of the magnetic-tape output system.

Information is written on the tape, under control of the computer, in a format acceptable by the tape-to-card converter. Information for controlling the converter is written in two of the six channels on the tape by the computer system. One channel is used for marking off units of information to be converted and punched. Each unit consists of 66 digits to be punched and a space recorded after each group of 11 decimal digits. Thus, one channel is used to identify the 66 decimal digits to be punched within the unit.

A magnetic-tape playback unit is used to reproduce the information on the magnetic tape prepared by the computer and transmit the information units on the tape to the tape-to-card converter.

The conversion unit, or "storage register," consists of a magnetic drum (Fig. 8) that acts as buffer storage between the magnetic-tape output and the card punch. It contains circuitry for controlling the operation of the system and for decoding the serial binary-coded decimal output of the tape playback unit into the parallel form required for input to the card punch.

The magnetic drum in the converter is divided into two sections, each capable of storing the 66 decimal digits of one information unit. Units of input information are read alternately into these two sections. During read-in to one section, the information unit stored in the other section is converted, transferred, and punched on one card. The conversion consists of setting up a row of 66 relays once for each of the ten numerical row locations on a card. The setting up is accomplished by searching the unit of information stored on the drum for decimal values that coincide with the numerical code of the row under the punch dies. The detection of coincidence results in the closing of relays whose locations in the row correspond to the locations of the digits in the unit of information. After each row is set up in the relays, the punch dies are actuated by the closed relays. When the conversion of one line is completed, the functions of the two sections of the drum are reversed. Speeds of the punch and the magnetic-tape playback unit are controlled so that both operate continuously.

Selector switches for emitting 14 decimal digits are included as part of the tape-to-card converter. Information that remains constant throughout the conversion of the data on one tape may be stored in these switches and punched on each card. The advantage of using this conversion equipment is that the computer system may be cleared for the reduction of more data by relegating the tabulating function to the conversion equipment. If the volume of output warrants it, more than one tape-to-card converter may be used.

Extension of the Basic Data Processing System

The basic system as described is appropriate for the reduction of static or quasi-static data gathered from such sources as engine test stands and wind tunnels. In tests of this sort, all pertinent conditions are checked and adjusted by the operators of the test before the taking of data. The operator does not press the button to record data until he is satisfied the system is stabilized and the data will be meaningful. Thus, all data recorded may be processed and the computer may economically be given the job of discarding bad data. On the other hand, in the case of dynamic tests, the operators have very little control over the conditions of the test other than the decision to start

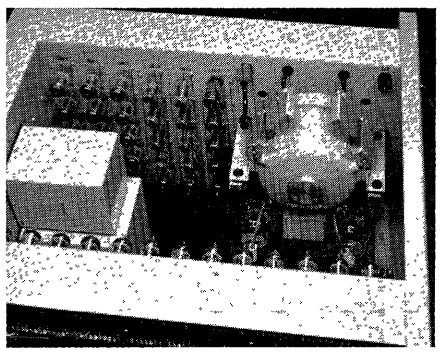


Fig. 8. Magnetic drum—tape-to-card conversion unit

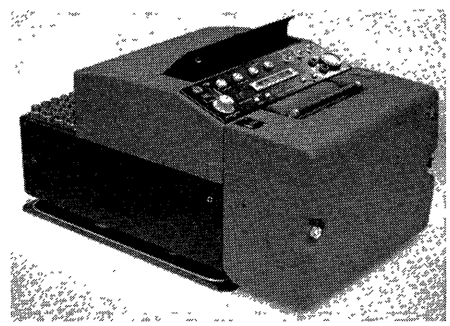


Fig. 9. Multichannel recording oscillograph

TIME TRACK ON OSCILLOGRAPH RECORD						
BINARY VALUE	0000	0111	0010	1001	0000	0001
DECIMAL VALUE	0	7	2	9	0	1

(⋀ = 1 ⋁ = 0)

Fig. 10. Serial binary code—oscillograph record

taking data. Once the recording begins, the operator relinquishes control to the pre-programmed data-recording system. Since the operator cannot be sure when significant system responses will take place, insignificant data must be recorded to be sure of recording areas of significance. Examples of this type of test are the multichannel recordings of many variables during a test flight, using an air-borne tape recording system, and the recording of data on tape at a telemetering ground station during test flight of missiles. In such cases many insignificant data are recorded. Estimates of 10 per cent useful data seem realistic. It would be economically foolhardy to attempt to run the bad data through the entire system. Although the computer can be programmed to edit the data after they are digitized, this is an expensive operation and drastically hinders the producing of useful reduced data. A more economical means of editing can be provided utilizing many components of the basic system.

Editing Procedures

So that the test engineer (editor) may have a presentation of the recorded data in order to decide which areas of the data are significant, the magnetic-tape playback unit reproduces and transfers all channels on an analogue tape to a multichannel recording oscillograph (Fig. 9). The information recorded in the time channel on the analogue tape is converted to digital time points by the analogue-to-digital converter, and time points are transferred from the time register or counter and recorded on one channel of the oscillograph record in the serial binary-coded decimal-digit form illustrated in Fig. 10. The editor uses these time points for controlling the editing operation.

In many cases long lengths of the analogue tape will contain information of no interest in any channel. In the editing process these parts are cut out and a new tape is spliced together.

Time points identifying the points where the analogue tape is to be cut are punched into paper tape. The analogue tape is placed in the playback unit and the information recorded in the time channel is converted to digital time points in the time counter or register. These time points are transferred to an editing control unit where they are compared with time points read from the paper tape. On coincidence between the time point from the time counter or register and the time point read from the paper tape, the playback unit is halted and the operator can mark the tape for splicing.

The amount of insignificant data can be reduced further by punching on paper tape time points that identify the beginning and end of significant data in individual channels (or groups of channels if a commutator is to be used) and using this tape to control, via the editing control unit, the recording of data samples and time points on the digital tape. Recording will take place only between the start and stop time points punched on the tape.

These methods of editing can reduce elapsed time from receipt of analogue tapes to tabulated final results in many cases as much as 80 per cent.

System Checks

The computer serves as the primary checking device for system errors. These checks are mainly programmed, although some built-in checks are provided.

The computer can be programmed to compare time-point discontinuities with a prepared start-stop time-point table. Inconsistencies between time points and data samples can be recognized. The computer can be supplied with tolerances for the data samples representing different measurements and can use these tolerances as a check. Other programmed checks can be devised, and the extent of checking will depend upon the characteristics of the data, the ingenuity of the programmer, and the time that can be economically allotted for checking.

The computer will automatically recognize, through built-in checks on read-in, some types of errors that occur in recording and will recognize and correct data samples that exceed the scale of the digital converter.

Conclusion

Once the computing routines and information-handling procedures have been programmed for the computer, this data-processing system makes the transition from physical phenomenon to interpretable analytical information about it nearly automatic, so far as the engineering user is concerned. He must still make certain determinations about modes of recording his data, must exercise choice among output units, and must know the operation of the system to make the best application of it to his problem, but the detail work involved in other methods of converting one kind of reading to another and correlating different kinds of measurements has largely disappeared.

From the operator's point of view the system is extremely simple, since all units are provided with easily interpretable displays to be monitored and with automatic alarms.

The magnetic-tape medium is central to the speed and versatility of such a building-block system. In military applications the advantage of speed is apparent. Equally important is the value of such systems for all industry in closing the gap between design and adequate evaluation, and hence eliminating much of the lag between design and profitable production.

References

1. Recording of Precision Data on Magnetic Tape, **R. L. Sink.** Paper *54-35-3*, Instrument Society of America, Pittsburgh, Pa.

2. Telemetering Standards for Guided Missiles. Research and Development Board, United States Department of Defense, Washington, D. C.

3. Transforming Theodolite Data, **Henry Schutzberger.** *Proceedings*, Computational Seminar, International Business Machines Corporation, New York, N. Y., 1949.

Discussion

E. L. Harder (Westinghouse Electric Corporation): With what accuracy can analogue data be recorded on magnetic tape and read? What form of modulation is used for the accuracy stated?

K. L. Austin: As for the range of data samples, three decimal digit samples may be recorded; these range from 0 through 999. Less than 1-per-cent error is quoted for this system of conversion.

What form of modulation? The form of modulation actually accepted by the system is pulse train, pulse width, or voltage level modulation. All other types of modulation must be converted into one of these forms.

D. B. Jordan (Sylvania Missile Systems Laboratory): To what extent were data interpreted by that step of the system?

K. L. Austin: Within the computer itself an interpretation process is necessary in tearing apart the data. The data samples and time points must be interspersed on the digital tape. This requires that an interpretation process be used to break the words apart and put them in suitable form.

D. B. Jordan: You mentioned in the steps of the over-all program that there was an interpretation, when you spoke of it originally, following the ATD conversion. Is this merely a sorting of data or ordering of the data as it comes off?

K. L. Austin: That is all I meant to imply, that it is an ordering of the data.

ORGANIZATION OF THE JOINT COMPUTER COMMITTEE

A. Name and Object

1. *Name.* This Committee shall be known as the Joint Computor Committee, herein identified by the abbreviation "JCC".
2. *Sponsorship.* The JCC shall be jointly and equally sponsored by the following three organizations:
Association for Computing Machinery
American Institute of Electrical Engineers
Institute of Radio Engineers, Inc.
3. *Object.* The Committee shall aid in the promotion of close co-operation and co-ordination in the activities of the sponsoring societies related to the field of computer engineering and allied arts and sciences. Its objects shall be scientific, literary, and educational in character and harmony with the aims of its sponsors.

B. Field of Interest

1. *Specific Scope of Sponsors.* The general scope of the JCC shall be that which stems from the scope of the Association for Computing Machinery, the Committee on Computing Devices of the AIEE, and the Professional Group on Electronic Computers of the IRE.

2. *Major Interests.* The major field of interest of the JCC shall be the engineering aspects of the design, development, manufacture, and use of computers, but shall also include an interest in the various activities that contribute to this field or utilize the products or techniques of this field. The term "computers" shall be interpreted broadly to include data- and information-handling and processing systems useful in the solution of scientific computation, business accounting, and industrial and military control problems.

C. Committee Formation and Management

1. *Committee Structure.* The JCC shall consist of fifteen (15) members, with four (4) appointed by each of the three sponsoring societies, together with the following three ex-officio members:
The president of the Association for Computing Machinery
The chairman of the AIEE Committee on Computing Devices
The chairman of the IRE Professional Group for Electronic Computers

The twelve (12) appointed committee members shall be divided: two (2) from each society from the East Coast, and two (2) from each society from the West Coast. The East Coast group shall be known as the Eastern JCC, and the West Coast group as the Western JCC. The three ex-officio members of the JCC shall be members of both groups. Where possible, the East and West groups should hold committee meetings during the respective conferences, and members of both East and West Coast groups will be invited to these meetings and participate with equal voice.
2. *Tenure.* The terms of office of the committee members appointed by the sponsoring society shall be 2 years, one half the members being appointed each year. Terms of office shall coincide with the calendar year, but members shall continue to serve until their successors are appointed. Only one member of each society on each Coast will be replaced each year. It is recommended that members from each Coast be suggested by the respective active organizations on that Coast. If there is no active group on a particular Coast, then the selections must be made by the heads of the three national groups.
3. *Selection of JCC Officers and Steering Committees.* The Eastern JCC and the Western JCC each shall annually elect one of its members as chairman, and another as secretary, as soon as expedient following the appointment of the new committee members. The officers' terms shall normally coincide with the calendar year, but incumbents shall continue to serve until their successors are elected.

When a conference is sponsored by the JCC in the East or in the West, the nucleus of the steering committee for this conference shall be, respectively, the Eastern JCC or the Western JCC. The chairman, East, and the chairman, West, shall serve, respectively, as the chairman of the Eastern and Western Conference Steering Committees. Each chairman under the direction of his own committee shall have the power to appoint additional members to it where required to carry out its work. A conference steering committee shall include a technical program chairman, a local arrangements chairman, and a publications chairman. This steering committee will work with local organizations in the actual initiation and management of the conference.
4. *Local Arrangements Chairman.* The local arrangements chairman shall be responsible for all local arrangements, including: finance, registration, inspection trips, exhibits, hotel arrangements. He shall also be responsible for all conference publicity, both local and national.
5. *Technical Program Chairman.* The technical program chairman shall be responsible for implementing all phases of the technical program. This includes: selection of, and arrangements with, all speakers and session chairmen, procurement of written material as required by the publication committee and local arrangements committee.
6. *Publications Chairman.* The publications chairman shall be responsible for the preparation of the Proceedings of the conference.
7. *Joint Secretary-Treasurer.* Any questions of JCC policy not covered by the charter should be decided by the chairmen of computing groups of the sponsoring societies who are ex-officio members of the JCC. These three men will unanimously appoint a "Joint Secretary-Treasurer," who on matters of this sort will serve as liaison between the sponsoring societies and the East and West Coast groups, and who may be delegated such responsibilities for co-ordination on a national level as the three chairmen of the computing groups of the three sponsoring societies may unanimously decide. Examples of such duties would usually include:

(*a*). Act as representative of the three societies in establishing reasonable and consistent conference policies;

(*b*). Obtain clearance from the three societies for specific conference operational and fiscal plans;

(*c*). Arrange for a final summary operational and fiscal report of each conference to be distributed to the JCC, to the three sponsoring societies, and to other interested parties;

(*d*). Preparation and maintenance of a specialized, unified mailing list on the basis of previous conference registrations.

(*e*). Other responsibilities as specified in Section D-2.

D. Committee Activities

1. *Conferences.* The JCC shall sponsor annually one or more conferences. Such conferences are not to be considered as being in competition with, or a replacement of, the regular

activities of their sponsoring societies. Instead, the conferences are to represent the combined efforts of the sponsors to produce major technical meetings of a specialized nature. Each conference is to treat a selected theme in the computer field in a thorough and authoritative manner; inspection trips and exhibits are to be featured; formal and informal discussions are to be encouraged; and a conference report is to be published. The quality and coherence of the technical programs shall be such that the publications resulting from the conferences will be useful as current and authoritative text or reference books covering various phases of the development of the computer field.

A JCC conference sponsored in the East shall be the specific responsibility of the Eastern JCC, and a JCC conference sponsored in the West shall be the specific responsibility of the Western JCC. The following are a few examples of the scope of operation of either JCC group in planning a conference in its own area:

(*a*). Decision on location of conference.

(*b*). Decision on topic or theme of conference.

(*c*). Decision on conference duration.

(*d*). Selection of chairman and advice concerning organization of local committees necessary to carry on the conference.

(*e*). Decisions on whether or not to include exhibits, and, if so, what kind.

(*f*). Decisions concerning publication of Proceedings.

(*g*). Fix a scale of rates for registration, cost of Proceedings, etc.

Not less than one conference nationally, and not more than one conference on each Coast, shall be sponsored annually by the JCC. The final authority for sponsorship of any conference shall reside with the JCC as a whole, and through it with the sponsoring societies.

2. *Other Activities.* The JCC as a whole, under the chairmanship of the Joint Secretary-Treasurer, may establish other committees as required to carry out its work. Such committees may, for example, be established to study and recommend long-range co-ordinated plans for conferences, exhibits, and publications in the computer field, and to accumulate and preserve by the preparation of reports and manuals the experience gained through its various specialized operations. The sponsoring societies may also, if it is mutually agreeable, request the JCC to undertake other projects in the computer field which may be particularly expedited through the joint and co-ordinated action of the committee.

E. Financial Considerations

1. *Conference Financial Matters.* The previous Joint Computer Conferences and their publications have been so successful from a monetary point of view that financial support of future conferences appears definite. However, in order to insure that the basic responsibility for the management of future conferences always springs from, and resides in, the three sponsoring societies, the Joint Computer Committee shall treat each future conference as an independent project requiring separate approval and initial loans from, and financial accountability to, the sponsoring societies. The Finance Chairman of the Eastern JCC and of the Western JCC will each be responsible to these societies through the Joint Secretary-Treasurer for the financial affairs of his own group.

2. *General Financial Procedure.* Approximately one year in advance of a conference on either Coast, the JCC group on that Coast shall form, with itself as a nucleus, a Conference Steering Committee. This committee shall formulate initial plans for the time, place, and subject, and submit a tentative plan of action, together with a conference budget, for approval to the sponsoring societies through the Joint Secretary-Treasurer. If the plans are approved by the sponsoring societies, the Joint Secretary-Treasurer shall request a loan of not greater than $300.00 from each of the sponsors, and forward the money to the Conference Steering Committee.

3. *Disposition of Surplus Funds.* All surplus funds resulting from the operation of a conference shall be distributed equally to the three sponsors by the Joint Secretary-Treasurer within six months after date of conference.

4. *Committee Expenses.* The JCC shall formulate a budget of expenses which it expects to incur in connection with its activities, but not chargeable directly to a particular conference; as for instance, committee stationery, preparation of a unified mailing list, and the like. This budget will be submitted to each of the sponsoring societies accompanied by a request for an appropriation from these sponsoring societies covering these expenses. These funds will be turned over to the Joint Secretary-Treasurer, who will account for them to the sponsoring society.