T-107

Proceedings of the Western Joint Computer Conference

Contrasts in Computers

Presented at Los Angeles, Calif.
May 6–8, 1958

Sponsored by:
AMERICAN INSTITUTE OF ELECTRICAL ENGINEERS
THE ASSOCIATION FOR COMPUTING MACHINERY
THE INSTITUTE OF RADIO ENGINEERS

March 1959

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

Printed in the United States of America

Price $4.00

T-107

Proceedings of the Western Joint Computer Conference

Contrasts in Computers

Presented at Los Angeles, Calif.
May 6–8, 1958

Sponsored by:
AMERICAN INSTITUTE OF ELECTRICAL ENGINEERS
THE ASSOCIATION FOR COMPUTING MACHINERY
THE INSTITUTE OF RADIO ENGINEERS

March 1959

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

Printed in the United States of America

Price $4.00

Welcome Address

On behalf of the 1958 Western Joint Computer Conference, I welcome you, and I hope your stay will be pleasant and profitable.

As most of you know, the conference is one of two Joint Computer Conferences held each year, sponsored jointly by AIEE, the Institute of Radio Engineers, and the Association for Computing Machinery. I think that you know that the Eastern Conference rotates among four cities, Washington, New York, Philadelphia, and Boston; and that this Western Conference will, at least by 1961, be alternating between San Francisco and Los Angeles.

I would like to call attention to a number of features of our conference; following the opening session, the technical program presents six invited panel discussions and six discussions of contributed papers. The topics of these several panels investigate a number of controversial subjects pertinent to our industry. The contributed sessions present papers in several areas of application and design for circuit systems; all significant discussions are incorporated in the proceedings of the conference.

Willis H. Ware
Conference Chairman

National Joint Computer Conference Officers

Chairman

M. M. ASTRAHAN.....International Business Machines Corporation

Vice Chairman

H. H. GOODE.....University of Michigan

AIEE Representatives

J. G. BRAINERD.....Moore School, University of Pennsylvania
J. F. KALBACH.....ElectroData Division of Burroughs Corporation
R. A. IMM.....International Business Machines Corporation
G. GLINSKI.....ElectroData Division of Burroughs Corporation

IRE Representatives

L. C. NOFREY.....Marchant Research, Inc.
N. H. TAYLOR.....
.....Lincoln Laboratories, Massachusetts Institute of Technology
R. D. ELBOURNE.....National Bureau of Standards
H. H. GOODE.....University of Michigan

ACM Representatives

P. ARMER.....The Rand Corporation
S. FERNBACH.....University of California, Livermore
A. PERLIS.....Carnegie Institute of Technology
F. M. VERZUH.....Massachusetts Institute of Technology

WESTERN JOINT COMPUTER CONFERENCE

Chairman

W. H. WARE ..... The Rand Corporation

Vice Chairman

H. T. LARSON ..... Aeronutronic Systems, Inc.

Secretary

R. E. UTMAN ..... ElectroData Division of Burroughs Corporation

Conference Committees

Finance

K. W. UNCAPHER ..... The Rand Corporation

Publications

A. DOWLING ..... Ramo-Wooldridge Corporation

Technical Program

M. PHISTER, Jr. ..... The Rand Corporation
W. F. GUNNING ..... Systems Division of Beckman Institute
C. T. LEONDES ..... University of California, City of Los Angeles

Printing

W. A. FARRAND .... Autonetics Division of North American Aircraft, Inc.

Registration

M. HOWARD ..... Ramo-Wooldridge Corporation

Hotel Arrangements

J. J. SEIDMAN ..... National Cash Register Company

Exhibits

D. F. WEINBERG ..... Ramo-Wooldridge Corporation

Public Relations

A. R. McKAY ..... Lockheed Missiles Systems

Trips

P. M. BROWN ..... Aeronutronic Systems, Inc.

Women's Activities

NAN GLENNON ..... Ramo-Wooldridge Corporation
BILLIE KEESEY ..... Telemeter Magnetics, Inc.

Publicity Consultant

DAVID PARRY ..... David Parry & Associates

Exhibitors

Adcon Corporation ..... Boston, Mass.
Aeronutronic Systems, Inc. ..... Van Nuys, Calif.
Aladdin Electronics Division of Aladdin Industries, Inc. ..... Pasadena, Calif.
Ampex Corporation, Instrumentation Division ..... Redwood City, Calif.
Amp, Inc. ..... Harrisburg, Pa.
Audio Devices, Inc. ..... Los Angeles, Calif.
Autonetics ..... Downey, Calif.
Bendix Computer Division ..... Los Angeles, Calif.
Benson-Lehner Corporation ..... Los Angeles, Calif.
C. P. Clare & Company ..... Los Angeles, Calif.
Coleman Engineering Company ..... Los Angeles, Calif.
Computer Control Company, Inc., Western Division ..... Los Angeles, Calif.
Computer Control Company, Inc., Eastern Division ..... Wellesley, Mass.
Datamatic, Division of Minneapolis-Honeywell Regulator Corporation ..... Newton Highlands, Mass.
Digital Equipment Corporation ..... Maynard, Mass.
Electronic Associates, Inc. ..... Long Branch, N. J.
ElectroData Division of Burroughs Corporation ..... Pasadena, Calif.
Electronic Engineering Company of California ..... Santa Ana, Calif.
Encyclopedia Britannica ..... Los Angeles, Calif.
Engineered Electronics Company, Subdivision of Electronic Engineering Company of California ..... Santa Ana, Calif.
Ferranti Electric, Inc. ..... Hempstead, N. Y.
Friden Calculating Machine Company, Inc. ..... San Leandro, Calif.
General Electric Company, Rectifier Department ..... Schenectady, N. Y.
G. M. Giannini & Company, Inc., Datex Division ..... Monrovia, Calif.
Great Books of the Western World ..... Los Angeles, Calif.
Hughes Aircraft Company ..... Culver City, Calif.
International Business Machines Corporation ..... New York, N. Y.
F. L. Moseley Company ..... Pasadena, Calif.
Motorola, Semiconductor Division ..... Phoenix, Ariz.
G. E. Moxon Sales ..... Los Angeles, Calif.
National Cash Register Company ..... Dayton, Ohio
Potter Instrument Company, Inc. ..... Plainview, N. Y.
Ramo-Wooldridge Corporation ..... Los Angeles, Calif.
J. B. Rea Company, Inc. ..... Santa Monica, Calif.
Remington Rand, Division of Sperry Rand Corporation ..... New York, N. Y.
Rese Engineering, Inc. ..... Philadelphia, Pa.
Research & Engineering, the Magazine of Datamation ..... New York, N. Y.
Royal McBee Corporation ..... Port Chester, N. Y.
Soroban Engineering, Inc. ..... Melbourne, Fla.
Sprague Electric Company ..... North Adams, Mass.
Stromberg-Carlson, San Diego Division ..... San Diego, Calif.
Telemeter Magnetics, Inc. ..... Los Angeles, Calif.
Teletype Corporation ..... Chicago, Ill.
John Wiley and Sons, Inc. ..... New York, N. Y.
Shephard Winters ..... Los Angeles, Calif.
Walkirt ..... Inglewood, Calif.
Wright Engineering Company ..... Pasadena, Calif.
Sponsor's Booth, ACM, AIEE, IRE

Contents

Page

WELCOME ADDRESS, W. H. Ware ... 2

THE SOCIAL PROBLEMS OF AUTOMATION, H. T. Larson, *Chairman* ... 7

The Social Consequences of Automation, Harold D. Laswell ... 7
The Social Problems of Automation, J. B. Schafer ... 10
The Social Problem of Automation, Cuthbert C. Hurd ... 13
Discussion ... 15

LOGICAL CIRCUITRY FOR TRANSISTOR COMPUTERS, J. H. Felker, *Chairman* ... 17

Transistor Resistor Logic Circuits for Digital Data Systems, T. R. Finch ... 17
Direct Coupled Transistor Logic Circuitry, James B. Angell ... 22
Symmetrical Transistor Logic, R. H. Baker ... 27
IBM Current Mode Transistor Logical Circuits, J. L. Walsh ... 34
Discussion ... 37

SPECIAL DEVICES FOR COMPUTERS, R. Brown, *Chairman*

MicroSADIC, A High-Speed Data-Preparation System with Variable Format Output, Helmut Schwab ... 40
A Computer-Integrated Rapid-Access Magnetic Tape System with Fixed Address, R. L. Best, T. C. Stockebrand ... 42
The Dynamics of Toggle Action, Norman L. Kreuder ... 46
A Direct Access Photomemory Part I, Prototype Machine System, F. A. Litz ... 50
Direct Access Photomemory Part II, System Considerations, A. J. Critchlow ... 53
Discussion ... 56

DIGITAL COMPUTER SYSTEMS I, Gerald Estrin, *Chairman*

The Flow Diagram Approach to Computer Logical Design Using the NCR 304 as an Illustration, James Hudson, Walter Edwards, D. E. Eckdahl 59
Cascaded Variable Cycle Control as Applied to the 220 Computer, E. L. Glaser ... 63
The RCA 501 Electronic Data Processing System, Glen E. Poorte, Arthur S. Kranzley ... 66
The Univac M-460 Computer, J. E. Thornton, M. Macauley, D. H. Toth ... 70
A Special-Purpose Solid-State Computer Using Sequential Access Memory, W. A. Cornell ... 74
Discussion ... 80

CONTRASTING TOOLS AND TECHNIQUES FOR SIMULATION, D. Madden, *Chairman*

Electronic Differential Analyzers in Perspective, John McLeod ... 82
The Case for Combined Analog-Digital Simulation, Walter W. Varner ... 86
Digital Computer Solution of Differential Equations in Real Time, H. J. Gray ... 87
Discussion ... 92

ACTIVE ELEMENTS FOR THE MACHINE, R. E. Meagher, *Chairman*

Switching Transistors, I. M. Ross ... 93
Special-Purpose Tubes for Computer Applications, Saul Kuchinsky ... 96
Superconductive Devices, A. E. Slade, H. McMahon ... 103
Magnetic Switching, Jan A. Rajchman ... 107
Discussion ... 117

COMMAND STRUCTURES, J. W. Carr, *Chairman*

A Command Structure for Complex Information Processing, J. C. Shaw, A. Newell, H. A. Simon, T. O. Ellis ... 119
The Selection of an Instruction Language, W. Buchholz ... 128
System Design of the Gamma 60, Phillippe Dreyfuss ... 130
Discussion ... 133

DIGITAL COMPUTER CIRCUITS, S. Ruhman, *Chairman*

A Direct Read-Out Bistable Circuit and Some Applications of It, Ir. H. de Miranda, I. Rudich ... 134
Flow Gating, W. J. Poppelbaum ... 138
Minimum Transistor Logic Modules for Air-Borne Control Applications, Arney Landy, Jr. ... 141
Transistor Magnetic Core Bilogical Element, W. J. Dunnet, A. G. Lemack ... 144
High-Speed Circuit Techniques Utilizing Minority Carrier Storage to Enhance Transient Response, L. P. Retzinger ... 149
Discussion ... 156

DIGITAL COMPUTER APPLICATIONS, R. C. Minnick, *Chairman*

A Chess Playing Program for the IBM 704, A. Bernstein, M. De V. Roberts, T. Arbuckle, M. A. Belsky ... 157
Applications of Digital Computers to Problems in the Study of Vehicular Traffic, Walter Hoffman, Richard Pavley ... 159
The Role of the Digital Computer in Mechanical Translation of Languages, David L. Johnson ... 161
The Application of a Large-Scale Electronic Computer to the Assignment of Telephone Facilities, Milton Drandell ... 165
An Experiment in Mechanical Searching of Research Literature with RAMAC, F. E. Firth ... 168
Discussion ... 171

Contents (con't)

Page

LOGICAL DESIGN METHODS, G. M. Amdahl, *Chairman* ... 176

Block Diagrams in Logic Design, Lowell S. Bensky ... 177
Logical Design Methods, R. K. Richards ... 179
Machine Language in Digital Computer Design, H. L. Engel ... 182
The Advantage of Logical Equation Techniques in Designing Digital Computers, Victor L. Hesse ... 186
Discussion ... 189

VERY LARGE FILES, J. A. Postley, *Chairman* ... 193

Methods of File Organization for Efficient Use of IBM RAMAC Files, W. P. Heising ... 194

SPECIAL DEVICES II, Mr. Rogers, *Chairman*

The Design and System Aspects of the HD File Drum, H. W. Fuller, S. P. Woodsum, R. R. Evans ... 197
Transistorized Modular Power Supplies for Digital Computers, Theodore C. Gams ... 203
The Shiftrix-Machine Organization for High-Speed Digital Computation, Gerald Estrin ... 207
A Device to Facilitate Combined Analog-Digital Computation, B. L. Schwartz, G. Jenkinson, L. Winslow, B. Gordon, J. Solomon ... 212
Discussion ... 215

DIGITAL COMPUTER SYSTEMS II, P. Levonian, *Chairman*

Communication Between Computers, William S. Knowles, Irving L. Wieselman, Raymond Stuart-Williams ... 216
The Universal Data Transcriber. A New Approach to Data Conversion Equipment, Marvin S. Maxwell ... 225
A Universal Computer-Language Translator, R. B. Bonney ... 230
A Computer Oriented Toward Spatial Problems, S. H. Unger ... 234
The Magnetic Ledger Card Computer, Thomas P. Holloran ... 239
Discussion ... 243

The Social Problems of Automation

H. T. LARSON
NONMEMBER AIEE

DURING this conference a considerable number of discussions will be presented on the subject of computer design. Such computer conferences have been conducted for seven or eight years. The computer industry can be reckoned to be about ten years of age. During this time, a large number of computers, data processors, and other descendents of computers have been produced. Because of the time span over which the early machines have been in operation, and because of the rather large and increasing numbers of these machines that are now being produced and used, it is reasonable to say that the world has now felt a considerable cumulative effect from the work these machines have performed.

How have these machines been of help? A very broad and overlapping classification can point to three major areas of activity:

1. Science and Engineering. Computers have proven to be unusually powerful tools wherever computation and numerical analysis are employed. The speed or economy of these machines have made it possible to perform computations and analyses which were formerly infeasible. The total effect of the application of computers undergoes an enormous amplification, because of the great variety of scientific and engineering fields in which computation contributes, and because of the never-ending application of scientific and engineering discoveries. Our everyday life is affected in a multitude of ways.

2. Military Control Mechanisms. A considerable fraction of the computer design effort over the past 10 years has has been aimed at the incorporation of these devices in weapons of warfare. They are being increasingly applied where men or other equipment are proving to be too slow, too large, too weak, too inaccurate, or too unreliable. The military work can be measured in terms of more powerful weapons and greater military strength. As usual, the techniques evolving from military effort find wide and useful application in nonmilitary fields.

3. Business Data Processing and Industrial Control. The capabilities of computer-type equipment are applicable to the performance of business processes such as financial accounting, sales forecasts, inventory control, study of alternate plans of action, etc. Furthermore, computer equipment is finding increasing application in the control of complex industrial processes. In this role, the computer is beginning to play an important part in the continuing industrial revolution. This revolution has been characterized by increasing replacement or augmenting of men's muscles, starting with the mechanization of simple, repetitive, manual tasks and moving to more and more powerful and complex mechanizations. In contrast to the mechanization of manual tasks, the computer can perform some of the types of activities that men perform mentally. This simple fact lies behind the ability of the computer to replace men in control of functions. Again, simple and repetitive mental or information-processing tasks are among the first to be mechanized, and more complex controls are on the way. They are likely to be used whereever they outperform men in terms of speed, accuracy, or operating costs. The modest progress to date in this area indicates that the total effect of such applications is going to be extensive.

What has been the over-all effect of the work these machines have done? Is it good or bad? How has it affected everyday life, health, politics, work, education, and leisure? Are new problems being created, are old problems being compounded, or is this computer activity all for the good? These are the questions asked of our opening session speakers. Are there any social problems associated with automation? If so, what are they, and what is the proper way to meet them.

H. T. LARSON is with Aeronutronic Systems, Inc., Van Nuys, Calif.

The Social Consequences of Automation

HAROLD D. LASWELL
NONMEMBER AIEE

A COMPREHENSIVE analysis of the impact of automation upon society would consider every sector of the social process. In the process of social interaction man pursues "value outcomes" through institutional practices utilizing resources. He pursues knowledge (enlightenment), political power, and various other value outcomes.

Enlightenment Outcomes

The development of computers has evolved a formidable instrument of enlightenment. Whether this potentiality is realized in practice depends upon many factors, of which one is the recognition of what can be done. An enlightened opinion on any problem of policy is a complex matter contingent upon the performing of five intellectual tasks: the clarification of goals, the perception of trend in the degree to which goals have been achieved, the analysis of the factors that condition trends, the projection of future developments (if policies continue unaltered), and the invention and evaluation of alternative policies.

Automation technique can be of enormous help in performing these intellectual tasks for every problem of public policy. There is no reason why working models of social history and of the future should not be part of the ordinary equipment of educational institutions and of institutions of civic decision. The entire social process can be portrayed in alternative models which show such estimated consequences as the following: the effect of at least two levels of expenditure upon arms; the consequences of cutting the hours of work over a given period to 6, 4, and 2 hours a day; the consequences of making available at various rates a cheap oral contraceptive to the populations of specified countries; the effects of altering the residential arrangements of the population in centers

HAROLD D. LASWELL is with the Yale University, School of Law, New Haven, Conn.

and subcenters of various size and location; the results of lengthening the life span during assumed periods of 10, 20, or more years, or even of abolishing death.

If every neighborhood, city, region, private association, and organ of government is to have continuing access to clarified information new institutions must be devised to supplement our present channels of public information. For instance, a means of popular instruction that has been applied to astronomy can be extended to the field of public opinion and decision making. The author refers to the possibility of a social planetarium. The social planetarium will enable visitors to rehearse the past and to foresee the future. Alternative policies relating to economic progress and stabilization, for example, can be presented step by step. In this way the meaning of alternative courses of action can be grasped by most of the population, not only by a specialized few. Competing interpretations can be candidly set forth in presentations approved by competent specialists so that the viewer-participant can arrive at a well-disciplined estimate of the likelihood of one or another outcome. Computing instruments are indispensable to the processing of vast bodies of data pertinent to the needs of social planetaria.

One advantage of automation technique is that it permits more exact appraisals to be made of the functioning of courts, administrative commissions, and other organs of decision. It is probably feasible to translate the routines of some modern logics into machine routine; this would mean that it is feasible to program the tax code or any other system of allegedly logical prescription for machine handling. The effort to do so will disclose many removable ambiguities. It will also make it practicable to test the strength of other-than-logical factors in bringing about decisions by pitting machine models built to function logically against contemporary and historical agencies.

Political Outcomes

Previously the discussion was about the possible impact of automation upon an enlightened view of puzzling and controversial issues. This also has great significance for the survival of popular government, which, as political philosophers and political scientists have pointed out, depends upon an informed public. Now consider the direct impact of automation upon political power. The obvious and basic point will be emphasized that no technical operation or program is introduced into a social vacuum. Nor does automation, per se, have "inherent" collective consequences. Factors in the context determine the aggregate effect by affecting the location, timing, and scale of the operation or program.

At first glance automation would appear to constitute a landmark in the historical processes that favor democratic forms of government. The machine does at last become the slave and give free time to free men for the consideration of public business. But since the effect of more leisure depends upon many factors in the social context besides leisure, it cannot be safely concluded that the sequence from automation, leisure to democracy is certain. Concern for public affairs and willingness to take an active part in civic life depend upon the perspectives of those concerned. There is evidence, though not definitive, that as modern Americans have more leisure they vote less often in elections and they expect pressure groups, executives, political party machines, the press, and public officials to look after the common good. Those who concern themselves with the distinctive specialties of modern times, such as science and engineering, are not as a rule conspicuous among the active leaders of community policy at local, state, national, and international levels. Evidently there is an assumption that public affairs is a specialty and that civic judgment has no peculiar opportunities or obligations.

In the intricate web of modern life, to civilize is to specialize. But a governing process is needed that continually clarifies the goal values of the whole community and appraises the degree in which current trends coincide with these value objectives, or diverge therefrom. An exercise of judgment is called for in which the individual acts as a whole person, not merely as an engineer, the head of a family, the alumnus of a college, a raiser of orchids, a member of a social fraternity, a man of international travel, or as one who plays a hundred other roles. The civic process is the governing process in which, ideally, contradictions and tensions are resolved in a common plan of purpose and method.

If democratic forms are to be kept vital, styles of life must be developed which take advantage of the leisure time that automation can afford. During basic training in the educational system some familiarity should be acquired with the decision-making process of society, and of the place of individuals and organizations within it. During the early phases of intensive professional education and practice, it is probable that one will become too specialized to play a significant civic role. But this phase may typically pass into a stage of professional growth at which one becomes an administrator of research or moves to general management. If one remains at the research level, one's development typically becomes more routine and less absorbing of the whole energy of the personality, leaving more margin for other concerns, including civic affairs. Even when most acutely preoccupied professionally, one usually belongs to professional and other associations which are equipped with staffs and committees. By supporting associational activities which are directed to public issues one may contribute somewhat effectively to civic life.

In these days, one cannot afford to overlook the world of politics. Does the advent of automation promise to intensify the world crisis or not? The fundamental fact in the global arena is the expectation of violence, which is the assumption that wars, though not necessarily global wars, are probable. Given the expectation of violence, the powers in world politics group themselves in opposing coalitions for defense and offense. Contemporary world politics conform approximately to a bipolar pattern in which the United States and Russia are dominant. In this setting, the continued growth of science and technology has contributed to an armament race of unprecedented danger. Automation speeds up the tempo of whatever it touches; and it is touching the preparation of arms and the conduct of mutual surveillance. Hence, automation speeds up the crisis and accentuates the bipolar antagonisms of world politics.

Will this result necessarily continue? There is evidence that automation technique makes it easier now than in the past to install modern technology anywhere on the globe. The question is whether the United States will supply enough investment capital to countries in the non-Soviet world to take advantage of this opportunity to develop strong allies; or whether we will underwrite only enough examples of superior technology abroad to provoke resentment that the United States has not done more. The top and middle leaders of the former colonial peoples are especially sensitive to this situation. Moreover, it cannot be taken for granted that automation will be installed by managers

who use policies relating to labor, public, and governmental relations which will keep difficulties at a minimum.

Is it likely that the middle powers (like Great Britain, France, or West Germany) will be able to improve their political position in relation to the United States and Russia, as they make fuller use of automation? Or will the strong grow stronger as the automation revolution proceeds? The author invites expert judgments here. This is a crucial area for joint estimates by physical scientists and engineers, and political scientists. The tentative expectation is that bipolarity will be sharpened and that the balance of power will not revert to the pattern of the last century, which was dominated by relations among a few great powers.

Perhaps it is easier to predict that, as automation advances, centralization will continue in the handling of all instruments of national policy, whether military, economic, diplomatic, or idealogical. The necessity for comprehensive and instantaneous information, coupled with the need of operational co-ordination, are factors that further centralize control.

The trend to centralize is also a trend to governmentalize. The cost and scale of military preparation favor the spread of this influence from national departments of defense throughout the whole economy. The new instruments of automatic operation, in this context, give hands as well as eyes and ears to official directives.

An examination of the newer technology discloses a remarkable impact upon the unit of effective military action. The most complex weapons and weapon centers are supervised and tended by rather small teams. Infiltration and surprise depend upon small, agile units; and sabotage and espionage networks call for such structures. With the increased use of automation, components of the total machine become vulnerable to the well-placed few.

Automation has an important psychological effect by reducing emotionality in combat preparations or activities. The machine intervenes between fighters; the destruction of a target is a result of a number of complex and impersonal sequences. In a sense modern war and war preparation are always cold and impersonal.

Cutting down on emotionality puts the emphasis upon intellectual appeals either to maintain or to undermine the dependability of personnel in sensitive positions. The ideological instrument of world politics is made more important, since it may be used to reach the key individuals in man-machine combinations.

In all likelihood, owing to pressures for social conformity, automation is having and will have large repercussions upon respect for individuality. Modern technology has developed a repertory of devices capable of penetrating barriers of privacy. Some of these relate to behavior such as the recording microphone or concealed photography; others refer to the inner life (narcosis-inducing devices, lie detection polygraph machines, etc.). Although traditional regard for freedom from snooping has erected many safeguards against arbitrary police investigation, these barriers are already dissolving in the name of security and loyalty inquiries, and of "moral" or "job" pressure to submit voluntarily to deception tests. In the past, the efficiency of police networks suffered from shortage of personnel. The installation of automatically monitored surveillance instruments makes it possible to penetrate the remaining barriers to privacy, and to redouble the pressures toward cautious conformity, not only to lawful prescriptions but to the informal prescriptions laid down by "Mrs. Grundy." The potentiality exists of monitoring not only prisons, schools, offices, plants, barracks, training, and recreational fields, but of surveying traffic flows, etc.

Limits to this process may be wanted in order to maintain areas of individual privacy and freedom. If so, it will be necessary to stop drifting and take the positive step of drawing up, adopting, and administering codes of freedom. Otherwise, the world will be comfortable only for people who have no unconventional impulses, no unpretty habits, no objectionable behaviors of any kind, no novel conceptions of rectitude. Man will be approaching the time when he automizes himself into conformity, into seeming rectitude. Paradoxically, a license to be unobserved for awhile may become one of the-principal rewards of meritorious conformity.

Affection Outcomes

Another important set of social outcomes relates to affection, not only in the sense of intimacy and friendship, but of emotional identification with such social units as the "Nation State." Automation requires the long-term trend to continue, which increases the amount of invested capital per producer. It is often said that this has brought about the dwarfing of the individual by skyscrapers, giant power plants, and such. To some extent men proudly identify the primary ego, which is relatively puny, with gigantic achievements of the kind. At the same time, however, many careful investigators allege that our large-scale modern society alienates many millions of its members. Attachments are slowly withdrawn from the larger and more inclusive entities to smaller and more compassable units. Presumably, this alienation affects the United States, Great Britain, and all of the older industrial nations more than it does, for example, the Soviet world. Where technological progress is relatively new, fewer elements in the population are disenchanted with its results in terms of art, morality, or other values of their kind.

Well-Being Outcomes

In a related vein are some questions about how mental and physical well-being is connected with technology. One sequence runs from the technological environment to an impact upon affection (alienation) or rectitude (rigid conformity), which has already been mentioned. Do the pressures of alienation or conformism in turn affect the psychosomatic equilibrium of individuals? Is the demand for tranquilizing drugs, for instance, to be interpreted as a symptom of cumulative stress whose origin is partly in the technological environment which automation intensifies? Can any significant part of the rising incidence of neurosis, of "acting out," and of other psychiatric disorders be attributed to the same set of factors?

Skill Outcomes

To offset whatever adverse effects there may be in terms of well-being, the effect of less drudgery per day upon human creativity can be weighed. If in fact, the hours of compulsory or semivoluntary work are cut down, perhaps human abilities will seek creative outlets on a hitherto unimagined scale. Undoubtedly this will give rise to problems. The ungifted, for example, are problems. But some of those who appear devoid of wit and talent are victims of our current inability to perceive new talent potentials for new and contributory modes of expression.

Respect Outcomes

A brief word about respect outcomes. Scientists and engineers are today en-

joying much public appreciation. This will grow more stable and rational as civilization takes its shape. But there are many tensions of transition, not the least of which is the predicament of literary and artistic men and women. One brigade of writers and artists is articulately unhappy with the times; they are conducting a rear-guard action that stigmatizes the techno-scientific culture of the age, and dramatizes the inner turbulence of the alienated in spirit. It is, perhaps, true that much of this is a reaction to a sense of humiliation; and that the humiliation comes from unwillingness or incapacity to acquire the knowledge essential to achieve a sense of belonging to the world of our epoch. If modern man is to be safe from crises resulting from resentments that stem from humiliation, it is important to speed up the process of sharing at least a minimum comprehension of the meaning of the scientific outlook. Surely, with all our resources and intelligence, this is not beyond our capabilities.

The author has said nothing about economic outcomes. Others will look after that. The aim of this paper has been to suggest some of the impacts of the process of automation upon other outcomes in the social process such as enlightenment, politics, rectitude, affection, well-being, skill, and respect. The greatest of these, in terms of potential, is enlightenment.

The Social Problems of Automation

B. J. SCHAFER
NONMEMBER AIEE

AMERICAN industry appears to be moving into a period of technological change which staggers the imagination. While a layman may have read a great deal about the new automated processes, and may have had considerable actual experience with the practical impact of automation, it is difficult to visualize fully the changes which are taking place.

Americans are fascinated with the new technology, and are a people who always have been entranced by mechanics, by science, and by the wonders which can be accomplished by machines and instruments. Americans are inclined to glorify mechanical, technical, and scientific progress. But the current revolutionary change in technology has reached a point that is sometimes frightening. In the past technical progress has come about by relatively easy stages. Each new invention affected only a limited segment of industry and economy. Each new machine represented only one step forward on an old process. Mechanical programs did not jump overnight from hand shovels to 10-cubic-yard earth-moving machines, but instead moved by easy stages over a period of decades. Each of us, as amateur mechanics, could savor and enjoy the step-by-step changes in the machinery we loved to watch in operation.

Now, automation has brought about the expectation of overnight revolution in the way we apply energy and build things. The change not only threatens to be abrupt, it also is so great as to outrun the imagination. Laymen have to some degree lost their capacity to enjoy watching change as spectators, and are hard put to understand the new devices which are coming into factories and offices.

We continue to admire, indeed, we continue to virtually worship, the scientists who create the fantastic new methods and machines. Yet we are filled with apprehension, for there is a vague and uncertain feeling that these scientists are so engrossed with the almost supernatural nature of their instruments that they have forgotten the natural needs of mankind. The author does not make the charge that the scientists have forgotten the people; only that many people have a feeling of apprehension. At times the trade union movement shares that apprehension when we see the industrial applications in certain cases. I am sure the scientists are just as interested as I am in dispelling this apprehension and creating a climate in which the layman can trust the scientists, and the scientists can feel that they are working solely for the welfare of people.

Panel discussions such as the one which produced this group of papers certainly contribute to a better understanding between those gentlemen who are working electronic marvels and the laymen who are amazed by it all. Discussions like this can serve to provide us with mutual education and perhaps a better understanding.

Speaking as a representative of organized labor, I can assure you that labor does not oppose automation as such. We are wholeheartedly in favor of the application of better, faster, easier, and cheaper methods of performing work, so long as these methods are applied with due consideration of the needs of the people. My only thesis is that the technology of man must always serve people; people must never be made the slaves of technology.

Organized labor has no doubt that in the long range automation will benefit the human race. Anything which increases the production of goods and services for the use of people, and anything which reduces human toil, ultimately will bring about a more pleasant life. Labor does not want to return to a past era of hard manual labor with picks, shovels, and other hand tools. Mechanical progress already has relieved labor of much of its unpleasantness and we certainly do not want to stand in the way of technological progress which will relieve us of still more toil.

This is our long-range view. In the short range, labor resistance will be found from time to time in the application of new automated processes which bring hardship to individual workers and groups of workers who are displaced by new equipment, with resultant personal and family dislocations. Whenever new equipment is applied under circumstances which take into consideration first the welfare of the people concerned and secondly the question of mechanical efficiency, there will be no substantial resistance from labor. But whenever new equipment is installed purely on the basis of production efficiency and without regard for the human beings who suffer thereby, resistance and resentment will be built up.

All changes in our way of doing things, whether these changes be economic, administrative, or technological, should be made on the basis of what will best serve the people. Labor believes that industrial planning should start

---

B. J. SCHAFER is with the Oil, Chemical and Atomic Workers International Union, AFL-CIO, Denver, Colo.

from the first premise: "How will this affect people?" and that other factors should be secondary.

In considering the needs of the people in applying automation, the problem should be considered from two standpoints:

First, the general economy of the United States and the free world must be considered and there must be general economic adjustments to accommodate the increased production made possible through automation.

Second, the welfare of the individuals directly affected by each specific case of automation must be considered and provision must be made to eliminate or at least minimize the hardship of change on these individuals.

Taking these two considerations one at a time, the general economy will be commented on first. In organized labor, and particularly in the industrial unions where the most direct effect by automation is seen, labor's economic philosophy is quite simple. We believe that above all there must be maintained a balance between productive capacity and purchasing capacity. Whenever there is more purchasing power than there is productive power, inflation results. Whenever there is less purchasing power than there is productive power, the results are declining markets, curtailed industrial operations, and a recession such as the United States is experiencing today.

As production increases, so must purchasing power. Theoretically, this could be accomplished while holding wages level by cutting prices in the same proportion that productivity rises. Practically speaking, in the society of pressure groups, it seems virtually impossible to bring about price cuts. Therefore, we revert to the position that wages must increase in proportion to the increase in productivity if we are to maintain a balance between purchasing power and productive power.

Over the long haul, American productivity as a whole has in the past increased at a rate of about 3% a year. Over the long haul, labor has managed to secure wage increases, that is, increases in real wages, not counting inflation, at approximately the same rate. This means that in the long run there has been maintained a productive power-purchasing power balance. There have been notable exceptions to the over-all trend; for example, in the late 1920's, purchasing power declined below productive power and the United States had a terrible depression. In the past few years there has been another decline in purchasing power, relative to productive power, resulting in a recession.

The rapid advancement of productive capacity through new automated processes can bring about maladjustments more sudden and more severe than we have experienced in past occasions of relatively simple mechanical advancement. In the past, new equipment in a given situation increased productivity by only a few percentage points. Some of the actual and projected automation projects of today can increase productivity in a given process by several hundred per cent, and can do it overnight.

It is going to be difficult to make economic adjustments necessary to maintain our production-consumption balance as rapidly as the technological changes can be made because the elimination of entire departments, geographical changes in business locations, and disruption of competitive relations of companies caused by these technological changes create work force dislocations affecting great segments of the economy and population. This creates problems that stagger the imagination. But these changes must be made or chaos will result.

The law of supply and demand cannot be depended upon to make the necessary adjustments. The law of supply and demand appears to have been repealed, for all over this nation oversupply and rising prices are seen side by side. For example, the automobile factories have been partially automated, productivity has vastly increased, there is an oversupply of cars, yet prices do not fall. The oil industry already is highly automated, productivity is increasing at the rate of six or seven per cent a year, the supply is overflowing the storage tanks, and still prices stand firm.

Since the law of supply and demand no longer appears to operate, it is obvious that some other forces must be applied to maintain balance. The labor movement seeks to restore the balance by securing higher wages. These higher wages would compensate for increased productivity and would provide a consumer demand in line with the increased supply of goods and services. labor is under considerable criticism because of its constant pressure for higher wages, but in fact the insistence on raising wages is the only force that labor can see which is at present working toward the maintenance of production-consumption balance.

Of course the balance between production and consumption can also be maintained through working shorter hours. Certainly that work which is available should be distributed equitably to all, whether that means a 40-hour week, a 30-hour week, or even a 20-hour week, if we are to maintain consumer purchasing power. Already there is a slight downtrend in working hours, not only in the reduction of the workday and the workweek, but also in the reduction of the workyear, through the reducing of overtime and the granting of longer vacations and more holidays. These current reductions are but a drop in the bucket compared to what may be possible in the future.

Organized labor favors shortening of working hours because it believes that with shorter hours and modern technology, enough can be produced for all. However, this approach to the balancing of purchasing power and producing power is secondary in our opinion to the more beneficial approach, in the immediate future at least, of raising our consumption levels still higher by increased wages. There are still many people who do not have a decent standard of living and there are a multitude of others who lack many of the modest conforts and luxuries. There is a great deal of room in this country, despite its appearances of prosperity, for the development of a higher standard of living. Roughly 20% of our families still earn less than $2,000 per year while the studies by the Heller Budget Group establish a recommended minimum budget requirement for a family of four at over $5,000 per year in most of our urban areas.

Labor strives for higher wages and for shorter hours. This, we believe, not only benefits working people but helps to balance the economy to the benefit of all segments of society. An alternative to our role as a pressure group balancing the economy, and I grant we accomplish this end rather unevenly and uncertainly, would be a planned economy in which some authority or other would dictate the amount of production to be permitted, the amount of labor to be employed, and the amount of purchasing power to be made available to labor and other consumers. There is tremendous resistance in this nation to any sort of planning of the economy, and labor generally shares the opposition thereto because we generally feel it is contrary to the principles of our democratic way of life; but all of us, in every line of endeavor, be it investment, management, engineering, or labor, should bear in mind that if we do not show the statesmanship to maintain a balanced economy,

then there will be chaos, and planned economy may be forced upon us.

And before the idea of a planned economy can be entirely cast aside on the grounds that it is onerous and smacks of dictatorship, we had best give some thought to some degree of economic planning within the framework of our democratic institutions or else we may increase the danger of a dictatorially planned economy at a future date.

Frankly, organized labor has serious concern over the ability or willingness of American industry to provide the necessary leadership or statesmanship to maintain balance without planning or to permit planning within our democratic system. American industry has demonstrated again and again, and is daily demonstrating at present, that it is primarily concerned with profits and largely unconcerned with the general welfare. While each individual corporate official may concede that something needs to be done to maintain economic balance, none is willing to take the lead in definite action toward this end. The social leadership of many of our important industrial leaders today seems to be limited to the making of speeches criticizing labor unions when in fact the pressure of unions to maintain wages and therefore purchasing power, is today the only force maintaining a semblance of balance.

The economy during the past several years and up until approximately the past year has been able to absorb the impact of automation. Our ability to consume has kept pace with the increase in productivity made possible through new technology, thus maintaining a healthy economy. Speaking of the industry with which I am most intimately acquainted; I will point out that the oil-refining industry increased 51% in its productive capacity and the market demand for its products during the decade of 1947 to 1956. This came about, by the way, with the number of productive workers employed remaining virtually unchanged during the decade. In fact, the number of blue-collar workers in the oil industry actually has declined slightly in recent years.

So we had in the oil industry a labor force of approximately unchanging size, a rapid acceleration in productive capacity due to new technology, and a market which would absorb that expanding production. During this same period, domestic oil demand has gone from 5,452,000 to 8,797,000 barrels, and refining throughout has risen equally in volume in that period. But in the past year, the market demand for oil products has levelled off. It has not declined, at least not more than about one percentage point; it has simply stood still. And this mere levelling off of demand has caused troublesome upsets in the oil industry. Thousands of workers are being laid off and now the companies even claim that their profits are dropping, although they are still quite handsome. This demonstrates the point that we cannot tolerate expanding production in the absence of expanding markets. This demonstrates that purchasing power must increase; it is not enough that purchasing power stand still, that it rest upon a plateau; it must expand.

Perhaps you feel that I have dwelt too much on economics. But economics and automation cannot be separated. It would be pleasant if we could consider automation as a thing apart, but we cannot, for automation is the biggest thing that has hit economics since the development of the capitalist system. At all times bear in mind the effect of automation on economics and thereby the effect on society. Those men who would apply automation to our machinery have an obligation to consider its economic impact, and they have a responsibility to support such measures as may be necessary to maintain economic balance.

Of course no single case of automation will within itself destroy the economic balance. But a single installation in a single plant or office can be quite upsetting to the human beings employed there. We have built up a system of society in which nearly all working people are pretty well committed for life to a given job. To move to new employment, if possible at all, is difficult and costly. Each individual's security tends to be wrapped up in a single corporation and he cannot move at will from one company to another without losing the benefit of experience and training which often is not transferrable to a new job, and without impairing economic benefits such as pension rights and the like. Further, most industrialists simply refuse to hire a man more than 40 years old, and for some jobs he had better be under 30.

So there is a situation in which a human being becomes part of a particular machine or a particular corporation. He doesn't always fit in as an integral part of another machine. There has been less progress and less attention devoted to the interchangeability of people than in the interchangeability of our precision-measured machine parts. These people who are rendered obsolete or displaced by automation cannot be cast aside as are the old machines rendered obsolete by new technology. They cannot be sold for scrap at the Pittsburgh mills. They have to be considered and treated as human beings.

Therefore, in each case in which automation is applied, provision must be made for the people displaced. If the economy is expanding and if the particular company involved is expanding, it is often possible to transfer people from one job to another within the company. This has been taking place in recent years, or at least it had been until the current recession came about. Even these transfers involve problems. A man who has spent 20 years learning the skills of operating a piece of equipment should not be demoted to an unskilled rate of pay when, through no fault of his own, that piece of equipment is replaced. Every effort must be made either to retrain him for work on the new equipment or to maintain his established rate of pay even if no similar work remains available.

Many companies are putting extreme pressure on middle-aged men to retire these days. Men of 50, with 15 years to wait for a pension and with weighty family problems are being cajoled and pressured to take early retirement on inadequate pensions, to take pay cuts, to take heavy work they cannot handle. This is simply not an acceptable solution. The more often it happens, the more bitterness will build up among working people.

Labor recognizes that it is sometimes costly for companies to take care of the surplus manpower found on hand when they automate, but we believe that the companies should make provisions for taking care of the people just as they do when they make financial provisions to write off old equipment and arrange to buy multimillion dollar new production units. As in these cases, money can be ear-marked to provide for the old people. Alternatives must be found to the loss of dignity, worthiness, productivity, and self-respect which comes to a man when he is denied an opportunity to work and earn a living. If such alternatives cannot be found, then no machine, no method, no technology is worthwhile for all of the wonderful production it can turn out.

In summary, I will say that in the application of new technology, every person concerned should adopt a motto

such as: "Consider first the human being." We should ask ourselves the following questions, both in general and in each new case under study: "Does this new process help the human race? Will it bring more good things to the people? Will it destroy the usefulness and the dignity of a single human being?"

The answers to all of these questions are important. Automation does bring the promise of more good things for less toil. Properly applied, it will bring to the most humble citizen a standard of living that no one dreamt of 50 years ago. But in all cases, proper application must be made, with due consideration of the human beings who are affected by the almost supernatural machines which we are now putting into use to increase efficiency and production.

The Social Problem of Automation

CUTHBERT C. HURD
NONMEMBER AIEE

RATHER than the announced title, the subject might more appropriately be: "Automation As It Has Demonstrably Affected Man's Working, Man's Playing, and Man's Thinking." The field of electronic computing which is today identified with automation, is now 10 years old. Consequently, there is no need for speculation concerning what might happen in the future. Rather, examine what has happened in the 10 years just past. Ten years ago it was thought that all manner of social problems might be created by the introduction of the electronic computer. In the light of experience, it can now be stated that given appropriate foresight and planning, the introduction of electronic computers can be accomplished in such a fashion that the fears of 10 years ago are proved groundless. Automation has extended man's capability as a worker and as a thinker.

The experience on which the analysis can be based is indeed great. During the past 10 years, computers have been applied in the field of science, in the field of business, in the field of the military, and are now being thought of increasingly for applications in the field of automatic process control. There are at least 200 giant machines in use; there are at least 1,000 middle-sized computers in use; and there are several thousand small electronic computers in use. Surveys have been made concerning the usefulness of these machines and their effect upon the people surrounding them. Case studies have been made and the results reported at scientific meetings. Moreover, we are still a small fraternity; we see each other frequently, and speak frankly concerning our experiences. Thus, a great deal is known about our 10 years of experience, and examples will be used rather than statistics in support of the statement that the fears of 10 years ago were groundless.

The opening statement indicated that the discussion would concern automation and information processing as they apply to man at work, man at play, and to man's thinking. The author would now like to withdraw the second topic because it is his observation that those who are engaged in the field of automatic information processing are faced with so many absorbing and exciting fields for development that they are finding their leisure and their playtime in their work. Stated directly, most of us are working harder than ever. This might mean that one man's leisure is another man's automation and it reminds me of the story of an American meatpacking executive who, vacationing in Spain, visited a bull fight in Barcelona. "Wasn't it interesting?," asked a Spaniard, later. "Yes, it was," admitted the visitor, and hesitated a moment, trying to be polite, "but it's an awful bother, isn't it? We do all that automatically back home."

Now to relate automation in information handling to man working and to man thinking.

Man Working

This section encompasses three topics: New professional opportunities, position enlargement, and co-operation between professional people.

New Professional Opportunities

It is clear that a whole new profession has arisen in a fashion which interacts with the development of electronic computers. Not only has a new profession arisen, but there has been an extension of a number of existing professions. The new profession is that which is commonly known as programming. This word is not completely precise and is used variously to include problem analysis, programming per se, coding, and, on somewhat rarer occasions, the word programming is used to include also the task of carrying out the solution of the problem on the machines. If programming is thought of particularly in one of the first three senses, it is clear that there is a professional activity involved. That is to say, it is possible to describe in a formal manner the abilities which are required for success, the activities which comprise programming, and the educational attainments or experience which gives promise of success in the field. Indeed, there are now perhaps fifty universities in the United States which have inaugurated courses of instruction in the field using medium or large-scale machines for laboratory purposes. Based on the number of machines installed and on the magnitude of the educational effort, both on the part of manufacturers and universities over the past few years, one can estimate that there are some tens of thousands of professionals already engaged in the field of programming. This profession is growing vigorously.

In addition to the new profession, the desirability of applying automation in information handling has been realized in almost every branch of the physical sciences, the social sciences, and beginning now, in the humanities. Thus, it is common to require that students in some of the engineering sciences be required to learn computing, and indeed, Professor Morse, Director of Computation at Massachusetts Institute of Technology once said that he looks forward to the time when every graduate at Massachusetts Institute of Technology will be familiar with computation and will be able to use it as a tool in his profession.

This new profession and the extension of existing professions has been described not only because of its importance in itself, but because the new profession has a direct relation to position enlargement, which is the next topic under discussion.

CUTHBERT C. HURD is with International Business Machines Corporation, New York, N. Y.

Position Enlargement

Position enlargement is an important consequence of automation in information handling. As automatic methods have replaced either semiautomatic methods or manual methods, workers have perhaps joined the newly created profession of programming or they have been given an opportunity to remain in existing positions, but in such a fashion that their work is less routine and more exciting. That is, their positions have been enlarged; they are more demanding, intellectually. There has been much in the literature on this topic, and therefore this paper will only cite an example that is directly within the author's experience. In 1954, it was decided that in the International Business Machines Corporation plant in Poughkeepsie, the production control perpetual inventory system would be placed on a *702*. (The *702* has now been replaced by the *705*.) A program of orientation for all of those engaged in the perpetual inventory department was immediately started; and as the planning, coding, and installation was accomplished, the orientation and educational program was continued. Many of those who were initially engaged in the inventory department in the plant have continued to be so engaged, but with two differences. First, productivity in the department has greatly increased because of the faster availability of more accurate information. Secondly, the character of the work has changed sharply. The work is now primarily inductive in character and consists of arriving at plans of action with respect to machine parts which are reported by the *705* as requiring attention. The position has been enlarged with the consequent increase in professional satisfaction.

Co-operation between Professional People

In this section the discussion turns to co-operation between those who are engaged in the field of automatic information handling. It has been demonstrated repeatedly that electronic computers create their own environment. Thus, the satisfactory application of machines to one task immediately seems to suggest additional uses. Therefore, there are always many tasks waiting to be approached. We never catch up.

It is imperative, therefore, that duplication of effort be eliminated wherever possible, and thus there have originated co-operative efforts among programming groups. The author has often marveled at the extent and at the effectiveness of this co-operation. In an economy which is highly competitive and in which proprietary information is held with great care, the co-operation among programming groups has grown rapidly and is illustrated, for example, by the SHARE organization.

This section will be concluded with the remark that as a result of the introduction of automation in information handling, there has grown up a new profession, existing professions have been extended, workers have been allowed an opportunity to enter this new profession, workers have been given the opportunity for additional education, workers have been given the opportunity for position enlargement, and finally, there has arisen a great amount of co-operation among groups which might normally be regarded as competitors.

Man Thinking

The most important effect of automation on man is its effect on the extent and the power of his thinking. First, man can now contemplate certain adventures which would be impossible without automatic methods of information handling. Secondly, perhaps it may be that the productive power of man's thinking has been increased by and through the use of computers.

With respect to what man can now contemplate, only two illustrations will be given. First, it is apparent that any idea of space travel depends directly upon using a computer as a designer and upon using a computer as a controller. In this field the only limitation on distance of travel is considered to be the limitation of the speed of light, and even this possible limitation has been challenged recently.

Secondly, because of automation in information handling, it is now possible to contemplate a future simulation of the United States economy and the answer to questions such as are presently widely discussed: Should the United States change economic parameters (i.e., tax rate) and thus attempt to terminate a recession, or not?

An effect then of automation is that man can now contemplate the ideas of space travel and of applying servo methods to the economy.

Has the power of man's thinking been increased; is man getting smarter because of automation in information handling? All here recognize that we do not even know what this question means. John von Neumann, from whose work and inspiration all of us profited greatly, was never able, even with his great powers of analysis and introspection, to arrive at a description of thinking which was satisfactory to him. Roger J. Williams, a pioneer worker in the "Science of Man," wrote in a recent article[1] "While I do not wish to oversimplify the functioning of the brain, I think that we can safely say that it has something to do with the thinking process." This statement was taken out of context with the hope that all will read the complete article. It is used to indicate how much has yet to be learned about thinking.

Even though lacking precision of definition, all would agree, perhaps, that terms such as problem solving and learning are connected with thinking and with intelligence. It is therefore not only desirable but essential that one considers how startled anyone would have been 10 years ago had one been told of some of the developments which are now active, which are in the forefront of our activities, and which seem possible only as a cumulative effect of 10 years of experience in computing. As examples, refer to the work of Simon and Newell in theorem proving by machine by nonalgorithmic methods and to the work of Friedberg in how to solve problems without giving the machine an explicit method.

It may be that people themselves are not smarter but at least it seems to me that the problems which people now propose to solve have scope and generality which would have been completely unimaginable 10 years ago. Indeed, it is the author's view that the power and extent of man's thinking has been increased because of his experience with automatic computing machines.

Automation has been beneficial to man working and to man thinking. The fears of 10 years ago are proved groundless.

Reference

1. Chemical Anthropology—An Open Door, **Roger J. Williams.** *American Scientist*, New Haven, Conn., vol. 46, no. 1, Mar. 1958, pp. 1–23.

Discussion

Harold D. Laswell: The important phrase in Dr. Hurd's extraordinarily valuable summary was, given appropriate foresight and planning, that there had thus far been no social problems, and I think that this is precisely where the emphasis needs to be put. It is of the utmost importance to make explicit the assumption about the context in which such instruments as we are talking about are used. In making these assumptions, we are talking about the conceptions of advantage and disadvantage that the various industries, firm leaders, owners, investors, and other decision makers of the complex social processes of the United States, and ultimately of other countries, will be involved in. It is, in all probability, quite beyond the possible scope of any one profession to control the social consequences of its specialized activity in society.

This is feasible in a limited scale during the initial period in which the new social innovation is taking place. As long as it is relatively puny in scope, it is possible for people of good will, for example, to control considerably the administrative devices which are made use of in an effort to anticipate the kind of difficulties that may arise.

I wish to emphasize this point because I think that it is important for professional groups, in this century particularly, to make use of all of the instruments that are at our disposal to anticipate the consequences of controlling part of the world balance of power, of controlling some industries faster than others. These are the interesting questions, the ones which can be affected by the advanced planning that was referred to by Dr. Hurd. I suspect that it is appropriate that, as this profession extends its scope and functions, the profession might interest itself even more than it has to date in undertaking to contemplate and foresee these consequences of social action.

With reference to the comments concerning scope and generality of thinking and what has already been done to increase man's thought effectiveness in these particulars, this is especially true in the areas of the social disciplines. It is already most obvious in the case of certain anticipations of population shifts. The effect is also quite apparent and significant in relationship to a variety of productivity.

In the other branches of the social disciplines, there has as yet been minimum social impact, and one of my hopes is that it will be more possible for us to develop decision-making processes that will make it appropriate for us to develop simulations of the major consequences of policy action, to which reference has been made.

Now then, I might conclude by saying that I welcome especially the very immediate and important emphasis which has been put on the problem of domesticating these new devices of production and of social thinking. With reference to the emphases which have been given by labor's representative, it is plain that we will want to consider the source and extent of policy referred to, in case industrial management is slower in waking up to the potentiality of the situation, and in case that management does not entertain the enlightened policies that Dr. Hurd has described in the case of inventory control. Also, we are confronted by other problems of social intervention. The question will come, "How far will we go, and through which channels, to supplement the failures of voluntary interpretation of self-interest that coincides with a social advantage?"

Two last remarks: As indicated previously, among the consequences that we might consider at one point is the effect on the psychosomatic integrity of people in our society, and also the significance for the skillful outcomes in society.

I think that perhaps one remark here is worthy of underlining for the purpose of discussion. It is often pointed out by individuals that it may be that some people in our society are ungifted, and it is quite clear that the ungifted people are problems; i.e., those whose predispositions we do not know how to utilize in ways we regard as socially contributory and that the individual accepts as satisfactory.

Now the one point I wish to make clear is who at any given time are called the ungifted. It is not necessarily the group that we will call the ungifted 10 years hence. The remarkable development is the result of modern science and technology, the expansion of methods of discerning capabilities to discover that 10, 20, or 30 years ago our notions about who were the ungifted children were singularly beside the point, and that a great many of the ungifted children were merely good in mathematics, and their teachers were not. I commend to ourselves the implications of this particular point, and suggest that we need to keep our minds open, as usual, to the possibility of discovering that those who are "ungifted" reflect only light from the darkness of our own current and scientific knowledge.

B. J. Schafer: Mr. Chairman, since I rather overstayed my leave awhile ago while on the podium, I desire not to impose on these good people too much. I think, however, one brief statement about the reference to enlightened management becomes a point that is extremely important to we, from the side of labor who constantly go to the bargaining table and have duplicate discussions of all that has happened here, where we discuss with the many hundreds of companies with which we bargain, our analysis of this thing called spreading technology and is effects on all of our society. I do not limit my statement to the effect on our people because we have learned that this problem is rather general in character, and in those discussions with management we are having almost the identical clash of ideas and ideals that have occurred here. I hope, however, that management might come to the bargaining table with some reservations about some of the things we say, again to see this broad social impact and take note of it, and recognize that there are things that have to be done.

I have had some personal experience; I have seen the displacement, and it is rather difficult when a man who is 50 years old finds himself without a job as a result of the introduction of some new machinery in, say, a pipe line which might have employed 2,000 people and extends from a refinery, say, in Houston, Tex., to, say, Harrisburg, Pa., and finds itself in a period of 6 years so completely automated that the force has been cut to a figure less than half of that which previously existed. Many of those people are reduced to returning to farms, and to doing other things. Those are some of the practical things that we around the bargaining table discuss, and those are some of the things that we sometimes find management reluctant to do anything about.

So I would hope that the same kind of understanding that we are arriving at here this morning in this broad discussion of the problem will become more general in character in all sections of our social structure. I would like to question my distinguished colleague a bit more. Is it correct that, "The trend to automation creates," as I understand it, in Dr. Laswell's analysis, "a trend to centralize?" I thought that might lead to other centralization trends. I believe he made reference to a centralization in government trends in planning. I would like to inquire as to what his thinking was with respect to that point. I could view it with alarm, or I could view it as something that might be good.

Harold D. Laswell: If I may reply, at the moment I am not viewing with alarm, nor am I viewing with enthusiasm. I was undertaking to characterize the profitable course of future development, suggesting that the instruments to which we are referring here are likely to be made use of on the basis of dominant purposes of the social context in which these instruments are being employed.

Now the dominant purposes characterizing a nation's status in a time of continuing mobilization and militarization are quite likely to continue in the direction of centralized direction of the allocation of the economic resources, for example, for purposes which are called national security. Under these circumstances the motivations are great in the direction of developing the instruments not only of central planning, not only certain kinds of central operational check-up control, but also to develop the instruments of surveyance. All of these mean that the government, the institution that is authorized to speak in the name of the whole community, comes to be regarded as the agency with the greatest responsibility for performing each of those operations in central fashion, and coordinating a variety of the resources of activities of the community to accomplish those purposes. What I am suggesting is that, in the years to come, in all probability, we will see the devices made use of to increase the degree of centralization of this character. This increases the degree of centralization of this character. This is all that I intended it to mean, but, needless to say, I welcome questions on this point.

B. J. Schafer: I want to ask the question because I, too, pointed to the fact that there is a certain trend toward centralization in thinking and control, but that we, in the trade union movement, customarily felt that too much centralized planning smacked of a totalitarian approach that we might not like. I think that other people have much the same concept, but I gather from

your statement that this is something to be thought of, and warned of, and that may be a national trend, and I can see where that is a possibility.

Cuthbert C. Hurd: I want to comment on one of Mr. Schafer's ideas which he repeated, and, therefore, I am sure he is attaching a great deal of importance to it. He said at one time that there were prophecies which might bring about improvement on the order of several hundred per cent and which might be incorporated almost overnight.

I match this with the experience which our industry has had with the military application of the guided missile. It is only now, in 1958, after the expenditure of billions of dollars and the application of some of the best brains in America, that the missiles can be controlled by computers.

Let us talk about process control with computers in maybe 10 or 12 years, compared with a field in which we still do not know whether computers are going to succeed, the field of application of data-processing machines in business. What is happening is this: The busiesses that started out first with manual methods then evolved to punch-card methods, finally are ready for large automatic machines. However, the period of time it takes to study and install and make a large automatic machine profitable, even after all of this evolution, is about 3 years. In our field I do not know of the possibilities for overnight improvements of several hundred per cent, which might bring into being the dislocation of the entire departments, and the disruption of activities. I do not know, and I guess that no one else does either.

Harold D. Laswell: I would very much like to hear a comment upon some of the implications that I referred to for the development of institutions to clarify policy alternatives. You may recall that I said something about the possibility of developing the social planetarium idea. Clearly, if one is to develop simulation models of the future allocation of population and resource, to characterize many of the future institutional processes in society, we will need the kind of instruments that will make it possibe to develop these forecasts well in advance. I would welcome comment on this type of fantasy as to whether it is a simple extension of what is already under way, or not.

H. T. Larson: This is one of the more sophisticated applications of the computers. Perhaps Dr. Hurd would be best able to give us a feel as to whether or not progress has been made in that direction.

Cuthbert C. Hurd: I suppose that the organization of such societies as Solametric and Psychometric, etc., which gives indication of fact, whose work is in the field of psychology and sociology, and where there is the possibility of making a major science of those fields, in itself is a very helpful sign. I think that the development of models, even in science, always takes much longer than one hopes. It seems to be that the development of models in social sciences takes a fairly large amount of time.

H. T. Larson: I expect that it will proceed very much as the physical sciences, and that models will be developed as a result of experiments and analysis. We have all read articles that have postulated economic models, but I do not know whether these have been used on computers.

Cuthbert C. Hurd: Rather recently there was an indication from Massachusetts Institute of Technology and also one from Harvard, that they saw the possibility of simulating the Canadian economy. The Canadian economy has been worked on, from the standpoint of simulation, for 20 years.

Which brings up a point. The computer industry has co-operated closely on automatic coding techniques for mathematical problems, and at the same time we realize that the simulation techniques on digital machines are becoming more and more important. I know of no concentrated, concerted, co-operative effort to develop automatic coding techniques to make simulation easy on a digital machine. I hope that the President of Share organization is here, and that he will start to work on this question.

We should all congratulate the Technical Program Committee for inviting these distinguished guests here today. I do not know when I have sat down and had someone get up and mention five or six ideas that had never occurred to me; and this has happened today. To mention only a few: The idea of using the computer as enlightenment in a political situation; and the expectation of violence and how it is related to computer development. These are startling ideas and completely new to me. Then tie this in with Mr. Schafer's comments that labor is not afraid of automation, what labor is asking for is enlightened management. Enlightened management, I take it, will consider the new ideas that Dr. Laswell has brought to us. We are only ten years old, and I look around here and I can see perhaps five or six of the small handful of people who started computing in this country 10 or 12 years ago. What are those fellows doing now? They are all managers. It is perfectly evident that not only they, but almost everyone in this room, increasingly, as automatic computing takes effect, are going to have positions of greater and greater responsibility, greater and greater managerial responsibility. You have been challenged today with new ideas, and you are the managers of the future in automation.

H. T. Larson: I think that that will serve very nicely, Dr. Hurd, as an up-beat note for the closing of our sessions.

I think that I speak for all of you, as well as for myself, in thanking Professor Laswell for giving us the benefit of his many studies over the years, and for applying his insight into the possible consequences of automation. Similarly I want to thank Mr. Schafer for so effectively stating the case for organized labor, and Dr. Hurd for so effectively speaking on behalf of the computer industry.

Logical Circuitry for Transistor Digital Computers

J. H. FELKER
NONMEMBER AIEE

MAN'S FIRST logical circuitry used gravity to supply the basic motive power in automatons of various complexities that were created from hydraulic devices[1] long before arithmetic was mechanized. Apparently these first automatons were used in mystic shrines and temples to impress the credulous. In these modern days our automatons are called giant brains and they are enshrined behind plate glass in temples of business efficiency. And so history repeats itself.

The logic circuits of a computing system are sometimes distinguished from the memory circuits by having the property of a fixed relation between output and input, whereas the relation in a memory is one that can be altered. Consider, however, a circuit assemblage with M input leads and N output leads. If the assembly is a logic circuit, the output is regarded as the logical consequence of the input signal. On the other hand, the assembly might be a memory circuit in which the input is an address and the output is the number stored at that address. In a permanent memory such as a ring translator[2] or a photographic plate scanned by optical means,[3] the relationship between input and output is fixed and the distinction between such a memory and a logic circuit is very obscure to say the least.

In common usage, an arrangement for which the relation between input and output can be compactly stated is called a logic circuit. An arrangement in which the relationship between input and output cannot conveniently be defined except as a tabulation of inputs and corresponding outputs, is usually called a permanent memory or a translator. Such distinctions are probably not very significant and in the talks which these remarks introduce, the term "logical circuitry" will be used not only to include arrangements in which the relationship between input and output is permanent, but also to include the bits and pieces of memory that are frequently associated with such circuits.

When the transistor was first announced, it seemed clear to some that the digital computer, which for a long time had been an idea in search of efficient mechanization, would be able to escape from the awkwardness of vacuum-tube circuitry into a more natural realization. The first transistors, however, were unruly devices. Even today most computing is being done with vacuum tubes. However, no one (to the author's knowledge) is planning new machines based on vacuum tubes.

In February of 1952[4] and December of 1954,[5] the author was given the opportunity to report to the Joint Computer Conference some of the work at Bell Telephone Laboratories on the application of point contact transistors to digital computers. It is a pleasure to again appear before a Joint Computer Conference and participate in this discussion of the alternate ways of using transistors in logical circuitry. The discussions will show how the speakers have exploited the junction transistor to produce sets of logical building blocks that are as stark, uncomplicated as, and far more reliable than, gear teeth meshing in gear teeth. The authors will present their material in a form that permits direct comparison of their proposals. At the end of the session it is the authors' hope that there will be participation by the audience in producing a group evaluation of the four sets of building blocks.

References

1. Le Monde Des Automates (book), **A. Chapuis, E. Gelis.** Paris, France, vol. 1, Chapt. 2, 1928.

2. No. 5. Crossbar AMA Translator, **T. L. Dimond.** *Bell Laboratories Record*, vol. 29, Feb. 1951, pp. 62–68.

3. Photographic Techniques for Information Storage, **G. W. King, G. W. Brown, L. N. Ridenour.** *Proceedings*, Institute of Radio Engineers, New York, N. Y., vol. 41, Oct. 1953, pp. 1421–28.

4. The Transistor as a Digital Computer Component, **J. H. Felker.** Special Publication S-44, Review of Electronic Digital Computers, *Joint AIEE-IRE Computer Conference*, Philadelphia, Pa., Feb. 1952, pp. 105–09.

5. Performance of TRADIC Transistor Digital Computer, **J. H. Felker.** *AIEE Proceedings of the Eastern Joint Computer Conference*, Philadelphia, Pa., Dec. 8–10, 1954, pp. 46–49.

J. H. Felker is with Bell Telephone Laboratories, Murray Hill, N. J.

Transistor Resistor Logic Circuits for Digital Data Systems

T. R. FINCH
NONMEMBER AIEE

THE DESIGNER of electronic circuits for digital systems frequently finds himself sandwiched between system objectives and requirements on the one side and the maze of component reliability and cost uncertainties on the other. Most of the time he feels generous when he says "We proceed with some logical use of art." This comes about because decisions based on pure judgment are sometimes forced by the necessity for proceeding ahead when analytical methods and definitive data, that would specify circuit choices and system worth, are not available.

Many creative circuit engineers are not too unhappy about this state of engineering understanding because it permits almost unlimited freedom to invent, analyze, and experiment with circuit designs. However, the assumed situation does lead to the following comment by this paper.

"The principal challenge in the field of digital system design is to provide systematic methods for over-all system synthesis: that is, to be able 1. to make precise statements on over-all system parameters, and 2. to generate system functions, system blocks, and circuit configurations that can be precisely and analytically related to the over-all system requirements."

T. R. Finch is with Bell Telephone Laboratories, Inc., Murray Hill, N. J.

In many cases (but not all by any means) the synthesis of new devices and circuits has been proliferous. The device and circuit choices are many for system environments ranging from missiles to business machines to industrial automation. However, the "truth tables" for the selection of components and circuit configurations are few and this situation is probably the primary reason for this group of invited papers on logic circuits. A stated objective, or perhaps hope, is that some rules for circuit selection will be generated through the presentation and comparison of common goals.

System and Circuit Objectives

This paper on Transistor Resistor Logic Circuits (TRL) for Digital Data Systems concerns itself with systems that face the circuit designer with the axioms of what Jean Felker calls the "Age of Complexity."[1] These are:

1. Face up to statistics. Pay attention to the tyranny of numbers.
2. Give the devil his due. Know your environment.
3. Insure bulkheads against spreading disaster. Provide protection for weak parts.
4. Use simple circuits. Design around simple components so that performance can be specified completely and anomalies that affect system performance do not exist.

Among the systems that demand design attention along the lines of axioms 1–4 is a data-processing system that has the following characteristics and which led to the TRL circuit proposed:

1. A large-scale system that contains:
   (a) Tens of thousands of active elements.
   (b) Hundreds of thousands of passive elements.
   (c) Perhaps millions of electrical connections: some soldered, some pressure contacts.
2. A large-scale system that demands several hours of assured performance in order to realize any useful output and requires hundreds of hours between failures for economic output.
3. A system in which economics predominate. Unreliability is a component of cost.
4. A system in which operational malfunction (sporadic error) is very costly. Downtime is an economic penalty but not a catastrophe as in real-time dominated machines.
5. A system in which increased circuit speed is valuable only if efficient machine use of speed can be realized in order to reduce the cost of data processing.

Table I. Characteristics of 2N393 Micro-alloy Transistor

1. D-c Current Gain
   D-c Beta >40, $I_C = 5MA$, $V_{ce} = 0.125V$
2. Speed of Response
   Rise time (T_0) $\leq 0.5\mu$ Second
   $I_{B1} = 0.125MA$, I_C $5MA$
   Storage Time (T_1) $\leq 0.6\mu$ Second
   $I_{B1} = 3MA$ $I_{B2} = 0.3MA$, $I_C = 5MA$
   Fall Time (T_2) $\leq 0.5\mu$ Second
   $I_{B1} = 3MA$, $I_{B2} = 0.3MA$, $I_C = 5MA$
3. Collector Leakage Current
   $I_C \leq 70\mu A$, $V_C = -5V$, $V_{BE} - 0.1V$

These system characteristics have led to the following circuit objectives:

1. The basic logic circuit when used in large numbers in many different machines of the system must provide the most economic system, including design charges, manufacturing costs, installation expense, and maintenance expense.
2. The circuit building block should provide, within the requirements imposed by the preceding item, flexibility for logical redesign and universality for general use.

Analysis of the required rate of data flow in the system and the resulting statistics of the necessary logical operations, i.e., the incidence of additions, comparisons, memory access cycles, etc., led to trial paper design for the System Control and Calculator in which approximately 7,500 logic circuits were used. A count of the number of inputs (fan-in) and the number of outputs (logical gain or fan-out) assumed for each logical building block indicated that 90% of the logic circuit needs could be met with a circuit that provided simultaneously a maximum fan-in of five and a maximum fan-out of five.

A characteristic of many data-processing systems is the large number of simple arithmetical operations that include memory read-write access. An economic balance is usually achieved in such systems through almost equal split of the total operational time between memory access and logic. For the size of random access store required in the large-scale system described, and considering the present development status of memory devices, an economic read-write cycle for a solid state store seems to be in the region of 6 to 10 microseconds. Allocation of a like time interval to the logical operations

Table II. Circuit Equations and Circuit Factors

1. On Condition
$$\frac{V_C}{\beta R_L} \leq \frac{V_C}{N_O R_L + R} - \frac{V_{be}(N_i - 1)}{R} + \frac{V_A - V_B}{R_A}$$
2. Off Condition
$$R_A(V_{ce} - V_{bo} + V_n) \leq \frac{R(V_A + V_{bo} - V_n)}{N_i}$$
3. Speed Condition
$$T = \frac{T_O}{4} + K_1 \left[\frac{\frac{V_C N_i}{N_O R_L + R} + \frac{V_A - V_{be}}{R_A}}{\frac{(V_{be} - V_{ce}) N_i}{R} - \frac{V_A - V_{be}}{R_A}} \right]$$

V_C = collector supply voltage
V_A = bias supply voltage
V_{be} = base turn on voltage
V_{ce} = maximum collector on voltage
V_{bo} = maximum base off voltage
R = gating resistor
R_L = collector supply resistor
R_A = bias supply resistor
N_O = fan-out
N_i = fan-in
β = transistor saturated current gain
T = average delay per stage
T_O = rise time at maximum gain
K_1 = proportionality constant relating $\frac{I_{B1}}{I_{B2}}$ to T_1 & T_2
V_n = noise margin

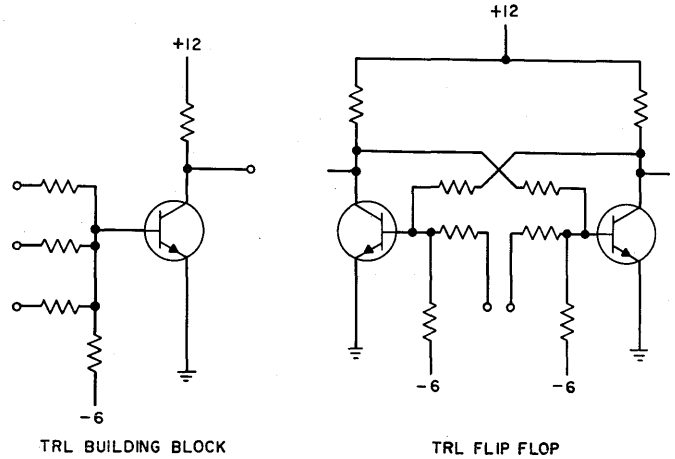
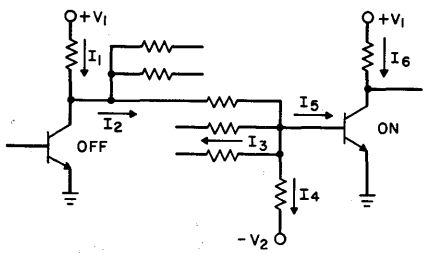


Fig. 1 (left). TRL logic building block and flip flop



$$\beta_{S\,MIN} \geq \frac{I_{6\,MAX}}{I_{5\,MIN}} = \left(\frac{I_{6\,MAX}}{I_{1\,MIN}}\right) \times \left(\frac{I_{1\,MIN}}{I_{2\,MIN}}\right) \times \left(\frac{I_{2\,MIN}}{I_{5\,MIN}}\right)$$

$$\beta_{S\,MIN} \cong \left(\text{CURRENT SOURCE FACTOR}\right) \times N_O \times \left(\frac{I_{B1} + I_{B2}}{I_{B1}}\right)$$

Fig. 2 (right). Typical interconnection of TRL circuits

Table III. Qualitative Comparison, DCTL, TRL, Symmetrical Logic

	TRL	DCTL	Symmetrical Current Logic
1. Logic capabilities (d-c transistor gain = 30)	Fan-out and fan-in from 5 to 10	Fan-out + fan-in ≃ 7	Fan-out and fan-in from 5 to 10
2. Delay per stage (30 mc transistor)	0.25–0.75μS	0.2–0.6μS	0.03–0.1μS
3. Noise consideration	Insensitive	Ground plane noise	Capacity and inductive coupled noise
4. Equipment realization problems	Flexible and easily manufacturable	Closely packed components for small ground plane	Low capacity wiring
5. Power drain	Low	Low	High
6. Initial transistor requirements	Moderate	Severe	Moderate
7. Mean time to failure	Long	Intermediate	Short

of a parallel organized calculator led to a stage propagation time of 0.5 microsecond maximum.

Summarizing: In addition to the previously listed circuit objectives, more specific circuit requirements developed are:

1. Allowable inputs per stage (fan-in) ≧ 5.
2. Allowable outputs per stage (fan-out) ≧ 5.
3. Maximum propagation time per stage ≦ 0.5 microsecond.

Circuit Selection and Description

The selection of logical circuits for a large-scale data processing system involves the consideration of many factors. The relative importance of these factors will vary depending upon the particular application. However, to some extent the following considerations generally apply: economy, reliability, flexibility, and designability.

In many cases, the operational requirements can be met using any one of several circuit forms. The circuit designer may be faced with a choice, at this point, between a relatively complex circuit form that permits a high ratio of circuit speed to device speed and thus provides an opportunity to use a wide range of suitable transistors or a relatively simple circuit form of few parts that can be completely analyzed but has a somewhat lower ratio of circuit speed to intrinsic device speed and permits less latitude in the selection of a transistor.

The decision to use the relatively simple TRL circuit to meet the system requirements resulted primarily from the following:

1. Rapid advances in semiconductor technology, primarily solid-state diffusion techniques, have provided very reliable high-speed transistors that can be uniformly produced at low cost. There are many indications that improvements in semiconductor technology will continue and that more-than-adequate circuit speed will be obtainable without resorting to "speed up" configurations.
2. System reliability depends, in large part, upon component life and operational margin statistics. Simple circuit configurations provide the system designer with a means for solving the "tyranny of numbers" problem.

The basic TRL logic circuit and flip-flop are shown in Fig. 1. Although n-p-n transistors are shown, either p-n-p or n-p-n devices can be used throughout. The building block circuit consists of an n-terminal resistor gate followed by a common emitter-inverting transistor amplifier.[2] The gate is an "or" gate to off-ground signals and an "and" gate to ground signals and by interconnecting building blocks of this single type, any combinational logic circuit can be implemented. As shown, the flip-flop consists of two building blocks in a re-entrant connection. This paper assumes that the reader is familiar with the logical design procedure for developing system units such as adders, registers, etc., from the basic building-block logic circuits.

The typical interconnection of several basic circuits is shown in Fig. 2, and a number of operational parameters of TRL are illustrated. The number of resistive legs on the input gate is referred to as the fan-in and the number of circuits which can be driven from the collector node specifies the logical gain or fan-out. An important parameter is the circuit delay, which determines the rate at which signals propagate through a logical network. The maximum propagation time determines the time which must be allowed between gating operations and, therefore, determines the system speed. For this reason, the maximum propagation or delay time has been considered as one of the system parameters.

The three TRL operational parameters described, fan-in, fan-out and, maximum-signal propagation time, are closely related. Within limits, performance exchanges can be made between them in

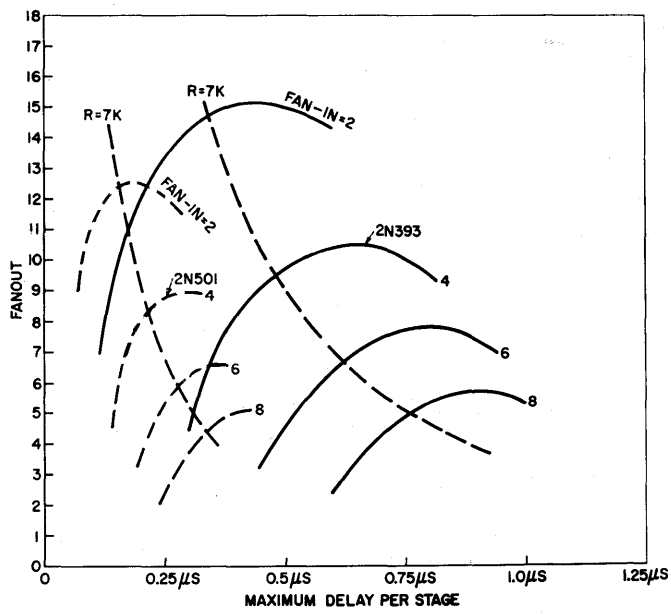


Fig. 3 (left). TRL design-capability curves using 2N393 transistor

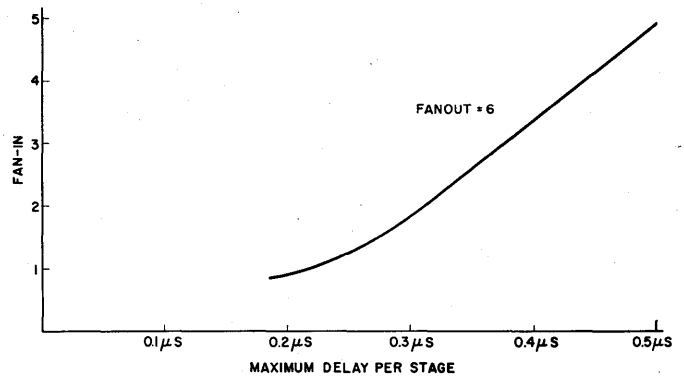


Fig. 4 (above). Maximum delay per TRL stage as a function of circuit fan-in. 2N393 transistor

Table IV. Estimated System Component Count and Economic and Reliability Factors

			TRL	DCTL
Components per system		Transistors	7,500	20,000
		Resistors	42,500	10,000
		Component connections	107,500	80,000
Failure rate including component connections (% failure/10^3 hours)	1958	Transistor	0.08	0.08
		Resistor	0.001	0.002
	1962	Transistor	0.02	0.02
		Resistor	0.001	0.002
Relative cost factors	1958	Transistor	45	45
		Resistor	1	0.33
		Component connections	0.13	0.13
	1962	Transistor	12.6	16.6
		Resistor	0.8	0.33
		Component connections	0.07	0.07
Relative first cost of components	1958		2.9	6.7
	1962		1	2.5
Mean time to failure-hours (component degradation) only	1958		156	62
	1962		520	238

order to achieve an optimum design for a particular application.[4] As shown in Fig. 2, the TRL transistor acts as a switch which, when conducting, shunts the current from the collector current source to ground. During nonconduction, this source current divides among the output paths and, consequently, the current supplied to any one coupling resistor is inversely proportional to the fan-out. Any one of the resistor gating paths of a TRL building block must provide sufficient current during the drive state to, 1, turn on the following transistor switch, 2, supply the base bias current, and 3, supply leakage current through the other fan-in resistors which may all be returned to ground through low voltage "on" transistor switches at the far end. Therefore, the higher the fan-in, the greater the leakage current and the smaller the current available for turning on the transistor switch. Thus, logic circuit gain (fan-out) can be increased at the expense of fan-in, within limits, and vice versa.

The principal propagation delay encountered in most TRL circuits results from minority carrier storage delay due to deep current saturation of the transistors. Conduction of reverse base current during the turn-off transient reduces both the storage time and the fall time of the transistor switch. The bias current sink and the leakage path back through the fan-in resistors provide reverse base drive. Thus, high reverse base current can be supplied, with resulting minimization of the signal propagation delay, at the expense of fan-in and fan-out.

An additional characteristic of importance, particularly when high reliability is of major interest, is the margin against false operation caused by inductively or capacitively coupled signals. This margin is provided in TRL by the resistive interstage network that provides, 1, a protective back bias on the base of an off transistor largely independent of transistor characteristics, and 2, a dissipative filter that protects against impulse noise.

Another parameter of interest is the operational temperature range of the circuit. Although this will not be considered in detail by this paper, because of assumed air conditioned environment, preliminary tests have indicated that TRL is relatively insensitive to a moderate temperature change. Suitable operation, meeting full logical requirements, has been achieved over a temperature range of 50 to 125 degrees Fahrenheit with germanium transistors.

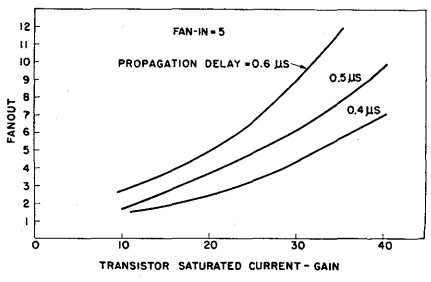


Fig. 5 (left). Effect of transistor current gain, beta in saturation, on TRL speed and logical capabilities

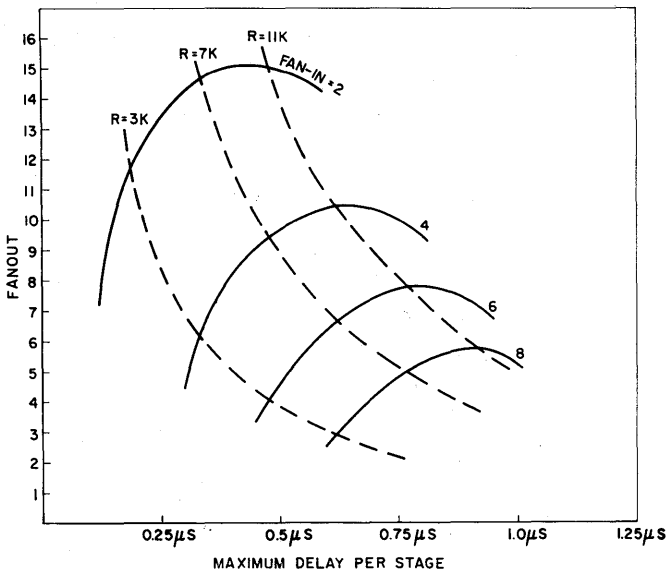


Fig. 6 (right). TRL design capability curves for 2N393 and 2N501

Circuit Design Tolerances

A question of considerable importance to the circuit designer is the extent to which worst conditions must be taken into account. For example, should end-of-life tolerances in the worst direction on all components be assumed to occur simultaneously without detrimental effects on the circuit operation? The answer to this question is not clear. However, it is clear that the larger the number of building block circuits to be assembled in a system, the more conservative the design must be. Or, as it was put earlier, design margins must face up to statistics. Accordingly, the design described here provides for proper circuit operation under the assumption of end-of-life conditions on all components. In particular, 5-% tolerances are assumed for the resistors, 4-% regulation is assumed for the power supplies and a 25-% reduction in transistor current gain, beta, is also assumed. The reader undoubtedly recognizes that worst-worst design and end-of-life condition on all components are severe performance limitations and may demand compromise in many cases. The following performance data, however, have assumed such restrictions.

TRL Performance

The design capabilities of a TRL building block circuit using a *2N393* germanium microalloy transistor are shown in Fig. 3. The essential transistor requirements are listed in Table I. TRL circuit analysis was undertaken using the three

equations shown in Table II, and the computation was accomplished using an IBM *650* computer.

The curves shown in Fig. 3 indicate optimal building-block circuit designs for particular requirements. It can be seen that, within limits, exchanges can be made between fan-in, fan-out, and speed. The dashed lines indicate the gating resistor values which optimize the designs. Corresponding to any particular point on the curve, design values of the other circuit components are also specified. The circuit flexibility for system parameter exchanges (fan-in, fan-out, and propagation speed) is particularly inviting and important to the circuit and system designers. The curves are calculated assuming operation at room temperature and a minimum margin against false turn-on of 150 millivolts. This is considered to be more than adequate in a normal environment and reasonably well designed equipment.

It can be seen from Fig. 3 that a building block design with a maximum fan-in of five and fan-out of six will have a maximum signal propagation time of 0.5 microsecond. Preliminary investigations indicate that such a design will be adequate to handle more than 90% of the building block applications.

It should be emphasized that the performance indicated here is under worst circuit operating conditions and assumes end-of-life component tolerances all in the worst direction. In particular, it is assumed that maximum fan-in and fan-out in all building blocks will be utilized. The maximum signal-propagation time of a building block is closely related to the fan-in of the circuit, since under worst circuit conditions all inputs may supply current to the base of an "on" transistor, saturating this transistor very heavily. The turn-off of such a stage will consequently be somewhat slower than the turn-off of a stage with fewer active inputs. Thus, if lower fan-in building blocks are used, or if there are known restraints on the input variables, somewhat faster circuit operation may be expected.

The relationship of signal propagation time and circuit fan-in for the particular design previously mentioned (coupling resistors 5,100 ohms, design optimized for fan-in of five and fan-out of six) is shown in Fig. 4. It can be seen from this figure that under actual operating conditions, considerably faster propagation speeds can be expected. This, in fact, has been observed in an experimental decimal adder using 150 TRL circuits of the previously mentioned design. Average propagation

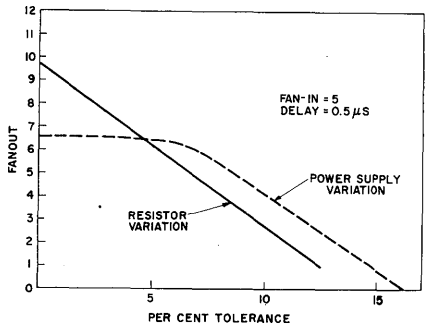


Fig. 7. Effect of power supply and resistor variations on TRL logical capabilities

delay of 0.2 microsecond per stage, or less, has been observed in this equipment. The fan-in ranged from one to five.

The relationship of the saturated current gain beta of the *2N393* transistor and the fan-out of a TRL circuit versus propagation delay, for a fixed fan-in of five, is shown in Fig. 5. From this data the effect of beta degradation on TRL performance can be determined.

The TRL design curves for the somewhat faster *2N501* microalloy diffused transistor are compared with those of the *2N393* in Fig. 6. It can be seen from Fig. 6, that a fan-out of 5, a fan-in of 5, and a signal propagation time of 0.2 microsecond can be achieved with the *2N501*. This probably can be realized by a direct replacement in the TRL circuit because it is seen that the logic resistor remains approximately the same value.

The effects of resistor and power supply tolerances on the design of TRL circuits are illustrated in Fig. 7. The use of a *2N393* transistor is assumed in a circuit with a fan-in of 5 and a signal propagation delay of 0.5 microsecond. The available fan-out is plotted as a function of the resistor tolerances in one case and power supply tolerances in the other, and assuming worst conditions.

Comparative Performance

The final choice between circuits for a particular application usually must be based on a cost comparison. Total cost is difficult to determine because a number of factors which enter into the final cost of the system are difficult to obtain. The initial cost of the system and the expense of the machine downtime due to maintenance and malfunction, are factors to be considered and the economic importance of each must be ascertained.

A rough qualitative comparison of three circuit approaches, TRL, direct-coupled transistor logic, (DCTL) and symmetrical current logic, is listed in Table III. The first cost of the components, wiring, and assembly can be determined quite readily. The evaluation of the cost of maintenance and repair, however, is somewhat more difficult. In order to evaluate this for the system described, the System Control and Calculator was designed using, on a trial basis, several different circuits. In order to illustrate the approach used, data will be presented for two designs: DCTL and TRL. These two circuits would normally be competitive where simplicity, low power, and economics predominate and high speed may be of secondary importance. The estimated component counts using the two circuits are listed in Table IV. It should be noted that in the DCTL design a much larger percentage of the building block capabilities in terms of fan-in and fan-out was utilized as compared with the TRL design. This is shown by the relatively low transistor ratio of less than three to one. It is expected that increased logical design sophistication might decrease the number of TRL transistors. Thus, the TRL design affords greater flexibility for redesign, and improved economy and reliability may be achieved through more efficient use of the TRL building block characteristics.

Table IV also lists the estimated 1958 and 1962 figures for the relative cost factors and failure rate of the various components used in the analysis. The TRL resistor is taken as the reference with a cost weighting of unity. The relative 1962 DCTL transistor price factor is assumed slightly higher than that for the corresponding device used in TRL due to the more extensive transistor testing and selection required for the DCTL application. In fact, the tight requirements on the transistor will probably result in decreased interest in DCTL for future large-scale systems. The assumed relative price factor for the TRL resistors is predicated on the use of molded deposited carbon resistors with an end-of-life tolerance of 5%. Based on these estimates, the relative initial-system price factors and the anticipated failure rates for the DCTL and TRL System Control and Calculator units have been determined and are shown in Table IV. The mean-time-to-failure computation is based on estimated component degradation only. The TRL design is shown to be more economic and to provide a longer mean time to failure using both the 1958 and 1962 data. These results are due to the use of the resistor, an economical and reliable component, as the gating element in the TRL circuit rather than the transis-

tor as in DCTL. Another very important advantage of the TRL circuit over DCTL is the greater margin afforded by the resistor interstage network against false operation caused by extraneous signals in the ground connections. This improved margin not only implies greater freedom from sporadic errors but also permits greater flexibility in the design of the physical equipment.

Direct comparison has not been made with the other logic circuit configurations, such as nonsaturating current steering circuits, because, in general, systems employing these circuits use more components to meet high-speed objectives, which are not the dominant requirement of the system described. Hence this paper contends that systems employing such circuits are more expensive and perhaps, because of higher component count, are less reliable. Diode logic seems less attractive than TRL because diodes are not competitive with resistors as gating elements, either in terms of cost or in terms of reliability.

Summary

The final chapter on reliability and economics in large-scale data processing systems may not be written for several years. However, the planning and design choices leading toward extensive electronic mechanization are now under way. This paper contends that the logical use of logic devices is leading toward implementation that employs simple circuits. That is, circuit simplicity extracts the best of the past developments on reliable, low-cost components, such as stable d-c power sources and resistors, and couples these with the best of modern semiconductor technology. Such a choice, for the medium-speed data processing system here assumed, is the transistor resistor logic circuit.

A large part of the material presented in this paper represents the work of the writer's colleagues, Mr. E. G. Rupprecht and Mr. Q. W. Simkins. It is hoped that a more extensive and detailed analysis of TRL will be published at a later date.

References

1. Complexity with Reliability, **J. H. Felker.** *Student Quarterly*, Institute of Radio Engineers, New York, N. Y., Sept. 1, 1956.

2. Transistor NOR Circuit Design, **G. H. Royer, W. D. Rowe.** AIEE Transactions, vol. 76, pt. 1, July 1957, pp. 263–67.

3. Arithmetic Operations in Digital Computers (book), **R. K. Richards.** D. Van Nostrand Company, Inc., Princeton, N. J., 1955.

4. Transistor Resistor Logic Circuit Analysis, **Q. W. Simkins.** *Proceedings*, Solid State Circuits Conference, Philadelphia, Penna., Feb. 20, 1958. (To be published.)

Direct-Coupled Logic Circuitry

JAMES B. ANGELL
MEMBER AIEE

DIRECT-COUPLED transistor logic circuitry (DCTL), first announced in 1955, has provided many desirable features for digital computer use.[1] Among the principal characteristics of DCTL are extremely simple circuitry, a minimum number of auxiliary components, very low dissipation and power-supply requirements, and the possibility of extraordinary compactness. Various special-purpose machines using DCTL have been built, and at least one class of general-purpose computer has been designed and is in operation. In this paper, the unique features and limitations of DCTL will be considered in detail, together with a brief description of the circuitry and its principles of operation.

Circuit Description

DCTL circuitry takes advantage of the fact that certain types of transistors have useful current gains and impedance ratios with the collector junction slightly forward biased (in saturation), whereas the usual mode of operation calls for a reverse bias on the collector. DCTL operation is unique to transistors; a vacuum-tube equivalent would require useful operation with the control grid more positive than the plate.

Given the possibility of a transistor running with zero or slightly forward collector bias, the basic inverter circuit shown in Fig. 1 becomes possible. This figure shows three cascaded common-emitter inverters, using p-n-p transistors. If the first stage is conducting (in saturation), its collector-to-emitter voltage is sufficiently low that the second stage is held off (nonconducting). With the second transistor not conducting, a sufficiently large current is drawn from the base of the third stage, through the second-stage load resistor, to put the third stage in saturation.

In order to appreciate the possibility of this modus operandi, consider the transistor characteristic curves shown in Fig. 2. These curves are representative of most "wafer" transistors, including alloy-junction, surface-barrier, and similar types, which have negligible series resistance in the emitter and collector leads. On the left of Fig. 2 is shown a set of collector characteristics for the common-emitter connection. These plots of collector current versus collector-to-emitter voltage, with base current the fixed input parameter, show that the transistor maintains good current gain and output impedance with collector voltages of 0.15 volt or less; such curves are representative of both silicon and germanium wafer transistors. The common-emitter input characteristics, shown on the right, indicate that the threshold of significant conduction occurs only with forward biases (negative for p-n-p transistors) greater than 0.1 volt for germanium or 0.6 volt for silicon. Thus it is obvious that the collector-to-emitter voltage of a conducting transistor can be maintained at a lower value than the threshold required for conduction at the input of a succeeding transistor.

Given a DCTL inverter, it is simple to make a flip-flop by connecting two inverters together, as shown in Fig. 3. The mechanism by which the gates affect the flip-flop is as follows: When a gate conducts, its collector-to-emitter voltage becomes low, thus turning off the flip-flop transistor whose base is connected to that gate, in turn causing the other flip-flop transistor (the one whose collector is tied to the gate in conduction) to conduct. Quite obviously, more than one gating circuit could be connected in parallel to either side of the flip-flop, so that the activation of any one or more of the gates on one side would set the flip-flop.

James B. Angell is with Philco Corporation, Philadelphia, Pa.

Many organizations, particularly Philco Corporation, Bell Telephone Laboratories, and Burroughs Corporation, have contributed to the understanding and perfection of direct-coupled transistor logic circuitry. The author is especially grateful to the many colleagues at Philco Corporation and to E. G. Clark and H. J. Tate of Burroughs Corporation, who assisted in the preparation of this paper.

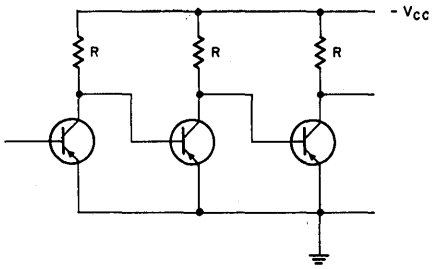


Fig. 1. DCTL inverters

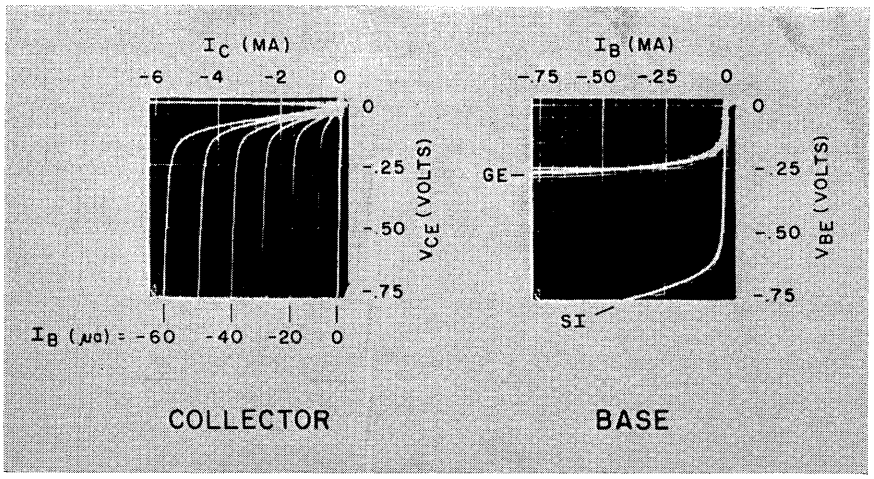


Fig. 2 (right). Transistor common-emitter characteristics

Gating functions can be achieved with transistors either connected in parallel or stacked in series. Examples of parallel and series gates are shown in Figs. 4 and 5. If the transistors are normally in the nonconducting state, the parallel gate could be called an "or" gate, since the output circuit conducts if one or more of the inputs are activated; similarly, the series gate is an "and" gate, since its output conducts only if all the inputs are simultaneously activated. An example of the simultaneous use of both gates to form a half-adder is shown in Fig. 6. The complementary inputs are normally derived from the two sides of flip-flops storing the appropriate input characters. It should be noted that an output signal in the sum or carry channel is the complement of the usually desired signal, in that a "one" corresponds to the low-impedance (conducting), or low-voltage condition. Therefore, it may be necessary in some cases to use an inverter to obtain the appropriate sense for the output signal.

It can be seen that, in general, fewer resistors than transistors are employed in these circuits, and that there is relatively little need for capacitors. One circuit in which a capacitor is required is the one-shot multivibrator, shown in Fig. 7. This circuit is used to generate delays for timing signals, and is triggered by the gating transistor shown in the figure, as suggested by the waveforms sketched there.

Typical practice shows that an over-all system uses roughly 400 resistors per 1,000 transistors and less than 10 capacitors per 1,000 transistors.

Transistor Gain

DCTL imposes a rather severe limitation on the amount of useful gain which can be demanded of each active element. This limited gain results from the need for operating a conducting transistor with low collector-to-emitter voltage. In practice, the number of loads (transistor bases) which can be driven from a single inverter (fan-out) is normally limited to 2 to 3 units in the case of surface-barrier transistors, or perhaps up to six loads in the case of higher gain alloy or microalloy units. The number of gates which can be used to control a single load (fan-in) is somewhat higher, usually set at 1.5 to 2 times the fan-out capability.

In order to understand how these limits are reached, it is desirable to consider the tolerances required for operation of a given interstage. Such an interstage, with n paralleled "or" gates driving m paralleled loads, is shown in Fig. 8. Two conditions are necessary in this interstage. First, when all the gates are off, it is necessary to ensure that all the loads are on. Secondly, it is necessary to guarantee that conduction in any one of the gates will turn off all the loads. The first of these two conditions can be assured by the inequality shown in equation 1.

$$\frac{|V_{CC}(\min) - V_{BE}(\text{on})|}{R_{\text{MAX}}} > nI_C(\text{off}) + mI_B(\text{on}) \qquad (1)$$

In this relationship, the term on the left, the smallest possible load current, is made greater than the leakage current in all the gates plus the minimum required load current in all the loads. The node voltage $V_{BE}(\text{on})$, is a fixed voltage limit, which must appear both in the transistor specifications and the circuit design equations; since it is a limit voltage it requires no tolerances. The second condition is indicated by equation 2,

$$I_C(\text{on}) > \frac{V_{CC}(\max) - V_B(\text{off})}{R_{\min}} \qquad (2)$$

which shows that the collector current of one conducting gate must be greater than the maximum possible current in the load resistor, with the node voltage equal to the specific limit voltage required to maintain the load transistors in the nonconducting state. The important transistor factors required to meet these conditions are as follows:

1. The input impedance of a conducting transistor must not be too low. This fact is assured by setting a maximum limit on "on" base current $[I_B(\text{on})]$ for a particular base voltage, $[V_{BE}(\text{on})]$. This limit is necessary in order to ensure that no one transistor will take too much of the available driving current.

2. The output impedance of a conducting transistor must not be too high. One method of ensuring this condition is to specify that the collector-to-emitter voltage $[V_{CE}(\text{on})]$ be less than a certain maximum value for given collector current and base voltage. An alternative method would be to put a minimum limit on collector current for given base-to-emitter and collector-to-emitter voltages. The ratio of the collector current in either of these alternative tests to the base current in the first test is the minimum transistor gain, which is normally substantially larger than the circuit gain. This ratio, $I_C(\text{on})/I_B(\text{on})$, is usually much smaller than the h_{FE} (current gain) of the transistor in the nonsaturated region ($|V_{CE}|>0.3$ volt).

3. The "off" collector current in a nonconducting transistor must not be too large. This condition is met by putting a maximum limit on collector current $[I_C(\text{off})]$ for a specified base voltage $[V_{BE}(\text{off})]$.

A representative specification for a surface barrier transistor intended for DCTL operation is shown in Table I. The first three quantities in this table ensure that the transistor will function in DCTL circuitry. The last three quantities relate to the high-speed performance of the transistor. Other transistor types have been recommended for use in DCTL operation. The *2N240* surface barrier transistor is specified for switching service, and meets all the requirements for DCTL operation; however, the specification does not guarantee the input impedance. A silicon-alloy transistor, the *2N496*, has also proven satisfactory for DCTL operation, here again, the specified switching

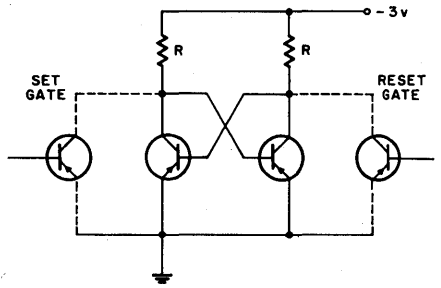


Fig. 3. DCTL flip-flop

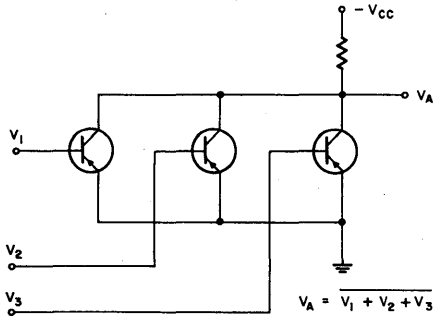


Fig. 4. Parallel (or) gate

characteristics include all the quantities for DCTL operation other than the input impedance.

In order to have compatible connections from one DCTL logical element to another, $V_{CE}(\text{on}) \leqq V_{BE}(\text{off})$. In the case of an "and" gate of two or more transistors, it is necessary to sacrifice gain to reduce the collector-to-emitter voltage, in order that the sum of all these voltages in the series gate will not exceed the "off" base-to-emitter voltage of the loads.

It is possible to relate the transistor gain to component tolerances and circuit gain quite explicity in the case of DCTL. For the single-level interstage shown in Fig. 7, this relation is easily achieved from equations 1 and 2. By solving these two inequalities for the ratio of collector current to base current in conduction, one gets

$$\frac{I_C(\text{on})}{I_B(\text{on})} > m\,\frac{R_{\max}}{R_{\min}} \times \frac{|V_{CC}(\max) - V_B(\text{off})|}{|V_{CC}(\min) - V_B(\text{on})| - nR_{\max}I_C(\text{off})} \qquad (3)$$

This result shows that the transistor gain must be greater than the product of the circuit gain, m, and factors accounting for the tolerances on components. Similar expressions are derivable for more complicated interstages (such as those involving "and" gates). In order to include the effect of component tolerances and "off" leakage current, together with

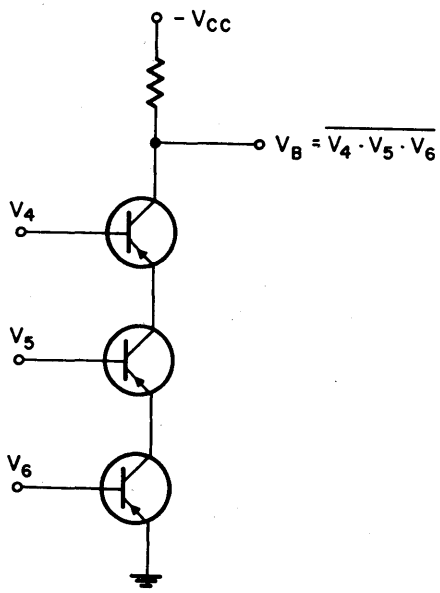


Fig. 5. Series (and) gate

any aging effects in the transistor, it is customary practice to set $I_C(\text{on})/I_B(\text{on}) > 2m$.

Propagation Time

Two transistor parameters are of principal importance in determining the switching time of DCTL circuitry. First, the transit time, of which the reciprocal of the α-cutoff frequency is a good indication, affects the rise and fall time of a gate. Secondly, hole storage is important, because transistors are driven hard into saturation in DCTL circuitry and show appreciable hole-storage delay time when being turned off.

Switching speed is noticably affected by the extremely low collector voltage at which all the transistors in a DCTL system normally operate, since the α-cutoff frequency decreases with decreasing collector voltage. For a typical surface-barrier transistor, as defined in Table I, gate switching times are of the order of 0.03 to 0.1 microsecond (μsec). Using such transistors in a flip-flop, transition times of the order shown in Fig. 9 are achieved.[2] It is interesting that the switching time decreases with increasing load; this decrease arises because the additional loads reduce the base current, and therefore the degree of saturation, of the flip-flop transitors.

With germanium alloy transistors having 5-volt α-cutoff frequencies of the order of 10 megacycles (mc), switching times of from 0.5 to 2 μsec are experienced. These switching times are generally greater than would be expected, when comparing α-cutoff frequency with

Table I. L-5132 Characteristics

$I_B(\text{on})$ <0.4 milliampere, (ma)	(I_C=2.5 ma, V_{BE}=0.293 volt)
$V_{CE}(\text{on})$ <0.075 volt	(I_C=2.5 ma, V_{BE}=0.293 volt)
$I_C(\text{off})$ <85 microamperes	(V_{CE}=−3 volts, V_{BE}=−0.1 volt)
I_{CO} <3.0 microamperes	(V_{CB}=−5 volts)
$f_{\max}$ >30 megacycles.	(V_{CB}=3 volts, I_C=−0.5 ma)
C_c <6.0 micromicrofarads	(V_{CB}=−3 volts)
$rb'C_c$ <1,500 micromicroseconds	(V_{CE}=−3 volts, I_C=−0.5 ma)

the corresponding quantities of surface-barrier transistors, because of the low-voltage operation in DCTL; this fact emphasizes the importance of the effect of low-voltage operation on switching speed.

The switching times observed in DCTL circuits are perhaps 1.5 to 2 times as great as are observed with the same transistors used in conventional resistance-capacitance circuitry having the same gain. However, since higher gains are normally employed with R-C circuitry the speed differential is not as great as 2 to 1. Of course, more refined high-speed circuits, employing nonsaturating or emitter-follower techniques, are appreciably faster, often by a factor of three or more.[2]

Reliability

There are many factors influencing the reliability of switching circuits. These factors include both internal effects, such as variations in component parameters and in power supplies with age or temperature, and externally generated effects, caused by induced signals (noise) from other circuits or stray fields. Contributing to good reliability in DCTL circuitry are the very-low power dissipation per logical element and the use of a single, uncritical power supply. Mitigating against good reliability are the very-low voltage levels employed within the system and the appreciable dependence on temperature of the "off" current of individual active elements.

Component Tolerances

As suggested by equation 3, it is possible to trade stage gain for component tolerances or for tolerances on supply potential. In general, it has been found desirable to use stable resistors with one-percent tolerances, and design circuits on the basis of at least 3 to 5% variation in component values. With regard to the transistors, in most cases limit (rather than bracket) tolerances are all that is required. For example, if the gain is greater than a certain minimum, or if switching speed is above a certain minimum, no trouble is encountered. The

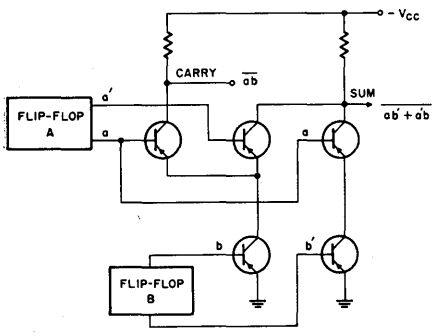


Fig. 6. DCTL half adder

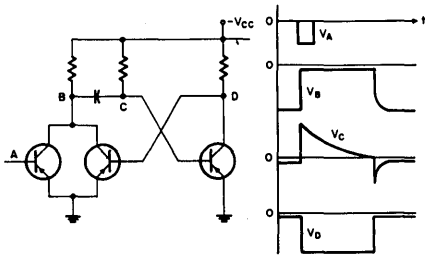


Fig. 7. One-shot multivibrator

one exception is the base-to-emitter "on" voltage for a given degree of conduction. Upper and lower bounds are required here in order to avoid hogging of the available current by one transistor in a multiple load. Fortunately, the input voltage for a given input current in a transistor is not sensitive to aging; in fact, the input current for a given voltage (a more severe condition) does not seem to vary significantly with age. This constancy of input voltage with age is not purely coincidental. It is to be expected that the input impedance in conduction is constant, because it is governed primarily by base spreading resistance (r_b') and emitter saturation current, I_{ES} (not I_{EO}, which includes a time-varying surface leakage current), both of which depend only on geometry and semiconductor resistivity.

DCTL has one particular advantage regarding life which should be noted. The extremely low dissipation in this circuitry subjects the transistors, and resistors, to a minimum of damaging stress from either voltage or temperature rise. Because of this extremely conservative operation, transistors have demonstrated excellent reliability in operating DCTL systems. Experience in various systems has shown failure rates (system malfunction) of less than one failure per 10^7 transistor hours. With such performance, it is possible to design systems with 10,000 transistors and still experience 1,000 hours of useful operation between malfunctions.

Extensive life tests on DCTL transistors have shown that two parameters are mainly responsible for the majority of deterioration with age. First is the "off" leakage current, which tends to increase with time. Secondly, the "on" collector voltage [V_{CE}(on)] for given base and collector currents tends to increase with time (because of the drop in current gain, h_{FE}). Both of these deteriorations are thought to be associated with contamination of the semiconductor surface near the active transistor.

Noise

Some kinds of noise are comparatively unimportant in DCTL whereas others are potentially severe. Because of the very-low impedance level of "on" circuits, capacitive pickup is generally negligible. Power-supply fluctuations are usually unimportant, because voltage translation circuits are not required and the majority of a complete logical system can, if desired, be run from a single supply. On the other hand, because of the small voltage swings which are experienced in DCTL, inductive coupling in either multiconductor cable or via ground leads may be troublesome.

It is generally felt that with germanium transistors a maximum induced noise of the order of 25 millivolts can be tolerated; this figure amounts to 10% of the normally encountered voltage swings

Many alternatives have been suggested for keeping the magnitude of inductive or ground-lead coupling within bounds. In the layout of a single printed-wiring board, it is frequently possible to use separate ground leads for susceptible circuits, or to use wide ground planes for common circuits involving both high-current pulses and susceptible gates. For connections between boards or between parts of a system, various techniques have been evolved. One interesting possibility for this application is the use of load (controlled) circuits which are normally "on" and are turned off by application of an incoming signal; with such operation, a short noise pulse will be ignored by the combination of the low-impedance load and by the hole-storage delay time in the load. A second technique for interplane coupling involves the use of higher voltage swings, and a definite "off" bias, afforded by R-C coupling. Fortunately, either of these techniques adds little complication to an over-all system, because the number of interplane connections can be minimized by designing large printed-wiring boards. In many cases, more than 200 transistors can be mounted on each board, a feat made possible by the small number of auxiliary circuit components.

Finally, contact noise of plug-in connectors is important in DCTL because of the low impedance level that is used in the active circuitry. Here again, large boards with many active elements per board have proven desirable in minimizing this trouble.

Temperature Dependence

Temperature dependence of transistor parameters creates one of the severe limitations of DCTL circuitry. The principal offender is the "off" current, I_C(off), for a fixed base voltage. This current varies exponentially with temperature, as does I_{CO}, and changes by as much as 8% per degree Centigrade, (C). Because of this severe temperature dependence, most DCTL systems using germanium transistors are not designed to operate above 35 to 40 degrees C, although with considerable sacrifice of gain it is probably possible to push this upper temperature limit to 50 degrees C. The upper limit with silicon transistors is in the range of 90 to 125 degrees C. Fortunately, because of the very low power dissipation, modest cooling equipment suffices to keep the system environment within necessary limits in cases where the maximum ambient temperature is excessive.

A second, far less important, effect of temperature is that switching speed tends to decrease with increasing temperature. This decrease is attributable to two phenomena, the increases of r_b' and

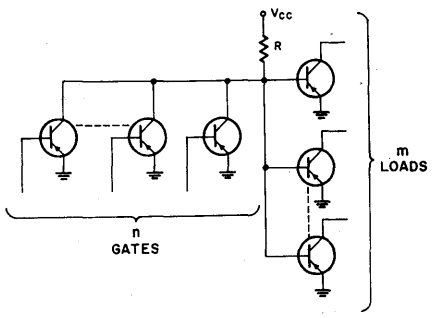
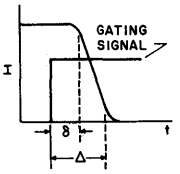


Fig. 8. DCTL interstage



SYMMETRICAL BASE-LOADS

	0	1	2	3	4
δ	75	55	45	40	30
Δ	110	90	70	65	55

SWITCHING TIMES IN MILLIMICROSECONDS

Fig. 9. DCTL flip-flop switching times

of h_{FE} with increasing temperature. The base spreading resistance, r_b', limits the flow of base current during turn-off, while the increase in h_{FE} increases the degree of saturation.

While the upper temperature limit is fairly definite and restrictive, germanium DCTL circuits can be cooled considerably below room temperature without undesirable effect. Although the transistor current gain, h_{FE}, decreases with decreasing temperature, the "off" base voltage, V_B(off), for a given collector current I_C(off), increases; the effects of these two changes tend to compensate each other.

Silicon-transistor DCTL circuitry behaves somewhat differently with temperature than does germanium circuitry, principally because of the lower current gain and higher base voltage of silicon transistors. Because current gain increases with temperature, it is found that silicon DCTL circuitry has wider margins of operation at high temperature than at room temperature. An example of this fact, based on data taken in 1956, is shown in the "schmoo" diagram of Fig. 10. This diagram contains plots of the upper and lower limits of "handle" voltage over which flip-flop operation is achieved.[3] The handle voltage of the DCTL flip-flop is the supply voltage to one side of the flip-flop, with the other side fixed at 3 volts. The particular test circuit on which these data were based employed deliberately selected extreme transistors, one with high gain and one with low gain. It can be seen that the margins of operation are much less critical (broader) at high temperature.

Design Predictability

Most of the important factors in DCTL circuitry are highly predictable, because of the extreme simplicity of the circuit modules. The transistor static parameters are, for the most part, in very close agreement with transistor theory, such as that propounded by Ebers and Moll. The relation between transistor characteristics and circuit performance is also predictable in many cases; an example of this predictability is given by equations 1, 2, and 3.

The prediction of switching times is again relatively simple, except for the fact that transistor parameters such as the α-cutoff frequency vary rapidly with operating point near the saturation region.

The two areas in static design where predictability is difficult are the stacked "and" gates and circuits operating at rather high currents. In the latter case, transistor nonlinearities become appreciable. This nonlinear region may begin at current levels as low as a few milliamperes for some extremely small transistors, particularly surface-barrier types.

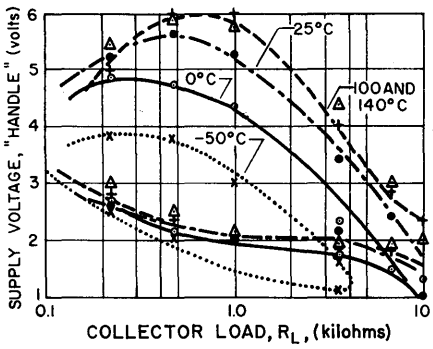


Fig. 10. Operational boundary (schmoo) diagram of silicon DCTL flip-flop

Marginal Testing

The possibilities for good marginal testing procedures are limited with DCTL, because there is generally only one power-supply potential. The usual procedure has been to split this supply into two sections, one supplying each side of all the flip-flops in the logical circuitry. The disadvantage of this procedure is that it renders the circuits somewhat more susceptible to power supply noise, since differentials can exist in the voltages supplied to the two halves of a flip-flop. Nevertheless, since there is little else that can conveniently be varied, this technique is still being employed for marginal checking.

Physical Properties

DCTL probably represents close to the ultimate of circuit simplicity and minimization, within the bounds of presently available diode or triode logical elements. (Such complex logical elements as the 4-terminal full adder recently described by R. F. Rutz[5] will no doubt make for even greater circuit simplicity at such times as they become producibly practical.) In general, only one class of transistor and one or two values of resistor are required for a complete logical system. Because of the small total number of components, the number of interconnecting nodes is likewise kept to a minimum.

The over-all cost of a DCTL system is very comparable with that of an equivalent system employing alternative circuit techniques, at least in the case of special-purpose computers. Admittedly, the cost per component is high, because most of the components are transistors and not inexpensive resistors, and because the transistors have to meet fairly stringent specifications regarding low-voltage current gain and impedances. However, because of the circuit simplicity and ease of design, engineering and assembly labor is minimized. Therefore, even though the component cost is high, the over-all cost in the manufacture of a limited number of special-purpose machines is fully competitive with that of a comparable design using alternative circuitry. One might say that DCTL eliminates a large fraction of the nodes of a logic system in exchange for a comparatively large number of fairly tight transistor specifications.

The logical design of a DCTL system is very close to the final electrical layout, because each logical element is normally equivalent to a very simple combination of transistors and at most two resistors. In fact, DCTL circuitry is very much like relay logic circuitry, with the exception that the individual elements are single-pole, single-throw switches with an interconnection between input and output circuits. Nevertheless, within the framework of specific rules of logical design, which are more restricted than in the case of relays, the conversion from design to finished hardware is very straightforward.

System Features

DCTL systems are probably more adaptable to compact packaging than any other competitive system, because of the small number of components and the lower power consumption of this circuitry. Although the circuitry demands rigid transistor requirements, it treats the transistors gently, so as to minimize the rate of aging. Probably the principal areas in which DCTL systems are presently applicable are those in which space and power consumption are at a premium.

The application of this class of circuitry will be greatly extended when economically competitive silicon transistors become available, both because of the higher maximum temperature and, more particularly, the easing of logical design rules made possible by the higher base-to-emitter potential in conducting silicon transistors. Various transistor types have already been specified for DCTL, thus eliminating any problem of component availability.

Summary

Direct-coupled transistor logic circuits are distinguished from conventional

switching circuits by a substantially smaller number of components and connections, and by extremely-low power consumption. Circuit simplicity and low dissipation are obtained at the price of limited gain, small voltage swings, and a comparatively low upper limit on internal temperature. Rather severe requirements on transistor parameters, particularly input impedance and saturation voltage, are compensated by almost negligible dissipation and maximum voltage requirements. The total cost using DCTL is comparable with other techniques, because the tightly specified transistor eliminates considerable complexity in system design and manufacture.

References

1. Surface-Barrier Transistor Switching Circuits, **R. H. Beter, W. E. Bradley, R. B. Brown, M. Rubinoff.** *Convention Record*, Institute of Radio Engineers, New York, N. Y., pt. IV, 1955, p. 139.

2. High Speed Transistor Computer Circuits, **S. Y. Wong, A. K. Rapp.** *IRE-AIEE Transistor Circuits Conference*, Phila., Pa., Feb. 1956. (Not published.)

3. High-Temperature Silicon-Transistor Computer Circuits, **James B. Angell.** Proceedings of the Eastern Joint Computer Conference, *AIEE Special Publication T-92*, Dec. 10–12, 1956, pp. 54–57.

4. Large-Signal Behavior of Junction Transistors, **J. J. Ebers, J. L. Moll.** *Proceedings*, Institute of Radio Engineers, New York, N. Y., vol. 42, Dec. 1954, pp. 1761–72.

5. Two-Collector Transistor for Binary Full Adder, **R. F. Rutz.** *IBM Journal of Research and Development*, New York, N. Y., vol. 1, July 1957, pp. 212–22.

Symmetrical Transistor Logic

R. H. BAKER
NONMEMBER AIEE

THIS paper discusses symmetrical transistor switching circuit techniques. The circuitry is designated "symmetrical circuits" because the basic circuits employ both p-n-p and n-p-n transistors in approximately equal numbers which result in networks that exhibit a high degree of operational and topological symmetry. The major practical advantage of this circuitry over other types lies in the ease with which basic circuits may be integrated into a digital system. This advantage is made possible through, 1, keeping the circuit configurations flexible (to satisfy many different system applications), 2, worst-case circuit design with regard to component stability and system loading requirements, 3, use of only standard commercially available components, and 4, insisting that all present-day circuit designs be applicable to known future device trends. The major criticism of the symmetrical circuit techniques is that the circuits require a greater number of components for a specific equipment than do other types of circuitry that can be engineered to build the specific system. However, it is felt that this criticism is not justified in this case for two reasons. They are: 1, The additional cost of components required for symmetrical logic is more than counterbalanced by the shorter system realization time required. This is particularly true for an establishment, such as the Massachusetts Institute of Technology Lincoln Laboratory, that is engaged in system research and development. And, 2, that symmetrical logic techniques are more adaptable to future developments in the device, circuits and system area than other types of circuit logic.

It is the purpose of this paper to show why, when, and under what conditions symmetrical circuit techniques should be used, along with probable future developments, through the discussion of the circuit techniques and relating their circuit capabilities to the solution of system problems.

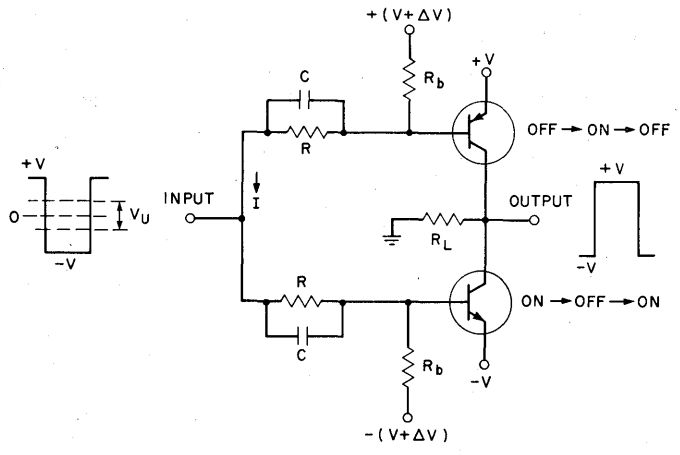


Fig. 1 (right). Symmetrical buffer inverter

I. Basic System Requirements

A. Initial

The initial specific system for which the symmetrical circuit approach was intended had the following requirements:

1. General-purpose computer applicable to real time problem solution.
2. All solid-state machine.
3. Nonairconditioned ground environment.
4. Synchronous machine with a basic clock rate of 3 microseconds, (μs).
5. Memory capacity of 8,192 registers, 28 bits in length, with a memory cycle time of 6 μs.
6. Semiportable housing.
7. Minimum conception to completion time with a minimum of staff.
8. Maximum reliability with minimum maintenance.

B. Additional

It was also the goal of the circuit designer to build circuitry that was sufficiently flexible to meet requirements for all other ground-based digital-data processing systems contemplated at the time (there were several).

II. Basic Circuit Modules

A. General Comments

Aside from the general system requirements of Section I, there are several other component-circuit-system requirements worthy of special note. They are:

1. That the circuits require minimum-specification components (single-ended specifications when possible).
2. That all components be commercially available stock items.
3. That the circuit techniques be sufficiently flexible to incorporate future higher performance devices to satisfy future

R. H. Baker is with the Massachusetts Institute of Technology, Lincoln Laboratory, Lexington, Mass.

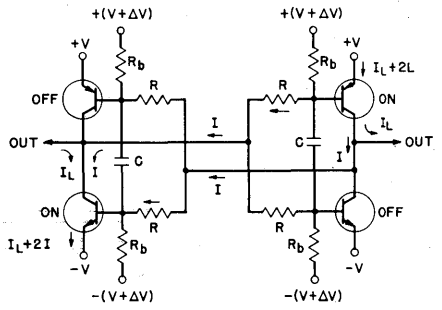


Fig. 2. Basic symmetrical flip-flop configuration

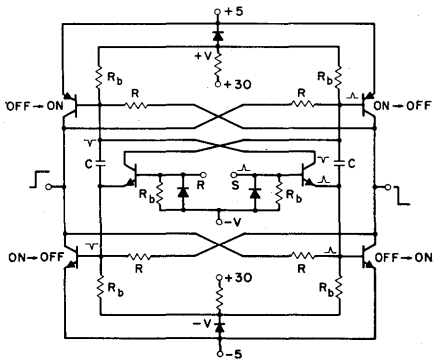


Fig. 3. Symmetrical flip-flop

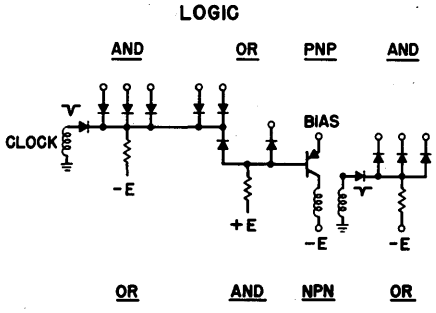


Fig. 4. A-c logic nets

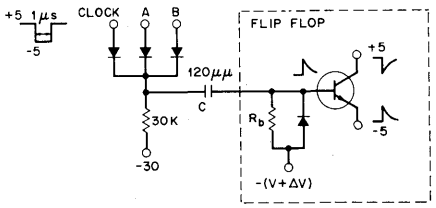


Fig. 5. Trailing-edge logic

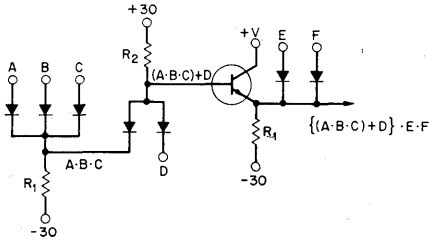


Fig. 6. Multilevel logic nets

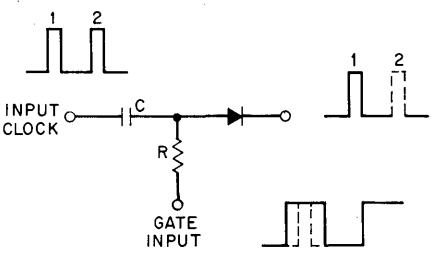


Fig. 7. CRD gate

higher-performance system requirements that may be needed.

4. That the circuit designs concentrate on fundamental device-network-system principles, thereby pointing out future trends in these areas and furthering the state of the art.

B. Buffer Inverter

With the aforementioned system and circuit requirements in mind and considering the transistor field as it existed in 1955, it seemed advisible to utilize p-n-p transistors to generate fast "positive-going" transients and n-p-n transistors to generate fast "negative-going" transients. This was first done in the form of a buffer inverter shown in Fig. 1. Some of the salient features of the circuit shown in Fig. 1 are:

1. Each transistor has positive base current applied while in the off condition and the numerical value of this current is equal to $\Delta V/R_b$.

2. Resistor values R and R_b along with ΔV may be chosen to yield a small uncertainty region (V_u). V_u may be centered around ground or shifted up or down over a large range.

3. R_L is merely a standby resistor and therefore can be almost any value.

4. The circuit is current demand, that is, it draws current (power) from the supplies in accordance with the load requirements.

5. The circuit power gain is approximately equal to β (grounded emitter transistor current gain), i.e.,

$$P_g=\frac{P_{out}}{P_{in}}=\frac{2v\beta I}{2vI}=\beta \qquad (1)$$

6. The circuit is fast. This is so because transistor hole/electron storage effects are minimized.

7. The transistor specifications are not critical.

8. The circuit has high utility and is flexible, i.e., the circuit may be: (a) used with any supply voltage within reason (note! Not necessarily equal positive and negative supplies), (b) designed with different values of R and R_b for the p-n-p and n-p-n to accommodate nonsymmetrical loads; (c) used with different-type transistors to create unusual circuit effects, etc.

C. Flip-Flop

D-C Considerations

Two buffer inverters of the type shown in Fig. 1 may be connected together to form a symmetrical flip-flop as shown in Fig. 2.

The d-c considerations of this circuit are the same as for the buffer inverter previously discussed. However it is worthwhile to point out that the ratio of total power drain on the supplies to the power delivery at the output terminals is almost unity, i.e.,

$$\frac{P_{out}}{P_{supplies}}=\frac{P_{out}}{P_{out}+P_{standby}}=\frac{2vI_L}{2vI_L+4vI} \qquad (2)$$

where $I_L=I\beta$ (3)

$$\text{then } \frac{P_{out}}{P_{supplies}}=\frac{2v\beta I}{2v\beta I+4vI}=\frac{1}{1+\frac{2}{\beta}}$$

$$=\text{efficiency} \qquad (4)$$

Transient Considerations

The important transient considerations of the symmetrical flip-flop may be seen with the aid of Fig. 3.

Salient points to note about the transient behavior of the network are:

1. The hole storage of the saturated transistor is minimized because large current is drawn from its collector during the storage time, thus clearing out the stored minority carriers quickly.

2. The rise and fall times, of the output, after the storage time is directly related to the ability of the conducting transistor to delivery current at the collector terminal and the terminal capacitance.

3. Transistor dissipation due to transient effect 1 and 2 set an upper limit to the average pulse repetition frequency of the circuit equal to approximately one-fifth the value of the transistor frequency cutoff (α_{co}), *i.e.*, the use of 10 megacycles, (mc) α_{co} transistor result in a circuit pulse repetition frequency (prf) of about 2 mc maximum.

4. The accumulation of charge on capacitors C by the transmission of trigger energy through it sets an upper limit to the maximum prf of about $0.15\alpha_{co}$.

5. It is important to note that restrictions number 3 and 4 allude to the average maximum prf; the "burst" repetition rate can be considerably higher. This can be shown as follows:

Let g=accumulated charge on capacitor C during one trigger pulse

$$\text{Since } g=C\,\Delta E \qquad (5)$$

where ΔE represents the change in voltage across C caused by one trigger pulse;

$$\text{then } g\leq iT \qquad (6)$$

where T=time between trigger pulses and i=charge current necessary to just discharge C in time T

Thus, as T is made small, i.e., the pulse repetition frequency is increased ($1/T$ = prf), then ΔE increases and at the limiting prf both transistors will be conducting at the same time (which usually causes one of them to burn out). In order to prevent ΔE from becoming too large, i must be increased (to discharge C in time T) by decreasing the values of R and R_b. There is a practical limit to increasing i, however, because decreasing R increases the transistor drive, thus lengthening the transistor storage time which in turn increases the transistor dissipation. However, this is an average effect and the burst prf (say for an interval of 100 trigger pulses) can be a factor of 3 to 5 higher.

6. It should also be noted that the network of Fig. 3 is triggered at all four points simultaneously, which minimizes the delay time without resorting to speedup capacitors across the coupling resistors R, and consequently rendering a flip-flop that is exceedingly difficult to false-trigger through load transients.

D. Logic

General

In most of the low-speed (below 500 kc prf) data processing systems built by the group to date, the nets have been constructed with diodes utilizing trailing-edge logic and pulse-level gating of the voltage mode. (Diodes must be able to sustain the full logic levels voltages in the reverse direction.) Transistor emitter follower circuits have been used internal to the nets, where additional current gain is needed. In higher-speed data processing systems (above 500 kc prf) current mode diode nets are used and delays are inserted to circumvent race problems. (Current mode nets are nets that operate with low voltage swings; the current is shunted from one path to another as in Fig. 4.)

Low-Speed Logic

1. Trailing-edge logic

An example of the trailing-edge logic is shown in Figure 5. Since each flip-flop can deliver 20 milliamperes, (ma) and each gate draws 1 ma, the "fan out" from a flip-flop to single level gates is 20. The "fan in" (number of gates that can drive a flip-flop) is almost unlimited. When 2-level nets are used the fan out is 5.

2. Multilevel nets

When more than two levels of logic are used, emitter followers are employed as shown in Fig. 6. This process "and-or-if-and-or" can be carried on indefinitely with two exceptions. They are: 1. that care is taken to guard against too much negative power being fed from the nets back into the flip-flops and, 2. that the nets do not oscillate. The first condition can be overcome by shunting resistors across the input to the nets (note that this wastes power) or using clamp diodes across the flip-flop outputs. The second condition is prevented by inserting buffer inverters after every fourth level in the nets. The over-all fan out for general two level nets of the voltage mode is 5.

3. Capacitor resistor diode gating (CRD Gating)

A simple inexpensive method of pulse-level gating is shown in Figure 7. CRD gating also has the advantage of built-in delay to circumvent circuit races. The disadvantage to this gating method is that it is inherently slow, the maximum speed (or minimum time between gating operations) being limited to about three times the time-constant RC. Another limitation is the signal-to-noise ratio unless care is taken to limit the clock pulse amplitude to less than the gate waveform amplitude. However, this latter limitation is easily taken care of with symmetrical circuits in that the wave-form amplitudes are automatically limited to the supply voltage values.

E. Negative Resistance Circuits (NRC)

General Comments

The fully symmetrical bistable circuit discussed in Section II-C (Figs. 2 and 3) can be used to generate very fast rising and falling transients (sharp leading and falling-edge square waves for example). However, for the reasons discussed, the circuit cannot be operated at a prf that would seem consistent with the fast transients. There are several specific reasons why this is so; but generally one can say that the circuits employ a great deal of "speedup" over drive (via the base-to-base capacitors) and since conventional speedup cannot be obtained without reactive elements, the energy stored during the transient period must be recovered during the circuit "rest" (static) interval.

It happens, however, that circuits can be built with much higher repetition rate with little sacrifice of transient time. This is accomplished by utilizing p-n-p and n-p-n transistors and coupling the two together to form a negative resistance circuit,[1] then paralleling two such negative resistance circuits to form a bistable circuit.[2]

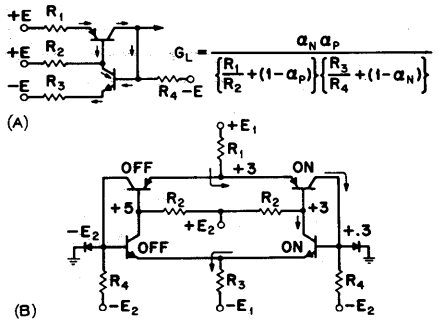


Fig. 8. Negative resistance bistable circuit

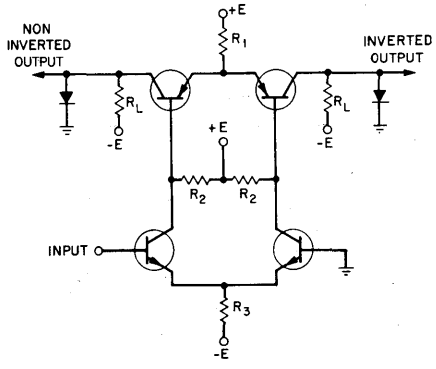


Fig. 10. Para-phase buffer inverter

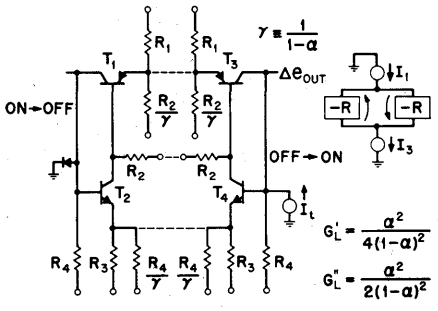


Fig. 9. Transient state of NRC flip-flop

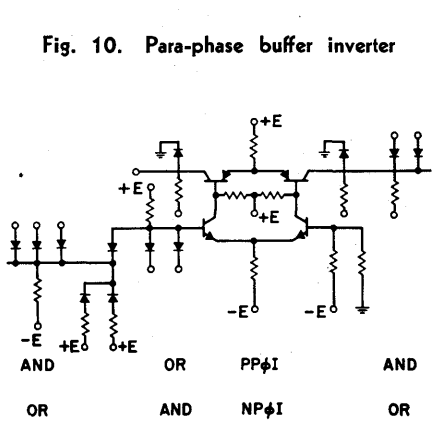


Fig. 11. D-c logic nets

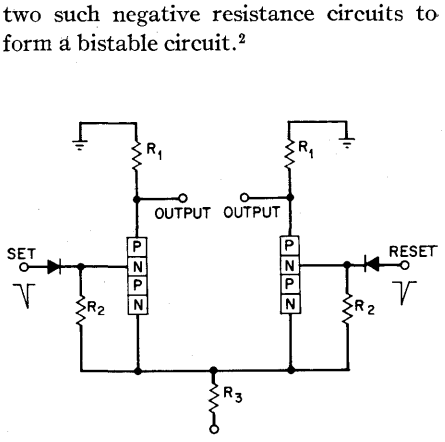


Fig. 12. Thristor NRC flip-flop

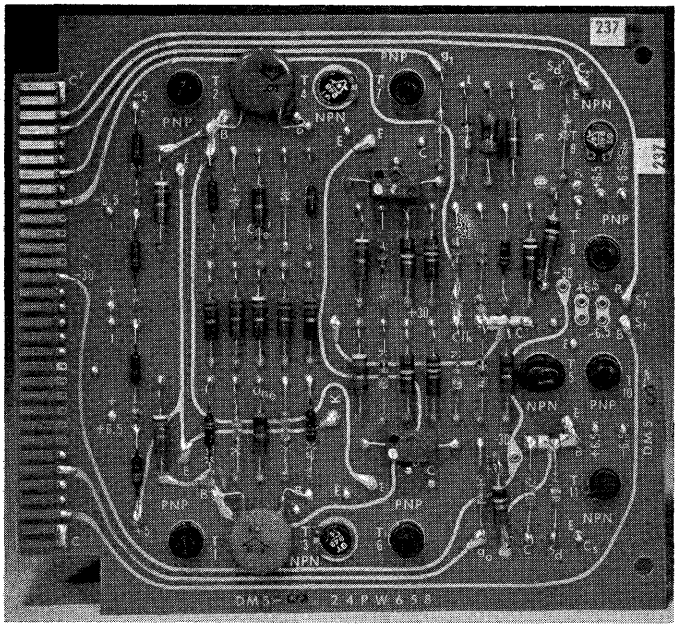


Fig. 13. Flip-flop module

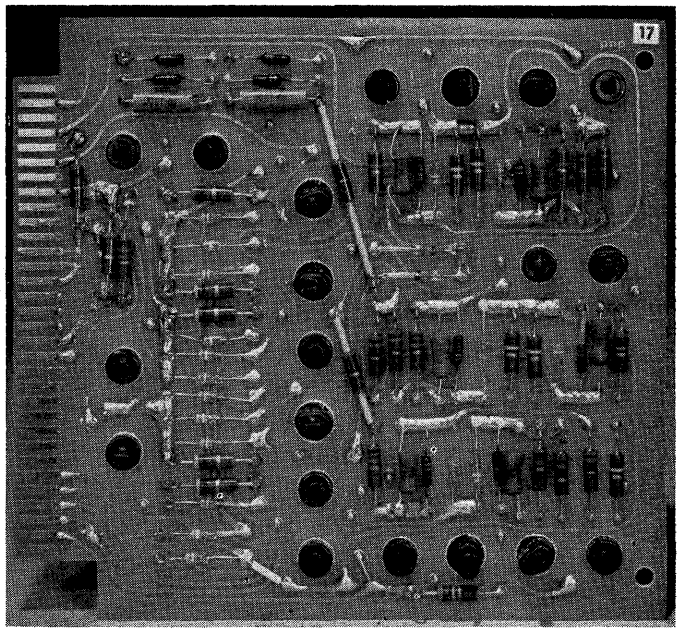


Fig. 14. Logical net circuit module

Bistable Negative Resistance Circuit

1. Static conditions

Consider the circuit of Fig. 8. When two transistor of opposite types are connected as shown in Fig. 6(A), the resulting circuit is as if a negative resistance was connected between the emitter terminals.[1] The gain of this network can be easily controlled by external means as indicated by the loop current gain equation shown in the figure; i.e., the current gain G_L is primarily dependent upon the product of the ratios of R_1 to R_2 and R_3 to R_4 (assuming a α_n, $\alpha_p \approx 1$). Thus, if two such circuits are emitter-coupled as shown in Fig. 6(B), the "on" pair will be stable if $R_1 > R_2$ and $R_3 > R_4$. It is also important to note that from the application point of view this circuit is very designable in that $+E$, $-E$, R_1, R_2, R_3, and R_4 may be varied independently rendering design easy for a variety of power supply values, etc. Also, the stability of the circuit is grounded base current gain sensitivity; i.e., α sensitivity, not β sensitivity which the circuit stable for a wide range of transistor parameters.

2. Transient conditions

The fact that the ultra-stable network of Fig. 8(B) may be made unstable is shown in Fig. 9. Referring Fig. 9 when the "off" pair (T_3 and T_4) is turned on, the emitter resistances (R_1 and R_3) are shunted by the base resistors (R_2 and R_4) of the opposite transistor pair (T_1 and T_2). This is an unstable situation as may be seen from the current gain equation of Figure 8(A). (Note: Under no condition, except when the transistors have a common base current gain less than one-half, can the conducting pairs have a gain less than unity.) Under these conditions the four transistor combinations can be characterized by two parallel negative resistance supplied by a current source whereby one-half "avalanches on" while the other "avalanches off." Thus we see that; 1. circuit regeneration is accomplished without speedup reactive components (this situation allows the circuit to be retriggered immediately after the transient without waiting for reactive components to recover), and 2. the circuits allow wide tolerance to: a. transistor parameters, b. supply changes, c. resistor values, and d. temperature (note that under static conditions, the circuits are effectively grounded base).

Experimental circuits designed on the NRC principal utilizing graded-base transistors have been operated at pulse repetition frequencies exceeding 30 mc and are conveniently designed to operate at a prf of 10 mc.

NRC Buffer

A buffer inverter is shown in Fig. 10. The same static and transient conditions

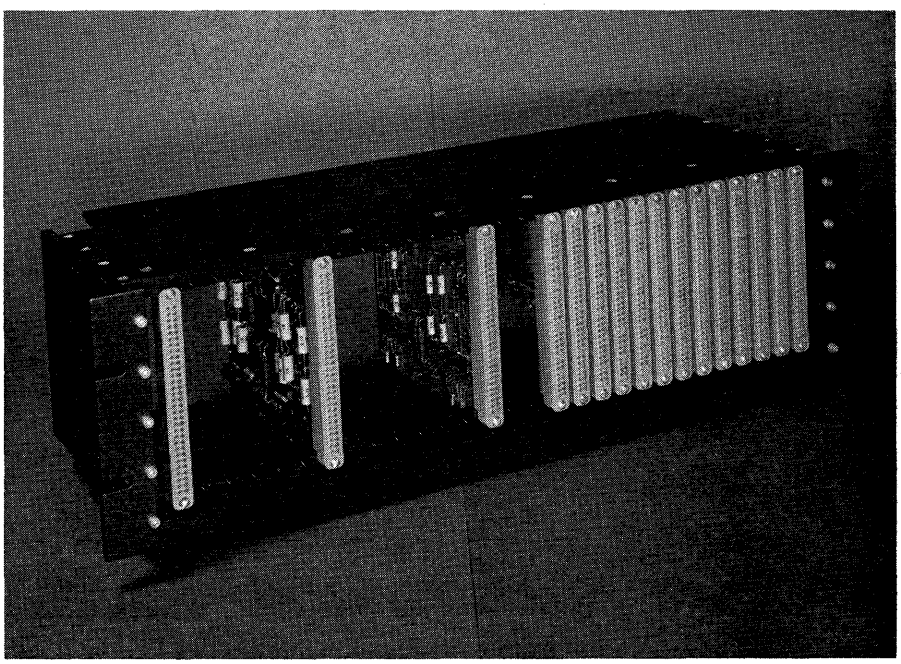


Fig. 15. Subrack

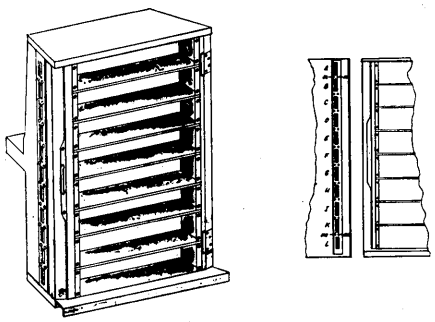


Fig. 16. Subrack holder

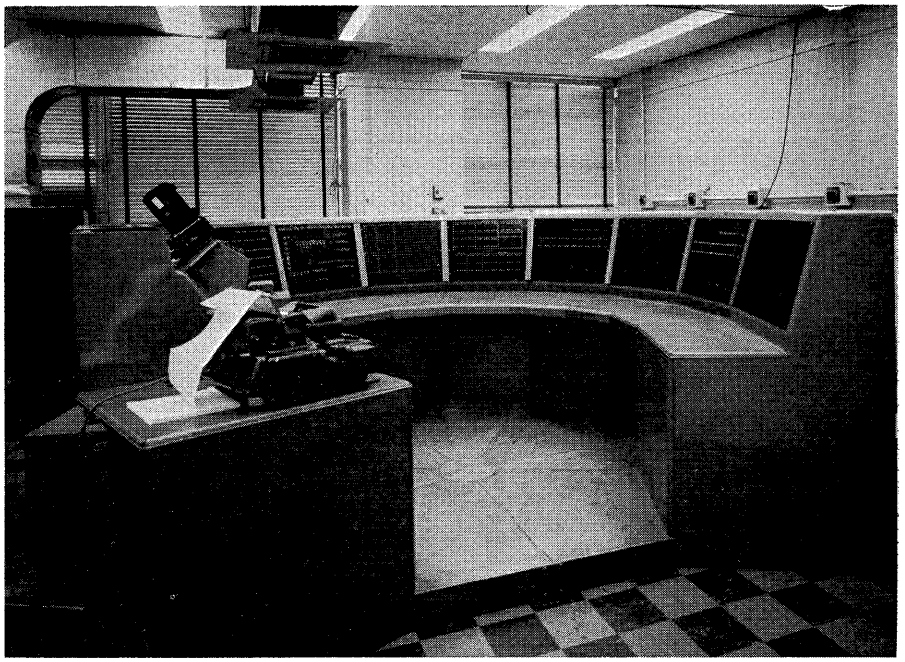


Fig. 17 (right). Computer console

that apply to the flip-flop are applicable in this case.

Fast Logical Nets

1. D-c coupled

A practical approach to building fast diode direct coupled nets is to keep the currents through the diodes low (thus enabling the diode to switch on and off quickly), keep the voltage swings low (thus minimizing the effects of shunt capacity) and to use high-gain NRC buffers. This approach is shown in Fig. 11. In addition to keeping the power level low, important considerations of this network are:

1. The logic is buffered after two levels, i.e., and-or-buffer-and-or-etc.

2. The circuit utility is increased from a system standpoint by having both inverted and non-inverted output available.

3. The clip levels are easily adjusted by providing a bleeder network at the base of the right hand n-p-n transistor.

4. The symmetry of the circuitry and the resulting system organization (supplies, etc.) allow an n-p-n para-phase invertor (NPϕI) easily obtainable by inverting the supplies, diode polarity and transistor types.

2. Fast a-c logical nets

An example of fast a-c nets is shown in Figure 4. As in the d-c nets the power level is kept small. The net clip level is adjustable through varying the emitter potential (bias).

Other important considerations for this type of pulse net are:

1. The a-c load per input reflected to the clock source is minimized because the clock source merely turns off the input diode.

2. The diode reverse breakdown voltage may be low (2 to 3 volts) which gives the diode manufacturer freedom to concentrate upon diode speed.

3. Symmetry concepts allow the use of "opposite polarity" designs.

4. Optimum speed bias conditions may be utilized for the transistor through the freedom to vary the transformer turns-ratio.

These considerations show why it is possible to construct a-c nets with a propagation time of 10 millimicroseconds, (mμs) (and-or amplifier propagation time).

Extension of NRC Techniques

It is the author's belief that NRC techniques offer one of the more attractive approaches for both higher speed and simpler, more powerful circuit design. The reasons for this are:

1. NRC techniques which utilize p-n-p and n-p-n transistors do give rise to high prf circuitry that is convenient to design and use.

2. It has been shown[1] that the four-terminal p-n-p-n devices are equivalent to a pair of transistors.

3. Circuitry work with three-terminal p-n-p-n transistors (RCA thyristor's) show that it is possible to construct a bistable circuit utilizing fewer components than do ordinary techniques (see Fig. 12).

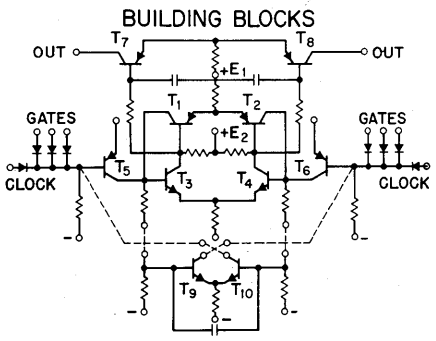


Fig. 18. Building block concept

III. Features of Basic Circuit Modules

The various circuit features that directly affect system design are discussed in the form of Tables I through IV. The author would like to point out that it is difficult to discuss the relative merits of different circuit approaches as applied to systems with a finite amount of words. The reason of this is that evaluation by necessity rests heavily upon experience and, hence, judgment, which is difficult

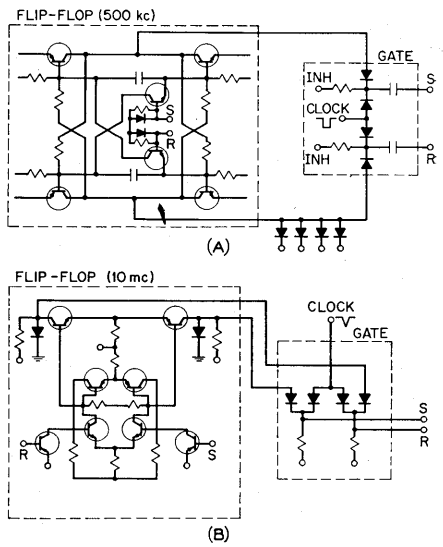


Fig. 19(A). Low-speed shift register stage (B). High-speed shift register stage

Table I.

Type Circuit	Propagation Time	Gain	
		Fan In	Fan Out
Symmetrical buffer	0.3 μs—5 mc units 0.03 μs—50 mc units See discussion Appendix I	1	10
Symmetrical flip-flop	0.3 μs—5 mc units 0.03 μs—50 mc units Appendix I	10	10
Trailing-edge logic	depends upon clock-rise time and diode delay 0.05 μs	15	1
CRD gating	diode delay time 10 m μs	15	1
Multilevel diode transistor nets	dependent upon power level Δv, ΔI 0.1 μs level with ordinary diodes and 5 mc transistors	5	5

Table II. Reliability

Circuit Type	Component	Power	Transistor	Noise	Temperature, Degrees C.	Design-ability
Buffer Fig. 1	Very good noncritical	Very good the circuits set own level	Very good	Fair	−35 +65 positive base current supplied	Excellent
Flip-flop Fig. 3	Very good noncritical	Very good sets own level	Very good noncritical	Fair	−35 +55 operation dependent upon transistor storage times	Good
Trailing-edge gating	Good dependent upon R_c time constant and clock pulse width	Very good not amplitude sensitivity R_c time constant dependent	Good clock and gate set by supply		Excellent	Good
CRD gate	Good	Excellent	Good amplitude sensitive amplitudes set by supply		Excellent	Excellent
Multilevel diode transistor nets	Fair See Appendix II	Fair	Fair	Fair		Good

to set down on paper. This is why the data are presented in tabular form. Every effort has been made to make the tables as complete as practically possible; and it is only after long deliberation and even then in some cases with reluctance that the quantitative values have been assigned. The key to the qualitative description is:

Excellent. . . almost all that could be desired
Very good. . noncritical, performs well
Good. . . . . all right under almost all conditions with normal care
Fair. . . . . works, but some care in design must be taken

In special cases where further explanation than that given in the tables is needed, the comments appear in the appendixes.

IV. Physical Properties

A. Circuit Modules

As noted in the introduction, the data processing group represented has chosen to sacrifice economy of components in order to save on "the circuits-to-system" realization time (the desirability of this is discussed further in Section V). Fig. 13 is a typical flip-flop board with associated input and output nets. Fig. 14 shows a typical logic board from the Arithmetic Section of the machine. Note that the component density was kept reasonably low to facilitate speedy implementation of the circuits for the system.

B. Subrack

Fig. 15 shows the circuit broad receptacle and subrack. One subrack holds 32 circuit boards of the type shown in the two preceding slides.

C. Subrack Door

Fig. 16 describes the type of rack used for mounting the subrack. The door (subrack holder) accommodates 10 subracks yielding a card density of 320 circuit cards per door.

D. Computer Console

Fig. 17 shows a picture of the computer console containing 10 doors or about 3,000 circuit cards.

E. Computer Comments

This computer contains about 20,000 transistors and approximately 150,000 diodes. The computer (which is now a working prototype) was built in about 30 staff years. Original estimates for the time to build this system was about one-hundred staff years. Considering the cost of research and development in a modern research laboratory per staff year, it is difficult to see how any other possible circuit-system techniques could have yielded a lower total prototype system cost. Indeed if one now projects the saving in time (staff years) along with the time advantage of having a working system for research studies (our major goal), we believe one will find a total saving in actual dollars by a factor of 2 to 4 over any other possible approach. If one further projects the saving in time to construct other data processing systems with the powerful time-saving circuit techniques discussed, it is difficult to see any other circuit solution for this work.

V. System Comments

Analysis of the difficulties that were encountered while constructing *CG24* showed that some of the fundamental device circuit system problems that are significant are:

1. Use of high-speed transistors. Although this was impossible at the time *CG24* was started (early 1956), the use of high-speed devices throughout the machine would lessen the clock source problem by allowing lower power nets to be used as well as utilizing fewer devices.

2. Build the machine into a smaller volume. Surprising though it is, the major trouble with nets arose, not from static considerations but rather from transient considerations. The fan out-fan in problems on a static basis are far less important than propagation time considerations. This is because the solution to fan out-in problems are simple in nature, where the solutions to propagation time problems are not usually so. Indeed some of the problems involving timing have been found difficult to even analyze, requiring a subtle knowledge of programming, devices, circuits, and hardware techniques.

3. Use of a-c nets. Closely coupled with the just preceding discussion is the practical problem of "debugging" the machine. The more liberal use of a-c nets facilitates the partitioning of nets for trouble shooting without the excessive use of dummy loads, etc.

4. Use of larger modules. The reliability of the machine has been extremely good from the standpoint of component failures. It would seem justifiable then to utilize a higher component packing density and more circuits per board with a resulting saving in space, lead length, etc.

Table III. NRC Circuits

Type Circuit	Propagation Time	Gain	
		Fan In	Fan Out
Flip-flop Fig. 8	30 mμs based on 50 mc transistors	15 diode current nets transistor driven	7
Buffer Fig. 10	30 m μs	1	7
D-c nets	50 m μs through an and-or- buffer combination	15	7
A-c nets	10 m μs	15	15

Table IV. Reliability

Circuit Type	Component	Power	Transistor	Noise	Temperature, Degrees C.	Designability
NRC	Excellent	Good	Excellent	Excellent	−35 +55	Excellent
Flip-flop Fig. 8			α sensitive circuits			α sensitive emitter current set
Buffer	Excellent	Good	Excellent same as flip-flop	Good	−35 +55	Excellent
D-c nets	Excellent	Excellent	Excellent	Good	−35 +55	Good
A-c nets	Excellent	Excellent	Excellent	Fair low level nets inherently have poor S/N ratios	−35 +55	Good

5. System packaging. Future trends indicate faster devices, circuits and systems. Present packaging techniques will prove the major stumbling block to the realization of fast systems.

6. One of the most powerful system aids that exist is a flexible set of circuits, particularly in this era of rapid technology advances. With this in mind, the building block concept shown in Fig. 18 is considered to be important. The circuit makes use of a basic NRC flip-flop (T_1 through T_4) along with a set of delayed outputs (T_7 and T_8) and a set of input amplifiers (T_5 and T_6). The circuit may be made more universal by supplying a set of delay switching current sources (T_9 and T_{10}) to completely circumvent circuit races.

VI. Universality of Techniques

Fig. 19(A) shows a single-stage diagram of a Low-Speed Shift Register (up to 500 kc prf). Fig. 19(B) gives the circuit diagram for a high-speed shift register (up to 10 mc prf). The principal feature of the low-speed stage is the ability to drive heavy loads. The main feature of the high-speed stage is the fast shift rate that can be attained.

It is believed that these techniques are well suited to: 1. realizing digital-data processing systems from basic circuits in a minimum length of time, 2. advancing the circuit art as such, 3. rendering feedback information to the device field, 4. utilizing the most powerful logical techniques. Therefore, they should prove interesting to and be used by: 1. any industrial concern interested in prototype machine design, large or small, 2. any research and development concern that is primarily system-oriented where the circuits are a means to a system end, 3. any establishment interested in studying the device-circuit-system relationship, i.e., component manufacturers, educational institutions, etc., because it is felt that the circuits represent realistic compromise in the device-circuit-system area for present and future techniques, and because the circuits and associated techniques are based upon fundamental consideration, realistic limitations, and practical experience.

Appendix I. Propagation Time

1. *Symmetrical buffer. Fig. 1*

The delay time through the circuit is approximately equal to the sum of the conducting transistor (old state) minority carrier storage time, and the rise (or fall) time of the transistor to be turned on. The minority carrier storage in the transistor to be turned off is shortened by the fact that a large current is drawn from its collector by the opposite conducting transistor. The fall (rise) time of the output is fast because standby current is not required and, therefore, all of the collector current of the conducting transistor (new state) is available to charge shunt capacity and to drive the load.

2. *Symmetrical flip-flop. Fig. 3*

Since the symmetrical flip flop is formed from two buffer inverters, the propagation time is about the same as that for the buffer inverter circuit, i.e., about 0.3μs for 5 mc transistors, and about 0.03μs for 50 mc transistors.

There is no loop delay in the circuit because the initial transient does not require feedback as the circuit is triggered at all four transistor base terminals.

Appendix II. Multilevel Net Propagation Time

The propagation time of a multilevel net (type shown in Fig. 4) is difficult to set because the time depends upon the power (impedance) level at which the net is operated. However, nets of this type can be readily built which exhibit a propagation time of about 0.1μs per stage utilizing ordinary diodes and 5 mc transistors.

References

1. Four Terminal P-N-P-N Transistors, **J. J. Ebers.** *Proceedings*, Institute of Radio Engineers, New York, N. Y., vol. 40, Nov. 1952, pp. 1361–364.

2. **R. H. Baker, E. J. Chatterton, Jr., A. C. Parker.** *Technical Report No. 130(TR130)*, Dec. 1957, pp. 7–10.

IBM Current Mode Transistor Logical Circuits

J. L. WALSH
NONMEMBER AIEE

THE CURRENT mode circuits discussed in this paper and in preceding papers[1] are intended for use in a very large, high-speed digital computer. From the circuit standpoint, this machine could be classed as a mixed synchronous-asynchronous system in which the outputs of chains of logic are often sampled by clock pulses. The system requires circuits which have delays of approximately 20 millimicroseconds per circuit. The basic circuit philosophy discussed here is well suited to the properties of the drift transistor.

Properties of the Transistor

The speed of response of a transistor switching circuit, neglecting stray capacitances, depends on the frequency response and the time constant of base resistance-collector capacitance. Of equal importance is the delay due to minority carrier storage, particularly when this delay approaches the maximum to which one wishes to restrict a circuit. As an example, if one wishes to restrict the circuit delay to 20 millimicroseconds and the saturation delay is 10 millimicroseconds, then only 10 millimicroseconds can be allowed for the transition to the switching threshold of the stages being driven. However, if the saturation delay is eliminated, a full 20 millimicroseconds can be allowed for transition to the switching threshold of the load stages. Clearly, then, a transistor switch operated out of saturation will not have to produce as steep rise or fall times to maintain the same circuit delay as a switch which is driven into saturation.

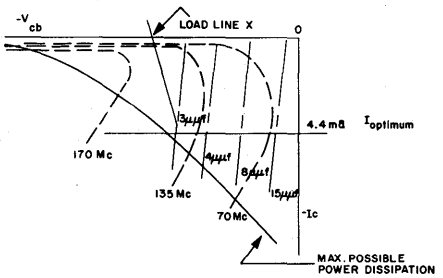


Fig. 1. Drift transistor characteristics

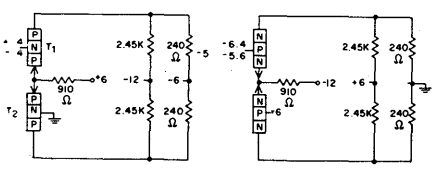


Fig. 2. Basic current switches

The frequency response and the collector capacitance are marked functions of the d-c operating point. The situation for a drift transistor is shown in Fig. 1, where curves of constant frequency cutoff and constant collector capacitance are plotted as a function of collector-to-base voltage and collector current. The collector capacitance varies inversely with the 1/3 power of collector voltage, but remains relatively constant as current is varied over a wide range. The contours of constant frequency cutoff bear some resemblance to a family of rectangular hyperbolas. In general, frequency cutoff increases as collector reverse bias is increased. However, for a fixed value of collector voltage, the frequency cutoff will decrease when the current exceeds the optimum value shown in Fig. 1. Also, the frequency cutoff is poor at very low currents.

The curves of Fig. 1 indicate that when a transistor operates on a load line such as *x*, frequency response and collector capacitance will approach an optimum within the hyperbola of allowable power dissipation. The disadvantage of this load line is that, when on, the transistor is required to dissipate more standby power than would be required with a load line that extended into the saturation region.

Basic Current Mode Switch

Consider the basic current mode switches shown in Fig. 2. The circuits are differential amplifiers with one input reference to ground in the p-n-p circuit, and −6.0 volts in the n-p-n circuit. The input signal to the top transistor, T_1, swings about ground, but only by an amount sufficient to switch current completely into either transistor T_1 or T_2.

In the p-n-p circuit of Fig. 2 the top transistor, T_1, is off when the input potential is at +0.4 volts and the bottom transistor, T_2 is conducting. When the input potential is at −0.4 volts, the bottom transistor is biased off and the top transistor conducts. Since the potential changes at the input are very samll the 910-ohm resistor and the +6-volt supply constitute a constant current source, and there is little difference in the current supplied to the top or bottom transistors. The output signal is developed across the 240-ohm load resistor returned to −6 volts. In order to make the output signal swing about −6.0 volts, a small current bias is added through the *2.45K* resistor returned to −12 volts. With this arrangement, the output potential varies from an off value of −6.4 volts to an on value of −5.6 volts.

The p-n-p circuit of Fig. 2 has an input referenced to ground, and outputs referenced to −6 volts. Because of the 6-volt difference between input and output, a p-n-p switch cannot drive another p-n-p

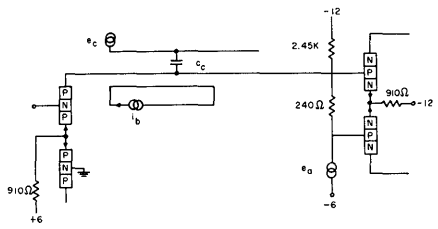


Fig. 3. Noise problems

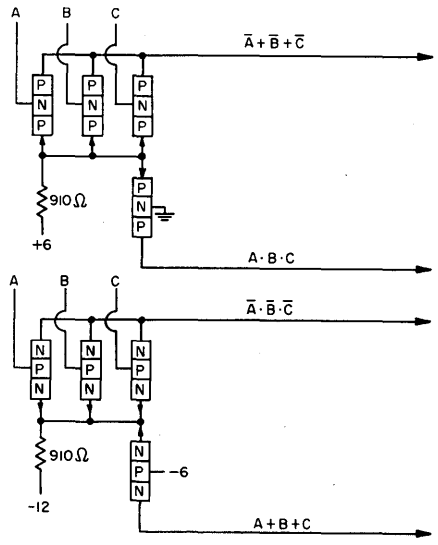


Fig. 4. Current switching "and" and "or" circuits

J. L. WALSH is with International Business Machines Corporation, Poughkeepsie, N. Y.

The circuits described in this paper are the work of the circuit design group of the International Business Machines (IBM) Product Development Laboratory, Poughkeepsie, N. Y. This work was supported by the Air Force Cambridge Research Center.

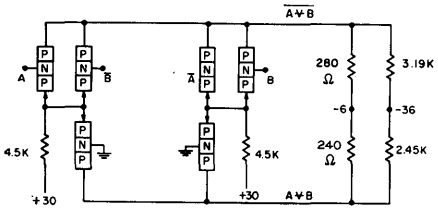


Fig. 5. Exclusive "or" circuit

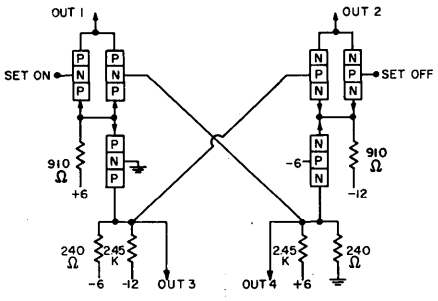


Fig. 6. Flip flop

switch. This difficulty is overcome by constructing a complementary circuit with n-p-n transistors. As shown in Fig. 2, the n-p-n circuit is reference to −6 volts and may be driven by a p-n-p circuit. The outputs of the n-p-n switch swing around ground and are suitable for driving p-n-p circuits. The two circuits of Fig. 2 are the basic current mode switches. A fundamental rule in their use is that p-n-p circuits always drive n-p-n circuits, which in turn may drive p-n-p circuits.

It is interesting to compare the characteristics of the basic current mode switch to the criteria required for high-speed switching, dictated by the transistor and shown in Fig. 1. The basic current mode circuit has a small load resistor similar to load line x in Fig. 1. Also, signal swings are small and the collector diode is reverse biased by 6 volts, so that the transistor operates in a region of good frequency response and low collector capacitance. Finally, optimum current can be switched by the proper choice of the emitter current source. With the exception of the very small region of poor frequency response at low currents, through which the operating point must pass as the transistor is turned on or off, the operating point is always in a region where the transistor bandwidth is high. Experimental results indicate more current should be switched than the transistor optimum of 4.4 milliamperes shown in Fig. 2. This increase in current over the transistor optimum was necessary because the input capacity of the stages being driven and the stray capacitance, rather

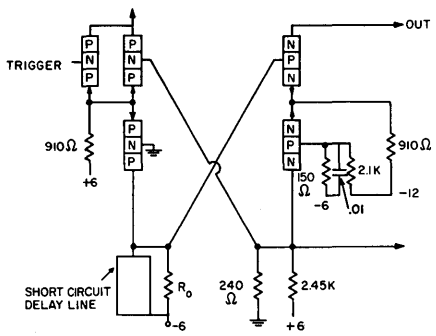


Fig. 7. Single-shot

than the time constant of the driver, were limiting speed. A further increase in current over the experimentally determined optimum will cause frequency response to fall off, and slower switching will result. In general, an increase in collector reverse bias will increase speed. However, a point of diminishing returns is reached, and a small compromise is made between speed and power dissipation. In Fig. 2 a collector supply of −6 volts and a current source of 6.5 milliamperes were used and judged optimum.

Before concluding the discussion of the basic switch, some mention should be made of noise problems and of the noise susceptibility of the circuit. When the load network is located close to the stages being driven, the basic current mode switch is not susceptible to power supply noise. Three possible noise generators are shown in Fig. 3. First, a noise generator (e_a) is inserted between the −6.0-volt reference supply and the load to represent noise on the −6-volt supply. Because of the ratio of the resistors in the coupling network (2.45K to 240 ohm), virtually all of the noise voltage will be applied to the base of the top transistor as well as to the base of the bottom transistor. When the noise is applied in this manner, it will cause little trouble, because the circuit is a differential amplifier, and switching can be accomplished only by changing the potential of one base with respect to the other. Noise on the collector bias and the emitter source supplies (−12 volts and +6 volts) will have to be of large amplitude to cause trouble, since these supplies are separated from the circuit by large resistors that join the circuit at points of low impedance.

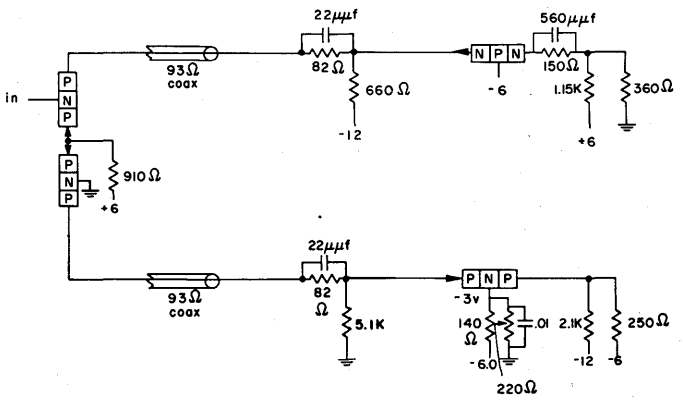
Inductive coupling, which might come from adjacent signal wires and which is represented by the noise current generator (i_b) in Fig. 3, is not troublesome. This is because the basic current mode switch presents a high output impedance in the loop through which any inductively coupled noise current must flow.

Of the three noise situations shown in Fig. 3, the problem of capacitively coupled noise from adjacent signal wires, represented by the voltage noise generator (e_c) and the signal wire capacitance (c_c), is the most critical. This is not serious, because the input node has a low impedance (220 ohms). One modification which will make the circuit more sensitive to noise is the inclusion of a peaking coil in the coupling network in series with the 240-ohm load resistor. The peaking coil will raise the node impedance, and capacitive coupling from adjacent wires will be more of a problem. Also, the peaking coil will act as a low pass filter in series with noise generator (e_a), and the differential amplifier property of the circuit will be decreased.

Logic Circuits

By providing additional transistors, the basic current switch of Fig. 2 may be extended to perform logic. Two logical circuits capable of performing the "and, or, and inversion" connectives are shown in Fig. 4. For this discussion, a binary "one" is defined to be the most positive input to a switch regardless of whether the signal in question is referenced to ground or −6.0 volts. One logical feature of current mode circuits can be seen in Fig. 4. That is, there are two logical outputs and they are complements of each other. As

Fig. 8 (right). Circuits for terminating transmission lines



a logical "one" is defined here, the top outputs are inverted and the bottom outputs are normal. The fact that inverted outputs are always available eliminates the need for a separate inverter building block in a system. In a long chain of logic where inversion is frequently required, the result is a reduction in over-all delay.

The two logical circuits of Fig. 4 will generate the necessary logical connectives required in a system. However, a separate "exclusive or" building block would be an advantage, since three of the logical blocks of Fig. 4 would be required to generate the "exclusive or" statement and also there would be a delay of two logical blocks in cascade involved. The simple arrangement of logical blocks shown in Fig. 5 will generate an "exclusive or" statement with a delay of only one logical block. Only the p-n-p version of the circuit is shown, but the n-p-n version can be formed in a similar manner. The "exclusive or" circuit consists of two parallel "and" circuits which generate $A \cdot \overline{B}$ and $\overline{A} \cdot B$. With the four inputs connected as shown, only one of the "and" circuit outputs can be conducting at any one time. Therefore, the "and" circuit outputs can be connected to form the "or" circuit required to complete the "exclusive or" function. Under the input conditions $\overline{A} \cdot \overline{B}$ or $A \cdot B$, the inverted outputs of both "and" circuits will be conducting and two units of current will flow into the load network. This network is designed to give a normal output only when both sides are conducting, and in this manner the inverted "exclusive or" statement is obtained. The inverted outputs will supply only one unit of current when the normal "exclusive or" inputs $(A \cdot \overline{B} + \overline{A} \cdot B)$ are present. However, because of the special coupling network, the output signal will not be large enough to switch the load stages. The "exclusive or" circuit requires both the normal and complemented input signals. This is no problem, since both normal and complemented outputs will always be available from the driving sources.

No discussion of a set of switching circuits for a computer would be complete without reference to some means of storage and some means of generating a well-defined pulse. The storage circuit in this case is shown in Fig. 6. It consists of two basic switches cross-coupled to form a symmetrical bistable flip-flop. The flip-flop can be set in either position through the pull-over transistors on either side. An "or" function can be built into the flip-flop by paralleling the pull-over transistors. Operation is in no way different from the logical block circuits of Fig. 4, previously described.

A basic current mode single-shot is shown in Fig. 7. The circuit again consists of two basic switches cross-coupled together, but in this case one side is through a short-circuited delay line. The bottom n-p-n transistor is biased off by the network at its base. The short-circuited delay line is terminated at the sending end, and the pulse width at the output is determined almost entirely by the time required for the wave front at the delay line input to travel down and then back up the delay line.

In any computing systems, situations are encountered where it is necessary to drive loads located at a considerable distance from the driving source. This may be done by driving conventional coaxial transmission line and terminating the coaxial line with either of the circuits shown in Fig. 8. Both the arrangements shown here are driven by a basic current switch. The n-p-n line terminator at the top of Fig. 8 is a Class A grounded-base amplifier. When the top output of the basic switch driving the line is off, the n-p-n transistor conducts and approximately 6.5 milliamperes flow into the current sink formed by the 660-ohm resistor and the −12.0-volt supply. When the top transistor conducts, the emitter current of the n-p-n grounded-base stage is reduced to 0.5 milliampere. The input impedance of this stage has a small inductive component and an impedance of 11 ohms. The 82-ohm resistor is added to increase the total impedance to 93 ohms and match the characteristic impedance of the coaxial line. The small capacitor compensates for the inductive input component. The value of the series resistance can be changed to match lines of different characteristic impedance if desired. The p-n-p line terminator at the bottom of Fig. 8 operates in the same manner. In this circuit the base is biased to −3.0 volts so that the output signal will be referenced to −6.0 volts. The n-p-n circuit differs from the p-n-p circuit in that it translates the output of the basic p-n-p switch from −6.0 volts up to ground level. Because of this, the n-p-n grounded base amplifier can also be used as a coupling means between two p-n-p logical blocks.

In concluding the discussion of the basic circuits, reference should be made to the component tolerances used in the design and the speed of operation that has been achieved with these circuits. Some design information will now be summarized.

Power supplies	±4%
Resistors	±3%
Transistors	$\beta = 20$
	70 megacycles $\leq f_{co} \leq$ 150 megacycles
	5 micromicrofarad $\leq C_{te} \leq$ 15 micromicrofarad
	40 ohms $\leq r_{bb} \leq$ 80 ohms
Emitter base breakdown	>2.5 volts

Circuit delays range from 6.0 to 22 millimicroseconds, the longer delays being associated with the larger loads. The basic logical block is designed to have a fan-out of 3 bases. The number of logical inputs is dependent on the number of loads. Maximum inputs are: 6 inputs for a load of three bases, 8 inputs for a load of two bases, and 10 inputs for a load of 1 base. In general, circuit delays are a function of the load and are not greatly affected by the number of logical inputs.

References

1. Millimicrosecond Transistor Current Switching Circuits, **H. S. Yourke.** *Proceedings*, IRE-AIEE Conference on Transistor and Solid State Circuits, Philadelphia, Pa., Feb. 1957.

2. High Speed Transistor Computer Circuit Design, **R. A. Henle.** Proceedings Eastern Joint Computer Conference, *AIEE Special Publication T-92*, Dec. 1956, pp. 64–66.

Discussion

T. R. Finch: The selection of a logical circuit for digital systems reminds me, in some aspects, of the procedure that a family goes through in selecting trees for the new one-acre lot. The house is located on a very barren plot, and since they would like to enhance their investment, they decide to buy some trees, and, of course, they want to get their money's worth. Well, they can invest in some evergreens which are very glamorous and present a very spectacular show above ground. Depending on where they live and the climate, in a number of years they may be able to call their little one-acre plot their dream place. However, they had better give some serious consideration to the root structure and soil as well as to the glamor above ground if they are indeed to enhance their investment.

I think to a certain extent the design of large-scale systems involves similar problems. It is fun to design a glamorous circuit with many storage elements. Sometimes we cannot understand exactly how they operate, but they operate like a "whizz-bang."

Companies that have been in the computer business for a long time, have to balance off system requirements and bring into being a system that has system efficacy. If there is available a whizz-bang system control and calculator that operates at 100-megacycles but must stand around for an hour and wait for the peripheral equipment to perform its data-processing task, then the system efficacy and low cost are most likely imaginary. What we are interested in is the integrated job of reliability at the least cost.

Transistor resistor logic (TRL) is not a glamorous circuit, it is just a "work-horse" circuit that gives you a great deal for your money. I think that the TRL circuit designer lives up to his responsibility of providing a basic circuit that can be understood, has a certain amount of flexibility, and provides the speed that is required across a rather flexible speed range. With TRL, and presently available transistor, is available, propagation time of the order of a tenth of a microsecond. It will not do this with all logical flexibility in all cases, but it is often more advantageous to exchange logical capabilities for the speed needed at lowest cost.

Another point on which I would like to comment is how we view the use of the transistor as a logical element as opposed to the use of another device in this application. This choice depends upon the size of the system, reliability, and cost principally. (We believe that in the end almost all characteristics, including reliability, have to be priced out.)

At this meeting we have seen circuits which use transistors for logical input, other circuits which use diodes for logical input, and TRL which uses resistors for logical input. We believe that resistors, given the same selection that semiconductors will be given, will prove more reliable and more economical and will justify their use wherever speed is not completely dominant.

My basic contention is that people who are using transistors for each logical input are indulging in "rich living," and they have got to get something out of it. What do you get out of it? In current mode switching you get speed. With direct coupled transistor logic (DCTL), you get simplicity and I contend that at the present time this is not enough to offset stiff transistor requirements and inadequate protection against sporadic error. However, I think that DCTL should be commended; it has been a real catalyst since its circuit simplicity has triggered us toward the evolution of digital systems employing simple logical circuits. I think now is the time to leave DCTL because of the problems related to the severe requirements on the transistors. As time goes on, these severe requirements are going to keep the DCTL transistor higher priced than the moderate requirements on other systems, and because the transistor is used for logical input it is going to be an expensive system.

In TRL, I think that there is a balance, a flexible balance, that permits the logical designer to meet the system's engineering needs. A study has been made to determine the efficacy of a complete intergrated system that places speed requirements on various blocks, sorters, and calculators, input-output stores, etc., and the study resulted in a group of requirements on logical speed. Once the speed requirements on the circuit are set, what do you want to optimize? The answer is the economics because reliability, maintenance, and the first cost of designability all are priced out.

The design of TRL circuits is flexible and permits a variety of speeds. Given a year or two, the diffusion-type devices are going to move the alloys out of business, and you will be able to obtain, for the same amount of money, diffusion-type devices of high-speed low-capacity reproducibility and reliability. Our premise in TRL is that it is a good bet to take advantage of what is going to be offered.

As to our thinking in regard to bringing a system into being a couple of years from now, we are betting on the diffusion-type devices with a simple circuit that we can understand and design, and can use in large numbers. We can realize our speed requirements, and we can obtain the various economics from the simple circuit by just scaling the impedance level up or down. This gives us the desired amount of protection against malfunction. If we want to go to higher speed we just lower the impedance level at the inner-stage, and we get high speed with some sacrifice of logical rules, or else we have to use more power.

Our premise again is that d-c power, as it has been refined throughout the years, is the best buy that you can get. You should invest in d-c power in many cases, although this is not true for air-borne applications. I think, if you wanted to design restricted, small scale special-purpose air-borne computers, you would get most value from carefully designed DCTL circuits: low power, low power consumption, and operation on low voltage.

I do not think that I would use TRL if I had a real time problem on a time-sharing basis where I had to get down to millimicroseconds. I do not know what I would use for sure, but I might just have to use some of the transistor rich living I mentioned. I have great confidence in the performance of diffused junction transistors, and I am not afraid to use them, but I do think that it is going to be some time before they are as cheap and as reliable as resistors. We believe the same is true for diodes. With the same care and attention given to the fabrication of all devices, I think that the resistor is going to outrun the diode and transistor for a long time to come.

James B. Angell: I have one disadvantage in arguing with Mr. Finch, in that basically I am speaking of DCTL predicated on experience. This is not just new and glamorous; we have had experience with it. So, to some extent what he, and I and others, in fact, will say about DCTL must be based on a fair amount both of design and actual usage and experience.

Before I go on with other comments, I would first like to augment to some extent the question of cost consideration that Mr. Finch started. This is that it is not really the cost per logical element which is of primary consideration in any computer, but it is rather the cost of, or per, logical decision. Certainly there are some cases where there are real time problems with a time scale of doing decisions sufficiently slow so that we are not primarily concerned with how many computations you can do per second. It is true throughout the majority of cases, if you can do the job faster, you can do it with fewer elements. You are not necessarily going to come out with the conclusion that DCTL is the right answer. I merely wish to augment Mr. Finch's thought.

What is the primary concern? What does it cost to make a given decision? DCTL is certainly not one of the faster types of circuit. It is roughly comparable in speed with what is called resistor-capacitor coupled transistor logic (RCTL), I believe.

This brings us to another point regarding cost. At least in the case of a limited number of machines a very strong factor in the cost is the designability of the circuit. That is, how easy it is to go from paper designs up to the final layout of the machine. I do not know this fact in the case of transistor resistor logic, but in DCTL I know that it is relatively straightforward. This circuitry can be compared almost to a relay logic circuit when you consider each relay to be of monopole, monothrow arrangement; the transistor is single-pole, single-throw relay, with, unfortunately, a connection between input and output. Nevertheless, the transformation from original layout to final machine is relatively straightforward once the original layout has been designed on the basis of simple logical rules.

Again with regard to cost, to do a given job in a given length of time with the slowest circuits will require more total circuits, and it has been found by experience with DCTL, and in various other cases, that reliability is not only on the basis of components, but also on the basis of interconnection of the components. How many nodes you have in the circuit becomes quite important; the total number of contacts, the total number of gadgets, if you will, that you can put in conveniently on a given panel. I add this point to indicate the fact that there is indeed something to be considered not only from the standpoint of a minimum number of circuits, but also for doing a given job in a given length of time.

Another factor to be considered is that of minimum component count. Minimizing the number of costly components does not

necessarily make for minimum over-all cost. It has been experienced and stated by one of the organizations that has been involved with DCTL that the over-all cost of a DCTL machine is not much more than the cost of an RCTL machine of equivalent capacity. Furthermore, it will require less design time because of the simplicity of the design and the smaller number of actual component values that must be determined.

Finally, I would like to leave you with the thought that the logic described by Mr. Walsh is a complementary version of DCTL, in that both use transistor logic.

R. H. Baker: I do not think that basically there is any difference between the circuitry that we can argue about, but for the sake of discussion I will make comments about each.

First of all, let me consider my own circuitry. On one hand the cost of the components is expensive, but on the other hand if we estimate 75 man-years for doing a job we save 30 kilo-dollars per man year, which is about what one does in a research laboratory. Then this is more than the total cost of all the parts of the machine that we were to build, and for this research job I do not see how we could have done it any other way. Our goal here is to build something for a system evaluation. I do think, however, that the cost of the components will go down, and that this circuitry of ours will become simpler and cheaper. It does have a high-performance record, and as time goes on it can be made simpler.

Now I would like to ask a question about TRL. To show no favoritism I will agree that resistors are more reliable than diodes, and if you can say that reliability is measured by how much per cent something changes, then a resistor changes less percentagewise than a diode does. On the other hand, there is a real question with the greater variation that you can stand with the diode than you can with the resistor because the latter does consume part of the margin of linear device. I am not at all sure that, with the greater allowable variation of diodes, that resistors offer a truly more reliable way. I also say that a hermetically sealed resistor (but it would be expensive) would be very effective.

I also want to point out that the designs of a paper machine are nothing like getting a machine to work. I think that some of the transient problems are the ones that are going to be really hard to solve, not the d-c problems in TRL. The problem is: One has a machine designed on paper, and you build it up and run a program through it to see if certain parts of the program make errors. This is usually the type of error where you must allow yourself enough flexibility to overcome the difficulties once you have started running the machine, transient-wise.

With regard to DCTL, I just cannot believe that a 3-terminal device is going to be as reliable as a 2-terminal device. I think the diode is more reliable than a transistor because a diode is governed more by a bulk phenomenon than is a transistor, hence, the transistor is more amenable to surface conditions affecting it than is a diode.

I think that the same comment is true of the circuitry described by Mr. Walsh. I think that a diode per logical input is expensive. On the other hand, I do not think that he expects this to last very long. In our position, we do not have to have something planned 5 years ahead, and this is where we differ basically from IBM, where the use is commercial as well as military. I think that by the time we go to his type of logic there will probably be something out with a little faster speed which will put diode logical input out of business. I think that it will probably be a p-n-p-n type of device produced by companies such as Radio Corporation of America, with three and four leads.

J. L. Walsh: I find it hard to speak in rebuttal since our problems are different than, for instance, the areas in which Mr. Finch operates. I have no argument with resistor logic, and I can certainly see that in many slow-speed applications it is economically worthwhile.

The advent of the drift transistor has opened up new areas of real high-speed operations, and here I do feel that saturation techniques will not be adequate. We will not in the future be able to neglect the ultimate storage delay which might be 5 or 4 or 3 millimicroseconds.

Chairman Felker: I have a series of questions to present to our speakers. The first question comes from E. J. Gauss, University of California in Los Angeles, for Mr. Finch: "Could not a gain in speed be obtained by the use of diode clamps to limit swings, and thus prevent saturation?"

T. R. Finch: Yes. This is a common way to operate and to use more power and what I call a "speed-up configuration." I commented on this when I pointed out that for a large number of applications falling in the fields of industrial control or business automation, the basic simple circuits with modern-day junction transistors will meet speed-up requirements without resorting to clamps for nonsaturation arrangements.

Chairman Felker: The next question is also from Mr. Gauss, for Mr. Angell: "Could not the "hogging" of current by one of several paralleled bases be minimized by the use of series resistors?"

James B. Angell: First of all, to minimize hogging, the manufacturers of transistors put in what is called "a base spreading resistance." This is not very well controlled, and indeed it has had some effect.

On the question of hogging, it is fairly stable with time so that if you have a transistor that meets the specifications initially, it will continue to meet that characteristic, not one of age. Second, to add such a base resistance does indeed slow down the turnoff of the transistor. If you will recall Fig.1 of my paper, when the first transistor is activated the second one is supposed to go off. The higher the base resistance of the transistor going off, the slower it turns off, until eventually it gets as bad as TRL. Of course, by adding the resistor you are adding an additional node. Experience has shown in general that when a circuit has relied on transistors to avoid hogging, it has been adequately successful, at least to date.

T. R. Finch: I wish to comment on Mr. Baker's comparison of DCTL to a degenerate design of TRL. This is definitely not the case because the big problem of DCTL is in its lack of adequate protection against malfunction resulting from a forward-biased "off" transistor, which is one of the strong differences between DCTL and TRL.

Chairman Felker: From A. Wennstron, Hughes Aircraft, for Mr. Baker: "Is diode recovery a serious problem in your high-speed circuits? What diode type do you use?"

R. H. Baker: Yes, it is a problem. However, I would like to point out that one manufacturer is now making a fairly fast diode. Although it was not meant to be particularly fast, if you are really after speed, you can get a diode made in the laboratory within a year that will have recovery time in the order of a millimicrosecond.

Chairman Felker: This question is addressed to Mr. Baker and Mr. Walsh, from S. Disson, Burroughs: "Please comment on pulse propagation problems in circuits with 5- to 50-millimicroseconds stage-delay characteristics, particularly on when and why coax is necessary. Also would Mr. Baker please comment on susceptibility of his circuits to coupled noise?"

J. L. Walsh: I think we gain when we use coaxial for cable that goes beyond 3 feet; below that, down to 8 inches, we use twisted wire. Beyond that, as far as propagation time, you have to keep track of the delay in the coaxial cable.

R. H. Baker: We do not know the answer to this question; this is what we are working on. We have a feeling that coax is not the answer because it makes the machine too big. It seems to me that if you are going to make the machine run fast, it is going to have to be small to keep the propagation time down. The only way to do this, I think, is to invent a new package where everything stays close to the ground plane. I do not think that there is a simple, general solution to this. All that I can say at the present time is that we have potential designs, and we are trying to get a solution.

Chairman Felker: This is for each panelist, and is from J. C. Hawkins, ALWAC: "Mr. Baker gave a component count figure of 20,000 transistors and 40,000 diodes for one machine. Would other panelists comment on, or estimate the number of components required for typical general-purpose computers using the type of circuits described?"

T. R. Finch: We made a comparison for a part of our system using about 7,500 TRL circuits. The difficulty here is that we worked to the needs of a system, an engineering analysis of what the blocks had to do in the way of terminal speed, if we came up with a solution that was faster it had little value unless the component count was reduced. The prime, dominant characteristic of Mr. Baker's circuit is to use more components to get speed. Also, Mr. Walsh's circuits are more complex than ours, so the comparison was made with DCTL. From our analysis it looked as if DCTL ran from $2\frac{1}{2}$ to 1 greater in transistor count, and counting joints and cost placed on resistors and transistors. Our estimate for the block, as of 1958 and of 1962 both, was that the

TRL was $2^1/_2$ to 1 less expensive for the same speed performance in the block.

Chairman Felker: Mr. Hawkins asks, "I believe that you stated that you had 34,000 transistors. How much would that be?

T. R. Finch: Twenty thousand transistors, and no appreciable number of diodes, and we are averaging about five resistors per transistor.

Chairman Felker: Do you have some figures on that, Mr. Walsh?

J. L. Walsh: I would not care to compare numbers with numbers used in TRL machines, as compared with the other numbers discussed here. Frankly, the comparison would be relatively favorable, but I do not think that the machines were designed to do the same job. I think it would be quite meaningless to point out what order of numbers has been used in TRL machines.

Chairman Felker: Here is a question from D. Weisser, IBM, for Mr. Baker: "Do you make any effort to select transistors for minimum minority carrier storage?"

R. H. Baker: The answer to this is, "No." With later-type transistors that we are working with now, storage is practically nonexistent.

Chairman Felker: We now have a question posed to all speakers from Louis Kurkjian, Hughes Aircraft: "Which of the 4 systems described are more applicable to synchronous and asynchronous operation?"

J. L. Walsh: I think that the system I described is applicable to either synchronous or asynchronous operation, but certainly there is less of a problem at high speed if you run equipment as synchronous.

Chairman Felker: I do not see why any of these circuits could not be used in both types of machines.

James B. Angell: DCTL is used almost completely in asynchronous machines.

T. R. Finch: All systems somewhere must be synchronized. As far as TRL is concerned, you wait for it to propagate and set up on registers, and then it is clocked out.

Chairman Felker: R. K. Richards, Consulting Engineer, asks of all speakers: "With all the virtues of the transistor circuits you have described, how do you account for the fact that existing tube computers are not experiencing competition from corresponding transistor computers?"

J. L. Walsh: I would be happy to answer that one. The existing tube computers are not really competitive from the standpoint of the ultimate use to be achieved. I think that most people are not designing replacements, but rather are designing equipment to do other bigger jobs faster.

Chairman Felker: The fact is that everyone is planning a transistor computer. I do not believe anyone is planning new vacuum-tube machines. I would expect that in 5 years most of the computers will be transistor based. In the military field transistors have pretty well won out competitively.

James B. Angell: There is no doubt about it.

J. L. Walsh: I would like to make another comment on that fact. Until we get a vacuum tube that has a diode drop equal to the diode drop of the transistors, I think that the transistor will be pretty difficult to beat for high-speed operation.

T. R. Finch: There is a comment I would like to add. It is difficult to supersede quickly a system implementation that has been refined for over 40 years.

Chairman Felker: This next question comes from O. S. Goda, Collins Radio Company, and is addressed to Mr. Finch: "Transistors *2N393* and *2N501* are relatively expensive devices at $7.50. How do you justify economy when 0.25-microsecond propagation time is obtainable with a diode OR-transistor NOT circuit using alloy-junction 8-megacycle, $1.50 transistor?"

T. R. Finch: If I had to bring into being next month a large-scale system, and it would be the only system I was going to design and put to work, then I think that the comment is well founded. I made the point that we were investing in the future, and that future is that the diffused-junction transistor, within a reasonable time on the order of 2 years, is going to be less costly, easier to make, and high speed. Beyond a period of 2 to 4 years I would question that you will be seeing many alloys around in a new system.

Chairman Felker: This question is for Mr. Angell, and comes from C. E. Baker, McDonnell Aircraft: "Exactly what does the increased number of transistors of DCTL, as compared to RTL, buy in the way of speed, reliability, and complexity?"

James B. Angell: In terms of speed for a given transistor, I would expect that the ratio is something like two; that is, DCTL is between one and two times faster.

What it buys in the way of reliability, I cannot say.

In complexity I think, without question, that there are slightly fewer nodes using DCTL because each element in DCTL is an active element with gain. You do not require occasional inverters or transistors to reinstate the loss you have had in the dissipating elements. Each element is active in this case.

R. H. Baker: In other words, they are both slow.

Chairman Felker: The next question is from L. P. Retzinger, Litton Industries, for Mr. Baker: "Do any oscillation problems arise from using emitter followers in logic?"

R. H. Baker: There definitely are, if you use followers in a row without isolating the diodes.

Chairman Felker: Mr. Retzinger also asks Mr. Finch: "Do you place a restriction on max-β, since storage time is a function of excess base current?"

T. R. Finch: We have debated about doing this, but we have not done so, so far. With the devices that we have been using, the microalloys and the microalloy diffused base, the requirements on the transistors are a minimum, d-c beta of 40 collector current of 5 mills with a possible "on" voltage of 125 millivolts. The worse-worse computation we have programmed on our computer, as I have shown you, did not put restriction on beta and included devices with d-c beta exceeding 100. I do think that we would get improved logical rules if we did.

Chairman Felker: Here is a question from W. Libaw, Magnavox Research Laboratories: "Do you use a pulse to trigger flip-flops? If so, is there a problem of gating the 'old' state of the flip-flop with the clock pulse?"

T. R. Finch: The way we are operating so far is straight d-c level. Although we are not treating the flip-flop any differently than a d-c logic stage, we have been giving consideration, as one might expect, to capacitor-coupled flip-flops. I might mention that the basic d-c level logic circuit covers the order of 90 per cent of logical needs, with the larger part of the remaining 10 per cent in the control circuit. Special pulse triggers were not used for the speeds described.

Chairman Felker: Here is a question from W. Woods, RCA, for Mr. Finch: "In calculating the performance curves for TRL circuits, what specifications are assumed for minority-carrier storage, and how well can this characteristic be controlled in practice?"

T. R. Finch: Minority carrier storage effects are controlled by, 1. transistor design in regard to lifetime of storage bodies and the volumes of these bodies, 2. the depth of saturation due to forward drive, and 3. the sweep out reverse current. Since we were primarily interested in the worst-worst performance, we placed requirements on the transistor turn off under our most severe logical conditions, in our case, 5 active inputs before turn off, and minimum available sweep out current. The forced beta became about 3, and the maximum allowable turn off time for the *2N393* was specified as 0.35 microsecond. Under these conditions, it was not unusual to find delay variations ranging down to less than half of the specified maximum.

MicroSADIC, A High-Speed System with Variable Format Output

HELMUT SCHWAB
NONMEMBER AIEE

MICROSADIC is a data acquisition and handling system. It is, therefore, the link between the test object and the electronic computer in an advanced scientific or technical study. Fig. 1 shows the cabinet which contains the central data processing electronics of the MicroSADIC system. It contains the data sources for time and selectable title constants. It also contains the commutators, the digitizer, the central program unit, and a power supply. Data are coming in through transducers from the test directly or through frequency modulated-pulse duration modulation (FM-PDM) radio link through telemetering equipment. The MicroSADIC output is usually magnetic tape. The tape unit is of the same size as the unit shown in Fig. 1. The MicroSADIC meets the need for high speed in data processing, high accuracy, compatibility with all standard computers, reliability, and outstanding flexibility. Its design for universal use allows the build-up of a variety of complete data handling systems. The following paragraphs will show how these characteristics are achieved. Special attention is given to the characteristic of flexibility which makes it possible for the MicroSADIC to create basically different data output formats as required for different computers with their contrasts in format and input specifications.

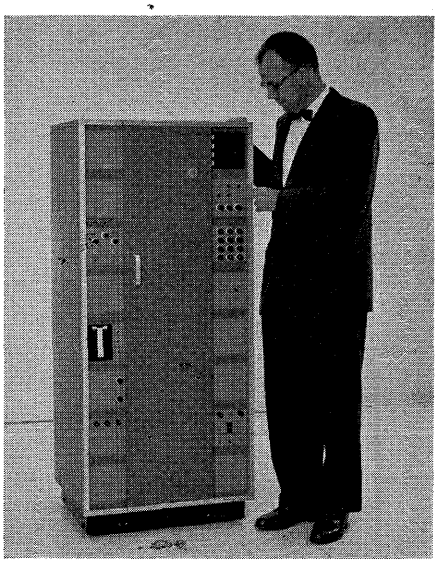


Fig. 1. Basic MicroSADIC cabinet

Data Sequencing and Coding

The operation of the over-all system is illustrated by Fig. 2. The test object may be a missile, a windtunnel model, a power engine, a chemical plant, or any other complex system. The transducers feeding the MicroSADIC system provide electrical signals proportional to the test values. Examples of the great variety of these are pressure, temperature, stress, and angle of attack. Two different types of test have to be considered, the air-borne test and the ground test.

In case of an air-borne test, see Fig. 3, the transducer outputs may be fed into an air-borne commutator. This switching device sequences the data and feeds its output into the air-borne signal converter. The air-borne signal transmission normally use pdm and FM modulations. A ground station records such signals on magnetic tape. The tapes are brought to the test center and played back into a demodulator. The demodulator provides straight pdm outputs for a pdm digitizer. It also demodulates the FM-FM signals into amplitude modulated (AM) signals. Another MicroSADIC digitizer converts these signals into digital data in the desired code. The operation is simpler in the normal ground test, (see Fig. 4). In that case the analog transducer outputs are given to a standard commutator which feeds directly into the analog digitizer.

Digital transducer outputs are also brought to a commutator. Thereby, the sequence of their recording is controlled. This sequence is important since the later computer program has to be established on this order in the incoming data flow. In Fig. 5 note how the different data sources feed into the central program unit. The pdm digitizer had been mentioned before. The commutator for analog signals is connected to the digitizer which feeds into the programmer.

Another data source is the time information. Most tests are of the "dynamic" type and, therefore, require time

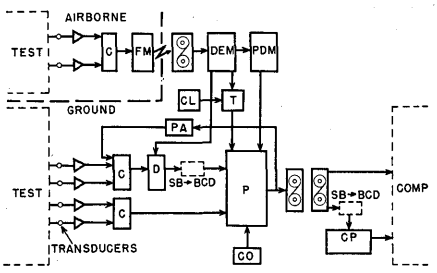


Fig. 2. MicroSADIC system, block diagram

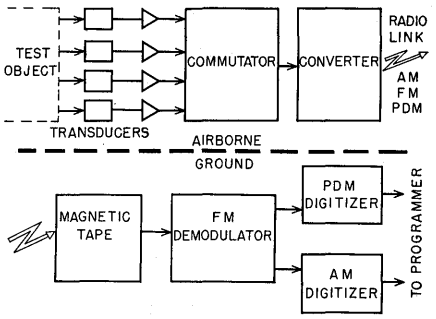


Fig. 3. MicroSADIC inputs for air-borne tests

information for each data point. Static tests make use of the time unit for counting test points and other purposes. There are also switch combinations which give digital constants and may be used for test identification (date) or special computer information (program selection).

In summary, then, there are five digital data types:

1. Constants.
2. Counter output from time accumulator or decoder.
3. Digitized data from the analog digitizer.
4. Other data.
   (*a*). From the subcommutator for digital inputs.
   (*b*). From the PDM digitizer.

Types 4(*a*) and 4(*b*) will not normally be used together. Therefore, normal MicroSADIC build-ups use four types of data. The sequence within each group is determined by the commutator and the code is determined by the digitizer. Straight binary or binary-coded decimal are mostly preferred. The *sb/bcd* translator in Fig. 5 is used in cases where both codes are wanted simultaneously. It might also be that the general system is selected to be straight binary but that for a special test *bcd* output is wanted.

The translator is based on the "doubling" principle. The numbers are serially shifted from an *sb* register into a *bcd*

HELMUT SCHWAB is with Consolidated Electrodynamics Corporation, Pasadena, Calif.

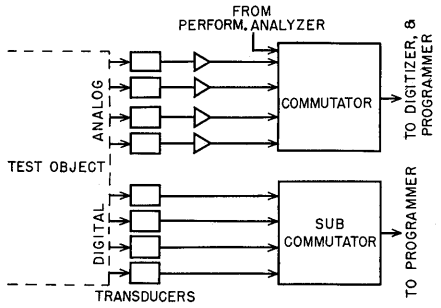


Fig. 4. MicroSADIC inputs for ground tests

register and the content of the *bcd* register is doubled with each step.

Fig. 6 shows possible output uses. The output tapes of the MicroSADIC system can be played back directly into one of the standard computers and they can also be played back into a tape to card converter and the cards may be sent to the computer. In this conversion again a code translator may be used for the same reasons as mentioned before.

The Programmer for Operation Mode and Format

The programmer is the link between the digital data sources and the tape, hence, the computer. This determines its task as follows:

1. Operational control of the MicroSADIC System.
2. Format control of the data flow.

The great variety of test applications and especially the contrasts between different computer characteristics require an exceptional flexibility of the programmer.

Operational Control

The operational control determines the chronological length of the test and, thereby, the length of the computer problem. It also determines what types of data

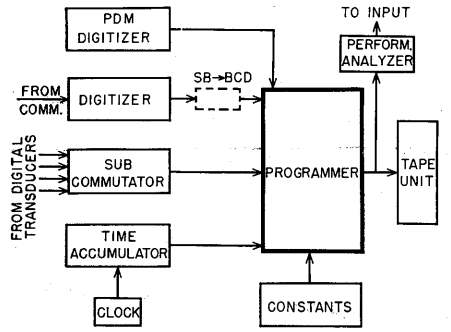


Fig. 5. Central units of the MicroSADIC system

are to be handled in each data block. The computer program has to conform to this selection.

Two types of operational control are provided: data-sequence control, and time-sequence control. The data-sequence control allows the writing of one of the following data combinations:

1. Constants only.
2. Time, digitized data, other data.
3. Constants, time, digitized data, other data.
4. File mark only.

Program 1 is used in order to flag the data flow to the computer or the magnetic tape. This can have the meaning of a computer instruction, title, identification numbers, etc.

Program 2 is typical for a normal test run.

Program 3, with the repetition of the constants in each data block, is useful in formats for punched card output.

Program 4 marks the end-of-test and serves to stop the computer at this point.

The time-sequence control offers three selections:

1. One sampling period only.
2. Repeated sampling periods.
3. Continuous operation.

Selection 1 is used for system checkouts. It might also be useful in static tests where one data point characterizes sufficiently each setup.

Selection 2 signifies that at any given time only one sampling period is recorded. But such commutation periods are triggered periodically, controlled by the time unit of the MicroSADIC system. Repetition rates can be anywhere between milliseconds and hours.

Selection 3 is the normal operation for a dynamic test with continuous data acquisition. It requires separate start and stop commands. The start-stop function is another operational control of the over-all system.

Format Control

Data are commonly stored in the form of pulses on a magnetic tape. The magnetic pulses are all identical. It is their location on the tape which determines the content of their message. Therefore, a strict order of writing pulses on the magnetic tape must be determined. This order is called the "format" and is different for different computers. These contrasts in computers become more important as more computers come into

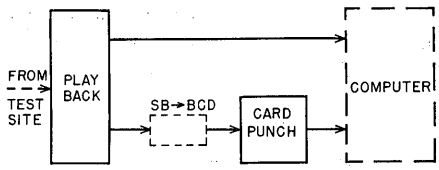


Fig. 6. MicroSADIC system outputs

use and as more computer types appear on the market. It is hoped that computer manufacturers will give attention to the point of format standardization. Standard values should be decided upon for the number of tracks, types and location of parity checks, length of data words, alignment of writing heads for magnetic tape, and quantitative characteristics of the magnetic pulses. The remaining variable of number of words in a data block should be sufficient to take care of necessary contrasts in computer characteristics.

Fig. 7 shows 16 formats which a normal MicroSADIC programmer is able to write. They are grouped around the main computers and their peripheral equipment. This concerns the International Business Machines (IBM) *704* and *709* with IBM card punch or printer, IBM *650* with IBM card punch or printer, MilliSADIC card punch, Remington Rand *1103A* with Remington Rand card punch or printer, and some special formats for high-data handling speed, which require special editing equipment. More formats can be written in as far as they represent a combination of the characteristics of the 16 formats which have been shown in Fig. 7.

The great flexibility of the MicroSADIC programmer is achieved by its operational design philosophy. This means that the over-all operation is subdivided in a number of fundamental suboperations. Each such suboperation is performed by a special building block. These building blocks again are designed to offer great flexibility. Some of these subunits are:

FORM. NO.	COMPUTER	CODE	$d/_s$	$s/_w$	$s/_{bl}$	$d/_T$	$d/_{CO}$
1	704–709	SB	11b	3	s.p.	36b	12
2		BCD	3	2		9	
3			<6	1			
4	704–709, PERIPHERAL		3	2	20	6	
5			<6	1	10		
6	IBM PERIPHERAL		<4		15		
7			<6		10		
8	704, M-SAD PUNCH		3	2	20		EXT.
9			<6	1	10		
10	M-SAD PUNCH		<4		15		
11			<6		10		
12	IBM 650		<5	2	100	<90	10
13	R.R. 1103A	SB		3	s.p.	36b	12
14	R.R. 1103A, PERIPHERAL	BCD	3	2	30	$6/_{12}$	$12/_6$
15			<6	1	15		
16	ED. EQUIP.	ANY	ANY		ANY	ANY	12

Fig. 7. Available tape formats for the MicroSADIC system

1. The digit counter which counts the number of characters in a word (optional 5, 6, 10, or 12).

2. The data group counter (optional any number from 1 to 10).

3. The sample counter (optional 1, 10, 15, 20, 30, 45, 60).

4. The block gap generator (optional, any time).

5. The parity check generator (optional odd or even check) and several more.

The selection of a specific performance is done by two means, patchboard and switch.

PATCHBOARD

A 100-point patchboard allows the selection of all formats in Fig. 7 and additional specific modifications. Normally one format is needed corresponding to the local computer. However, in some places there are different computers available, for instance, a smaller local computer for quick evaluation and a big central computer for final evaluation. Sometimes two different central computers are available and the selection depends upon their work load, so, a fast format change is then needed. Some tests may be made twice in two different formats in order to make greatest use of the computer facilities. Then, the number of wanted formats is patched and specific points patched to the switch inputs.

FORMAT SWITCH

This is a multiple, double-throw switch, which effects a fast change of the patching pattern. Two or more different formats may, thereby, be selected by the turn of a knob.

Standard selections could be:

a. IBM *704, sb* (format 1)
b. IBM peripheral, *bcd* (format 6)
or
a. IBM *704*, and peripheral (format 4)
b. Remington Rand *1103, s.b.* (format 13)
or
a. IBM *650* (format 12)
b. MilliSADIC card punch (format 8)
or any other combination

It should be kept in mind that the data processing speed is different for the different formats. This may motivate the computer and format selection. It also shows the influence of the "contrasts in computers" on the data-acquisition systems and thereby on the original test itself. It is believed that the MicroSADIC system succeeds in giving the liberty to scientists and engineers to select the best way for their data processing problem. However, the limitations inherent in the computer characteristics can not be avoided.

The Over-All System

The beginning of this paper mentioned the importance of high speed in data processing, high accuracy, reliability, flexibility, and completeness as main system characteristics. The following figures show how MicroSADIC is able to realize the following goals:

MAXIMUM SPEED: Format 16: 10,000 samples per second average, s.b. code, each corresponding to 3 decimal digits. (20-kc clock rate)

ACCURACY: $\pm 0.1\%$

RELIABILITY

Industrial reliability can be defined as the ratio of working time to trouble-shooting time for the system. The working time is mostly given by the components' life time. MicroSADIC uses only long life components and transistors as active elements. The circuit design tolerances correspond to the known end-of-life values for each components for instance $\pm 15\%$ for resistors. The trouble shooting time is greatly reduced by MicroSADIC's modular and operational design and numerous checking and test facilities, marginal checks included.

FLEXIBILITY

The writing of 16 different formats is standard. Additional modifications are possible. Two different codes can be written. Input data can be digital, analog, or pdm.

COMPLETENESS

Special equipment (specific transducers, code transformers, etc.) allows system combinations for practically all tests. The standard equipment covers the essential applications as known from the past and expected in the future.

EXPERIENCE

In order to give an idea of the background of the data acquisition technique, it can be stated that 30 data-preparation systems (SADIC and MilliSADIC) have been installed during the last 5 years by this company only. MicroSADIC represents the latest state of the art in components, manufacturing, and systems design.

The MicroSADIC provides an efficient input means to many computers with their contrasting input conditions.

A Computer-Integrated Rapid-Access Magnetic Tape System with Fixed Address

R. L. BEST
NONMEMBER AIEE

T. C. STOCKEBRAND
NONMEMBER AIEE

THIS paper describes the internal tape library system planned for the *TX-2* computer at Lincoln Laboratory, Massachusetts Institute of Technology (MIT). One hundred magnetic tape transports will be under the control of a central electronic system; the system will have a storage capacity of 10^{10} bits and an access time of about 30 seconds. It is particularly well suited for use with a computer of large random-access storage capacity such as *TX-2*, which has a core memory of $2^1/_2$ million bits. A simple tape transport having a high-speed search mode with redundant information transfer will make the ultimate library system for a computer, reliable and relatively inexpensive.

The tape transports are controlled by electronic circuits closely integrated with the computer. A permanent, constant-density timing track on the tape provides the speed reference for the control circuits and makes possible fixed position address-

R. L. BEST and T. C. STOCKEBRAND are with the Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, Mass.

The research in this document was supported jointly by the United States Army, Navy, and Air Force under contract with the Massachusetts Institute of Technology.

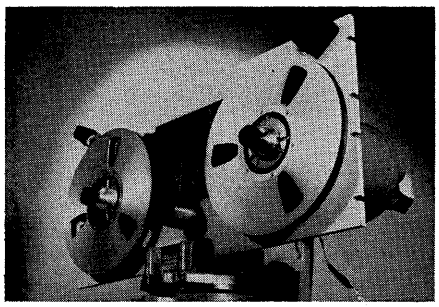


Fig. 1. Tape transport mechanism

ing and a variable rate of information transfer. The transport does not employ a constant speed capstan: two conventional 3-phase motors pull the tape in either direction. The control system varies the speed by modifying the torque on the driving motors after comparing the timing track bit rate with the desired bit rate. By this means the computer can select an appropriate speed for recording data in real time over prolonged periods. A separate channel of block marks enables the computer to locate information blocks at any speed, including high-speed search at 920 inches per second (ips). Read and write speeds are 30 to 100 ips and acceleration time to these speeds is 1/2 to $1^1/_2$ seconds.

High reliability in the transfer of information to and from the tape is gained by making five channels out of ten redundantly paired tracks. Thus in the ten-track head assembly there are three information channels, one timing channel, and one block mark channel. Appropriate head shielding reduces crosstalk and permits reading of the timing channel when other channels are being used for writing. Since the amplitude of the signal from the tape varies greatly with speed, a system of recording is used that allows the polarity of the flux change, rather than its amplitude, to be sensed in reading. High-gain amplifiers may then be used in which the output stages are normally saturated and the gain need not be closely controlled.

Computer/Tape Library Relationship

The primary objective, a large library of quickly accessible information, is provided by a large number of tape transport mechanisms controlled by a small number of circuits which are closely integrated with the computer. One feature of *TX-2*[1,2] is its multiple sequence control; that is, it can share its attention between equipments which are operating simultaneously in real time. The computer commands the selected tape drive to attain some mode of operation, then turns its attention elsewhere until the tape control tells the computer that the desired mode has been attained. Multiple sequence control also enables the computer to vary the speed of the tape during information transfer: this feature would be used, for example, if the computation itself depended upon external, real-time events and the information transfer had to be synchronized with the results of the computation.

Another feature of *TX-2* is its large, random-access core memory[3] which can store large blocks of information at one time and which can be used as a buffer while the tape is accelerating. The control element for the tape library accepts the computer's commands rapidly but does not require instant response by the computer to events in the tape system. Generally the control takes care of the simple local problems and leaves the complex problems of operation to the computer, and therefore the programmer.

The tape mechanism, which has a wide variety of speeds, can search through a reel of tape much faster than the computer can accept information. For this reason, blocks of information on the tape are tagged with block marks that can be read at speeds up to the maximum, 920 ips. The read and write instructions command only a single 9-bit transfer between memory and tape. A series of such instructions is necessary to transfer large blocks of information; the instructions must occur at an average rate determined by the tape's speed.

Transport Design and Motion Control

Mechanical Design

The tape transports used in this system were made as simple and fool-proof as possible: they consist of a read-write head assembly, two reels, two drive motors, and a tape guide. The drive mechanism has no capstan. Thus a good deal of mechanical complexity is eliminated and a wide range of tape speeds is made possible. Fast starts and stops are precluded, however: 1/2 second and $7^1/_2$ inches of tape are required to reach 30 ips.

Figs. 1 and 2 show the transport mechanism. The motors are flange mounted, 1,800 rpm, 3-phase induction motors of the conventional type which have roughly constant torque characteristics when operating well below synchronous speed. The horsepower (hp) rating, and therefore the torque, is as high as possible, limited by the tensile strength of the tape. One-eighth-hp motors, each driven by a magnetic amplifier, provide the proper torque to operate 10-inch reels mounted directly on the motor shaft and loaded with polyester tape, 0.001-inch thick and 1/2-inch wide. Maximum tape speed is about 920 ips when the driving reel is full.

The head assembly and guide are shown in the insert, Fig. 2. The relatively large, constant radius of the guide reduces the pressure between tape and guide: At speeds above 20 ips the tape floats on an air cushion and is thus easy to edge guide. Skew, caused by nonuniform tape tension across the width of the tape and by variations in tape width

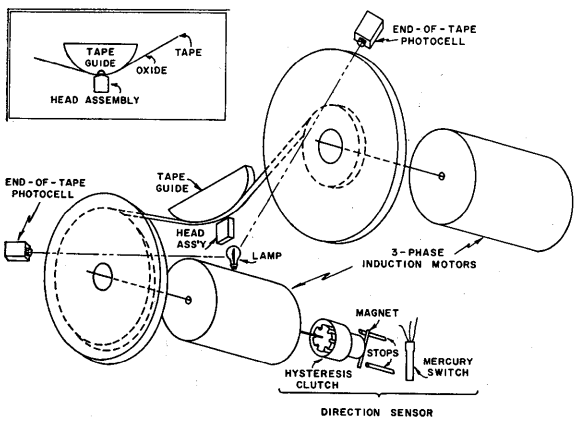


Fig. 2 (left). Tape transport mechanism

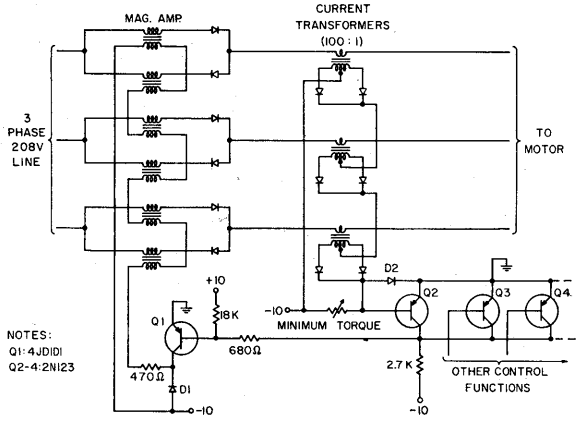


Fig. 3 (right). Motor control

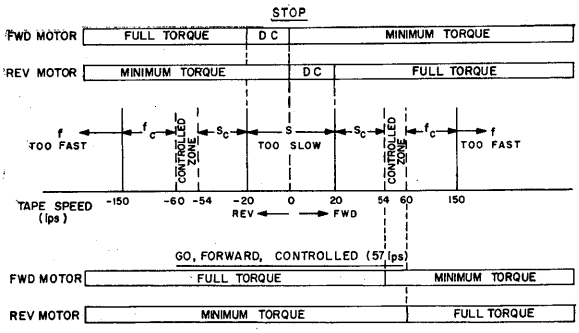


Fig. 4 (left). Motor torque versus tape speed for several command states

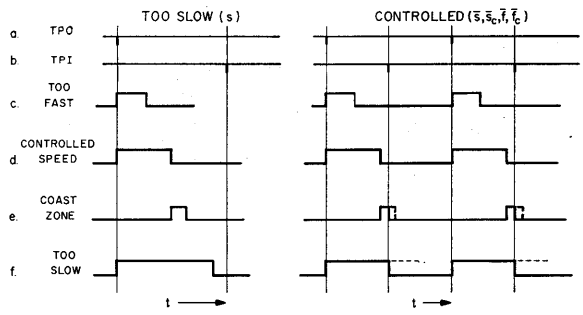


Fig. 6 (right). Speed - sensing waveforms

is minimized. There is no wrap around the head. Variations in tape tension, which are large in this transport, do not, therefore, cause excessive pressures on the head and wear is reduced. Because only short wave lengths (0.0025 and 0.005 inch) are used in the system, the area of tape-head contact need not be large.

The direction in which the transport is moving is determined by a sensing device mounted on the rear shaft extension of one motor. The sensor consists of an iron cup dragged against one of a pair of stops by hysteresis from a star-shaped permanent magnet on the motor shaft. The cup operates a mercury switch by rotating an attached magnet. This scheme gives positive direction information even at the slowest tape speed. A mercury wetted contact switch provides computer-level signals to the control without contact bounce and with good reliability.

Motion Control

Each motor can generate torque in only one direction to pull the tape from one reel to the other. The control of the motors is therefore simpler than if torque had to be reversed. Since tension is limited by tape strength, acceleration is relatively slow. A sudden change of torque, which might allow a loop to form, is prevented by a long time constant in the control windings of the motor magnetic amplifier.

To stop the tape, full torque is first applied by the trailing motor until the tape speed falls below 20 ips: at that point d-c is applied to the trailing motor to bring the tape to a smooth stop. The direction sensor indicates which motor is trailing. With d-c in the motor field winding the rotor will resist applied torque even at zero velocity due to the hysteresis in the rotor. Voltage is never completely removed from either motor in order that some tape tension always be maintained. The end of the tape is sensed by a photoelectric cell which receives light through transparent leaders at each end of the tape. The timing track is continued on the edges of the 100-foot transparent strips so that the control element will know when the tape has fallen below 20 ips as previously described.

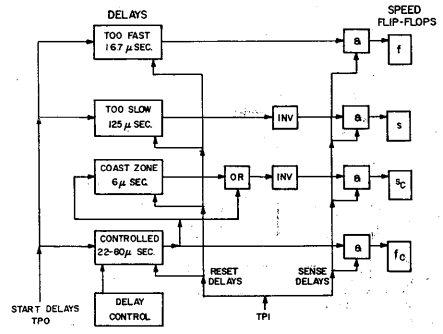


Fig. 5. Speed-sensing logic

The control circuit for one of the motors is shown in Fig. 3. The transistor, Q-1, regulates the current through the control windings of the 3-phase magnetic amplifier, and it switches between saturation and cutoff at various duty cycles. When Q-1 is cut off, D-1 conducts, so that the control winding-time constant is determined solely by its own inductance and the 470-ohm resistor. This time constant is made long enough to prevent abrupt torque changes and to average the control current.

Feedback was included to provide close control of minimum torque. Too much minimum torque will either allow the tape to creep or require excessive d-c holding currents in the trailing motor. Too little torque will fail to overcome static friction, and allow a loop of tape to form. The feedback prevents large variations in the output current of the magnetic amplifier which would be caused by unbalanced line voltage or small variations in reactor control current, especially when the amplifiers are nearly cut off. The feedback signal is derived from the sum of currents in all three motor leads. The diode D-2 limits the sum output voltage of the current transformers to 10 volts and thus keeps the voltage drop across their primaries negligible under high current conditions. To generate full torque, one of the other transistors such as Q-3 is saturated, cutting off the magnetic amplifier control current independent of the feedback and allowing full current to flow to the motor.

The direct current which is applied to the trailing motor when the transport is slowing to a stop is switched to one lead of the motor by a relay contact (not shown on Fig. 3). The d-c flows into that motor lead and out the other two, back through the magnetic amplifiers. Although the magnetic amplifiers are biased into a low torque condition they will pass the d-c (approximately 1 ampere) since the average voltage across any one reactor must be zero.

Digital Speed Control

To determine which motor should receive full torque, minimum torque, or d-c, the desired condition of the transport is compared with the existing one. As shown in Fig. 4, the motion control is based on a group of speed domains: too fast (f), faster than controlled (f_c), slower than controlled (s_c), and too slow (s). Various torque commands are shown as a function of speed and direction for several desired conditions.

The speed sensing logic is diagrammed in Fig. 5. The speed is detected by comparing the interval between timing pulses from the tape with the delays of delay units, as shown in Fig. 6. Two pulses are generated from a tape timing channel as it travels over the head, Fig. 6(A) and (B). The first is used to fire three delay units, two of which, Fig. 6(C) and (F), establish the boundaries between the area of usable speeds and too fast or too slow. The third delay unit, Fig. 6(D), can be set by the computer to any one of several delay times representing speeds in the useful range and provides close speed control at preset and selectable tape speeds. It in turn drives a fourth delay unit, Fig. 6(E), to provide a controlled zone. In this condition the transport coasts. The second timing pulse occurs at a time determined by the speed of the tape. It is used to sense the condition of

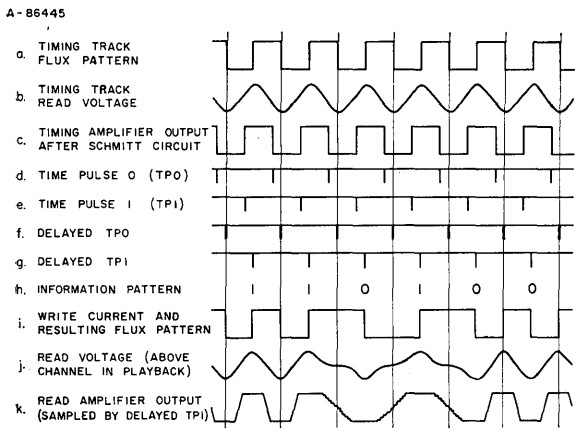


Fig. 7 (left). Phase-modulated nonreturn-to-zero waveforms

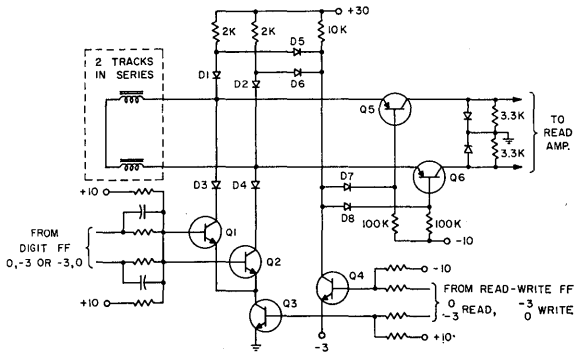


Fig. 8. Read-write switch and write circuit

the delay units and to set flip-flops to define the speed domain. It is also used to reset the units so they have time to recover before the next "set" pulse. The dotted waveforms in Fig. 6(E) and (F) indicate this resetting when the tape is in the controlled-speed zone. The delay units must be resettable, and the delay time of one must be capable of electronic variation.

Head Assembly and Read-Write Method

Head Assembly Design

The 10-track head assembly contains five channels: three information, one timing, and one block mark. Each channel consists of two redundantly paired tracks; the tracks in each pair are nonadjacent to minimize the effect of a speck of dirt lifting a portion of the tape. The timing channel occupies the two outside tracks which are heavily shielded from the interior ones in order that the timing channel may be read while the others are being written.

The timing channel controls tape speed, information density, and the fixed address feature. It assures constant information density regardless of tape speed and makes possible the changing of a single word in a message. Its density must therefore be known and constant. It is permanently recorded, either with a constant-speed capstan temporarily attached or on a separate constant-speed machine.

Read and Write Principles

Since the amplitude of the signal from the tape varies greatly with speed, a system of recording is used that allows the polarity of the flux change to be significant rather than its amplitude. High-gain amplifiers may then be used in which the gain need not be constant. Fig. 7 shows the flux pattern and other waveforms. The idealized timing-track flux pattern consists of 200 complete cycles of flux per inch, or 400-flux reversals per inch, see Fig. 7(A). The timing track read voltage, Fig. 7(B), is the expected derivative waveform from a 0.0005-inch gap looking at a signal of this density. The signal is amplified and squared in a Schmitt circuit, Fig. 7(C); the finite hysteresis of the Schmitt circuit delays the signal slightly as shown. Time pulses are generated from the transitions of the Schmitt circuit; time pulse 0 (TP0) from the negative transitions and time pulse 1 (TP1) from the positive transitions, 7(D) and 7(E). The time pulses must then be slightly delayed, 7(F) and 7(G), so that the information flux pattern may be written in phase with the timing flux pattern. The delay is a function of tape speed and is varied by an analog voltage fed to the delay circuit. The analog voltage is in turn derived from a circuit whose output is a predetermined function of the average frequency of the time pulses fed to it. The flux is laid on the tape in phase with the timing flux so that information may be read or written while the tape is moving in either direction.

The delayed time pulses control the transfer of information to the writing flip-flops. Delayed TP0 transfers the bit to be written to the flip-flop which is controlling write current; delayed TP1 complements the flip-flop. Thus a flux change is written in the center of each line corresponding to the bit to be written; there may or may not be flux changes between the lines. A typical information pattern and resulting ideal flux pattern are shown in Fig. 7(H) and 7(I). The voltage which would be read from this channel during read time is shown in Fig. 7(J). Notice that there is a saturation signal at the center of each line, whereas, in between lines there is sometimes a signal and sometimes not. The signal is then amplified more than necessary, Fig. 7(K). The amplifier has enough gain so that one of the redundant tracks may be completely separated from the head by a speck of dirt while a half-amplitude signal is being received from the other track; a saturation signal will still be delivered by the amplifier at the center of the line. The amplifier is strobed by delayed TP1, so that the logic doesn't know what the amplifier output looks like at any other time. The saturation output received at the center of each line with phase-modulated nonreturn-to-zero recording also allows the tape to be read correctly with plenty of amplifier gain margin over a wide range of tape speeds.

Read and Write Circuits

The read-write switch and write circuit for one digit are shown in Fig. 8. During "write," $Q4$ is cut off and $Q3$ is saturated. With $Q4$ cut off, its $10K$ collector resistor lifts the bases of $Q5$ and $Q6$ towards $+30$, leaving them cut off and the read amplifier disconnected. The silicon diodes at the amplifier input prevent any large voltage excursions from reaching the amplifier. With $Q3$ saturated, the digit flip-flop will cause either $Q1$ or $Q2$ to also be saturated. With the circuit values shown, 15 milliamperes (ma) will flow through the two series-connected tracks and 30 ma through the saturated transistor ($Q1$ or $Q2$). The direction of current flow through the tracks is determined by the flip-flop state; i.e., whether $Q1$ or $Q2$ is saturated.

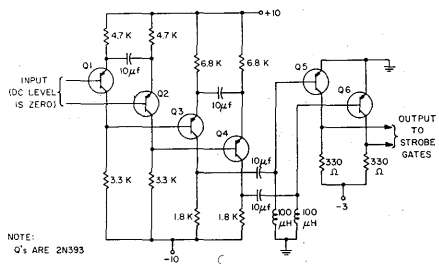


Fig. 9. Read amplifier

During "read," $Q3$ is cut off and $Q4$ is saturated. $Q4$ takes the full write current through diodes $D5$ and $D6$, back biasing diodes $D1$ through $D4$. With $Q4$ saturated diodes $D7$ and $D8$ are back biased, allowing $Q5$ and $Q6$ to be saturated, thus connecting the two series-connected tracks to the read amplifier at a d-c level of approximately zero.

Read Amplifier

The read amplifier, Fig. 9, has two difference-amplifier stages and one output stage with more than enough gain to give a saturation output signal at a tape speed of 20 ips. In the first two stages, the common mode gain per stage is less than unity while the difference signal gain is approximately beta. The low common mode gain insures that power supply noise will not be amplified. Each transistor ($Q1$-$Q4$) is biased to a constant d-c operating point of approximately 3.8-smitter-collector volts and 1.9-ma collector current. The capacitors shown must only be large enough to have negligible signal attenuation at 20 ips, the lowest tape speed of interest. The lowest frequency signal will be at 20 ips and 100 cycles per inch [alternate ones and zeros; see Fig. 7(J)] for a frequency of 2 kc. The highest frequency will be at 920 ips and 200 cycles per inch (all ones or all zeros) for a frequency of 184 kc. The micro-alloy *2N393* transistors have more than enough bandwidth for this application. The signal amplitude at the input to $Q5$ and $Q6$ is large enough so that, most of the time, one of these transistors is saturated. The output signals are of a computer-type amplitude (0 or -3) and are sampled by TP1 using conventional *TX-2* logical circuits.[4]

References

1. Functional Description of TX-2 Computer, **J. M. Frankovich, H. P. Peterson.** *Proceedings of the Western Joint Computer Conference,* Institute of Radio Engineers, New York, N. Y., 1957, pp. 146–50.

2. The Lincoln TX-2 Input-Output System, **J. W. Forgie.** *Ibid.,* pp. 156–60.

3. Memory Units in the Lincoln TX-2, **R. L. Best.** *Ibid.,* pp. 160–66.

4. Transistor Circuitry in the Lincoln TX-2, **K. H. Olsen.** *Ibid.,* pp. 167–71.

The Dynamics of Toggle Action

NORMAN L. KREUDER
NONMEMBER AIEE

IT IS common in the design of regenerative circuits to perfect the d-c (static) design on paper and the a-c (dynamic) design in the laboratory. In the process of working out the static design, a certain amount of engineering judgment can be used so that the resulting circuit will give approximately the required dynamic performance, but rarely will the d-c-designed circuit pass all the performance specifications without some changes dictated by laboratory tests.

Although it is awkward to calculate directly the effects of loading and component variation on stability, switching time, and triggering characteristics, it is even more awkward to measure these effects experimentally because interaction of all the components makes it difficult to find the worst combination.

This paper presents a design method using an intermediate step, i.e., the negative resistance curve, between the static and dynamic design. The effects of loading and component variation show up clearly as changes in the negative resistance curves and, in turn, the altered dynamic performance can be calculated easily from these curves.

The method is here applied only to toggle (flip-flop) design, but the extension to monostable circuits in simple. In principle, the method is applicable to other regenerative circuits such as blocking oscillators.

The paper includes the derivation of negative resistance curves from the circuit parameters, a method of evaluating toggle performance from the curves, and an example in the form of a transistor toggle.

Description of the Curve

Fig. 1(A) shows a typical toggle using p-n-p transistors. Suppose the resistor sizes have been chosen to produce some standard swing (V_0 to V_1) with some standard power supply voltages, (E_1 and E_2). The object is to evaluate this proposed design.

A variable voltage source is shown connected to one collector, with suitable meters for current and voltage measurements. If the source voltage is either V_0 or V_1, no current will flow since V_0 and V_1 are the stable, open-circuit output voltages. Some current will flow at other voltages.

When T_1 is conducting, and saturated, the current will be zero at a voltage very close to zero. If T_1 's collector is forced more negative, current will rise rapidly along the saturated collector characteristic, (R_s). This is shown as A, B in Fig. 1(B). Since collector current cannot exceed $\beta_1 i_b$, where i_b is the base current, T_1 will pull out of saturation when the collector current reaches $\beta_1 i_b$.

The measured current will continue to rise, but more slowly, (B,C) and the slope is now that of R_1, ($R_2 + R_3$), and the grounded-emitter collector resistance of T_1, all in parallel.

Up to this point, T_2 has been cut off, but as the collector voltage of T_1 rises, so does the base voltage of T_2. When T_2 begins to conduct, the circuit behavior alters radically. The current through T_1 is reduced as the collector voltage of T_2 drops, so less current, rather than more, is required from the external source. In fact, as the voltage is increased further, the current required drops to zero and reverses. (C,D,E on Fig. 1(B).) When T_1 is completely cut off, the current again begins to rise, and the slope is now R_1, R_2, and a very large collector resistance all in parallel. At $E_F = V_1$ the current crosses the axis and will continue to rise linearly until breakdown.

This negative resistance (NR) curve describes completely the output characteristics of the circuit, just as the plate curves describe a tube. Load lines can be drawn that will indicate loading capability, as will be shown later. In addition it will be shown that the NR curve is an aid in deducing triggering thresholds (duration and amplitude) and switching time.

Derivation of the NR Curve from the Circuit Parameters

It is not necessary to make the previous measurement to arrive at the NR curves for a circuit. Consider Fig. 2(A). Assuming the base swing is small compared to E_2, the current through R_3 (i_3) is nearly constant at E_2/R_3. When T_1 is saturated, its collector current (i_c) is $(E_1/R_1) - i_3$. The drop across the right

Norman L. Kreuder is with Burroughs Corporation, Pasadena, Calif.

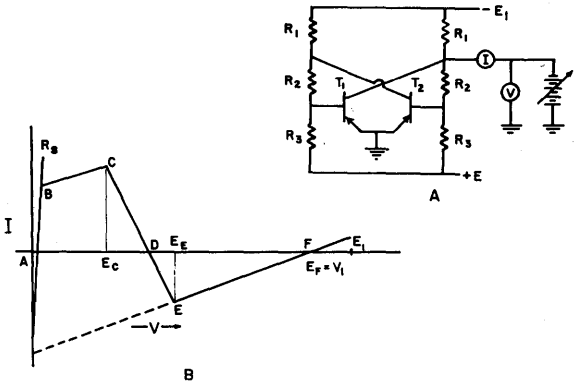


Fig. 1 (left). Typical toggle and negative resistance curve

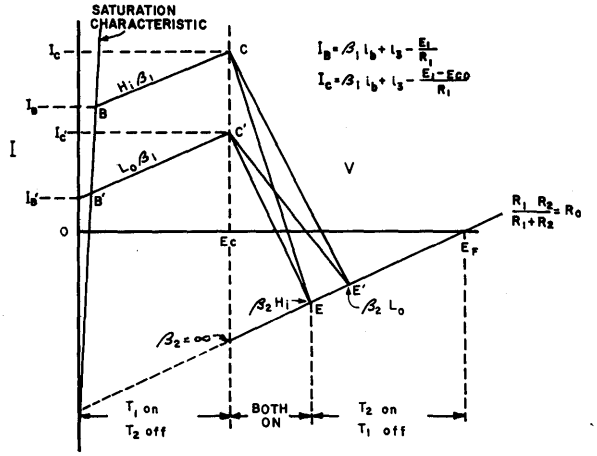


Fig. 2B (right). Resulting NR curve

hand R_2 (E_{co}) is then i_3R_2 and that is the amount by which T_2 is cut off. The collector voltage of T_2 is R_2E_1/R_1+R_2, neglecting the base-emitter drop of T_1. The base current (i_b) of T_1 is $(E_1/R_1+R_2) - i_3$. The collector current of T_1 must be less than βi_b in order for T_1 to be saturated.

The foregoing paragraph must be obvious to anyone who has ever gone through the d-c design of a toggle. It serves only to define a few terms.

Again imagine the collector voltage of T_1 to be forced negative by some external device. The external current will rise steeply along T_1's saturation curve until the collector current is $\beta_1 i_b$. (The rise to I_B is actually a little steeper than this, as R_1 and (R_2+R_3) are in parallel with T_1's saturation curve.) If, at this point, the collector voltage is still low, a good approximation for alloyed transistors, the current that the external device must insert is T_1's collector current plus i_3 minus the amount inserted through R_1. That is, $I_{ex}=\beta_1 i_b+i_3-(E_1/R_1)$. This current is noted as I_B on Fig. 2(B). The subscript B corresponds to the point B on Fig. 1(B). A positive I_{ex} will be taken to mean an electron current into the toggle circuit.

For external currents greater than I_B, the collector voltage increases more rapidly. T_1 is then in the linear region, so the slope B, C is r_c/β_1 in parallel with R_1. Another parallel path is (R_2+R_3) but it is usually large compared to R_1. Up to point C, T_2 has been cut off but when the collector voltage of T_1 reaches i_3R_2, T_2 begins to conduct. Notice that this voltage is independent of β, so that $E_c=E_c'$. The external current at this point, $I_c=\beta_1 i_b+i_3-(E_1/R_1)+(E_{co}/R_1)$, which is just like I_B except for the last term.

It is simpler now to skip to point E_F. Since here T_1 is cut off, r_c is large, and the slope through E_F is simply R_1 in parallel with R_2. E_F is just R_2E_1/R_1+R_2.

Starting at E_F and reducing the voltage of the external source, the external current will be determined by the resistance $R_1R_2/R_1+R_2=R_o$. As the voltage is reduced, the base current of T_2 will be reduced, and T_2 will come out of saturation. When the collector voltage of T_2 has risen to E_{co}, T_1 will again begin to conduct because its base voltage will have risen to ground. The current in T_2 when T_1 is on the verge of conduction is then $I_v=(E_1-E_{co}/R_1)-i_3$ and this quantity divided by β_2 is the base current required in T_2. The current through the right hand R_2 will be $(I_v/\beta_2)+i_3$ and the required voltage on T_1's collector is:

$$\left(\frac{I_v}{\beta_2}+i_3\right)R_2=\left[\frac{E_1-E_{co}}{\beta_2R_1}+\frac{i_3(\beta_2-1)}{\beta_2}R_2\right]$$

For very large β_2, this voltage approaches i_3R_2 which is also E_c.

Notice that, beginning at zero volts, the point reached, C or C', depends only on β_1, and, beginning at E_f, the point reached, E or E', depends only on the term β_2.

The range of voltage C-E is the active region, and, if the β's are constant here (a good assumption), and T_2's base-emitter drop is constant (a poor assumption), the lines C-E will be straight. It is easy to take into account the effect of T_2's base-emitter voltage, as will be seen later.

An additional useful bit of information is I_E, the maximum current the external source can remove from the circuit without preventing bistability.

$$I_E=\frac{E_F-E_E}{R_o}=\left(\frac{E_1R_2}{R_1+R_2}-\frac{R_2(E_1-E_{co})}{\beta_2R_1}-\frac{R_2i_3(\beta_2-1)}{\beta_2}\right)\frac{R_1+R_2}{R_1R_2}$$

$$=\frac{[\beta_2R_1-(R_1+R_2)][E_1-i_3(R_1+R_2)]}{R_1^2\beta_2}$$

Evaluation of Circuit Using NR Curves

Driving Capability

The circuit margin with respect to β changes can be seen on the NR curves since the points of circuit stability are the points where the NR curves cross the load line with positive slope. The unloaded case is a special one in which the load line is a zero-current line and therefore coincides with the horizontal axis.

Fig. 3 shows two load lines on a typical pair of NR curves. R_A is returned to some supply voltage E_A. R_B is returned to ground. These would be typical of "and" and "or" gates in computer circuitry. For the low β_1 curve, the toggle is not bistable with load R_A, since there is only one point of intersection of the load line and the NR curve. With load R_B, the toggle is bistable for both low β_2 and high β_2. Marginal load lines would pass through C or C' and E or E'.

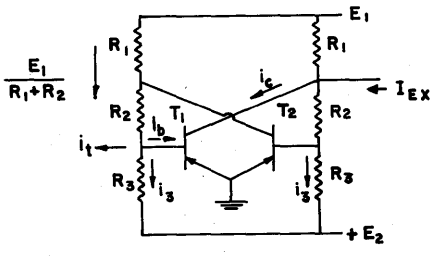


Fig. 2(A). Circuit for analysis

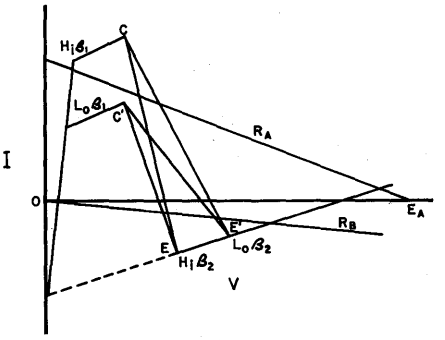


Fig. 3. NR curves with typical load lines

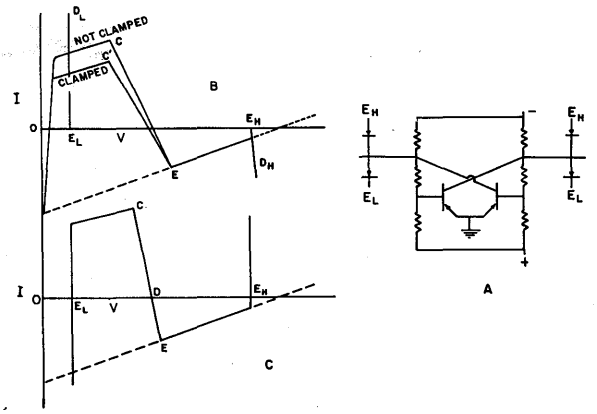


Fig. 4. Two representations of the effect of clamps

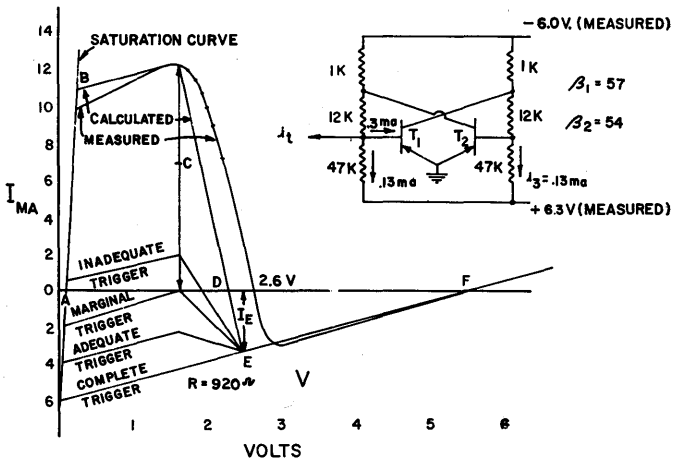


Fig. 6. Calculated and measured NR curves for a typical circuit

Usually, other considerations come into play before lack of bistability becomes a factor. A certain tolerance on output voltage must be maintained. When this tolerance is specified, a load line can be constructed to produce limit voltages and the minimum load resistance thus determined.

Clamps are a special case of resistive loading. Their effects can be shown as in Fig. 4, where D_H and D_L standardize the output voltage at E_H and E_L. D_L may not be necessary for standardizing output swing, but is often used to prevent saturation in the transistors. Since T_2's collector is not allowed to go to E_F, T_1's base current is reduced. I_c is then less than in the unclamped case. An alternative display of the effect of clamps is Fig. 4(C), where the diodes have been considered as part of the toggle, rather than as part of the load. The NR curve has been reshaped by adding together the diode current and the NR current at each output voltage. Similarly, built-in load resistors can be included in the toggle's NR curve by adding the resistor current to the NR current at each voltage. (When adding, it must be remembered that the load resistances were plotted as negative when they were loads, and must be reversed in sign when they are to be considered as part of the toggle.)

D-C Trigger Threshold

In many cases it is advisable to trigger a toggle by turning "off" an "on" transistor rather than vice versa because the trigger pulse is applied to an active (conducting) element. That type of triggering will be covered in this paper. Triggering "on" is a straightforward application of the same techniques.

Refer to Fig. 5. Suppose the trigger is to be applied to the base of T_1 (which is conducting). Current in T_1 must be reduced till its collector rises to E_{co}, the amount by which T_2 was cut off. At this point T_2 comes into conduction and helps cut T_1 off. The current in T_1 at the verge of regeneration is $\beta_1 i_b = (E_1 - E_{co}/R_1) - i_3$. The base current $i_b = (E_1/R_1 + R_2) - i_t - i_3$. Solving for i_t (minimum), $i_t = (E_1/R_1 + R_2) - i_3 + (i_3/\beta_1) - (E_1 - E_{co}/\beta_1 R_1)$ which is seen to be I_c/β_1. Therefore, the minimum trigger current is just $1/\beta_1$ times the height of the NR curve above the axis. If $\beta_1 i_t > I_c$ there will be only one point of intersection (point F) and the toggle will flip to that state and stay there when the trigger is removed.

The effect of triggering can be shown by reshaping the NR curve to take into account the reduced base current. Fig. 5(B) shows an NR curve with several different triggering levels.

Pulse Trigger Threshold

Because of capacitive loading and finite current rise times in transistors, one can expect a higher trigger threshold when triggering a toggle with short pulses. This threshold is difficult to specify in general, but for any particular case, graphical methods will produce a good answer.

The same statements apply to switching time and waveform for different triggering levels. The numerical example to follow will illustrate all the aforementioned techniques and also demonstrate the method for solution of the waveform and threshold problems.

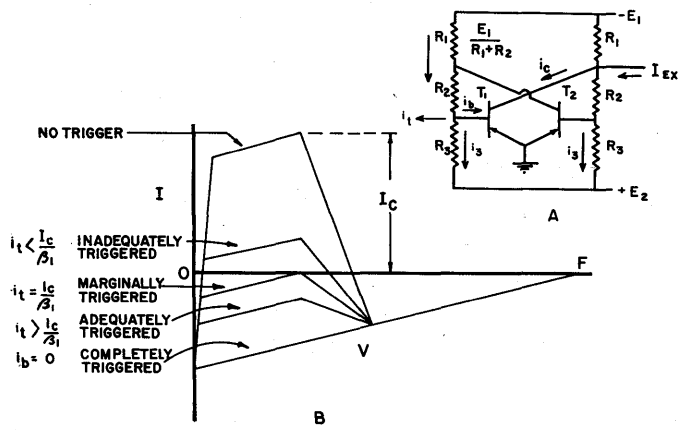


Fig. 5. NR curves showing the effect of trigger current

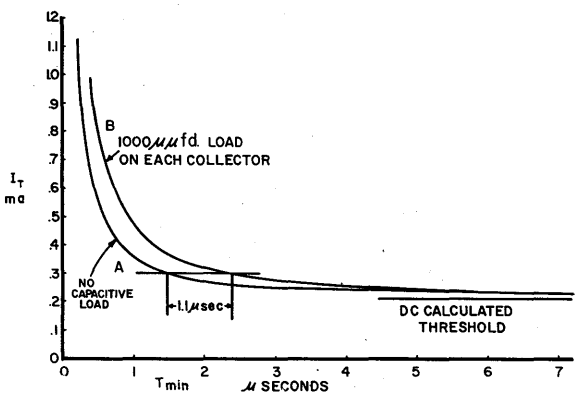


Fig. 7. Threshold trigger current versus trigger time, with and without capacitive load

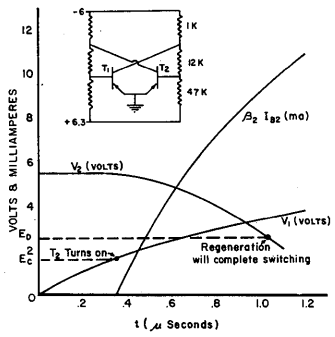


Fig. 8 (left). Toggle transient and essential calculations

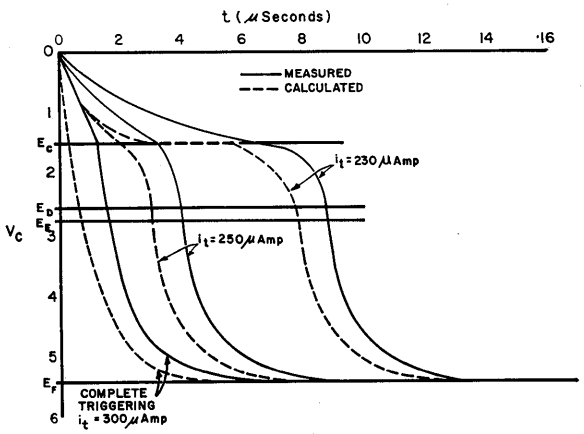


Fig. 9B (right). Collector waveform for three values of trigger current

$$i_{b2} = \frac{E_H}{R_2}(1-e^{-\frac{t}{RC}}) - \frac{E_2}{R_3}$$

$$I_{c2} = \beta_2 i_{b2} = \beta_2\left[\frac{E_H}{R_2}(1-e^{-\frac{t}{RC}}) - \frac{E_2}{R_3}\right]$$

$$I_{c2}(S) = I_1 + I_2 \quad I_1 = \frac{V_{c2}}{R} \quad I_2 = V_{c2}SC$$

$$I_{c2}(S) = \beta_2\left[\frac{E_H \frac{1}{RC}}{R_2(S+\frac{1}{RC})S} - \frac{E_2}{R_3 S}\right]$$

$$V_{c2}(S) = \frac{I_{c2}(S)}{C(S+\frac{1}{RC})} = \frac{\beta_2}{RC^2}\left[\frac{E_H}{R_2 S(S+\frac{1}{RC})^2} - \frac{E_2 RC}{R_3 S(S+\frac{1}{RC})}\right]$$

$$V_{c2}(t) = \beta_2 R\left[\frac{E_H}{R_2} - \frac{E_2}{R_3} - \left(\frac{E_H}{R_2}(1+t/_{RC}) - \frac{E_2}{R_3}\right)e^{-t/RC}\right]$$

On the NR curves, Fig. 1(B), D is a point of unstable equilibrium, as the load line (in this case the horizontal axis) crosses the NR curve in a region of negative slope. If the circuit is brought near point D by an external force and then released, it will regenerate to point A or point F, depending on which side of D it was on when released. To inquire as to what will happen if the starting point is exactly D, one must consider ever-present noise, and similar practical difficulties. Toggles will not stay at point D for the same reason that pencils will not balance on their points. The important point is that for successful triggering, the circuit operating point must pass E_D by the time the trigger has gone away. Otherwise, the circuit will return to its starting point.

Example

Fig. 6 shows a typical toggle circuit and its measured and calculated NR curves. The disagreement at low voltage, point B, is due to the erroneous assumption of constant β at low voltage and the neglect of r_c. The offset in the negative resistance region is due to the fact that the base-emitter drop of T_2 must be added to the calculated straight line CDE. This results in excellent agreement with the measured curve. The calculated threshold trigger current is:

$$I_c/\beta_1 = 12.3/57 \backsimeq 215 \text{ microamperes}$$

The measured threshold trigger current is 220 microamperes. Note that the "complete triggering" current is 300 microamperes. That is the trigger current that reduces T_1's base current to zero.

The current available for the external load at any voltage is the height of the NR curve at that voltage. If the load is largely capacitive, the available current can be integrated to find the time required to turn on T_2. Knowing T_2's base current and β, T_2's collector current can be integrated into its load capacity until the collector voltage changes enough to supply the threshold current to T_1. At this time the trigger can be removed and the regeneration can continue. Before this time has elapsed, the toggle will resume its former state if the trigger is removed. To this threshold time must be added the "built-in" delays in the transistors. If tubes are used, these built-in delays can be neglected.

Fig. 7 shows the trigger amplitude versus duration threshold for the typical toggle. Curve A indicates the magnitude of the built-in delays in these particular transistors. Curve B shows the similar curve with 1,000 $\mu\mu$fd on each collector. The difference at i_t=300 microamperes is 1.1 microseconds. Note the agreement between measured and calculated low-speed threshold.

Three hundred microamperes represents complete triggering in this case, since that is T_1's base current. Above 300 microamperes, the trigger is removing stored carriers from T_1's base, and this speeds up the transistors considerably. (Refer now to Figure 8A). We will now calculate the additional trigger duration required when each collector is loaded with 1,000 micromicrofarads ($\mu\mu$f). The point of unstable equilibrium (E_D) is 2.6 volts in this circuit, so that is the voltage that T_2's collector must descend to before the trigger is removed. The sequence of events can best be shown graphically, as in Fig. 8(A). At $t=o$, the current in T_1 is reduced, and the collector voltage increases as the load capacitor charges through R_0 (920 ohms). This voltage asymptotically approaches −5.5 volts in the case of complete triggering. When the collector voltage of T_1 exceeds 1.6 volts (E_{co}), T_2 begins to conduct, and its current increases with the same waveform that appeared on T_1's collector, but with the obvious time displacement. The collector current of T_2 flows into a load consisting of R_0 and 1,000 $\mu\mu$f in parallel. Eventually the collector voltage of T_2 will fall to E_D (−2.6 volts), and regeneration will take over. At that instant, the trigger current can be turned off and regeneration will complete the change-of-state. Fig. 8(B) shows the essential steps in the calculation of minimum trigger time for the case of complete triggering. Note that V_{c2} in Fig. 8(B) represents the change in T_2's collector voltage from the initial state of −5.5 volts. The change

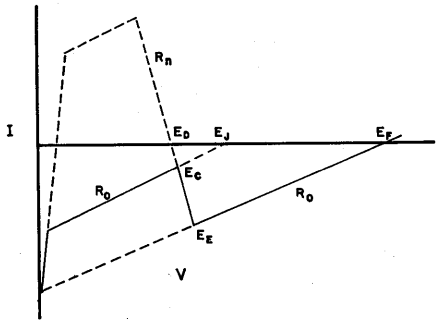


Fig. 9(A). NR curve for adequate triggering

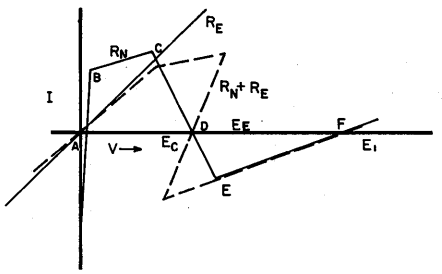


Fig. 10. Alteration of NR curve due to common emitter resistor

required is $5.5-2.6=2.9$ volts. Upon substituting, it is seen that V_2 reaches 2.6 volts after a delay of 1.07 microseconds, and this figure compares favorably with the 1.1 microsecond found experimentally and illustrated previously in Fig. 7.

Another good check point is near the point of marginal triggering. Notice the NR curve for adequate triggering in Fig. 9(A). The current is of the form (E_J/R_0) $(1-e^{-t/RC})$ between $E=0$ and $E=E_c$. The voltage on the capacitor is obviously a simple RC rise. From E_c to E_E there is another RC rise, but here R is negative, so the exponent is positive. In a sense, the current "approaches" E_D/R_n as $t\rightarrow-\infty$. Above E_E, the rise is again a simple RC, approaching E_F. For triggering levels just over the margin, the waveform on T_1's collector should resemble the three exponentials. Fig. 9(B) shows the actual and predicted waveforms for two values of triggering current just over the minimum and for complete triggering.

The Effect of a Common Emitter or Cathode Resistor

A common emitter resistor affords another path of regeneration and can therefore be expected to alter the shape of the NR curve. Fig. 10 illustrates this effect. The voltage drop across R_E, the common emitter resistor, has been added to the active circuit voltage and the NR curve of the composite circuit is shown dashed. It is assumed that the emitter return voltage is changed along with R_E in such a fashion as to keep the emitter current the same as before at points A and F.

When $R_E=-R_N$, the composite NR curve becomes vertical in the regenerative region. For larger R_E, the slope reverses in this region. It may seem surprising to have three consecutive intersections of the NR curve and the horizontal axis in which the NR curve has positive slope, and the casual observer may be led to believe that there are three points of stable equilibrium. Intersections with positive slope, however, are points of stable equilibrium only in systems that are open-circuit bistable, and in cases where $|R_E|>|R_N|$, it is both open-circuit and short-circuit bistable.

One was able to explore the NR curve experimentally when R_E was small because the constant-voltage source constituted a vertical load line. That is, there was only one intersection of the (vertical) load line and the NR curve. But for large R_E, the current is no longer a single-valved function of voltage and the circuit is both open-circuit and short-circuit bistable. When R_E is bypassed with a capacitor, the behavior becomes more complicated. For short input pulses, R_E can be ignored, but for d-c loading, it must be taken into account.

Special Considerations When Using Tubes

For tube circuits, the NR curve is not so easy to find analytically. The experimental method in the beginning of this paper can be used, although it is slow. The power supply can be replaced by a transformer, though, and the NR curve traced at a 60-cycle per second rate on an oscilloscope. Connect the plate voltage to the horizontal amplifier and insert a metering resistor to derive a current signal for the vertical amplifier. Then, the effects of variations in tubes, resistors, or voltages can be seen immediately. Since tubes go into or out of conduction more gradually then transistors, the corners of the NR curve will be rounded. This could be an advantage in analysis because the NR curve may be well represented by the simple cubic $I=K_1(E-E_D)+K_2(E-E_D)_1^3$ where E_D is again the point of unstable equilibrium.

Conclusions

The usefulness of NR curves in toggle analysis has been demonstrated, and a method has been given for the derivation of such curves from the circuit parameters. The next step, apparently, is to turn things around and use the NR curves as the intermediate step in toggle synthesis. Given the requirements, output voltage and tolerance, load driving requirements, etc., it should be possible to draw an NR curve that fits the requirements, and then to derive the component values and tolerances directly from the NR curve. Whether this can be done in practice remains to be seen.

Direct Access Photomemory

Part I. Prototype Machine System

F. A. LITZ
NONMEMBER AIEE

IN RECENT years, the International Business Machines Corporation (IBM) research laboratory at San Jose, California, has sustained a group effort directed toward the investigation of a very-large capacity direct-access digital storage system using a unique self-developing photographic medium.

A feasibility model, designed around this film medium, with a 10^9 alpha-numeric character capacity, has been completed. It was under taken purely as a research project to explore the capabilities of this medium, and there are no present plans to produce it. This machine, the Direct Access Photomemory, was used as a carrier to determine the problems associated with the radically different photo file as opposed to the more familiar magnetic systems. Early in the machine's development, design parameters were established with the primary purpose of forcing advanced technological development in all phases of the work. These were as follows:

Capacity: 10^9 alpha-numeric characters, consisting of 10^7 100-character records
Storage density: 1,000 bits per lineal inch, equivalent track spacing of 0.006 inch
Access time: one second maximum
Input: 100-kc bit rate, asynchronous
Output: 100-kc bit rate, asynchronous

Direct Access Photomemory

A description of the Direct Access Photomemory will reveal little resemblance in basic components to magnetic storage systems. Light sources, lenses, and a transparent film replace the more familiar magnetic head and oxide coated tape, disk, or drum surfaces. One of the major differences encountered from the point of view of system is the char-

F. A. Litz is with International Business Machines Research Laboratory, San Jose, Calif.

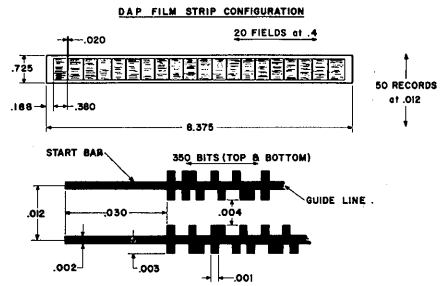


Fig. 1. Film strip organization

acteristic nonerasability of the film used. This makes the photo file a ledger book where updating of records is accomplished by subsequent recording of revised data, thereby providing an inherent audit trail.

The machine configuration will be discussed in five parts: 1. the film medium; 2. film-strip organization; 3. file organization, film strip selection, and transport mechanisms; 4. input shutters and recording optical system; and 5. the output scanning system.

Film Medium

An ideal photo file film would have the following specifications: it should be dry developing and erasable; speed of recording and erasing on the order of microseconds; spectral sensitivity outside the range of the visible spectrum to avoid inadvertent exposure; exposed area absorption characteristics peaking in the visible to permit readout with available light sources and photo detectors; excellent stability; good mechanical strength and hard surface coating; and high resolution.

The film medium adopted, referred to in this discussion as Chalkley Film, after the inventor Dr. E. Chalkley, fundamentally incorporates all of these requirements to a greater or lesser degree with the exception of erasability and microsecond response. Chalkley Film consists of an organic dye, chemically classified as a pararosaniline leuconitrile, coated in a gel solution on a clear cellulose acetate or Mylar substrate. Unactivated, the compound is transparent and insensitive to visible light. When exposed to ultraviolet radiation between 2,400 and 3,200 angstrom units, the dye turns purple with an absorption peak at 5,750 ganstroms. Thus, the film is exposed with ultraviolet radiation and data thus recorded may be scanned out with a visible light beam. Being an organic dye, film resolution is molecular and system resolution is limited by the optical system, coating imperfections, dirt and scratches.

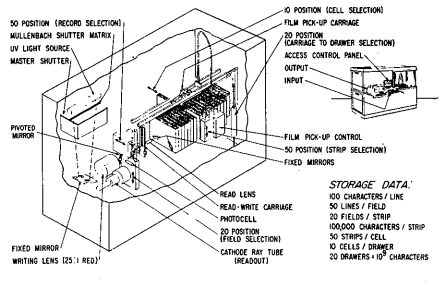


Fig. 2. Schematic of the feasibility model

Film Strip Organization

Film strip size, thickness, data organization, and density were fundamentally determined by a combination of system, mechanical, and optical considerations. These are basically: ease of addressing, registration tolerances, positioning accuracy, recording and reading optical components, mechanical strength required for high-speed film transport, and anticipated dust and dirt obliteration.

As shown in Fig. 1 each strip is approximately 8 by $^3/_4$ by 0.005 inches and contains 1,000 100-character records arranged in 20 columns of 50 records each. One or more columns normally would be considered a unit or ledger record. A 7-bit nonreturn-to-zero (NRZ) code was adopted. Bit size was established at 0.001 inch wide by 0.003 inch high with 350 bits recorded above and 350 below the 100-character record centerline. A 3-digit number, 000 to 999, defines each of the 1,000 address locations on the strip.

File Organization, Film Strip, and Transport Mechanisms

Numerous file organizations were considered in an effort to provide maximum flexibility from a systems viewpoint, one that would be simple to fabricate and that imposed the least severe mechanical drive- and strip-transport mechanism specifications. The feasibility model file is essentially a rectangular tub file. It is capable of storing 10,000 strips containing a total of one billion characters. There are 20 rectangular drawers which can be positioned under the access mechanism. Each drawer holds 10 removable cell modules of 50 filmstrips each. The strips are 0.020 inch on centers and a loaded cell weighs 4 ounces. The file bin dimensions are approximately 30 inches long by 12 inches wide by 10 inches deep.

As shown in Fig. 2, the strips are located parallel to the drawer axis. A pickup mechanism moves on fixed ways perpendicular to the drawer axis. To select a strip, the appropriate drawer is positioned with that cell under the pickup mechanism way, and the pickup mechanism is simultaneously positioned over the drawer and the appropriate strip through a coarse-fine mechanism. The strip is physically removed from the cell and transported to the optical station where it is inserted in a carriage that in turn positions the strip to the one of a thousand possible record positions relative to the optical axis. After recording and/or reading operations are completed on a particular strip, the strip is returned to its original cell position.

The basis for the loosely held removable cell module lies in the elimination of the necessity for holding rigid precision fabrication tolerances over the entire file structure. Accuracy in strip selection is obtained by caging or detenting the loosely held cell module to the pickup carriage.

A second advantage of the removable cell module is the systems flexibility it provides by permitting cells to be interchanged between files or easily stored outside of the machine for historical purposes.

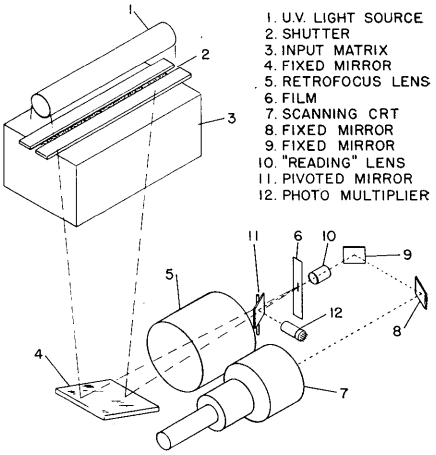


Fig. 3. Optical arrangement

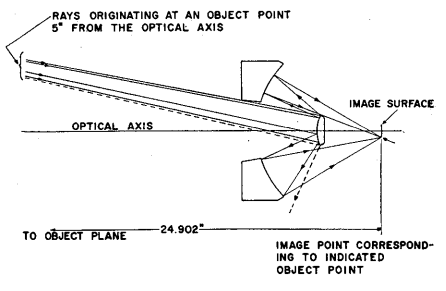


Fig. 4. Ray trace for f/s retrofocus lens

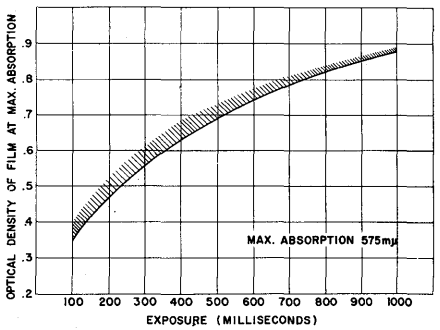


Fig. 5. Ultraviolet system recording speed

All drive mechanisms employed were open-loop, nonreset electrohydraulic devices capable of responding to direct binary control signals and positioning relatively large loads to discrete positions with a high degree of accuracy and speed.

Drive positioning accuracy requirements were as follows:

Optical station carriage
- Fifty-position line selection ±0.001 inch
- Twenty-position column selection ±0.005 inch

File
- Drawer: 10-position cell selection ±0.005 inch
- Pickup mechanism: 20-position drawer selection ±0.005 inch
- Pickup mechanism: 50-position strip selection ±0.003 inch

Input Shutters and the Recording Optical System

Data recording comprises photographing an electromechanical shutter matrix representation of the input data. Input data received in character groups, parallel by bit, serially by character, is buffered by the bit-shutter matrix composed of 700 individually actuated shutter elements.

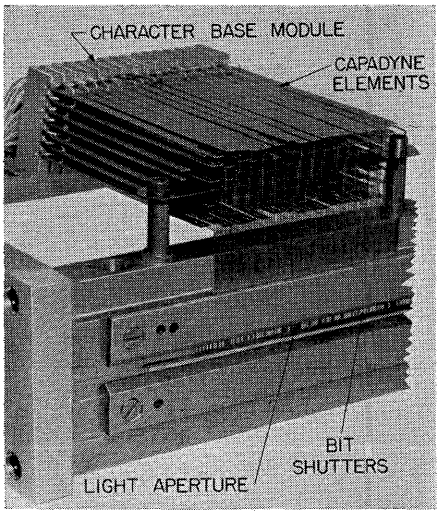


Fig. 6. Input shutter array

Design considerations relative to the optical layout shown in Fig. 3 are to some degree dictated by the geometry of the light source and the input shutter matrix.

The ultraviolet recording optics consist of a 2,000-watt high-pressure mercury-arc light source delivering approximately 200 watts in the region of 2,400 to 3,200 angstroms over its 12-inch length, and a 25 to 1-minification retrofocus lens system.

The recording lens, having no refractive elements, is free of chromatic aberration in the specified range and has a speed of f/2. Chromatic range specified was 2,400 to 3,200 angstroms and the lens provides a reduction of 25 to 1 of the approximately 10 by 0.5-inch shutter matrix object. Lens resolution was better than 100 lines per millimeter. Depth of focus for this system was established at ±0.0005 inch. Fig. 4 shows the ultraviolet lens-ray trace path. Fig. 5 shows the system exposure time as a function of optical density of the Chalkley Film.

The bit-shutter matrix consists of an array of miniature shutters, each 0.025 inch wide by 0.075 inch high, conforming to the specified 100-character record configuration. Each bit shutter is actuated by an electrostrictive driving element produced by the Mullenbach Company. In addition to providing the necessary force to actuate a bit shutter when charged with a data voltage pulse, the element provides a short time buffer to permit serial-to-parallel conversion of the input data and an asynchronous recording operation.

In the circuitry, the electrostrictive driving elements appear as approximately 0.03 microfarad capacitors requiring a voltage pulse on the order of several hundred volts for a desired 0.075 inch deflection. Fig. 6 shows a partial assembly of the input shutter array. Controlled film exposure is obtained by opening a main shutter between the ultraviolet light source and the bit shutter matrix for a preset exposure period.

Output Scanning System

Recovery of digital data is accomplished by scanning a 100-character record line with a visible-light spot provided by a flying-scanner cathode-ray tube. The record-modulated light is then detected by a multiplier phototube, the video shaped, decoded, and transmitted to the output system.

Although mechanical scanning was considered feasible, the flying-spot scanner is considered more desirable for two reasons. First, it is asynchronous and, second, there is no dead time between the notifying pulse and information output, such as exists in a drum storage system.

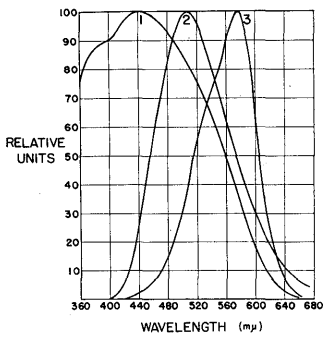


Fig. 7. Spectral characteristics of read-out components

1. Spectral sensitivity of S-11 photocathode
2. Spectral emission of P-24 phosphor
3. Absorption of hydroxy ethylated para-rosanaline

The scanning system employed, in conjunction with a data-synchronized clock, eliminates the requirement for extremely accurate positioning and registration that would normally be needed with a synchronous clock system. The latter system used with a recording density of 1,000 bits per lineal inch would require a combined film registration and positioning tolerance of ±0.00025 inch as opposed to the ±0.010 inch proved adequate for the Direct Access Photomemory system employing a data-synchronized clock system.

The spectral characteristics of Direct Access Photomemory readout components are shown in Fig. 7. An optimum design requires that the scanning light-source spectral distribution be matched to the absorption characteristics of the exposed film and the spectral sensitivity of the multiplier phototube. This match is imperative to optimize recording time since, for a given signal-to-noise ratio, a less dense exposure could be used for a matched rather than a mismatched system.

The scanning cathode-ray tube used was a Dumont *K1393P24* magnetic deflection tube. The spot size is on the order of 10 thousandths and appears as a one thousandth spot at the image plane after passing through a 10-to-1-minification lens system.

Signal-to-noise ratio was in excess of 5 to 1 for an optical density in excess of 0.65 for a "1010" pattern. Fig. 8 shows a typical phototube output. Aperture distortion is noted which results

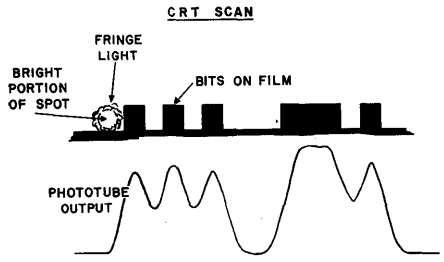


Fig. 8. Typical phototube output

from the finite spot size and a gaussian spot light distribution. It should also be noted that the d-c signal level varies with recording density in addition to variations in the information pattern when the spot size approaches the size of a single bit.

Operating Characteristics

All basic functions required in the Direct Access Photomemory file model, input, output, and, film handling, have been demonstrated with sufficient success to establish fundamental feasibility of a high-density random-access coded photo file, disregarding the originally specified access time. In all instances, positioning-accuracy specifications were met. The maximum cycle time to select a film strip from the 10^9-character file, record a 100-character record, read it back, and return the strip to its original cell position, was established at 2.75 seconds with a minimum time of about 1.6 seconds.

Remaining Problem Areas

Problems still to be resolved basically revolve about the film storage medium. Increased speed and stability are essential. Spectral characteristics of the exposed film require substantial improvement. The development of a suitable reversible film medium would considerably enhance this class of memory system. High-speed film handling mechanisms also require additional development.

Conclusions

The advantages gained by this machine development are primarily reflected in component-technology advances in the areas of high-resolution photography, self-developing high-resolution photographic recording media, high-speed precision positioning and film handling mechanisms, and flying-spot scanner systems.

Although there are no plans to incorporate the Direct Access Photomemory into a commercial business system in the near future, development of photographically based systems will play a big part in, and have a significant impact on tomorrow's new business machine technology.

Direct Access Photomemory

Part II. System Considerations

A. J. CRITCHLOW
NONMEMBER AIEE

THE Direct Access Photomemory resulted as a continuation of the International Business Machines Corporation (IBM) research interest in random access memories for data processing which began with the Random Access Memory Accounting Computer (RAMAC).[1] It was realized that large-capacity random access memories allowed the use of entirely new methods and techniques to drastically shorten data processing while providing valuable new services to the user. The model described in the first part of this paper was a research attempt into a new storage medium having the necessary characteristics.

Immediate access to stored information allows "in-line" as opposed to "batch" processing of business transactions, thus eliminating sorting and collating operations, and provides continuously updated records. If a large capacity file could hold all the records required by a company and maintain complete detail on several classifications, then the many separate reporting operations now required might be eliminated.

A preliminary study disclosed several business, industrial, and governmental applications where one-billion-character files could be used to handle the routine daily operations. Examples of these are: engineering change letter, parts requirements and status files in large aircraft plants, state automobile-license bureau records, policy status records (in large insurance companies), traveler's checks control in banks, and manufacturing control operation sheets.

System Considerations

A suitable system employing the Direct Access Photomemory must make optimum use of the following characteristics.

1. One-billion-character storage.
2. One-second access time.
3. Static storage allowing asynchronous and discontinuous input and output.
4. Nonerasable storage to which information can be added at any time.
5. Storage medium automatically replaceable under program control. Also, removable film cartridges which may be filed for historical reference.

It is immediately apparent that there are three areas of contrast with the usual magnetic storage. These are: nonsynchronous versus synchronous storage; easily replaceable versus erasable memory; "wasteful" versus "thrifty" use of memory due to the large capacity available.

The design of any system using the Direct Access Photomemory should make optimum use of the particular characteristics of the Direct Access Photomemory which follows.

Nonsynchronous Versus Synchronous Storage

Rotating magnetic drums or disks, or high-speed tape files frequently require buffer storage to provide a speed match to input and output devices. In some larger machines, several hundred cores are used for this purpose.

A photomemory is static rather than dynamic and can be slaved to accept or present information at any desired rate up to its limit. In many cases the need for a buffer storage may be reduced or even eliminated.

An example that may be considered is input from punched cards directly into the photomemory. Since the electrostrictive elements have buffering properties and can be read into at any speed up

A. J. CRITCHLOW is with International Business Machines Research Laboratory, San Jose, Calif.

to 100 kc per second, any machine that operates at slower rates may feed information directly into the electrostrictive elements without the need for an additional buffer. Normally, this is accomplished by using the clock cycle or timing cycle of a synchronous device to control the stepping of information into the photo memory.

Another case in which this static property is valuable is in reading from the photomemory into a magnetic processing drum or magnetic tapes for entry of information into another computer which may be at a remote point. Now, the cathode-ray flying spot scanner may be slaved to the clock cycle of the drum or tape so that information is read from the photomemory at the speed required by the synchronous device.

Easily Replaceable Versus Erasable Memory

It is interesting to note that there is little difference between erasable and "updatable" memories if memory is infinite. The need for selective alteration of memories was brought about primarily by the use of magnetic storage systems of limited capacity. The same result is obtained by crossing out or otherwise identifying old information and the cumulative adding on of the new information as is done in ledger bookkeeping. An advantage of retaining the old information is the historical record and audit trail it provides. Of course, storage capacity is used at a rapid rate, but this is unimportant if the old information may be easily filed and an up-dated record prepared quickly and automatically.

There is an old rule of thumb in data processing which says "80% of the volume is accounted for by 20% of the items." This adage expresses the realization that some accounts are much more active than others and that a data-processing system must take this fact into account. Suppose that a careful study of the problem to be mechanized has been made and some such figure is established. (It is important to note that a large data-processing system has a large "statistical inertia" since it is based on a large population which changes slowly.) How is it handled on the Direct Access Photomemory?

If an indexed addressing system is used, and this is usually the best way to address large files, the address assignments can be completely arbitrary so that very active files can be assigned more space than less active files. Suppose monthly revision of the files is required. Then, space is assigned in blocks of standard size which are greater than the maximum activity to be expected during this time period. Sometimes, however, unexpected activity will occur so that it is necessary to provide an "overflow" address to prevent loss of information in case of activity exceeding the allotted space.

Several attempts have been made to devise a way to compare erasable and nonerasable storage; so far, no satisfactory general formula has been found.

There are three areas to be compared.

1. In those applications where historical information and audit trails are important, the nonerasable memory has advantages.

2. When transaction rates are high, the number of different accounts small, and only summary information is required, the erasable memories have definite advantages.

3. An intermediate application area where large numbers of accounts are maintained with fairly low activity per account, requires detailed evaluation to determine which is best. In this intermediate area, it appears that to compete, a nonerasable memory should have about ten times the capacity of an erasable memory with the same cost, access, and reliability characteristics. Since the potential storage density of photomemory may be of the order of 50 times as great as magnetic memory, a strong competitive situation exists and peripheral equipment may well be the deciding factor.

"Wasteful" Versus "Thrifty" Use of Memory

When memory becomes plentiful, many processing operations become easier. Computations once made are stored rather than recomputed. Unit records become larger to make available the seldom used data required for exceptions. Tables of rates, terms, and prices are used to reduce computation. Programs are written out in full rather than using "loop" operations to save storage. Also, it is possible to provide numerous cross indexes to facilitate the preparation of scheduled monthly reports.

The following example was chosen because it is easy to understand and illustrates the way in which a large capacity memory may be used to save processing time or processing equipment in a business data processing machine.

Mortgage Loan Accounting Application

General Description

Suppose a mortgage-loan accounting system is considered for banks or savings and loan associations. Skipping over the details of entering information into the file, which can be handled by the usual punched card or keyboard input techniques, the over-all system requirements can be considered.

A mortgage loan accounting system requires a mortgage record for each mortgage held. It would be very desirable to store in one place a record of all payments made and a corresponding record of current balance, interest charges, taxes and insurance paid, and other information regarding the mortgage.

Assume that a mortgage loan accounting system must handle 30,000 mortgages for an average of 20-years' life, with 15 entries per year for each mortgage and a line of 100 characters for each entry. Then, each line will contain identification and classification data, a constant payment amount, a monthly charge to escrow and its current balance, monthly charge to interest, the current principal, and any surplus payment amount.

Initially, the record for one mortgage will contain one full line of information and more than 300 empty lines, each capable of holding 100 characters. As payments are made, one line per payment will be filled and the account up-dated. After 10 years of operation the stored information should amount to about 450 million characters or 45% of capacity. This is wasteful use of memory indeed. However, it has three advantages which justify its use:

1. Active accounts can be processed directly against memory on a random access basis, with all the information required to process the account directly available.

2. Inquiries on account balances and payments are easily available on an inquiry basis.

3. Periodic reports required by the Federal Housing Authority or for tax purposes can be filled from the data directly available on an automatic basis.

In order to save processing time in preparing routine reports, it is desirable to prepare duplicate files by routine posting of information during in-line processing. For example, a report of mortgages by district, dwelling type, and occupation of mortgage holder can be maintained simply by posting to a selected ledger or subledger and allowing ample space for additions during the report period.

A detailed example of one part of the mortgage loan procedure is described next to indicate the solution to one of the problems that arise in processing against such a large file, in an order unlike the internal sequence of the records.

Delinquent Mortgage Loan Procedure

1. Provide a separate section of the Direct Access Photomemory file which contains a

list of all mortgage numbers in due-date sequence. All months of the year can be handled with 31 possible due-dates so that 31 strips, one for each day, can be used. (This is a storage capacity of 3.1 million characters.) Each line on the due-date strips will contain 10 mortgage numbers of 10 digits each.

2. Each day the strip index for a single past-due date will be sequentially scanned to obtain the internal Direct Access Photomemory address identifying the mortgage balance record for each mortgage on the strip.

3. When the record is found, its payment status is checked. If it is unpaid, the surplus payment amount is checked to determine whether it covers the required payment. If not, a listing of past-due mortgage payments can be made and the past due notices prepared on an associated typewriter or printer.

Addressing a Billion-Character Memory

Two general techniques are used for addressing random access memories.

1. Address conversion by mathematical calculation. This excellent technique is well known and will not be discussed further in this paper. A good description is given by W. W. Peterson.[2]

2. Addressing by consulting a continuously maintained index is particularly appropriate for the Direct Access Photomemory and its use is described next.

In a large aircraft factory, for example, there may be 500,000 separate parts which have been assigned numbers over a long period of time. Changes in numbering systems, design changes, or obsolescence may have reduced the number of active part numbers to perhaps 100,000. Each part number may contain as many as 20 alphabetic and numeric characters in any sequence. Since all permutations of 20 characters represent a number far beyond the total storage capacity of the file, if provisions were made for all possible entries, a means must be found to reduce the external address to an internal address which more accurately represents the actual storage required for all the part numbers.

Taking advantage of the large "statistical inertia" inherent in a large file, the full file information can be arranged in some alpha-numeric order based on the needs of the system. This ordered file may then be arbitrarily divided into groups of equal size for indexing purposes with some assurance that changes in this grouping will not be required often. Group size is controlled by the natural divisions of the memory itself in much the same way that dictionary divisions are controlled by page size.

Primary Indexing

In the Direct Access Photomemory, this first index gives the location on an individual strip. Thus, a 20-digit alpha-numeric plus a 4-digit strip number would be needed for each of 10,000 strips, or a total primary index storage of 240,000 characters using three "primary index" storage strips. A simple auxiliary storage would determine which primary index storage would be selected. Actually, it might turn out to be desirable to use ten strips, each partially full, for the primary index if the first digit of the external address was numeric and none of the resulting ten groups exceeded the capacity of the primary address strip. Instead of the first digit of the external address any other digit could be used to provide this index.

Note that primary index requirements depend only on external address length and the number of storage divisions used. The number of accounts to be stored does not enter. Storage designation may be completely arbitrary if desired.

Secondary Indexing

Secondary indexing would be provided at the top of each strip for the following reasons.

1. Reduction of access requirements, since the index and the required information are on the same strip.

2. Ease of maintenance of index. Within the smaller divisions of the index changes would be expected more often than within the large divisions. When a strip is to be rewritten its index may also be modified as required.

Maintenance of the index is greatly simplified by taking advantage of the high-speed scanning abilities of the cathode-ray tube scanner. This scanning speed allows the whole index to be scanned by cathode-ray tube beam deflection to search for an item so that index information may be added serially in chronological order rather than by interfiling of new information. When new accounts or part numbers are to be added to the file, the index term may be added at the end of the index list on the strip. This scanning of the index is especially simple for files which have less than 100,000 accounts since there are then only ten index terms per strip. In these cases, it is desirable to associate other information, such as overflow indication, with the index terms.

An external-internal addressing system of this kind is capable of handling external addresses which consist of names, descriptive words, or codes of various kinds, thus eliminating many processing steps now carried out by humans. Such an index requires large amounts of storage, however, and may not be feasible for small memories.

Conclusions

The Direct Access Photomemory program was a research effort in the direction of building memories which are large enough and cheap enough so that all of a business' information could be made available very rapidly on a random access basis. If this concept is realistic, as IBM thinks it is, major changes can be expected in data processing and much more complete integration of information.

Perhaps it should be re-emphasized that the Direct Access Photomemory has been devised in research to prove the feasibility of certain ideas. No attempt has been made to make a machine suitable for production, and there are no current plans to produce this machine.

However, the results of this program indicate that serious consideration must be given to the systems problems that appear if the cost of the "master file" memory of the large data-processing systems of the future is eventually to become negligible. Some of these questions have been touched upon, and indications of the probable answers have been given. Much work remains to be done to provide definite answers to the way in which machines, and procedures, should be organized to take full advantage of the potential available.

References

1. The Random-Access Memory Accounting Machine, Part I, System Organization of the IBM 305, **M. L. Lesser, J. W. Haanstra.** Part II, The Magnetic-Disk, Random Access Memory, **T. Noyes, W. E. Dickinson.** *IBM Journal of Research and Development*, New York, N. Y., vol. 1, no. 2, Jan. 1957, pp. 62–75.

2. Addressing for Random-Access Storage, **W. W. Peterson.** *IBM Journal of Research and Development*, New York, N. Y., vol. 1, no. 2, Apr. 1957, pp. 130–46.

3. A High-Scanning-Rate Storage Device for Computer Applications, **D. M. Baumann.** *Journal of the Association for Computing Machinery*, New York, N. Y., vol. 5, Jan. 1958, pp. 76–88.

4. Photographic Techniques for Information Storage, **G. W. King, G. W. Brown, L. N. Ridenour.** *Proceedings*, Institute of Radio Engineers, New York, N. Y., vol. 41, Oct. 1953, pp. 1421–28.

5. A Permanent High Speed Store for Use with Digital Computers, **R. D. Ryan.** *Transactions*, Institute of Radio Engineers, Professional Group on Electronic Computers, vol. EC-3, Sept. 1954, pp. 2–5.

Discussion

Chairman Brown: The first question for Mr. Schwab is from H. G. Cragon, Aro, Inc.: "How many analog inputs to the digitizer and what are the voltage levels required?"

Helmut Schwab: The system has a certain flexibility. The fundamental commutator block has 20 inputs. The next larger block has 100 analog inputs. You can stack together as many such blocks as you wish up to 1,000 inputs and additional groups of 1,000 through intermediate circuitry. The analog voltage required is between ±3V full scale. The one that is required is between zero and three volumes. This requires for certain tests, amplifiers between the transducers and our system. That is what you have seen in the block diagrams.

Chairman Brown: R. F. Johnson, Northrup Aircraft asks, "Did you say that MicroSADIC can write a standard format IBM *704* input tape which can be used on off-line equipment?"

Helmut Schwab. If you mean by off-line, such peripheral equipment as an IBM card-punch, then the answer would be "yes." There are formats which go directly to the IBM *704* that cannot be used in peripheral equipment because they have very long blocks which give a very high average sampling rate. However, there are other formats which go to the IBM *704* as well as to the peripheral equipment, and finally, there is a group of formats which are specifically made for peripheral equipment.

Chairman Brown: R. L. Richman, Navy Electronics Laboratory asks, "How do you provide for a format for use with a Datatron *205*?"

Helmut Schwab. This problem is at present under study in our laboratory. It is possible to analyze the format and compose the different format components in the MicroSADIC in the right way for the *205*. However, at this time I would not like to commit myself on this problem. In a matter of days I should be able to give you a firm answer concerning this question. Indications at this time are that it is entirely possible. At least, I think all Milli-SADIC formats are accepted by the *205*.

Chairman Brown: The first question for Mr. Stockebrand is from C. L. Baker, Rand Corporation: "How is the tape head held in contact with the tape? On what unit is the constant density timing track written?"

T. C. Stockebrand: It is not in contact with the tape. As the tape passes over the guide, there is a little section by the head in which the tape is flat. You move the head until it just barely touches the tape. In fact, you can see through it under most conditions of operation. We have to be careful not to wrap the tape around the head, because there would be finite variations in tape tension which would, in some cases, jam the tape into very close contact with the head and normal little air cushions would be removed.

In other cases, when the machine is coasting, it would cause a large gap. So, we use no head wrap and in fact, the tape is not in contact with the head most of the time.

Chairman Brown: The second part of the question is: "On what unit is the constant density timing tract written?"

T. C. Stockebrand: We have an attachment which we clamp on the machine, which has a hysteresis synchronous motor to drive tape through the machine at a uniform rate. Then we write the track by using an oscillator, or we could write the track on another machine, say, the central bit. If we had an interchangeability problem, we could write the central several tracks on another machine, then put the tape on this machine and operate at any controlled speed and transfer the information from the arbitrarily written tracks to the timing tracks. In this way we should be able to compensate for any skew in the driver. At the present time we are not sure which scheme we will have to use.

Chairman Brown: We have a question from the floor.

The question is as follows: "Can you record at 400 inches per second and read it at 100 inches per second?"

T. C. Stockebrand: The usable range of speed of any machine is determined; the upper limit is determined by a computer's external devices, and the lower limit is determined by the signal. The upper limit can be at any speed at which the device to which this is associated is able to take care of it. So, yes, you can write at 400 and you can read back at any speed which you would like to set. In our case, the computer can select either of two speeds, near the high and near the low of the operating extremes and can oscillate back and forth between those speeds, so the average speed is whatever the computer would desire.

Chairman Brown: There is another question from the floor: "What is the maximum recording density?"

T. C. Stockbrand: The recording density is 200-information bits per inch, and this is fixed by the constant density timing track which is written at 400 zero crossings per inch, and it does not vary.

Chairman Brown: A. M. Nelson, Magnavox Corporation asks, "What is the induced signal on the clock channel due to writing on an adjacent track?"

T. C. Stockebrand: Perhaps Mr. Best is better qualified to answer that.

R. L. Best: We obviously cannot have a crosstalk signal comparable to the signal being picked up on the tape. So, the head is specially constructed with a heavy shielding between the timing channel and the other channels. We have the timing channels as the two outside channels. They are separated from the inner body of the head by heavy laminated shielding.

Chairman Brown: E. Goldstein, Bell Telephone Laboratories asks, "How do you prevent crosstalk from write currents from interfering with the reading of the permanently recorded clock tract?"

T. C. Stockebrand: I believe the answer to that question has already been discussed.

Chairman Brown: Mr. D. K. Sampson, Remington-Rand Univac asks, "What is the role of the hysteresis clutch on the block diagram?"

T. C. Stockebrand: If you mean the little star-shaped affair at the end of the motor, when you stop, you have to know which reel is the trailing reel so you can apply power to that one. So, we have a little cup which is dragged by a star-shaped magnet between two stops. When the tape is moving in one direction it is at one stop, and when it is moving in the other direction it is at the other stop. We use this as an emergency switch to tell us in which direction the tape is going for stopping purposes, and also for logic purposes in the electronics.

We chose this design because it will work even when the tape just barely creeps through the machine as it might in some stand-by conditions. In a normal speed sensor there is always some band of very slow speeds in which it will not function.

Chairman Brown: Mr. Nothman, Gilfillan Brothers asks, "How do you generate the variable delay for the clock pulses?"

T. C. Stockebrand: We have not figured that problem out as yet, either, but we have some plans. We can very simply generate an analog voltage that is proportional to the average rate of the timing pulses. This will give us an analog voltage which is proportional with the tape speed. We can then, in turn, use this varying voltage to give us the variable delay.

Chairman Brown: Mr. Houde, National Cash Register Company asks Mr. Litz, "What is the energy in watt seconds to write 1 by 3 mil bit to a density of 0.65?"

F. A. Litz: Approximately 1 20-milliwatt second.

Chairman Brown: Mr. R. A. Kudlich, Bell Telephone Laboratories asks, "Have you made any estimates as to the cost of a Direct Access Photomemory system?"

A. J. Critchlow: I am afraid I cannot give out that information.

Chairman Brown: Mr. R. A. Kirsch, National Bureau of Standards asks, "Please reconcile your 100-kc writing rate with the 500-millisecond exposure time that you quoted."

A. J. Critchlow: The 100-kc writing rate is the rate at which information can be accepted from an external source. That is the rate at which the electrostrictive elements will set up using the circuitry developed. Of course, you cannot receive information continuously at that rate.

Chairman Brown: Mr. C. H. Richards, Convair-Astronautics asks, "How do you tell when a strip is full and how long does it take to change a strip?"

F. A. Litz: When the strip is inserted into the read-write station, it should be scanned to determine the number of records so far

recorded. This count would then be entered into the positioning mechanism counter. If the strip is full, it would have to be returned to a historical file cell and a new strip put into position. With the present system this would take probably one and a half seconds.

Chairman Brown: Mr. J. Cornell, National Cash Register Company asks, "How do you cope with the phototube waveform? What means of shaping it is employed?"

F. A. Litz: Basically, we were clamping the output and employed a data-synchronized pulse system; so we effectively clamped the output, clipped it and data synchronized the clock using two oscillators controlled by the falling edge of a bit. The clock would be resynchronized so that it would account for such things as jitter in the waveform.

Chairman Brown: Mr. McAllister, Douglas Aircraft asks, "Are all of the 20 drawers removable and may they be replaced? What is the estimate of commercial availability?"

A. J. Critchlow: The drawers were designed to be easily replaced; one drawer or any number of drawers. In addition, each cell is individually replaceable. Here is a cell, one cell with 50 strips in it; this has a capacity of 5 million characters. This can be filed in a filing cabinet very easily.

F. A. Litz: We might say just a little more about the cell. It fits in a file drawer having 20-cell slots, so it is very easily removable.

A. J. Critchlow: As regards availability, as we have mentioned, this is a research effort and I do not think I have any more to say on the subject.

Chairman Brown: Mr. W. Myers, Eastman Kodak Company asks, "What rate will the shutter elements accept data if the electronics were changed? What is your thinking on multiple access to the file?"

F. A. Litz: Fundamentally, the shutter elements are electrostrictive devices that work as capacitors in the circuit. If you are willing to supply the circuitry that will provide the charging current, you can charge elements with about 5-microsecond pulses and the element itself will not move for something on the order of 10 milliseconds. Then it will retain its charge for some period determined by its normal leakage impedance. Concerning the multiple access for this, we chose the moving drawer file, so that several access ways could be placed over the file. Then it would be simply a matter of keeping track of which drawers were over which access. This is strictly a control problem.

Chairman Brown: Mr. C. O. Carlson, National Cash Register Company asks, "How was spot size defined and how measured? What was the primary limitation for the quoted signal to noise ratio? Compare cost of shutter buffering versus core buffering."

F. A. Litz: The spot size fundamentally was defined to include the fringe light and was basically measured by a photographic technique looking at the waveform itself for a 1010 pattern. The primary limitations to the signal to noise ratio is the amount of light that is available and the relative density of the exposed film.

For the second part of your question, I would say this: I think you have two different types of systems here. At the present time prototype basis shutter buffering is probably not cheap, compared to core buffering, by probably a factor of maybe two, three, or something on this order. However, the shutter actuator would provide several things. In addition to being a buffer, it is a mechanical transducer which provides the force necessary to pull the shutter open. So, if you had a core buffer receiving the information, then you would have to have shutter drivers which would be the small solenoids or some other type of electromechanical transducer.

Chairman Brown: Mr. I. L. Wieselman, Telemeter Magnetics, Inc. asks, "It takes 500 milliseconds for photography of 100 characters. Is this correct?"

F. A. Litz: Well, this is with the present system. Actually, we know ways of improving this recording time. One of them would be a better spectral match between the exposed film and the scanning light spot. Here we can gain a factor or two. We already have light sources available to us which have more energy and useful color spectrum than the recording light source we used. One 2,000 watt source has about 200 watts of usable energy, in the 2,300 to 3,200 angstrom range. This light source has 50% more usable energy than the one used. Recording time could also be decreased through development of a more sensitive film, which is certainly in order.

Chairman Brown: I. L. Wieselman also asks, "If you do not buffer output, do you get into trouble trying to read information from film; how do you recognize the bits?"

A. J. Critchlow: I do not believe we got across the idea of the self-clock system. This was developed for the Ramac and is described in the Ramac operation manual. Essentially, this technique develops a clock track from the information which is being read. The information may be read at any speed, slow or fast, and develops a clock track or a clock signal which is continually changed. This is done by starting and stopping an oscillator for each bit read. Every time a bit is read, you start an oscillator. This oscillator then provides a clock signal until another bit is read. Then you start another oscillator. You always have one oscillator that is in synchronization with the speed of the reading. If you change the reading rate it is only necessary that the oscillator information speed be changed.

We worked out means for controlling the oscillators used to control the sweep rate of the cathode ray tube. We are able to take a clock track or clock signal from outside sources and use it to generate a clock which is synchronous with the clock developed from the reading of the bits.

Chairman Brown: J. E. Morse, Eastman Kodak Company: "In reading the signal waveform shown, what is taken as signal in computing the signal ratio of 5 to 1?"

F. A. Litz: Fundamentally, peak-to-peak, voltage that was referenced to the relative optical density and a 1010 pattern. In other words, I indicated that you have a variation in the peak-to-peak voltage as a result of variations in information patterns, which is what happens when the spot size approaches the size of a bit, which was the case.

Some people might wonder why we did not reduce the spot size still further optically, and the answer to this question is: As you reduce the size of the spot optically, you lose light and in general do not gain anything by reducing the spot size beyond a certain point.

Chairman Brown: J. E. Morse also asks, "Can old data be marked to indicate its obsolescence to the scanner?"

A. J. Critchlow: This can be done because the self-developing film is always sensitive so you can always add information to it. The thought here was that we would provide both an overflow indication and old data indication along the edge of the strip. There is one square that is not vacant unless the information is new, and it is marked. This indicates the information was old. You can go further and classify the old data by data period, if you wish.

Chairman Brown: J. H. Jacobs, Consolidated Electro Dynamics asks, "Is the Chalkley film stable to sunlight?"

F. A. Litz: This is one of the problems, of course. The film would fog in sunlight because it is sensitive to ultraviolet light. However, the plastic material of which the cell is made filters out the ultraviolet in sunlight. In general, a plastic covering over the file would eliminate exposure from ambient light.

Chairman Brown: J. Heid, General Electric Company asks, "What precautions are taken to eliminate errors arising from dust?"

F. A. Litz: Actually, dust and dirt are a problem, and this is part of the design basis for the bit size that was used. We actually were able to record bits as high as 10,000 bits per inch. The film resolution itself is molecular and our reading and writing system optics carries a resolution of more than 100 lines per millimeter. Although recording very dense records, we were not at out limit at 1,000 bits per inch. This was selected because of anticipated dust and dirt infiltration. At this density, we did not have any particular trouble from dirt. We were operating with an open file and with the most sloppy environmental conditions possible.

Chairman Brown: J. M. Bennett, University of Sydney asks, "How is the recorded information fixed? Is this done line by line?"

F. A. Litz: We have answered that question.

Chairman Brown: P. Dreyfuss, Bull Company, asks, "How long would it take to load or reload the complete file?"

F. A. Litz: It depends upon what you mean by loading and reloading. Actually, loading the file normally would consist of

placing the first record in a ledger record on every strip for which you have ledger addresses. This results in quite a large number, and it depends upon how you count them.

I suppose in loading the file you are talking in terms of several hundred hours. If you want to scan out a billion-character file and you had all the records filmed and scanned out every record, not just the last record in a line, this would take about 50 hours.

Chairman Brown: H. V. Flesch, I. T. T. Laboratories asks, "Is the processing rate, or rather, the processing capacity limited to the number of $1^1/_2$-second periods in the day in which there are 50,000 times the number of records per strip, if the strip retrieval time is the major portion of the one and a half seconds?"

A. J. Critchlow: Actually, strip retrieval time and writing time are about equal. It is according to how you load the film. The time estimate that we used for loading is based on the strip access. The point we did not mention is that, having once gained the strip from the file and put it in a rewrite station, the strip could be indexed in about 20-milliseconds per line.

In fact, if you went through these strips sequentially, I think the time would be somewhat less than the time mentioned.

Chairman Brown: H. V. Flesch also asks, "You had the example of 30,000 accounts. When would the billion bits be filled and what are the over-all process capabilities and the limitations on transactions per day."

A. J. Critchlow: It would take a second and a half minimum per access multiplied by the number of transactions you would normally process in a day, assuming that the logical transactions take no time. You could perhaps handle many more transactions in either of two ways; by greater multiple access, or by having one strip always ready for processing. In that way you can completely hide the effect of the strip access time by having two or three accesses feeding the same rewrite station. You are limited to the writing time of the strip itself, which is about 500 milliseconds, so under optimum conditions you could probably handle at least one transaction per second, and in an 8-hour day this is in the order of 30,000 transactions, which is enough for most applications we considered, and which is certainly enough for many applications you can think of. At this continuous rate the billion character file would last about $1^1/_2$ years.

Chairman Brown: "What was the primary cause of noise? Was it due to the fact that density varies when you are writing? We would like to know what kind of electrical noise you got in the photomultiplier?"

A. J. Critchlow: The basic noise with which we were concerned in our system was the normal light leakage around the shutter elements.

Chairman Brown: "Is this what determined the 5 to 1 ratio?"

A. J. Critchlow: This partially determined the 5 to 1 ratio. Other factors were previously discussed.

The Flow Diagram Approach to Computer Logical Design Using the NCR 304 as an Illustration

JAMES HUDSON
NONMEMBER AIEE

WALTER EDWARDS
NONMEMBER AIEE

D. E. ECKDAHL
NONMEMBER AIEE

A FLOW diagram approach to the logical design and mechanization of computers and other digital equipment has served as a basic design method for the formulation and construction of a medium-size data processor, the NCR *304*.

The flow diagram method used by the National Cash Register, (NCR) Electronics Division resulted from an application to logical design of techniques that the computer programmer uses in developing a computer routine. That is, the programmer's method of developing a routine by constructing a flow diagram which denotes the desired sequence of computer commands was applied to the logical design problem of describing the combinations and sequences of computer operations necessary to the execution of a command. In the same way that a programmer may design a flow using only a general knowledge of a computer's repertoire, it was recognized that a logical designer may lay out a command flow, having in mind only the general character of the functional units of the machine he is designing.

Once the entire command list is represented in flow diagrams the designer is then able to specify in detail the set of interconnections the computer must have, just as the programmer details the set of instructions which the blocks in his flow represent.

To discuss these concepts in more concrete terms, it is desirable to examine the NCR *304* in some detail. Table I displays the more pertinent characteristics of the *304* system. Fig. 1 is a block diagram of the *304* central processor, indicating the major functional units of the machine.

There are:

1. The main store of ferrite core memory with a flip-flop buffer register, M, which also serves as a working register.
2. A second working register, S, which is used for shifting and for intermediate storage.
3. An arithmetic unit composed of an adder and several flip-flop counters.
4. A memory address flip-flop register, L.
5. An auxiliary memory address flip-flop register, A.
6. A control unit consisting of an instruction register, I, a cycle or sequence counter, N, a decoding matrix, and a bank of operation amplifiers which furnish the outputs of the control unit.
7. A pulse counter, P, for defining the subdivisions of a cycle or word-time.
8. A set of timing and decision flip-flops which augment the pulse counter and control as will be discussed shortly.

Given the functional block diagram the logical designer has two problems:

1. To describe, usually in Boolean algebra, what interconnections must be made between the various blocks in the functional diagram.
2. To tabulate for each computer command the time-sequence of interconnections which will produce the desired results.

In practice the designer must propose a set of functional blocks, devise a set of tentative interconnections, establish flow diagrams for those commands which would seem to have the least in common, and then proceed to lay out the remainder of the command flows, modifying the interconnections where necessary and even adding functional units to the block diagram.

Returning to the *304* Fig. 1, to illustrate some of these ideas it is necessary to describe the computer's cycle or word-time. The word-time consists of a fixed sequence of clock intervals which are defined by the pulse counter. These intervals are allocated to:

1. Control set-up.
2. Memory read.
3. Logical manipulation of operands.
4. Memory write.

in the order listed.

The control set-up results in the

JAMES HUDSON, WALTER EDWARDS, AND D. E. ECKDAHL are with National Cash Register Company, Dayton, Ohio.

Table I. NCR 304 Data Processing System Characteristics

Input	Output	File Storage	Computer Structure	Command Types	Memory	Circuitry	Control
Punched paper tape photo-electric reader	Paper tape punch	Drum file	3-Address	Arithmetic	Coincident-current, ferrite arrays	Transistors for counter and address decoding and for the Adder	A flow diagram approach used in description and preparation of computer control
Punched card photo-electric reader	Line printer	Magnetic tape file	Alphanumeric	Logical	20,000 or 40,000 character capacity	Transistorized Eccles-Jordan flip-flops	Each command level realized by the incorporation of a set of magnetic cores corresponding to the set of blocks in the command flow
Magnetic tape	Magnetic tape		Serial by character, parallel by bits	Decision	20 microsecond read-write cycle	Transistors for amplification and inversion	
Electric typewriter	Electric typewriter		Partial word addressing	Editing	66 bit parallel accessing	Germanium diodes for all interconnecting gates and for clamping	Each core threaded by those operation lines which must be activated to realize the logical operation in that block of the flow
			Automatic relative addressing	Data processing		Molybdenum permalloy ribbon cores for logical decoding and summing of control matrix outputs	
			Automatic linking on subroutines	Tape file			
			Automatic program monitoring	Drum file			
				Input-output			

establishment of the set of operations which are to be used in this word-time. These are simply the logical gates which are to be active during the read, logic, and write intervals.

The core memory is then accessed at the cell specified by the address register, or, under certain control conditions, at one of a number of special working cells, *M*1, *M*2, *M*3, etc.

The logical manipulation may take the form of a copying of the memory buffer, into the working register, *S*, or a combining of *M* and *S* through the adder; or a transfer of a portion of *M* to the *A*-register. Or several of these operations may be carried out at the same time. The simultaneous manipulation of data in several registers is a common occurrence in the mechanization of the *304*.

At the completion of the ten clock intervals of the logic period, the memory write circuitry copies the contents of the memory buffer, *M*, into the cell which was cleared during memory read. And the cycle starts again.

Consider the flow diagram, Fig. 2, which is very nearly the add flow for the *304*. Each block in the flow has two aspects. One is that the block represents a fixed interval of time, the word-time. The second is that each block is a unique control configuration which implies a unique combination of operations of active logical gates.

In block 0, the control selects a special memory cell, *M*1, in which is kept the address of the next instruction. This address is transferred from the memory buffer to the memory address register, *L*, and to the auxiliary address register, *A*, during the logic interval. Also during this interval the adder is used to augment the address which is to be restored to the cell, *M*1, for the next instruction address.

In block 1, the first instruction word, which contains the instruction code and three addresses, is read from the cell selected by the address register, *L*. The instruction code is transferred to the instruction register, *I*; the three addresses are copied from the memory buffer into the *S* register; and the auxiliary address register, *A*, is augmented by one. The memory cell is then restored with this first instruction word, and as this block is terminated, the augmented address from the *A* register is copied into the *L* register in parallel.

It is then possible in block 2 to access the cell containing the second instruction word. This word contains partial word extractors which are copied into counters

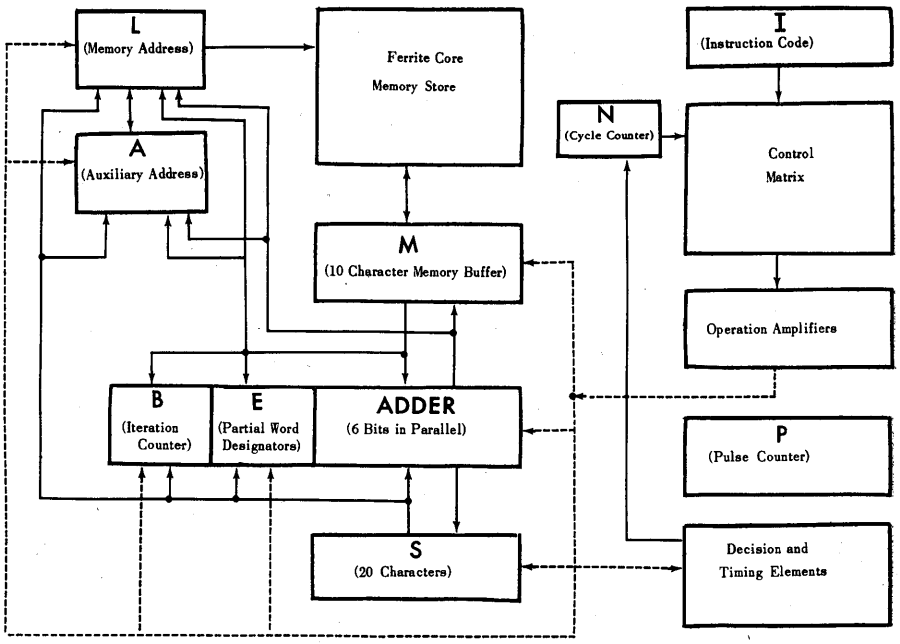


Fig. 1. NCR 304 processor

in the arithmetic section, a relative address designator which is copied into the *A* register, a monitor code which is compared with certain console switches, and a mode digit which is copied into an extension of the instruction register.

As this word is restored to its memory cell, the relative address cell location in *A* is transferred into *L* in parallel.

The computer then advances to block 3 where the memory cell specified by *L* is accessed. The resulting contents of the memory buffer are added to the *S* register, producing absolute addresses for command execution.

In block 4 the two operand addresses, *a* and *b* are transferred from the *S* register to *L* and *A*, and the putaway address, *c*, is transferred to the memory buffer from which it is copied into a special working cell, *M*2, for later referral.

These five blocks which have just been described are almost identical for all command levels in the *304*. This is the process by which the machine's registers are set up for command execution.

Before proceeding to the remainder of the add flow, it is instructive to summarize the characteristics of the machine which have appeared in these first blocks.

Fig. 3 is a more detailed flow chart with a shorthand notation for indicating the operations which are to occur in each block.

Notice that certain operations have been used several times in these first 5 blocks, e.g., *M*-select, *L*-register, $M \rightarrow A$, $A \rightarrow L$.

It is also characteristic of the *304* that the adder is used quite frequently for bookkeeping, such as augmenting the command location as in block 0, augmenting the *A* register as in block 1, and for adding the relative addresses to the index cell as in block 3.

Finally, there are a total of 48 distinct and independent operations in these first 5 set up blocks in the actual *304* flow. These 5 blocks for all commands amount to 1/5 of the total number of blocks in the machine, and require 1/10 of the 480 available machine operations.

Referring again to Fig 2, the *a* and *b* operand addresses are held in *L* and *A* respectively in block 4. In block 5, the cell containing the *a* field is accessed, and *a* is extracted from *M* into *S* under the control of the counter containing the partial word designator for *a*.

In block 6, the *a* field is right-justified as it is transferred into the working cell, *M*2. The putaway location *c*, which was stored here temporarily in block 4, is transferred to the *A* register, and the *b* address in A is copied into the *L* register. The *a* address is discarded.

In block 7, the *b* field is read into *M* and extracted into *S* under the control of the counter containing the partial word designator for *b*. The signs of both operands being now available are compared, the comparison having three possible results:

1. Signs are the same, implying no complementing of either operand is desired.

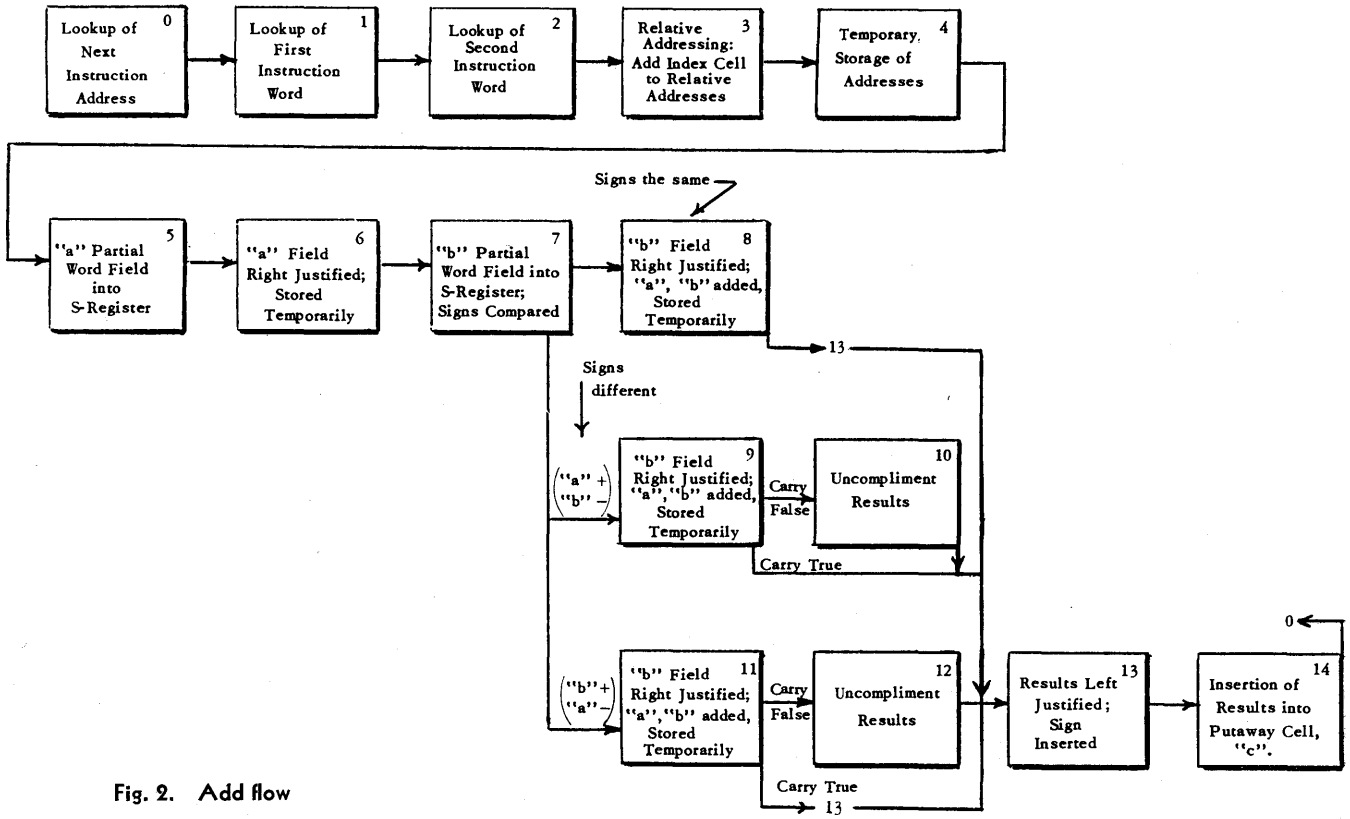


Fig. 2. Add flow

2. a is positive, b is negative implying that b is to be complemented as the addition occurs.

3. b is positive, a is negative implying that a is to be complemented as the addition takes place.

The flow from block 7 illustrates graphically the action that the machine control must take for each of these eventualities. It will advance to block 8 if the signs are the same; it will skip to block 9 if a is positive, b negative; it will skip to block 11 if a is negative, b positive.

This skipping ability which the machine control has allows the logical designer to specify as many as three alternate arbitrary jumps from each block in a flow as well as the ability to count to the next block. These jumps or transformations of the control counter are another result of the operation lines furnished by the control section. In effect, the control establishes those operations which are to occur in one block time, and the numbers of those blocks which may follow, one of which is chosen by decisions made within the block time.

This facility also makes it possible to mechanize iterative processes with the same sequence of blocks by testing for a certain condition at the end of the loop, and either counting or going back to the start of the sequence.

In Fig. 2, if the signs are the same, the control proceeds to block 8 and the b operand is right-justified as it is added to a which has been copied from $M2$ into the M register. The resulting sum in M is restored to $M2$. From block 8 an unconditional jump is made to block 13.

In block 13, $M2$ is copied into the M register and then is left-justified in S in preparation for putaway. The c address is transferred from A to L.

In block 14, the C cell is copied into the M register, and an insert type of operation transfers the results from S into the allocated portion of M. M is then copied back into the C cell location, and the control returns to block 0.

If the a sign is positive, but b negative the control goes from block 7 to block 9. As the addition takes place, the b input to the adder is complemented. Also the adder "carry" indicates at the end of the operation whether the result is valid or an uncomplementing operation is required. Here again the control establishes a possible skip to block 13 or a count to block 10 for uncomplementing.

A like procedure occurs in blocks 11 and 12 for the case of a negative, b positive.

The add command requires then 15 possible control configurations for its execution. Each add command specifies the following actions:

1. Augment the a, b, and c portions of the first instruction word by the corresponding a, b, and c portions of index cell R, to obtain the absolute addresses for operands and putaway.
2. Extract and right-justify operand a.
3. Extract and right-justify operand b.
4. Obtain sum of a and b and left-justify.
5. Insert results into putaway cell.

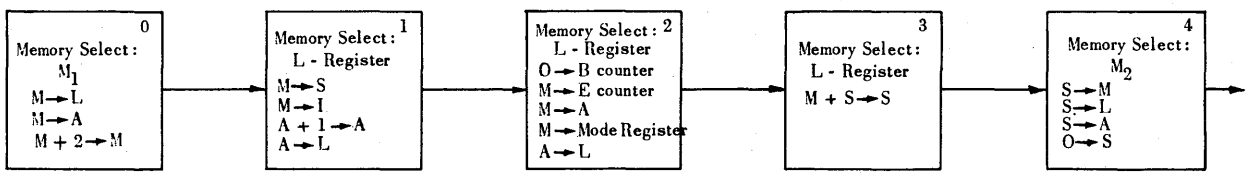


Fig. 3. Detailed flow diagram

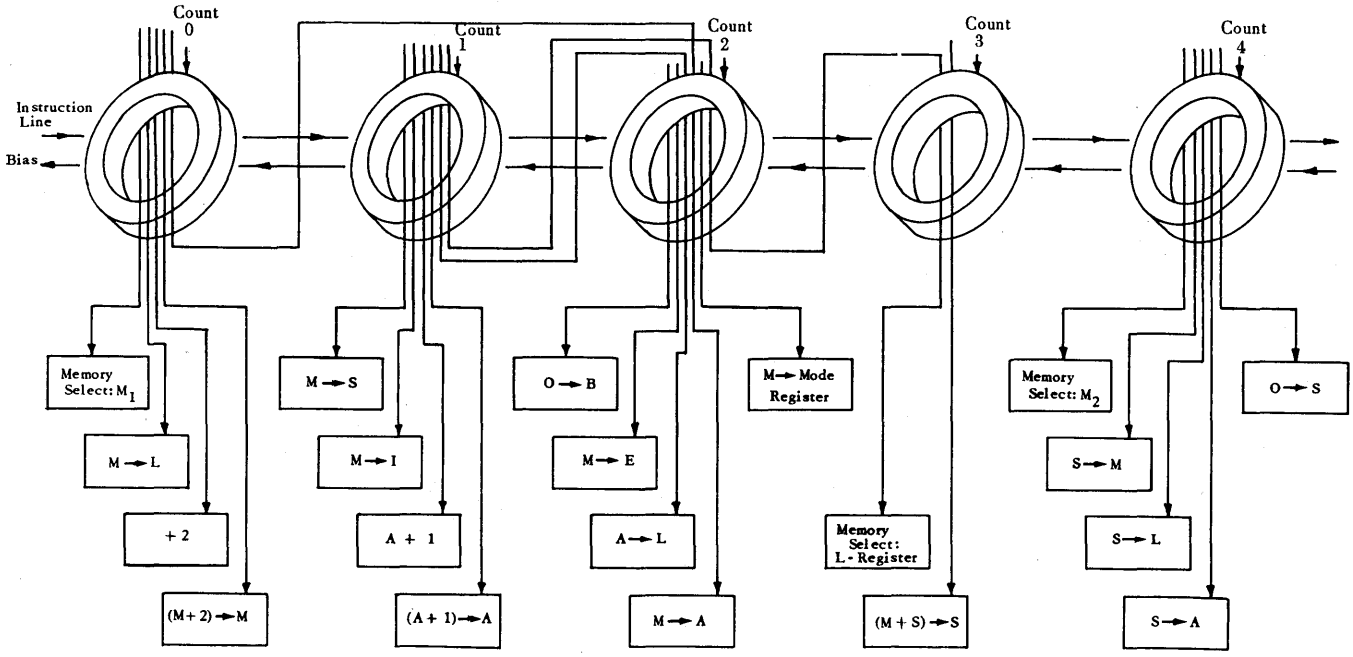


Fig. 4. Core matrix decoding

A total of 82 operations are required to effect these steps.

There are 40 instructions in the *304* of which approximately 1/3 are relatively simple commands similar to the add flow. Another 1/3 of the commands are for data processing and editing. And a final 1/3 are input, output, and tape file commands with extensive bookkeeping and branching requirements.

Turning to the mechanization of the control for this rather formidable command list, a matrix of ribbon cores is used to decode the instruction register and control counter into the individual blocks of a command flow. The matrix consists of a 40 × 40-array, but with cores wired in only for the blocks actually used. As an example, there are 40 intersections on the add command level, but only the first 15 have cores.

The matrix has a bias winding which holds all cores in a "minus 1" state. One of the 40 lines decoded from the instruction register will carry a current sufficient to cancel the bias in those cores which it threads. Then as the control counter changes from state to state, its decoded outputs will, one by one, turn over the cores at the intersections with the instruction line. Hence, as the control counter changes, a change of flux occurs in the core corresponding to the new state, and this flux change must activate those operation amplifiers which are to be used during that block time. For this purpose each operation amplifier is linked to all those cores for which that operation is required. As an example, Fig. 4 shows 5 cores which represent the first 5 blocks of a command set up as discussed previously. There are 20 sense windings in this portion of the array corresponding to the 20 distinct operations which were listed in Fig. 3. Some of these only thread one core of this section; others thread 2 or 3 cores. Each sense winding eventually terminates on a latching amplifier which will be activated if any core on its winding is turned over.

In the *304* matrix, there are approximately 900 cores, with an average of 15 sense windings threading each core. There are 480 sense windings emanating from the matrix, each terminating on a latching amplifier which activates a set of logical gates by supplying proper voltages to the product resistors of these gates. One would imagine that 480 operations would imply a certain redundancy. This is actually the case and arises for two reasons. One is the difficulty in optimizing a design of this complexity; but more important is the necessity of minimizing command execution time. This latter reason accounts for most of the redundancy in the operation list of the NCR *304*.

Another implication of this large operation list is that given this list and given the requirement of a new command, a logical design can be prepared for this new command which will probably not require any additions or changes to the functional units other than the installation of a core plate wired for the new command.

In summarizing, a little thought leads one to the following conclusions about the *304* command mechanization. In executing a design from specifications involving rather complex commands, a flow diagram approach results in a presentation of the types of interconnections and the sequence in which these interconnections are made in the various command levels. This presentation shows the compromise which the designer made between execution time and minimization of equipment.

The control mechanization is nearly optimum for the *304* command list for three reasons.

1. Flexibility in modifications of the machine is achieved by specifications of different wiring patterns for the sense windings in the control matrix.

2. Additional command capabilities were more easily included than would have been the case with previous types of control because these were achieved by the addition only of cores to the matrix, and modifications to the sense windings' paths.

3. The logical design of commands such as multiply, divide, merge, sift, summarize, which have an iterative character are more efficiently executed using the "count" and "skip" features to effect these iterative loops.

Finally, it is felt that the flow diagrams are one of the most easily understood methods of machine description and facilitate a rapid training of maintenance personnel.

Cascaded Variable Cycle Control as Applied to the 220 Computer

E. L. GLASER
NONMEMBER AIEE

THE subject of this paper is basically micro-programming as a design tool, the program of the meeting notwithstanding. This subject of microprogramming has received a great deal of attention in the last several years, primarily as a possible method of making the so-called general purpose computers more flexible. It has been propounded by some that microprogramming offers a technique whereby the user would be free to define, in some degree, the machine organization. However, microprogramming has other rather intriguing possibilities. Some of these have been realized recently at the ElectroData Division of the Burroughs Corporation.

Several years ago work was begun on new control techniques. One of these techniques that showed promise is the one under discussion. This method of control was first applied to the Datatron *220*. This control principle was employed on this project primarily to facilitate detail design and thus to shorten the production lead time. The method was not selected for economic reasons although it now appears that some economy of componentry has been achieved because of it.

The logical requirements for the microcontrol are similar to those of an interpretive program. A command must be fetched from the memory if necessary command arithmetic takes place, and the command is analyzed and executed. In the *220*, the model for discussion, all data is handled serial by digit, parallel by bit, except in the case of access to the parallel-parallel core memory and two broadside address transfer pads.

Fig. 1 represents the flow of data within the central processor and between this processor and the high-speed memory. The broadside shifts between the address buffer and the address register and program counter can be seen. The various microcommands available to the designer are those dealing with this configuration. The different data pads can be closed by means of gates. Any register may be shifted right, certain decades may be counted, and in the case of the adder-subtractor, either addition or subtraction can be ordered.

Since all data manipulation takes place a digit at a time, the same operation will often be applied to several digits which comprise the word or subword. Two types of pulses are used to accomplish this necessary repetition and nonrepetition of function. A digit pulse is used

E. L. GLASER is with Burroughs Corporation, Pasadena, Calif.

Acknowledgment is given to Lloyd Cali who was invaluable in the design of the *220* and for his original thinking in this basic principle of control.

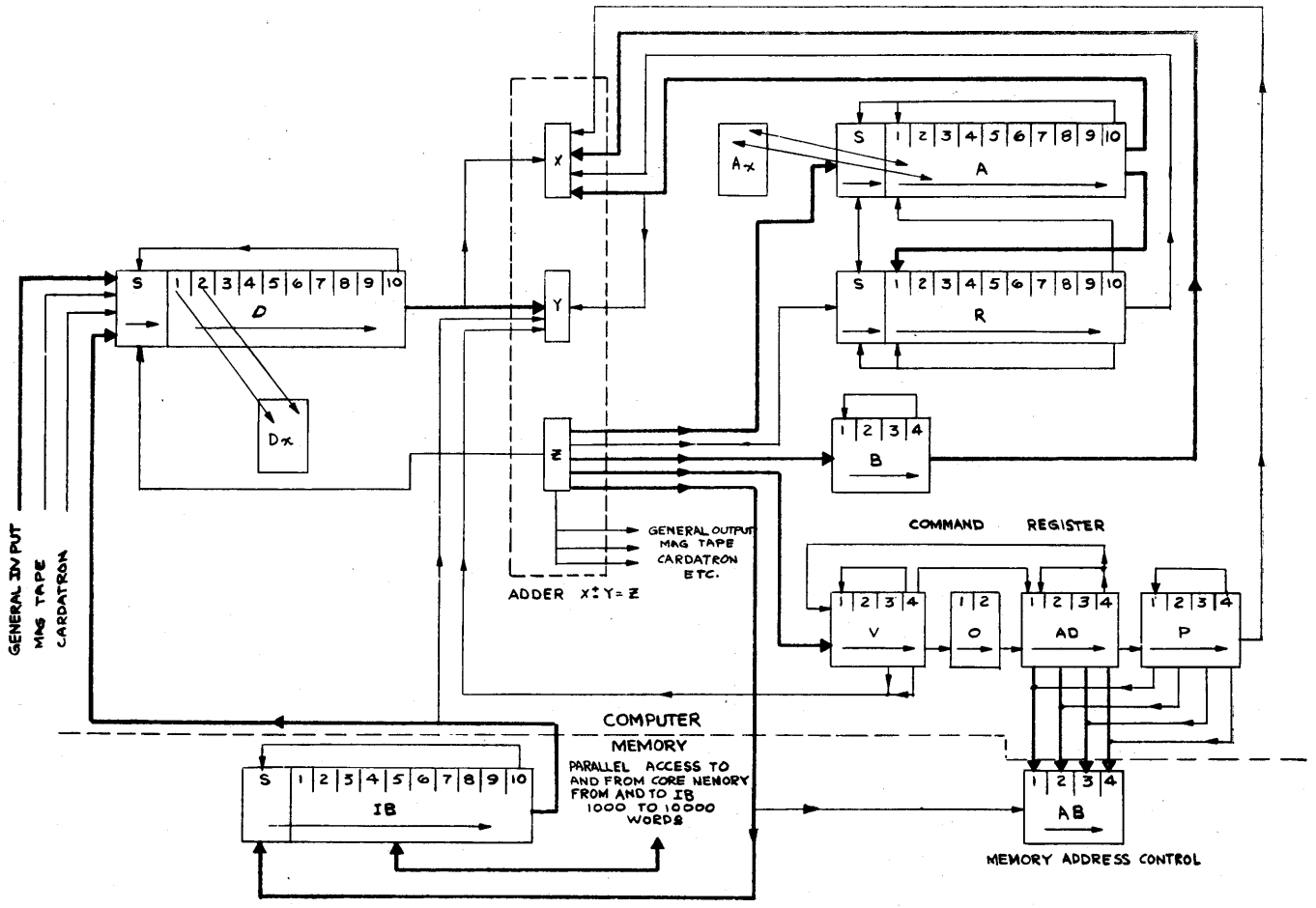


Fig. 1. Flow chart of the 220 computer

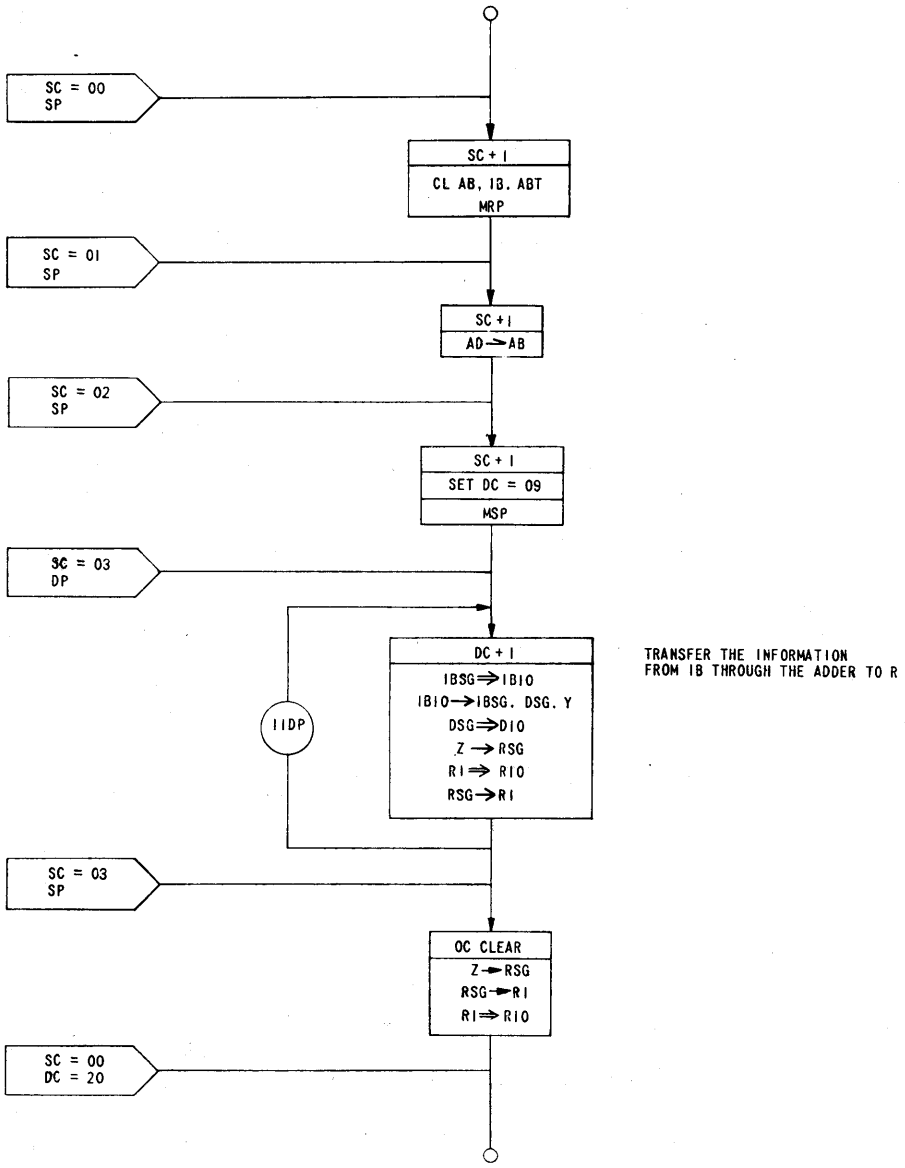


Fig. 2. Flow chart of the load R command

to apply the same operator to a number of digits. The distribution to the several digits is by means of shifting the contents of a register. Groups of these digit pulses are separated by sequence pulses. The sequence pulses may be thought of as the decision-making pulses. Any number of digit pulses may be in a group, as a consequence, the concept of word time has no meaning even though the *220* is a fixed word-length machine.

A digit counter is used to count the number of digit pulses in a group. This counter is designed to seek a terminal value which is, in the *220*, equal to the number 20. When this counter is set to some value less than 20 it will immediately start counting in a positive sense until the value of 20 is reached. A terminal value of zero could have been used in conjunction with a decreasing count; however, the present method was picked because of easier training. This counter can, in any event, be used to define both sequence and digit pulses. Digit pulses occur, by definition, when the digit counter is less than 20. These pulses are used to increase the counter in order to meter the digit pulses. Sequence pulses occur when the counter is equal to 20, the terminal or rest state.

A second counter is employed in the micro-control for the purpose of defining the subcycles in any operation. This sequence counter in the *220* is a 4-stage, binary counter taking on all values from 0 through 15. This counter is set and counted by sequence pulses only. Thus, any sequence pulse is defined by the setting of the sequence counter and in turn, defines the next setting of the sequence counter. A digit pulse is defined by the setting of the sequence counter. The setting of the digit counter is not cited here since its purpose is to meter the number of digit pulses. Its initial setting for each group of digit pulses is the only value of interest and this setting is made and defined by a sequence pulse. As a result, the same pulse that sets the sequence counter sets the digit counter.

The contents of the sequence counter is not in itself sufficient to define the set of micro-operations to be carried out by the associated digit and sequence pulses. It is therefore necessary to have some method by which the coded machine order can be related to the microprogramming cycles, or subroutines. This need is satisfied by the mechanism of a standard-order decoding matrix affixed to the order portion of the command register. During command execution, the output of this matrix defines the order and is gated by the output of the sequence counter to define the suborder. The three functions of digit counter, sequence counter, and order matrix could have been combined; however, the attending logical complexity was too fearsome to contemplate for long. The maximum values for the contents for both control counters was picked as minimal for the *220* system.

Fig. 2 is a flow chart of the load *R* command. This command loads the *R* register from memory. The first pulse *SP* 0 is common to all commands and merely clears the debris from the previous fetch. The second pulse sets up the address in the address buffer and the third pulse orders a memory read. All of these pulses are sequence pulses and each defines the next settings of the control counters. So far the digit pulse groups have been of length zero. *SP* 2 also sets the digit counter to 09 which is 20 minus 11. Since 11 pulses are needed to shift the word to the *T* register, this setting is the correct one. The final *SP* is used to shift the *R* register into proper position and at the same time all control imaginable is cleared, or otherwise disposed of.

It can be deduced from this explanation, hopefully, that some sort of timing flip-flop is still needed to differentiate between the two basic states of a classic stored program computer. When this flip-flop is in the fetch state, the output of the order matrix is blithely ignored and the sequence counter and the timing flip-flop alone control the fetch cycle. During execution this flip-flop is in the opposite state. This state is changed at the end of each half cycle of the computer. The fetch state is also used during the control of

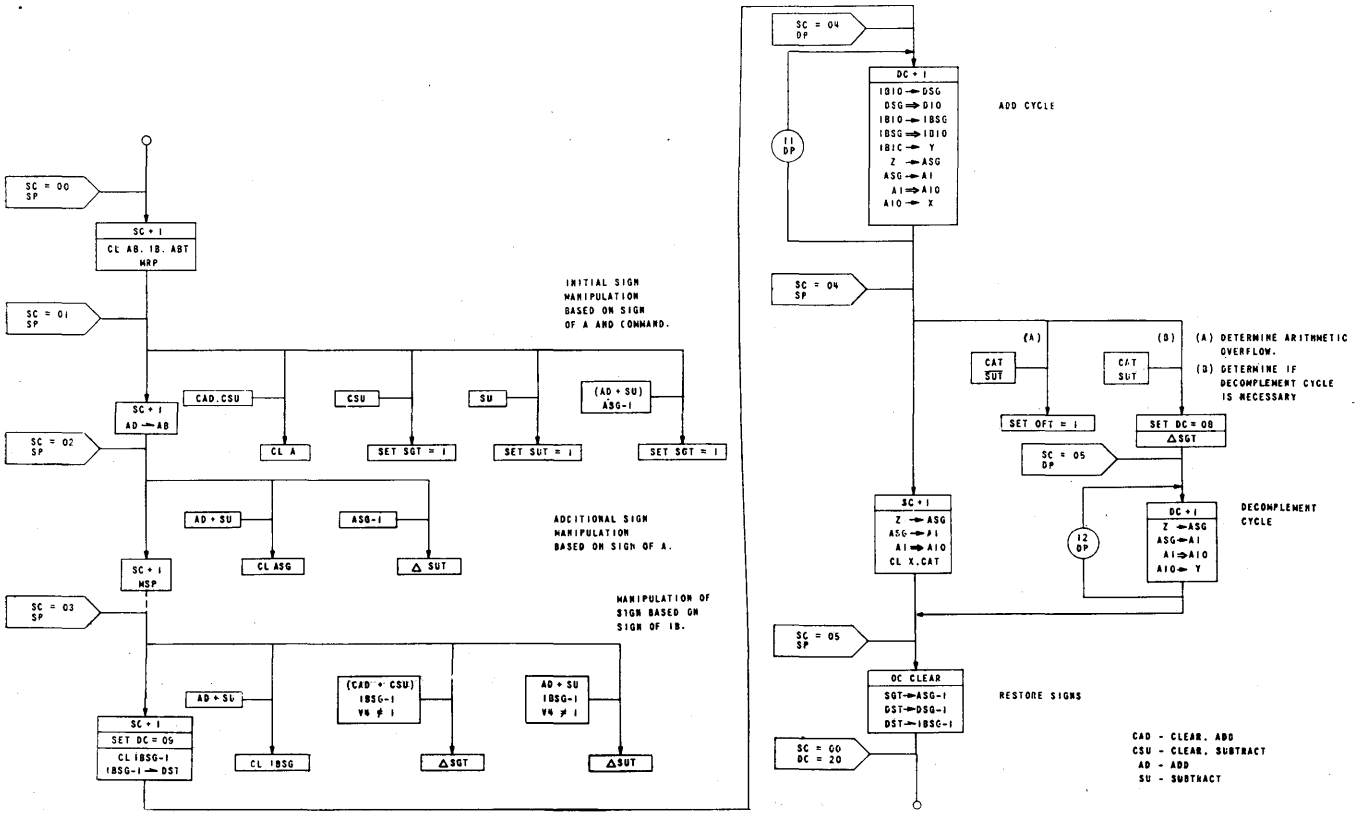


Fig. 3. Flow chart

cycles from the console. Micro-operations are not, and should not be, dependent on the contents of the order register.

A number of extraneous flip-flops are still needed in the control section to store binary decisions. Several are there because of the command list. These include the overflow and compare indicators. Some are there because of the nature of the universe. The carry flip-flop is a good example. One or two are used to make life easier for the designer. An example of this class is the subtract flip-flop. This controls the adder-subtractor. In all cases, combinational logic could replace this flip-flop; however, the presence of this flip-flop makes possible the use of sequential rather than combinational logic at a reasonable saving in complexity. Since the data for controlling this flip-flop are available only in a sequential fashion, it follows that no loss of performance is encountered.

A second example of command control is the fixed point add group. Two types of decisions must be made in this group. The entire group of fixed point add commands are compressed into one cycle so that the exact micro-order to be followed is the usual function of the sequence counter and the order matrix. The sign control also enters into the choice of micro-order. A second class of decision is that which is necessary after a complement addition. If the result of the addition is in complementary form, a decomplement cycle is ordered. In all other cases this cycle is skipped. This decision is made by conditional setting of both the digit and sequence counter. (See Fig. 3.) In the case in point, only the digit counter is involved. However, the conditional setting of the sequence counter is prevalent.

An analogy may be drawn between this type of microprogramming and normal coding. The machine order to be designed is the subroutine name. The setting of the sequence counter is equivalent to the step number. The micro-orders are equivalent to the normal commands in programming. The digit counter is similar in operation to a repeat command as used in some machines. The use of this format for command decision has already proved its worth in making it possible to employ young, relatively untrained engineers in the detail stages of design. This also serves as a good basic training ground for these young engineers and gets them into design that much faster.

The translation from the flow chart to a print is not as formidable as one might think. The use of data processing on the flow charted information can group inputs to circuits and produce loadings on outputs. A set of circuit and wiring lists can be produced and from these to a print is straight forward although tedious.

This method of design is far from a panacea. Its main disadvantage is that it does not facilitate optimization of circuitry. Its prime value is in the speed that can be realized in the early detailing of a large system.

The RCA 501 Electronic Data Processing System

GLEN E. POORTE
ASSOCIATE MEMBER AIEE

ARTHUR S. KRANZLEY
NONMEMBER AIEE

THE RCA (Radio Corporation of America), *501* Electronic Data Processing System is a complete and new system in the intermediate- and large-scale performance class. It is a general-purpose system using all transistor logic. The system has been under development for approximately 3 years. The efforts of this program have resulted in a prototype system which is presently in test.

This system is distinguished by the following features:

1. High rates of data transfer are provided by magnetic tape, with complete retention of accuracy control measures.
2. The units comprising the system are completely transistorized.
3. The system is easily expanded in terms of high-speed memory capacity, number of tape stations handled, large scale random access memory, and input-output capacity.
4. Completely variable data organization is enhanced in the RCA *501* system.
5. Simultaneous compute functions with magnetic tape, printer, and paper tape reader operation are provided.
6. Programming flexibility is keyed to maximizing equipment usage with ease.
7. New versatility is provided for magnetic tape handling and address modification.

The system design for accuracy control is based upon a proper balancing of component reliability and built-in measures for checking data transfer and manipulation.

The RCA *501* Electronic Data Processing System is comprised of a computer and both on and off-line input-output units.

The RCA *501* computer actually consists of a number of integrated units. It includes a console, program control, high-speed memory, monitor printer with paper tape punch, paper tape reader, and associated power supplies.

The console typifies the functional design and human engineering which was applied to all units of the system. The console provides for complete monitoring of computer operation, including display of pertinent register and status level action during program testing and maintenance periods.

The program control facilitates addition of up to 63 tape stations, up to 262,144 characters of high-speed memory, and on-line high speed printing. It is basically a modified two address system for use with completely variable data organization.

Program control provides for electronic switching of up to eight tape stations. Additional increments of eight tape stations are provided by tape selecting units which connect to the basic tape selecting unit of the program control. Maximum capacity of the program control is 63 tape stations. Switching time is negligible at the approximate level of 10 microseconds.

The monitor printer is mounted on a separate table and includes a paper tape punch. Its basic use is for operational control, program testing, and exceptional types of output.

The paper tape reader is under programmer control and operates at the basic rate of 400 characters per second. The paper tape reader with its reel mountings is housed in a separate cabinet.

The computer high-speed memory is available in increments of 16,384 characters up to a maximum of 262,144 characters. All character locations are individually addressable. In one 15-microsecond cycle, the programmer has access to any one character, or four characters in parallel, with regeneration included. (See Fig. 1)

The RCA *501* tape station has a read-write rate of 33.3 kc and moves tape 100 inches per second either forward or reverse. (See Fig. 2)

Input equipment in the RCA *501* is designed to accommodate expansion. Eighty column cards are converted to magnetic tape in two ways. A card reader transcribes directly to magnetic tape with editing reserved for the computer. This same unit may also be connected to a card editor which edits each card and records these data on tape in accordance with a programmer-prescribed message format. The combined unit is referred to as a card transcriber. The card reader transcribes cards at a maximum rate of 400 cards per minute. Fig. 3 illustrates one view of the card reader.

An RCA *501* tapewriter and tapewriter-verifier are included in the product line to handle the original preparation and verification of punched paper tape for subsequent on-line transcription into the system using the paper tape reader.

The output equipment is similarly designed to permit convenient expansion. A line printer is provided in the system which is designed to operate on-line from the computer or to operate off-line with a data editor. The on-line printer accepts edited data and is controlled by the computer. The off-line electro mechanical printer is formed by driving the line printer from the data editor. Editing control is provided by a plugboard and a tape loop for complete horizontal and vertical editing of data received from a magnetic tape. Fig. 4 is a view of the printer mechanism which prints at the rate of 600 lines per minute and up to a 120-column line.

In order to accommodate the punching of 80 column cards from magnetic tape, a transcribing card punch is included in the system. This unit provides for punching of cards at a maximum rate of 100 cards per minute.

Large scale random access memory is also available for on-line operation with the RCA *501* computer. A random access file control unit which provides for connection of up to 32 random access files is provided for direct trunk connection to the computer. Each random access file will store 1,500,000 characters with the average random access time to any data being 192 milliseconds.

The RCA *501* tape station handles magnetic tape which is 3/4 inch wide and available in 2,400-foot reels. The recording density is $333^1/_3$ characters per inch and tape speed is 100 inches per second in forward or reverse. Dual recording of data is retained as a primary accuracy control measure and also to enhance magnetic tape life. Each information bit is recorded on two separate tracks on magnetic tape, each one of which is capable of producing a standard signal during reading.

Transistorization

Probably the most significant engineering advance in the RCA *501* Electronic Data Processing System is the completely new all-transistor logic. Working in conjunction with the RCA laboratories in Princeton, N. J. the Semi-Conductor Division, was able to specify

GLEN E. POORTE and ARTHUR S. KRANZLEY are with Radio Corporation of America, Camden, N. J.

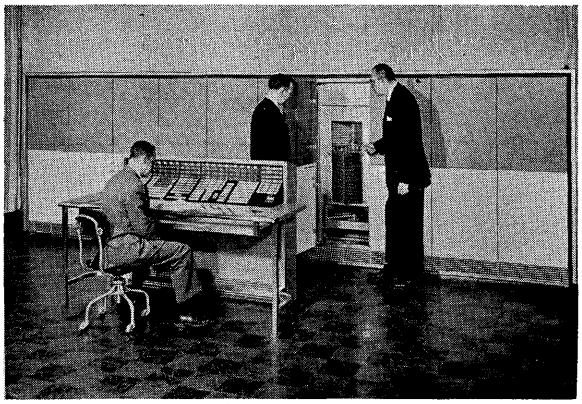


Fig. 1 (left). The RCA 501 computer, showing high-speed storage bay

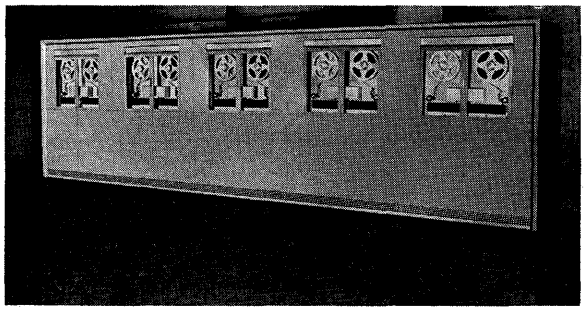


Fig. 2 (right). The RCA 501 tape stations

and tests types of transistors uniquely suited for the system application.

Transistor types resulting from this work are now being used in the RCA *501* system circuits and logic packages. The use of transistors has increased reliability, reduced physical size, and reduced power requirements.

For a computer with a high-speed memory capacity of 32,768 characters, approximately 14,000 transistors are used. The transistors are mounted and dip soldered on small printed wiring boards, as seen in Fig. 5, which in turn are dip soldered to the larger printed wiring board.

Over 90% of the system is constructed from one basic normal operation register, (NOR) circuit, which is wired on the small boards referred to as a submodule. This circuit is a two-input-type unit used for AND, OR, power and indicator driver circuits. It is also used to construct flip-flop register circuits by cross coupling the inputs and outputs. The printed wiring pattern on the large board dictates the plug-in logic configuration. Plug-ins are assembled in a rack as shown in Fig. 6.

Expansibility in the RCA 501 System

Expansion is built into the RCA *501* system in terms of the number of tape stations and tape selecting units used, in the capacity of high-speed memory, and in the type of input and output units.

These features are provided by a system design which minimizes user investment during initial stages of program testing, conversion, and production. Further, these provisions are made to permit the user to expand his production in terms of volumes and variety of operations. All of the features are designed in such a way that major disruption of the operating installation is avoided.

The basic tape selecting and buffer unit in program control, permits connection of one to eight tape stations to the computer. (See Fig. 7). For the purpose of expansion, each trunk line so provided may be connected to a tape selecting unit-*B* which has eight additional trunks, each of which can be connected to tape stations or file control units. With the addition of these units, the system can be expanded to include up to 63 trunk lines to the computer. (See Fig. 8).

The expansion of the system in terms of total number of tape stations connected to the computer and the increase of the high-speed memory from the basic 16,384 characters in modules of 16,384 characters to a total of 262,144 characters are both directly related to the scope of the application handled, as well as the over-all efficiency of the system. Tape station and high-speed memory increases add measurably to the efficiency of sorting and merging. They add substantially to the procedural content of computer capabilities in terms of greater program capacity and the introduction of a wider variety of input and output types to each operation.

Input and output volumes are related directly to the need for expansion from on-line forms of input and output to off-line forms. In addition, the introduction of the card editor and the data editor as parts of the card transcriber and electromechanical printer respectively, improve over-all system efficiency by providing parallel operation and greatly relieving the computer of editing chores.

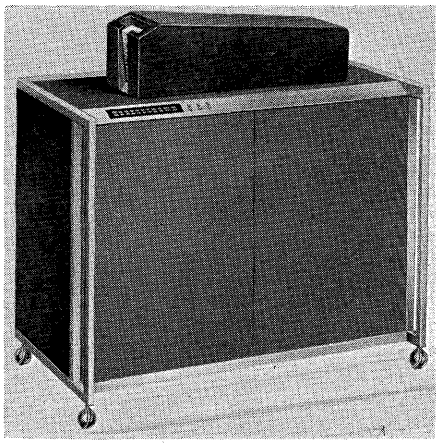


Fig. 3. The card reader

Data Organization

Completely variable data organization is enhanced in the RCA *501* system. The processing of data in its most natural form, avoiding arbitrary restrictions and editing procedures, is the design objective. Binary coded RCA characters consist of six information bits and one parity bit. The RCA code is alpha-numeric, including all letters, numbers, punctuation and special marks, and control symbols. In the RCA system, characters necessary to specify a particular unit of information, such as a stock number or a policyholder's name, are organized as items in their natural form. These items are organized into messages which have unique significance, such as policyholder's reference file, a transaction, etc. A second form of data may be written on magnetic tape without regard to message structure. These data are referred to as a block and consist of eight or more characters which may be handled by the computer for special applications.

The natural computer language, namely the handling of data in the form in which it occurs or in the form in which it corresponds to procedural logic, is a system advantage which increases the efficiency of all equipment items in the system and in particular the effective utilization of tape stations, high-speed memory, and the computer proper. Actual experimental analysis of applications converted to RCA electronic data processing systems reveals a far greater return in terms of data storage reduction than was originally anticipated.

Computer Logic and Simultaneity

The computer logic design uses a two address instruction code which is made

up of eight characters. One character specifies the operation, three characters each specify the two addresses A and B and one character specifies the N code. The N code is used to call for either one or two out of seven static or dynamic register locations. Static registers refer to high-speed memory storage locations, whereas dynamic registers refer to flip-flop-type registers since the contents may constantly change during the operation cycle of an instruction. The contents of these registers are added to either or both of the addresses A and B. The instruction is then executed by using the sum of the address(es) and the modifier as the final address(es). This modification does not alter the instruction stored in the high-speed memory.

The execution time of an instruction depends upon the instruction and the data being processed. For example, in a decimal addition, the instruction performs an end-around carry or is terminated after processing the most significant digits of the operands and the operands' length may be completely variable.

The instruction execution time is the sum of a series of 15-microsecond status levels, each of which is equal to the memory access time. The status level is in turn divided into series of six time pulses each 2.5 microseconds in duration.

The computer uses a bus scheme as shown in Fig. 9. All registers have access to those buses, thus permitting the programmer a high degree of freedom in transferring and storing data and address information in the registers. Registers are used in place of address counters. A simplified binary adder is used to modify the contents of registers on the memory addressing bus by addition or subtraction. This permits character location addressing in either an ascending or descending mode, which in turn adds versatility to data manipulation.

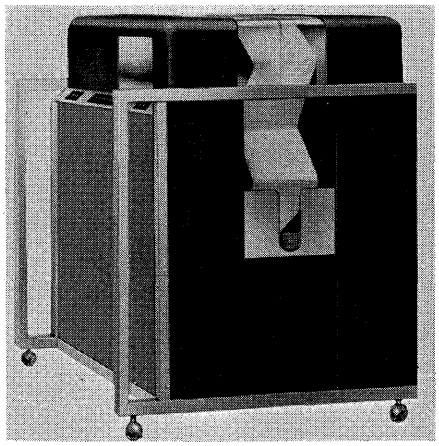


Fig. 4. The on-line printer

As seen from the Fig. 9, simultaneous access to four characters is possible. This decreases the time required for forming and data transfers. No data-length restrictions result from this arrangement, since all character locations are individually addressable.

Simultaneity in the computer has been added to make use of the low access cycle of the high-speed memory during the execution of certain instructions. For example, most of the execution time of a tape instruction is consumed in tape movement. Therefore, an interrupt technique is used which permits a compute function to proceed simultaneously with a second instruction. The compute instruction is interrupted only when access to the high-speed memory is required by data as controlled by the second instruction.

To accomplish this, a normal (NOR) and a simultaneous operation register (SOR) is used as shown in the diagram. These registers specify the operation in process, such as decimal add, etc. Instructions are performed in either the normal or the simultaneous mode. All instructions are initiated and may be performed in the normal mode, but only potentially simultaneous operations may be performed in the simultaneous mode. Potentially simultaneous instructions include certain magnetic and paper tape, monitor print, on-line printer and random access File operations. If the simultaneous mode is free, transfer from the normal to the simultaneous mode will take place by transferring the contents of NOR to SOR and A register to S register. This transfer is normally automatic but may be controlled by the program. If the simultaneous mode is already in use, the transfer of the potentially simultaneous instruction from normal mode will take place upon completion of the simultaneous instruction without interruption and regardless of how far the instruction has progressed in the normal mode. Upon completion of this transfer the next instruction is formed in the normal mode and the execution of it initiated.

Another technique of value both functionally and economically is incorporated in the on-line printer logic. The on-line printer consists of a rotating, etched character drum with an attached character identification and timing disk. Solenoid driven hammers strike the paper, forcing the paper and a carbon ribbon against the rotating drum for character printing. Paper motion equipment is also included. Each one of the 51 print characters, such as the "A", is etched in line for a total of 120 positions and the whole line is parallel to the drum axis. Thus, all similar characters in one print line, such as the "A," are simultaneously printed. A line of print is complete after one complete drum revolution. The logic to perform this function requires that the printed data be edited in the computer. These data from the computer are compared against the character identification disk, the results of which

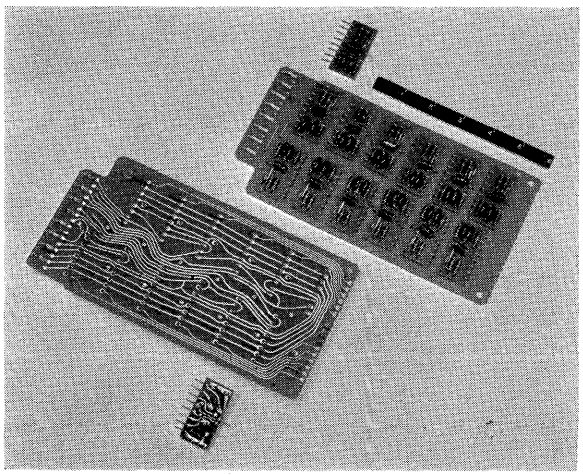


Fig. 5 (left). The transistor submodule and plug-in

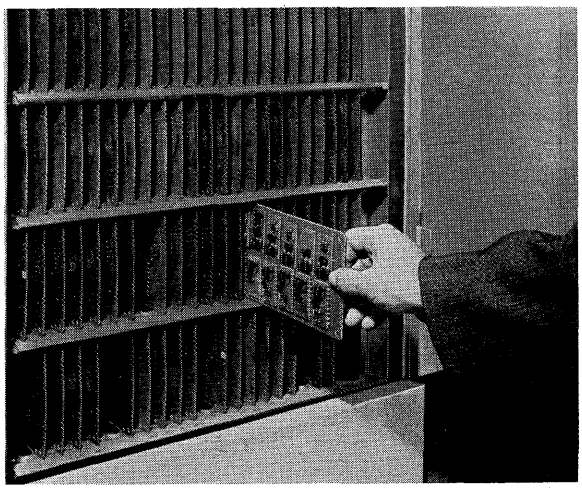


Fig. 6 (right). A rack assembly, plug-in side

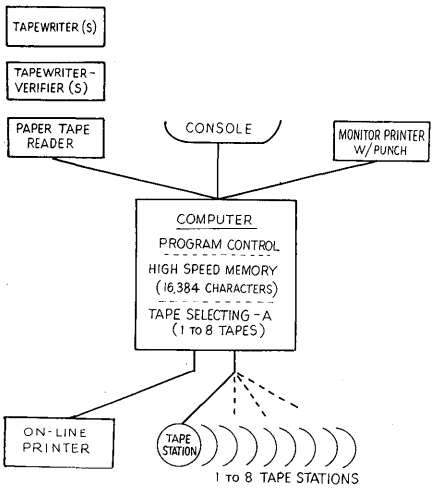


Fig. 7. The basic RCA 501 system

are transferred into a 120-position shift register. The characters of this drum line are printed by parallel shifting this information out of the shift register. This information in turn drives the print hammer solenoids. Following this, the next character compare cycle begins. The process is repeated for each print character until a complete line is printed. The computer then instructs the line printer to advance the paper through the desired number of lines.

Programming

Programming complexities are normally traceable to the difficulties imposed by the inherent design of a data processing system when attempting to realize the full potential of the system's effective capacity.

One important programming feature of the RCA *501* computer is the provision for automatic relative addressing of both programs and data. This feature is facilitated by the availability of the static and dynamic registers which are under programmer control. Beginning and end addresses of variable size messages are located by use of static registers. Within a message, any item may be located by use of the "locate *n*th symbol" instruction. Relative references can be made to a series of consecutive items by using final contents of the registers, which are dynamic registers. When processing reference files using the simultaneous tape instructions, it may be desirable to have alternating areas for reading and writing. The ability to alternate data areas in this fashion evolves in the *501* computer the simple problem of transposing the contents of two of the static registers. In this way, extensive program modification or duplication is entirely avoided.

Programs can be written relative to a fixed point by utilizing static modifier registers. In order to float or transfer addressing of programs, it is necessary only to relocate the program in the memory and change the setting of the register. Programs can also be written completely self-relative for the computer. The programmed instructions are set up so that they are relative to the setting of the program register, the contents of which hold the high-speed memory location of the next instruction to be executed. This is the *P* register shown in Fig. 9. Programs written in this manner can operate without modification from any position in the memory.

Another over-all feature of the RCA *501* computer is the emphasis upon efficiency in tape handling. Sorting is improved substantially by the incorporation of reverse read, the use of locate *n*th symbol, four-character parallel transfer, modifier register operation, and by programmer access to all registers. Batching of messages after internal block sorts is itself facilitated by locate *n*th symbol, multiblock write, and directly addressable tapes.

The handling of transaction input and reference files is greatly improved by binary operations for sensing, random distribute and multiblock write for data handling, and the miscellaneous features for data shuffling such as are provided by reason of an individually addressable character memory and control symbol

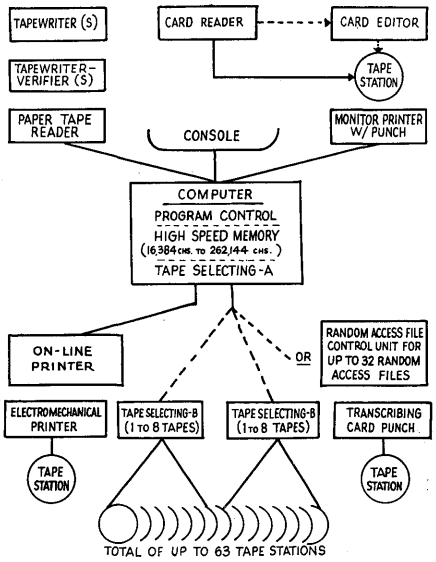


Fig. 8. An expanded RCA 501 system

logic as provided by the RCA *501* computer.

Accuracy Control

Extensive field experience with other RCA computing systems has permitted the adoption of a well-balanced approach to accuracy control in the RCA *501* system. Both equipment and programmed checks are used. Evaluation of the system logic, coupled with field experience pinpointed critical areas where hardware checks were required. Areas

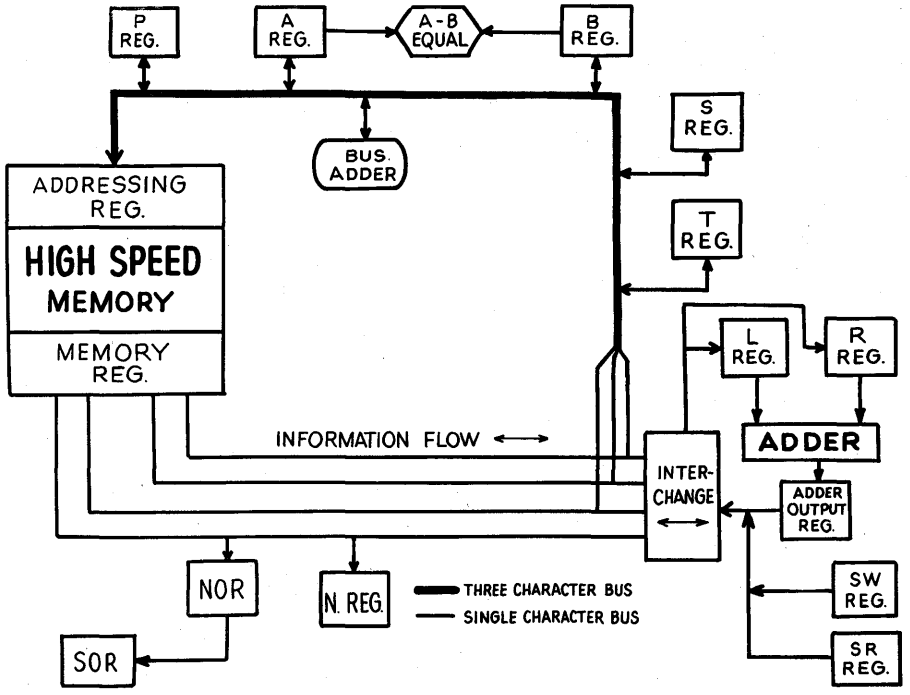


Fig. 9. The RCA 501 computer, block diagram

of a less critical nature may be program checked, thus avoiding unnecessary checking circuitry. But, in all cases, the information is checked in each unit as it flows from the system input to the system output.

Several types of equipment checks are performed in the system. Included in these types of checks are parity, all and only, double arithmetic and compare, tape data format, dual recording, and echo checking.

A seventh bit is added to the six-bit character which provides a parity count to insure against the loss of bits. In general, parity is checked whenever data enters or leaves a unit in the system. In the computer, parity checks are performed whenever data are internally transferred.

The all and only checking feature insures that all and only those characters to be processed are actually processed. These checks are found in equipments such as the electromechanical printer and the card transcriber.

The double arithmetic and compare is used in the computer to insure correct operation of the adder. This check involves the comparison of the arithmetic result against the complemented arithmetic result which also includes the complemented carry. This check is performed on a character by character basis, thus permitting variable length operands and result, without restriction.

The tape data format check insures the proper organization of data on tape by checking the sequence of special symbols as the information is written on, or received from tape.

Dual recording of magnetic tape information is used to insure reliable reading and enhance tape life. Two tracks representing each bit are displaced by approximately half the tape width and are serially connected, thus writing identical information in two tracks of the tape. Information read from either track of the tape is sufficient to provide the correct output.

The echo check feature verifies the parity of the tape-head writing current to insure correct receipt of data at the tape head.

Summary

In summary, the RCA Electronic Data Processing System is presently undergoing prototype tests in Camden, N. J. First production systems will be delivered in the last quarter of 1959.

This system is intended to provide for closer fitting of an equipment system to application characteristics. The system design provides for this fit at minimum user investment levels. It also provides for graduated levels up to a maximum efficiency level which is geared to fit large scale application requirements.

The major engineering advance of all transistor logic is directly related to an objective for substantially improving the reliability level of a data processing system with corresponding reductions in power, size, and over-all system cost.

The Univac® M-460 Computer

J. E. THORNTON
NONMEMBER AIEE

M. MACAULAY
NONMEMBER AIEE

D. H. TOTH
NONMEMBER AIEE

THE Univac *M-460* is a computer of advanced design from both the logical and engineering standpoints. It is the intent of this paper to describe its more interesting features.

The Univac *M-460* computer is a large scale high-speed general-purpose digital computer especially suited to rapid, complex processing of large quantities of data. The computer is a parallel, single address computer with fixed point, one's complement binary arithmetic. A computer word in the *M-460* is 30 binary digits in length with the provision for treating the two 15-bit halves independently, if desired. The memory section has a capacity of 32,768 30-bit words with a cycle time of 8 microseconds, and a read access time of 2.5 microseconds.

Special advanced logical features give the *M-460* a great deal of programming flexibility. From the standpoint of the number of instructions required for a given operation, the effect is much as if two-address logic were employed.

For purposes of discussion, the computer may be divided into four sections: control, arithmetic, input-output, and memory. The block diagram of Fig. 1 shows the interconnections of each of the four sections. The main registers of the computer are shown by rectangles. Some of the principal functions of each section are indicated within the round-edged blocks.

The control section includes a 30-bit instruction register U, a 15-bit program address register P, seven 15-bit address modification registers, B_1 through B_7, and a 15-bit auxiliary register R. A computer instruction is executed by first acquiring an instruction word from the memory location designated by the P register and then performing a number of operation sequences controlled by the instruction register U. Normally the content of the P register is advanced by one count for each instruction so that instructions are executed in numerical sequence. Program branching may be performed by changing the contents of the P register in jump instructions. A special feature of the control section permits program branching by another method. A skip designator, active in most instructions, permits an extra one-count advance of the P register if certain arithmetic criteria specified in the instructions are met. Thus the next sequential instruction may be conditionally skipped.

The organization of the instruction

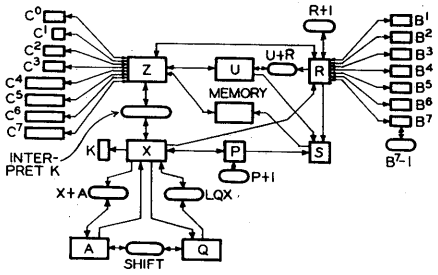


Fig. 1. Block diagram showing the interconnections of the four sections

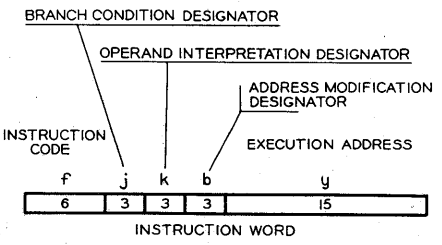


Fig. 2. Organization of the instruction word

J. E. THORNTON, M. MACAULAY, AND D. H. TOTH are with Remington Rand Univac, Division of Sperry-Rand Corporation, Saint Paul, Minn.

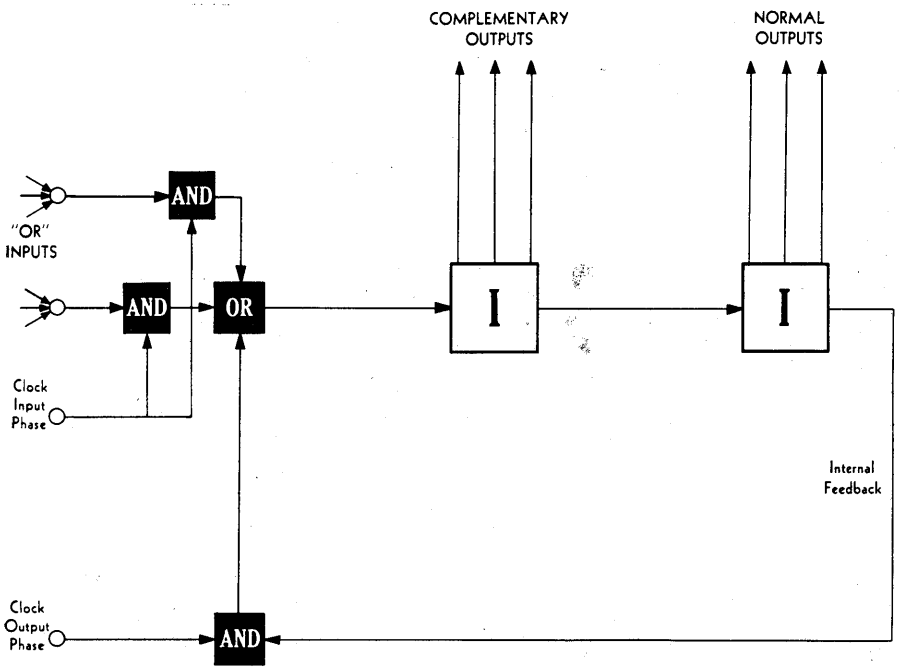


Fig. 3. Block diagram of the logical building block

word is shown in Fig. 2. There are 62 basic instructions designated by *f*, the upper 6 bits of the instruction word. These basic instructions are supplemented by three designators *j*, *k*, and *b* and a basic execution address *y*. In understanding the execution of an instruction, each of the following four main sequences should be considered:

A. Read next instruction
B. Read operand
C. Arithmetic
D. Store operand

Read Next Instruction. An instruction word is acquired from memory during the *A* sequence and is transferred to the *U* register. While the memory restore cycle is being completed, a modification is made to the lower half of the instruction. The basic execution address is modified by the addition of the content of one of seven *B* registers specified by the *b* designator of the instruction. This modification is complete by the time the instruction memory reference is finished. If the *b* designator is zero, no modification results.

Read Operand. An operand is obtained during the *B* sequence, the source of the operand being specified by the execution address as previously modified by the *b* designator. The operand interpretation designator, *k*, is active during the *B* sequence. Depending upon the the value of *k*, the operand may be the lower order 15 bits of the instruction word, the upper or lower 15 bits of a memory location, or the full 30-bit word contained in a memory location.

Arithmetic. Any arithmetic to be performed in the instruction is completed during the *C* sequence. No memory reference is required in this sequence.

Store Operand. If the result is to be stored, this storage action occurs during the *D* sequence. In the *D* sequence the operand interpretation designator is again used. Depending upon the value of *k*, the result may be stored in either half or all of a memory location, or the result may be transmitted to the accumulator or *Q* register.

The last designator active in the instruction is the branch condition designator *j*. This designator is used in three ways involving the *B*, *C*, and *D* sequences. First of all, most instructions utilize the *j* designator to specify whether the next sequential instruction is to be skipped as a result of the condition of the accumulator or *Q* register at the end of the current instruction. Second, the jump instructions utilize the *j* designator to specify whether a jump is to be performed depending on the condition of the accumulator or *Q* register. Finally, instructions requiring the use of a *B* or *C* register utilize the *j* designator to specify the particular *B* or *C* register to be employed.

The sixty-two basic instructions may be classed into 6 groups which include instructions involving data transfer, arithmetic, logical operations, jumps, indexing, and special operations. Instructions are basically single-address and are supplemented by the previously described designators. The special instructions include repeat, initiate input buffer or output buffer, compare, and masked comparison. Instruction execution times range from 11 microseconds for add constant to 130 microseconds for divide. Typical instruction time is 20 microseconds.

The arithmetic section of the Univac *M-460* comprises three 30-bit registers,

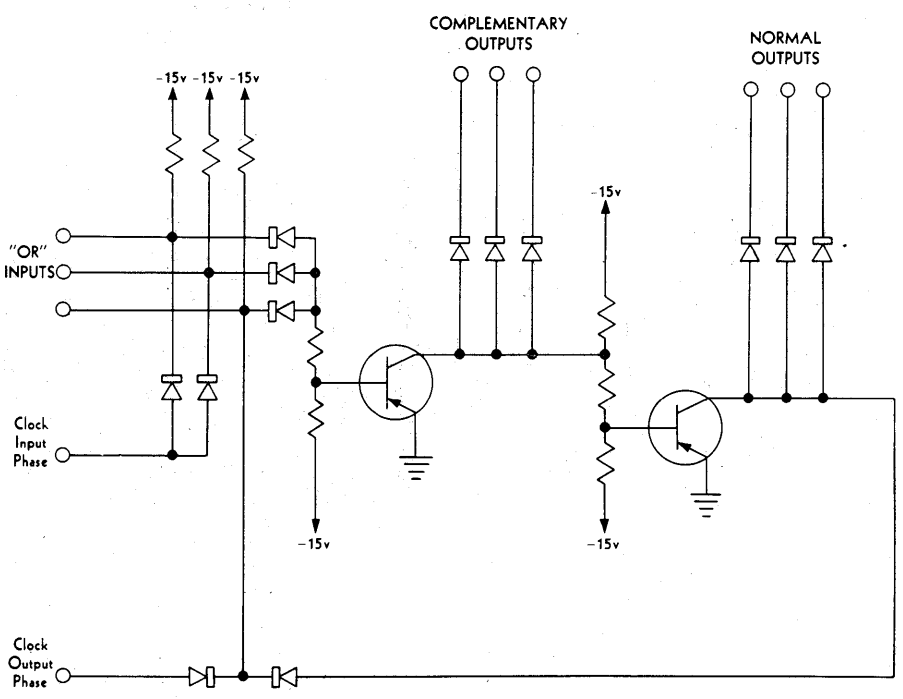


Fig. 4. Schematic of the logical building block

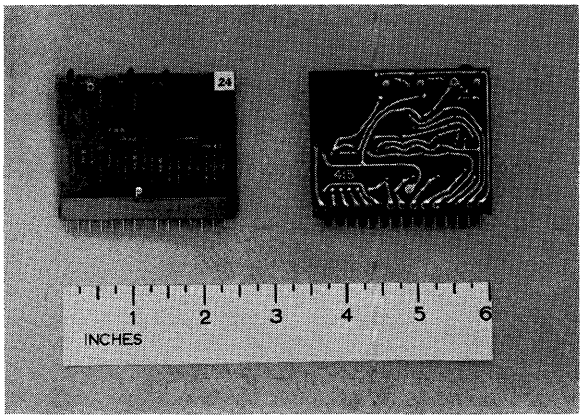


Fig. 5 (left). Logical building block components mounted on cards

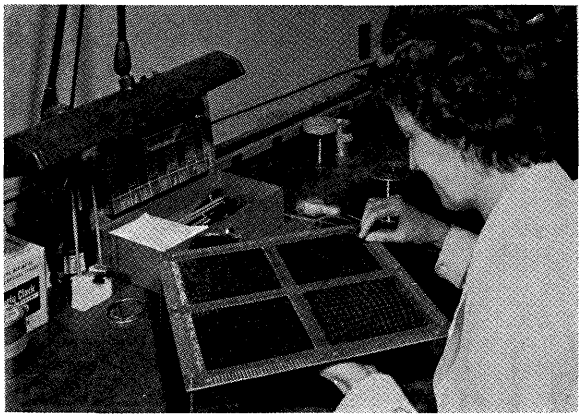


Fig. 7 (right). The memory section containing a coincident - current magnetic core memory

the accumulator, the Q register and the X register. Fixed-point binary arithmetic including add, subtract, multiply, and divide is performed in these registers. Double-length products obtained from multiplication instructions and double-length dividends before division instructions are held in the combined accumulator and Q register, used as one 60-bit register. The least significant 30 bits reside in the Q register, and the most significant 30 bits are held in the accumulator. The accumulator and Q register have high-speed shifting properties. The contents may be shifted right or left independently or as one 60-bit register. A shift of 30 places requires a total of $3^1/_2$ microseconds. Operands are always transferred to and from the arithmetic section through the X register.

The memory section is a coincident current, random-access ferrite-core memory of 32,768 words with a 30-bit data register Z and a 15-bit address register S. The other three sections of the computer have direct independent access to memory. Instructions, operands, and input-output data may be stored in any desired location in the memory. One complete memory cycle comprising a read and write operation is performed in 8 microseconds, the read access time being 2.5 microseconds. Since the memory employs destructive read-out, a read-memory reference is always completed by rewriting the information which was previously read. Similarly, a write memory reference is always begun by reading the contents of the designated memory location, thus clearing that location before writing. All memory references, either read or write, require a comp'ete memory cycle and the same sequencing mechanism controls both functions.

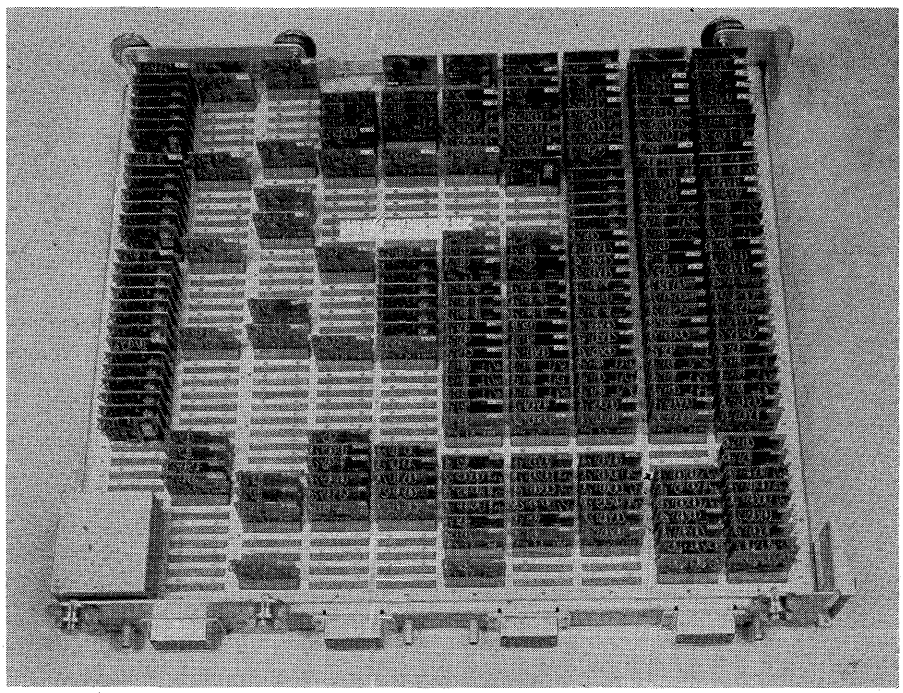


Fig. 6. Logical building blocks mounted on a flat chassis

An important characteristic of the Univac *M-460* is the versatility of its input-output section. Seven parallel registers are provided for the transfer of data to and from the computer. A total of 30 channels are available without additional switch gear; eighteen channels for input and twelve channels for output. One channel of each register may be activated at any one time. Furthermore, each channel may be connected to a number of external devices.

The concurrent operation of a number of external devices requires control communication as well as data communication. One of the seven registers of the input-output section is designated as the function register C_3. It performs the control communication function. For example, a transfer of data between an external device and the computer often requires deactivating all other devices on the assigned channel and activating the desired device. Other representative requests by the computer could be start motor, stop motor, advance, transmit, receive, etc. Requests may also come to the computer from the external device through the function register. Computer handling of such requests can be divided into two groups, scan and interrupt. In the first group, scan, the computer program examines the input channels of the function register periodically to determine whether a request has been made. In the second group, interrupt, the computer program is interrupted by means of an interrupt line provided on each of the input channels to the function register. This interruption causes the computer to halt its current program and to jump to a special interrupt routine. Special precautions are taken to prevent any conflicts between multiple interrupts, loss of current data, etc.

Another important feature of the input-output section is its ability to transfer data to and from memory independently of, and concurrently with, arithmetic

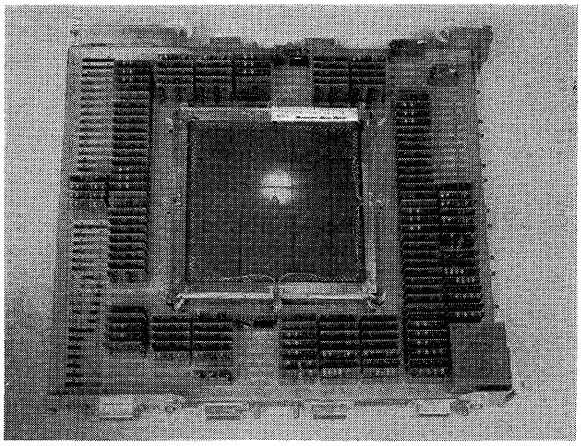


Fig. 8 (left). One of the five separate memory chassis

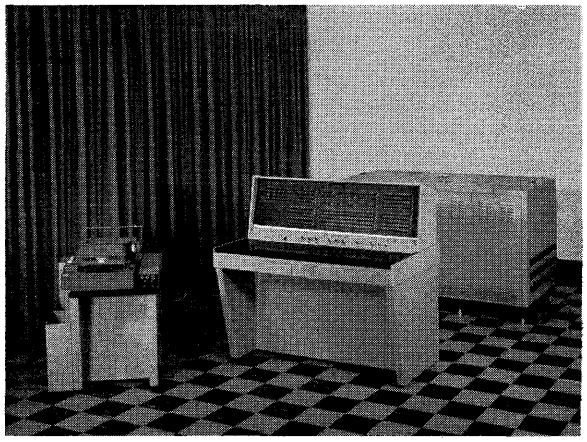


Fig. 10 (right) Maintenance console and monitor typewriter

operations. Data may be transferred by direct program transfer or by an independent buffer control, which is initiated by the program. The initiation of a buffer operation on a given input-output register establishes an initial memory address and a terminal memory address. Thereafter, data is transferred between memory and the external device at a rate controlled by the external device. The initial memory address is advanced for each transfer and compared with the terminal address. When the two addresses are equal, the Buffer operation is made inactive for that register until initiated once again by the computer program. Six such buffer operations may be operating concurrently using a built-in priority system, which allows the highest priority transfer to proceed at any instant.

A lockout system is employed to properly sequence requests for memory references since a buffer operation and the main computer program might request the simultaneous use of memory. The maximum input-output rates range up to over 50,000 words per second depending upon the communications structure. Since the words are 30-bits long, a maximum rate of over 1.5 million bits per second is possible.

An additional feature of interest is the inclusion of a real-time clock facility. Real time to the nearest millisecond is available to crystal oscillator accuracy.

It is surely apparent at this point that the Univac *M-460* computer is a powerful computer having high-program speed, a large high-speed memory, and a versatile input-output system. The remainder of this paper discusses the construction and design methods employed.

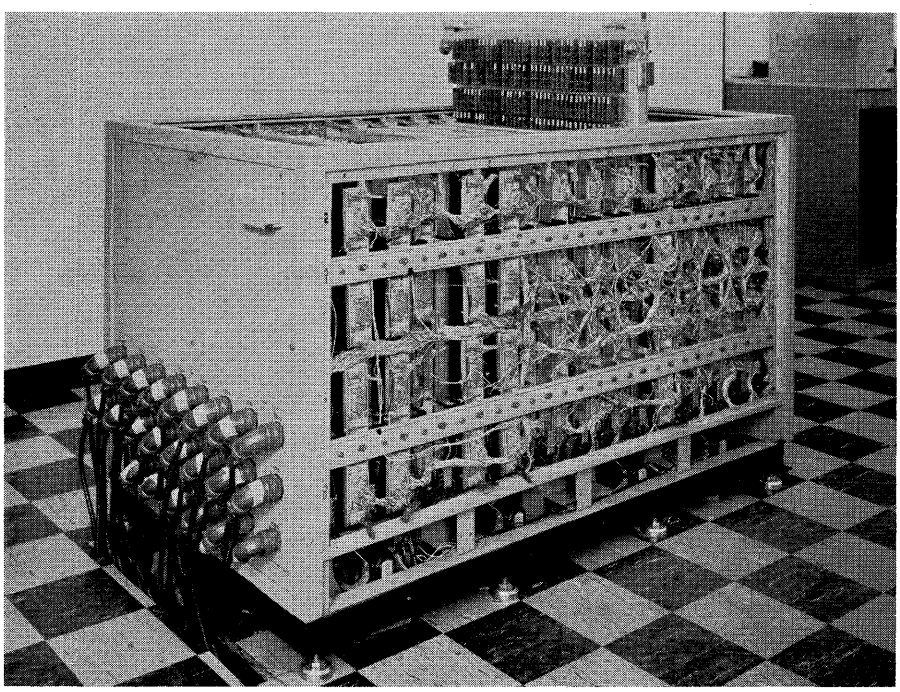


Fig. 9. The nine logic chassis and five memory chassis mounted in cabinet

The logical sections of the *M-460* are implemented with transistor building blocks shown in block diagram form in Fig. 3. Each building block is made up of two inverters, which provide the normal and complementary outputs, plus "and-or" logical inputs to the first inverter. All logical decisions are gated by a square-wave clock at the entry to each building block. A feedback path is provided from the second inverter to the first and is gated by the opposite phase of square-wave clock. The operation of a building block follows a continuous 2-phase sequence: 1. the input phase, when the logical decision is made, and 2. the output phase, when the result is captured and held for output. In Fig. 4, the schematic diagram of the logical building block shows the two transistor inverters and the diode "and-or" logic. Each phase of the clock is 0.5 microsecond in duration giving an information rate of 2 megacycles.

Mechanized design methods, described in a previous paper, were used with this building block. These mechanized design methods made possible a very short design and development period. For one thing, the *ERA 1103* (Univac Scientific) computer was programmed to simulate, verify, and correct the logical design of the *M-460*. Even more important, the *1103* was used to provide optimum building block placement, complete wiring tabulations, chassis maps, cross-reference tabulations, and parts lists.

The logical building blocks are constructed of fibre-glass cards with an etched circuit of copper foil bonded to one side. The dimensions of the cards are 2 inches by 2.5 inches. As shown in Fig. 5, components are mounted on the side of the board opposite the wiring. A 15-pin connector at the base of the card allows each building block to be easily removed. These logical building blocks are mounted on a flat chassis, shown in Fig. 6, whose

dimensions are 28 inches by 30 inches. Each chassis contains its own power supply and 2-phase clock "slaves."

The memory section contains a coincident-current magnetic core memory made up of ferrite memory arrays, shown in Fig. 7, and transistor building blocks. Memory current drive of 0.4 ampere is obtained by using high-current germanium transistor switches. The complete memory is divided into five separate memory chassis, one of which is shown in Fig. 8. The memory chassis are identical in size to the logic chassis. Each memory chassis contains its own power supply and is completely interchangeable with the other four memory chassis. Considerable effort was devoted to the orientation of memory arrays and transistor building blocks on this chassis in order to obtain an optimum electrical arrangement.

The nine logic chassis and five memory chassis are mounted vertically in a "deep freeze" cabinet as shown in Fig. 9. The total volume of this cabinet is less than 2 cubic yards. The interconnections between these chassis are made in the two side planes of the cabinet. The external equipment cables connect into the two end planes. Chassis are completely removable for test and repair. Standard auxiliary units shown with the computer in Fig. 10 are the maintenance console and the monitor typewriter. (The high-speed paper tape reader and punch are not shown). Other auxiliary units such as the magnetic tape and magnetic drum are presently being developed for use.

Many test programs have been operated, ranging from simple service and diagnostic tests to more complicated system operation with other equipments. Also, the *M-460* has carried on a "conversation" with a magnetic-switch test computer, played musical tunes through its input-output section, and has performed a number of other unusual feats. The performance of the Univac *M-460* leaves no doubt as to the excellent future of solid-state digital computers.

Reference

1. A Progress Report on Computer Design, S R. Cray, R. N. Kisch. "Proceedings of the Western Joint Computer Conference," *AIEE Special Publication T-85*, Feb. 1956, pp. 82–85.

A Special-Purpose Solid-State Computer Using Sequential Access Memory

W. A. CORNELL
NONMEMBER AIEE

Synopsis: A feasibility model of a new, special purpose computer using diode-resistor logic and high-speed pulse amplifiers has recently been completed. Magnetic-core logic circuitry is used to implement one part of the program generation function. Memory is provided by quartz ultrasonic delay lines and associated transistorized drivers and amplifiers. This paper describes the operation and characteristics of the computer and contains a description of a special "data simulation" technique which was used to test the computer in the laboratory.

THE versatility of large-scale general purpose computers in meeting the increasing needs of business and industry for data processing commands considerable attention and respect. It is, therefore, fairly easy to keep abreast of new developments in these large machines. It is often more difficult to remain well informed about the large variety of smaller computers which are being designed for very special applications. This paper describes one such machine which has been completed at Bell Telephone Laboratories as part of a research program carried on under contract with the Bureau of Ordnance of the United States Navy.

The Prediction Computer, as it is called, is a feasibility model of a special purpose, digital computer which uses diode-resistor logic and high-speed pulse regenerative transistor amplifiers. Vacuum tubes are used only for clock power generation and in the display circuits. Magnetic-core logic circuitry is used to implement one part of the program generation function. A sequential access memory is provided by quartz ultrasonic delay lines and is entirely solid state including the radio-frequency circuitry. This is a simple and compact form of memory and the usual disadvantages of access time are not troublesome because of the special character of the problem and the particular logical organization of the computer.

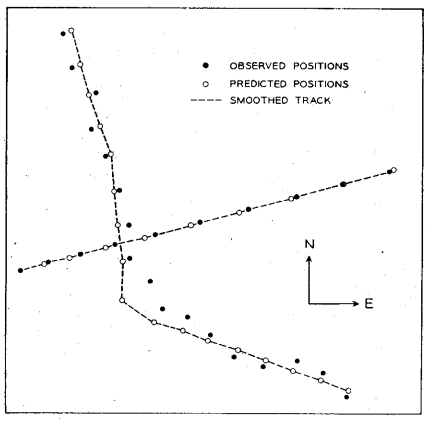


Fig. 1. Exemplary track-while-scan problem

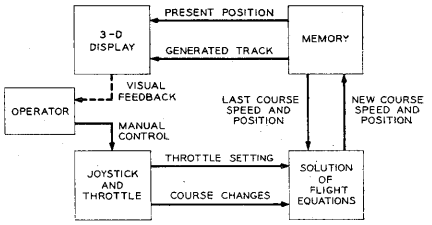


Fig. 2. Simulation flow diagram

Consider for a moment the special purpose for which the prediction computer was designed. Investigation of the application of high-speed digital techniques to a track-while-scan problem was desired. Successful analog systems had been built which provided analog tracking equipment on a per-track basis. For large numbers of tracks, a considerable reduction in equipment would be possible by time-sharing high-speed digital equipment. This was a research program involving three steps: the formulation of a systems proposal, research on circuitry and devices, and a demonstration of feasibility under laboratory conditions. In concept the problem is simple. A search radar provides intermittent position data in three dimensions on a large number of aircraft. These data must be operated on so as to generate a substantially continuous track for each individual aircraft. To do this data will be accepted from the radar in the form of one observation of the position

W. A. Cornell is with Bell Telephone Laboratories, Inc., Murray Hill, N. J.

of each aircraft per radar antenna scan. Each aircraft must then be tracked with sufficient accuracy to distinguish it from every other aircraft and to identify it to other equipment as required. This process may be thought of as involving two distinct operations: first, the association of new radar data with the appropriate stored track and second, the calculation of rates, prediction of future position, and generation of a continuous track by interpolation or extrapolation. The first operation is essentially a sorting problem which will not be discussed further except to say that a system design for a sorter was formulated which influenced the design of the prediction computer to a considerable extent. The second operation is performed by the prediction computer and can be illustrated by reference to Fig. 1. Here is represented successive observations of the positions of two aircraft by sequences of dots. These observed positions are perturbed by noise. The prediction is made on the assumption of unaccelerated, straight-line flight with appropriate weighting of the observed data and the last predicted position. As a conse-

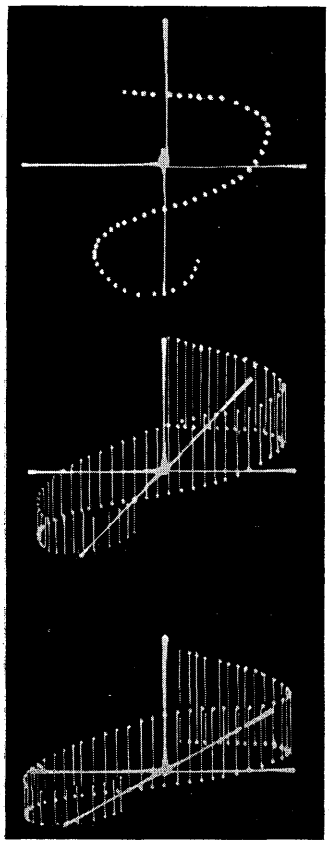


Fig. 3. Output display for typical flight path

Top: Plan view
Center: Oblique view θ, 45 degrees
Bottom: Oblique view θ, 30 degrees

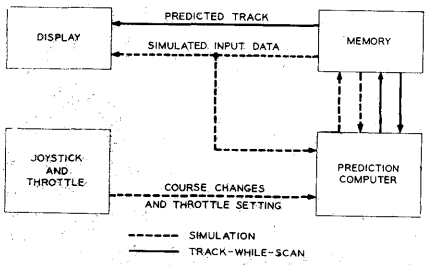


Fig. 4. Track-while-scan flow diagram

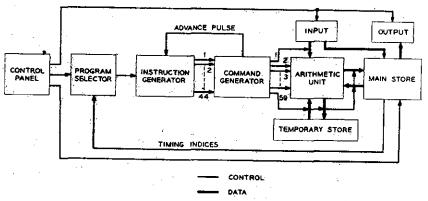


Fig. 5. Prediction computer block diagram

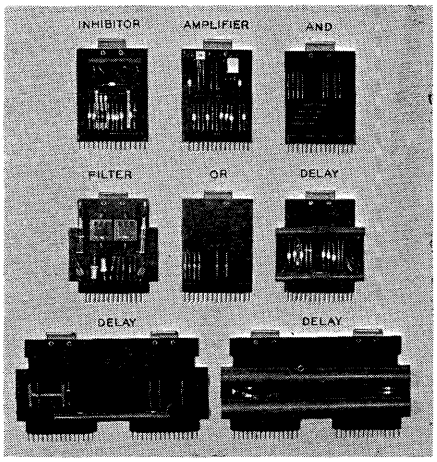


Fig. 6. Circuit packages

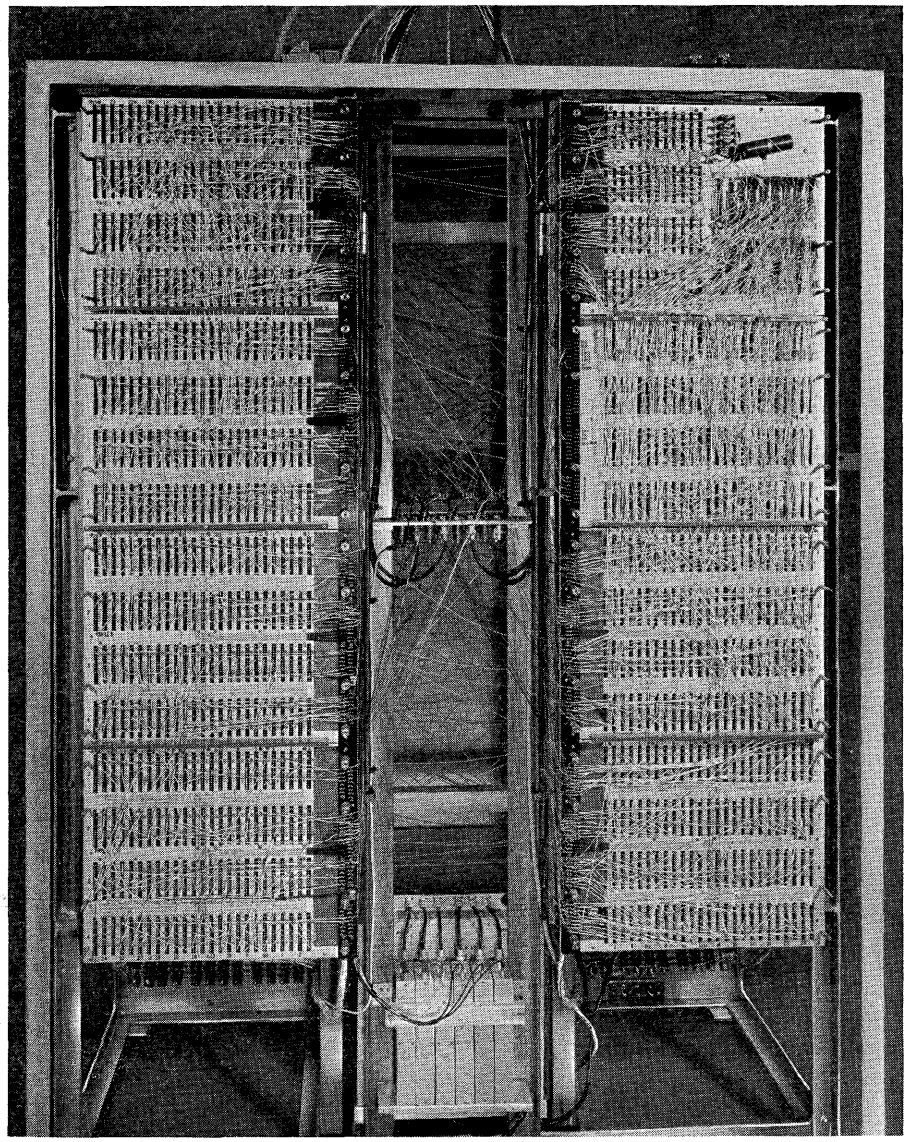


Fig. 7. Prediction computer

quence smoothing is introduced and the system has a certain response time to changes in course or speed of the aircraft.

It may be instructive to consider how the requirements of the problem influenced the design of the computer. For 50 simultaneous tracks, the computer must accept input data from the sorter at rates of the order of 35,000 bits per second. The arithmetic operations required for co-ordinate conversion, rate computation, smoothing, and for prediction of future position are reasonably simple; a program of about 100 steps is adequate. However, these computations must be repeated for each track every radar scan and must be done in real time. Memory requirements are moderate, less than 1,000 words being adequate for data on 50 tracks. The over-all computing speed required is high, but if one can operate at pulse rates in excess of one megacycle, one can still use a serial arithmetic unit with the attendant saving in equipment over parallel operation. The specialized repetitive program, which need not be changed, together with the serial arithmetic unit led to the choice of a serial memory (this was also in keeping with the proposed design of the sorter). These and other considerations led to the design of a serial, synchronous machine operating at a 3-megacycle bit rate and having the following general characteristics:

1. Word length is composed of 12 binary digits with a fixed binary point and scaling such that all numbers lie in the range from -1 to $+1$. Negative numbers are expressed as true complements.

2. Approximately 1,000-point contact transistors are used as pulse regenerative amplifiers along with about 12,000 germanium diodes for the diode-resistor logic.

3. 636 words of internal memory are provided by two ultrasonic delay lines using transistorized radio-frequency circuitry. Electrical delay lines are used as temporary storage in the arithmetic unit and as access circuitry for the ultrasonic memories. An additional 318 words of ultrasonic delay line memory are used as an input buffer storage.

4. Programs are wired into the machine and automatic sequencing of programs is provided for. Program steps are controlled by a 168-step magnetic-core stepping switch.

5. The arithmetic operations provided are addition, subtraction, multiplication, division, shift, and round-off. Addition time is 4 microseconds and multiplication time is 48 microseconds.

6. Major control loops contain circuits which are single error correcting in order to prevent transient errors in one program from affecting the succeeding programs.

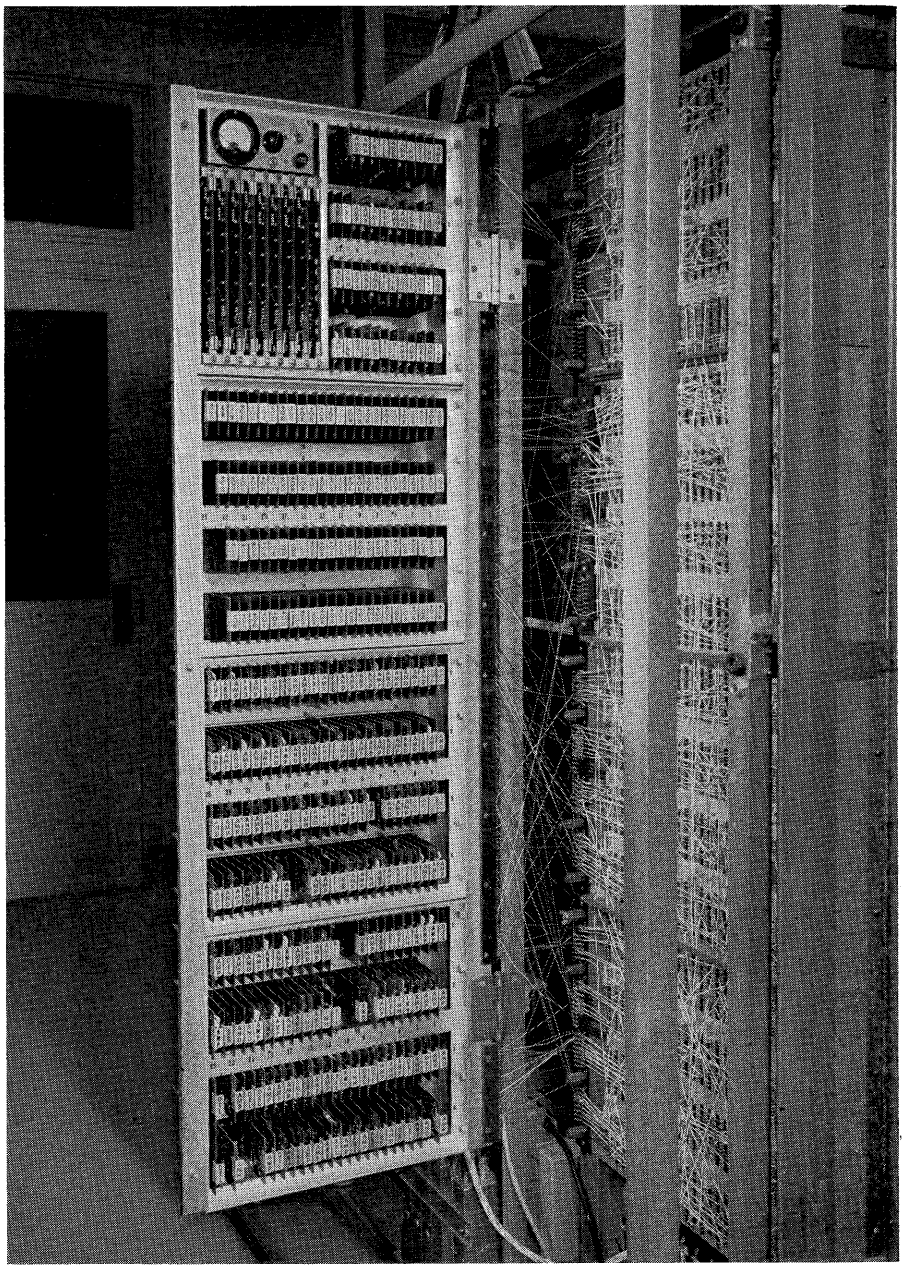


Fig. 8. Oblique view of prediction computer

7. Output in the form of aircraft tracks, is displayed on a cathode ray tube which has been arranged to present either an oblique projection of the three dimensional data or a plan view in ground co-ordinates. For test purposes, output data can also be recorded digitally on a pen recorder.

The operation of the prediction computer will be described in two steps. First, how such a machine would operate as a part of a complete track-while-scan system will be briefly discussed and then how it was tested and demonstrated in operation under laboratory conditions will be discussed.

In a complete track-while-scan system, the prediction computer has stored in its memory the predicted present position of each aircraft being tracked and also the predicted rate for each aircraft. Input data is supplied in the form of a new observed position for each aircraft. These data arrive at more or less random times but each observed position carries with it the identity of the track to which it belongs. The computer calculates a new rate on the basis of the observed position, the predicted position from memory, and the predicted rate. It then continually adjusts the stored position on the basis of the stored rate until a new observed position is supplied. The output of the computer consists of these stored values of present position for each aircraft being tracked. These may be displayed visually for evaluation. The operator's job is to monitor the

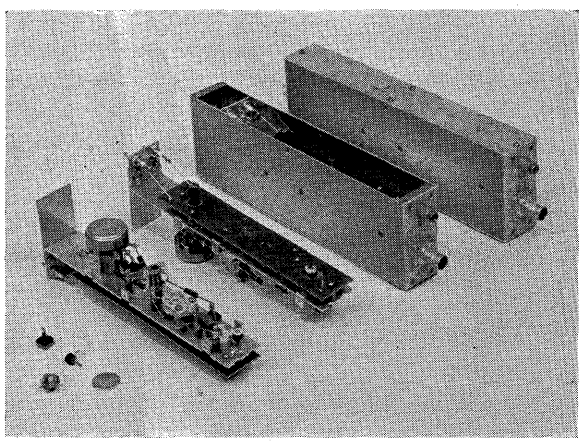


Fig. 9 (left). Radio-frequency transmitter for memory

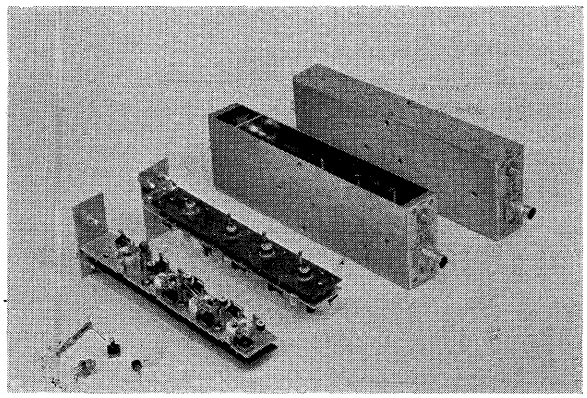


Fig. 10 (right). Radio-frequency receiver for memory

operation of the machine by watching the display. He may initiate new tracks, terminate tracks, and change the operation of the computer with respect to any individual track if this seems advisable.

The operation of the prediction computer under laboratory conditions is somewhat different because of a problem which often arises when one builds a piece of special purpose equipment which is part of a large system: namely, how to test and demonstrate the machine without recourse to large amounts of auxiliary equipment? In this case it was wished to demonstrate the prediction computer without the use of a radar and without being forced to build a sorting mechanism and expensive display equipment. Two approaches were available. A high-speed data generator which could be attached to the input of the computer could be built or the input of the computer could be connected to an auxiliary memory which could be preloaded at low speed and then read out into the computer at the computer pulse rate. The latter approach was chosen. Since a memory of reasonable size could only keep the computer busy a small fraction of the time, a substantial amount of free time existed between computer programs. Why not use some of this free time to let the computer calculate its own input data? Accordingly a program was wired into the machine which would permit it to simulate the flight of an aircraft and thereby generate a sequence of positions which can be used as input data. This process is shown in the flow diagram of Fig. 2. The operator is provided with a joystick and throttle controls which permit him to control the course and speed of a hypothetical aircraft. At regular intervals the computer samples the output of these controls and proceeds to solve simplified equations of flight to compute the speed, course, and position of the simulated aircraft. At specified intervals of time the computer sends position data which it has calculated to the input-buffer store section of its memory. The result is a series of discrete positions which can subsequently be used as input to the computer in solving a track-while-scan problem. The entire contents of the buffer store may be used to store successive positions of one aircraft, or alternatively the operator may cause a lesser number of positions to be stored for each of several flight paths that he wishes to generate. The operator is aided by a display which presents to him not only the current position of his aircraft but also the past history of the flight path as stored in the computer memory. Fig. 3 shows how a typical flight path would appear on the display. Certain constants are built into the program which limit the maximum speed in level flight and which cause speed to be a function of angle of climb or dive. Similarly the maximum turn acceleration is limited so that the turning radius for a given joystick setting is a function of the speed.

Two features of the process just described should be emphasized. First, this is a laboratory device for testing and demonstrating the computer. The joystick, throttle, and attendant flight path generation have no counterpart in an operational track-while-scan system. Second, the operator referred to is, in reality, the man who is demonstrating and testing the computer. He performs different functions from the operator of an operational track-while-scan in that his prime function is to assist in generating a problem and not to monitor the solutions to the problem.

The operation of the prediction computer in the laboratory can be carried on in several different ways, two of which are of particular interest. In the first of these the input store is loaded with data-representing position reports on one or more aircraft from successive radar antenna scans. This can be accomplished by means of the simulation program or manually if desired. After the store has been loaded, the computer calculates a smoothed track, the most recent position of which is displayed on a cathode ray tube or recorded digitally on the pen recorder or both. In the second mode of operation, the computer automatically selects between the simulation program and the tracking programs so that in effect it calculates its own input data as it proceeds with the tracking problem. Fig. 4 is a flow diagram for this operation. The operator "flies" his aircraft with the help of the computer which alternately solves the flight equations and then uses these results as input to the tracking problem. Both solutions are presented to the operator who can then watch both the position of "his own aircraft" (which represents radar position data supplied to the computer) and the computer's attempts to track him as he maneuvers. This use of a real time computer to compute synthetic input data for test purposes suggests that such a scheme might be useful in an operational machine. It would provide a ready means of checkout when the machine is off line, and such a program could also be interspersed with the normal operating programs when the machine is operational. Since the correct solution can be predetermined by substituting constants for the variable outputs of the joystick and throttle, a monitoring means is made available which can detect any malfunctions.

There are a total of five programs wired into the computer. Two of these are diagnostic routines for test purposes. A third is the simulation program and the remaining two programs implement the track-while-scan function. The length of each program is made equal to a multiple of the circulation time of the

memory so that at the end of each program the results can be immediately returned to storage. Furthermore, the next track to be processed can now be read from the memory. This interplay between arithmetic operations and memory access is effected by deriving the timing of the programs from index marks in the circulating memory and serves to minimize time lost in searching the memory.

Fig. 5 is a block diagram of the computer. Heavy lines indicate data paths and light lines indicate control signals. Control signals progress from left to right becoming more detailed as they pass through the several units. The highest level of control is exercised by the program selector which receives control information only from the setting of controls on the control panel. Its function is to select the proper wired in program, remember what program is in process, and change to another program at the appropriate time. It also receives timing indices from the memory which enable it to synchronize the arithmetic operations to the memory cycle. It is a multistate circuit consisting of several single error-correcting memory cells and logic circuitry for the generation of action signals to other parts of the computer. These signals are a function of inputs and the current internal state. The instruction generator, under the supervision of the control unit, generates the sequences of instructions such as multiply, read from memory, and store, which comprise each of the programs. It consists of a 168-step magnetic core counter for ordering the successive steps of a program and a translation matrix for each wired program. This matrix converts each step of the counter into the appropriate instructions for that step of the selected program. At the output of the instruction generator, signals are in the form of a single pulse on the appropriate lead. Conversion of arithmetic and memory transfer instructions into detailed switching orders such as "open gate A," "read sign," and "shift partial product" is accomplished by the command generator. This unit is made up of a number of timing pulse generators and a logic network. Input instructions in the form of a single pulse on one of many leads are converted into sequences of pulses called commands on several leads at the output. Just prior to the completion of each command, a pulse is sent back to the instruction generator to cause the counter to advance one step. Inputs to the arithmetic unit consist of detailed action signals from the command generator and data from the memory and from the input to the computer. This unit performs serial addition, subtraction, multiplication, and division. Multiplication is handled by a modification of the Booth algorithm and division is carried out by a nonrestoring process. Provisions are made for round-off, division limiting, and determination of absolute values. "Scratch pad" storage in the form of electrical delay lines provides for storage of intermediate results between memory access periods.

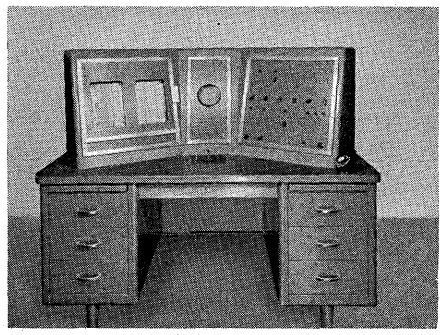


Fig. 11. Control console

The diode-resistor logic comprises the three logic operators "and," "or," and "inhibit." These are built on printed circuit cards as are the pulse regenerative amplifiers. The pulse regenerative amplifiers[1,2] provide output pulses 0.16 microsecond long at about five volts and can drive loads requiring up to 17 milliamperes. The use of pulse-type logic is advantageous in permitting more rapid turn-off of the point contact transistors which was a problem because of carrier storage. The low-voltage low-impedance circuitry permits wiring between panels to be done without the need for coaxial cables or special driving amplifiers. In addition to the logic and amplifier packages, various electrical lengths of delay are provided. Three different packages are used to accommodate delay lines from a fraction of a bit to about five bits in length. The basic circuit packages are interconnected so as to form more complicated switching circuits such as half-adders, memory cells, and gates. Fig. 6 shows most of the cards which are used in the computer. The filter is typical of a few special purpose circuits. It consists of filter condensers for the power supply buses and tank circuits to correct the phase and amplitude of the 4-phase 3-megacycle clock supply to the pulse regenerative amplifiers. Other special purpose circuits not shown are decoders for converting the digital output to analog form for the display and pulse stretchers in order to convert the high speed pulses to a form suitable for operating the magnets of the pen recorder. A maximum of 100 cards can be mounted on a panel 17 inches by 14 inches by 4 inches. Fig. 7 shows a view of the computer which consists of twenty-three such panels mounted in six bays. The wiring in the open center portion is signal wiring between panels and is run so as to minimize crosstalk between leads. The memory is located at the bottom of the center portion. The six rectangular boxes contain the radio-frequency circuitry and immediately behind them is the oven which contains three quartz delay lines. Fig. 8 is an oblique view showing one of the bays opened on its hinges. The magnetic core counter occupies the left-hand side of the top panel on this bay. Immediately below it is the program "or" matrix which constitutes the wired in programs. The delay lines are multireflection polygons using barium titanate transducers.[3,4] Figs. 9 and 10 are the transmitter and receiver used with the memory. The former is essentially a pulsed oscillator which generates 0.16-microsecond bursts of 15-megacycle carrier. The latter is a broadband (12-megacycle) amplifier, centered at 15 megacycles, plus detector and pulse amplifier. All transistors are diffused base germanium and the receiver gain is about 35 decibels.

The control console is shown in Fig. 11. To the right is the control panel containing switches which permit the selection of programs and operating modes. Means are provided here for manually altering the data in the memory and for operation of the diagnostic test routines. Lamps are provided for monitoring the operation of the computer. The output display oscilloscope is in the center position. This is a standard laboratory oscilloscope and circuitry for generating the special display is located at the rear of the console. The 40-pen recorder is shown on the left side of the console. This instrument is useful in recording selected data from the computer memory. If a problem with a predetermined solution is given to the computer, the results of the computer's action at various stages in the computation can be checked. The circuitry necessary to select a required memory position, read out at a 3-megacycle rate and convert to signals capable of operating the pen magnets is located on the main frame of the computer. On the desk top directly in front of the oscilloscope are the throttle

and joystick controls for the flight simulation program.

Since its completion the computer has been operated for a period of about 3 months on a 24-hour a day basis. Component failures have been few and reliability has been good. Although transient errors have occurred occasionally in individual programs, the provision of error correcting circuitry in the higher levels of the control has prevented propagation of errors from one program to the next. Information has been recirculated in the memories for several days with no errors. The simulation routine has not only been useful as a demonstration tool but also as a rapid means of checking the operation of the computer. The problem solved by the prediction computer is a very special one in which similar operations are performed many times on a fairly large amount of input data; the simplicity of the final equipment and its performance under test have indicated the advantages of the rather special design employed.

References

1. Transistor Amplifiers for Use in a Digital Computer, **Q. W. Simkins, J. H. Vogelsong.** *Proceedings*, Institute of Radio Engineers, New York, N. Y., vol. 44, Jan. 1956, pp. 43–55.

2. A Transistor Pulse Amplifier Using External Regeneration, **J. H. Vogelsong.** *Ibid.*, vol. 41, Oct. 1953, pp. 1444–450.

3. Characteristics of Ultrasonic Delay Lines Using Quartz and Barium Titanate Ceramic Transducers, **J. E. May, Jr.** *Proceedings*, National Electronics Conference, Chicago, Ill., vol. 9, 1953, p. 264.

4. Low Loss, 1,000 Microsecond Ultrasonic Delay Lines, **J. E. May, Jr.** *Ibid.*, vol. 11, Mar. 1956, pp. 786–790.

Discussion

Chairman Estrin: The first question is addressed to Mr. Hudson, and comes from Mr. E. Schreiner, Olivette: "Are the control matrix cores reset after each program counter step? If so, how?"

James Hudson: The individual amplifiers are latching-type amplifiers, and they respond only to single polarity signals. The sense winding is bipolar in order to reduce the noise due to the shuttles in the core. The sense winding is wound first, then the other latching amplifier has to wait for reset in order to be set up. This means that the core is brought back by the application of both the instruction line and the program counter lines. Then both lines are removed as it carries the core back so that either at one time or another the latching does meet the proper polarity.

Chairman Estrin: We have another question for Mr. Hudson from Mr. Schreiner: "You used the words 'operation time' and 'word time'. What is the difference in meaning and what are these times?"

James Hudson: The "operation time" may have been referring to the complete command; the "word time" is essentially the fixed-cycle memory-read logical operation. The control at the beginning is to establish what memory cell, that is, what logical operation will take place.

Chairman Estrin: The next question is from Mr. R. A. Kudlich, Bell Telephone Laboratories: "Are all block times equal? If so, how long is a block time?"

James Hudson: For all internal operations without input-output the block time is a fixed time. There are 24-pulse times in the word, each have $2^1/_2$ microseconds or 60-microsecond words.

Chairman Estrin: The first question is for Mr. Poorte, and is from Mr. C. W. Einolf, IBM: "RCA publicized a 155-card-per-minute punch development some months ago; is this the punch being used, or are you using a standard IBM punch? Also, is the *501* meant to replace the Bizmac 2?"

Glen E. Poorte: The punch you saw is the RCA development, it is not an IBM punch. We expect in the future to accelerate speed to 150 words per minute, at the present time we are operating at a hundred.

Number two, the *501* is not meant necessarily to replace Bizmac 1 or 2. It does, however, reach down farther into the medium-sized scale, as you can see from the complement.

Chairman Estrin: The next question is for Mr. Poorte from Stephen Young, Burroughs Corporation: "For simultaneous magnetic-tape read and write operations, is a buffer used for the input data, or is the write operation interrupted in the line printer? Is a specially edited magnetic tape required, and must the tape block contain 120 characters?"

Glen E. Poorte: To answer the first question regarding the simultaneous magnetic tape and reads, the procedure here was glossed over quickly due to time elements. There is a small 4-character buffer which accepts characters on the input for magnetic tape to accept information up to 33 kc, which is operating range; and for the tape output there is another small 4-character buffer. We transferred the four characters simultaneously on the output buffer, and we write each character at the appropriate time. In the meantime, information from the input, which has already been received from the 4-character buffer, can be transferred into the high-speed memory; this occurs simultaneously. To compute the same it requires an interrupted 8-memory cycle to accept information from the input, or to write information to the magnetic tape.

With regards to the line printer, the magnetic tape is not necessarily especially edited tape. It can be accomplished by plugboard and paper tape loop. The tape that is referred to as a tape block is not necessarily true. We refer to it as a message in the RCA system, and it may contain any number of characters, although 120 characters are all that will be printed.

Chairman Estrin: The next question if for Mr. Thornton from S. S. Rideout, Brookhaven National Laboratories: "Please discuss further the role of Univac Scientific in the logical design of *M-460*. What was actually done on the computer?"

J. E. Thornton: This was discussed at this conference 2 years ago. What we do is to formulate the building block structure of the computer in algebraic form and enter it into the Univac System verify format, that is, electrical and logical format, and simulate the operation phase by phase in the computer. Finally, we gather all the data after the desired design has been completed, and formulate with tabs and placement, the cards in the chassis. We do not do any logical deductions, the designer still does this job.

Chairman Estrin: We have a 3-part question from Neal J. Dean, Booy Allen and Hamilton for Mr. Thornton: "Is there a separate buffer for each input-output device. Also, how large is the buffer?"

J. E. Thornton: No, there is not a separate buffer. Each of the registers is under the control of one buffer control. As for the term "buffer" itself, we have a conflict; most people consider a buffer as a section of the memory, but as referred to here, it is the control of the memory having to do with transferring words in and out of the computer. The buffer here is considered a portion of the memory.

Chairman Estrin: "Is this device presently operational, and for what applications?"

J. E. Thornton: This device was designed and constructed for the United States Defense Department. It is classified and it is presently operating.

Chairman Estrin: "Are there plans to market this device; when, and for what approximate price?"

J. E. Thornton: I must refer you to our contract department for this information.

Chairman Estrin: We have another question for Mr. Thornton from Paolo Ercoli, I.N.A.C., Rome, Italy: "What kind of marginal checking do you perform on *M-460* computer? Have you any figure of reliability on transistor replacement rate?"

J. E. Thornton: The only marginal checking we perform is on the variation of the power voltage supplied to the computer itself, which in turn varies the d-c voltages of the chassis. We have used this very, very little, and, as a matter of fact, we have not conducted any systematic routine maintenance at this point.

As to the transistor reliability or replacement, most of the work at this point is in program debugging, and as a result we have not replaced very many transistors at all. I do not think that this is a consistent result of the maintenance. However, wherever we refer back to previous models of the computer or to our technique for routine maintenance and replacement of the various diodes we are successful, since the circuit used is very, very similar.

Chairman Estrin: The next question for Mr. Thornton is from Mr. S. Shohara, Hughes Aircraft Corporation: "What are the temperature limits for which the memory was designed? Also, if the environment is controlled, what variation is allowed on drive currents?"

J. E. Thornton: The temperature limits for which it was designed are military limits.

Chairman Estrin: S. Shohara further asks, "Is there a difference in the design of the memory?"

J. E. Thornton: There is not a real difference. In this case we do enclose the stack.

We have observed a variation of drift currents on the order of $\pm 5\%$; and together with the variation of voltages at the center currents of $\pm 10\%$, approximately.

Chairman Estrin: Here is a question from J. Cornell for W. A. Cornell: "Must the memory recycle period bear any relation to the antenna scan period?"

W. A. Cornell: No, there is no connection between the memory recycle period and the antenna scan period. The memory recycle period is about one millisecond, and the antenna scan period assumed here, was 4 seconds.

Chairman Estrin: From Mr. R. K. Richards, for Mr. Cornell: "Why were point-contact transistors with pulse-type circuits and diode logic chosen rather than the TRL circuits described yesterday by your colleague, Mr. Finch?"

W. A. Cornell: This is history. TRL was not invented then. The circuitry goes back a few years when the only high-speed transistors available were 3-megacycle pulse rate, or point-contact, and, having decided to go ahead to use them at high speed, pulse-type techniques were used. We got minimum trouble from the carrier storage by only turning them on for 6 microseconds.

Chairman Estrin: This question comes from John Poivinen, General Electric Company, for Mr. Poorte: "Did I understand correctly that the *501* address codes are alphanumeric?"

Glen E. Poorte: The address code uses

alphanumeric information. I cannot think of it in terms of alphanumeric however. Each operand, each character in the operand is from 00 to 77, and it can represent alphanumeric characters. Therefore, the address is actually made up of 16 possible alpha characters, 8-RCA and 8-*501* characters.

Chairman Estrin: Mr. D. R. Pickering, California Institute of Technology and Jet Propulsion Laboratory asks Mr. Poorte: "In the RCA *501* computer, what is the average time required for addition, subtraction, multiplication, and division? What type of transistors are used?"

Glen E. Poorte: With regard to the first question, it is difficult to answer. We do use complete variable data lines for addition, and subtraction would be essentially the same, eliminating the end-run area. A typical example would be to take six characters in each operand, and it would be somewhere in the neighborhood of 350 microseconds.

With regard to division and multiplication, I would not attempt to keep time on this. However, it is in a series of additions for the multiplication and shifts. For division it is a series of subtractions and shifts.

Chairman Estrin: Here is a question for Mr. Poorte from Mr. S. H. Unger, Bell Telephone Laboratories: "How many inputs are there to each 'nor-gate' package?"

Glen E. Poorte: The "nor-gate" package is a 2-input device. I told you that we use them for power drivers and inverters; and you will notice in this particular case that we tied two inputs together. There is a second package that I did not mention in the paper which is essentially the same circuit with a single input.

With regard to the type of transistors used, we are using the *2M583*, an RCA transistor, not necessarily because it is RCA, but because it is a very good transistor. The second type that we are using is *2N579*, which is used to drive the high-speed memory cores.

Chairman Estrin: This question is from H. Zeidler, SRI for Mr. Cornell: "How do you accommodate intersecting or coincident flight paths (two dimensions and three dimensions)?"

W. A. Cornell: This is handled in one of two ways: If there is enough amiss in the intersect pass, it is carried through; otherwise it requires manual intervention of the operator's paths.

Chairman Estrin: Another question for Mr. Thornton from Mr. R. Frohman, National Cash Register Company: "With the computer package as shown in your paper, is cooling necessary? What is the power required by the computer?"

J. E. Thornton: We only use room temperature air with fans. The total computer power is 1,200 watts.

Chairman Estrin: H. V. Flesch, I.T.T. Laboratories asks Mr. Thornton: "Does the buffer control sense if a bin is full (input) or empty (output)? Do you allow an interrupt routine to be interrupted?"

J. E. Thornton: I have to decide what is meant by a "bin" here. First of all let me explain what happens in initiating a buffer operation for output. Let us say that a certain area of memory is allocated to transferring information out; if this is a 130-bit word of information, each of these will be transferred out one at a time at a rate of the output device. The operation requires that the buffer control of the computer decides when all the information is available. No examination of the data is performed at this time. The answer to the other question is, no. We do not allow interruption on an interrupt. What happens here is, if the interrupt designator is set at this time of interrupt, it is cleared out after the interrupt is performed by a very clever means.

Chairman Estrin: Mr. S. W. Palpus, Palpus Engineering Company, asks Mr. Poorte: "Will the *501* be provided with a large-scale random-access memory equivalent, for example, to the IBM Ramac?"

Glen E. Poorte: Yes. The slides you saw show that the file is capable of handling a million and a half alphanumeric characters each, and each file control unit is tied to a trunk line capable of handling 33 devices. If you wish to extend this, you can tie in more file-control units and decontrol even more of the file devices.

Chairman Estrin: Another question from Dr. Flesch, I.T.T. Laboratories, for Mr. Poorte: "Would you please explain the random access file system with regard to: 1, descriptors on which searches can be made and 2, simultaneous search capabilities, if any? also, what about inquiry handling?"

Glen E. Poorte: I would like to start with the last one. There is a simultaneous search feature in the file control unit. It accepts criteria, and searching and computing go on simultaneously. When the criteria has been reached on the file unit itself, the transfer then takes place into the high-speed memory by the use of interim technique.

Regarding the second part of the question, one search at a time unless there is more than one file control unit in the system. Then there can be more than one searching going on simultaneously.

If I understand the question correctly, the criteria can be any item in a message which is contained on the file, and the criteria is actually limited to 16 characters in length. Does this answer the question?

Chairman Estrin: Mr. Flesch adds: "By handling I meant for manual operation, to step up to the machine and have an answer."

Glen E. Poorte: Oh, this is handled in exactly the same way: the criteria is manually inserted in the computer console through paper tape, or simply inserting it through the buffers, and a search can then be made on the file. If you wish to have it printed on a line printer, it can be done by means of a high-speed memory after the transfer has taken place, or it can be transferred out by way of the monitored printer for copies.

Chairman Estrin: From Mr. R. A. Kudlich, Bell Telephone Laboratories, for Mr. Thornton: "How many, and what types of input devices and output devices would be operating simultaneously to achieve the 50,000 words per second input-output rate?

J. E. Thornton: The 50,000 words per second would be anything from one device, one register, up to dovetailing and operating simultaneously on several of the registers to an upper maximum limit.

Chairman Estrin: We have a number of questions here from Quentin Cornell for Mr. Poorte: "What is the random access store media? What is the random access time? What is the maximum access time? What is the minimum access time?"

Glen E. Poorte: Minimum access time is almost instantaneous. Maximum time would be twice the 192, or 384 milliseconds maximum.

Chairman Estrin: Mr. Cornell also asks: "What is the contemplated lease cost per month for a random access store module?"

Glen E. Poorte: There is a lease policy. However, I am not in a position to quote these prices.

Electronic Differential Analyzers in Perspective

JOHN McLEOD
NONMEMBER AIEE

IN THIS paper's contribution to the discussion of Contrasting Tools and Techniques for Simulation, "Electronic Differential Analyzers in Perspective", the emphasis will be on perspective. There is no other field of technical endeavor in which the picture is so often thrown out of focus and distorted by point of view as that which this paper will attempt to clarify. It is not difficult to realize why this is true. The field of simulation, dependent as it is on both the technique and equipment of machine computation, is comparatively new and growing fast; so fast, in fact, that even those who have been in it for some time are unable to keep up with all developments. And people who have become involved more recently have usually been so busy trying to apply the tools available to the problem at hand that they have scarcely had time to look over the other fellow's shoulder. Hence this discussion.

So that others will understand the author's comments, even if they do not agree, simulation will be defined as the use of electronic analog and/or digital computers to determine the dynamic behavior of physical systems. The computer mechanization may or may not operate on a one-to-one time ratio with the physical system being simulated, and it may or may not include parts of the physical system. Other methods of simulation are recognized but will not be discussed here.

Simulations of the kind under consideration may be mechanized on direct-analog, electronic differential analyzer, digital differential analyzer, general-purpose digital, or special-purpose computers, or combinations of these. And, although the choice of which to use is more often dictated by availability than suitability, the assumption will be made here that suitability is the only factor of interest.

The remarks in this paper will be confined to describing the characteristics, both good and bad, of analog computers (as the author sees them), and rude comparisons with other computers which may sometimes be coerced into something similar to real simulation will be avoided!

The term "analog computer" refers to a group of "operational amplifiers" and related equipment suitable for solving sets of ordinary simultaneous nonlinear differential equations. Operational amplifiers are capable of accepting several voltage inputs, multiplying them by constant integers, and summing and/or integrating the result with respect to time. They are usually associated in an analog computer with a stable reference voltage and power supply, a number of potentiometers for "multiplying" by fractions, multipliers for multiplying by variables, resolvers for changing co-ordinate systems, function generators for introducing empirical relations, switches, relays, etc. for controlling, and meters for monitoring operation, and finally recorders and plotting boards to record the results. All of these components operate on voltages, which are made analogous to the variables of the physical system. In the general-purpose analog computer the components are connected together by means of patch-cords much as telephones were before the advent of automatic exhanges.

For the benefit of those who are not familiar with the setup of analog computers (and experience has shown there are many too many), a simplified example will be presented to emphasize what the author believes to be the most important characteristics of these computers.

Assume that the effect of changing various parameters of a short-range rocket is to be studied in order to arrive at an optimum design. To begin, assume a constant weight and a constant thrust for a definite duration of time and use the basic equation:

$$F=ma \tag{1}$$

Solve for the highest derivative,

$$a=F/m \tag{2}$$

and define the terms:

a = acceleration along the flight path

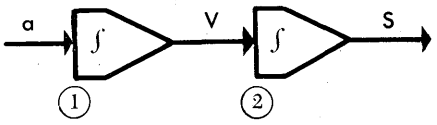


Fig. 1. Double integration of acceleration to obtain displacement

F = summation of longitudinal forces acting on the rocket
m = total mass of the rocket.

For the time being consider that the thrust and the aerodynamic drag are large compared to the weight of the rocket, i.e., $F=F_T+F_D$.

It has been stated that the thrust will be a constant, T, until the cutoff time t_c, and then it will be zero, so that:

$$F_t=T\ (0<t<t_c)$$
$$=0\ (t\geq t_c)$$

It is known that the drag force, $F_D=-1/2\rho V^2SC_D$ where

ρ = atmospheric density
V = the velocity of the rocket along the flight path
S = the equivalent frontal area of the rocket
C_D = the drag coefficient.

Now, if this is a short-range subsonic rocket little error is introduced by letting

$$k=-\frac{1}{2}\rho SC_D$$

so that $F_D=kV^2$ and the complete longitudinal equation reduces to:

$$a=\frac{(T+kV^2)}{m} \tag{3}$$

Now note how straightforward the mechanization of a problem on an analog computer is.

Basic to the procedure is what, at first, may seem an invalid assumption. This is that the voltage analogous to the value of the highest-order derivative, in this case, a, can be made available when it is needed.

Therefore in Fig. 1, assume that somehow a voltage can be generated analogous to the acceleration, a. If so, this voltage can be integrated with respect to time in the integrating amplifier (1) to obtain a voltage analogous to the rocket velocity V. If desired, it is possible to integrate that voltage in amplifier (2), to get a voltage which will be proportional to the distance, S, traveled along the flight path.

Now, consider validating that original assumption. What is needed to generate a voltage analogous to a?

Looking again at equation 3, it is seen that T is needed as a function of time; therefore, one constant voltage proportional to T, and another voltage which increases proportionally with time are required. These voltages are obtained from the computer reference

JOHN McLEOD is with Convair Astronautics, San Diego, Calif.

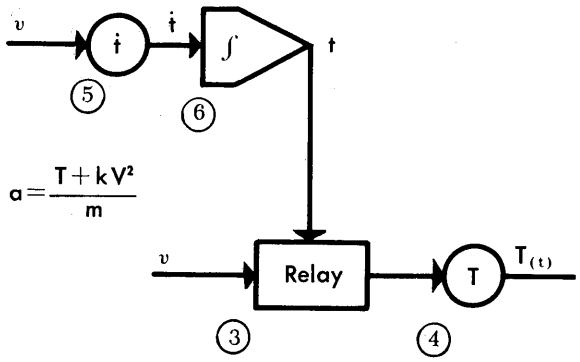


Fig. 2. Generation of thrust as a function of time

voltage v. For T this is supplied through a relay (3) to the scale-factor potentiometer (4) which adjusts it to some convenient value, proportional to T. For t, the reference voltage v is adjusted by means of the potentiometer (5) to some convenient value, $\dot{t}$, and then integrated in amplifier (6) to give a voltage, t, which increases linearly with time. The normally closed relay (3) is set to open when $t=t_c$, which operation cuts off the simulated thrust.

Referring once more to equation 3, it is seen that a voltage proportional to kV^2 is also required. This can be obtained as shown by multiplying V, which is already available from amplifier (1), by itself in the multiplier (7) to produce V^2 which can then be multiplied by k in pot (8) to give a voltage proportional to kV^2. Now, adding this to T in the summing amplifier (9), and multiplying by $1/m$ in pot (10) will give $T+kV^2/m$, which is equal to a. So the assumption that a could be made available was justified, and the computer is connected to solve the required simple nonlinear second-order differential equation.

Heretofore, the author has referred to velocity and direction along the flight path. Usually, however, one is interested in the shape of the trajectory, and this re-

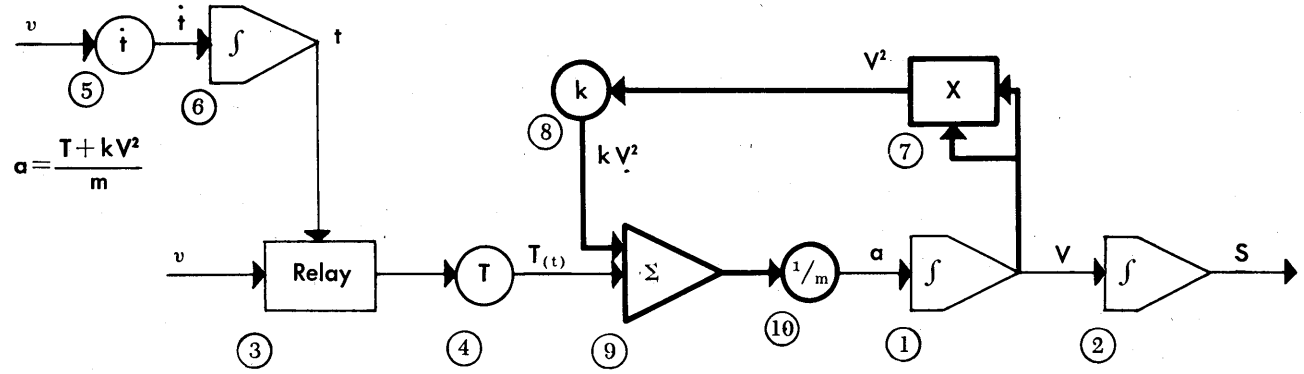


Fig. 3. Generation of the drag term (heavy lines) and combining it with the previously generated terms to solve the basic equation

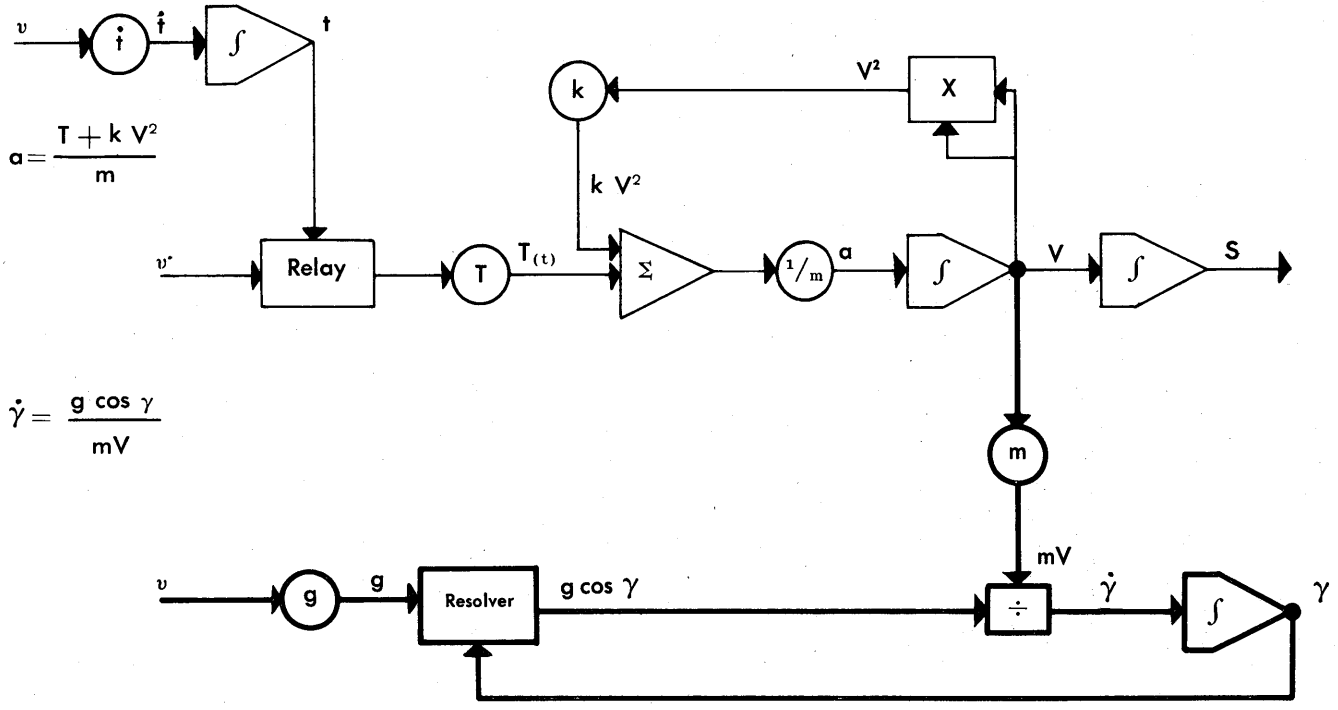


Fig. 4. Generating the flight-path angle (heavy line)

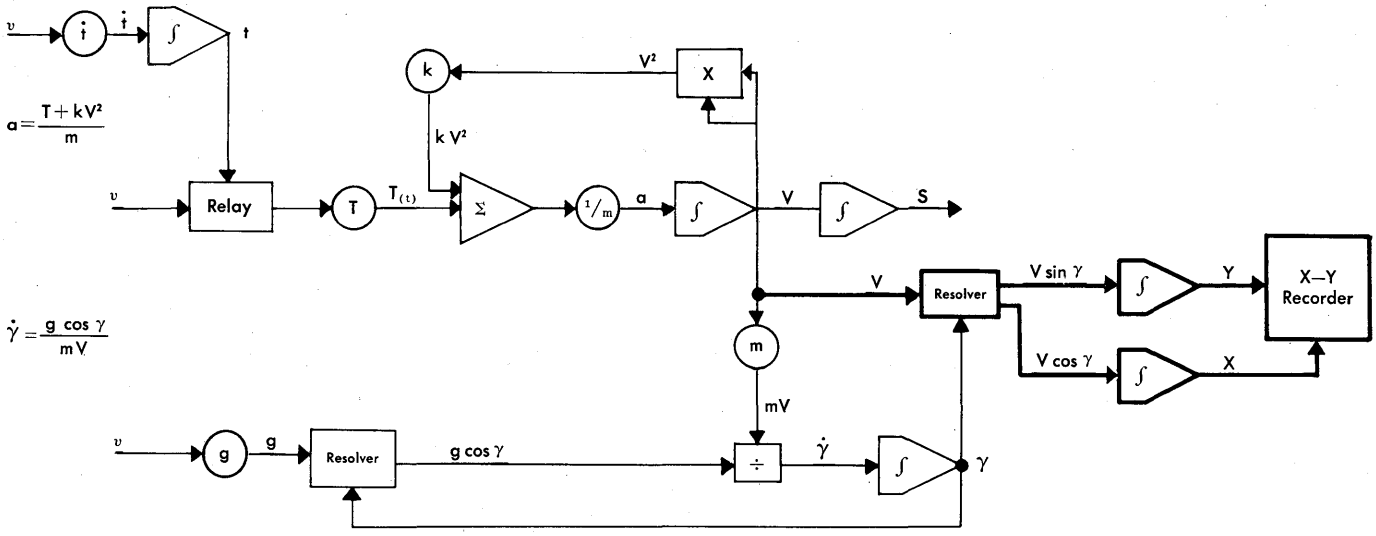


Fig. 5. Additions required to the trajectory

quires that the direction of the flight path, or flight-path angle, γ, be known as a function of time. This can be found by writing the basic centrifugal force equation, $F_c = \omega^2 rm$ in the form

$$\dot{\gamma} = \frac{Fn}{mV}$$

where $\dot{\gamma}$ is the rate of change of flight-path angle (equal to ω), F_n is the summation of the forces acting normal to the flight-path angle, which must be equal to F_c, m is the mass, and V is the instantaneous linear velocity of the rocket.

Thus, if one considers a "no-lift" trajectory (one in which there are no aerodynamic forces acting normal to the flight path) the component of gravity normal to the flight path, $g \cos \gamma$, will be the only force turning the rocket. Now the relationship $\dot{\gamma} = g \cos \gamma / mV$ can be written and mechanized to find γ by adding the computing components as shown in Fig. 4.

If an x-y plot of the trajectory is desired, it can be obtained by adding components as shown in Fig. 5 to mechanize the relationships $V_x = V \cos \gamma$ and $V_y = V \sin \gamma$, and integrating them to obtain and record x and y.

If it is desired to examine the rocket performance in more detail the simulation can be made more detailed. As shown in Fig. 6, if the rocket weight is not small compared to the thrust and drag, a term $g \sin \gamma$ must be added to the acceleration. This term is already available from the resolver which produced $g \cos \gamma$. If the rocket is high-speed, so that C_D changes nonlinearly with velocity, and long-range so that the altitude, and therefore ρ changes considerably the drag force, $F_D = 1/2 \rho V^2 S C_D$, must be generated by adding the function generators and multipliers as shown, and resetting pot (5) to take care of the only remaining constants 1/2 and S. Computing elements can as easily be added to take care of other effects of interest, such as the change in mass as fuel is consumed. The value of any of the variables in the system, which are represented by voltages generated in the computer, can be plotted out against time or against other variables, as desired.

Now in this problem only the two-dimensional translational equations in the pitch-plane have been considered. When the three dimensional systems are considered, and rotational and cross-coupling effects are included, the mechanization can, and does, get complicated. Nevertheless, this simple example illustrates one of the most important characteristics of the analog computer: An engineer can build with computer components, just as with system components, a system as simple or as complex as may be required for his purpose. Furthermore, as he builds up the simulation he can maintain a one-to-one correspondence between computer components, or groups of components, and system components, groups of components, or functions, as he sees fit. Thus the analog computer can be made a recognizable model of the physical system under study, and a powerful and flexible experimental tool is created.

Some advantages are obvious:

1. Changes in design parameters can be simulated by twisting a dial, or at most by changing a function in a function generator. For instance, in the previous example, the effect of increasing thrust can be observed by increasing the setting of pot (4). Or the effect of aerodynamic design can be studied by changing the C_D function generator.

2. Changes in basic design can be simulated by changing patch-cords to change the basic model. Perhaps one would like to know what would happen if some fins were put on the rocket in the example. This would require additional computer components to generate voltage proportional to the lift and drag forces created by the fins. These voltages would then be plugged into the a and $\dot{\gamma}$ summing junctions.

3. The engineer is forced to become intimately familiar with the operation of the physical system. For this reason it has been said that this alone would justify the use of an analog computer, even if it never actually "solved" problem one!

Other advantages of the analog computer are the speed with which it can operate and the "language" which it speaks.

The speed of the analog computer results from its operating in parallel with respect to time, i.e., it operates on all of the voltages at the same time, adding, integrating, multiplying, etc., simultaneously. The real importance of this speed is not merely a saving in the operating time necessary to solve a problem, though this is sometimes important, but stems from the fact that it allows the computer to operate in "real" time, i.e., in a one-to-one time relationship with the physical system. This not only makes the model more realistic, but is absolutely necessary if components of the actual system are to be used in the simulation, a practice which is sometimes desirable, or, if the response of the system components is in question, necessary.

Advantages of the analog language

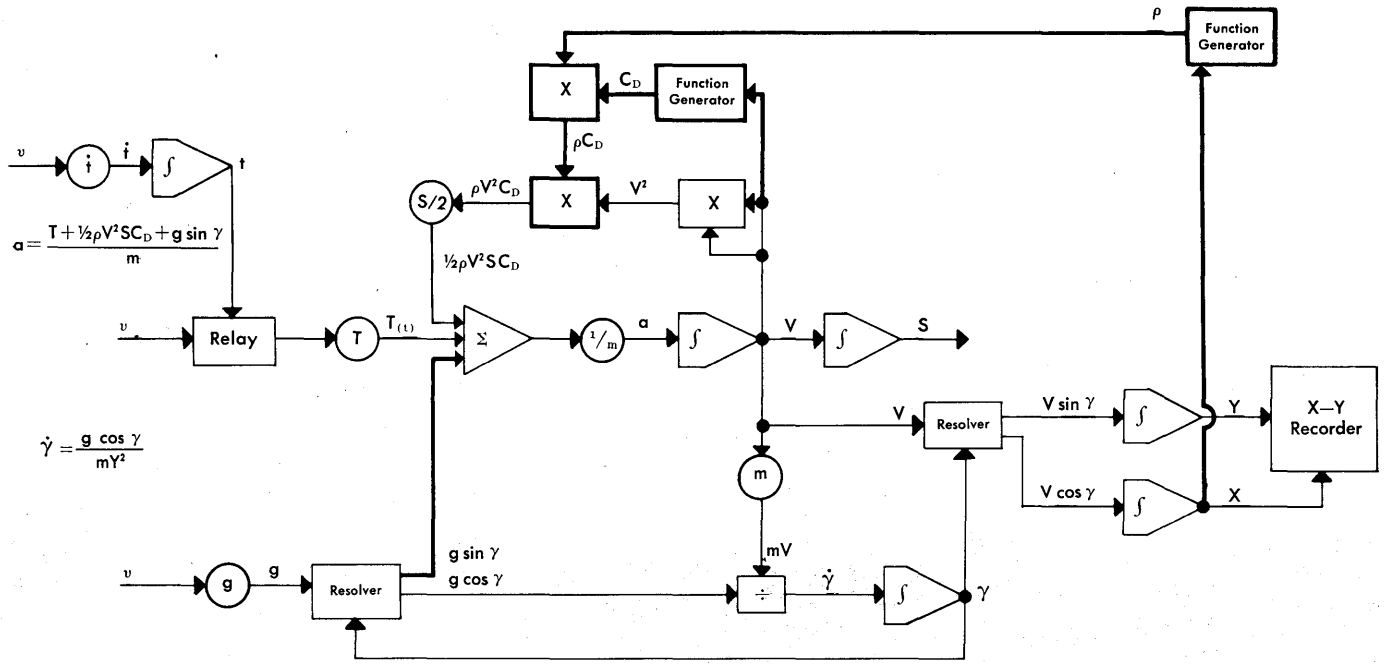


Fig. 6. Additions required to account for the effect of gravity and a changing air density and drag coefficient

usually accrue to the electronic differential analyzer at both the input and output. Most of the inputs represent quantities in nature, which in general are analog, that is, continuous, rather than discrete. Therefore no "translation" is necessary. At the other end, the output is usually for human consumption, and as most engineers are human and prefer graphs to numbers for the analysis of dynamic behavior they understand the computer. Rapport is established!

So much for the advantages of electronic differential analyzers; have they no shortcomings? They have!

The one you will probably hear talked about most is their lack of accuracy. The author belives this is overemphasized. True, although most individual components will produce errors of only about one part in a thousand these may build up in a complex problem to 2 to 5% or more. However, analog computers are usually used to simulate hardware the performance of which is not predictable, or measurable, with accuracy that great and often not as great. Furthermore, computer errors build up in the simulation in the same way that hardware and measurement errors do in the physical system, so for simulation, the analog computer is usually as accurate as it has to be.

A more serious difficulty is that of answering the question "How does one know he is right?" Qualified answers may be obtained by incorporating some of the following procedures.

1. Component checks.
2. Static checks.
3. Numerical checks.
4. Comparison with the physical system.
5. Engineering judgment.
6. Clairvoyance.

Usually some combination of the foregoing is used to build up the necessary degree of confidence. However, checking is still a problem, and is recognized as such by the computer manufacturers who have made, and are making, progress in alleviating it.

One inherent characteristic which may or may not be a disadvantage, depending on the problem, is the inability of the electronic differential analyzer to integrate with respect to any variable other than time. In simulating most physical systems this is no handicap because it is only necessary to integrate with respect to time. In other cases, if it is only necessary to integrate with respect to some other one variable, the problems can be set up in such a way as to represent that variable in the system by time in the simulation.

Another inherent characteristic of the analog computer gives rise to two difficulties. This is the fact that an analog computer setup grows in complexity and number of components with the complexity of the physical system being simulated. Therefore cost goes up and reliability goes down. There seems to be no way around this!

As to the cost, a rule of thumb is that on the average three-operational amplifiers will be used for every order of the set of differential equations being mechanized. This may vary from a little more than one to ten or more (for problems involving a large amount of algebra). Unless the type of problems to be solved is defined, three per degree is about as good a guess as can be made.

The price of an analog computer will vary from about $700 per amplifier to perhaps ten times that amount. The lower figure includes a minimum of auxiliary equipment as contrasted to the more elaborate computers with a complete complement of nonlinear, input-output, and special checkout equipment. The cost of the average installation will probably run between $2,000 and $3,000 per amplifier, or from $5,000 to $10,000 per order of the set of equations. Circumstances may alter these figures by an order of magnitude in either direction, but if one must guess without additional information this is about as accurate as possible.

The foregoing would seem to indicate that if one is willing and able to spend the money he can simulate systems of any complexity. Not so! As stated, the reliability of an electronic differential analyzer, as with anything else, goes down with the complexity of the mechanization. Thus, a point is reached when the equipment simply cannot be made to give useful answers. The author cannot tell where this point is, because it depends on the problem, the equipment, the opera-

tor, and the maintenance man. And, as two humans are involved in the operation of this equipment, the author will not even hazard a guess!

Some people are cowed by the mere idea of an electronic computer, while others can successfully manage complexes of several hundred amplifiers and auxiliary equipment. Moral: pick operators and maintenance men with the same care with which equipment is chosen, and pay them accordingly!

The Case for Combined Analog-Digital Simulation

WALTER W. VARNER
NONMEMBER AIEE

THE IDEA of combining the good features of both analog and digital devices is certainly not a new one. Numerous examples can be found in history dating back to the ancients who kept time by a slowly burning cord with equally spaced knots tied in it. Each time a knot burned, this fact was tallied (digitally) and the subintervals were estimated by interpolation (analog). One more recent example is the odometer on all automobiles where the whole miles are given digitally while the tenths are estimated on a nonquantized analog-dial wheel. Another example appears in many digital clocks where the hours and minutes are given digitally on dials, but where the seconds are estimated on a continuous analog-scale wheel. Often in determining areas enclosed by curves, a combination of digital and analog methods are used. For example, the largest possible portion of the area may be blocked out as a rectangle whose dimensions are digitally determinable and its area computed by forming the digital product of its dimensions. Then the remaining area or areas are estimated by using a planimeter, an analog device.

There is an almost unlimited number of such examples that could be given to show that practical men have realized the importance of combined digital and analog computation, using the digital methods for greater accuracy and analog methods for simplicity and convenience. Today with elaborate analog and digital computers in widespread use it was again inevitable that practical men would seek to combine the salient features of each to produce a superior hybrid.

In previously presented papers, the following reasons were among those given for using only analog computing equipment for simulation.

1. Real time simulation is easy on an analog computer.
2. Actual analog hardware can be used in the loop.
3. An analog computer permits the engineer to get a "feel" for his problem by observing the results at the same time parameters are varied with potentiometers.
4. Analog equipment is comparatively inexpensive.
5. Analog computers require relatively short programming, debugging, and checkout time.

Similarly in the digital presentations, some of the reasons given for preferring only digital equipment for simulations were:

1. Digital computers are more accurate.
2. Digitally computed results are reproducible.

The strong features given for each type of computer are, in turn, the weak features of the opposite type. Thus, with each of the computer types having complementary strong points, it was only a matter of time before the interconnection of them was undertaken. It is natural to ask how it is possible to combine the advantages of analog and digital computers without combining their limitations; i.e., why is the resultant simulation not as slow as the digital computer and as inaccurate as the analog? The answer is, of course, the way in which they are combined. The analog equipment is used to simulate the high-frequency effects and noise while the digital equipment is used to simulate such things as relatively slow-speed navigational computers.

In the early part of 1954, under the direction of Dr. Walter H. Schwidetzky, Chief of Computers and Simulation at Convair-Astronautics, tentative specifications were written for an high-speed combined analog-digital and digital-analog converter. Among the more important considerations involved were the following:

1. The number of bits of digital information that should be converted.
2. The number of analog-digital channels and the number of digital-analog channels.
3. Conversion time for each channel.

After an evaluation was made, it was decided that a dynamic range of 100,000-to-1 would be more than adequate. To accomplish this, it was necessary to have 17 bits plus a sign bit where the most significant bit represented 50 volts and the least significant bit represented slightly less than 1 millivolt (mv) (actually 100×2^{-17} volts).

For all work which could be contemplated, it was felt that 15 channels of analog to digital conversion and 10 channels of digital to analog conversion would suffice. Conversion time for each channel was limited by the state-of-the-art and the specified 100 microseconds per analog-digital (μsec per A-D) conversion and 25 μsec per D-A conversion have still been only approximately met.

It is interesting to note that originally the specified maximum sampling rate for any one channel of analog to digital conversion was from 1 to 20 per second and for digital to analog it was 1 to 100 per second; however, the equipment has successfully operated at analog-to-digital sampling rates of over 4,000 per second and at digital-to-analog sampling rates of over 5,000 per second. These speeds are limited by the *704* execution time and could be approximately doubled if a faster computer were available.

After all of the specifications were determined, the development and construction contract was let to Epsco, Inc., Boston in early 1955. While the 25-channel converter was being designed and built, combined simulations were performed using an Epsco Datrac single-channel analog-to-digital converter and single channel digital-to-analog converter to connect an *1103* computer with the analog computer on a trial basis. These tests were carried on successfully through the first half of 1957 proving the soundness of the technique. Tests were run on a simplified guided missile in-flight simulation and results were excellent.

Incidently, to investigate other uses

WALTER W. VARNER is with Convair-Astronautics, San Diego, Calif.

for a high-speed converter, the Datrac was also used to digitize analog voltages from telemeter tape at rates of 4,000 words per second.

The 25-channel Epsco Analog-to-Digital and Digital-to-Analog Converter (Addaverter) was delivered untested and not fully debugged in April 1957, and because of modifications suggested by subsequent tests, it was September 1957 before active testing got under way. Until December 1957 the Addaverter was used actively in several different applications such as simulations, checking analog multiplier accuracy, and the digitizing of analog-telemetered data from magnetic tape. During the time from December 1957 to May 1958, the Addaverter was moved to the new astronautics plant and checked out with an International Business Machines *704* in preparation for complete missile simulations.

Combined analog and digital computer simulation is admittedly an expensive method since it requires the use of three costly units of equipment, and at the same time presents the problem of effective utilization of each of the units since considerable skill and maintenance is necessary to make all three operate satisfactorily at the same time. System debugging and checkout time is expected to be held to a minimum by the use of a *704* simulator and by careful checkouts of each of the units before they are interconnected. With the limited experience encountered so far, it seems that efficiency is increasing so that with advancements in the state of the art and increased operating proficiency, combined computer simulations will be about 90% as efficient to debug and check out as a purely analog simulation, (which lacks needed accuracy, and about 90% as accurate as a purely digital simulation which is never possible if analog "hardware" is included in the loop. Once debugged, it has been determined that large combined simulations can be set up for re-runs for production faster than all-analog runs of comparable over-all magnitude.

In a combined analog-digital simulation a compromise must always be made between what percentage of a given simulation will be put on the analog computer and what will be put on the digital computer. At the present time an average simulation uses from 500 to 2,000 digital machine instructions and at present digital computer speeds, relatively low sampling rates of one to one hundred per second are necessary to permit this number of digital computations to be made per sampling time increment.

In some simulations where an analog-digital system is being simulated by the Addaverter system the sampling rate is made the same to increase the validity of the simulation. In one such simulation the sampling frequency was 2 per sec. The computations desired were limited and efficient techniques were used to stay within the 500-millisecond computing (ms) time available. The actual computation used required about 400 ms. Thus 100 ms of time was available for furture modifications and expansions. In other simulations where the sampling frequency is not fixed by the simulation, it is determined by the amount of time required for the digital computation and usual care must be used in choosing sampling rates so that the conversions will not be made so often that intolerable rounding errors will occur and yet often enough to prevent excessive truncation. As computer speeds increase in coming years, a greater percentage of each simulation will be made on the digital machines for accuracy reasons. It is conceivable that ultimately only the actual analog hardware in the loop will be handled by analog equipment.

To change viewpoint for a moment, alternate methods of performing accurate, real time simulations should be mentioned. With the development of transistorized and high-speed electronic digital-differential analyzers and recently announced compatible high-speed converters, there is indicated the possibility that many small-magnitude combined simulations where extreme accuracy is not necessary may be done more cheaply by using an electronic digital differential analyzer, a small analog computer for making parameter variations and including hardware, and a converter to link the two.

As a final indication of the necessity and importance of combined analog and digital simulation it should be noted that aircraft and missiles have now become so complex that they, themselves, are in many cases controlled by self-contained combination analog-digital systems. To design and develop such systems and to investigate their behavior with superimposed noise, it is essential that combined analog-digital computing equipment at least one order of magnitude better be available to simulate them.

Digital Computer Solution of Differential Equations in Real Time

H. J. GRAY, JR.
NONMEMBER AIEE

DIGITAL computers have been used for some time now to obtain the solutions of differential equations. The first electronic digital computer, the Electronic Numerical Integrator and Computer, (ENIAC)[1] was designed and built at The Moore School of Electrical Engineering under contract with the Aberdeen Proving Ground for the specific purpose of integrating the differential equations of motion of a projectile. This computer was very successful at this task. Digital computers have been used to prepare tables of mathematical functions which are solutions of differential equations and one occasionally reads that a digital computer has been used to check, but not in real time, the results of an analog simulation of some system described by differential equations. When the subject of the digital computer solution of differential equations in real time is considered, questions that have to be answered arise in connection with speed, accuracy, cost, ease of maintenance, and flexibility. In this paper, it will be attempted to discuss these points within the framework of a specific application, the Operational Flight Trainer.

Operational Flight Trainers (OFT's)

H. J. GRAY, JR. is with the Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia, Penn.

The work reported here is the work of many people whom the author wishes to thank, especially Dr. Morris Rubinoff, Mr. Herbert Gurk, Mr. Cornelius Eldert, and Dr. F. J. Murray.

This work was performed in connection with contracts N6onr24915, Nonr551(02), and N61339-272, with the United States Naval Training Devices Center, Office of Naval Research, Port Washington, N. Y.

have been used for many years in the training of airplane pilots, a notable example being the Link Trainer. Present-day OFT's are complex devices employing computers which solve the equations of motion of the aircraft for which the pilot is being trained while the student pilot sits in a cockpit which furnishes the inputs to the computer. The solutions of the equations of motion are modified by further computation to provide the instrument readings and stick forces which are returned to the student pilot. In addition the instructor can send input data to the computer and can receive output data. A block diagram of an OFT[2,3,4] computer is shown in Fig. 1 which shows the character of the inputs and outputs.

Nature of the Problem

Before discussing the factors that are related to the choice of computer, it will be attempted to give the reader some further indication of the nature and magnitude of the problem. If the aircraft can be considered as a rigid body one has the equations:

(dots indicate differentiation with respect to time):

$$\dot{u} = gX/W - gl_3 - wq + vr$$

$$\dot{v} = gY/W + gm_3 - ur + wp$$

$$\dot{w} = gZ/W + gn_3 - vp + uq$$

$$I_x\dot{p} = L + (I_y - I_z)qr$$

$$I_y\dot{q} = M + (I_z - I_x)rp$$

$$I_z\dot{r} = N + (I_x - I_y)qp$$

$$\dot{l}_3 = rm_3 - qn_3$$

$$\dot{m}_3 = pn_3 - rl_3$$

$$\dot{n}_3 = ql_3 - pm_3 \qquad (1)$$

where u, v, w are the linear velocities of the aircraft referred to axes imbedded in it; p, q, r are, similarly, angular velocities; l_3, m_3, n_3 are three of the nine direction cosines relating the airplane axes to ground axes; g is the acceleration of gravity; I_x, I_y, I_z are the principal moments of inertia; W is the weight; X, Y, and Z are the aerodynamic forces; and L, M, and N are the corresponding torques. In addition to these equations, there are equations for X, Y, Z, L, M, and N under a variety of conditions, equations assuring the normality and orthogonality of the direction cosines,[5] equations relating instrument readings with the dynamic variables, etc. The total number of equations for a typical subsonic jet aircraft is such that when typed to a reasonable density, approximately 26-

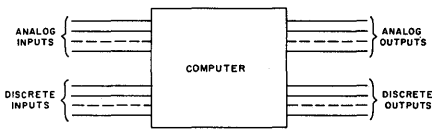


Fig. 1. Block diagram of an OFT computer

8½ by 11 inch sheets of paper are required. The equations have been programmed for Universal Digital Operational Flight Trainer, (UDOFT) a digital-computer-activated flight trainer designed by the Moore School and now being built by Sylvania Electric Company. This computer will take approximately 11 milliseconds to perform the computations necessary to advance one interval in the numerical integration of the differential equations including all subsidiary computations. For a typical supersonic jet aircraft, the computer takes approximately twice this time, an indication of the amount of computation involved in the case of supersonic aircraft. It is apparent, then, that the problem under discussion is of appreciable magnitude.

The Computer[4]

All of the trainers in existence use analog techniques in order to perform the functions of the block labeled "computer" in Fig. 1. (At the time of the writing of this paper UDOFT had not yet been put into operation.) Certain reasons have been put forth giving the desirability of using digital techniques. (It should be mentioned that the number of analog inputs and outputs to the computer in Fig. 1 was found to be equal to the number of discrete or digital inputs and outputs required in the case of the simulation of the afore-mentioned aircraft. Therefore, the question of compatibility of computer type with inputs and outputs does not arise.) Foremost of these reasons is that a digitally computed flight solution is as accurate as the numerical method used, and analog computer problems such as amplifier drift, errors due to component drift, etc., do not arise. Other advantages of digital techniques will be stated later.

Digital Techniques

Two types of digital machines were considered at first, the digital differential analyzer (DDA), and the digital computer of the general-purpose stored-program kind with, possibly, some modifications to make it better suited to the problem. At that time the only existing DDA was the MADDIDA, which was too slow. In addition, a hypothetical DDA using the best of digital computer techniques also proved to be too slow. The main reasons were that the method of integration requires a small integration interval to get good results, addition using integrators is essentially equivalent to counting, and much of the computation is essentially serial in nature. On the other hand, a hypothetical digital computer using the same digital computer techniques, while still too slow, showed more promise of success. This led to the development of the UDOFT. The characteristics of UDOFT are as follows:

Address structure: single address
Number representation: signed 20-bit magnitude
Information rate: 1.2 millicycles per second 5 phases
Order memory: 4,095 words, 20 bit per word (6-bit order type, 12-bit number memory address, 1 relative address bit, 1 parity bit)
Number memory: 4,094 words, 22 bits per word (20-bit magnitude, 1-bit algebraic sign, 1 parity bit)
Input: punched card reader used to load problem
Output: Teledeltos printer available for test purposes
Discrete inputs: 64 toggle switches
Analog inputs: 24 Gray code wheels
Discrete outputs: 24 on-off signals
Analog outputs: 64 multiplexed signals
Operation times: addition, subtraction, and similar operations, 5 microseconds; multiplication, 10 microseconds; divide, 105 microseconds.

In addition to high accuracy as a reason for the use of digital techniques, there are several additional reasons for the use of a stored-program digital computer. These are derived from the characteristics of such computers and are as follows:

1. A digital computer is flexible and can be changed by programming from the simulation of one plane to another. This flexibility also makes it possible to change flight conditions during a test run or to alter aerodynamic coefficients gradually in order to test the effects of such changes on the flight of the plane and the response of the pilot.

2. One digital computer should be able to solve the equations for several cockpits simultaneously, allowing a group of trainees to receive simultaneous instruction, either in independent flights, in flight formations, or in simulated combat.

Comparison of Digital and Analog OFT's[6]

A summary comparison of digital and analog OFT's will now be made on the basis of accuracy, speed, flexibility, ease of maintenance, and cost.

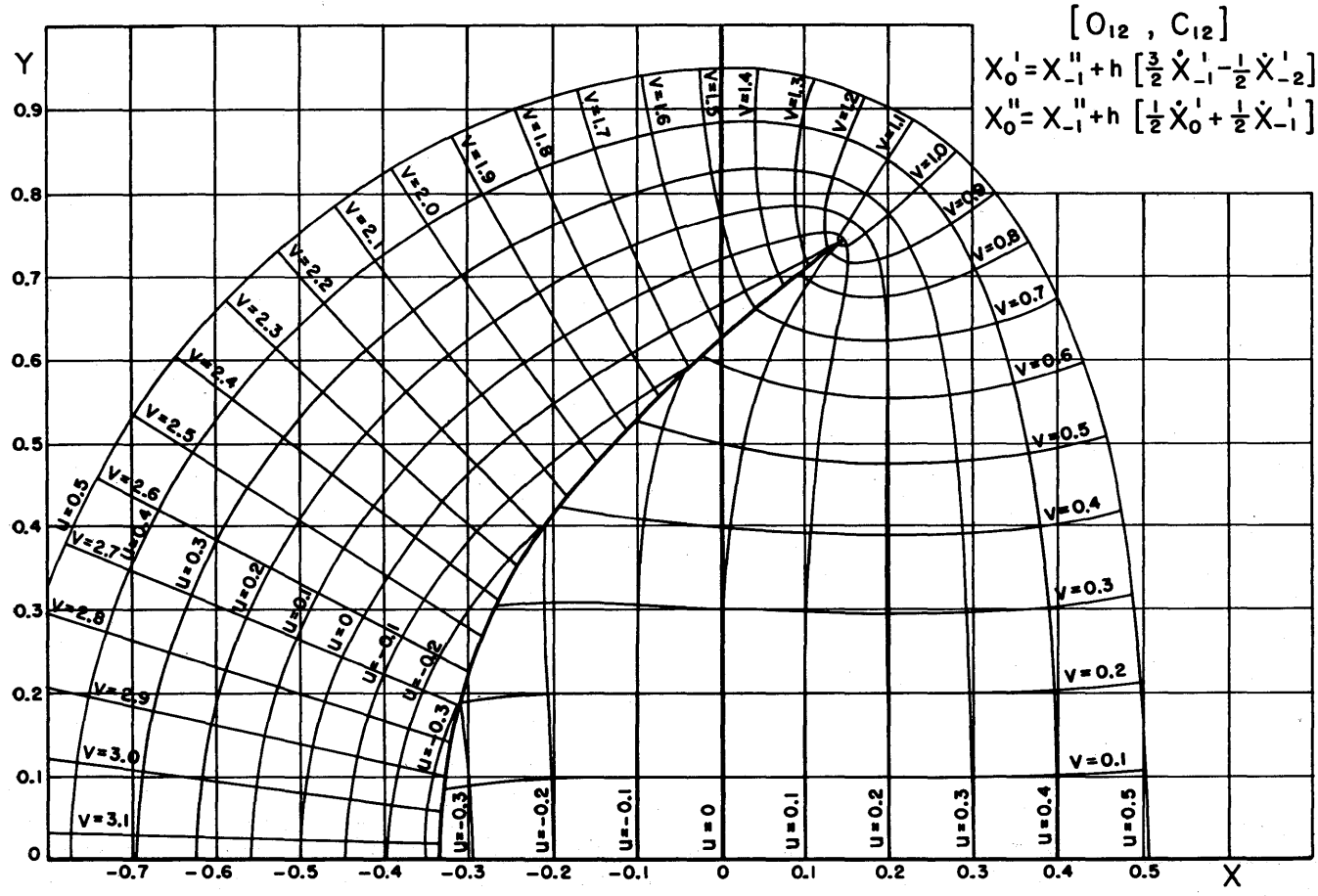


Fig. 2. Stability chart for classical $[O_{12}, C_{12}]$

Accuracy and Speed

The analog OFT's, operating as they do in real time, have been found accurate enough to satisfy the desires of experienced pilots in most cases. Accuracy of results for an analog machine depends on the degree to which the various components of the computer approximate the functions which they are simulating.

The digital OFT is as accurate as the numerical method used. Actual calculations of simulated maneuvers have been carried out using the methods of the digital OFT. These have given quite accurate results. A mathematical theory has been developed enabling the prediction of the maximum interval permissible in the numerical solution of the differential equations. This interval is several times larger than the time it takes UDOFT to perform the calculations. The mathematical theory will be discussed shortly.

Flexibility

Proposed analog computers would be able to simulate any one of a number of airplanes, although existing types would essentially have to be rebuilt to do this. The proposed computers would use plugboards for different planes in order to change numerical values (i.e. resistances) and mathematical relations (connections). A number of different plugboards would be needed for each plane.

For a digital OFT, the airplane to be simulated could be changed by reprogramming the computer. As with the plugboards, the programming would have to be prepared in advance.

The cockpit connections in present analog OFT's are often tied directly to the computer, in some cases being mechanical connections. Such connections might cause some difficulty when the planes being simulated are changed. For a digital OFT, all connections are made via multiwire cable connector. One would anticipate, however, that this could be done in the proposed analog OFT's.

A digital computer, if it is fast enough, can solve the equations of more than one aircraft. The arithmetic unit of UDOFT is fast enough for at least three aircraft and there will be even faster digital computers in the future. It is true, however, that the UDOFT design allows only enough memory space for one aircraft.

To simulate more than one aircraft on an analog computer requires the addition of sufficient analog components to make more than one analog computer. Such a simulation seems to be quite impossible, at present, on a single analog computer. Thus simulation of two aircraft would require two analog computers.

Ease of Maintenance

The components of both the analog and digital computers gradually drift or deteriorate. In an analog computer this gradual change in components shows up as a gradual change in results. In a digital computer, however, such changes in components have no effect on the accuracy of results until some component has deteriorated substantially. Then the error is discrete and immediately noticeable. Thus, it is generally easier to detect the malfunctioning of a digital machine than that of an analog machine.

Cost

OFT's, both analog and digital, are, at the present time, quite expensive pieces of equipment. Modular construction of the digital computer and mass production of modules may make the digital OFT competitive in cost with the

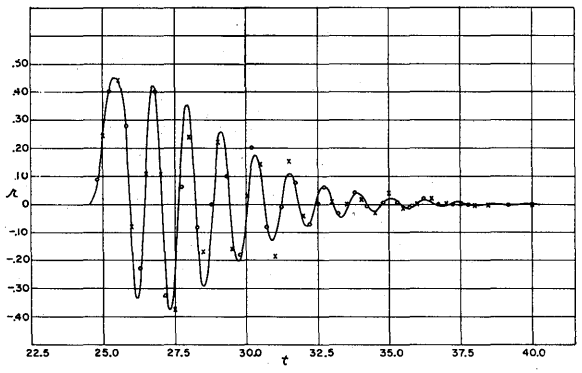


Fig. 3. Immelman turn-r

——Classical rC_{41}
● $[O_{30}, C_{31}]$ mod Gurk
× O_{33} mod Gurk

analog OFT. Existing data, while meager, show the digital OFT to be somewhat more expensive than the existing analog OFT's. One might expect however, that the proposed analog OFT's, being more complex, would be more expensive than the existing analog OFT'S.

Numerical Methods[7]

Success of a digital OFT and some of its superiority are derived from the quality of the numerical methods used and the ability of the mathematical theory to predict the accuracy of the simulated behavior of the aircraft. It is the numerical method together with a fast computer that makes the digital computer solution of differential equations in real time possible. One would expect to see in this "space age" more and more need for the real-time solution of differential equations and for these reasons a brief exposition of the mathematical theory follows.

Quadrature Methods

In the numerical solution of the differential equation

$$\dot{x}=f(x,t) \tag{2}$$

one common way is to approximate the value of x_n by a formula of the type O_{NM}, called an "open" formula

$$x_n=\sum_{j=1}^{N} a_{j0}x_{n-j}+h\sum_{j=1}^{M} b_{j0}\dot{x}_{n-j} \tag{3}$$

from a sequence of values of x:

$\ldots x_{n-3}, x_{n-2}, x_{n-1}$ where x_j and $\dot{x}_j$ are respectively the values of x and its derivative, computed from equation 1, at $t = jh$ (j an integer), h is the quadrature interval (a constant), and a_{j0}, b_{j0} are the known coefficients of the quadrature formula. This yields a new sequence $\ldots x_{n-2}, x_{n-1}, x_n$ which, together with equations 2 and 3 enable one to obtain x_{n+1}, x_{n+2}, etc. In this way the solution of the differential equation is approximated by the values of x at times separated from each other by multiples of h.

The coefficients in the formula given by equation 3 can be obtained by the so-called polynomial method, which for a given formula is equivalent to choosing the coefficients so that the positive integer R is a maximum, where the equations $\dot{x} = O$, $\dot{x} = 1$, $\dot{x} = t$, $\dot{x} = t^2, \ldots \dot{x} = t^R$ are solved exactly by the quadrature formula. This procedure involves the solution of a set of simultaneous algebraic equations. The formula, O_{22}, obtained in this way is,

$$x_n=-4x_{n-1}+5x_{n-2}+h[4\dot{x}_{n-1}+2\dot{x}_{n-2}]$$

When the coefficients are obtained in this way, the formula is called "classical".

If the formula makes use of $\dot{x}_n$,

$$x_n=\sum_{j=1}^{R} a_{jc}x_{n-j}+h\sum_{j=0}^{Q} b_{jc}\dot{x}_{n-j} \tag{4}$$

it is called a "closed" formula, C_{RQ}. Initially, a guess at x_n must be obtained, usually by an open formula. Equation 4 is used repeatedly to generate iteratively over one interval successively better values of x_n before proceeding to the next interval. This is called "method rC_{RQ}". An example of a classical closed formula is C_{23}, Milne's formula.

$$x_n=x_{n-2}+h\left[\frac{1}{3}\dot{x}_n+\frac{4}{3}\dot{x}_{n-1}+\frac{1}{3}\dot{x}_{n-2}\right]$$

A third method uses an open formula, O_{NM}, to estimate x_n and follows it by a single application of a closed formula, C_{RQ}. Such a method is called a "mixed method" and is denoted by $[O_{NM}, C_{RQ}]$. A classical mixed method uses classical O_{NM} and C_{RQ} formulas.

Application of any of the these methods to equation 1 requires the evaluation of the right-hand sides of each of the equations once in an interval for open and mixed formulas, compared to many such evaluations in an interval for any rC_{RQ} method. Hence, the rC_{RQ} methods are not suited for real-time computations.

Stability Charts[8]

In the use of any of the open, closed, or mixed formulas to solve differential equations in real time, it is necessary to have some way of comparing the asymptotic behavior of the computed and true solutions. (Asymptotic behavior is of importance because of the long duration of the computation.) This is accomplished by means of "stability charts."

If the differential equation

$$\dot{x}=\lambda x \ (\lambda \text{ complex}) \tag{5}$$

is solved analytically, its solution is found to be

$$x=x_0e^{\lambda t}$$

The real part of λ determines the rate at which the solution grows or decays and its imaginary part is 2π times the frequency of the oscillations in the true solution. Hence λ is a natural frequency of equation 5.

When equation 5 is numerically solved using any of the quadrature formulas mentioned, a set of points result which, it has been proven, can be fitted exactly by

$$\sum_j c_j e^{\mu_j t}$$

provided the correct number of terms and the correct values of the μ_j are chosen. The μ_j are the natural frequencies in the computed solutions. The asymptotic behavior of the computed solution is determined by the μ_j having the largest real part. If this μ_j is denoted by μ_1, then the computed solution is very nearly given by

$$x_n=c_1e^{\mu_1 t};\ t=nh$$

It also has been shown that if μ_1 is close to λ, then c_1 is close in value to x_0. Therefore, if μ_1 is close in value to λ, the computed solution of equation 5 will be close to the true solution of equation 5. A stability chart is a picture showing how μ_1, a complex number, is related to λ, also a complex number.

A stability chart for classical $[O_{12}, C_{12}]$ is shown in Fig. 2. It has been defined that

$$z=x+iy=h\lambda$$

$$w=u+iv=h\mu_1$$

The use of the chart is best shown by an example. If the equation $\dot{x}=(-2+i3)x$ is solved by this formula and a step of $h = 0.1$, one has $z = -0.2 + i0.3$. This yields a value $w = -0.19 + i0.29$ read from the chart or $\mu_1 = -1.9 + i2.9$. Whether or not this is satisfactory depends on the nature of the problem. However, if $h=0.2$, $z=-0.4+i0.6$, $w=-0.26+i2.19$, and $\mu_1=1.3+i11$ it is clearly unsatisfactory.

The validity of the application of stability charts has been extended rig-

orously to linear systems having several natural frequencies and to linear systems containing forcing functions. As yet no rigorous proof exists extending the validity of the stability charts to varying-parameter linear systems and to nonlinear systems. However, experimental computations have shown that in every case tried, the stability charts have successfully predicted the behavior of the computations. This is not too surprising in the case of not-too-severe 2-dimensional aircraft maneuvers because the equations of motion are not greatly nonlinear. However in the case of a particularly violent maneuver, the Immelman turn, results were surprisingly good as predicted. In this maneuver, the aircraft makes half an inside loop followed by a roll through 180 degrees, to that the aircraft finishes the maneuver right-side up. The violence of this maneuver is indicated by the fact that there was a maximum acceleration of 7 g during the inside loop.

Computations were first made using a small interval with classical rC_{41} to provide a reference standard. Computations were then repeated using two nonclassical methods, O_{33} mod Gurk, and $[O_{30},C_{31}]$ mod Gurk, using an interval of 50 milliseconds. The yawing velocity, r, is plotted in Fig. 3. The greatest error appeared in this variable, yet the results are quite acceptable for simulator use. Note that the results asymptotically approach the reference standard and differ only during the transient oscillations.

Expansion in Series[9]

The frequency μ_1 is clearly a function of λ. Under certain conditions it is possible to obtain a power series expansion of μ_1 in terms of λ. In the case of classical rC_{NM} formulas, a particularly interesting result is obtained as follows:

$$w=\mu h=z+\frac{1}{N+M}\left[\frac{N!(M-1)!}{(N+M-1)!}\right]^2 z^{L+M}+\ldots;\ N\geqslant M-1 \quad (6)$$

where $z=\lambda h$.

Tables of the coefficient of z^{N+M} have also been obtained for $N<M$-1 and are fairly straightforward to obtain for the open formulas also. Equation 6 is useful in estimating μ_1 when more accuracy is required than is afforded by the stability chart.

Synthesis of Nonclassical Methods[10]

The complex frequencies, μ_1, and λ, are related by a function, $z=f(w)$, for open, closed, and mixed methods. The characteristics of this functional interdependence are determined by the coefficients of the quadrature formula which also appear in the function, $f(w)$. Hence, the characteristics of the stability chart are also dependent on the values or these coefficients. From knowledge of the type of functional relationship between z and w, it has been possible to develop some general tools useful for synthesis of completely new formulas having good stability charts. (A good stability chart is one where in a large region around the origin the function $z=f(w)$ approximates the identity mapping, $z=w$, hence $\mu_1=\lambda$.) One synthesis method is to impose less than the maximum number of conditions required for the determination of the coefficients by the polynomial method and then to chose the remaining coefficients in such a way as to maximize the good region. The greatest success, however, has been obtained by using what is called the "shifting technique." Often, a classical or mixed classical method has a stability chart which exhibits a good region extending into the right-half plane. An example of such a method is classical $[O_{12},C_{12}]$ shown in Fig. 2. If the chart were shifted such that the $u=0.2$, $v=0$ point coincided with the $z=0$ point and were renamed $w=0$, an excellent quadrature method for real-time computations would result. The mathematics to do this is quite simple and the resulting stability chart has yielded the two nonclassical methods $[O_{30},C_{31}]$ mod Gurk, and O_{33} mod Gurk.

Conclusions

At the present time digital computers seem to be more than competitive with analog computers and digital differential analyzers in the real-time solution of differential equations. UDOFT, for example, is more than adequately fast for its application. The flexibility of a digital computer is also of great value in the OFT problem. For example, much of an analog OFT is involved in operations other than the solution of the equations of motion of an aircraft. These "pinball machine" operations can be and are going to be performed by the digital computer. There appears to be much that can be done with digital computers in real time and it is hoped that this illustration of an application will stimulate more investigation.

References

1. The ENIAC, **J. G. Brainerd, T. K. Sharpless.** *Electrical Engineering*, Feb., 1948.

2. The Organization of a Real-Time Simulator. **H. J. Gray, Jr.** *Convention Record*, Institute of Radio Engineers, New York, N.Y., pt. 1. Radar and Telemetry, vol. 85, Mar. 1953.

3. Digital Computers for Real-Time Simulation, **M. Rubinoff.** *Journal*, Association for Computing Machinery, New York, N.Y., vol. 2, no. 3, July 1955, pp. 186–204.

4. A Digital Computer for Use in an Operational Flight Trainer, **W. H. Dunn, C. Eldert, P. V. Levonian.** *Transactions*, Institute of Radio Engineers, VEC-4, no. 2, June 1955.

5. A Technique for Real Time Simulation of a Rigid Body Problem, **H. J. Gray, Jr., M. Rubinoff, H. Sohon.** *Mathematical Tables and Other Aids to Computation*, Washington, D.C., vol. VII, no. 42, Apr. 1953.

6. Anal. vs Digital Computers—A Comparison, **Morris Rubinoff.** *Proceedings*, Institute of Radio Engineers, Oct. 1953, pp. 1254–62.

7. Numerical Solutions of Differential Equations, **H. M. Gurk, M. Rubinoff.** *Proceedings of the Eastern Joint Computer Conference*, Special Publication T-70, Dec. 1954, pp. 58–63.

8. Numerical Methods in Digital Real-Time Simulation, **H. J. Gray, Jr.** *Quarterly of Applied Mathematics*, Providence, R. I., July 1954.

9. Propagation of Truncation Errors in the Numerical Solution of Ordinary Differential Equations by Repeated Closures, **H. J. Gray, Jr.** *Journal of the Association for Computing Machinery*, vol. 52, no. 1, Jan. 1955, pp. 5–17.

10. The Use of Stability Charts in the Synthesis of Numerical Quadrature Formulae, **H. M. Gurk,** *Quarterly of Applied Mathematics*, Apr. 1955.

Discussion

Walter W. Varner: I have one afterthought on the Addaverter. I pointed out in my talk that it is quite difficult to get the three pieces of machinery working at the same time. I did neglect to mention that one of the other great difficulties is the fact that there are people in this loop, and getting people to work with these three pieces of equipment; that is, scheduled, and so forth, is also quite difficult.

I do not want to create the impression that we were able to predict the error exactly, but we were able to estimate the error well enough to determine whether or not the solution was good enough to meet the requirements of people who posed the problem for the machine.

Chairman Madden: A question for Mr. Gray from M. K. Haynes, IBM Corporation: "What method and formula for quadrature have you found to be optimum, and exactly what is the formula used in UDOFT?"

H. J. Gray: So far we have found optimum, a method called O_{33} Mod Gurk, and it is the equivalent (this method) to [O_{30}, C_{31}] Mod Gurk. Our optimum method gives us the largest region around the origin which is good. This method is obtained from the one shown in the paper, by shifting the point u equals 0.2 to the origin, and the mathematics for finding the coefficients is quite straightforward.

Chairman Madden: A question for Mr. McLeod by J. Murphy: "In your discussion you mentioned the resolver which produced a rapid answer to parts of your formula. Could you explain how the resolver works?"

John McLeod: For the purposes of this discussion the resolver is a black box which has two inputs and two outputs. In one mode of operation the input may be the slant range R of a radar target and gamma, the elevation angle. The outputs would then be R cos gamma, the surface distance, and R sin gamma, the elevation of the target. Conversely, in the other mode of operation, the surface distance R cos gamma, and the elevation R sin gamma may be the inputs, in which case the outputs would be the elevation angle gamma and the slant range R. As was indicated, in the most popular kind of resolver, the voltage representing gamma will position a servo driving one or more sine-cosine potentiometer across which the voltage representing R is applied.

Switching Transistors

I. M. ROSS
NONMEMBER AIEE

TRANSISTORS are being used more and more frequently as components of computing systems. This increasing popularity can be attributed to their small size, high efficiency, and potential reliability. The size and efficiency of transistors are sufficiently well known to require no further discussion here. On the reliability question, it is pertinent to note that transistors have been in operation in some telephone applications for several years with failure rates of about 0.05% per thousand hours. This figure compares favorably with that for the best vacuum tubes and there is every confidence that newer devices will exhibit much higher reliability.

The first part of this paper is devoted to a discussion of the electrical characteristics that make conventional transistors (n-p-n and p-n-p) attractive in computing applications. The second part deals with a family of 4-region (p-n-p-n) devices which are now under development and in some cases in early production. These devices exhibit a bistable characteristic and their use may lead to a considerable simplification of computer circuitry.

Conventional Transistor Switches[1]

A common form of switching transistor is shown in Fig. 1. Because such a device is frequently made by alloying techniques, it will be referred to it as an "alloy-type" transistor. One essential feature of such a structure is that the emitter and collector regions are metallic and hence, do not introduce appreciable series resistance in the emitter and collector leads. A further feature is that the emitter and collector junctions are opposite one another and are nearly the same size, the collector usually being somewhat larger than the emitter. Thus, the device is close to being symmetric. Because of the lack of series resistance in the collector and emitter leads and also because of the symmetry of the device, this transistor is probably the most suitable for switching applications. It is further the most amenable to accurate analysis.

The simplest form of switching circuit using an n-p-n transistor is shown in Fig. 2. If the base current is very small or even negative, the current through the load resistor is of the order of the leakage current across the reverse biased collector junction and is microamperes or less. Thus, effectively all the supply voltage appears across the transistor which therefore acts like an open switch. When a current I_b flows into the base lead, a current $I_b\alpha/1-\alpha$ will flow in the collector circuit. The collector voltage approaches zero as the collector current $I_b\alpha/1-\alpha$ approaches V/R_L. Effectively, all the supply voltage then appears across the load resistor and the transistor behaves like a closed switch. If the base current is increased beyond this critical value, the current that would be collected by a reverse biased collector becomes greater than the current that can be supplied through the load resistance from the voltage source. In fact, the collector junction becomes forward biased in order to reject or reinject some of the current and maintain the collector current close to the limiting value determined by the external circuit. Thus, for large values of base current, both the emitter and collector junctions are in the forward biased direction and the emitter to collector voltage, which is the difference of the two forward biases, becomes very small, as small as a few millivolts. This condition of operation with both junctions forward biased and the collector current saturated is referred to as the saturated condition. It should be noted that in the saturated condition, the emitter to collector voltage is less than the emitter to base voltage by the magnitude of the collector forward bias. Thus, if the collector of a saturated transistor is directly connected to the base of a second transistor the second transistor will be maintained in its off condition. The use of this mode of operation can lead to very simple switching circuitry.

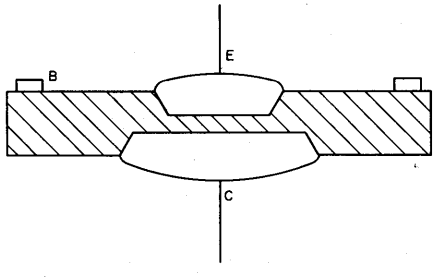


Fig. 1 (left). Section of an alloy-type transistor

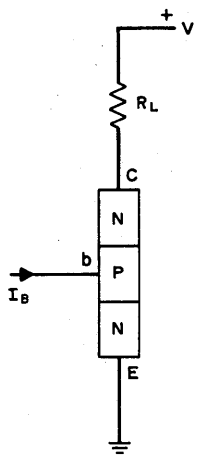
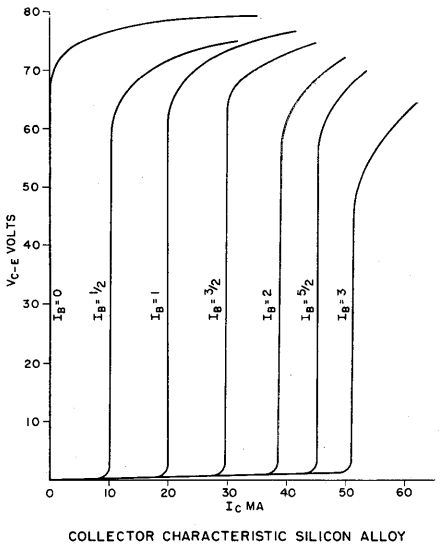


Fig. 2 (right). Simple transistor switching circuit



COLLECTOR CHARACTERISTIC SILICON ALLOY

Fig. 3. Collector characteristics of a silicon alloy-type transistor

A family of collector characteristics with base current as the parameter is shown in Fig. 3 for a silicon alloy-type transistor. For d-c operation, the most important parameters are:

1. The leakage current in the open or off condition. This depends on the dimensions of the device, the method of fabrication and most critically, on the material from which the device is made. For germanium transistors, the leakage current is of the order of microamperes at room temperature and doubles approximately every eight degrees Centigrade with increasing temperature. For silicon, the leakage currents are of the order of millimicroamperes and the temperature dependence is somewhat smaller than in germanium. For this reason, silicon transistors are preferred over germanium in many switching applications.

2. The "breakdown" voltage in the off condition. As the applied voltage approaches this value, the leakage current increases rapidly and the device no longer represents an open switch. Transistors can be designed to have breakdown voltages as high as 50 to 100 volts.

I. M. Ross is with Bell Telephone Laboratories, Murray Hill, N. J.

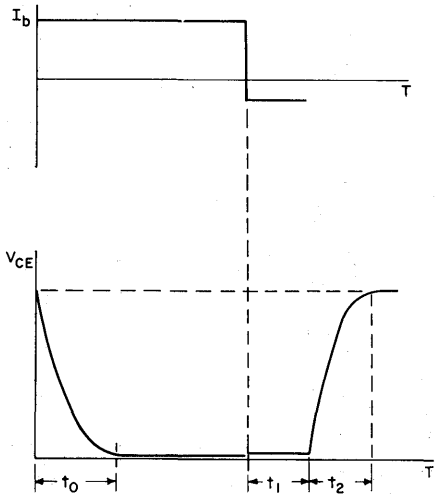


Fig. 4. Transistor switching transient

3. The voltage in the on condition which for alloy-type transistors can be as low as tens of millivolts.

4. The current amplification factor α- which determines the current gain, that is, the ratio of control to controlled current. In typical transistors, α is in the neighborhood of 0.9 to 0.99, giving a current gain in the circuit of Fig. 2 between 10 and 100.

Of prime importance are the times required to switch the transistor between the off and on conditions. These switching times depend not only on the parameters of the transistor but also on the particular circuit in which it is used. Unfortunately, no really satisfactory figures of merit have been devised to characterize the transient behavior of switching transistors. The discussion here is limited to a qualitative description of the transient effects and their dependence on transistor parameters.

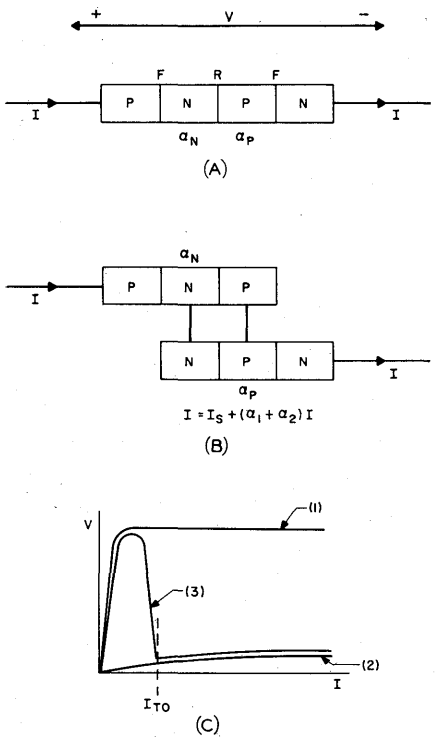


Fig. 5. Diode showing (A) schematic representation (B) 2-transistor equivalent circuit (C) VI characteristics

Fig. 4 shows the transients which occur when a large signal pulse is applied to the base lead of the transistor in Fig. 2. The collector voltage starts out along an exponential fall determined by the amount of drive, the current gain and the transit time for carriers from the emitter to the collector junction. This transit time, which is inversely proportional to the frequency cut-off, f_α, of the transistor, is determined primarily by the base thickness, decreasing as the square of the base thickness. Thus, the thinner the base region the shorter the turn-on time t_o. In the saturated condition, both the emitter and collector junctions are forward biased injecting charge carriers into the base region and, as a result, the amount of charge stored in the base during the saturated condition is greater than in the unsaturated condition. This additional stored charge is roughly proportional to the lifetime, τ, for minority carriers in the base and to the excess of base current over that necessary to drive the transistor into the saturated condition. When the base drive is removed, this additional stored charge must be removed before the collector voltage can start to rise. There is thus a storage time t_1 between the removal of base drive and the start of increase of voltage at the collector. The storage time t_1 depends on the amount of stored charge and the rate at which the charge can be removed. If the device is turned off by open-circuiting the base, the stored charge will decay with essentially the characteristic lifetime τ. If however, the device is turned off by applying reverse bias across the emitter-base junction, the stored charge will also be drawn out at the emitter contact and, in the limit, can be removed in a time close to the transit time across the base. Thus, the storage time, t_1, depends on the degree to which the device was driven into saturation, the lifetime in the base region, the transit time across the base and the nature of the turn-off signal applied to the base contact. At the termination of the storage period the collector voltage begins to rise exponentially at a rate which is again determined by the frequency response of the device.

The device parameter common to the three delay times is thus the frequency response of the transistor which is closely related to the width of the base region. Since speed is usually the most important characteristic in a switching circuit, the designer is frequently willing to trade increased speed for a decrease in some other performance parameter. The diffused-base transistor is a typical example of such a compromise. Using this structure, the device designer can achieve thinner base regions and hence, increased frequency response but usually at the expense of several ohms of series resistance appearing in the collector lead.

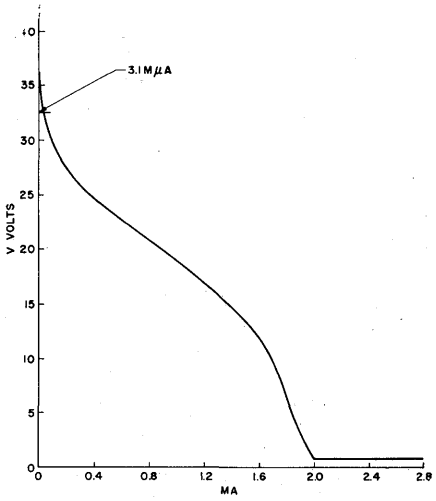


Fig. 6. VI characteristic of a silicon p-n-p-n diode

It should be emphasized again that the delay times observed in a transistor switching circuit depend not only on the properties of the transistor but also on the particular circuit used. Transistors are now available which, in practical switching circuits, yield turn-on, storage, and turn-off times each in the range of 10 to 100 millimicroseconds. Hence, switching rates of 10 to 100 megacycles

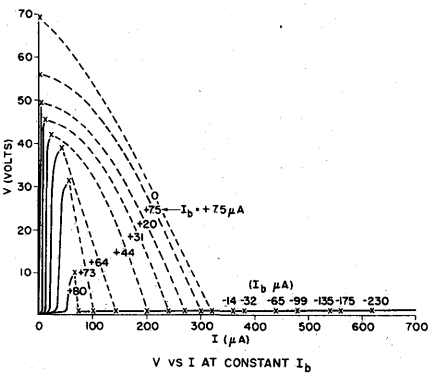


Fig. 7. Experimental characteristics of a 3-terminal p-n-p-n transistor

are practical. There is no doubt that the trend in manufacture towards higher frequency transistors will continue over the next few years and hence, devices should be available which will yield switching rates considerably higher than 100 megacycles.

Four Region Devices[2,3]

Fig. 5(A) shows a schematic diagram of a p-n-p-n diode biased so that current flows from left to right. This direction of current tends to make the outer two junctions forward biased and the middle junction reverse biased. The device is designed so that the outer regions are good emitters into the middle regions and that there is substantial transmission of carriers across the middle regions. Thus, we can look upon the device as being a combination of two transistors, as shown in Fig. 5(B), there being two separate emitters and a common collector. Defining the alphas of the p-n-p and n-p-n transistors as being α_n and α_p respectively, the current across the middle reverse-biased junction is the sum of the following three components: a leakage current I_s due to the bias across the middle junction itself, a hole current $\alpha_n I$ due to injection from the p-type emitter and an electron current $\alpha_p I$ due to injection from the n-type emitter. Since the sum of these three components must equal the total current I in the device,

$$I_s+(\alpha_n+\alpha_p)I=I \qquad (1)$$

which gives for I

$$I=I_s/[1-(\alpha_n+\alpha_p)] \qquad (2)$$

From equation 2 it is seen that, if the sum of alphas is less than unity, the current is a multiple of the leakage current, and therefore, the VI characteristic of the device is similar to that of a reverse biased p-n junction, curve 1 of Fig. 5(C). Now consider what happens if the sum of alphas is greater than unity (since each alpha can be as large as unity, the sum can easily be greater than unity). Equation 1 shows that for this case the injected current that would be collected if the middle junction were reversed biased is greater than the total current that flows. Just as in the case with the saturated transistor, the middle junction becomes forward biased to reject or reinject some of this current. Hence, for sum of alphas greater than unity, all three junctions are forward biased and the net voltage across the device is approximately equal to the forward bias across a single p-n junction. The VI characteristic is then similar to that of a forward biased diode, curve 2 of Fig. 5(C). Since, in silicon transistors, α increases with increasing current, it is possible to design a silicon p-n-p-n diode in which the sum of alphas is less than unity at low current and greater than unity at high current. The resulting VI characteristic is shown as curve 3 of Fig. 5(C) where I_{to} is the current at which the sum of alphas goes through unity. This characteristic is similar to that of a gas discharge tube and as with the gas tube a load line can be chosen to give two stable states of operation, one at low current and high impedance and the other at low voltage and low impedance. Unlike the gas tube, the sustain voltage, i.e., the voltage in the on condition, is very low, less than one volt. Hence, as a switch, a p-n-p-n diode can be much more efficient than the gas tube. Fig. 6 gives the characteristic of a typical silicon p-n-p-n device. This diode in the high impedance condition has a leakage current of a few millimicroamperes shunted by a capacitance of about 10 micromicrofarads. In the on condition the sustain voltage is about 0.8 volts and the dynamic impedance about 2 ohms.

The transient behavior of a 4-region diode is similar to that of the 3-region transistor being characterized by "turn-on," "storage," and "turn-off" times. The turn-on time is essentially determined by the transit times across the two bases. In developmental models, turn-on times as low as 10 millimicroseconds have been observed. Storage time is again determined by the lifetime, the transit times in the bases and the extent to which the device was driven into the low impedance condition. Little information is yet available on the values of storage time achieved in this device; however, values comparable to those for transistors should be possible.

An essential feature of the p-n-p-n diode is that it reaches the low impedance condition when the current through the device reaches a critical value, that is, the value for the sum of alphas equal to unity. In the 2-terminal device, current is caused to increase by the diode going into the breakdown region. An alternate way in which the current can be made to increase is to put a contact onto one of the middle regions and forward bias the outer junction. It is then possible to switch from the high impedance to the low impedance condition without biasing up to the breakdown voltage. The family of characteristics obtained from an experimental 3-terminal device is shown in Fig. 7 and is reminiscent of that for a thyratron tube. However, unlike the thyratron, it is possible to turn off the device by pulling current out of the base contact.

There are two other useful ways in which the p-n-p-n diode can be switched. The first method makes use of the fact that semiconductors are photosensitive. If sufficient light is shone on a p-n-p-n diode in the off condition, the current can be increased beyond the switching current I_{to}. The device can, therefore, be used as a photo switch. The second method involves the fact that there is a shunt capacitance associated with the reverse biased middle junction. Thus if a sharp rise in voltage is applied across the device, the displacement current necessary to charge the middle junction can turn on the device. Although this transient turn-on may find useful applications, it may also act as a speed limitation in other applications.

The bistable nature of the characteristics of p-n-p-n devices makes them attractive components for numerous computing applications. As illustrated in Fig. 5, the device is equivalent to two interconnected transistors and hence, its use can result in a considerable decrease in the number of components necessary to perform a given function. It is also possible that in the future, complete functional circuits can be built into one semiconductor device. For example, using p-n-p-n material, experimental models of 4-stage counters have been made, the transfer from stage to stage occurring totally within the semiconductor. These devices, which are electrically analogous to gas stepping tubes yet the size of a normal transistor, have operated at counting rates up to 1 million per second.

References

1. Large-Signal Behavior of Junction Transistors, **J. J. Ebers, J. L. Moll.** *Proceedings*, Institute of Radio Engineers, New York, N. Y., vol. 42, Dec. 1954, pp. 1761–72.

2. P-N-P-N Transistor Switches, **J. L. Moll, M. Tanenbaum, J. M. Goldey, N. Holonyak.** *Ibid.*, vol. 44, Sept. 1956, pp. 1174–82.

3. Three Terminal P-N-P-N Transistor Switches, **I. M. Mackintosh.** *Transactions*, Professional Group on Electron Devices, *Ibid.*, vol. 5, Jan. 1958.

Special-Purpose Tubes for Computer Applications

SAUL KUCHINSKY
NONMEMBER AIEE

SPECIAL purpose tubes have received acceptance as active elements in the performance of many fundamental electronic functions. Three general classifications of such tubes are of special interest in computer applications: beam switching tubes, in-line alphanumerical indicators (Nixie), and cathode ray readout tubes (Charactron, Typotron).

The beam switching tube is a high-vacuum electronic distributor and multi-output device which uniquely performs such functions as counting, timing, programming, sampling (telemetering), dividing, coding, matrixing (memory addressing), and binary decoding. It differs from other active elements in that a single cathode controls an electron beam to any one of ten constant current output positions, each of which has provisions for bistable beam locking and high impedance switching.

Gas-filled in-line alphanumerical indicators, commonly called "Nixies", represent another type of special purpose tube with characteristics not readily available by other means. Elements in the form of numerals, letters, or special characters may be selected by applying voltage to appear as a glowing cathode in a simple gas discharge. The phenomenon whereby the visual glow discharge is larger than the actual cathode permits the display of all in-line characters in a common viewing area. The device is an unusually efficient electronic to visual converter since almost all of the electronic energy is concentrated in a visual glow of relatively narrow optical bandwidth. The eye acts as a natural filter in distinguishing this glow in high ambient light. Both decimal and biquinary-type readouts are described.

Charactrons and Typotrons represent still another type of important special purpose tubes. These cathode ray tube types summarily described herein are performing increasingly important computer functions where high-speed printout and readout characteristics are primary requirements.

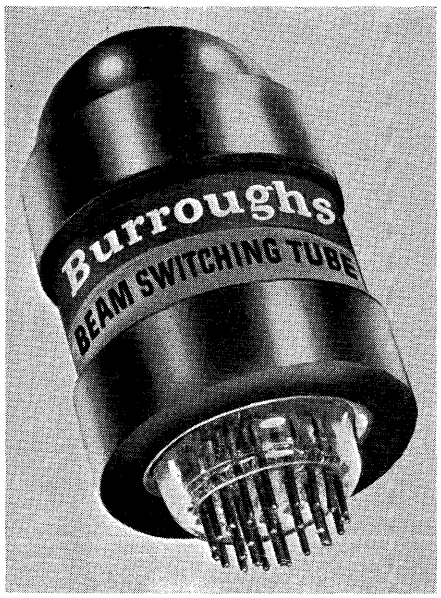


Fig. 1. Beam switching tube

Beam Switching Tubes

The beam switching tube is a 10-position high-vacuum constant current distributor (Fig. 1). It consists of ten identical "arrays" located radially about a central cathode (Fig. 2). Each array comprises, 1. a spade which automatically forms and locks the electron beam, 2. a target output which makes the beam current available with constant current characteristics, and 3. a high impedance switching grid which serves to switch the beam from target to target. A small cylindrical magnet is permanently attached to the glass envelope to provide a magnetic field which, in conjunction with an applied electric field, comprise the crossed fields necessary for the operation of this tube.

The tube may be made to operate in almost any conceivable distribution or switching mode such as 1. the tube may be in a clear or cut-off condition, 2. an electron beam may be formed in any one of the ten positions, 3. the electron beam may be switched sequentially, 4. the electron beam may be switched at random from any one position to any other position, and 5. the electron beam may be switched and cleared (cut itself off).

The impedances of the three electrodes are such that building block versatility exists. A distributor or switch of any number of positions may be obtained, since there are simple methods of arranging circuit parameters and interconnections between positions in the same envelope or between separate tubes. The same versatility permits accepting binary inputs directly for converting to decimal or other codes.

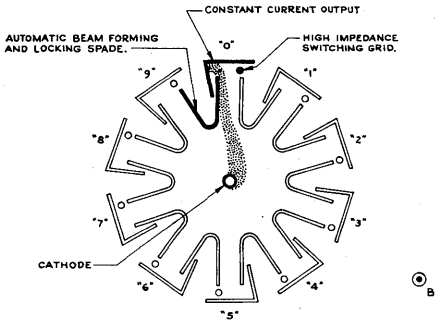


Fig. 2. Beam switching tube cross-section

Fig. 3 indicates a typical test and operating circuit for the type *6700* Tube. The operation of the tube can best be understood by studying the characteristics of the three basic elements comprising each array.

Spade Characteristics

Clear Condition

The spade electrodes directly affect the magnitude and shape of the electron beam in the area between the cathode and the spades. The tube will always be in the cut-off state when power is first applied if there are no provisions for beam forming. The spades are commonly connected to their supply voltage through individual series load resistors. When all of the spades are at $B+$ potential, the tube is equivalent to a magnetron diode in the cut-off condition, since the magnetic field prevents any electrons from reaching the outer arrays.

Beam Formation—Static From Cut-Off

The beam may be formed in any one of its ten "on" positions by sufficiently lowering the potential of the respective spade with either a d-c voltage or a high-speed pulse. Each spade has a negative resistance characteristic due to the crossed electric and magnetic fields. The resultant bistable states are shown in Fig. 4 by the solid "static" curve intersected by the series spade resistor load line. When the spade potential is lowered to approximately 60% of the spade $B+$, the negative characteristic will provide automatic lock-in at a point near zero or cathode potential. Thus, the one spade which forms and locks the beam is near cathode potential, while the remaining

Saul Kuchinsky is with Burroughs Corporation, Plainfield, N. J.

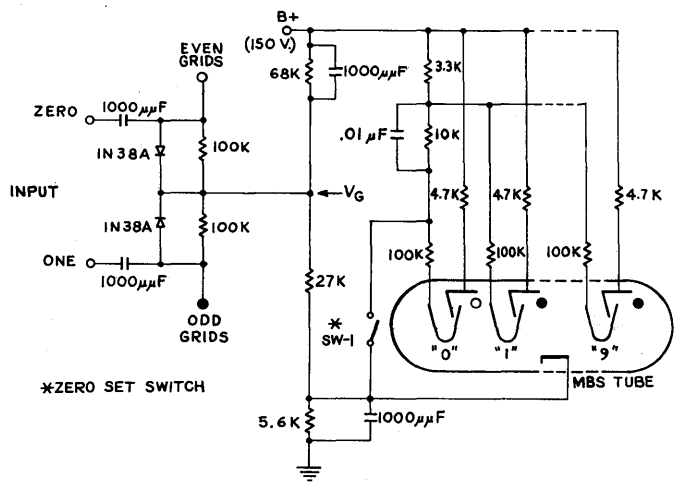


Fig. 3. Basic circuit of beam switching tube

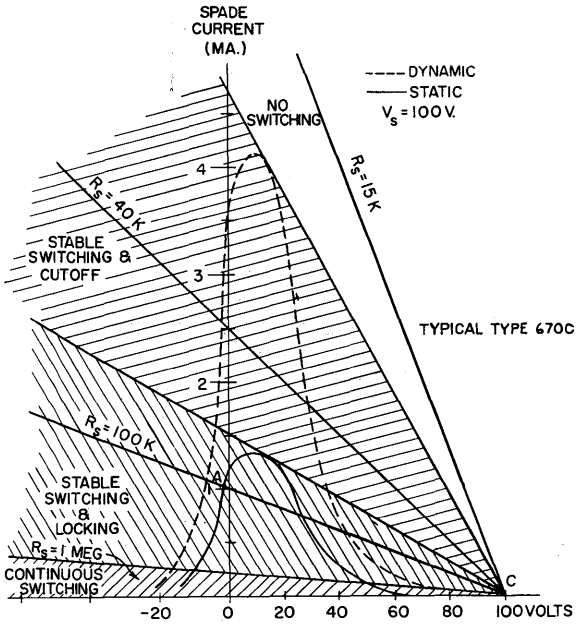


Fig. 4. Beam forming and locking spade characteristic

ones are at $B+$. The curved shape of the beam is the result of the radius of curvature of the electrons determined by the combination of electric and magnetic fields as shown in Fig. 5.

Beam Formation. "Dynamic"

When a beam has been formed on a spade, it can remain there indefinitely, or it can be advanced in many ways. One method is by lowering the switching grid voltage to a value where it will change the electric field in the area between spades so that enough of the electron beam is diverted to the leading spade to cause that spade voltage to be lowered and assume its locked-in stable state. The entire beam current is effective in quickly switching and lowering the potential of the leading spade. The lagging spade will remain at near zero potential for a longer time determined by its RC constant. An instantaneous condition results with two spades near cathode potential.

The resultant dynamic spade characteristic curve, indicated by the broken line shown in Fig. 4, is due to the broader electric field obtained by two spades being at near cathode potential dynamically during sequential switching. This characteristic indicates the ability to design switch and clear load lines useful for turning the beam on in a second tube after which the beam in the first tube will automatically clear or cut itself off.

The simplicity and wide operating range obtained by the spade resistor load line, whether used for beam forming and locking, beam switching and locking, or beam switching and clearing (using load line which intersects dynamic characteristic only) is indicated in Fig. 4.

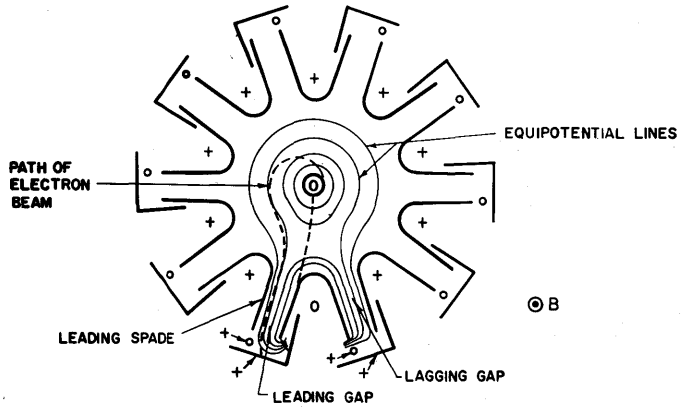


Fig. 5 (right). Beam formation. Equipotential lines

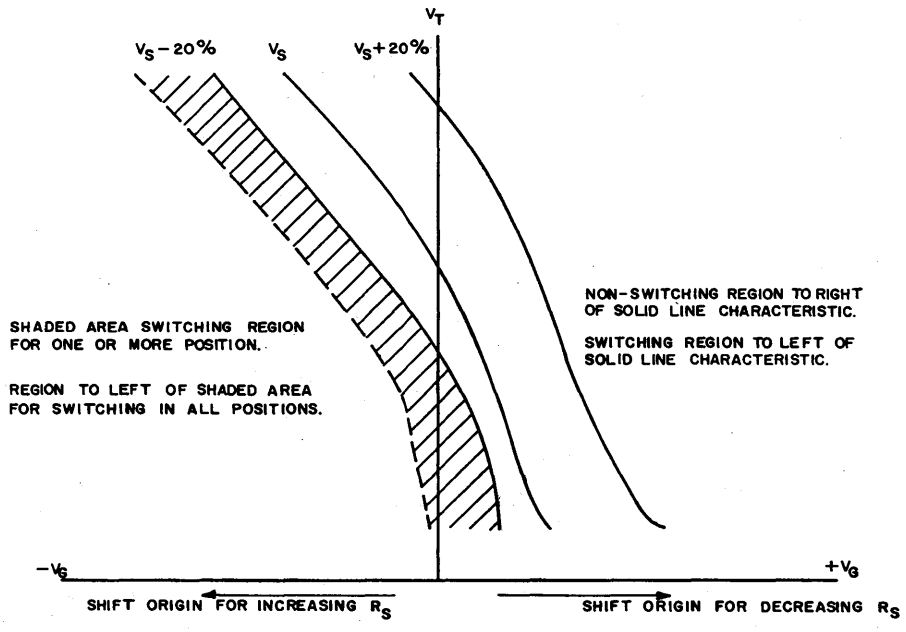


Fig. 6. Effect of operating parameters on grid switching characteristics

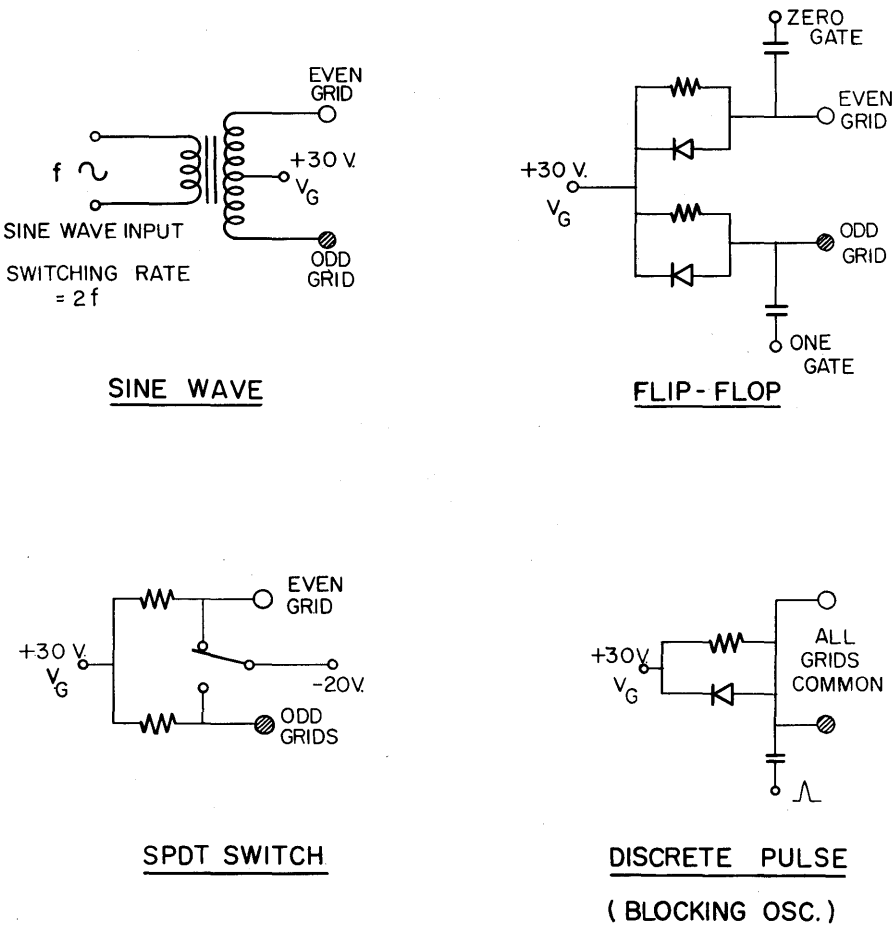


Fig. 7. Grid input circuits

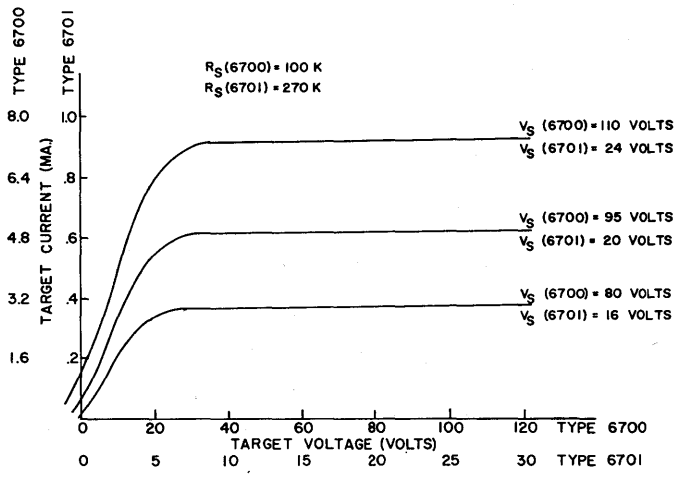


Fig. 8. Output characteristics

Setting to Zero or any Position

The beam may be formed to any position from the cut-off condition by either a pulse or d-c voltage as explained under Spade Characteristics. Once the beam is formed, it is usually advisable to clear and reset the beam to obtain random presetting. The basic circuit, Fig. 3, illustrates a simple one-switch method whereby a simultaneous clear and zero set action takes place. Closing the switch lowers the common V_s to below cut-off, clearing the tube. When the switch is opened, the common V_s recovers to $B+$ at a faster rate than the zero spade and its associated capacity, causing the beam to form to this position. This technique can be used in any or all positions and be accomplished electronically at high speeds.

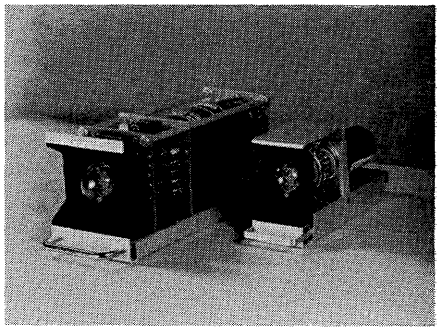


Fig. 9. Beam-switching-tube decade counters

THE SWITCHING GRID

The grid is normally the electrode used for sequential switching since it performs its function without drawing any appreciable current. Because of their shape and position, a negative voltage or pulse to the grid electrodes will effect a very fast and uniform switching. The fixed polarity of the magnetic field determines the direction of sequential switching which is clockwise as shown in Fig. 2.

Fig. 6 shows the relationship of other operating parameters to the switching characteristic. Biasing the grid voltage to the right of the characteristic will result in the beam remaining locked in. By obtaining the d-c grid bias from a resistive bleeder between the spade supply voltage and the cathode, the proper voltage relationship between these two elements will be maintained for stable operation despite comparatively large variations of power sources.

In each position the beam is affected only by the individual grid with which it is associated. The grids are connected in two groups, the odd numbered grids in one group and the even numbered in the other. Thus it is possible to use a d-c input in push-pull fashion and still secure single position stepping.

This binary type of input (flip-flop or push-pull) represents the most positive method of driving the beam switching tube. The requirements on such drivers may be made comparatively light because of the high impedance of the grids.

The adaptability of the grid to almost any type of switching input is illustrated in Fig. 7. Voltages are typical for tube type *6700*.

TARGET OUTPUT

The output efficiency of the beam switching tube is without equal as a multiposition current distributor. All of the beam current is put to work in the one selected position without any current being wasted in the other nine positions.

Approximately 15% of the beam current

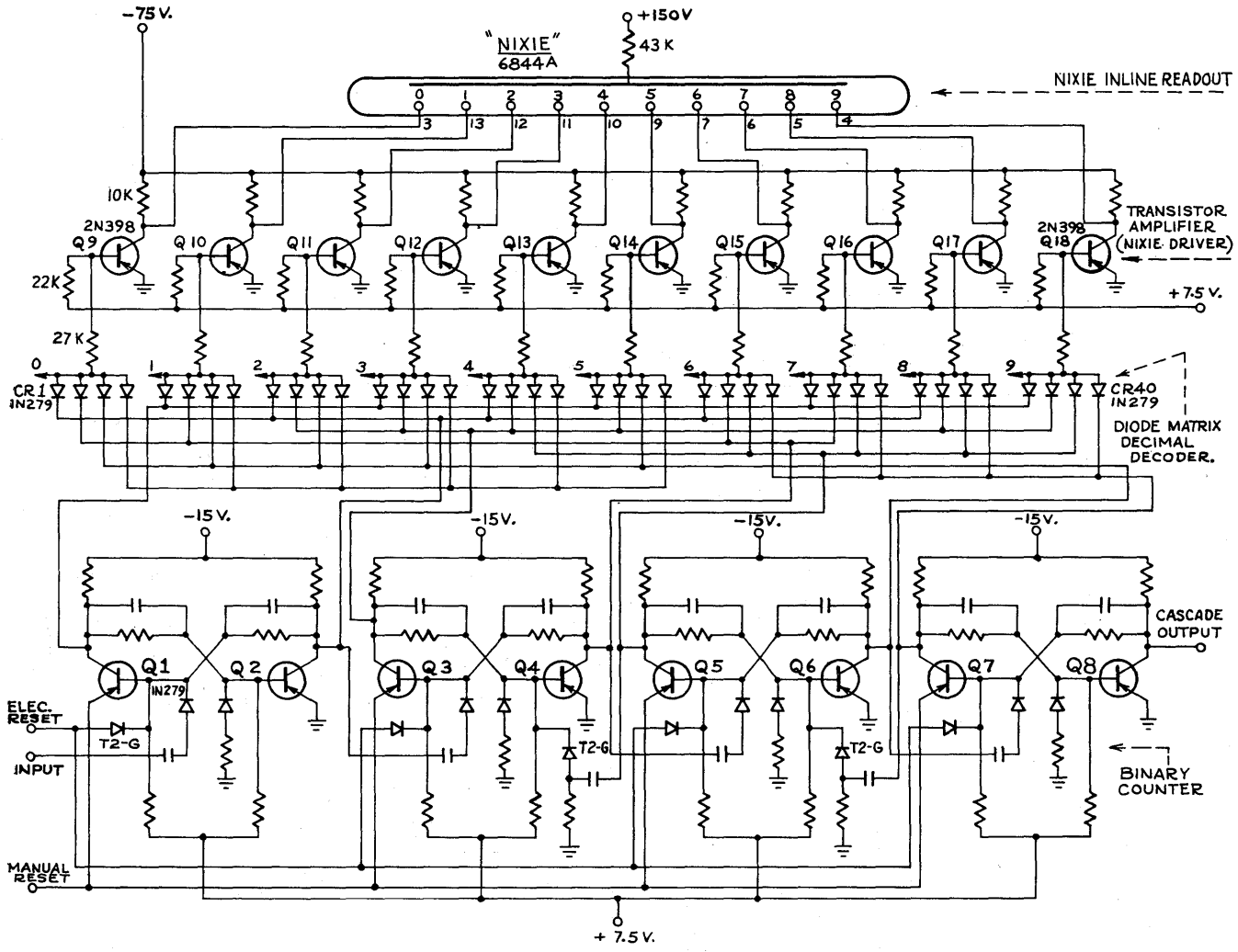


Fig. 10. Transister decade-counter circuit with Nixie readout

is used by the spade to automatically lock the beam while the other 85% is available at the target output with the constant current characteristics of Fig. 7. There is negligible crosstalk between outputs. Normally, the target output load line is made to intersect the straight portion of the curve approximately equal to the spade supply voltage. By operating closer to the knee of the curve, higher output currents may be obtained (using higher V_s) without exceeding its conservative one watt power rating. A wide variety of resistive load lines and target supply voltages can be used if the relationship $[V_t - (I_t R_t)] = V_s$ is observed. For example, 200-volt outputs may be obtained from a type *6700* Tube where V_s and R_s are selected for 5.5 milliampere, (ma) target current (typically $V_s = 100V$ $R_s = 100K$) by using a 37-K target load resistor and a 300-volt target supply. (In applications requiring target output voltages of 100 volts or greater, it is usually advisable to use 20 micromicrofarad, ($\mu\mu$f) or greater spade by-pass capacitors to prevent the large target pulse from causing spade switching instability. By the same token bypass condensers (0.1 microfarad μfd) are used to prevent inductive overshoot when driving relays. Diode clamps could also be used to prevent the target voltage going below the knee of its characteristic.)

The ability of the target to drive large capacities; e.g., long leads, without appreciably affecting normal operation is desirable in many electromechanical operations and represents another feature of the tube. The $B+$ efficiency of the beam switching tube as a 10-position distributor compares very favorably with either vacuum tube or transistor flip-flop techniques. Substantially all of the $B+$ current drain of the beam-switching tube circuit can be put to work in the one useful position.

Reliability

Beam switching tube reliability can be directly attributed to the principle of crossed electric and magnetic fields upon which its operation is based. Its stability does not depend on any gas or secondary emission principles which are difficult to control. Beam switching tubes have been tested thoroughly by many individual government and commercial agencies. Limits have been established on almost all possible mechanical and operational parameters to obtain reproducibility and dependability. Tubes are available to *MIL-E-1/1058* specifications.

Life

The beam switching tube is one of the most reliable of vacuum devices, with a life potential of 50,000 hours. Experience has indicated that a circuit may be designed reliably around beam switching tubes with an end point emission level of 85%. This compares most favorably with other components.

The lack of a close-spaced control grid inherently contributes to this factor. The currents and voltages used are many times less than those for which the cathode is rated. The particular beam shape and

the bistable tube characteristic which locks the receiving spade at near cathode potential both tend to minimize the effects of ion bombardment. The beam formation has the property of using a different portion of the cathode for each beam position in a manner which results in time sharing and minimizes the effects of poor emission.

Temperature

Beam switching tubes are noteworthy as being one of the electron devices least sensitive to either high or low temperatures. Operation with ambient temperatures of 100 degrees Centigrade, (C) have been common. Tubes with special processing schedules have been successfully made for continuous operation at 200 C and higher. Every completed tube and magnetized magnet assembly goes through a cycle at 150 C to permanently fix the silicone cement which holds the precise magnetic field alignment. Random tests made with repeated cycling at these temperatures have shown negligible affects.

Rugged Structure

The rugged box-like symmetrical structure is balanced and supported evenly at all points, both within the mica and to the glass envelope. It is firmly held through many tie points to the multilead stem. Finally, the tube floats within the rubberized silicone cement that attaches the glass envelope to the permanent magnet to obtain an additional protective effect.

Mount Assembly by Automation

The beam switching tube introduces the first use of a multielectrode assembly jig using principles of automation to a production vacuum tube. The 20 spade and target parts are precisely fixed and simultaneously inserted in both the top and bottom mica around the central cathode. The ten switching grid wires are inserted in a similar manner. This automation in the assembly of so many parts results in a premium ruggedness as borne out by shock and vibration tests.

Beam Switching Tube Types

The present family of beam switching tubes includes seven tube types. The selection of the proper beam switching tube is generally straightforward. The mount structures are geometrically and mechanically similar. The theory and application of all tube types are alike. Prime differences concern magnetic field strength, magnetic shielding, internal spade load resistors, external connections, physical dimensions, and the effects of these parameters on operating voltages, output current, and switching speeds. These tube types are:

1. General purpose *6700* (Fig. 1)
2. Low voltage *6701*
3. High speed *MO-10R*
4. Shielded general purpose *BD-301* (Fig. 9)

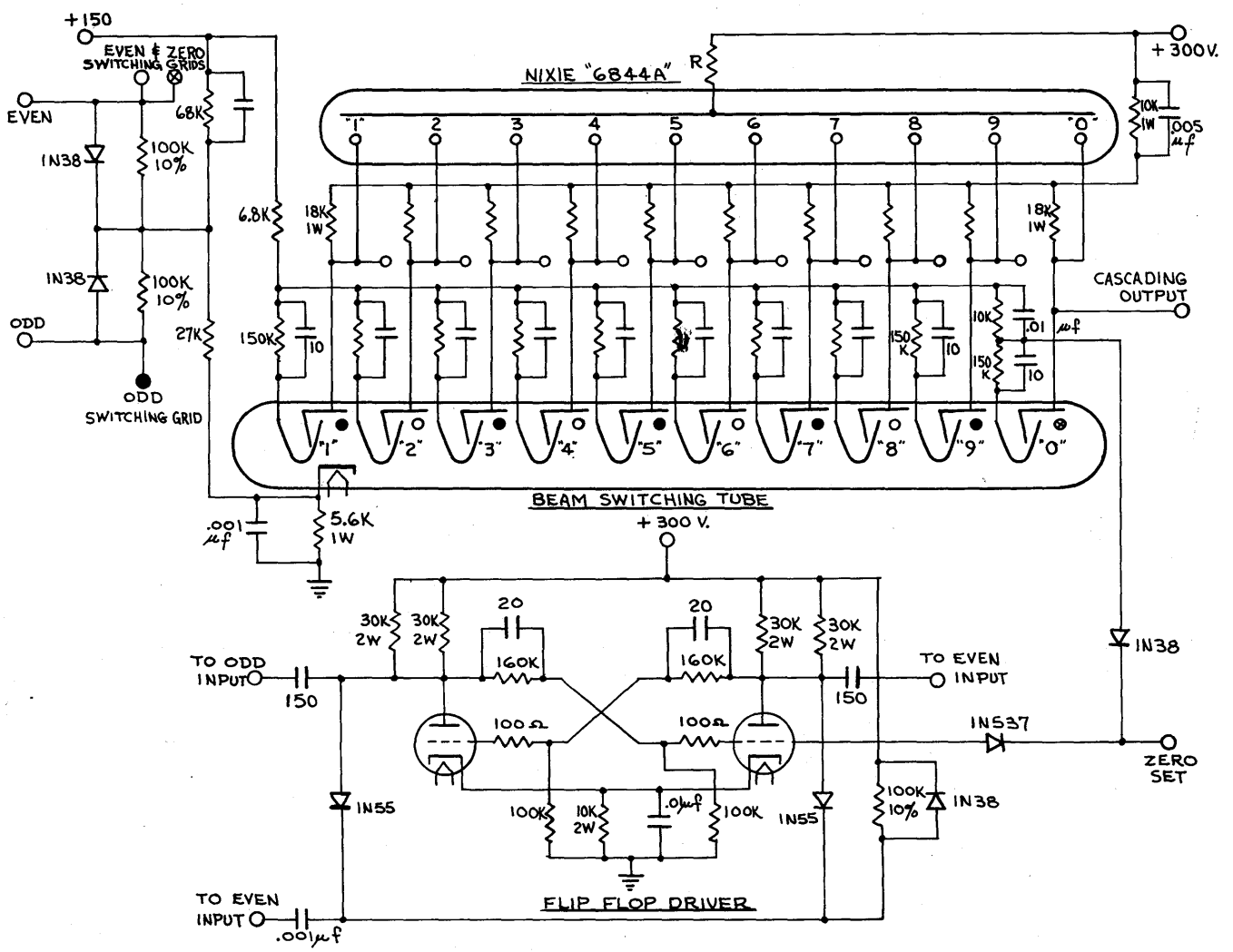


Fig. 11. Beam switching tube decade-counter circuit with Nixie readout

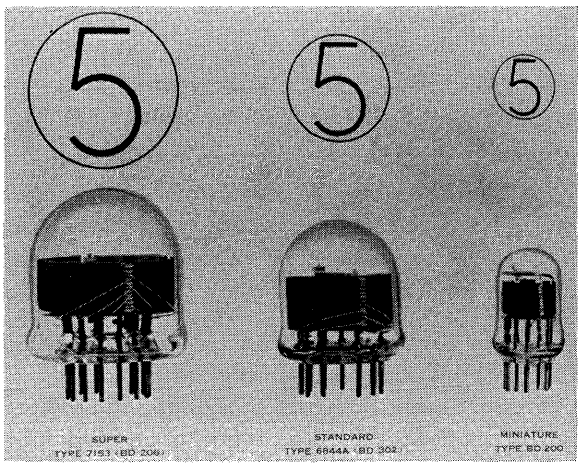


Fig. 12 (left). Nixie indicator tubes

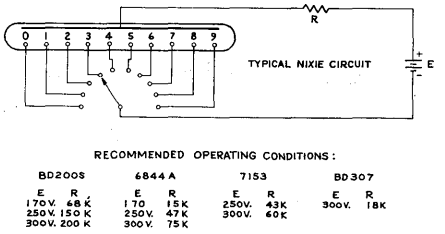


Fig. 13 (right). Nixie circuit

5. Shielded low voltage *BD-308*
6. Miniature *BD-203* (Fig. 9)
7. High current *BD-311*

The reduction in size obtained through the use of the miniature tube type *BD-203* is apparent in Fig. 9 illustrating two decade counters with Nixie readout. The larger commercial counter used type *BD-301*. The smaller experimental counter uses type *BD-203*. The larger tube types are still generally preferable where higher current outputs are required.

Beam Switching Tube Applications and Circuitry

The beam switching tube differs from other active elements such as tubes or transistors in that its single cathode controls an electron beam to any one of ten constant current output positions each of which has provisions for bistable beam locking and high impedance switching. The beam switching tube can replace the functions of from 4 to 20 vacuum tubes or transistors which could perform similar functions through the relatively complex technique of combining bistable or binary stages.

There appears to be a popular misconception that the total use of transistors will result in the ultimate technical characteristics of power, speed, versatility, reliability, serviceability, and cost, regardless of their function. To the contrary, there are numerous distribution and switching applications where the beam switching tube decisively outperforms transistors in these important considerations. Fig. 10 illustrates a transistor decade counter with Nixie readout as an example for comparison of the two techniques. The basic limitations of the transistors in this application is obvious. Each transistor represents a bistable element which performs the decimal distribution in a relatively devious manner. As is shown, it is necessary to use 8 transistors to perform the binary counting, 40 diodes to perform the binary to decimal conversion (where decimal output is required), and 10 additional transistors to amplify the diode matrix output sufficiently to satisfy the voltage requirements of the Nixie indicator tube. Fig. 11 indicates the use of one beam switching tube, therefore replacing the 18 transistors and 40 diodes. These advantages are surprisingly typical wherever decimal-constant current outputs are required. It is interesting to note that the $B+$ power requirements for the transistor version are considerably higher than the beam-switching tube version. This generally more than compensates for the heater wattage of the single beam switching tube. For example, the transistor power requirements for Fig. 10 are as follows: 70 ma at 75 volts, 2.8 ma at 150 volts, 12 ma at 7.5 volts, and 80 ma at 15 volts. The use of a high voltage n-p-n transistor, which is not yet available, might reduce the current requirements at 75 volts appreciably. By comparison the commercial beam switching tube decade counter type *DC-105*, which counts at 1.5 megacycles, presets to zero in less than 1 microsecond, operates Nixie, and has all ten decimal outputs available for printout, only requires 30 ma at 300 volts including requirements for its high-speed binary driver. The characteristics of this beam switching tube commercial counter are considerably extended in comparison with the transistor developmental counter.

However, when using such a decimal counter in computer or system work there are considerable periphery circuits required around the basic distributor or counter. These often logically can and should be performed by tube and transistor logic. The beam switching tube because of its high impedance inputs, its constant current outputs, and its operating voltage flexibility lends itself unusually well to combination with these other active elements. Careful technical evaluation will often indicate that the proper combination of beam switching tubes with transistors, tubes, and other active elements will represent the ultimate technique in the performance of many counting, distributing, and switching functions.

Electronic In-Line Readout Tubes. Nixie

Gas-filled indicator tubes represent a relatively new class of special purpose tubes which have recently received wide acceptance. This simple tube contains stacked elements in the forms of thin metallic numerals or letters. Application of a negative voltage to the selected character with respect to a common anode results in its becoming the cathode of a simple gas discharge diode. Only the selected information is visible in a common viewing area due to the phenomenon whereby the visual glow discharge is considerably larger than its thin metallic source.

The device is an unusually efficient electronic to visual converter since all of its electrical energy is converted into a neon glow of relatively narrow optical bandwidth. The eye acts as a natural filter in distinguishing this glow in high ambient light. Both decimal and biquinary-type indicators have been made.

This new tube type exhibits the following features: 1. all-electronic with a minimum of power required, 2. high-speed rate of change, 3. simplicity, 4. wide temperature operating range, 5. uniform characteristics from tube to tube and number to number, 6. human-engineered numerical design, 7. comparatively low cost, 8. small volume for number size, 9. light weight, 10. rugged, 11. good readability for number size.

Decimal Nixie

The most common type of Nixie is a gas-filled cold cathode tube which contains all the numerical digits "0" to "9". Three commercial numerical indicator tube types are shown in Fig. 12. These

tubes are similar in construction and characteristics with the major exceptions being size and voltage-current requirements. Current requirements vary from less than 1 ma for the smallest size to approximately 10 ma for a 3-inch diameter tube type. Life in the order of from 3,000 to 5,000 hours has been commonly attained. Special tube types with life in the order of 10,000 hours have been made.

Fig. 13 illustrates a simple circuit for operating Nixies directly from a switch. The switch could, of course, be replaced by a current source such as beam switching tubes, vacuum tubes, or transistors, supplying sufficient current to ionize the numeral completely. Since the Nixie has a common anode, it may be prebiased to approximately one-half of its 170-volt ionization rating, reducing the external voltage swing required (Fig. 9). D-C operating voltages are recommended. When using a-c or pulsed voltages, peak currents must be kept within ratings to insure long life.

The question is often asked, "How fast will Nixie 'count'?" The ionization time is a function of the applied voltage and is in the order of 5 to 20 microseconds. Nixie could be photographed at rates as high as 50 kc. However, in most applications the Nixie is only used to indicate the end result when a high-speed counter stops. Thus, a Nixie coupled to a megacycle beam switching tube counter will still read correctly whenever the counter stops long enough to be viewed.

Biquinary Nixie

Numerical indicators of the decimal type require one input lead for each cathode or character to be displayed or a total of ten leads per tube. Many systems and computers use binary logic and, therefore, require binary to decimal converters which can be relatively complex and expensive. The binquinary Nixie was designed to meet the requirements for both direct operation from binary logic and a means to reduce the number of leads for display systems using large quantities of Nixies at remote locations.

The biquinary tube is of the same mechanical construction and size as the decimal Nixie. It differs in internal mount construction in that the numerals are divided into two sections of five characters or digits each. An individual anode is associated with each grouping of five. In numerical biquinary tubes all the even numerals are associated with an even anode, while all the odd numerals are associated with an odd anode. A shield element isolates the tube sections electronically. Two cathodes (one even, one odd) may be simultaneously selected (by connecting characters in common 0-1, 2-3, 4-5, etc.), but only the one associated with the properly energized anode will ionize and be visible. In practice the anodes are operated from the lowest weight binary (0-1) which determines the odd or even display. The five pairs of cathodes are energized from simple resistive bleeders obtained directly across the proper binary outputs. Fig. 14 illustrates a typical biquinary test circuit.

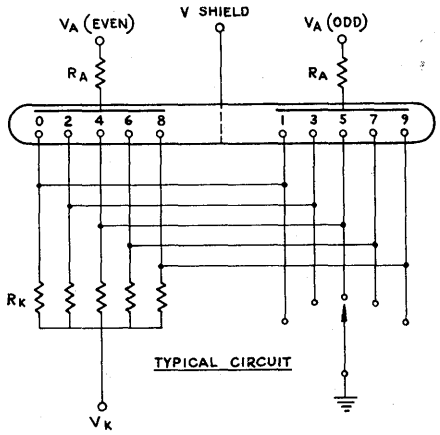


Fig. 14. Biquinary test circuit

Cathode Ray Readout Tubes

Special-purpose cathode ray tubes in the field of computer readout, flight information display, and high-speed printout have made major advances in recent years. As one example, the ability of tubes such as Charactrons and Typotrons to select and display all alphanumerical characters at high speeds have been developed to a very practical and useful art.

These tubes generally consist of an electron gun as means for focusing and selecting any one of 64 alphanumerical characters on a matrix plate and for positioning and displaying the selected characters as desired upon the tube face. Computer readout speeds of 15,000 characters per second are possible. Indications of the high-quality definition can be obtained from their ability to print any characters at any one of over a million discrete locations on a page. Control circuits necessary to operate these tubes are relatively complex but have been developed to a practical stage. Reference is made to papers presented at the 1956 and 1957 Wescon and Western Computer Conferences and at the 1957 and 1958 National Institute of Radio Engineers Conference for detailed descriptions on this family of tubes and their applications.

Conclusions

Three classes of special purpose tubes have been described which have received wide acceptance in recent years. Improved tube types and increased application can be predicted in the future for all three categories.

A low-voltage, high-current beam switching tube operating with $B+$ voltages from 5 to 10 volts and switching currents in the order of 20 ma is conceivable. Low-voltage Nixies operating directly from transistors also appear to be within the scope of practicality. Improvements in Charactron and Typotron characteristics will also undoubtedly be forthcoming, possibly with respect to reducing the complexity of the control circuitry and the tube design itself.

Consideration of special purpose tubes is indicated wherever their inherent characteristics lend themselves to the performance of functions not readily feasible by other means. The use of special purpose tubes will continue to offer unique solutions in many fields as the electronic art becomes more complex and the requirements for active elements become more severe.

Superconductive Devices

ALBERT E. SLADE
NONMEMBER AIEE

HOWARD McMAHON
NONMEMBER AIEE

OVER the past 2 years there have been several suggestions advanced for computer devices which utilize the unique properties of superconductors. Some of these take advantage of the strong nonlinear dependence of resistance on magnetic field which makes possible the construction of high-speed switches. Other proposed devices utilize the ability of a superconductor to carry a so-called persistent current which offers the possibility of constructing memory elements.

To date, a variety of devices of both types have been constructed and studied, and while many developmental problems remain to be solved, there is a strong conviction among those who are working in the field that the phenomenon of superconductivity will eventually provide useful computer devices.

Superconductivity

Superconductivity[1] was discovered by Kammerlingh Onnes in 1911. While in the process of extending electrical resistance measurements to the newly available liquid helium temperature range he discovered that the resistance of mercury dropped abruptly to zero at approximately 4 degrees above absolute zero. Soon thereafter many other metals were observed to exhibit similar characteristics. Onnes recognized that he had discovered a completely new state of matter and he named it the superconductive state.

Since absolutes are rare in nature it became a matter of considerable interest to determine if the resistance in the superconductive state is truly zero or only exceedingly small. Many experiments have been performed to detect any small residual resistance and it is now generally accepted that the resistance is indeed zero. It has been observed that a current induced in a closed superconductive circuit will continue to flow for years with no measurable attenuation. Currents of this type are known as persistent currents.

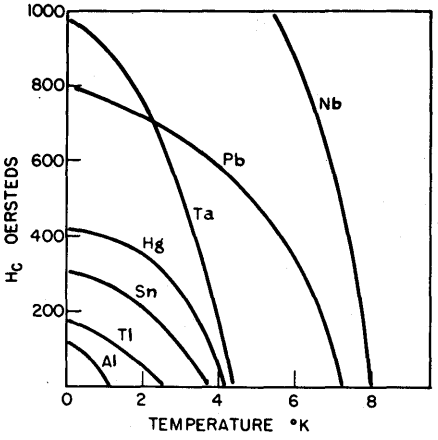


Fig. 1. Critical field versus temperature for various superconductors

It was soon found that a superconductive circuit cannot carry an indefinitely large current. In each case there is a critical current above which the metal recovers its normal resistance. Further investigation revealed that the superconductive transition is determined not only by the temperature but also by the magnetic field strength at the surface. The relationship of magnetic field and temperature for several known superconductors is shown in Fig. 1. Only the total strength, but not the direction, of the magnetic field at the superconductive boundary is effective in restoring the normal state. The limit on current-carrying ability of a conductor has been shown to be consistent with the local magnetic field strength resulting from the current.

A superconductor is unusual not only because of its infinite conductivity, but also because of its remarkable magnetic properties. It is to be expected that the magnetic induction inside an infinitely good conductor will remain constant with time, because any change in external magnetic field will result in the generation of undamped eddy currents which will cancel the effects of the external field change. It is also to be expected that any current flow will be confined to the surface of the body. Both of these conclusions can be derived readily from Maxwell's induction equation.

At first it was thought that this argument, which established B as a constant, was adequate to characterize completely the superconductive state. However, in 1933, it was discovered that the magnetic induction is not only invariant but equal to zero at all times, a fact which cannot be derived from the criterion of zero resistance. Thus the ideal superconductive state is not only infinitely conducting but, in addition, is also perfectly diamagnetic.

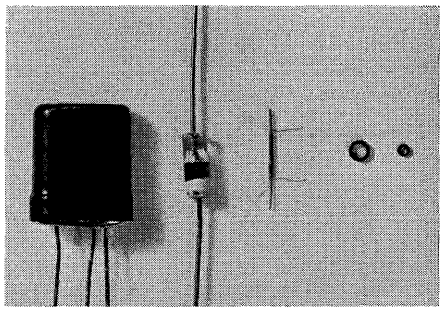


Fig. 2. Comparison of a transistor, a germanium diode, a cryotron, and two magnetic cores

The Intermediate State

It has been found that a real superconductor cannot be described simply as a body of zero electrical resistance. In fact the only superficial resemblance between real and ideal superconductors is that the electrical resistance is exactly zero for sufficiently small current flow. The precise way in which real superconductors differ from ideal ones is exceedingly complex and depends on such things as physical and chemical structure, amount of mechanical strain, shape of conductor, thickness of metal, and other factors.

Real superconductors, particularly in the form of thin films, show large deviations from ideal behavior. For example, the ability to carry current without becoming resistive is often lower than the ideal value by an order of magnitude. Also the strength of an externally imposed magnetic field required to destroy the superconductive state may be much larger than ideal.

Currently proposed theories of the intermediate state picture it as being composed of thread-like filaments of superconductive state and normal state somewhat like the fibers of a rope. A surface energy, either positive or negative, is ascribed to the boundary surface separating the superconductive from normal regions. This model is moderately successful in explaining most of the observed complexities, but is not sufficiently complete at present to eliminate the need for experimentation in the design of superconductive devices.

The Cryotron

Dudley Buck[2] was the first to consider seriously the potentialities of superconductivity in relation to computers. He proposed to utilize the sharp resist-

ALBERT E. SLADE and HOWARD O. McMAHON are with Arthur D. Little, Inc., Cambridge, Mass.

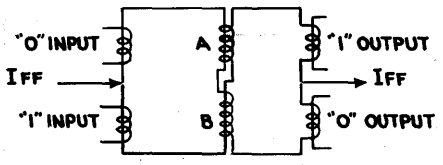


Fig. 3. Flip-flop circuit

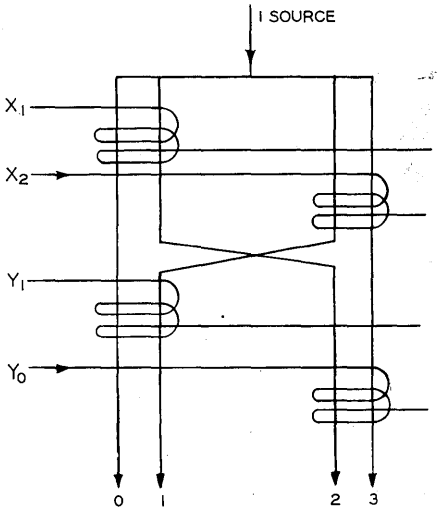


Fig. 4. Single-pole 4-position cryotron switch using multiple-gate cryotrons

Table I. Characteristics of Various Active Superconductive Switches

	Present Wire-Wound Cryotron	Proposed Wire-Wound Cryotron with Gate Deposited on Superconductive Core	Present Vacuum-Deposited Cryotron	Research May Make This Vacuum Deposited Cryotron Possible
Resistivity at 4.2 degrees K	10^{-6} ohm-centimeters, (cm)	10^{-6} ohm-cm	10^{-7} ohm-cm	10^{-6} ohm-cm
Cross-sectional area of gate	5×10^{-4} cm²	$T\pi d$	16×10^{-7} cm²	3.8×10^{-6} cm²	5×10^{-7} cm²
Length of gate that is quenched	0.5 cm	0.5 cm	0.1 cm	0.05 cm
Resistance of quenched area	10^{-3} ohms	3×10^{-1} ohms	2.6×10^{-3} ohms	10^{-1} ohms
Inductance of control	20×10^{-9} henry	8×10^{-9} henry	15×10^{-9} henry	1×10^{-9} henry
Inductance of gate	24×10^{-9} henry	24×10^{-9} henry	24×10^{-9} henry	1×10^{-9} henry
Inductance of cryotron	44×10^{-9} henry	32×10^{-9} henry	39×10^{-9} henry	2×10^{-9} henry
L/R time constant	40 microseconds	0.1 microseconds	15 microseconds	0.02 microseconds
Gain	3	2	2	2
Gain$\times R/L$	0.075×10^{6}	20×10^{6}	0.13×10^{6}	100×10^{6}
Complementing time of a flip-flop	200 microseconds	0.5 microseconds	75 microseconds	0.1 microseconds*

* *Note:* Very little is known about the thermal time constant of a cryotron. It is very likely that heat dissipation will limit the maximum repetition rate.

ance change with field strength to make an electronic switch, and suggested the name of "cryotron" for such a device. As a matter of convenience, Buck's cryotrons took the form of an insulated tantalum wire 0.009 inch in diameter, called the gate, around which is wound a 0.003-inch diameter niobium wire (called the control). Current in the control creates an axial magnetic field of sufficient strength to cause the tantalum gate to switch from the superconductive to the resistive state. The niobium control coil remains superconductive because its critical field is much higher than that of tantalum (see Fig. 1). Fig. 2 compares the size of a transistor, a germanium diode, a cryotron, and two magnetic cores.

Current Gain

The current gain of a cryotron can be defined as $I_{g(\max)}/I_{c(\min)}$, where $I_{g\,(\max)}$ is the maximum current the gate can carry without restoring its own resistance, and $I_{c(\min)}$ is the minimum current in the control coil necessary to make the gate resistive. If this gain is greater than one, then the current flowing in a gate can be sufficiently large to be used to energize a control coil connected in series. In cryotron circuitry it is only necessary for the gain to be greater than one, since the current flows in series through all of the output gates and coils. If it were necessary for the output of a single flip-flop to drive several loads that are in parallel, then the gain would have to be much greater.

However, in cryotron circuits, because of the zero d-c impedance, it is not feasible, in general, to utilize parallel circuits and a current gain slightly in excess of unity is therefore adequate. The ability to connect an indefinitely large number of controls in series with a single gate even though the current gain is only slightly greater than unity is advantageous under certain circumstances. However, the advantage is purchased at the expense of reduced frequency response, because of the increased inductance.

Frequency Response

The frequency response of a cryotron circuit is governed by the L/R-time constant of the current path which is in the process of being changed. In each circuit considered there will be a number of inductances and a number of resistances. However, the fundamental time constant of the cryotron as a device can be considered to be the inductance of a single control coil divided by the resistance of a single gate.

The time constant of the early wire-wound cryotron is about 100 microseconds. This obviously leaves much to be desired and is undoubtedly the most severe disadvantage of the cryotron as originally conceived.

Since the time constant is L/R or some small multiple, it can be shortened by reducing L or increasing R. Increasing R by increasing the length of the cryotron is not fruitful because L increases at the same rate as R. R can be increased by reducing the diameter of the wire but this raises many practical problems related to construction feasibility. R could also be increased by depositing a thin film of gate material on a nonconducting rod, thereby increasing the resistance of the gate because of its reduced cross-sectional area but keeping the size of the device unchanged. The resistance might also be increased by using an alloy of higher resistivity than tantalum.

There are fewer possibilities of reducing the inductance of the control coil. Reducing the diameter of the coil would reduce its inductance, but in order to preserve current gain it would be necessary to increase the number of turns per inch in proportion, which would cancel the advantage. The inductance may be reduced considerably by using a film gate as previously mentioned where the inside of the rod is filled with a superconductor. The control coil flux cannot penetrate the superconducting core and therefore, the effective diameter of the coil is reduced. The core must, of course, remain superconductive when the gate is quenched and it must be electrically insulated from the gate.

In Table I, the gain and time constant of the present cryotron and several proposed cryotrons are tabulated for comparison. Thus, it appears almost certain that an increase of two orders of magnitude can be achieved in the speed of cryotrons when used as active circuit components.

Cryotron Circuitry

Buck has discussed a variety of logical circuits that utilize cryotrons.[2] The nature of cryotron circuits differs considerably from those which use vacuum

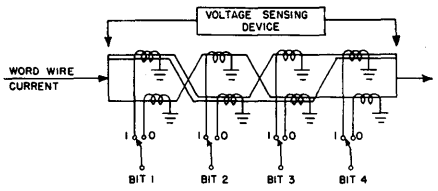


Fig. 5. Woven memory

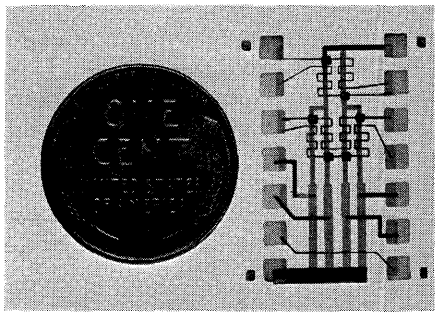


Fig. 6. A vacuum-deposited cryotron tree switch

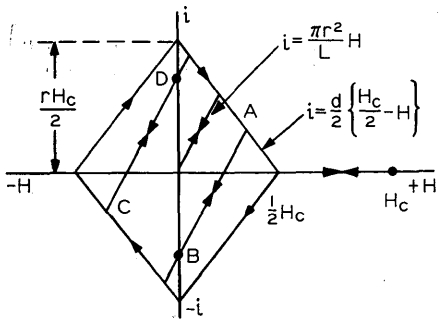


Fig. 7. Hysteresis of current in a closed ring as a function of applied axial field

tubes and transistors. Cryotrons more nearly resemble relays and as might be expected the logical circuitry is also similar. Because of the exceptionally low resistance involved, a current source is used. The current is usually allowed the choice of two paths, one of which is resistive, the other superconductive. Because of the infinite resistance ratio between the resistive gate and the superconductive gate, all of the current flows in the superconductive path.

A flip-flop with input and output gates, such as is shown in Fig. 3, illustrates the use of cryotrons in switching circuits. The actuating current I_{ff} is a constant regardless of the state of the flip-flop. If the current is flowing through the gate of cryotron A then it also flows through the control of cryotron B, making gate B resistive. The circuit is therefore bistable, because the current, once established in one of the two possible paths, will lock-in by causing the other path to be resistive.

The state of the flip-flop can be determined by measuring the resistance of the output gates; one gate will have zero resistance, the other a finite resistance. The state can be changed by energizing the control coil of the appropriate input cryotron, thus imposing a resistance in the conductive path and causing the current to divert into the alternative circuit. The characteristics and operation of the flip-flop and other cryotron circuits have already been described in more detail in other papers.[2,3]

Multiple Gate Devices and Circuits

An important distinction between transistor and cryotron circuits is the possibility in the latter of combining many devices into one. In some cryotron circuits the over-all system is in reality a single device.[4] Fig. 4 illustrates a 4-position switch in which the current labelled $I_{(source)}$ has a choice of four paths, each of which threads a unique course through the control coils. When the control coils are energized in pairs *i. e.;* X_1 or X_2 and Y_1 or Y_2, there is one, and only one, superconductive path through the switch. Any number of paths can be added to the switch by increasing the number of pairs of control coils. With ten control coil pairs there can be 2^{10} or 1,024 unique paths. It has been pointed out that such a switch could be constructed using more or less conventional weaving techniques. A 1,024-position woven switch using 0.002-inch wires would resemble a rope about a quarter of an inch in diameter and about 2 inches in length.

Another circuit similar to the multiple gate switch of Fig. 4 is the woven memory.[4] The woven memory, Fig. 5, is a logical network of multiple-gate cryotrons in which all of the stored information is permanent. Control coils are arranged in pairs, and wires are threaded through either the "one" or the "zero" coil of each pair. The memory differs from the switch in the respect that only a fraction of the possible paths contain a wire. Each wire woven into the memory represents a stored word. All word wires originate at a common junction and terminate at another common junction, and a constant total current is caused to flow through them. For any combination of parallel binary input currents to the pairs of control coils, there is only one possible superconductive path through the memory. If a word wire exists in that path, there will be no voltage drop across the memory. If that word wire were purposely omitted from the memory, the current is forced to flow through the remaining resistive paths. The presence or absence of a word is detected by monitoring the voltage drop across the memory. This type of static memory may be useful in translation and for function tables.

Vacuum-Deposited Cryotrons

The low switching speed and tedious interconnection problems associated with wire-wound cryotrons has led to a study of vacuum deposition techniques for constructing superconductive switching devices.

A circuit consisting of one or many cryotrons can be made by vacuum-evaporating in succession, layers of suitable metals and insulations through a series of masks. Fig. 6 is a vacuum-deposited, 4-position switch constructed in this manner. It consists of a deposition of tin gates which form the conducting paths of the switch, a subsequent deposition of silicon monoxide insulation in selected areas, a deposition of lead control crossings, and finally a deposition of silicon monoxide encapsulation. Vacuum deposition techniques offer the double advantage of increased switching speed, because of miniaturization, and simplicity of construction. Entire circuits can be made in a single operation, thus eliminating interconnection problems.

Vacuum-deposited cryotrons are still in the research stage. The nature of a thin superconductive film is not well understood at the present time. The restoration of resistance under the influence of external magnetic fields and internal currents cannot be predicted for films of various thicknesses even for pure materials. For this reason it is not possible to predict current gain and frequency response (or switching speed) from existing theory. The third column of Table I indicates the present state of the art for active circuit elements now possible. A considerable improvement is expected with further research and development.

Deposited film techniques of exactly the same kind havebeen used to construct high-speed passive switches. Fig. 6 is an example of a passive tree switch. Since there is no requirement for current gain in this case, the switching speed is proportional to the inductance of the gate only, and is independent of the inductance of the control. Optimum switching speed would be realized when the length of the

gate conductors is made small compared with the length of gate that can be made resistive. Maximum switching speed has not yet been determined for this configuration.

Persistent Current Devices

Several methods for utilizing persistent currents for memory devices have recently been discussed at the Fifth International Low Temperature Conference at Madison, Wisconsin[5,6,7] and elsewhere.[8] These devices are based on the establishment of a persistent current in a superconducting loop which will remain indefinitely until its direction is reversed by an external agency. The direction of the current is used to signify the storage of a "one" or "zero" bit of information.

The magnitude and direction of the current are determined in a somewhat complicated way by the magnetic or electrical history of the loop. Consider the case of a superconducting ring cooled in a region of zero magnetic field. If it is exposed to an axial magnetic field H, a circulating current will appear in the ring. The current is of such a magnitude and direction that it preserves the initial state of zero flux density within the area encircled by the ring. For a ring of radius r and a wire assumed to be small in diameter, compared with r

$$\pi r^2 \frac{dH}{dt} = L \frac{di}{dt} + iR = L \frac{di}{dt}$$

since the resistance R is zero. Integration gives the current i,

$$i = \frac{\pi r^2}{L} H$$

When the imposed field reaches a certain value, the vectorial sum of the field caused by the supercurrent and the imposed field becomes equal to the critical field at the outer edge of the ring and begins to restore resistance to the ring. At this point the total field at the outer edge of the ring is the sum of two terms. The contribution of the imposed field is $2H$ because of the distortion of the magnetic field about the circular cross-section of the superconducting wire that forms the ring and the contribution of the circulating current is $4i/d$, where d is the diameter of the superconducting wire. Hence the maximum value of i is given by:

$$2H + \frac{4i}{d} = H_{\text{critical}}$$

As the imposed field is increased further, the circulating current diminishes in such a way that the vectorial sum of the two fields remains at the value H_{critical}. Fig. 7 shows a hysteresis plot of the current in a closed ring as a function of the applied magnetic field. The boundaries of the diagram are determined by the requirement that the tangential component of the total magnetic field at any point on the surface of the ring must not exceed the critical field. The diagram can be circumscribed only in the clockwise direction. Any change in H that does not intersect the boundary produces a reversible change in i that is completely cancelled when the change in H is reversed. However, if H is changed in such a way that it exceeds the diagram boundary, the persistent current remaining in the ring when H is returned to its original value is permanently altered and may even have a reversed direction.

A superconducting ring is used as a memory element in the following manner. A positive transferse field somewhat less than $1/2H_c$ is applied and then removed. A persistent current is thus stored in the ring as represented by point B in Fig. 7. If the same positive transverse field is again applied and removed, the current will reverse along line AB and always leave the same persistent current stored in the ring.

A negative transverse field will change the current along line BCD and the direction of the current is reversed. A positive current can represent a "one"; a negative current, a "zero."

Readout can be accomplished by observation of the net change of flux within the ring. If the persistent current does not change direction, the output is symmetrical and can be integrated to zero; if the persistent current does change direction, the output is not symmetrical, and integration does not yield zero.

Details of various readin and readout schemes have been discussed in recent conferences and publications. Switching times as short as 10 millimicroseconds have been reported for a memory device similar in principle to the one just described.[8]

Persistent currents can be excited in superconducting loops by other than inductive methods, and several memories have been suggested which utilize such methods. Consider a circuit consisting of two superconductive self-inductances in parallel. Assume one of these to be a straight conductor which, of course, has a very low inductance; and the other to be a coil which has a much larger inductance. When a current pulse is applied to the parallel combination, it will divide according to the ratio of inductances. The straight conductor will carry the most current, and for some critical value, will become resistive due to the strength of its self-field. At this time the current will redistribute itself into the larger inductance. When the current pulse is removed, the straight conductor again becomes superconductive and the energy stored in the larger inductance will cause a current to persist in the ring formed by the two inductors. If the element is pulsed again in the same direction, the persistent current does not change direction. If the element is pulsed in the reverse direction, the persistent current is reversed, and an output voltage can be observed either with a small secondary coil or by measurng the voltage directly across the large inductance.

The elements are extremely small, and it is reported that a packing value of 300,000 elements per cubic foot can be achieved.

Conclusions

The speed of operation of wire-wound circuits can be improved in many ways. A direct reduction in size would give an increase in speed which is inversely proportional to the square of the dimensions, but for the wire-wound device, any major improvement in this direction is limited by construction and handling difficulties. Superconductive alloys with resistivities higher than those of tantalum may be found for use as gate materials, but it is unlikely that an improvement of more than an order of magnitude would result from this alone. Where current gain is not required, gate wires of smaller diameter may be used, and an arbitrarily long section may be restored to the resistive state, thereby achieving an arbitrarily high switching speed.

For switches which must have current gain, however, if an improvement of more than perhaps two orders of magnitude is to be achieved, a reduction in physical dimensions is certainly necessary. For maximum switching speed it is clear that thin superconductive films offer the greatest possibilities. Switching speeds already observed for some of the persistent-current memory elements[8] utilizing deposited films are of the order of 10 millimicroseconds.

Further development of deposited film techniques and devices should make the fabrication of extended and complicated circuitry a fairly straightforward matter. With the high speed and high density of logical elements which could result from this type of construction, it seems inevitable that superconductivity

will find practical applications to computer circuits.

References

1. SUPERCONDUCTIVITY, **D. Shoenberg.** Cambridge University Press, Cambridge, England, 1952.

2. THE CRYOTRON—A SUPERCONDUCTIVE COMPUTER COMPONENT, **D. Buck.** *Proceedings*, Institute of Radio Engineers, New York, N. Y., vol. 44, 1956, p. 44.

3. THE CRYOTRON CATALOG MEMORY SYSTEM, **A. E. Slade, Howard O. McMahon.** *Proceedings*, Eastern Joint Computer Conference, AIEE, Dec. 12, 1956.

4. THE WOVEN CRYOTRON MEMORY, **A. E. Slade.** *Harvard International Symposium on the Theory of Switching*, Cambridge, Mass., Apr. 2–5, 1957.

5. A COMPUTER MEMORY ELEMENT EMPLOYING SUPERCONDUCTING PERSISTENT CURRENTS, **E. C. Crittenden, Jr.** Program of the Fifth International Conference on Low Temperature Physics and Chemistry, Madison, Wis., Aug. 26–31, 1957. (In abstract form.)

6. THE PERSISTATRON, **M. J. Buckingham,** *Ibid.*

7. TRAPPED FLUX SUPERCONDUCTING MEMORY, **James W. Crowe.** *IBM Journal of Research and Development*, New York, N. Y., vol. 1, no. 4, Oct. 1957.

8. *Ibid.*, p. 295.

Magnetic Switching

JAN A. RAJCHMAN
NONMEMBER AIEE

IN THE last decade, magnetic cores made of square hysteresis loop materials have become important in the manipulation of digital information. New materials, devices, circuits, and systems are still being discovered and invented a an unabated rate.

This paper attempts to bring to light some of the underlying philosophy of many devices and circuits employing cores by describing selected examples.

A core with hysteretic properties has an infinite number of possible remanent states and offers the possibility of storage of information by means of the value of flux at remanence. Cores which have a square hysteresis loop have, in addition, thresholds of current required for producing irreversible flux changes and clearly defined saturation limits of flux (Fig. 1). Therefore these cores can be used for switching as well as storage. There are current-driven magnetic circuits depending on the current thresholds, and voltage-driven circuits depending on the flux limits.

Cores Directly Driven by Electronic Devices

Cores are passive devices which in high-speed applications must be driven by active electronic devices, tubes, or transistors. Perhaps the simplest applications are those in which the cores are driven and sensed directly by electronic devices and do not drive each other. The current-coincident memory arrays and the combinatorial switches are the outstanding examples of such directly driven circuits. The selection system in memory arrays in which the magnetomotive force on the unselected cores is at most half that on the selected core is well known and will not be considered here.[1–3]

Combinatorial switches can be made to select one, or several outputs from a large group. This is accomplished by using a number of windings each linking certain cores in series in such a way that energizing selected groups of windings will produce an algebraic sum of magnetomotive forces which exceeds the threshold of switchover on the selected core only. There is a considerable difference between such switches and the current-coincident memory, which is also combinatorial, in that the nonselected cores in the switch can support arbitrarily large magnetomotive forces tending to drive them in the direction of their existing saturation, whereas this is not permissible in the memory arrays because the state of remanence is unknown by definition. Typical are the following types of combinatorial switches.

A switch to select one out of *nxm* outputs by the choice of one input among *n*, and one input among *m*, can be made using an array of *nxm* cores. The cores are linked by *n* rows and *m* columns and also by a common winding carrying a d-c biasing current.[2] The input excitations switchover the selected core, and that core only, as one of the excitations neutralizes the d-c bias and the other produces the switchover. At the cessation of the selecting currents the d-c bias automatically restores the selected core. This feature is important since the two polarities of drive current always required in magnetic circuits are obtained without having to double the number of driving electronic circuits.

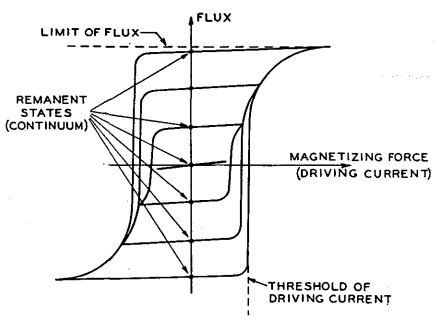


Fig. 1. Typical character of rectangular hysteresis loop cores

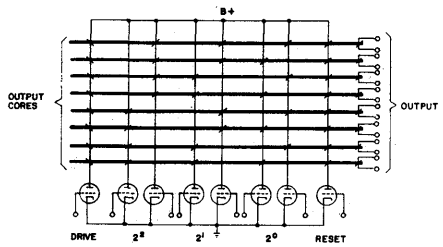


Fig. 2. Tube-driven decoder switch

Another example is what might be called a combinatorial decoder switch.[2] A typical case would be a switch with eight outputs and three inputs, whose purpose it is to select one output for every possible combination of inputs (Fig. 2). For simplicity, the cores are represented on the figure by heavy lines and the linking windings by 45 degree segments of line. The input signals are in pairs, each one linking half the cores. The first input links the cores by juxtaposed halves, the second by interlaced quarters and the third by interlaced eighths. The direction of winding is such that a current sent in one or the other branch of each input tends to magnetize the cores further into saturation. Consequently, the applications of the inputs have no effect per se. If during the presence of the inputs, all cores are energized in a direction tending to reverse their magnetization, only the core which is not inhibited by the input currents will have a net reversing magnetomotive force and therefore will be the only one to switch over. After the core has been switched, it can be restored by energizing a winding which links all cores. Decoding switches of this type have been used

JAN A. RAJCHMAN is with Radio Corporation of America, Princeton, N. J.

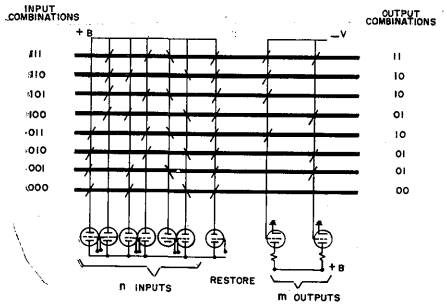


Fig. 3. Decoder-encoder or universal switch

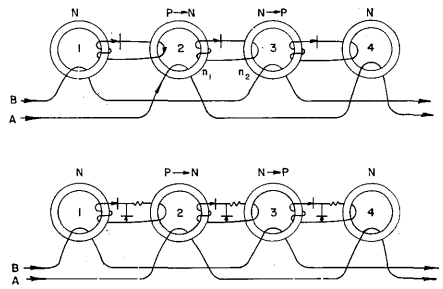


Fig. 4. Magnetic shift register

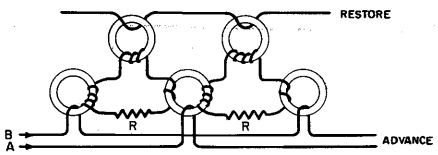


Fig. 5. Diodeless shift register using resistance coupling

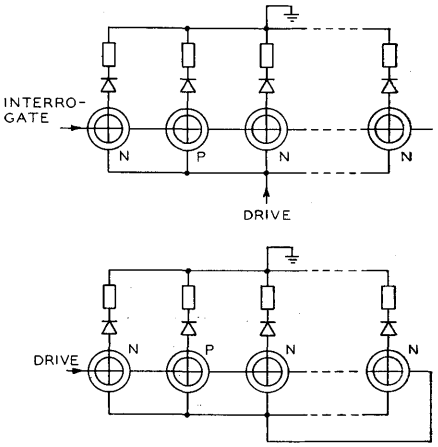


Fig. 6. Current steering principle

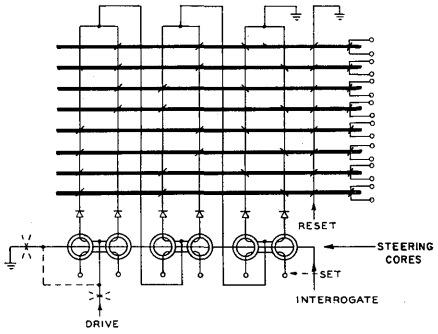


Fig. 7. Current-steering decoder

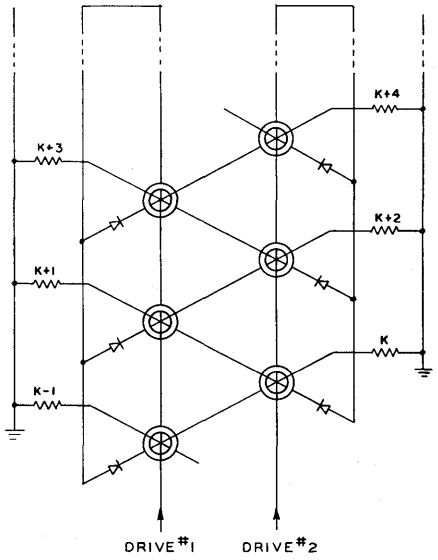


Fig. 8. Current-steering commutator

as input switches for magnetic core memories and many other applications.

The usefulness of the binary decoder switch can be broadened by using output windings that link, in series, certain cores according to a desired code (Fig. 3). Switchover of the selected core will cause output signals on those particular output windings which are coupled to it and those only. Any truth table relating any n inputs to any m outputs can be obtained in this fashion, demonstrating thereby the feasibility of obtaining any desired logical function. The system is practical for a relatively small number of inputs. However, it becomes impractical when the number of inputs is large, since the number of cores required, which is equal to the number of possible input combinations, grows exponentially (unless some input combinations are excluded). For example, for 20 inputs more than a million cores would be required. Also the power wasted in elastically disturbing many cores by many steps becomes prohibitively large.

Practical logical circuits cannot be treated merely as truth tables. Of necessity one must consider the specific logic to be accomplished. Furthermore, all the inputs may not be coincident in time. The logic may be a time schedule transformation. This requires, in general, that the logical switching be split into blocks and that it be possible to use the output of one core for the input of another.

Magnetic Shift Register

The classical example in which the switching of one core causes the switching of another is the magnetic shift register.[4] The operation of this device in which the pattern of settings of a row of cores can be shifted along the row is well known (Fig. 4). A few comments may be of interest. The information-carrying quantity is the magnetic flux. Flux is transferred from core to core without any loss because it is possible to make up for the ohmic losses in the coupling circuits by an excess of turns on the winding on the input core with respect to those on the output core. There must be provision for forcing the information to flow in the desired direction. Also, the core to which flux is being transferred must not be loaded by the next core of the row, as this loading would prevent its proper setting. To prevent this back-flow of information and detrimental loading, diodes are used in the coupling circuits.

The example of the shift register shows that the addition of diodes to magnetic cores provides power gain and permits transfer of information from core to core indefinitely. The gain results from the fact that the switching of a core to one polarity produces no load current and requires, therefore, only the amount of energy necessary to make up the losses in the core, whereas the switching of the core to the other polarity does produce a load current and transfer of energy from the source to the load. The logical input "primes" the core and determines thereby that an output is produced during the "drive" of the core. An unprimed core produces no output during drive.

The decoupling of the loading core during priming can also be achieved by using a resistive coupling instead of a diode and operating by priming at a relatively slow rate so as to make the resultant induced voltage (and current) small and driving relatively fast so as to make the resultant voltage (and current) high. In this way, the resistance seems to be in the circuit during priming, but seems to disppear during driving. Coupling cores between the switching cores are necessary to realize this artifice[5,6] (Fig. 5.)

The solution to the problem of transferring flux from core to core in the magnetic shift register can be applied to more complicated networks of interconnected cores which can perform a great variety of switching functions. Many such circuits have been built.

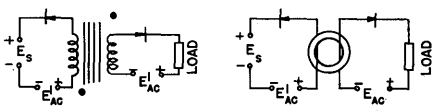


Fig. 9 (left). Principle of voltage-driven circuit

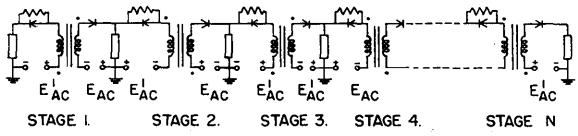


Fig. 10 (left). Voltage - driven commutator switch

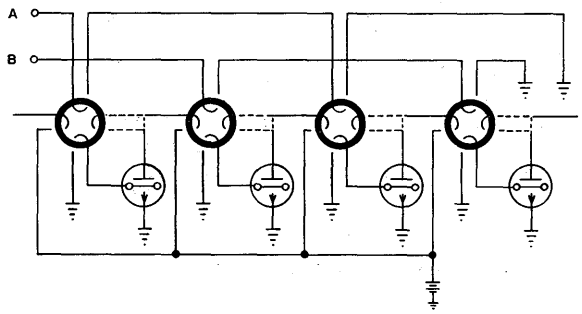
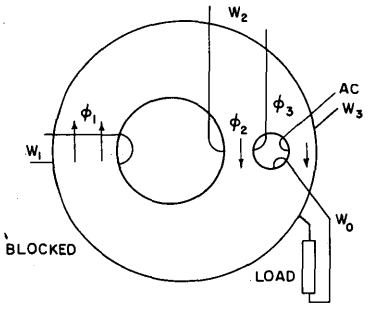
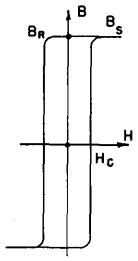


Fig. 11 (left). Transistor-coupled shift register





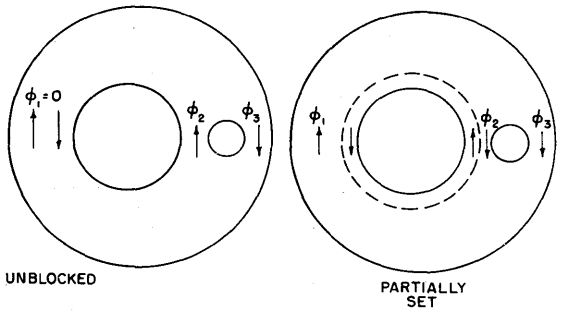


Fig. 12 (right). Principle of the transfluxor

Current Steering in Magnetic Circuits

Circuits[7,8] can be made in which the current from a source can be steered through one particular path among many possible paths. Consider the lower circuit of Fig. 6. Let all cores be set in one direction, the normal direction, except one which is set in the other, abnormal, direction. A pulse of current sent through the circuit and flowing through the common winding in a direction tending to bring all cores to the normal state, will switch over the abnormal core, inducing a voltage on its output winding which, with the proper choice of polarities, will tend to make the corresponding diode conducting and to cut-off all other diodes. Thus, the current will flow entirely through the selected branch.

For the steering to be successful, there must be more turns on the switching winding than there are on the branch windings (not shown on the simplified figure) in order that there be an excess of magnetomotive force-producing switch-over in spite of the selected branch current which necessarily opposes the effect of the driving current.

The prior setting of the core that selects the conductive branch produces no branch current in itself since the back-to-back diodes block any possible current flow. This setting is akin to priming the circuit. The desired branch current has an amplitude which is exactly the amplitude of the drive current source since the branch current is the drive current itself. Therefore it is independent of the core and diode properties.

The usefulness of the principle of current steering can be illustrated by its use in a binary decoder switch (Fig. 7). The three pairs of selecting windings that were driven by tubes in the previous example are now driven by current steering. Each tube is replaced by a core and diode. The parallel pairs of selecting windings are connected in series. The inputs to the switch set one core in each pair differently from the other. The drive brings all the input cores to a standard direction of magnetization, thereby switching one core in each pair. This causes the drive current to be steered through one branch of every pair. Consequently, all output cores but one will be subject to inhibiting currents and the selected one will be subject to a switching current, just as was the case in the tube-driven decoder. That particular core will be switched over and produce a signal in its output winding.

In the previously mentioned example, the outputs themselves do not have the benefit of current steering. The current steering is used merely for selection, but not for the outputs. However, it is quite possible to connect the series-connected parallel branches in series with the set of parallel output windings, each connected in series with a diode so that one can use current steering both for selection and for producing the output.[8]

Another example of a current steering circuit is one in which new settings are produced by the steered current itself. It is a "2-phase stepping register" which could also be thought of as a current steering commutator (Fig. 8). The purpose of the circuit is to deliver a given current successively to a number of loads. The steering cores are divided into two groups. A first drive current, steered by a given core, k of one group flows through a corresponding load and switches a core, $(k + 1)$ of the second group. In turn, a second drive current steered by the previously set core, $(k + 1)$ flows through its corresponding load and sets a new core, $(k + 2)$ in the first group and so on. The circuit is similar to the conventional magnetic shift register (used as a ring counter) with the important difference that the advance current itself is steered through the successive loads. It is possible to design the commutator so that most of the energy of the driving sources appears in the load and only a small part is dissipated in the steering cores and diodes. Experimental current-steered commutator switches have been operated with 80 per cent of the drive energy appearing in the loads. Currents of one to two amperes have been steered to successive loads, in time intervals of one to a few microseconds.[8]

The advance current can be modulated; i.e., the advance pulses need not have square tops nor need successive pulses be identical as long as their amplitude is sufficient for the advance action. However, to distribute modulated signals it is often convenient to separate the "drive" and "interrogate" functions. The cores are linked by another set of parallel branches which carry the loads in series with additional diodes. The additional

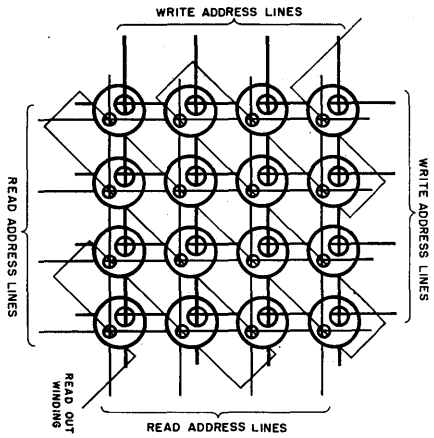


Fig. 13. Nondestructive memory. Crossbar switch

"drive" source of modulation is steered in succession through these branches due to the action of fixed amplitudes drive no. 1 and drive no. 2 which act as the "interrogate" and "set" for the circuit.[8]

Voltage-Driven Circuits

The combinatorial switches, magnetic shift registers, and commutator switches described thus far were all driven from current sources. A current source is a device furnishing a given current without regard to the voltage this current will produce in the load; hence it can be considered as a generator with infinite internal impedance. The current-steering principle epitomizes the current-driven type of circuit. Magnetic switches can also be driven by voltage sources; that is, by generators which produce a given voltage without regard to the current produced in the loads and which can therefore be considered to have zero internal impedance.

Some of the well-known voltage driven magnetic computer circuits depend on the use of the core as a synchronous magnetic amplifier.[9] Consider the core of Fig. 9 with an input and an output winding and with diodes in each circuit as shown. Power is provided to both windings; i.e., a-c sine waves of the same frequency and phase. Square waves or any other symmetrical waves can be used also. The two voltages E_{ac} and E_{ac}' have at any instant the relative polarities shown and have amplitudes proportional to the number of turns of their respective windings. The operation of the amplifier is in two steps: reset and gating. During the half-cycle with polarities as indicated in the figure, current can flow only in the input or reset circuit because of the direction of the rectifiers. This causes the magnetic core to proceed from saturation (or reset state) an amount dependent on E_s and E_{ac}'. If E_s is zero, the core is switched over by E_{ac}', if it is equal to E_{ac}' there will be no voltage across the cores since E_s and E_{ac}' will cancel so that the core will not switch over. When E_s is zero and the core switches over, there is no current in the output winding even though the direction of the diode would permit it because the voltage E_{ac}, which is greater or at least equal to the voltage induced on the output winding, keeps the diode cut off.

The second or gating step is in the next half cycle. The voltages are the reverse of those shown in the figure. If the core was not switched over in the preceding step ($E_s = E_{ac}'$), then the voltage E_{ac}, which tends to reset the core, appears almost entirely on the load while the load current simply brings the core further into saturation. If the core was switched over in the preceding step ($E_s = 0$) then the voltage E_{ac} will appear almost entirely across the output winding with only a small voltage drop across the load, due to the magnetizing current. It is seen, therefore, that if there is no input ($E_s = 0$) there is practically no output ($E_L = 0$), and if there is input ($E_s = E_{ac}'$) there is an output ($E_L = E_{ac}$). In the conversion of the intelligence signal from the input to the output several quantities can change. 1. The voltage E_{ac} can be greater than E_{ac}', 2. there can be more power in the output than the input because the input needs to provide only enough power to magnetize the core while the output can provide far in excess of that amount, and 3. the output occurs one-half cycle after the input.

It is of interest to comment on these three points. 1. The voltage E_{ac} can be greater than E_{ac}' simply because the core acts as a transformer. If the output of one stage is to drive another, it is advantageous to make E_{ac} greater than E_{ac}' in order to have some excess voltage to allow voltage losses in the coupling loop. 2. There is more power in the output than the input because an artifice is used to remove the load while the input

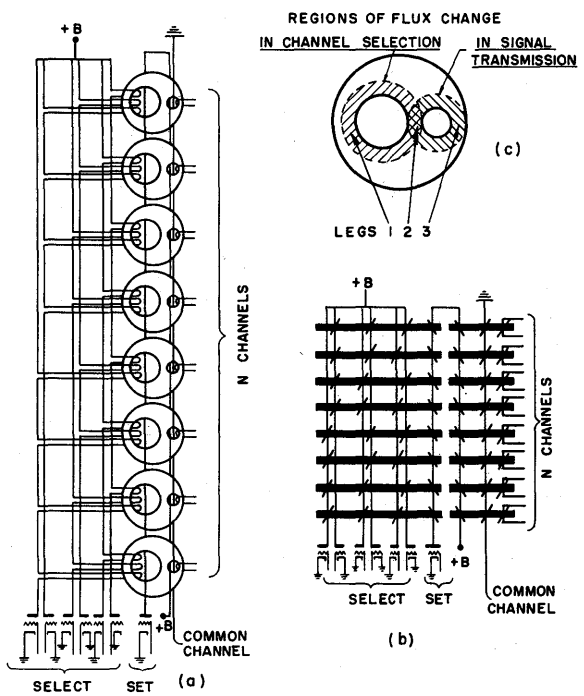


Fig. 14 (left). Coded channel selector

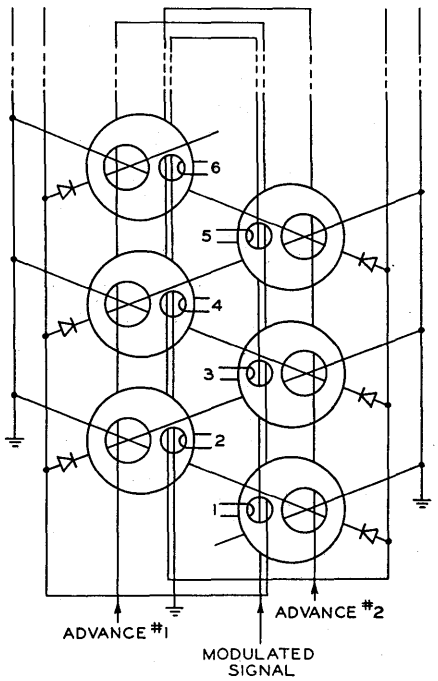


Fig. 15 (right). Channel commutator

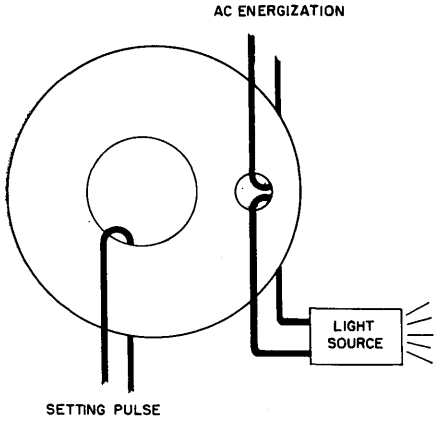


Fig. 16. Transfluxor-controlled light source

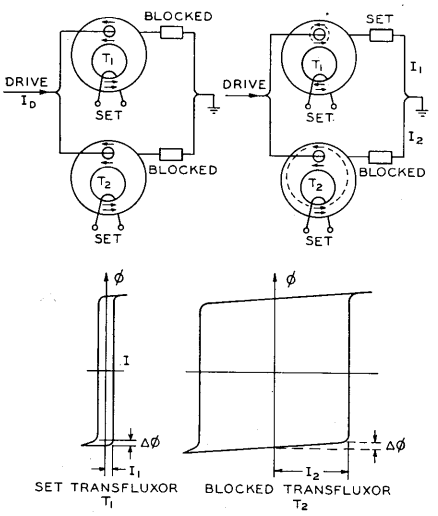


Fig. 17. Principle of transfluxor current steering

switches the core and to reinsert it in the circuit when the output switches the core back. The artifice here consists of biasing off the output diode, connected to be otherwise conducting. In current driven circuits discussed previously the artifice consisted in connecting the diodes so as to connect or disconnect the load, depending on whether it was the gating or control cycle. The input can be thought of as priming the core so that output can be obtained during a subsequent drive period. 3. In magnetic amplifiers there is always a delay between the output and input signals, of the order of a few cycles of the high-frequency power supply, which is required to allow for buildup. Here the delay is only half a cycle because the signal is synchronous with the power supply frequency. The delay, undesirable in conventional magnetic amplifiers, is exploited to advantage in computer applications.

A commutator switch, or a magnetic delay line as originally called, can be made by connecting many such one-core

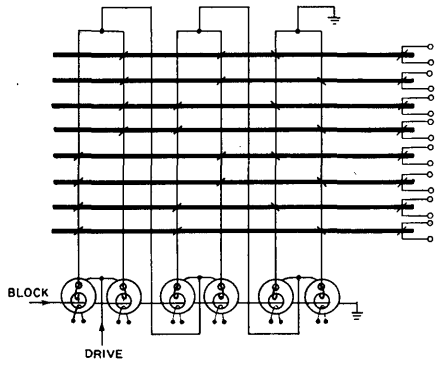


Fig. 18. Transfluxor-steered decoder switch

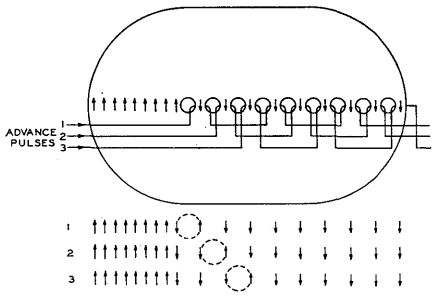


Fig. 19. Magnetic shift register by geometrical flux transfer

magnetic amplifiers in a row, as shown in Fig. 10. Here explicit use is made of the discrete time delay in each stage as well as of the intrinsic gain which prevents the degradation of the signal from stage to stage. The operation of this counter can be described in the following manner. A pulse appearing during a reset half cycle on the input of stage no. 1 is reproduced on the output of stage no. 1 on the second half cycle; this provides an input signal pulse for stage no. 2 and so the pulse is reproduced on the output of stage no. 2 on the third half cycle, etc. When a core is being switched over, the preceding and following stages must be decoupled, as was explained in connection with current driven commutators. The decoupling is obtained by forcefully biasing off the output diode when the input is allowed to conduct and vice versa.

Voltage-driven circuits have been investigated quite thoroughly and all-magnetic computers have been built[10] using unit blocks each including basically voltage-driven single-core magnetic amplifiers and a number of diodes. Some of the diodes provide the necessary elements required for obtaining gain and isolation from other units, while others are used for logic switching.

Transistor-Coupled Circuits

In the magnetic logic circuits previously described, the power originates

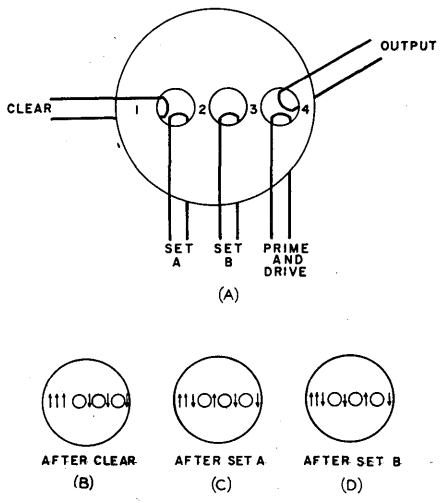


Fig. 20. Three-apertured transfluxor sequential gate

from a central source, and all elements of the circuit are passive and dissipate power. They require only very few driving tubes by centralizing all sources of gain and power and, therefore, minimizing the expense and possible trouble. This is in keeping with the tube-age pretransistor philosophy, that the complexity of a circuit is measured by the number of tubes, the tubes being the most expensive and the least reliable element of the circuit. With the advent of the transistor this may no longer be valid, as it becomes practical to contemplate many local elements with real gain interspersed between passive elements.

An example of such a circuit is a magnetic shift register with transistor coupling between cores as shown in Fig. 11. Actuation of one of the advance windings, e.g., A, causes the abnormal core (or cores) to switch from P to N, as in the previous example of a core-diode shift register. This induces a voltage on the winding connected to the base of the corresponding interstage transistor and thereby makes the transistor conducting. The relatively large collector current of the transistor switches over the next core from N to P, thereby effectively causing a shift of information. It is possible to improve the operation by a feedback coupling winding (shown dotted on the figure), on the core being reversed by the advance winding, by means of which the collector current of the transistor not only tends to switch over the next core, but also helps in turning over the core being actuated by the advance winding. This produces a "snap-action" and results in a greatly reduced value of required advance current, which now needs only to trigger the core.

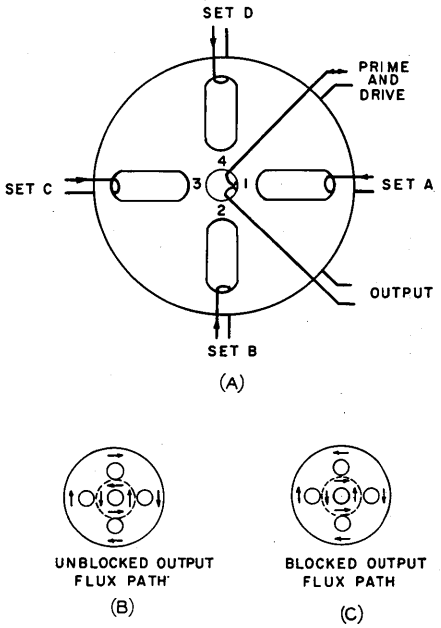


Fig. 21. Five-apertured flower transfluxor

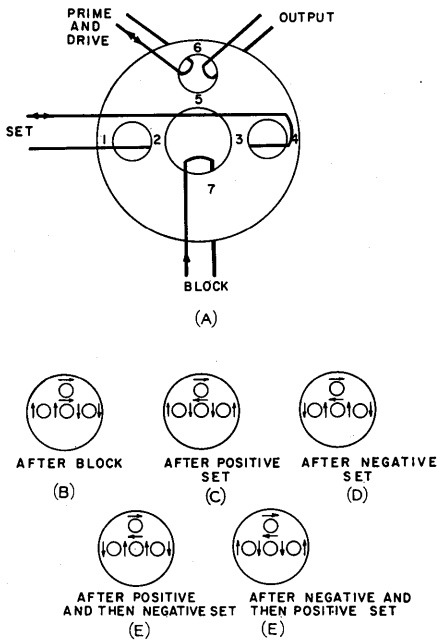


Fig. 22. Setting with a pulse of either polarity

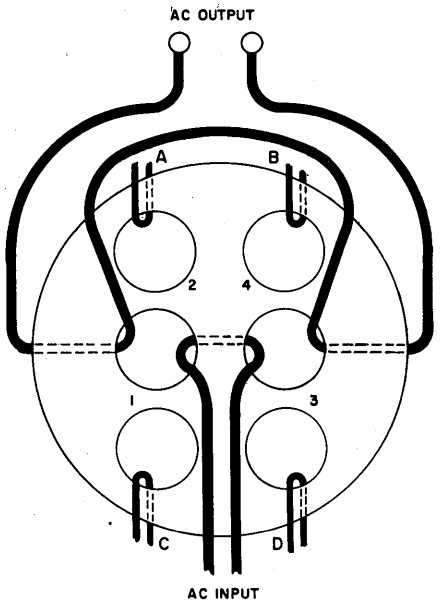


Fig. 23. Odd parity checker 6-hole transfluxor

Transfluxor

Magnetic flux, the information-carrying quantity, can be manipulated by geometrical transfers in branches of cores having a more complicated configuration rather than by electrical linkages between simple ring-shaped cores. This is an underlying idea of the transfluxor.[11,12]

In a two-apertured transfluxor having three legs, the amount of remanent flux in leg. no. 1, which is equal to the negative algebraic sum of fluxes in legs no. 2 and 3, determines how much transfer of flux between legs no. 2 and 3 is possible (Fig. 12). There can be an indefinitely long back-and-forth exchange of whatever amount of permissible flux was initially set in. The transfluxor is blocked when legs 2 and 3 are saturated in the same direction as one or the other leg denies any of the flux flow. The amount of setting depends on the amount of flux that can be changed in one leg before it is saturated. That same amount will necessarily appear in the other leg. The device can be used to store analog as well as digital information.

The nondestructive read-out property of the transfluxor permits making magnetic switches for selecting channels of transmission through which information can be transmitted for as long a time as desired. A class of such switches or storage systems is obtained by substituting a transfluxor for a core in the combinatorial core switches and memories described previously. As a result, a channel for bilateral transmission of signals is obtained in the selected element instead of a single transient at the instant of selection. Examples illustrating this possibility are:

1. Nondestructive read-out random-access memory.

An array of transfluxors can be linked with column and row windings, both through the large input aperture and through the small output aperture (Fig. 13). Current-coincidence setting of the transfluxors is possible because there is a threshold of current required to produce setting and, similarly, current-coincident read-out can be obtained because there is a threshold of current required to obtain read-out. This type of memory is useful when frequent and rapid read-out is necessary and relatively slow write-in can be tolerated.[12]

2. Cross-bar switch.

An array of transfluxors can also be utilized for a cross-bar switch (Fig. 13). Pulse or amplitude modulated signals applied to the column windings linking leg no. 3 will be transmitted to those row windings linking leg no. 3 which are coupled through unblocked transfluxors, but not those coupled through a blocked transfluxor. The setting of the transfluxors can be obtained by using the column and row windings linking leg no. 1.[12]

3. Coded channel selector

A decoder switch analogous to the core type can be made by using transfluxors. The large aperture is treated as if it were a simple core and is threaded by the input selecting setting windings (Fig. 14). As a result, a selected transfluxor can be set and all others blocked. Therefore, a modulated signal fed through the small hole of all transfluxors will be transmitted only to the one that is set.[12]

4. Channel commutator.

A device for opening channels successively from an array can be made by using a row of transfluxors. Here again the large aperture is considered as though it were a simple core and is connected as the cores of a shift register (Fig. 15). Current-steering type of connections are particularly suitable. After the termination of an advance pulse, one of the transfluxors is set while all others are blocked. Consequently, modulated signals can be transmitted through that channel only and will continue to be so transmitted until a new advance pulse selects the next channel.[13]

5. Control of light sources.

The transfluxor is particularly adaptable to the control of light sources, such as incandescent lamps, gas discharge lamps or electroluminescent cells. Let the light source be coupled to the output of a constantly energized transfluxor (Fig. 16). Light will be emitted in accordance with the amplitude of the setting, which is zero when the transfluxor is blocked and which can have any desired intermediate value up to a maximum. Possible applications include indicators to monitor the occurrence of pulses in digital systems, small arrays of lights to display numerals or letters, or large arrays to display more complex patterns and pictures. (These applications will be described in detail in a forthcoming publication.)

Transfluxor Current Steering

The previous examples show the usefulness of the nondestructive read-out property of the transfluxor. The isolation of the input setting and the output circuit of the device can be exploited for current steering.[8]

Transfluxor steering is achieved by the use of a transfluxor winding linking leg no. 3 connected in series with a branch load in each of a number of parallel branches. This is illustrated in Fig. 17 for the case of two branches. The steer-

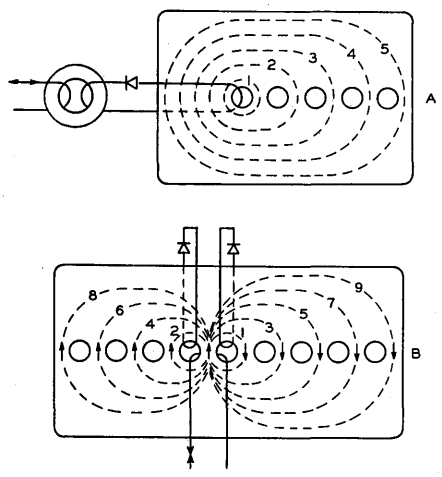


Fig. 24 (left). Flux counters

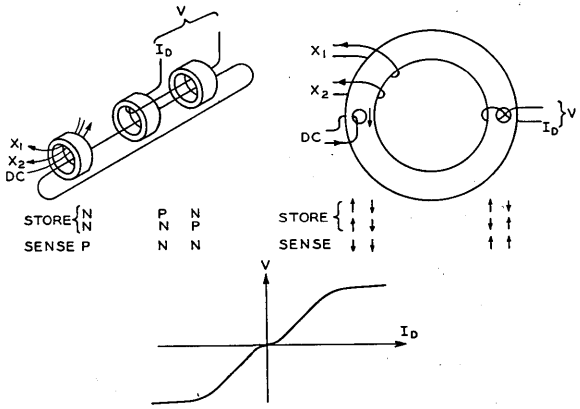


Fig. 25 (right). Flux limited drive of storing cell

ing action can be considered to be due to the difference in effective impedances of a blocked and set transfluxor. By setting the two transfluxors to different states, a preferential current flow is obtained. However, diodes are not required since the setting of the transfluxors causes no voltages in the output windings. The operation of the transfluxor steerer can be understood by following the detail of current drives and resulting flux changes. When the two transfluxors are blocked, the drive current will not be steered, but rather will be divided between the two branches, equally if the loads are equal. If one transfluxor is set; e.g. T_1 in Fig. 17, the drive current I_D divides into a relatively small current I_1 and a larger current I_2. This can be understood as follows: Assume first that both branch loads are zero; then the transfluxor output windings are directly in parallel and any voltages generated by the changing of flux in transfluxors T_1 and T_2 must be equal. Consequently, the total flux changed in each branch at every instant must also be equal as they occur at the same time. Since T_1 and T_2 operate on different B-H loops, transfluxor T_1 being set and T_2 blocked, it takes different magnitudes of current to produce equal flux changes. For reasonably square-loop magnetic material the blocked transfluxor T_2 requires a large current I_2 to produce the same flux change that a smaller current I_1 produced in the set transfluxor T_1. The ratio of currents I_2 and I_1, defined as the discrimination, in a real circuit with equal loads will start at a high value and tend asymptotically to unity as the currents I_1 and I_2 become equal. This results from the fact that eventually there are no longer any flux changes and the currents divide according to the load impedances. The discrimination is low for large loads since the voltages across the transfluxor windings are a small part of the total branch voltages. Consequently, the loads must be reasonably small to obtain good discrimination.

In the operation of the transfluxor steerer, branch currents are undesirable during "set and block," since they slow down these steps and consume power. As explained previously, during "set," there are substantially no branch currents. During "block," branch currents are zero because the fluxes in leg no. 3 of both transfluxors are reversed simultaneously with resultant opposition of induced voltages. The drive current is intense enough to produce spurious unblocking of the initially blocked transfluxor and thereby bring it to the same state as that of the initially set transfluxor which has been driven. The drive, being intense, provides fast flux switching. This results in an inherently fast overall operation, since the "block and set" steps, producing only negligible branch currents, can be performed arbitrarily fast.

Transfluxor steering of current through one branch of a parallel pair can be repeated in many pairs connected in series and is directly applicable to driving a decoding switch. The example of a decoder with $n = 3$ inputs and $2^n = 8$ outputs used to illustrate the tube and core diode decoders is used again in Fig. 18 to illustrate the transfluxor steering decoder. After one transfluxor in each pair is set by the inputs, the drive current is steered through the branch of each pair including the blocked transfluxor and thereby causes a net reversing magnetmotive force on only the single selected output core.

Multiapertured Transfluxors

The principle of the geometrical flux transfer can be generalized to cores with more than two apertures and three legs.[11,12,14,15] Two simple principles are sufficient for the understanding of most effects in these more complicated transfluxors. These are:

1. The total amount of flux through any plane section of a core must be zero to satisfy the condition of continuity of flux flow. Consequently, the transfer of flux from any branch of the core to other branches can be considered as lossless. The situation is analogous to the hydrostatics of an incompressible fluid, as the total amount of fluid remains constant no matter how the distribution is changed.

2. The path followed in any flux change for a given magnetomotive excitation will tend to be as short as possible. If several alternate paths are offered to the flux flow, the flux will take the shortest path unless it includes a saturated leg in the direction of the prospective flow. In that case it will take the next shortest path.

These principles are illustrated in a possible type of shift register consisting of a row of holes (Fig. 19).[13] Assume that initially the flux is directed downward in all the legs and upwards in an adequately wide return path leg. The device can be thought of as operating as follows. Energizing advance winding no. 1 will cause the first leg to be directed upward with a consequent downward partial change of flux in the wide leg. This advance current will have no effect in any but the first leg because no available unblocked paths exist except the one including the first leg. The application of the second advanced winding will cause the flux in the second leg to go upward and to restore the first leg to its initial downward condition. The application of the third advance winding will similarly transfer the upward state to the next leg. The successive applications of advance pulses 1,2,3,1,2,3, etc. will cause the upward state to be shifted along the row of legs. Three, rather than two, advance windings are necessary to in-

sure direction of information flow. Experiments show that such successive transfers of flux are possible. However, because of deviations from ideal loop rectangularity, at every step some flux is transferred to more distant legs. These partial flux transfers diminish the value of the nominal flux and also cause secondary parasitic transfers. Nevertheless, reasonably long geometrical shift registers of this type can be made.

The geometrical flux manipulations permit designing various specialized gates, as illustrated by the following examples:

A transfluxor with three apertures in a row, as shown in Fig. 20, can be operated as a 2-input sequential gate.[12] An output is produced if the two inputs A and B are applied to it in the order AB and no output is produced if either input is missing or if the two inputs are applied in the order BA. The operation is illustrated by the symbolic diagrams, Figs. 20(B), 20(C) and 20(D).

The transfluxor with five apertures arranged like a flower, Fig. 21, can be operated as a 4-input "and" gate.[12] The occurrence, in any order, of all four input signals A, B, C, and D is required to open the gate. The principle of operation depends upon the fact that the output flux via legs no. 1, 2, 3, and 4 around central aperture can be blocked by any one of the four legs and is unblocked only when the senses of flux saturation around central hole in all legs are the same.

A transfluxor which can be set by either polarity or setting pulse,[12] can be obtained by using four apertures, as shown in Fig. 22. It is apparent that either a positive or a negative set pulse will cause leg no. 5 to reverse its flux. For a positive set pulse this will occur with corresponding reversals of legs no. 2 and 4, and for a negative set pulse with reversals of legs no. 1 and 3. The output flux path via legs no. 5 and 6 is unblocked by the setting of leg no. 5. Blocking can also be of either polarity.

Another example of a multiple-hole transfluxor is a 6-hole "logicore" which can be used for a 4-terminal odd-parity checker.[14] The a-c input produces an electromotive force tending to change flux on a path around legs no. 1 and 2 and on a path around legs no. 3 and 4 (Fig 23). Depending on the input signals A,B,C,D, determining the flux around the outer holes, these two paths can be set or blocked. There will be no a-c output when both paths are blocked, or when both are unblocked, because flux changes on the two sides buck each other. This occurs either when no or any two inputs A,B,C,D have been energized. There is an a-c output when one path is blocked and the other unblocked, a case which occurs when one or three of the inputs A,B,C,D are energized. The 6-hole transfluxor can also be used for the exclusive "or" logic function and for a half-adder.[14]

Combination of Geometrical and Electrical Flux Transfer

It is possible to use a combination of geometrical flux transfer in a multiapertured core with further transfers by electrical linkages to other cores. As an example, consider what might be called a flux counter.[13] The middle aperture of a row of apertures is linked to a simple toroidal core through an electrical linkage including a diode (Fig. 24). The cross-sectional area of the toroidal core is made equal to that of the legs of the multiapertured core. Back-and-forth reversals of the toroidal core will cause the legs of the multiapertured core to be switched over in succession. This occurs because the reversal of the core in the direction in which the diode is conductive will cause a flux change through the nearest available leg in the multiapertured core. Reversal of the core in the opposite direction will not erase this setting because the diode blocks any current flow.

A similar flux counter can be made by using one of the legs of the multiapertured core as the standard unit of flux to be added per count (Fig. 24). Two electrical linkages with diodes are used on either side of the leg. One or the other of these linkages becomes conductive when the center leg is reversed and produces a current which tends to force the flux change in the direction of the other nonconductive linkage. Consequently, for successive reversals of the central leg the flux will be pushed to the left, to the right, to the left, etc.

Analogy Between Geometrical and Electrical Flux Transfer

Consider three identical cores linked by an electrical loop of negligible ohmic resistance (Fig. 25). One of these three cores will be considered as a switch core and the two others as a pair of memory cores. Assume that initially the switch core is in state N and the memory cores are either in the states NP or PN. The switchover of the switch core from N to P will bring whichever memory core was in state P to state N. This occurs because the total amount of flux linked by a zero-resistance loop must remain constant. The identity of the memory core being switched can be determined by the polarity of the induced voltage on a winding (called a digit winding) linking the two memory cores in series opposition. When the switch core returns to state N, it will tend to switch both memory cores to state P. If these are identical and there is no current through the digit winding, both cores will be switched halfway. A subsequent read-out, due to the switching of the switch core from N to P, would switch the two memory cores equally, and consequently no read-out voltage would be obtained. However, if during the return of the switch core from P to N, a current is sent through the digit winding which favors one core and hinders the other, the flux switched in one core will be greater than that in the other. Consequently, on a subsequent read-out, a voltage is obtained when both memory cores are brought to the standard N condition. The readout voltage depends on the amplitude and polarity of the write-in current in accordance with a typical characteristic curve as shown in Fig. 25. It is worth noting that analog information as well as digital information can be stored in such a pair of memory cores.

Systems for driving memories utilizing this principle of flux-limited drive have been described.[16] The switch as well as the memory proper is made of ferrite-apertured plates. The switch drives many parallel pairs of plates, the address selection being by rows X_1 and columns X_2. A d-c biased switch is particularly convenient. Corresponding apertures in a pair of memory plates are used to store one bit of information. The three cores of Fig. 25 can be thought to represent the apertures and surrounding material of one address location of the driving switch and one pair of memory plates.

A 3-apertured transfluxor can be used as an analog of the three electrically linked cores.[13] This is illustrated in Fig 25. Here the d-c links leg no. 1, and the selecting currents X_1 and X_2 link the central aperture; i.e., both legs no. 1 and 2. When the sum of energizations of the windings X_1 and X_2 exceeds that of the d-c bias by an amount sufficient to produce switchover between leg no. 1 and legs no. 3 and 4, the latter will be brought to saturation. If they have been previously left in unequal states of remanence, this will produce an output voltage on the digit winding linking legs no. 3 and 4. When the selecting currents X_1 and X_2 terminate, the d-c tends to transfer the total flux of leg no. 1 through legs no. 3 and 4. Assuming for

a moment that the path lengths 1 to 3 and 1 to 4 are equal, this would produce equal amounts of flux (equal to half that contained in leg no. 1) through legs no. 3 and 4. If, during this time, a digit current is sent through the digit winding, unequal amounts of flux will be transferred, because that current will favor transfer of flux to one leg and hinder it to the other. Consequently, the relation between the read-out voltage and write-in current will be similar to what it was for the three cores. It is interesting to note that the total amount of flux through legs no. 1,2,3, and 4 must not only be constant in this case, but identically zero. For this reason it is necessary to have an additional leg, leg no. 2, which is really a dummy, and serves merely to make up for the balance of flux.

If the lengths of path 1 to 3 and path 1 to 4 are not equal, as is true in the case of the core having the geometry shown in Fig. 25, there will be unequal division of flux between legs no. 3 and 4 even when the digit current is zero. Various geometries are possible to make these flux paths equal. One geometry consists of properly shaping the center aperture in the shape of a "U." Another consists of locating legs no. 3 and 4 on either side of leg no. 1 and splitting dummy leg no. 2 in two halves.

Very fast current-coincident setting can be obtained with either the electrically linked three cores or the 3-aperature transfluxors. Furthermore, the storage is differential and thereby gives a positive or a negative read-out signal for digital information rather than a signal or lack of signal as in conventional systems. Very fast access and read-out signals free of disturbances characterizes the system.

Setting of Transfluxor by Short Pulses

In most of the devices and circuits described so far, it was assumed that the energizing current pulses were of sufficient duration to produce as complete a switch-over as that due to pulses of much longer duration. It is interesting to consider the behavior of a transfluxor being set by much shorter pulses and the solutions to some of the problems encountered in such a case.[13]

Consider a 3-hole transfluxor as illustrated in Fig. 26. Let the setting windings be on leg no. 1. The total amount of flux that can be transferred to legs no. 3 and 4 and that can be subsequently interchanged between leg no. 3 and 4 is limited to the saturated value of flux in leg no. 1. Consequently the setting characteristic, or the amount of set flux as a function of the amplitude of a single pulse, has a saturated region as shown in Fig. 26. It is found by experiment that this characteristic is the same for a pulse of any duration, as long as that duration is greater than some minimum T_0. This time, T_0, might be defined as the nominal setting time of the transfluxor and is approximately equal to the switching time defined for optimum operation in a 2-to-1 regime for a memory toroid made of the same material. This time, T_0, depends on the nature of the material, and is typically between 1/2 and 5 microseconds for good square-loop ferrites.

It is found that the characteristic is quite different when the setting pulses are shorter than the nominal time T_0. For a given amplitude of setting pulse, much less flux is set than for the nominal-duration pulses. A typical curve for short setting pulses is also shown in Fig. 26. If many short setting pulses of a given amplitude are used, it is found that the setting will reach the same value as can be obtained with a single longer pulse of like amplitude. In many applications where it is desired to sample the instantaneous value of an analog quantity and store it for further manipulation, these effects are undesirable, for example, when it is desired to set a transfluxor by the coincidence of two or more pulses applied to leg no. 1.

Rapid coincident setting of a transfluxor is possible in spite of its relatively slow setting characteristic. The method is as follows. A d-c current is applied to leg no. 1 of the transfluxor in addition to the setting current-coincident pulses. Consider the case of two such pulses, one of the pulses being purely for selection and the other carrying the analog information to be stored. Let the amplitude of the first pulse and the maximum amplitude of the second be equal to the d-c bias. It is evident that the application of both pulses will set the transfluxor, whereas, the application of either one separately will have no effect on the setting. The desired level can be obtained in the desired time, provided the amplitude is chosen to be sufficiently large. It is interesting to note that after the cessation of the setting pulses the d-c bias will restore leg no. 1 to full saturation, but this will not erase the setting, as the flux flow will be to the closer leg no. 2. The magnetic circuit through legs no. 1 and 2 "protects" the setting of the flux in the output legs no. 3 and 4.

Three-hole cores operated as described above can be arrayed in rows and columns to provide high-speed memories for analog or digital information with non-destructive read-out properties.[13] It is also possible to operate 3-hole cores at the higher speeds made possible by a d-c bias with conventional destructive read-out.[17]

Conclusions

The examples of devices and circuits described in this paper illustrate the variety of switching tasks which can be accomplished by magnetics. Many of these tasks can be performed by other devices, such as transistors and diodes, vacuum tubes, cryotrons and other cryogenic devices, and by devices based on such physical phenomena as ferroelectricity, photoconductivity, electroluminescence, magnetoresistance, etc.

It is perhaps most significant to compare the art of magnetics to the art of semiconductor devices, as both are relatively well developed and both utilize solid-state devices at noncontrolled ambient temperatures. In general, storage is better accomplished with magnetics since it is an inherent property of the materials used and does not depend on a flip-flop or other artifice required with semiconductor devices. On the other hand, switching is better accomplished, in general, by semiconductor devices which have inherent gain and greater nonlinearity. Switching by magnetics requires more power than by semiconductors, and the driving power must be in the form of a-c or pulses, i.e., generated by active elements such as tubes or transistors. In spite of these disadvantages, magnetic switching is perferable in many cases including the following:

1. When the number of switched channels is very large. It is quite feasible to make magnetic switches with tens or even hundreds of thousands of outputs, whereas switches of similar size made with present-day semiconductors would be prohibitively expensive and not as reliable. Magnetic switches to address magnetic memories are a typical example of this case.
2. When the desired switching function is intimately related to a storing function. This is the case in most of the described applications of the transfluxor as well as in serial logic switching functions exemplified by the magnetic shift-register.
3. When extreme reliability is desired.

Magnetics, based on the existence of the electron spin, is one of the fundamental electronic tools for the manipulation of information along with the controlling of electronic charge motion in vacuum, semiconductors, and superconductors. It is naturally suited to

memories and is most likely to remain, the dominant technique in this application. Magnetic switching, as a corollary to magnetic storage and in many other applications, is also likely to remain a dominant art in the foreseeable future.

References

1. Digital Information Storage in Three Dimensions Using Magnetic Cores, **J. W. Forrester.** *Journal of Applied Physics*, New York, N. Y., vol. 22, Jan. 1951, pp. 44–48.

2. Static Magnetic Matrix Memory and Switching Circuits, **J. A. Rajchman.** *RCA Review*, Princeton, N. J., vol. 13, June 1952.

3. A Myriabit Magnetic Core Matrix Memory, **J. A. Rajchman.** *Proceedings*, Institute of Radio Engineers, New York, N. Y., vol. 41, Oct. 1953, pp. 1407–21.

4. Static Magnetic Storage and Delay Line, **A. Wang, W. L. Woo.** *Journal of Applied Physics*, vol. 21, Jan. 1950, pp. 49–54.

5. A Magnetic Core Gate and its Application in a Stepping Register, **G. R. Briggs.** *Report E-475*, Digital Computer Engineering, Massachusetts Institute of Technology, Cambridge, Mass., Oct. 1952.

6. Diodeless Magnetic Core Logical Circuits, **L. B. Russell. 1957** *Convention Record*, Institute of Radio Engineers, pp. 106–14.

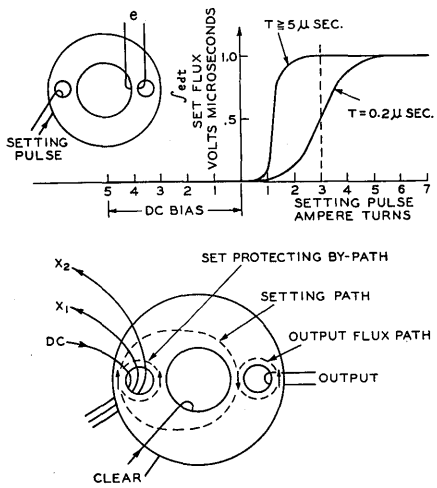


Fig. 26. Setting of transfluxor by short pulses

7. Pulse-Switching Circuits Using Magnetic Cores, **J. Karnaugh.** *Proceedings*, Institute of Radio Engineers, vol. 43, May 1955, pp. 570–84.

8. Current Steering in Magnetic Circuits, **J. A. Rajchman, H. D. Crane.** *Transactions*, Institute of Radio Engineers, vol. EC6, No. 1, Mar. 1957, pp. 21–30.

9. Magnetic Amplifier Circuits and Applications, **R. A. Ramey.** *Electrical Engineering*, New York, N. Y., vol. 72, Sept. 1953, pp. 791–95.

10. The Megacycle Ferrator, **T. H. Bonn.** *Special Publication T-85*, 1956 Western Joint Computer Conference, AIEE, Feb. 1956, pp. 118–19.

11. The Transfluxor—A Magnetic Gate with Stored Variable Setting, **J. A. Rajchman, A. W. Lo.** *RCA Review*, vol. 16, June 1955, pp. 303–11.

12. The Transfluxor, **J. A. Rajchman, A. W. Lo.** *Proceedings*, Institute of Radio Engineers, vol. 44, Mar. 1956, pp. 321–32.

13. Principles of Transfluxor and Core Circuits, **J. A. Rajchman.** *International Symposium on Switching Theory*, Massachusetts Institute of Technology, Cambridge, Mass., Apr. 1956.

14. Multihole Ferrite Core Configurations and Applications, **H. W. Abbott, J. J. Suran.** *Proceedings*, Institute of Radio Engineers, vol. 45, Aug. 1957, p. 1081.

15. Logic by Ordered Flux Changes in Multipath Cores, **N. F. Lockhart. 1958** *Convention Record*, Institute of Radio Engineers, Mar. 27, 1958.

16. Ferrite Apertured Plate for Random Access Memory, **J. A. Rajchman.** *Proceedings*, Institute of Radio Engineers, vol. 45, Mar. 1957, pp. 325–34.

17. Recent Events in Very High Speed Magnetic Storage Techniques, **W. W. Lawrence, Jr.** *Special Publication T-92*, Proceedings Eastern Joint Computer Conference, AIEE, Dec. 10–12 1956, p. 101.

Discussion

Chairman Meagher: The first question is from R. K. Richards, Consulting Engineer, for Dr. McMahon: "Would you please describe the cylindrical film cryotron in more detail? How do you obtain current gains greater than unity with this structure?"

H. McMahon: The idea is to substitute for the solid gate wire, a thin metallic film in order to reduce just the cross section, thereby increasing the resistance. This can be done by evaporating the metal down to a glass rod, and then depositing a layer of insulation, and finally wrapping a control winding on that, just as on the normal cryotron. This does not alter the current gain at all. It does increase the resistivity a great deal.

Now there can be a slight variant on that, but to my knowledge this has not been carried out, that is, to replace the glass rod in the structure with a super-conducting diamagnetic material. In other words, one might have a niobium wire, the insulating film, and a tantalum gate evaporated on that in the form of a thin shell, and finally the control winding. Now this not only increases the resistance of the gate, but it very greatly decreases the inductance of the control winding.

Now the second variant greatly decreases the inductance of the control winding, and thereby decreases L and increases R simultaneously, both by very sizable factors. You increase the speed perhaps by the order of a hundred to a thousand.

Chairman Meagher: The second question that Mr. Richards put was, "How do you obtain current gains greater than unity with this structure?"

H. McMahon: This structure does not change the current gain feature at all.

Chairman Meagher: The next question is for Mr. Kuchinsky from Mr. Turner: "Will you explain the method used in determining the 50,000-hour end-of-life figure and is it safe to predict this for computer application?"

Saul Kuchinsky: The unusually long life of the Beam Switching Tube can be predicted on the basis of comparing its inherent design features with those of standard tubes and discussing the reasons that these latter tubes have earlier life failures. These features primarily are: 1. The Beam Switching Tube is a high vacuum tube, 2. Its large cathode-anode spacing eliminates contact potential variations so harmful in standard tubes with close spaced grids, 3. A magnetic field minimizes ion bombardment of the cathode, 4. The Beam Switching Tube cathode is operated under relatively low current and voltage requirements, 5. Because of its unique beam forming characteristics, involving crossed electric and magnetic fields, output currents remain constant despite wide variations in cathode efficiencies.

Life test data are continuing on a small quantity of tubes beyond 25,000 hours of life as of this date.

However, the average time between failures of any component, regardless of its inherent life, is not easily determined and may be primarily a factor of quality control. Each Beam Switching Tube receives an aging test of over 30,000,000 stepping operations (72 hours at 120 cycles per second) before being shipped to a customer. This detects the early failures and insures an unusually high percentage of long life tubes. In this respect it may be comparative to predict 50,000 hours for Beam Switching Tubes on the same basis that 10,000 hours is predicted on some premium computer tube types.

Chairman Meagher: "Mr. Kuchinsky, do you have any added comments about the devices you have described?"

Saul Kuchinsky: I believe there will be a more realistic trend to use special purpose tubes as well as regular tubes in applications where distinct advantages over solid state devices of cost, temperature, servicing, replacement, and reliability are apparent prime factors. For example, several major "all-transistor" systems have already made exceptions of using Beam Switching Tubes where one tube replaces 18 transistors and many diodes in a decimal counter with individual digital outputs.

Special purpose tubes will undoubtedly get smaller and better with time. Special purpose solid state devices will become more competitive with special purpose tube devices. However, the unique characteristics obtainable by special purpose tubes will apparently make them a factor well worth considering in the foreseeable future.

Chairman Meagher: The next question is for Dr. Ross from Paul Low, IBM: "Would you comment on some of the 'field-effect' devices?"

I. M. Ross: You are thinking of field-effect devices for switching applications. The main drawback here is the efficiency of the device as a switch. There is the possibility of getting bistable operations by getting injection from the drain contact which might be of interest in switching applications, but I would suspect here that p-n-p-n would be more important.

The main application of the field-effect might be in transmission-type applications, where you are interested in not-too-high frequency response with high input impedance and high output impedance. There is a possibility that they result in lower noise.

Chairman Meagher: The next question is for Dr. McMahon from Mr T. G. Lesher, Hughes Aircraft Corporation: "How great are the problems of maintaining a liquid helium bath in a large computer system?"

H. McMahon: At the present time there are serious difficulties. Concerning helium, I think it is entirely fair to say that if there is to be a serious advantage gained from superconductive circuitry, or any kind of circuitry requiring some special temperature stabilization, then I think that it is fair to say that the problem of maintaining this ambient is no more difficult than any other temperature you wish to select. It is purely a question of putting in enough man hours to do the job. There is nothing mystical or difficult about it. The insulation problem, that is the thermal insulation problem, is somewhat worse, but by and large I think that it should be viewed as almost an engineering triviality if there is anything to be gained by doing it.

Chairman Meagher: The next question from Mr. Loeh, Convair-Astronautics, is for Dr. Ross: "Previously, the state of the art was limited by the available components, tubes and relays. Now we have many components all serving similar functions. Has anyone selected or developed selection criteria? If not, what would be possible selection criteria?"

I. M. Ross: There are not good criteria yet for selecting devices for particular applications. I do not think that we have sufficiently explored the possibilities of the devices themselves, nor have we explored the circuit applications. However, I can make the comment that eventually this choice will be a matter of reliability and cost.

Chairman Meagher: I have another question for Dr. Ross from Mr. M. Kliman, Sperry Gyroscope Company: "Will you elaborate on your description of the solid-state 4-stage counter? Have any other devices of this nature been built?"

I. M. Ross: The device consists of a number of p-n-p-n stages fabricated in one sheet of semiconductor, there being one contact common to all stages plus individual contacts to each stage. The individual stages exhibit the bistable action of a p-n-p-n device and hence, by using a common load resistor, one can insure that only one of the stages is in the "on" condition. By controlling the geometry of the individual stage, one can arrange that current flowing in one stage will trigger only one of its two neighbors. The resulting device is similar in concept to the gas stepping tube yet is the size of a normal transistor. Exploratory models have operated at counting rates up to one million per second.

Chairman Meagher: "Have any other devices of this nature been built?"

I. M. Ross: One attempt I know of used as the basic stage the filamentary transistor or double based diode. A few devices were fabricated but development was not continued because the p-n-p-n structure gave promise of better performance.

Chairman Meagher: Perhaps Dr. Ross will be willing to state, in a few minutes time, the direction in which solid state semiconductors will go in the next few years?

I. M. Ross: On transistors and diodes, I believe there will be continued effort on producing faster devices possibly resulting in one order of magnitude of higher speed. However, the main effort will probably be in the direction of improving reliability and lowering cost. There has been talk today about cores and the necessity for transistors to drive these cores. To do this, the transistors must be capable of switching amperes into the cores in times around 10 to 100 millimicroseconds. I think that in the future we can expect to see devices that will meet these requirements.

The other question that arises is "what about new material?" In order to obtain a higher temperature of operation, we must have larger energy gaps. This is one of the reasons for silicon being employed rather than germanium for higher temperature applications. The III-V compound semiconductors promise higher energy gaps and

some of them also have high mobility which is important in determining the base thickness required for a given frequency cut-off. We may see the development of some types of transistors made from compound semiconductors.

Chairman Meagher: "Dr. McMahon, do you have any added comments?"

H. McMahon: I think perhaps that I might just hedge by saying, what are the possible advantages of the cryotron circuit? What are their obvious limitations? Of the possible advantages, it seems to me the first is reliability, and this is not through any observational experience because they are not well enough along to say anything more about that. At very low temperatures there is nothing that can happen except a mechanical disruption or electrical burnout and mechanical destruction, and these are the only two things that could interfere with reliability. The cost looks particularly low when one looks at it superficially, and that is the only way one can look at it right now as the final device has yet to be invented.

There seem to be no difficult refining problems of any sort; and there do not seem to be difficult control problems concerning materials. It lends itself to large size simply because of the possibility of miniaturization, and the possibility of constructing the computer as a parallel kind of operation, rather than a sequential type where you deal with separate devices.

If superconducting circuits are ever used, it will be a different philosophy from the old one of building devices and hooking them up one by one. It will be more a matter of constructing large and complex circuits in one vacuum evaporation using suitable masks. It may be that other components will go in the same direction, we are working with blocks of components instead of single ones.

Perhaps I should mention the limitations: Obviously one of the limitations is requirement of low-temperature ambients. I do not regard this as a terribly difficult problem, but, generally speaking, it can be serious. Another limitation is, that because of the cost of this region of very-low temperature, one does not want to bring in large numbers of electrical communications. This implies that things like telephone switchboards might be very cumbersome, although otherwise it lends itself to the application.

I think that the limitation of the present devices is that, unfortunately, they do not combine too well with existing input and output circuits.

Chairman Meagher: "Dr. Rajchman, do you have any comments about the use of cores, or other magnetic elements?"

Jan A. Rajchman: I certainly think that magnetics remain the dominant technique. As I said, magnetic switching, as a corollary to magnetic storage and in many other applications, is likely to remain a dominant art in the forseeable future.

I think that the combination of magnetics and transistors is likely to play an important role in many applications.

A Command Structure for Complex Information Processing

J. C. SHAW NONMEMBER AIEE
A. NEWELL NONMEMBER AIEE
H. A. SIMON NONMEMBER AIEE
T. O. ELLIS NONMEMBER AIEE

THE general purpose digital computer, by virtue of its large capacity and general-purpose nature, has opened the possibility of research into the nature of complex mechanisms per se. The challenge is obvious: humans carry out information processing of a complexity that is truly baffling. Given the urge to understand either how humans do it, or alternatively, what kinds of mechanisms might accomplish the same tasks, the computer is turned to as a basic research tool. The varieties of complex information processing will be understood when they can be synthesized: when mechanisms can be created that perform the same processes.

The last few years have seen a number of attempts at synthesis of complex processes. These have included programs to discover proofs for theorems,[1,2] programs to synthesize music,[3] programs to play chess,[4,5] and programs to simulate the reasoning of particular humans.[6] The feasibility of synthesizing complex processes hinges on the feasibility of writing programs of the complexity needed to specify these processes for a computer. Hence, a limit is imposed by the limit of complexity that the human programmer can handle. The measure of this complexity is not absolute, for it depends on the programming language he uses. The more powerful the language, the greater will be the complexity of the programs he can write. The authors' work has sought to increase the upper limit of complexity of the processes specified by developing a series of languages, called information processing languages (IPL's), that reduce significantly the demands made upon the programmer in his communication with the computer. Thus, the IPL's represent a series of attempts to construct sufficiently powerful languages to permit the programming of the kinds of complex processes previously mentioned.

The IPL's designed so far have been realized interpretively on current computers.[7] Alternatively, of course, any such language can be viewed as a set of specifications for a general-purpose computer. An IPL can be implemented far more expeditiously in a computer designed to handle it than by interpretation in a computer designed with a quite different command structure. The mismatch between the IPL's designed and current computers is appreciable: 150-machine cycles are needed to do what one feels should take only 2 or 3 machine cycles. (It will become apparent that the difficulty would not be removed by "compiling" instead of "interpreting," to resurrect a set of well-worn distinctions. The operations that are mismatched to current computers must go on during execution of the program, and hence cannot be compiled out.)

The purpose of this paper is to consider an IPL computer, that is, a computer constructed so that its machine language is an information processing language. This will be called language *IPL-VI*, for it is the sixth in the series of IPL's that have been designed. This version has not been realized interpretively, but has resulted from considering hardware requirements in the light of programming experience with the previous languages.

Some limitations must be placed on the investigation. This paper will be concerned only with the central computer, the command structure, the form of the machine operations, and the general arrangements of the central hardware. It will neglect completely input-output and secondary storage systems. This does not mean these are unimportant or that they present only simple problems. The problem of secondary storage is difficult enough for current computing systems; it is exceedingly difficult for IPL systems, since in such systems initial memory is not organized in neat block-like packages for ease of shipment to the secondary store.

Nor is it the case that one would place an order for the IPL computer about to be described without further experience with it. Results are not entirely predictable. IPL's are sufficiently different from current computer languages that their utility can be evaluated only after much programming. Moreover, since IPL's are designed to specify large complicated programs, the utility of the linguistic devices incorporated in them cannot be ascertained from simple examples.

One more caution is needed to provide a proper setting for this paper. Most of the computing world is still concerned with essentially numerical processes, either because the problems themselves are numerical or because nonnumerical problems have been appropriately arithmetized. The kinds of problems that the authors have been concerned with are essentially nonnumerical, and they have tried to cope with them without resort to arithmetic models. Hence the IPL's have not been designed with a view to carrying out arithmetic with great efficiency.

Fundamental Goals and Devices

The basic aim, then, is to construct a powerful programming language for the class of problems concerned. Given the amount and kind of output desired from the computer, a reduction in the size and complexity of the specification (the program) that has to be written in order to secure this output is desired.

The goal is to reduce programming effort. This is not the same as reducing the computing effort required to produce the desired output from the specification. Programming feasibility must take precedence over computing economics; since it is not yet known how to write a program that will enable a computer to teach itself to play chess, it is premature to ask whether it would take such a computer one hour or one hundred hours to make a move. This is not meant as an apology, but as support for the contention that, in seeking to write programs for very large and complicated tasks, the overriding initial concerns must be to attain enough flexibility, abbreviation, and automation of the underlying computing processes to make programming feasible. And these concerns have to do with the power of the programming language rather than the efficiency of the system that executes the program.

In the next section a straightforward description of an IPL computer is begun. To put the details in a proper setting, the remainder of this section will be devoted to the basic devices that *IPL-VI* uses to achieve a measure of power and flexibility. These devices include: organization of memory into list structure, provision for breakouts, identity of data with program, two-stage interpretation, invariance of program during execution,

J. C. SHAW, A. NEWELL, and T. O. ELLIS are with The Rand Corporation, Santa Monica, Calif.

H. A. SIMON is with the Carnegie Institute of Technology, Pittsburgh, Pa.

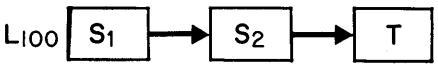


Fig. 1. A simple list

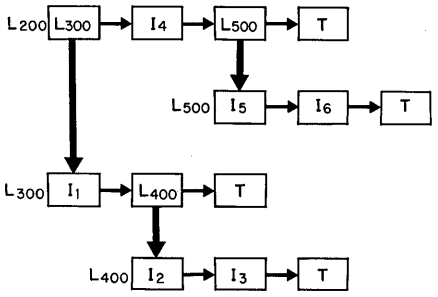


Fig. 2. A list structure

provision for responsibility assignments, and centralized signalling of test results.

List Structure

The most fundamental and characteristic feature of the IPL's is that they organize memory into list structures whose arrangement is independent of the actual physical geometry of the memory cells and which undergo continual change as computation proceeds. In all computing systems the topology of memory, the characteristics of hardware and program that determine what memory cells can be regarded as "next to" a given cell, plays a fundamental role in the organization of the information processing. This is obviously true for serial memories like tape; it is equally true from random access memories. In random access memories the topological structure is derived from the possibility of performing arithmetic operations on the memory addresses that make use of the numerical relations among these addresses. Thus, the cell with address 1435 is next to cell 1436 in the specific sense that the second can be reached from the first by adding one to the number in a counter.

In standard computers use is made of the static topology based on memory addresses to facilitate programming and computation. Index registers and relative addressing schemes, for example, make use of program arithmetic and depend for their efficacy upon an orderly matching of the arrangement of information in memory with the topology of the addressing system.

When memory is organized in a list structure, the relation between information storage and topology is reversed. The topology of memory is continually modified to adapt to the changing needs of organization of memory content. No arithmetic operations on memory addresses are permitted; the topology is built on a single, asymmetric, modifiable, ordinal relation between pairs of memory cells which is called adjacency. The system contains processes that make use of the adjacency relations in searching memory, and processes that change these relations at will inexpensively in the course of processing.

A list structure can be established in computer memory by associating with each word in memory an address that determines what word is adjacent to it, as far as all the operations of the computer are concerned. Memory space of an additional address associated with each word is given up, so that the adjacency relation can be changed as quickly as a word in memory can be changed. Having paid this price, however, many of the other basic features of IPL's are obtained almost without cost: unlimited hierarchies of subroutines; recursive definition of processes; variable numbers of operands for processes; and unlimited complexity of data structure, capable of being created and modified to any extent at execution time.

Breakouts

Languages require grammar-fixed structural features so that they can be interpreted. Grammar imposes constraints on what can be said, or said simply, in a language. However, the constraints created by fixed grammatical format can be alleviated at the cost of introducing an additional stage of processing by devices that allow one to "break out" of the format and to use more general modes of specification than the format permits. Devices for breakouts exchange processing time for flexibility. Several devices achieve this in *IPL-VI*. Each is associated with some part of the format.

As an illustrative example, *IPL-VI* has a single-address format. Without breakout devices, this format would permit an information process to operate on only a single operand as input, and would permit the operand of a process to be specified only by giving its address. Both of these limitations are removed: the first by using a special communication list to store operands, the second by allowing the address for an operand to refer either to the operand itself or to any process that will determine the operand.

The latter device, which allows broad freedom in the method of specifying an operand, illustrates another important facet of the flexibility problem. Breakouts are of great importance in reducing the burden of planning that is imposed

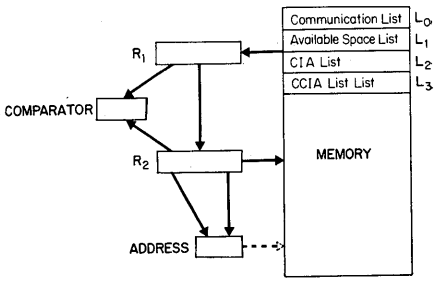


Fig. 3. Machine information transfer paths

on the programmer. It is certainly possible, in principle, to anticipate the need for particular operands at particular stages of processing, and to provide the operands in such a way that their addresses are known to the programmer at the appropriate times. This is the usual way in which machine coding is done. However, such plans are not obtained without cost; they must be created by the programmer. Indeed, in writing complex programs, the creation of the plan of computation is the most difficult part of the job; it constitutes the task of "programming" that is sometimes distinguished from the more routine "coding." Thus, devices that exchange computing time for a reduction in the amount of planning required of the programmer provide significant increases in the flexibility and power of the language.

Identity of Data with Programs

In current computers, the data are considered "inert." They are symbols to be operated upon by the program. All "structure" of the data is initially developed in the programmer's head and encoded implicitly into the programs that work with the data. The structure is embodied in the conventions that determine what bits the processes will decode, etc.

An alternative approach is to make the data "active." All words in the computer will have the instruction format: there will be "data" programs, and the data will be obtained by executing these programs. Some of the advantages of this alternative are obvious: the full range of methods of specification available for programs is also available for data; a list of data, for example, may be specified by a list of processes that determine the data. Since data are only desired "on command" by the processing programs, this approach leads to a computer that, although still serial in its control, contains at any given moment a large number of parallel active programs, frozen in the midst of operation and waiting until called upon to produce the

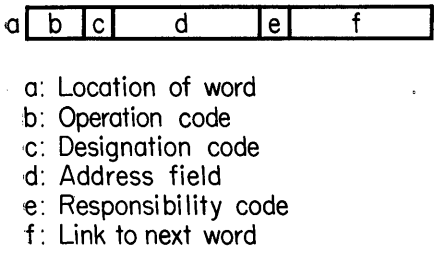


Fig. 4. IPL word format

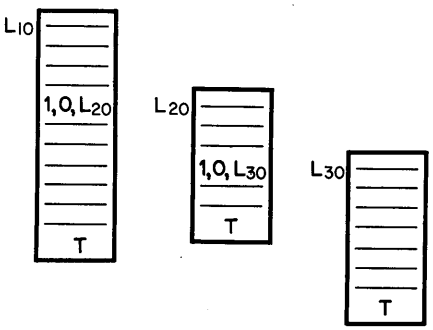


Fig. 5. A simple subroutine hierarchy

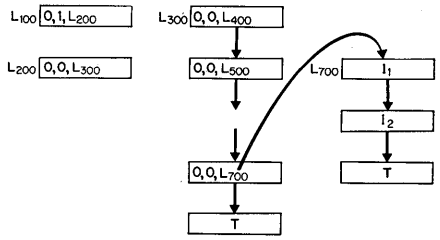


Fig. 6. Example of finding last item of last sublist

next operation or piece of data. This identity of data with program can be attained only if the processing programs require for their operation no information about the structure of the data programs, only information about how to receive the data from them.

Two-stage Interpretation

To identify the operand of an *IPL-VI* instruction, a designating operation operates on the address part of the instruction to produce the actual operand. Thus, depending on what designating operation is specified, the address part may itself be the operand, may provide the address of the operand, or may stand in a less direct relation to the operand. The designating operation may even delegate the actual specification of the operand to another designating operation.

Invariance of Program During Execution

In order to carry out generalized recursions, it is necessary to provide for the storage of indefinite amounts of variable information necessary for the operation of such routines. In *IPL-VI* all the variable information is stored externally to the associated routine, so that the routine remains unmodified during execution. The name of a routine can appear in the definition of the routine itself without causing difficulty at execution time.

Responsibility Assignments

The automatic handling of such processes as erasing a list, or searching through a list requires some scheme for keeping track of what part of the list has been processed, and what part has not. For example, in erasing a program containing a local subroutine that appears more than once within the program, care must be taken to erase the subroutine once and only once. This is accomplished by a system for assigning responsibility for the parts of the list. In general, the responsibility code in *IPL-VI* handles these matters without any explicit attention from the programmer, except in those few situations where the issue of responsibility is the central problem.

Centralized Signalling of Test Results

The structure of the language is simplified by having all conditional processes set a switch to symbolize their output instead of producing an immediate conditional transfer of control. Then, a few specialized processes are defined that transfer control of the basis of the switch setting. By symbolizing and retaining the conditional information, the actual transfer can be postponed to the most convenient point in the processing. The flexibility obtained by this device proves especially useful in dealing with the transmission of conditional information from subroutines to the routines that call upon them.

General Organization of the Machine

The machine that is described can profitably be viewed as a "control computer." It consists of a single control unit with access to a large random-access memory. This memory should contain 10^5 words or more. If less than 10^4 words are available in the primary memory, there will probably be too frequent occasions for transfer of information between primary and secondary storage to make the system profitable.

The operation of the computer is entirely nonarithmetic, there being no arithmetic unit. Since arithmetic processes are not used as the basis of control, as they are in standard computers, such a unit is inessential, although it would be highly desirable for the computer to have access to one if it is to be given arithmetic tasks. The computer is perfectly capable of proving theorems in logic or playing chess without an arithmetic adjunct.

Memory

The memory consists of cells containing words of fixed length. Each word is divided into two parts, a symbol and a link. The entire memory is organized into a list structure in the following way. The link is an address; if the link of a word a is the address of word b, then b is adjacent to a. That is, the link of a word in a simple list is the address of the next word in the list.

The symbol part of a word may also contain an address, and this may be the address of the first word of another list. As indicated earlier, the entire topology of the memory is determined by the links and by addresses located in the symbol parts of words. The links permit the creation of simple lists of symbols; the links and symbol parts together, the creation of branching list structures.

The topology of memory is modified by changing addresses in links and symbol parts, thereby changing adjacency relations among words. The modification of link addresses is handled directly by various list processes without the attention of the programmer. Hence, the memory can be viewed as consisting of symbol occurrences connected together by mechanisms or structure whose character need not be specified.

The basic unit of organization is the list, a set of words linked together in a particular order by means of their link parts, in the way previously explained. The address of the first word in the sequence is the name of the list. A special terminating symbol T, whose link is irrelevant, is in the last word on every list. A simple list is illustrated in Fig. 1; its name is L_{100}, and it contains two symbols, S_1 and S_2.

The symbols in a list may themselves designate the names of other lists. (The symbols themselves have a special format, so that they are not names of lists but designate the names in a manner that will be described.) Thus, a list may be a list of lists, and each of its sublists may be a list of lists.

An example of a list structure is shown in Fig. 2. The name of the list structure is the name of the main list, L_{200}. L_{200} contains two sublists, L_{300} and L_{500}, plus an

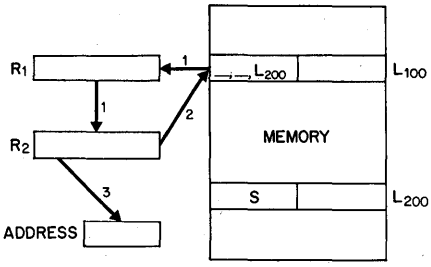


Fig. 7. Information transfers in c=2 operation

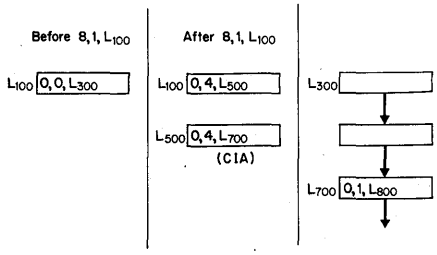


Fig. 8. Example of a data program

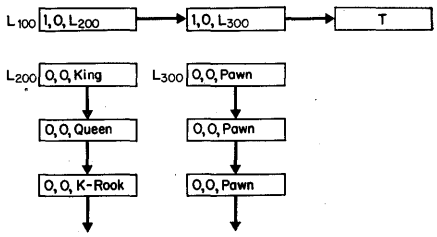


Fig. 9. Application of a data program to chess

item of information, I_4, that is not a name of a list. L_{300} in its turn consists of item I_1 plus another sublist, L_{400}, while L_{500} contains just information, and is not broken out further into sublists. Each of these lists terminates in a word that holds the symbol T.

Available Space List

A list uses a certain number of cells from memory. Which cells it uses is unimportant as long as the right linkages are set up. In executing programs that continually create new lists and destroy old ones, two requirements arise. When creating a list, cells in memory must be found that are not otherwise occupied and so are available for the new list. Conversely, when a list is destroyed (when it is no longer needed in the system) its cells become available for other uses, but something must be done to gain access to these available cells when they are needed.

The device used to accomplish these two logistic functions is the available space list. All cells that are available are linked together into the single long list. Whenever cells are needed, they are taken from the front of this available space list; whenever cells are made available, they are inserted on the front of the available space list just behind the fixed register that holds the link to the first available space. The operations of taking cells from the available space list and returning cells to the available space list involve, in each case, only changes of addresses in a pair of links.

Organization of Central Unit

Fig. 3 shows the special registers of the machine and the main information transfer paths. Four addressable registers accomplish fixed functions. These are shown as part of the main memory, but would be fast access registers.

Communication List, L_0. The system allows the introduction of unlimited numbers of processes with variable numbers of inputs and outputs. The communication of inputs and outputs among processes is centralized in a communication list with known name, L_0. All subroutines find their inputs on this list, and all subroutines put their outputs on the same list.

Available Space List, L_1. All cells not currently being used are on the available space list: cells can be obtained from it when needed and are returned to it when they are no longer being used.

List of Current Instruction Addresses (CIA), L_2. At any given moment in working sequentially through a program, there will be a whole hierarchy of instructions that are in process or interpretation, but whose interpretation has not been completed. These will include the instruction currently being interpreted, the routine to which this instruction belongs, the superroutine to which this routine belongs, and so on. The CIA list is the list of addresses of this hierarchy of routines. The first symbol on the list gives the address of the instruction currently being interpreted; the second symbol gives the address of the current instruction in the next higher routine, etc. In this system it proves to be preferable to keep track of the current instruction being interpreted, rather than the next one.

List of Current CIA Lists, L_3. The control sequence is complicated in this computer by the existence of numerous programs which become active when called upon, and whose processing may be interspersed among other processes. Hence, a single CIA list does not suffice; there must be such a list for each program that has not been completely executed. Therefore, it is necessary also to have a list that gives the names of the CIA lists that are active. This list is L_3.

Besides these special addressable registers, three nonaddressable registers are needed to handle the transfers of information. Two of these, R_1 and R_2, are each a full word in length, and transfer information to and from memory. Register R_1 receives input from memory; R_2 transmits output to memory. The comparator that provides the information for all tests takes as its input for comparison the symbols in R_1 and R_2. This pair of registers also performs a secondary function in regenerating words in memory: the basic "read" operation from memory is assumed to be destructive; a nondestructive "read" merely shunts the word received from memory in R_1 to R_2 and back, by means of a "write" operation, to the same memory cell.

A register, A, which holds a single address, controls references to the memory, that is, specifies the memory address at which a "read" or "write" operation is to be performed. References to the four addressable registers, L_0 to L_3, can be made either by A or directly by the control unit itself; other memory cells can be referred to only by A. Finally, the computer has a single bit register which is used to encode and retain test results.

The Environment

How input-output, secondary storage, and high-speed arithmetic could be handled with such a machine will be indicated. The machine manipulates symbols: it can construct complex structures, search them, and tell when two symbol occurrences are identical. These processes are sufficient to play chess, prove theorems, or do most other tasks. The symbols it manipulates are not "coded"; they simply form a set of arbitrary distinguishable entities, like a large alphabet.

This computer can manipulate things outside itself if hardware is provided to make some of its symbols refer to outside objects, and other symbols refer to operations on these objects. It could do high-speed arithmetic, for example, if some of its symbols were names of words in memory encoded as numbers as in the usual computer fashion, and others were names of the arithmetic operations. In

such a scheme these words would not be in the IPL language; they would have some format of their own, either fixed or floating-point, binary or decimal. They might occupy the same physical memory as that used by the control computer. Thus the IPL language would deal with numbers at one remove, by their names, in much the same manner as the programmer deals with numbers in a current computer. A similar approach can be used for manipulating printers, input devices, etc.

The Word and Its Interpretation

All words in IPL have the same format, shown in Fig. 4. The word a is divided into two major parts: the symbol part, $bcde$, and the link, f. It has been observed that the programmer never deals explicitly with the link, although it will be frequently represented explicitly to show how manipulations are being accomplished. Since the same symbol can appear in many words, the symbol occurrence of the symbol in the word a will be discussed.

A symbol occurrence consists of an operation, b, a designation operation, c, an address, d, and a responsibility code, e. The operation, b, takes as operand a single symbol occurrence, which is called s. The operand, s, is determined by applying the designation operation, c, to the address, d. Thus, the process determined by a word is carried out in two stages: the first-stage operation (the designation operation) determines an operand that becomes the input to the second-stage operation.

The Responsibility Bit

The single bit, e, is an essential piece of auxiliary information. The address, d, in a symbol may be the address of another list structure. The responsibility code in a symbol occurrence indicates whether this occurrence is "responsible" for the structure designated by d. If the same address, d, occurs in more than one word, only one of these will indicate responsibility for d.

The main function of the responsibility code is to provide a way of searching a branching list structure so that every part of the structure will, sooner or later, be reached, and so that no part will be reached twice. The need for a definite assignment of responsibility for the various parts of the structure can be seen by considering the process of erasing a list. Suppose that a list has a sublist that appears twice on it, but that does not appear anywhere else in memory. When the list is erased, the sublist must be erased if it is not to be lost forever, and the space it occupies with it. However, after the sublist has been erased when an occurrence of its name is encountered on the other list, it is imperative that it not be erased again on the second encounter. Since the words used by the sublist would have been returned to the available space list prior to the second encounter, only chaos could result from erasing it again. The responsibility code would indicate responsibility, in erasing, for one and only one of the two occurrences of the name of the sublist.

Detailed consideration of systems of responsibility is inappropriate in this paper. It is believed that an adequate system can be constructed with a single bit, although a system that will handle merging lists also requires a responsibility bit on the link f. The responsibility code is essentially automatic. The programmer does not need to worry about it except in those cases where he is explicitly seeking to modify structure.

Interpretation Cycle

A routine is a list of words, that is, a list of instructions. Its name is the address of the first word used in the list. The interpretation of a program proceeds according to a very simple cycle. An instruction is fetched to the control unit. The designation operation is decoded and executed, placing the location of s in the address register, A, of Fig. 3. Then operation b is decoded and performed on s. The cycle is then repeated using f to fetch the next instruction.

The Operation Codes

The simple interpretation cycle previously described provides none of the powerful linguistic features that were outlined at the beginning of the paper: hierarchies of subroutines, data programs, breakouts, etc. These features are obtained through particular b and c operations that modify the sequence of control. The operation codes will be explained under the following headings: the designation code, sequence-controlling operations, save and delete operations, communication list operations, signal operations, list operations, and other operations.

The Designation Code

The designation operation, c, operates on the address, d, to designate a symbol occurrence, s, that will serve as input, or operand, for the operation b. The designation operation places the address of the designated symbol, s, in the address register.

The designation codes proposed, based on their usefulness in coding with the IPL's, are shown in Appendix I. The first four, $c=0$, 1, 2, or 3, allow four degrees of directness of reference. They are usable when the programmer knows in advance where the symbol, s, is located. To illustrate their definition, consider an instruction a_1, with parts b_1, c_1, d_1, and e_1, which can collectively be called s_1. The address part, d_1, of this instruction may be the address of another instruction $d_1=a_2$; the address part, d_2, of a_2 may be the address of a_3, etc.

The code $c_1=1$ means that s is the symbol whose address is d_1, that is, the symbol s_2. In this case the designating operation puts d_1, the address of s_2, in the address register. The code $c_1=2$ means that s is s_3; hence, the operation puts d_2, the address of s_3, in the address register. The code $c_1=3$ puts d_3, the address of s_4, in the address register. Finally, $c_1=0$ designates as s the actual symbol in a_1 itself; hence, this means that b is to operate on s_1. Therefore, this operation places a_1 in the address register.

The remaining two designation operations, $c=4$ and 5, introduce another kind of flexibility, for they allow the programmer to delegate the designation of s to other parts of the program. When $c_1=4$, the task of designating s is delegated to the symbol of the word $d_1=a_2$. In this case, s is found by applying the designation operation, c_2 of word a_2, to the address, d_2, of word a_2. An operation of this kind permits the programmer to be unaware of the way in which the data are arranged structurally in memory. Notice that the operation permits an indefinite number of stages of delegation, since if $c_2=4$, there will be a further delegation of the designation operation to c_3 and d_3 in word a_3.

The last designation operation, $c=5$, provides both for delegation and a breakout. With $c_1=5$, d_1 is interpreted as a process that determines s. Any program whatsoever, having its initial instruction at d_1, can then be written to specify s. When this program has been executed, an s will have been designated, and the interpretation will continue by reverting to the original cycle, that is, by applying b_1 to the s that was just designated. It is necessary to provide a convention for communicating the result of process d_1 to the interpreter. The convention used is that d_1 will leave the location of s in L_0, the standard communication cell.

Sequence-Controlling Operations

Appendix II lists the 35 b operations. The first 12 of these are the ones that affect the sequence of control. They accomplish 5 quite different functions: executing a process ($b=1,10$), executing variable instructions ($b=2$), transferring control within a routine ($b=3,4,5$), transferring control among parallel program structures ($b=0,6,7,8,9,$), and, finally, stopping the computer ($b=11$).

A routine is a list of instructions; its name is the address of the first word in the list. To execute a routine, its name (i.e., its name becomes the s of the previous section) is designated and to it is applied the operation $b=1$, "execute s." The interpreter must keep track of the location of the instruction that is being executed in the current routine and return to that location after completing the execution of the instruction (which, in general, is a subroutine). All lists end in a word containing $b=10$, which terminates the list and returns control to the higher routine in which the subroutine just completed occurred. (The symbol T is really any symbol with $b=10$.)

Fig. 5 provides a simple illustration of the relations between routines and their subroutines. In the course of executing the routine L_{10} (i.e., the instructions that constitute list L_{10}), an instruction, $(1,0,-L_{20})$, is encountered that is interpreted as "execute L_{20}." In the course of executing L_{20}, an instruction is encountered that is interpreted as "execute L_{30}." Assuming that L_{30} contains no subroutines, its instructions will be executed in order until the terminate instruction is reached. Because of the 10 in its b part, this instruction returns control to the instruction that follows L_{30} in L_{20}. When the final word in L_{20} is reached, the operation code 10 in its b part returns control to L_{10}, which then continues with the instruction following L_{20}. (Only the b part, $b=10$, of the terminal word in a routine is used in the interpretation; the c and d parts are irrelevant.) This is a standard subroutine linkage, but with all the sequence control centralized.

The operation code $b=2$, "interpret s," delegates the interpretation to the word s. The effect of an instruction containing $b=2$ is exactly the same as if the instruction contained, instead, the symbol, s, that is designated by its c and d parts. One can think of the instruction with $b=2$ as a variable whose value is s. Thus, a routine can be altered by modifying the symbol occurrence s, without any modification whatsoever in the words belonging to the routine itself.

The three operations, $b=3,4$, and 5, are standard transfer operations. The first is an unconditional transfer; the two others transfer conditionally on the signal bit. As mentioned earlier, all binary conditional processes set the signal either "on" or "off." In order to describe operations $b=0,6,7,8,9$ the concept of program structure must be defined. A program structure is a routine together with all its sub-routines and designation processes. Such a structure corresponds to a single, although perhaps complex, process. The computer is capable of holding, at a given time, any number of independent program structures, and can interrupt any one of these processes, from time to time, in order to execute one of the others. All of these structures are coordinate, or parallel, and the operations $b=0,6,7,8,9$, are used to transfer control, perhaps conditionally, from the one that is currently active to a new one or to the previously active one. In this sense, the computer being described may be viewed as a serial control, parallel program machine.

The execution of a particular routine in program structure A will be used as an example. Operation $b=6$ will transfer control to an independent program structure determined by s; call it B. The machine will then begin to execute B. When it encounters a "stop interpretation" operation ($b=0$) in B, control will be returned to the program structure, A, that was previously active. But the "stop interpretation" operation, unlike the ordinary termination, $b=10$, does not mark the end of program structure B. At any later point in the execution of A, control may again be transferred to B, in which case execution of the latter program will be resumed from the point where it was interrupted by the earlier "stop interpretation" command. The operation that accomplishes the second transfer of control from A to B is $b=7$, "continue parallel program s." Thus, $b=0$ is really an "interrupt" operation, which returns control to the previous structure, but leaves the structure it interrupts in condition to continue at a later point. There can be large numbers of independent program structures all "open for business" at once, with a single control passing from one to the other, determining which has access to the processing facilities, and gradually executing all of them. Operations $b=8$ and 9 simply allow the interruption to be conditional on the test switch.

Notice that the passage of control from one structure to another is entirely decentralized; it depends upon the occurrence of the appropriate b operations in the program structure that has control.

When control is transferred to a parallel program structure, either of two outcomes is possible. Either a "stop interpretation" instruction is reached in the structure to which control has been transferred, or execution of that structure is completed and a termination reached. In either case, control is returned to the program structure that had it previously, together with information as to whether it was returned by interruption or by termination. Thus, $b=0$ turns the signal bit on when it returns control; $b=10$ in the topmost routine of a structure turns the signal off.

The operation, $b=11$, simply halts. Processing continues from the location where it halted upon receipt of an external signal, "go."

Save and Delete Operations

The two operations, $b=12$ and 13, are sufficiently fundamental to warrant extended treatment. For example, consider a word, L_{100}, that contains the symbol I_1:

Location	Symbol	Link
L_{100}	I_1	t

The link of L_{100}, t, indicates that the next word holds the termination operation, $b=10$. The "save" operation ($b=12$) provides a copy of I_1 in such a way that I_1 can later be recalled, even if in the meantime the symbol in L_{100} has been changed. After the "save" operation has been performed on $s=L_{100}$, the result is:

Location	Symbol	Link
L_{100}	I_1	L_{200}
L_{200}	I_1	t

A new cell, which happened to be L_{200}, was obtained during the "save" operation from the available space list, L_1, and a copy of I_1 was put in it. The symbol in L_{100} can now be changed without losing I_1 irretrievably. Suppose a different symbol is copied, for example, I_2, into L_{100}. Then:

Location	Symbol	Link
L_{100}	I_2	L_{200}
L_{200}	I_1	t

Although I_1 has been replaced in L_{100}, I_1 can be recovered by performing the "delete" operation, $b=13$. Before the "delete" operation is explained, it will be instructive to show what happens when the "save" operation on L_{100} is interated.

If it is executed again, it will make a copy of I_2. Therefore:

Location	Symbol	Link
L_{100}	I_2	L_{300}
L_{300}	I_2	L_{200}
L_{200}	I_1	t

Notice that the cell L_{200}, in which the copy of symbol I_1 is retained, was not affected at all by this second "save" operation. Only the top cell in the list and the new cell from the available space list are involved in the transaction of saving. The same process is performed no matter how long the list that trails out below L_{100}; thus, the save operation can be applied as many times as desired with constant processing time.

The "delete" operation, $b=13$, applied to the symbol I_2 in L_{100}, will now be illustrated. This operation puts the symbol and link of the second word in the list, L_{300}, into the first cell, L_{100}, and puts L_{300} back on the available space list, with the following result:

Location	Symbol	Link
L_{100}	I_2	L_{200}
L_{200}	I_1	t

The result is the exact situation obtained before the last "save" was performed.

In the description of the "delete" operation up to this point, only the changes it makes in the "push-down" list, in this case L_{100}, have been considered. The operation does more than this, however; "delete s" also erases all structures for which the symbol s (I_1 and I_2 in the examples) is responsible. When a copy of a symbol is made, e.g., the operation that initially replaced I_1 by I_2 in L_{100}, the copy is not assigned responsibility for the symbol ($e=0$ was set in the copy). Thus, no additional erasing would be required in the particular "delete" operation illustrated. If, on the other hand, the I_2 that was moved into L_{100} had been responsible for the structure that could be reached through it (if it were the name of a list, for example), then a second "delete" operation, putting I_1 back into L_{100}, would also erase that list and put all its cells back on the available space list. Thus "delete" is also equivalent to "erase" a list structure.

Communication List Operations

In describing a process as a list of subprocesses, the question of inputs and outputs from the processes has been entirely by-passed. Since each subroutine has an arbitrary and variable number of operands as input, and provides to the routine that uses it an arbitrary number of outputs, some scheme of communication is required among routines. The communication list, L_0, accomplishes this function in IPL.

That the inputs and outputs to a routine be symbols is required. This is no real restriction since a symbol can be the name of any list structure whatever. Each routine will take as its inputs the first symbols in the L_0 list. That is, if a routine has three inputs, then the first three symbols in L_0 are its inputs. Each routine must remove its inputs from L_0 before terminating with $b=10$, so as to permit the use of the communication list by subsequent routines. Finally, each routine leaves its outputs at the head of list L_0.

The b operations 14 through 19 are used for communication in and out of L_0. Their one common feature is that, whenever they put a symbol in L_0, they save the symbol already there, that is, they push down the symbols already "stacked" in L_0. Likewise, whenever a symbol is moved from L_0 to memory, the symbol below it in L_0 "pops up" to become the top one. (To be precise, the responsibility bit travels with a symbol when it is moved. Hence for example, $b=16$ and 17, do not, unlike the "delete" operation, erase the structure for which $1L_0$ is responsible.)

The four operations, $b=14,15,16$, and 17, are the main in-out operations for L_0. Two options are provided, depending on whether the programmer wishes to retain the s in memory ($b=14$ and 16) or destroy it ($b=15$ and 17). (The move in operation 15 has the same significance as in 16 and 17; the responsibility bit moves with the symbol, and the symbol previously in the location of s, is recalled.)

Operation $b=18$ is a special input to aid in the breakout designation operation, $c=5$. Recall that the latter operation requires d to place the location of s, the symbol it determines, in L_0. Operation 18 allows the process d to accomplish this.

Operation $b=19$ provides the means for creating structures. It takes a cell, for example L_{200}, from available space, and puts its name, as the symbol $(0,0,L_{200})$, in the location of the designated symbol, s. The symbol s, previously in this location is pushed down and saved.

Signal Operations

Ten b operations are primarily involved in setting and manipulating the signal bit. Observe that the test of equality ($b=20$ and 21) is identity of symbols. Since there is nothing in the system that provides a natural ordering of symbols, inequality tests like $s>1L_0$, are impossible. ($1L_0$ means the symbol in L_0.) It is necessary to be able to detect the responsibility bit ($b=22$), since there are occasions when the explicit structure of lists is important, and not just the information they designate. Finally, although the signal bit is just a single switch, it is necessary to have two symbols, one corresponding to "signal on" and the other to "signal off" ($b=26$ and 27), so that the information in the signal can be retained for later use ($b=28$ and 29).

The sense of the signal is not arbitrary. In general "off" is used to mean that a process "failed," "did not find," or the like. Thus, in operations $b=6$ and 7, the failure to find a "stop interpretation" operation sets the signal to "off." Likewise, the end of a list will be symbolized by setting the signal to "off."

List Operations

Both the "save" and "delete" operations are used to manipulate lists, but besides these, several others are needed. The three operations, $b=30$, 31, 32, allow for search over list structures. They can be paraphrased as: "get the referent," "turn down the sublist," and "get the next word of the list." They all have in common that they replace a known symbol with an unknown symbol. This unknown symbol need not exist; that is, the symbol referred to may contain a $b=10$ operation, which means that the end of the list has been reached. Consequently, the signal is always set "on" if the symbol is found, and "off" if the symbol is not found. One of the virtues of the common signal is apparent at this point, since, if the programmer knows that the symbol exists, he will simply ignore the signal. Instruction formats that provide for additional addresses for conditional transfers would force the programmer to attend to the condition even if it only meant leaving a blank space in the program.

To illustrate how these search operations work, Fig. 6 shows a list of lists, L_{300}, and a known cell, L_{100}. Cell L_{100} contains the reference to the list structure. The programmer does not know how the list, L_{300}, is referenced. He wants to find the last symbol on the last list of the structure. His first step is $(30,1,L_{100})$ which replaces the reference by the name of the list, L_{300}. He then searches down to the end of list L_{300} by doing a series of operations: $(32,1,L_{100})$. Each of these replaces one location on the list by the next

one. In fact, a loop is required, since the length of the list is unknown. Hence, after each "find the next word" operation, he must transfer, on the basis of the signal, back to the same operation if the end of the list hasn't been reached. The net result, when the end of the list is reached, is that the location of the last word on list L_{300} rests in L_{100}. Since in this example he wants to go down to the end of the sublist of the last word on the main list, he next performs $(31,1,L_{100})$. This operation replaces the location of the last word with the name of the last list, L_{700}. Now the search down the sublist is repeated until the end is again reached, at this point the location of the last symbol on the last list is in L_{100}, as desired. The sequence of code follows:

Location	Symbol	Link
	b c d	
	30,1,L_{100}	
L_{888}	32,1,L_{100}	
	4,0,L_{888}	
	31,1,L_{100}	
L_{999}	32,1,L_{100}	
	4,0,L_{999}	

The operations, $b=33$ and 34, allow for inserting symbols in a list either before or after the symbol designated. The lists in this system are one-way: although there is always a way of finding the symbol that follows a designated symbol, there is no way of finding the symbol that precedes a designated symbol. The "insert before" operation does not violate this rule. In both operations, 33 and 34, a cell is obtained from the available space list and inserted after the word holding the designated symbol. (This is identical with the first step of the "save" operation.) In the "insert before" operation ($b=33$) the designated symbol, s, is copied into the new cell, and $1L_0$ is moved into the previous location of s. In "insert after" ($b=34$), the designated symbol is left unchanged, and $1L_0$ is moved into the new cell. In both cases $1L_0$ is moved, that is, it no longer remains at the head of the communication list.

Other Operations

This completes the account of the basic complement of operations for the IPL computer. These form a sufficient set of operations to handle a wide range of nonnumerical problems. To do arithmetic efficiently, one would either add another set of b's covering the standard arithmetic operations or deal with these operations externally via a breakout operation on b (not formally defined here) that would move a full symbol into a special register for hardware interpretation relative to external machines: adders, printers, tapes, etc.

The set of operations has not been described for reading and writing the various parts of the word: *b, c. d, e,* and *f* (although it may be possible to automatize this last completely). These operations rarely occur, and it seemed best to ignore them as well as the input-output operations in the interest of simple presentation.

Interpretation

This section will describe in general terms the machine interpretation required to carry out the operation codes prescribed. There is not enough space to be exhaustive, therefore selected examples will be discussed.

Direct Designation Operations

Fig. 7 shows the information flows for $c=2$, an operation that is typical of the first four designation operations. These flows follow a simple, fixed interpretation sequence. Assume that instruction (–,2, L_{100}) is inside the control unit. The contents of L_{100} are brought into R_1, the input register, then transferred to R_2, the output register, and back to L_{100} again. The d part of R_2 now contains the location of s, and this location is transferred from R_2 to the address register.

Execute Subroutine ($b=1$)

When "execute s" is to be interpreted, the address register already contains the location of s, which was brought in during the first stage of the interpretation cycle. L_2, the current instruction address list (CIA), holds the address of the instruction containing the "execute" order. A "save" operation is performed on L_2, and s is transferred into L_2, which ends the operation. The result is to have the interpreter interpret the first instruction on the next sublist, and to proceed down it in the usual fashion. Upon reaching the terminate operation, $b=10$, the delete operation is performed on $1L_2$, thus bringing back the original instruction address from which the subroutine was executed. Now, when the interpretation cycle is resumed, it will proceed down the original list. Thus, the two operations, save and delete, perform the basic work in keeping track of subroutine linkage.

Parallel Programs

A single program structure, that is, a routine with all its subroutines, and their subroutines etc., requires a CIA list in order to keep track of the sequence of control. In order to have a number of independent program structures, a CIA list is required for each. L_3 is the fixed register which holds the name of the current CIA list. The name of the CIA list for the program structure which is to be reactivated on completion or interruption of the current program structure is the second item on the L_3 list, etc. Therefore, the L_3 list is appropriately called the current CIA list. The "save" and "delete" operations are used to manipulate L_3 analogously to their use with L_2 previously described.

Appendix III gives a more complete schematic representation of the interpretation cycle. It has still been necessary to represent only selected b operations.

Data Programs

In the section on list operations a search of a list was described. There the data were passive; the processing program dictated just what steps were taken in covering the list. Consider a similar situation, shown in Fig. 8, where there is a working cell, L_{100}, which contains the name of a list, L_{300}. L_{300} is a data program. There is a program that wants to process the data of L_{300}, which is a sequence of symbols. This program knows L_{100}. To obtain the first symbol of data, it does $(6,1,L_{100})$, that is, "execute the parallel program whose name is in L_{100}." The result is to create a CIA list, L_{500}, put its name in L_{100}, and fire the program. Some sort of processing will occur, as indicated by the blank words of L_{300}. Presumably this has something to do with determining what the data are, although it might be some bookkeeping on L_{300}'s experience as a data file. Eventually L_{700} is reached, which contains $(0,1,L_{800})$. This operation stops the interpretation, and returns control to the original processing program. The first symbol of data is defined to be $1L_{800}$. The processing program can designate this by $4L_{100}$, since the sequence of $c=4$ prefixes in L_{100} and L_{500} pass along the interpretation until it ultimately becomes $1L_{800}$. Now the processing program can proceed with the data. It remains completely oblivious to the processing and structure that were involved in determining what was the first symbol of data. Similarly, although it is not shown, the processing program is able to get the second symbol of data at any time simply by doing a "continue parallel program $1L_{100}$" ($b=7$).

One virtue of the use of data programs is the solution it offers for "interpolated" lists. In working on a chess program, for example, one has various lists of men: pawns, pieces, pieces that can move more than one square, such as rooks,

queens, etc. One would like a list of all men. There already exists a list of all pieces and a list of all pawns. It would be desirable to compose these lists into a single long list without losing the identity of either of the short lists, since they are still used separately. In other words form a list whose elements are the two lists, but such that, when this list of lists is searched it looks like a single long list. Further, and this is the necessary condition for doing this successfully, one cannot afford to make the program that uses this list of lists know the structure. The operation "execute s" ($b=1$) is precisely the operation needed to accomplish this task in a data program. It says "turn aside and go down the sublist s." Since it does not have the operation $b=0$, it is not "data." It is simply "punctuation" that describes the structure of the data list, and allows the appropriate symbols to be designated. Fig. 9 shows a data list of the kind just described. The authors have taken the liberty of writing in the names of the chessmen.

The stretch of code that follows shows the use of a data program for a "table look up" operation. The table has arbitrary arguments, each of which has a symbol for its value. A_1, A_2, etc. have been used to represent the arguments. To find the value corresponding to argument A_5, for example, A_5 is put in the communication cell with (14,0,A_5). Then the data program is executed with (6,0, L_{100}). Control now lies with the table, which tests each argument against the symbol in the communication lists: i.e., A_5, and sets the signal accordingly. The program stops interpreting ($b=8$) at the word holding the value only if the arguments are the same. In this case it would stop, designating L_{350}. If no entry was found, of course, control would return to the inquiring program with the signal off.

Location	Symbol	Link
L_{100}	20,0,A_1	
	8,0,L_{300}	
	20,0,A_2	
	8,0,L_{320}	
	20,0,A_5	
	8,0,L_{350}	t

Conclusions

The purpose of this paper has been to outline a command structure for complex information processing, following some of the concepts used in a series of interpretive languages, called IPL's. The ultimate test of a command structure is the complex problems it allows one to solve that would not have been solved if the coding language were not available.

At least two different factors operate to keep problems from being solved on computers: the difficulty of specification, and the effort required to do the processing. The primary features of this command structure have been aimed at the specification problem. The authors have tried to specify the language requirements for complex coding, and then see what hardware organization allowed their mechanization. All the features of delegation, indirect referencing, and breakout imply a good deal of interpretation for each machine instruction. Similarly, the parallel program structure requires additional processing to set up CIA lists, and when a data symbol is designated, there is delegated interpreting through several words, each of which exacts its toll of machine time. If one were solely concerned with machine efficiency, one would require the programmer to so plan and arrange his program that direct and uniform processes would suffice. Considering the size of current computers and their continued rate of growth toward megaword memories and microsecond operations, it is believed that the ilmitation already lies with the programmer with his limited capacity to conceive and plan complicated programs. The authors certainly know this to be true of their own efforts to program theorem proving programs and chess playing programs, where the IPL languages or their equivalent in flexibility and also in power have been a necessary tool.

Considering the amount of interpretation, and the fact that interpretation uses the same operations as are available to the programmer; e.g., the save and delete operations, one can think of alternative ways to realize an IPL computer. At one extreme are interpretive routines on current computers, the method that the authors have been using. This is costless in hardware, but expensive in computing time. One could also add special operations to a standard repertoire to facilitate an interpretive version of the language. Probably much more fruitful is the addition of a small amount of very fast storage to speed up the interpreter. Finally, one could wire in the programs for the operations to get even more speed. It is not clear that there is any arrangement more direct than the wired in program because of the need of the interpreter to use the whole capability of its own operation code.

Appendix I. c Operations (Designating Operations)

c Nature of Operation for (a) = b c d e.

0 (a) is the symbol s.
1 d is the address of the symbol s.
2 d is the address of the address of the symbol s.
3 d is the address of the address of the address of the symbol s.
4 d is the address of the designating instruction that determines s.
5 d is the address (name) of a process that determines s.

Appendix II. b Operations

b Nature of Operation

SEQUENCE-CONTROL OPERATIONS

0 Stop interpreting; return to previous program structure.
1 Execute process named s.
2 Interpret instruction s.
3 Transfer control to location s.
4 Transfer control to location s, if signal is on.
5 Transfer control to location s, if signal is off.
6 Execute parallel program s; turn signal on if stops; off if not.
7 Continue parallel program s; turn signal on if stops; off if not.
8 Stop interpreting, if signal is on.
9 Stop interpreting, if signal is off.
10 Terminate.
11 Halt; proceed on go.

SAVE AND DELETE OPERATIONS

12 Save s.
13 Delete s (and everything for which s is responsible).

COMMUNICATION LIST OPERATIONS

14 Copy s into communication list, saving $1L_0$.
15 Move s into communication list, saving $1L_0$.
16 Move $1L_0$ into location of s, saving s.
17 Move $1L_0$ into location of s, destroying s.
18 Copy location of s into communication list, saving $1L_0$.
19 Create a new symbol in location of s, saving s.

SIGNALLING OPERATIONS

20 Turn signal on if $s=1L_0$, off if not.
21 Turn signal on if $s=1L_0$, off if not; delete $1L_0$.
22 Turn signal on if s is responsible, off if not.
23 Turn signal on.
24 Turn signal off.
25 Invert signal.
26 Copy signal into location of s.
27 Copy signal into location of s, saving s.
28 Set signal according to s.
29 Set signal according to s; delete s.

LIST OPERATIONS

30 Replace s by the symbol designated by s, and turn signal on; if symbol doesn't exist ($b=10$), leave s and turn signal off.

31 Replace s by the symbol in d of s and turn signal on; if symbol doesn't exist, leave s and turn signal off.
32 Replace s by the location of the next symbol after d of s and turn signal on (s replaced by "0,4, (f, part of d of s)"); if next symbol does not exist, leave s and turn signal off.
33 Insert $1L_0$ before s (move symbol from communication list).
34 Insert $1L_0$ after s (move symbol from communication list).

Appendix III. The Interpretation Cycle

1. Fetch the current instruction according to the current instruction address (CIA) of the current CIA list.

2. Decode and execute the c operation:
If $c=3$ replace d by d part of the word at address d, reduce c to $c=2$ and continue.
If $c=2$ replace d by d part of the word at address d, reduce c to $c=1$ and continue.
If $c=1$ put d in the address register and go to step 3.
If $c=0$ put CIA in the address register and go to step 3.
If $c=4$ replace c, d by the c, d parts of the word at address d and go to step 2.
If $c=5$ mark CIA "incomplete," save it, set a new CIA$=d$, and go to step 1.

3. Decode and execute the b operation:
(Some of the b operations which affect the interpretation cycle follow.)
If $b=0$ turn the signal on, delete CIA and go to step 4.
If $b=1$ save CIA, set a new CIA$=d$ part of s and go to step 1.
If $b=2$ replace b, c, d by s and go to step 2.
If $b=3$ replace CIA by the d part of s and go to step 1.
If $b=10$ delete CIA.
If no CIA "pops up" turn signal off, delete CIA and go to step 4.
If "popped up" CIA is marked "incomplete" fetch the current instruction again, move $1L_0$ into address register and go to step 3.
Otherwise go to step 4.

4. Replace CIA by the f part of the current instruction and go to step 1.

References

1. Empirical Explorations of the Logical Theory Machine, **A. Newell, J. C. Shaw, H. A. Simon.** *Proceedings of the Western Joint Computer Conference,* Institute of Radio Engineers, New York, N. Y., Feb. 1957.

2. The Logical Theory Machine, **A. Newell, H. A. Simon.** *Transactions,* Professional Group on Information Theory, *Ibid.,* vol. IT-2, no. 3, Sept. 1956.

3. An Experiment in Musical Composition, **F. P. Brooks.** *Transactions,* Professional Group on Electronic Computers, *Ibid.,* vol. EC-6, no. 3, Sept. 1957.

4. A Chess Playing Program for the IBM 704, **A. Bernstein, et al.** *Proceedings of the Western Joint Computer Conference,* AIEE, 1958.

5. Experiments in Chess, **J. Kister, et al.** *Journal,* Association for Computing Machinery, New York, N. Y., vol. 4, no. 2, Apr. 1957.

6. The Elements of a Theory of Human Problem Solving, **A. Newell, J. C. Shaw, H. A. Simon.** *Psychology Review,* vol. 65, Mar. 1958.

7. Programming the Logical Theory Machine, **A. Newell, J. C. Shaw.** *Proceedings of the Western Joint Computer Conference,* Institute of Radio Engineers, Feb. 1957.

The Selection of an Instruction Language

W. BUCHHOLZ
NONMEMBER AIEE

Synopsis: The selection of instruction sets for recently developed computers indicates a trend towards more complex instructions than were used in early stored program computers. Instruction formats for several machines are cited. More complex instructions make possible shorter programs with a gain in speed and storage space.

It is not easy to define a good starting point for selecting the set of instructions to be built into a new computer. From the user's point of view, one should examine some typical applications for the computer and so derive an instruction set to fit the job to be done. An economy-minded engineer might prefer to define a tightly knit organization of input-output, storage, and registers, and select instructions around them. Neither approach is at all likely to satisfy the criterion of a high performance-to-cost ratio.

A user might define a small set of quite elaborate operations. His language may contain a great deal of redundancy, employing names or symbols instead of numeric addresses. A single statement may consist of numerous related specifications which ideally form a single operation. To design a computer which directly interprets lengthy and redundant statements every time it traverses the inner loop of a program would sacrifice performance while raising the cost of registers and decoding circuits.

The cost-conscious engineer might define a large set of simple operations around his concept of efficient data paths. These operations are apt to be so low in information content that even programs for simple tasks will become too long and take too much time. Although the cost would be lower, performance can drop much faster than cost, thus again giving an unattractive ratio of performance to cost.

In practice, the designer finds it necessary to iterate around both loops, starting with a trial set based on experience with earlier computers. The process is a complex one, involving a great deal of juggling and fitting. There certainly is not a unique instruction set that can be considered best. Nor is there any direct relationship between the instruction set and cost or performance. The iterative design process can, however, result in an instruction set which very substantially improves the ratio of performance to cost.

Contrasting the instruction language of earlier computers with recent developments may serve to illustrate a trend.

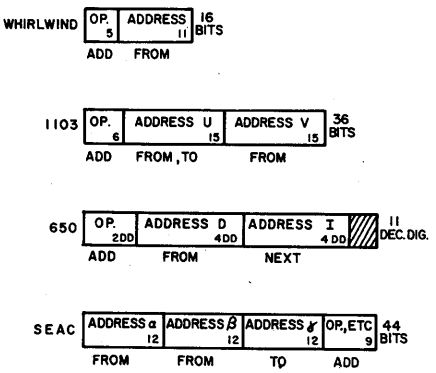


Fig. 1. Some classical instruction formats with one, two, and three addresses

Earlier Instruction Languages

The instruction formats of some of the earlier computers are reviewed in Fig. 1.

The Massachusetts Institute of Technology (MIT) Whirlwind computer represents the simplest of single-address instruction formats. It specifies the operation and the address of one of the operands. The other operand is implied to be in a working register.

The Remington-Rand *1103* (Univac Scientific) uses a 2-address scheme where two operands may be specified. The result may be returned to one of the two addresses.

The International Business Machines (IBM) *650* employs a different 2-address scheme. Only one address specifies an operand, the other operand residing in an implied working register. The second address specifies the next instruction.

W. Buchholz is with the International Business Machines Corporation, Poughkeepsie, N. Y.

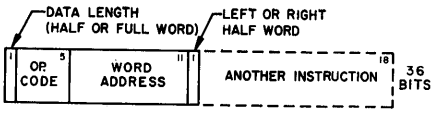


Fig. 2. Instruction format for IBM 701

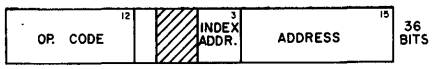


Fig. 3. Typical instruction format for IBM 704 and 709

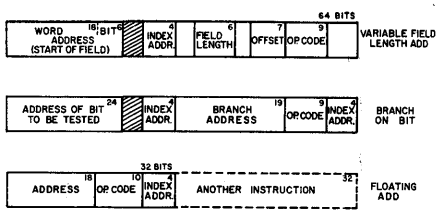


Fig. 4. Typical instruction formats for Los Alamos computer

This technique is advantageous with a revolving storage device because it permits locating instructions so as to minimize access time.

The National Bureau of Standards SEAC computer had available two instruction formats, one with three and another with four addresses. The 3-address format is shown. Two operands and a result may be specified.

The choice of these different formats often generated considerable discussion. In retrospect one wonders whether fitting instruction words to a desired data-word length was not as strong a factor in each choice as the intrinsic merit of the instruction format. The distinction is mainly whether one chooses to write related pieces of information vertically or horizontally on a sheet of paper. There is remarkably little difference between most of the early computers in the operations which they performed.

In all the early computers, simplicity was an important engineering consideration. After all, no one was quite sure in those days that the complex electronic devices parading under the imposing name of large-scale electronic data-processing machines would really get going.

The computers, however, were really usable and productive. They provided valuable experience for the designers of later computers. They clearly showed a need for much higher speed and much larger storage. At the same time, it became evident that speed could be gained and storage space could be saved by providing more built-in operations. A larger vocabulary can mean a quite drastic reduction in the number of instructions written and executed to do a given job. Floating-point arithmetic and automatic-address modification, or indexing, are two features which have become standard equipment on scientific computers. Alphabetic representation and variable field length have similarly become accepted as built-in functions for business data processors. The instruction set has been growing steadily in size and complexity.

The desire to specify more different things with one instruction has left no room in most instructions for more than one major address. The debate over multiple addresses has thus been settled by a process of evolution.

Evolution of the Single-Address Instruction

The illustrations for this evolutionary process will be taken from experience gathered at IBM over a number of years. The experience is not unique, and examples could have been chosen from other designs.

The IBM *701* follows the simple single-address pattern as in Fig. 2. To make efficient use of the word length selected for data representation, two instructions are packed in each word.

The IBM *704*, and later the IBM *709*, are both direct descendants of the IBM *701*, but they have a much bigger repertoire of instructions and features. As a result, the instruction has grown to fill the entire word as in Fig. 3.

Today's computing problems require larger memories. The address part of the instruction, therefore, was increased from 11 to 15 bits, giving 16 times the capacity of the IBM *701* memory. Three bits were added to specify indexing. The part of the instruction specifying the operation was increased from 5 to about 12 bits. Part of this increase was needed because the number of operations available to the user increased several times. Some bits were added to govern the interpretation of other bits, thus permitting more than one instruction format. For instance, there is a format in which two 15-bit quantities can be specified to provide a limited 2-address repertoire in the IBM *704* and *709*.

Some instruction formats of the new computer which IBM is developing for the Los Alamos Scientific Laboratory will be used to illustrate the current phase of this evolution. The goal of this project is to produce a computer which has about 100 times the performance of the IBM *704* on typical problems. This performance will, in part, be gained from new and faster components. Equally important though is a more sophisticated machine organization which greatly reduces the number of instructions that must be written and executed, particularly in the nonarithmetic portion of a program.

The basic word length of the Los Alamos computer, Fig. 4, is 64 bits, so almost twice that of the previous machines. The instructions, of course, expanded to fill every bit of this longer word. Only some of the functions are illustrated in Fig. 4. The first instruction permits the addition of fields of specified lengths which can be smaller than a full memory word.

The address has grown to 24 bits. The leftmost 18 bits can select one of 2^{18} full words in memory, which is 8 times the capacity of the IBM *704* and *709*. The next 6 address bits select a single bit of the 64 bits in the memory word. The bits in memory are numbered consecutively without regard to physical word boundaries imposed by the structure of the memory. Fields of different lengths may thus be stored in adjacent positions and addressed directly. The address used is that of the leftmost bit in the field.

A 4-bit index address specifies an index register to be used for automatic modification of the main address.

Another part of the instruction indicates the length of the field to be operated on. There is then no need to assign data bits to demarcate field lengths. The "offset" specifies a shift of the data to be added without the need for a separate instruction.

A variable field length "add" instruction thus requires four major specifications: address, index, length, and offset, as well as some minor ones not shown. With these it becomes possible to operate on any kind of binary coded data without format restrictions.

The second format illustrated is a "branch on bit" instruction. The 24-bit address specifies the bit to be tested, which may be any bit in memory or any bit in the addressable internal computer registers. A second address gives the location of the next instruction if the bit satisfies the branching condition. Two index addresses permit independent indexing of each of the two memory addresses.

The two examples illustrate the logical power of single instructions made possible by the long word length. Not all operations require this much informa-

tion in one instruction. A 64-bit instruction would be wasteful for the highly stylized floating-point and indexing operations with their fixed data formats.

The concept of varying the instruction length to suit the operation was explored. Vocabularies of half-length, quarter-length, and even eighth-length instructions were tried. While short instructions save space, it was found that the saving can be quickly offset by the extra bits needed to tell the computer how to interpret each format. The greatest economy of memory space and memory references was gained in a mixture of half-length and full-length instructions.

A floating-point "add" illustrates the half-length format. Only 18 bits of address are needed because a floating-point number occupies a full 64-bit word.

It has been found already that the flexibility and power of the new vocabulary will materially shorten the length of programs. In programming comparisons with the IBM *704* or *709*, reductions in program length up to a factor of 3 have been observed.

The Simplicity of Complexity

One may ask whether a more complex instruction set does not lead to more difficult programming. One answer is that programming can be simplified by adding instructions to complete a set (branch on plus, as well as branch on minus) and arranging them systematically. Another answer can be obtained by looking at the other extreme.

Van der Poel has shown[1] that the simplest instruction set theoretically consists of just one instruction. The instruction contains no operation code, only an address. Every instruction causes a combination of "subtract" and "store" to be executed; the difference replaces the contents of both the accumulator and the specified memory address. All other computing operations, including conditional branching, can be built up from this one instruction which is a very easy one to learn. But the programs needed to simulate no more than the elementary instruction set of early computers would be enormous. It is quite a task just to estimate the size of the program for a real job. It seems safe to say that the storage required would be gigantic, and a desk calculator would probably be faster.

A complex, but appropriate, language will, in fact, simplify the programmer's task as the problems to be done become more complex.

Relationship to Automatic Programming Languages

The advent of more powerful computers designed to tackle larger problems is thus accompanied by more elaborate and versatile instruction sets. Programs to do the same job require considerably fewer instructions and fewer references to memory. Or, to look at it another way, sequencing of simpler instructions stored in a relatively slow memory is replaced by internal sequencing with high-speed control circuits. This is a form of microprogramming using the fastest available memory, one made of transistor flip-flops.

Such an instruction set is still a long way from the "superlanguages" being developed under the heading of automatic programming. These languages are intended to simplify the task of the problem coder, not to raise the performance of the machine. The instruction set is an intermediate level between the programmer's language and the language of the elementary control steps inside the machine.

A 2-step process of translation is thus required. One is the programmed assembly of machine instructions from the statements in the superlanguage. The other is the internal translation of instructions to control sequences. The 2-step process is a matter of necessity at this stage of development to keep the complexity of the computer within bounds. It has the advantage that each language can be developed independently of the other to be most effective for its purpose.

At the level of the user, there may be a need to develop specialized languages which facilitate programming of different jobs with varying emphasis on arithmetic, logical operations, data manipulation, and input-putput control. At the machine level, where all these jobs come together, the need is clearly for a versatile and relatively unspecialized language. Perhaps the greatest demand on versatility is made by the process of translating from an automatic programming language to machine language. The performance of a computer on translating its own programs is a significant measure of how effective a tool the instruction set really is.

Reference

1. The Essential Types of Operations in an Automatic Computer, **W. L. Van der Poel.** *Nachrichtentechnische Fachberichte*, Fredr. Vieweg & Sohn, Braunschweig, Germany, vol. 4, 1956, p. 144. (Proceedings of a conference on Electronic Digital Computing and Information Processing held at Darmstadt, Germany, Oct. 1955.)

System Design of the Gamma 60

PHILLIPPE DREYFUSS
NONMEMBER AIEE

THE BULL Gamma *60* is a large-scale electronic data-processing system designed for general business use, as well as for engineering calculations.

The Gamma *60* incorporates many new design concepts both in the structure of its information handling devices, and in its novel control unit organization. It uses a resistor diode logic.

The Gamma *60* opens a new area in data processing whereby different functions, or even distinct independent problems can be solved simultaneously on a single machine without any previous planning or programming of the possible occurrence of these parallel operations.

The Gamma *60*, like other data-processing machines, handles different sets of information, and the difference depends upon:

1. The nature of the information; qualitative data, quantative data, or instructions.
2. The origin or destination of the information; input devices, internal processing and storage units, and output devices.

It is well known that no single code is optimally suited to the requirements of these different classes of information. These requirements are: information density, minimum storage requirements, coding efficiency, ease and speed of transcribing input and output codes to and from internal machine codes, high transfer flow, parallel versus serial, number of parallel channels, programming ease and economical storage allocation, and fixed versus variable "word lengths."

In order to remain as close as possible to an ideal code which would optimize

Phillippe Dreyfuss is with the Bull Company, Paris, France.

these often conflicting requirements, different codes are used in the Gamma *60*, depending upon the nature and location of information throughout the system.

Internal data code is used: Quantitative (numerical) data are coded in a 4-bit decimal code; qualitative (alpha-numerical) data are coded in a 6-bit alphanumerical code. The internal instruction code means that the instructions are coded in straight binary code.

As to the internal information length, the information quantum is called a "catena," and it is composed of 24 bits representing either 6 decimal digits, or 4 alphanumerical characters. This quantum must contain a multiple of 4 and 6 bits to represent a whole number of decimal or alphanumeric characters. Twenty-four bits was found to be a good compromise between the minimum 12 bits, which would lead to a too-low transfer flow from a parallel readout core memory, and 36 bits or more, which was judged as too large an information quantum. The catena is to be considered as the equivalent of a character in variable word length machines, but it cannot be called so, as it may contain several characters. It is transferred in series to and from the main memory.

Not wanting to call a "quantum" a word, or a set of characters a letter, (a word is a word, and a quantum is something else), a new word was made, and it was called a "catena." It is an English word and exists in Webster's although it does not in French. Webster's definition of the word catena is, "a connected series;" therefore, a 24-bit information item. The word catena will be used hereafter.

The internal code, therefore, has been defined. Now what are the external data codes? These depend primarily upon the information handling device involved. The Gamma *60* is designed to handle information relevant to any binary coded structure. Thus an 80-column punched card is considered as a 960-bit information item; 12 rows multiplied by 80 columns equals 960 possible punches; is stored as an exact image in 960 magnetic cores of the main memory with 2 card columns occupying one catena.

If necessary for further processing, this card code will be translated into internal code in a later operation. Some of the usual codes are: punched card codes, 80 or 90 columns; International Business Machines Corporation, Bull, or Remington line printer codes; punched paper tape codes, 5 and 8 channels; magnetic tape codes, etc.

A novel approach in logical structure has been developed for the Gamma *60*. The machine can roughly be divided into a main unit, including the magnetic core memory, and a super control unit, composed of a program distributor and a data distributor. There are an unlimited number of elements, each of which constitutes a complete data-handling device with its input-output storage and control units. These elements belong to one of these categories:

1. The arithmetic unit.
2. The logical unit.
3. The comparison unit.
4. The translating unit.
5. The magnetic-drum auxiliary storage unit.
6. The card reader input unit.
7. Tape reader input unit.
8. Card punch output unit.
9. Tape punch output unit.
10. Line printer output unit, etc.

The fundamental common characteristic of these elements is that each one is rectly connected to the main memory by two (some by only one) transfer buses, a distributing bus from the main memory to the elements, and a collecting bus from the elements to the main memory. No direct transfers between elements are possible. However, each element contains its own program and control unit, and can operate completely independently of any other element in operation, once it has been provided with an instruction.

The only common devices shared by all the elements are the main memory and transfer buses. The program distributor handles the super control problem of providing each element automatically and in due time with instructions, and of setting and removing all the necessary interlocks to avoid program interference. The data distributor handles the priority problem of time-slot allocation (one slot equals one catena) to demanding elements on the transfer buses.

Thus the Gamma *60* allows a completely integrated on-line operation resulting from the possibility of running several independent problems at the same moment and on a single line. To illustrate this feature the following problems could be performed simultaneously:

1. A general data-processing problem such as a payroll.
2. A scientific computation such as a matrix inversion or linear programming.
3. One, or several magnetic tapes to printer conversions.
4. One, or several punched cards to magnetic conversions.
5. One, or several punched tapes to magnetic tape conversions.

This would, of course, require an adequate number of elements, magnetic tape units, printers, card and tape readers, and enough magnetic core storage and drum storage to accommodate all of these programs and their corresponding working storage. But, this condition being fulfilled, nevertheless, the different programs would have been planned, written and loaded in the machine independently, and in complete ignorance of one another. As a result of on-line operation a large economy of hardware is achieved, as well as a higher degree of automation; all of the phases of a problem can remain under the control of a single stored program.

The dominant feature of this organization is, however, the over-all saving of time. In effect, the time loss resulting from solving several problems simultaneously will only exceptionally exceed a few per cent for each problem of the time that would be required to solve that problem alone.

As for operations generally handled offline, such as card-to-tape or tape-to-printer conversions, they will always keep pace with the maximum speed of the slow mechanical device, card reader, or printer.

Everthing is quite conventional. There is no high speed; there is nothing extraordinary in technology. The question was raised before by someone as to the possibilities of a symbolic code. The printer allows one to write a symbolic formula code such as: $y = \Sigma X_i(IJK)$-$P(1/2)$.

Now study the main unit and the elements. The following are characteristics of the main memory which is a saturated magnetic core memory. It has a capacity of 32,768 catenae representing 786,432 bits, 196,608 decimal digits or 131,072 alphabetic characters. The access time is 11 microseconds for 1 catena. Every catena is addressable.

The arithmetic unit performs the 4 arithmetic and the comparison operations both on program and on floating decimal-point numbers. The numbers are represented by two catena; the algebraic sign 1 bit, 10 coded decimal digits 40 bits; the decimal exponent (range 0 to 79) 7 bits. The arithmetic unit instruction allows 1 to 3 address operations, that is, 2 to 4 catenae.

A 3-address floating decimal-point addition takes 150 microseconds; a 1-address floating decimal-point addition takes 88 microseconds. The arithmetic unit also comprises four particular index registers for automatic address modifications.

The logical unit performs logical and

straight binary operations and some specialized operations, such as program loop counts, transfers in and out of subroutine, etc. It contains two registers.

The alphanumerical comparison unit performs the comparison of two items stored in the main memory. The length of the items in catenae is determined by the operation code. It can take any value from 1 to 16,384 catenae.

The comparison unit is used to translate codes and edit input and output information. None of the input or output devices are provided with any electromechanical means of editing or selection, and no plugboard or relay networks. All these functions are performed by the translating unit for which special instructions are provided. The line-printer paper skip is itself programmed. The speed and flexibility thus achieved surpasses greatly those obtained with electromechanical means and further increases the integrated on-line character of the Gamma *60*.

The comparison unit instruction allows a 2-address variable-length operation. The same unit handles transfers from one location of the main memory to another. The same instructions, two address and length, are used.

The magnetic drum units are auxiliary storage units exchanging information with the main memory. Their capacity is 32,768 catenae each; the average access time is 11 milliseconds. A magnetic drum unit instruction is of the same 2-address type and length as the comparison unit instruction. The first address is that of the magnetic drum, the second is that of the main memory; the operation code determines the direction of the transfer and length. A single operation can control the transfer of a full magnetic drum unit (32,768 catenae) to the main memory, or vice versa.

The magnetic tape units are auxiliary storage units exchanging information with the main memory. They use one-half inch Mylar tape comprising 8-information-carrying channels. Pulse packing density is 200 bits per inch; tape speed is 50 inches per second leading to a frequency of 10 kc. This frequency corresponds to a rate of 20,000 decimal digits or 13,333 alphabetic characters per second. The tapes may be read in forward and backward directions. A fast rewind is provided.

The card reader units will read 80 column cards at a rate of 300 cards per minute. Three reading stations allow a complete check. The first reading is compared with the second reading; if these differ, the third reading may either agree with one of the first two, or give a third different value. In that case, the card will be directed to a special card stacker.

Punched paper tape may be directly fed into the Gamma *60* through the tape reader units at a rate of 200 characters per second. Any type of tape with from five to eight channels and code may be used.

The card punch units are in fact card reader punch units. They read and punch 80-column cards at a rate of 300 cards per minute. Two card hoppers are provided; one for reading cards already punched, which can then be read by two reading stations before reaching the punch mechanism; the other for eventually feeding blank cards directly into the punch mechanism. Two stackers are provided.

The line printer units are based on the flywheel principle. They will print 300 lines a minute each. The line holds 120 characters; there are 60 characters for each position. The paper skip is controlled by the program. A skip instruction specifies the number of lines to be skipped or a skip to the next form. Printer checking uses the echo principle.

The number of elements that can be connected to the main unit is unlimited. To illustrate the possibilities of simultaneity, the following figures show the number of elements of the different categories that could operate simultaneously if they were to saturate the rate of flow of the transfer buses and main memory:

Comparison unit: 1
Logical unit: 1
Arithmetic units: 2 to 3 (depending upon the operation)
Translating units: 3
Magnetic drum units: 8
Magnetic tape units: 28
Line printer units: 312
Card reader punch units: 472

These figures are the justification of the control organization previously described. It shows the momentous gain that can be achieved in time and hardware with the adoption of on-line data processing in a system which allows simultaneous operations.

It is the author's firm belief that this will be one of the major trends in system development of the data processing machines of the 60's.

This is all that can be told about the Gamma *60* at this time. There is not enough time to go into more detail, but the author wanted to show that it is possible to run a computer data-processing machine with parallel operations, and that it is possible to have simultaneous functions, going along at the same time without any interconnection between the programs of these functions. Of course, it is a great time saver because the over-all loss in time of running, say a 5-tape-to-printer conversion, and running a tape sort at the same time; and running, say, invenory control, or any of ten programs—is not more than 5 %.

It is a question of putting in more units. Of course there is one limitation, the 2.4 megabits of the code storage. One cannot go further than that, and that is the limitation of the amount of simultaneous programming that can be handled by this conception.

Discussion

Chairman Carr: The first question is for Mr. Shaw from Quentin Correll, IBM: "What has the command structure you described been used for to date? What future uses do you see for it?

J. C. Shaw: There are other questions here that are along a very similar vein, so let us get them all in at the same time and I will answer them together.

Chairman Carr: The next question for Mr. Shaw is from L. D. Yarbrough, North American Aviation: "I understand that an information-processing language has been coded for Johnniac. How about other existing computers, (*704*, *1103*, etc.). Will you publish details of this project?"

It seems as if everyone has asked the same question. The next question for Mr. Shaw is from Mrs. Joanna Wood Schot, David Taylor Model Basin: "For what machine is the program you have discussed designed? Or is it designed for an arbitrary machine? What types of data have already been successfully processed using this routine, or is the whole thing indeed hypothetical in spite of your claim of its existence?"

J. C. Shaw: Three of these questions are along the idea: What has the command structure been used for to date? What are the future uses?

We are in the field of simulating human problem-solving behavior. To put it bluntly, we would like to create a machine that thinks.

Last year at Western Joint Computer Conference we presented two papers on the logic theory machine. At the 1955 Western Joint Computer Conference, a paper on the chess machine was presented.

The logic theory machine has done some interesting things. One which I might mention here is that it has discovered a new proof to a theorem in Chapter II of *Principia Mathematica*.

We are only now achieving the chess machine. However, I am looking forward to a paper in this conference by Dr. Bernstein and others having to do with the chess machine that they have coded at IBM. There has been a series of these information processing languages, so that the one I described is really IPL VI. The prior ones, which are essentially the same, were realized on the Johnniac, the Princeton-type computer built by the RAND Corporation.

The people working along the same lines at the Carnegie Institute of Technology have put an information processing language on the *650*, and also on the *704*. So, in the pure interpretive form, IPL's essentially the same as the one I described exist and are being used.

Chairman Carr: The first question for Mr. Dreyfuss is from myself: "Wasn't this differential address device discovered or invented in this country first by Samuel Alexander and others at NBS and the University of Michigan?"

Phillippe Dreyfuss: We were badly informed, and we had to rediscover it.

Chairman Carr: There is a question from the floor: "Are you able to use the instruction location counter as an index register?"

Phillippe Dreyfuss: Yes, any instruction, any current address counter can be used as an index register; but any internal memory location in fact can be used as an index register by substituting the address automatic feature, the automatic counter of the address. This is made necessary by the fact of having simultaneous operations. Let us take one like a drum transfer load, a current address of the drum telling it where the first address is, the first catena to be transferred. This transfer is going to take place only once in every eight cycles; it is not continuous, and it must be stored somewhere. The point which we have reached in the transfer is to know that the next transfer arrives eight cycles later, and it will have to translate it next. You have to have the feature of an automatic counter anyhow. We added another constant or even a variable.

Chairman Carr: I would like to express what I think a lot of people are thinking, and that is that the Bull Company is to be congratulated for very imaginative design. I would also like to ask you how far along is this design?

Phillippe Dreyfuss: Thank you. Well, the machine is being developed, and the first delivery is scheduled for delivery in one year. We have seven systems on order.

There is one other inevitable question that you all have in mind, so before anybody puts it to me, I will give it to you right away. The cost of the system is hard to determine, as it depends on how much and how many units you put in. Let us say that the most basic system would sell for about a million dollars.

Chairman Carr: A question from the floor: "How many people made up the group who designed the machine?"

Phillippe Dreyfuss: Four or five people.

Chairman Carr: Another question from the floor: "When you have several problems going on, how do you know where to start the new ones, how do you know that you have enough core memory?"

Phillippe Dreyfuss: You have to have some kind of core planning. It can be done by the supervisor, or by the machine itself. At some time the machine always counts how much core storage is available. It knows the loaded programs. You generally know how many core storages you are going to use, so the machine can do its own accounting. If you are trying to put in a problem that does not fit, it would just write out, "It does not fit."

As to how you are going to start a problem, there are just the sequences as I have mentioned. There is the cut operation, the sort of branch cut operation; every time we start a new sequence it comes from a branch or cut. A very special cut operation is put in on the console which is the start operation. The difference between this start operation and that of other machines is that you can restart the machine when it is once started by setting a new first address. If you have a new first address that you want, you press the button, and you start a new program. The other one may stop; in fact, I did not have a chance to tell you about all of the cuts. There are eight types of cuts. Also, I did not tell you about internal checking, but the machine has internal checking.

Chairman Carr: Again from the floor: "Do you anticipate trouble-shooting in the equipment as well as trouble-shooting programs?"

Phillippe Dreyfuss: That is why I started to tell you about the checking. Whenever an error occurs, if our checking is correct, this error should be automatically detected. What happens when it is detected? It puts one of the bits in a qualitative catena and it stops the sequence on hand, and it sends us to a diagnostic program. The first thing we have is 24 portions of elements to find out what happened. Two things may have happened: Either there is a possible corrective step taken by the machine itself; or it will stop the reading of the tape through the diagnostic program.

Of course the arithmetic program will just try to find out if it is a random error or not, and repeat the operation, if the operand is still available.

There is another type of cut that does exist. When a unit breaks down, the sequence in question has stopped, in fact just as if you had programmed a branch operation branching over to a diagnostic program. This diagnostic program cannot use the unit that was broken down, and, most important, it should not stop by a cut operation which would liberate the unit taken over by another program which is waiting for it. In this case, unfortunately, I did not have time to explain how the waiting line is established. It is automatically established there and there may be any number up to a hundred programs waiting for a certain program to become available. A chain-like system inside it allows it to repick the program as it arrives, whenever the unit becomes available.

Chairman Carr: From the floor: "Is this a relative diagnostic program that you are talking about?"

Phillippe Dreyfuss: It is a program; you can have as many as you can program.

Chairman Carr: From the floor: "Speaking of programming, have you developed a programming system for this machine, and can you give a brief discussion about it?"

Phillippe Dreyfuss: I wrote out one formula for you, and we have written a compiler which is in the mathematical type. We have more symbols to use on the printer, and a few modifications on the program. This program is now being coded, it is not coded as yet.

Chairman Carr: From the floor: "What are you coding it in, that is, what language?"

Phillippe Dreyfuss: In the machine we have used a 4-level code and it is a pure binary code. The first program interpretive routine that we have is what is called a second-level code, an alphanumeric symbolic code; one to one for each catena. This code that I have here uses a one-to-four-letter code to specify the type of catena. Then the addresses are in decimals. Of course the relative addresses are just alphanumerical codes.

A Direct Read-Out Bistable Circuit and Some Applications of It

IR. H. RODRIQUES DE MIRANDA
NONMEMBER AIEE

I. RUDICH
NONMEMBER AIEE

THE *6977* is a high-vacuum subminiature triode with a fluorescent anode primarily designed for display in transistor computer circuitry. It consists of a single-strand direct-heated filament having long life properties and requires 1 volt and 30 milliamperes (ma) a-c or d-c. Around this single-strand cathode is a cylindrical control grid. The anode is a gridlike structure coated with a *P-15* phosphor optimized for maximum brightness. The light output of the tube is visible from the side as an area approximately the diameter of the tube with a length of 1/2 inch. At zero bias, the plate current is approximately 0.6 ma and the tube can be cut off with from 2 to 4 volts, depending upon whether the filament voltage is a-c or d-c.

This tube, being a vacuum triode, can without objections be used as an active element with the advantage of combining two functions in this one device. As this tube will be mainly used for indicating the position of a bistable circuit, it is worth while considering this tube as an active part of this circuit. Fig. 1(A) and (1)B shows the average characteristics of the *6977* and its construction.

The use of complementary transistors in bistable circuits is well known. This principle of direct coupling of a p-n-p and n-p-n transistor providing a regenerative loop can be applied as well to the combination p-n-p vacuum tube.

A vacuum triode can very well be compared with a n-p-n transistor. However, there are some important differences to consider:

1. Operation can be in the negative grid voltage region.
2. When overdriving the tube, the anode-cathode voltage remains considerable.
3. The grid-input impedance stays very high in the operating region.

While points 1 and 2 are only facts which are not particularly advantageous or disadvantageous, the high-input impedance of the tube is definitely an appreciable feature. This gives the possibility of triggering the circuit with very little power.

The introduction of the *6977* only as an indicator tube for transistor flip-flops already means a big saving in components. It has made the amplifier stage between the indicator and the bistable circuit obsolete. The circuit to be described here is a further attempt in this direction.

In Fig. 2 there is a positive feedback from anode via the transistor to the grid of the tube.

"Off" Condition

Assume that for $V_g < -4$ volts, the anode current is zero. This means that there is no base current and no collector current (assuming the transistor can stand the high collector voltage). Nevertheless, because of the small currents in the loop $R_1R_2R_3$, the grid voltage stays < -4 volts. This situation therefore is stable. It is referred to as the off condition.

Transient I

Now consider that in some way, i.e., by externally applying a positive going pulse to the grid, the tube starts to conduct. This means that there is base current, and consequently, collector current in the transistor. The potential of A grows in the positive direction and because of the voltage division $R_1 - R_2$ point B also rises. This tends to increase the anode current more. This is (or can be) an unstable situation. (For the stability criteria see circuit analysis.)

"On" Condition

The current increases until the voltage drop over R_3 approaches 50 volts. The collector-emitter voltage becomes nearly zero and the transistor ceases to be an active element. This again is a stable situation. R_1 and R_2 must be chosen so that the level of B is, in this condition, about zero volt. R_3 can be chosen so that the transistor just reaches saturation. The analysis will go over these considerations in more detail.

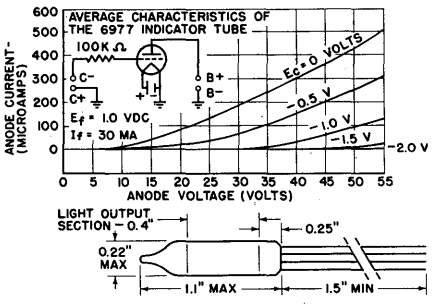


Fig. 1(A). Construction of the 6977

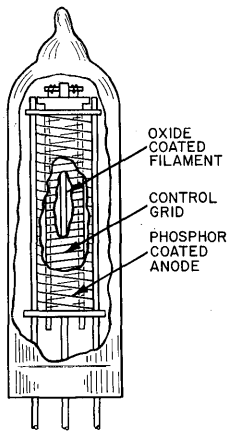


Fig. 1 (B). The 6977 is a high-vacuum subminiature indicator triode with a type P15 phosphor-coated anode

Transient II

Now, if by some means the anode current decreases somewhat, e.g., because a negative going pulse is applied to the grid, the base and therefore the collector current decrease. This makes point B negative and therefore decreases the anode current even more. This situation can be made unstable. Finally, the circuit terminates in the off-condition again.

Most fast-switching transistors cannot withstand such a high collector-emitter voltage as 50 volts. As for proper operation of the indicator tube, this supply voltage is needed and, as this voltage will come across the transistor in the off condition, a possible change of the circuit must be considered in order to protect the transistor. One of the possible solutions to this problem is given in Fig. 3(A).

In this circuit, as well as in Fig. 1, assume that R_1 and $R_2 >> R_3$ and R_4.

By inserting R_4 across the transistor, the collector-emitter voltage in the off condition becomes

$$\frac{R_4}{R_3+R_4} \times V_{\text{battery}}$$

For instance, if $R_3 = R_4$, the maximum collector voltage will become 25 volts.

In the on condition, R_4 has no influence upon the operation because the transistor then represents a very small impedance. In this kind of circuit, the transistor acts merely as a switch which is

IR. H. RODRIQUES DE MIRANDA is with Philips Research Laboratories, Eindhoven, Holland.

I. RUDICH is with Amperex Electronic Corporation, Hicksville, N. Y.

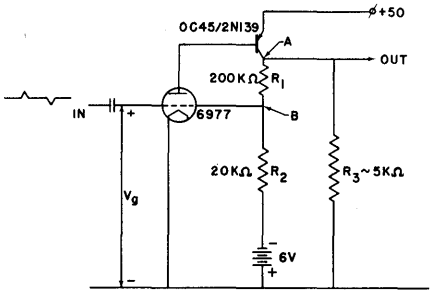


Fig. 2. Description of basic circuit

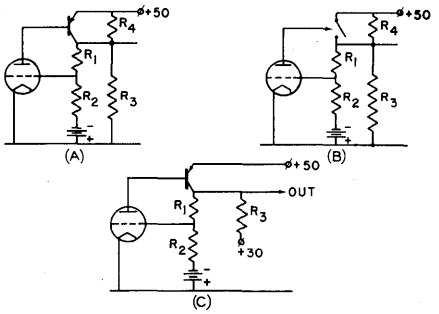


Fig. 3. Protection against high collector voltage

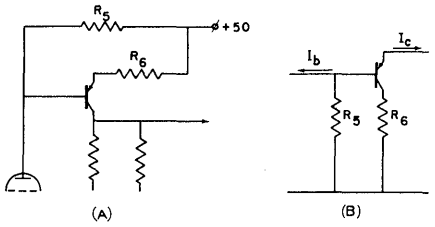


Fig. 4. Stabilization against spread and variation of β

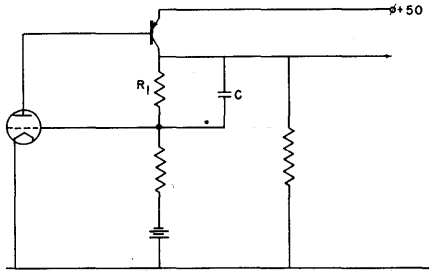


Fig. 5. Improving speed and sensitivity

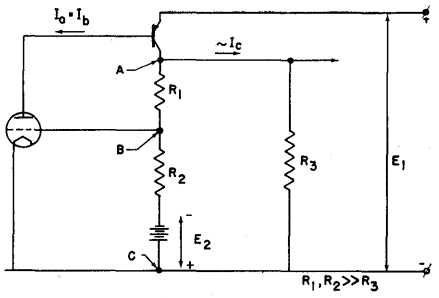


Fig. 6. Analysis of basic circuit

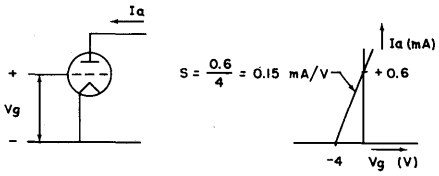


Fig. 7. Linear approximations

controlled by the base current (anode current of *6977*); see Fig. 3(B). This protection must be compensated for with a smaller output voltage and a higher battery power consumption.

In the circuit of Fig. 3(C), which is another possible solution, the power consumption is not increased but decreased. In this circuit, however, an extra battery-voltage level is needed. (This protection may not be necessary with some of the newer diffused-base transistors because of their higher breakdown voltages.)

Of course, the circuit could be designed so that even for the lowest expected value of β (base collector-current multiplication) the transistor saturates ($V_{ec} \approx 0V$) in the on condition. This means, however, that for higher values of β, the transistor is overdriven very badly, which gives higher dissipation and, what is even more important, lower switching speed.

There is, however, a very simple method of stabilizing β at the cost of some loop amplification; see Figs. 4(A) and (B).

If R_6 is chosen so that it is several times the emitter resistance and R_5 is several times smaller than β times R_6 (which is the approximate base input impedance), the effective β of the circuit $\bar{\beta} = I_c/I_b$ will approximate R_5/R_6.

Reasonable values for R_5 and R_6 are 600 and 40 ohms respectively. This stabilization is set the expense of a somewhat reduced power gain.

In general

$$\frac{1}{\bar{\beta}} = \frac{1}{\beta} + \frac{R_6 + r_e}{R_5}$$

in which

$\bar{\beta}$ = effective current gain of the circuit of Fig. 4(B)
β = current gain of transistor
r_e = internal emitter resistance

The stabilization of the current gain is defined as

$$S_\beta = \frac{d\beta}{d\bar{\beta}}$$

$$\bar{\beta}^{-1} = \beta^{-1} + \frac{R_b + r_e}{R_5}$$

Differentiated to $\bar{\beta}$ gives

$$-\bar{\beta}^{-2} = -\beta^{-2}\frac{d\beta}{d\bar{\beta}}$$

or

$$\frac{d\beta}{d\bar{\beta}} = \left(\frac{\beta}{\bar{\beta}}\right)^2$$

If, for instance, the effective β is lowered by a factor 4, a stabilization of 16 results.

$$S_\beta = \frac{d\beta}{d\bar{\beta}} = \left(\frac{\beta}{\bar{\beta}}\right)^2 = \left(1 + \beta\frac{R_6 + r_e}{R_5}\right)^2$$

For example, if

$\beta = 60$
$R_6 = 40\Omega$
$r_e = 10\Omega$
$R_5 = 600$

then

$$S_\beta = \left(1 + 60\,\frac{50}{600}\right)^2 = 36$$

and

$\beta/\bar{\beta} = 6$ or $\bar{\beta} = 10$

As another example, if

$\beta = 60$
$R_6 = 40\Omega$
$r_e = 10\Omega$
$R_5 = 3{,}000\Omega$

then

$$S_\beta = \left(1 + 60\,\frac{50}{3{,}000}\right)^2 = 4$$

and

$\beta/\bar{\beta} = 2$ or $\bar{\beta} = 30$

In Fig. 5 a method is given to improve the speed and sensitivity. By shunting the resistance R_1 with a capacitor, a much greater positive feedback is obtained at the beginning of the transient, which will make the switchover much faster.

The R_1C time should be much smaller than the output pulse time.

The analysis of the circuit in Fig. 6 is given in the following.

Assumptions

The approximate electrical characteristics for the indicator tube *6977* are as follows:

heater voltage = $V_f = 1$ volt
heater current = $I_f = 30$ ma
anode voltage = $V_a = 50$ volts
maximum light output at $V_g = 0$ volt ($I_a = 0.6$ ma)
zero light output at $V_g = -4$ volts (actually 3 volts or less) (one side of filament tied to ground)

For the analysis, the following assumptions are made:

1. In this first-order approximation neglect

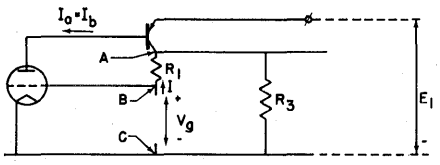


Fig. 8. Circuit analysis

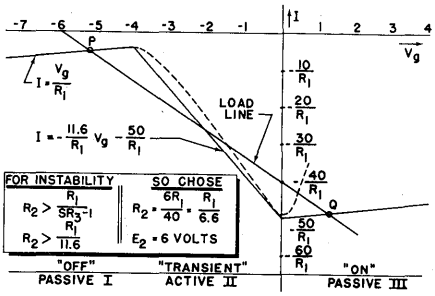


Fig. 9. Input characteristic showing negative resistance

the influence of voltage applied to the filament.

2. Assume a linear relation between grid voltage and anode current. A constant slope S is used throughout the working region of the tube; see Fig. 7.

3. Neglect the grid current in first approximation.

4. Assume β of the transistor is constant.

5. Assume the collector breakdown voltage is higher than E_1.

ANALYSIS

One must now determine two battery voltages, E_1 and E_2, and values of three resistances, R_1, R_2, and R_3, to obtain bistable action of the circuit. (See Fig. 8.)

The value of E_1 is given by the tube requirement $E_1 = 50$ volts. The value of R_3 is determined by the fact that in the on condition the transistor should be near saturation. So, $I_c R_3 \cong E_1$, but $I_c = \beta I_a$,

thus $R_3 \cong \frac{E_1}{\beta I_a}$

With

$\beta = 15$
$E_1 = 50$ volts
$I_a = 0.6$ ma
$R_3 = 5.6$ kilohms

Now there is the somewhat more complicated matter of choosing a right combination of R_1, R_2, and E_2. This choice determines the stability and triggering sensitivity. That the voltage V_{CB} should be 4 volts for off and zero volt for the on condition is only one of the requirements. Of course, R_1 and R_2 are chosen as high as possible because this will result in a high-input impedance level which makes the triggering power low.

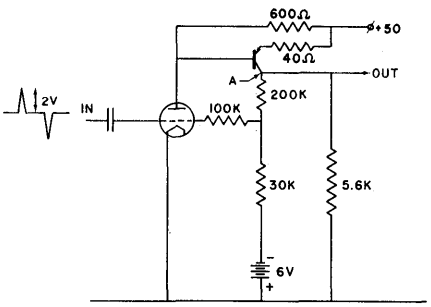


Fig. 10. Calculated basic circuit

To determine R_1, R_2, and E_2 take E_2 and R_2 out of the circuit. Looking at the terminals $B-C$ (Fig. 8) one sees negative resistance region limited on both sides by passive, and therefore, positive resistance regions.

In Fig. 7 the positive directions of V_g and I were chosen so that if an increase of I results from an increase of V_g, a passive element exist. If, however, I decreases if V decreases, an active or negative resistance element exists.

Consider how I and V_g are related.

The anode current is

$$I_a = S(V_g + V_0)$$

where $V_0 = 4$ volts.

The collector current is

$$I_c = \beta I a = \beta S(V_g + V_0)$$

The voltage across R_3 is

$$V_{AC} = R_3 1_c = \beta S R_3 (V_g + V_0) \quad \text{(assuming } R_1 >> R_3\text{)}$$

$$I = \frac{V_g - V_{AC}}{R_1}$$

$$I = \frac{-(\beta S R_3 - 1)}{R_1} V_g - \frac{\beta S R_3 V_0}{R_1} \tag{1}$$

This equation indicates that a negative resistance can be obtained if and only if

$$\beta S R_3 > 1 \tag{2}$$

where

$\beta = 15$
$S = 0.15$ ma/volt
$R_3 = 5.6$ kilohms $\beta S R_3 = 12.6 > 1$

If the known values are substituted in equation 1 the following results

$$I = \frac{-11.6}{R_1} V_g - \frac{50}{R_1} \tag{3}$$

In Fig. 9 this relation of V_g and I plotted. The passive regions on both sides limiting the active or transient domain are interesting.

REGION I: OFF, PASSIVE

If the grid voltage becomes more negative than $-V_0$ (-4 volts), anode and collector current stops. The differential resistance of $B-C$ now becomes R_1 (substitute in equation 1 $S=0$).

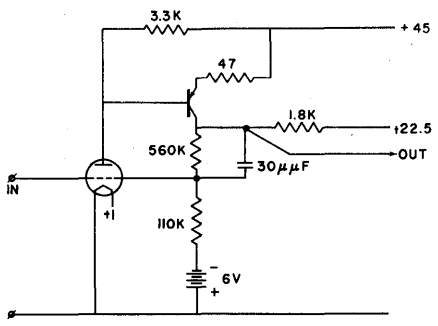


Fig. 11. Practical design

REGION II: TRANSIENT, ACTIVE

This is our calculated negative resistance. The dotted line gives the better approximation.

REGION III: ON, PASSIVE

If the voltage becomes greater than 0 volt, the transistor saturates (this also can occur by limiting the grid voltage swing because of the grid current or because of external diode action) and consequently stops being an active element. Now again there is a positive differential resistance of the value R_1.

If a load line is drawn, one sees that there are two stable situations (P=off and Q=on).

It was not easy to foresee that a stable on situation would result, which implies a positive grid voltage of 1.5 volts. The best solution to this dilemma is by inserting a series grid resistance of 100 kilohms as is in fact ordered by the tube manufacturer. It is not important that point B rises to a positive voltage, the grid stays very nearly at zero volt because of the grid current which starts to flow. Another way would be to choose a slightly higher value for R_3 in order to saturate the transistor at $V_g \approx -1$ volt.

Now back to the determination of R_1, R_2, and E_2. From the load line, Fig. 9, it is learned that:

1. Bistable action can only be achieved if the load line intersects the S characteristics at three points. This means that the instability criterion is

$$R_2 > \frac{R_1}{SR_3 - 1} \tag{4}$$

or in the case

$$R_2 > \frac{R_1}{11.6} \tag{5}$$

2. Looking at the graph it seems to be a reasonable compromise between stability and triggering sensitivity if the following is chosen

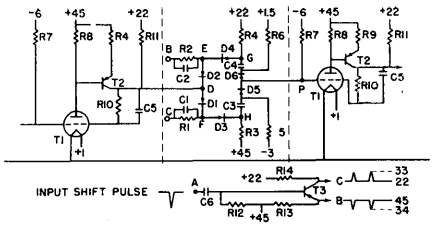


Fig. 12. Application of the circuit in a shift register

$$R_2=\frac{6R_1}{40}=\frac{R_1}{6.6}$$

and

$E_2=6$ volts

3. Now there is still some freedom in choosing the absolute value of R_2, and consequently R_1. It is chosen as high as possible, limited by the grid-cathode capacitance and switching speed. A value of 30 kilohms seems reasonable.

$$R_2=30 \text{ kilohms}$$
$$R_1=6.6\times R_2 \rightarrow R_1=200 \text{ kilohms}$$
$$E_2=6 \text{ volts}$$

4. The triggering sensitivity now is 2 volts toward both sides. Finally the basic circuit now looks like Fig. 10. Though the circuit of Fig. 10 operated satisfactorily, the actual circuit used was a little different, see Fig. 11.

As mentioned before, a collector voltage of 50 volts is above the ratings of the *OC45* which was used for its speed. As described previously (Fig. 4) R_3 now is connected to a positive voltage. Further it is seen that the feedback resistance of 560 kilohms is bypassed by a 30-micro-microfarad ($\mu\mu$f) capacitor to improve the speed. A rise time of 0.26 microsecond (μsec) and a fall time of 0.86 μsec were found.

The impedance level of the feedback voltage divider is chosen somewhat higher in the final circuit. By doing this one could avoid the 100-kilohm grid series resistance of Fig. 10 and still have a grid current limiting of the on position.

The purpose of the coupling circuit between the dashed lines is to provide point P with a positive pulse at the time a shift pulse appears at A, if the previous stage was on, and a negative pulse if the previous stage was off.

$r_1,r_2,r_3,r_4=56$ kilohms
$r_5,r_6=220$ kilohms
$r_7=100$ kilohms
$r_8=3.3$ kilohms
$r_9=47$ kilohms
$r_{10}=560$ kilohms
$r_{11}=1.8$ kilohm
$r_{12}=15$ kilohms
$r_{13},r_{14}=1$ kilohm
$c_1,c_2=47$ $\mu\mu$f
$c_3,c_4=100$ $\mu\mu$f
$c_5=30$ $\mu\mu$f
$c_6=0.001$ μf
$T_1=6977$
$T_2=OC\,45$
$T_3=OC\,45$
$d_{123456}=OA\ 86/1N480$

As is shown later the capacitors c_1 and c_2 act as a temporary memory during the shift pulse. During this pulse the previous stage has to be decoupled and the polarity of the output pulse at P should only be determined by the state of the previous stage before shifting.

The way the pulses take in this coupling circuit is now followed. There are three inputs B, C, and D. Points B and C receive positive and negative going pulses respectively from an external supply which serves the whole n-stage register. Point D is connected with the output of the previous stage. Therefore, point D can have two voltage levels, 45 volts (on condition) and 22 volts (off condition).

Assume that $V_D=45$ volts. The following conditions are considered:

1. Condition between two shift pulses.
2. Condition during shift pulse.

With reference to condition 1, the following applies:

$V_D=45$
$V_B=22$

d_2 is reversed-biased; c_1 is not charged.

$V_c=22$

d_1 is forward-biased and c_2 charges up to $\sim 45-22=23$ volts. d_4 is reversed-biased. d_3 is not carrying appreciable current because $V_F \sim V_H \sim 45$.

$V_H=45$
$V_G=22$

During shift pulse, the following applies:

$V_B=34$

Because of the voltage inertion of c_1, V_E lowers to 34. d_2 and d_4 are still nonconducting.

$V_C=33$

Because of the voltage inertion of c_2, V_F rises ~ 9 volts.

$V_F \approx 45+9=54$

d_3 is, therefore, during the shift, forward-biased and d_1 is reversed-biased.

$V_H \approx 54$

Therefore the positive going pulse is transferred through the lower branch and passes via c_4 and d_5 to the input of the second bistable circuit.

This circuit receives a positive pulse which switches it on if it was off and leaves it on if it was already on.

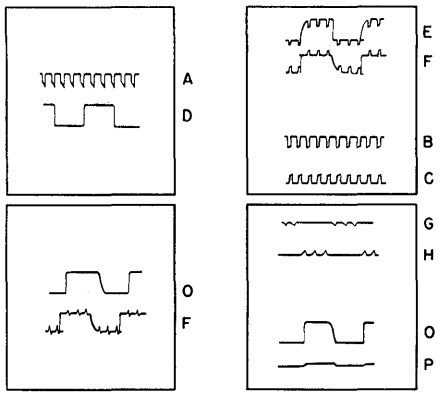


Fig. 13. Operation of shift register as shown by waveshapes

Note that a change of voltage on D, because of a possible change of state of the previous stage, does not influence the circuit during the shift pulse. Diodes d_1 and d_2 are both reversed-biased.

Assume that $V_D=22$ volts. A similar reasoning as in the 45-volt case can be given for the case in which the previous stage is off, and consequently the potential of D is 22 volts.

In this case a negative going pulse on P is found, which takes care that after the shift pulse the second stage is in the off condition.

To summarize, the voltage levels at the different points are given in Table I, which follows.

Table I. Voltage Levels

	Between Shift Pulse S		During Shift Pulse	
Points	$V_D=45$	$V_D=22$	$V_D=45$	$V_D=22$
D	45	22	45	22
B	45	45	34	34
C	22	22	33	33
E	45	22	34	11
F	45	22	56	33
G	22	22	22	11
H	45	45	56	45
P	0/−2	0/−2	>0	<−2

The purpose of c_3 and c_4 obviously is to separate the d-c level of points G and H from the pulse signal.

The diodes d_5 and d_6 in combination with r_5 and r_6 and the battery voltages -3 and $+1.5$ take care that the grid voltage of the *6977* can change between -3 and $+1.5$, without being loaded with the preceding circuit.

Testing the Circuit

For the test the input voltage at D was simulated with a square-wave generator with a voltage swing of 22 to 45. This generator was synchronized to 1/6 of the

shift-pulse frequency applied to A. The maximum shift-pulse frequency was 0.3 megacycles per second.

In the following figures the pulse patterns on the indicated points of the circuit of Fig. 12 are seen. The shift frequency was 100 kilocycles per second and the block frequency at point D was 16.7 kilocycles per second.

Conclusions

A direct read-out bistable circuit using one transistor and one high-vacuum indicator triode is described. The advantages of this circuit are:

1. Very low triggering power (~30 microwatts) because of high-input impedance (~100 kilohms).
2. High-output power available (~200 megawatts) on low-impedance level (~2 kilohms).
3. Simple circuit with fewer components than the conventional circuit.

Further, this paper contains a description of a coupling circuit which, together with a bistable circuit, provides one stage of a shift register.

Flow Gating

W. J. POPPELBAUM
NONMEMBER AIEE

Synopsis: The standard arrangement used to transfer information from one flip-flop to another requires severing elements in the connection between them. The number of diodes and triodes in these "gates" is often comparable to that in the flip-flop. This paper describes a scheme in which the transfer is effected by a simple diode going from the sending to the receiving flip-flop and by changing the supply voltage of one of the two circuits. The method appears to be attractive for small memories composed of flip-flops. It is also shown that all inputs and outputs can be tied together and that information can then be exchanged between two partially selected units.

A SIMPLE problem occurring in a computer is to transfer a zero or a one from one place to another in a selective fashion. The problem may be presented by discussing the transmission of information from one Eccles-Jordan flip-flop to another. It will be assumed that nonoverlapping voltage bands represent the zero and one signals, the bands being caused by parameter drift. An Eccles-Jordan flip-flop then has the following abstract properties (see Fig. 1): There are two low impedance outputs and two high-impedance trigger points. Points with the same number are in phase; points with a different number are out of phase. (In general, especially in so-called last-moving-point flip-flops, there may be a time lag between a trigger point and the corresponding "in phase output.") "Out 0" in the one state corresponds to the zero state of the flip-flop. "Out 1" in the one state corresponds to the one state of the flip-flop. Transmission of information between two flip-flops can be accomplished by connecting the outputs of the first to the inputs of the second by means of switches. The direction of flow of information will be determined by the asymmetry of the impedances; the lower impedance drives the higher impedance. This does not mean that the voltages in the connecting wires lie necessarily in the zero or one bands while the flip-flops are tied together. It does mean, however, that once the switches are opened, flip-flop no. 1 has copied the state of flip-flop no. 2 independently of the order in which the switches were opened.

The important point to note is that a gate in the form of two switches severs the connections which transmit information, thus allowing the triggered circuit to seek its levels inside the permitted bands. This severing action can be achieved by adding two diodes in front of trigger point 0 and trigger point 1. In Fig. 2 the situation is indicated for the case of positive logic (one signal voltages greater than zero signal voltages). If the points marked "in" are kept at the zero level, the trigger points are effectively disconnected, except perhaps for some small currents due to the difference in the applied zero voltage and the natural zero voltage of a trigger point. In order to allow the trigger points to seek their own level, the two inputs can be held at the most negative zero voltage. This will be termed a floating output of the diodes.

Note that one can also pull the trigger points down by using diodes in the opposite direction. Then, the floating output would be caused by the most positive one voltage. The discussion will, however, be limited to the first case.

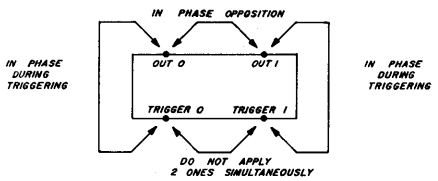


Fig. 1. Abstract representation of an Eccles-Jordan flip-flop

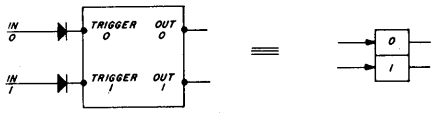


Fig. 2. Make-up of an Eccles-Jordan flip-flop

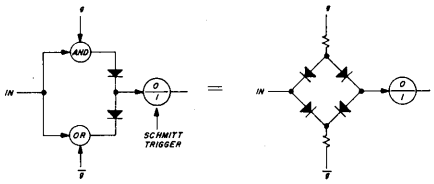


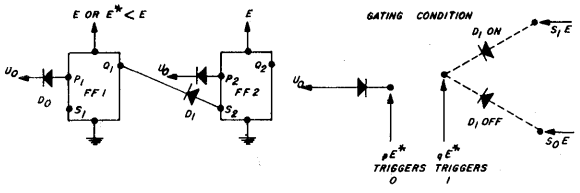
Fig. 3. Double gating a Schmitt trigger-type flip-flop

Transmitting information between two Eccles-Jordan flip-flops with input diodes now only necessitates the use of two "and" circuits with a sufficiently low output impedance. To inhibit the flow of information, a zero signal is injected into the second inputs of the "and's". This double-gating system can be simplified by setting the first flip-flop to the standard zero state by an initial clearing signal which is turned off before a single "and" (connected between the one sides) receives the gating one-signal which causes the conditional transfer. This clearing and gating is naturally slower than double gating since it essentially involves two distinct operations. It should be noted that in both these gating systems the trigger point is either left as it is (floating or zero input) or pushed up.

In flip-flops of the nonsymmetric variety, like a Schmitt trigger (operationally equivalent to an Eccles-Jordan with only two in-phase points accessi-

W. J. POPPELBAUM is with the University of Illinois, Urbana, Ill.

The author wishes to thank Mr. Neil Wiseman and Mr. Sylvian Ray of the Digital Computer Laboratory for their help with calculations and experiments. This work was supported in part by the Office of Naval Research and the Atomic Energy Commission under AEC Contract AT(11-1)-415.



ALL TRANSISTORS ARE W.E. GF45011
ALL DIODES ARE HUGHES 6A4-100
ALL RESISTORS ARE ±2%, 1/2 W
ALL POWER SUPPLIES ARE ±3%
ALL ALPHAS ≥ 0.98

MAX. INPUT CURRENT: 2.75 m.a.
MIN. OUTPUT CURRENT: 3.75 m.a.
OUTPUT VOLTAGE: −10 V, −20 V
SUPPLY VOLTAGES: −20 V, −45 V
SETTING TIME: < 50 m μ. sec.

Fig. 4 (left). The flow-gating principle. Vertical distances indicate potential above ground

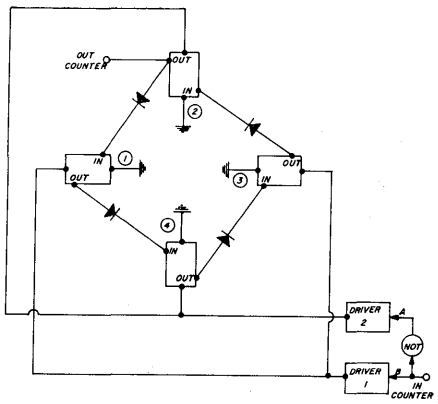


Fig. 6 (right). A flow-gating binary counter stage

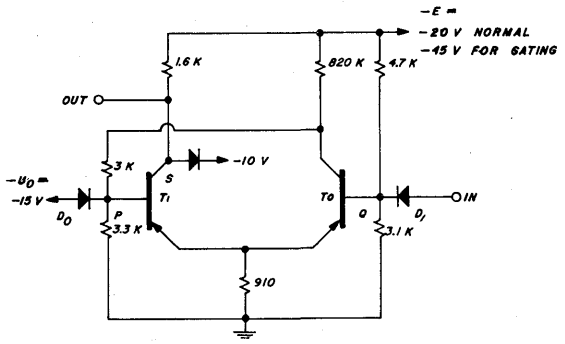


Fig. 5 (left). Layout of a flow-gating flip-flop

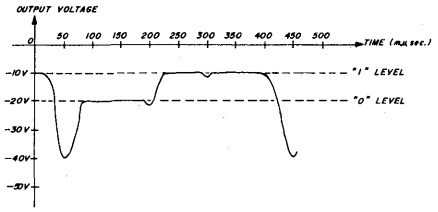


Fig. 7 (right). Output of a flow-gating flip-flop in a binary counter. Input 5-mc sine-wave

ble!), the double-gating problem is somewhat harder to solve. This can be seen in Fig. 3. A single trigger point has to be pushed up (to gate a one), or to be pulled down (to gate a zero) or finally to be disconnected from the incoming signal in order for the device to stay in its last state. A single diode is only able to transmit information in one direction, therefore, it is evident that going into the trigger point two diodes must be used. It turns out that one of the paths then necessitates an "or" gate and a second input which is the complement of the gating signal. Furthermore, inspection shows that there is no essential difference (even topologically) between the latter arrangement and a bridge modulator type of gate as indicated in the right-hand side of Fig. 3.

Although g and $\bar{g}$ do not have to be in exact phase opposition, the production of push-pull gating signals is rather cumbersome, especially when control applications, with only a small number of flip-flops connected to any given gating bus, are considered. The bridge modulator gating system can be and has been, used for registers. Here, the fact that two gates can be connected "upside-down" can be used to provide mutually exclusive paths.

The gating problem for the Schmitt trigger type of flip-flop is, of course, quite easily solved in case clearing precedes the gating. One "and" circuit and one diode are sufficient to set the circuit to the one state once it has been cleared to zero. The only objection to this method is, again, its comparatively slower speed since gating-in again involves two operations. The next section describes a gating system which is essentially a clearing and gating system, but in which the two operations occur simultaneously. Furthermore, no special clearing signal has to be provided from the outside.

The Flow-Gating Principle

The main idea in flow-gating is to vary the potentials of two flip-flops in such a way that "transfer diodes" connected between the flip-flops are conditionally conducting, depending on the state of the sending flip-flop when these potentials are made unequal. In the normal (equal) potential condition the transfer diodes are cut off and produce the severing action discussed in the last section. The word "flow-gating" has been chosen to characterize these systems, in which information flows up or down a potential gradient established between bistable elements.

It will be shown that the idea can be applied to bistable elements of a very general class. Take any d-c bistable circuit having two trigger points P and Q such that a sufficiently positive voltage applied to P triggers it into the zero state while a sufficiently positive voltage applied to Q produces the one state. Also suppose that there is a (low impedance) output S in phase with Q (the out-of-phase output could be discussed too). It is not necessary, or even desirable, that the voltage swings at P, Q, and S be the same. It will even be assumed that (under all tolerance conditions) Q and P have non-overlapping swings. Suppose, more specifically, that the voltages q and s at points Q and S have the property that, (1) and (0) designating the flip-flop state,

$$q(1)>q(0)>s(1)>s(0) \tag{1}$$

Consider the supply voltages of the circuit. Rename potentials in such a way that the lowest supply voltage is called ground, and obtain all other voltages from dividers between this new ground and the highest supply voltage E. This will not change the operation of the circuit. In particular, the circuit is still going to be bistable, and the three points of interest, P, Q, and S, will each exhibit two voltages $p(1)$, $p(0)$, $q(1)$, $q(0)$ and $s(1)$, $s(0)$, depending on the state. In each one of the states, however, all voltages are going to be proportional to E because of the d-c stability assumption. In other words,

$$\begin{aligned} p(1)&=p_1E \qquad p(0)=p_0E \\ q(1)&=q_1E \qquad q(0)=q_0E \\ s(1)&=s_1E \qquad s(0)=s_0E \end{aligned} \tag{2}$$

where $p_1p_0\ldots$etc. are constants. Equation 1 now simply becomes

$$q_1>q_0>s_1>s_0 \tag{3}$$

Under slightly idealized circumstances (negligible hysteresis, etc.) the critical trigger voltages to be applied to P and Q to trigger either a zero or a one are:

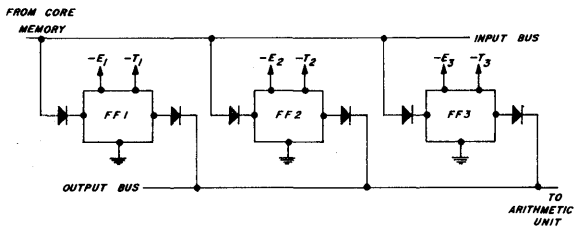


Fig. 8. A flow-gating buffer register memory

$$\frac{p_0+p_1}{2}E=pE \text{ (say)}$$

$$\frac{q_0+q_1}{2}E=qE \text{ (say)} \qquad (4)$$

To simplify the discussion, it will also be assumed that the output impedance is very low, i.e., s_0 and s_1, will be assumed independent of the load.

Now, connect two flip-flops fitting this description as in Fig. 4, i.e., connect Q_1 and S_2 through a transfer diode D_1, and connect P_1 through a diode D_0 to a fixed potential u_0. The value of u_0 will be determined later. It is evident in view of equation 3 that flip-flop no. 1 and flip-flop no. 2 are quite independent as long as their supply voltages are the same. D_1 will not be conducting for any combination of states. This corresponds to what was called a floating output in the preceding section.

To gate information from flip-flop no. 2 to flip-flop no. 1, E is lowered to a value E^*, such that the arithmetic mean of the two possible potentials of S (for E) becomes equal to the trigger potential of Q (for E^*). E^* is given by

$$qE^*=\frac{s_1+s_0}{2}E \qquad (5)$$

Furthermore, u_0 is chosen to be slightly larger than the trigger potential of P (for E), i.e.,

$$u_0>pE^*=\frac{p}{q}\,\frac{s_1+s_0}{2}\,E \qquad (6)$$

It is easily seen that equations 3, 4, 5, and 6, together with the connections of Fig. 4, guarantee a complete transfer of information between the flip-flops. For if flip-flop no. 2 is in the one state, $s_1E>qE^*$ and D_1 conducts, thus setting flip-flop no. 1 to the one state. If, however, flip-flop no. 2 is in the zero state ($s_0E<qE^*$), diode D_1 is ineffective (although it may conduct slightly), and diode D_0 is going to set flip-flop no. 1 to the zero state. There may, of course, arise the question, "Why is setting through D_1 possible while D_0 counteracts?." The answer is simply that it is always possible to arrange for s_1E to override u_0 by providing a large enough swing at the output.

Once the information has been received, the supply voltage is made to rise back to E; the state impressed is "trapped" in the process. That the circuit really cannot change its state as the E^* to E change occurs can be seen as follows. If a zero has been stored, D_1 is off and D_0 is on (see Fig. 4). As the supply voltage increases, Q and P become more positive. D_1 certainly cannot be turned on again. A similar argument holds when a one has been stored.

It is now clear why flow-gating is a sort of clearing and gating system in which both the clearing operation and the "and"-gate function are performed by modifying the potential of the whole circuit. Practically, this means that a fairly large current and a large voltage swing have to be provided by the gate drivers. If it is desired to obtain easier driver conditions, in particular if the gate-in signal has to be furnished by a flow-gating flip-flop, a separate driver transistor can be added, having the flip-flop between its collector and ground and a load resistor of appropriate size.

Practical Example of a Flow-Gating Flip-flop and Its Use

Consider the circuit of Fig. 5, giving what is probably the simplest flow-gating flip-flop, i.e. a modified Schmitt trigger. The trigger points P and Q are the bases of the two transistors, and the output S is the collector of the emitter follower T_1. Taking the output from this point does not impair the left-hand transistor's function in the flip-flop. Note that S is in phase with Q as was assumed in the last section. There is, however, one slight difference. For practical purposes p-n-p transistors were used, and this necessitates changing the sign of all potentials with respect to ground. E becomes $-E$, u_0 becomes $-u_0$, etc. The general theory is, of course, still valid after this change.

The circuit values are chosen in such a way that under the worst drift conditions, i.e., a 2% drift of resistor values, a 3% drift of voltage supplies and a transistor α anywhere between 0.98 and 1.00 (which means using selected *GF-45011* transistors), the circuit still works satisfactorily. In order to verify this, the exact circuit equations and inequalities were analyzed by *Illiac*, the electronic computer at the University of Illinois. The base-emitter drops were stored in table form for fixed values of the emitter current, and linear interpolation was used to obtain intermediate values. The analysis was preceded by an optimizing process in which all equations were linearized in the neighborhood of a given set of parameters. After several steps a near optimum solution was obtained and used as the basis for the analysis programs.

The figure gives all pertinent information, like input and output currents and output and supply voltages. A few indications about the "setting time" might be useful. The most reliable characteristic of a circuit is its operation time, which is obtained by connecting in series n similar circuits and measuring the propagation time of a given state from one end to the other (e.g., by connecting the circuits in cascade and observing the resultant oscillation). A similar procedure is difficult to realize for flow-gating, and it is more realistic actually to swing the supply voltages up and down for each gating operation. One way of doing this is indicated in Fig. 6 which shows the use of flow-gating in a binary counter stage.

The counter works as follows: Suppose that when there is a one (positive logic) at the input to the first driver, flip-flops no. 1 and 3 are in their normal, more positive state i.e., that they cannot receive information. Due to the "not" circuit, flip-flops no. 2 and 4 are then in the gating, more negative state, and information will be gated from 1 into 4 and from 3 into 2. When "in" now goes back to a zero, the contents of 2 and 4 will be sent to 1 and 3, respectively. If the initial pattern 0011 is put into the four flip-flops, cyclic shifting will be observed. Looking at one of the flip-flops (out), square pulses, the period of which is twice that of the incoming symmetric square wave are seen, i.e., there is a frequency divider. Of course, the device is asynchronous if alternate non-overlapping gating signals are provided by the two drivers.

Fig. 7 shows the output of a flip-flop when the input is a 5-megacycle sine wave. Satisfactory operation can be obtained at nearly twice that frequency. This shows that the setting time of a flip-flop is less than 50 millimicroseconds. Note that the normal output levels, i.e., -10 volts and -20 volts are overswung during the gating-in of the zero state. This does not interfere with the transmission of information to another stage since it simply cuts off the transfer diode by a greater margin.

It is interesting to count the number of

elements involved in Fig. 6 to the left of points A and B. This part corresponds to a "true-toggle-false-toggle" type of counter with its associated four gating "and's" and two decoding "and's" commonly used in asynchronous computers. Defining complexity by C=number of transistors + 1/2 number of diodes, the result is typically

C classical system = 25 − 31
C flow-gating = 16

Combination of Flow-Gating with Collector-Gating

Consider the Schmitt trigger of Fig. 5, and suppose that R_6 is returned to a separate voltage $-T$. It is then evident that as long as $-T$ is kept negative enough, no information can be transmitted out of flip-flop even if the receiving flip-flop is in its (negative) gating state. The transfer diode can never conduct. This means that the outputs of n flow-gating flip-flops can be tied to a common output bus and that this bus will carry the information of that one of the flip-flops which has $-T$ in the "sending range." Here the collector-supply voltage of the emitter-follower is therefore used to provide the "gate-out" signal. This will be called collector-gating. It should be noted that this is typically a gating method on the sending side.

Similarly, the inputs of n flip-flops can be tied to a common input bus and only that one which has its $-E$ made negative enough to gate-in will receive the information present on the bus. This flow-gating is typically a gating method on the receiving side. The common-input-bus and common-output-bus scheme is particularly attractive for a buffer memory connected between a slow core memory and a fast arithmetic unit of a computer. Fig. 8 shows the arrangement schematically.

The idea discussed in the last paragraph can be extended. All inputs and all outputs of n flip-flops can be tied to a common bus. To transfer information from flip-flop i to flip-flop k, $-T$ of flip-flop i is raised, and $-E$ of flip-flop k is lowered. The two partially selected flip-flops then have the same relative potentials as those described in the section headed "the flow-gating principle," i.e., the transfer diode can set the receiving flip-flop.

Flow-Gating Applied to Double-Gating

It is clear that flow-gating principles can be extended to include double-gating. The automatic clearing feature can then be dropped. This may not be very interesting as long as the simple transfer between two flip-flops is considered, but the possibility of using partial selection (described in the previous section) is quite attractive. In order to achieve this condition, output emitter-followers are added to a classical Eccles-Jordan flip-flop and their collectors are tied, through an appropriate load, to a variable supply voltage. This voltage is then raised for the sending flip-flop. The receiving flip-flop has the voltage supply of the cross-coupled inverters lowered until the bases of the inverters come into the receiving region where the two transfer diodes going into these points can (conditionally) conduct.

Finally, it should be mentioned that under some circumstances the condition of a single supply voltage for the bistable circuit can be dropped. All that is really needed is that all supply voltages $E_1E_2\ldots$ be simultaneously reduced or increased to kE_1, $kE_2\ldots$, k being some numerical constant. The fact that this simultaneous variation is practically quite difficult to achieve, reduces the multiple supply voltage scheme to one of only academic interest.

Minimum Transistor Logic Modules for Air-borne Control Applications

ARNEY LANDY, JR.
NONMEMBER AIEE

IN RECENT years, digital methods have found application in the real time control of aircraft and missiles. This real time control is accomplished by performing mathematical computations within some form of air-borne digital computing device on sampled input variables. This recent trend means that the hardware techniques need to be tailored to air-borne applications.

The logic modules described in this paper have been designed for a maximum of flexibility to permit the needed versatility for implementing and testing a wide variety of air-borne digital computing systems. Furthermore, they are based on design concepts which will allow them to be used as such, or with minor modifications, in a final system design. In particular, the digital logic modules have been designed to have the following characteristics:

1. Simple, small, and compact
2. Light in weight
3. Impervious to shock and vibration
4. Capable of high-speed operation
5. Operation over an extended temperature range
6. Use of modular construction for convenient replacement in servicing or making logic changes.

Because of their obvious advantages, transistors have been applied extensively in mechanizing electronic digital systems. However, presently available circuit techniques such as direct coupled transistor logic (DCTL) circuits require the use of large quantities of transistors. Layout techniques that have been used in the past are wasteful of space factors so important in air-borne applications.[1]

The transistor circuit modules described in this paper are believed to embody some new concepts tailored to contain the characteristics just listed.

Primary Sets of Logic Modules

In order to maintain a desirable balance between the requirements in circuit design to meet the previously listed characteristics, the primary criterion established during development was that a minimum number of transistors per function accomplished were to be used.

The type of logic that the circuits are designed to mechanize is a combination pulse and d-c level logic. The basic functions and symbols used are defined in

Arney Landy, Jr., is with Minneapolis-Honeywell Regulator Company, Minneapolis, Minn. These circuits were developed on a company-sponsored research project.

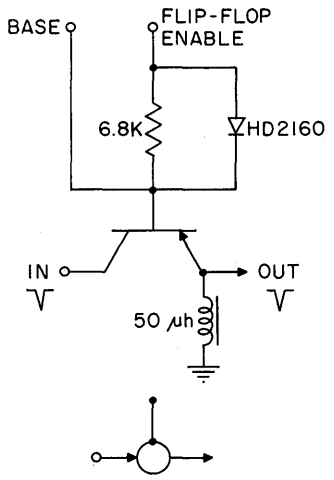


Fig. 1. And-gate module (two gates per module)

each circuit diagram. Since the complements of all logic inputs are available from the flip-flops, any logical expression can be implemented by means of "and" and "or" gates exclusively.

At present, the circuits have been employed in the development of an experimental air-borne digital-analog system which mechanizes inertial guidance, bombing, and steering equations. The system is in the preflight test stage now.

Two sets of logic modules containing the same circuit configurations have been developed:

1. One set using 4-volt 0.4-microsecond (μsec) pulses and inexpensive General Electric-type *4JD1A73* germanium alloy-junction transistors for medium-speed (up to 400 kc) applications of input-output transducers.

2. A second set using 4-volt 0.1-μ sec pulses and Philco-type *T1166/2N393* germanium micro-alloy transistors for high-speed (up to 5 megacycles) applications of the control, memory, and arithmetic sections.

Since several medium-speed applications exist, it has been economically feasible to retain use of the medium-speed set of circuits built with low-cost transistors. Since the second set is of primary importance, the descriptions of circuits will be limited to this set. Commercially-available components are used throughout.

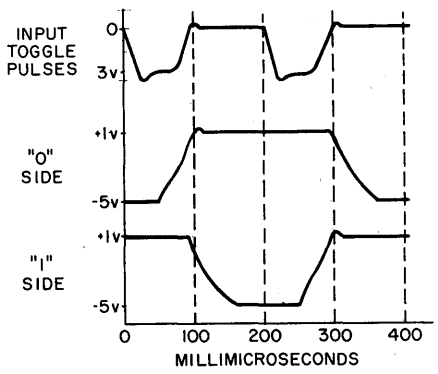


Fig. 3. Flip-flop switching times

Preliminary tests have been conducted successfully on a third set using Philco silicon transistors in the same circuit configurations. These units offer higher temperature range advantages.

The "And" Gate

In order to meet the criterion of minimum transistors to mechanize combination pulse and d-c level logic, an analysis was made of gating properties of transistors. By considering a transistor as a 3-terminal element, it was determined that a junction transistor could function as a gate by establishing an enable voltage or current in one element, applying a pulse voltage to a second element, and obtaining an output pulse at the third element. As a result of investigations concerning the six possible connections, one connection yielded the unique "and"-gate package shown in Fig. 1. This gate package has the following characteristics:

1. No pulse inversion occurs in passing a pulse through the gate, thereby eliminating transistors or transformers at the input or output normally necessary for performing this function.

2. Minimum pulse delay occurs in passing a pulse through the gate, i.e., ordinary electronic conduction as in a piece of wire.

3. The switch-on and switch-off times depend on the type of transistor selected, as shown in Table I.

The enable level is -5 volts for "on" condition and $+1$ volt for "off" condition. Both junctions are forward-biased by the negative swing prior to application of a pulse. The positive swing, together with the diode across the base resistor, provides rapid removal of stored carriers in the base region.

Use of a 50-microhenry choke in the emitter circuit maintains a high impedance to ground while still furnishing a relatively constant bias, regardless of the number of added circuits.

Flip-Flop

A conventional Eccles-Jordan saturation-type flip-flop shown in Fig. 2 is used. However, the low input impedance to the flip-flop is overcome through use of capacitor-trigger through series resistors. This trigger method is possible since no positive excursion exists on the trigger pulse. The minimum trigger level for 0.1 microsecond pulses is 2.0 volts.

A simple pulse "or" circuit configuration is available at the transistor base since the input impedance is low (about 300 ohms). Additional resistor-capacitor sets are decoupled when tied to the base. Complementing or toggle operation is accomplished merely by tying the 0 and 1 sets together. The clamping diodes from base to emitter re-establish d-c level of trigger pulses for reliable high-repetition rate operation. The flip-flop switching waveforms are shown in Fig. 3. The switch-on and switch-off times are 50 and 80 millimicroseconds, respectively. The delay before switching is slightly less than the pulse input width and allows the flip-flop to be set at the same time that it is being sensed. The flip-flop can

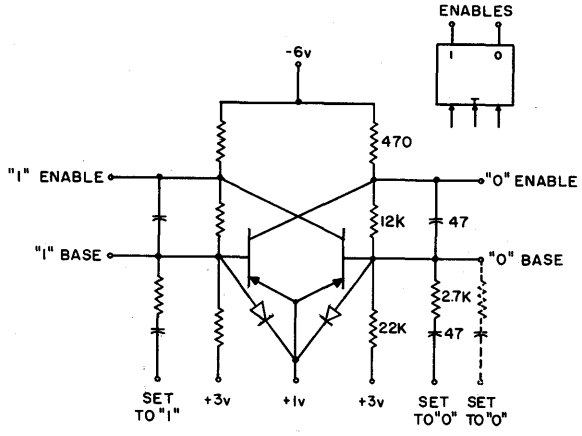


Fig. 2 (left). Flip-flop module

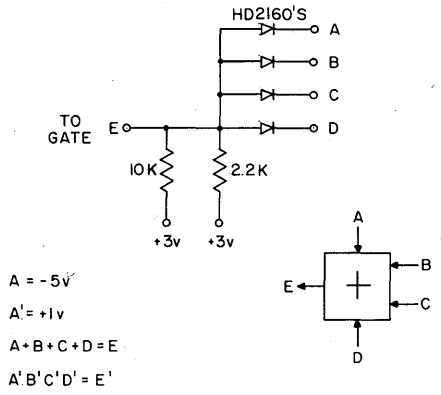


Fig. 4 (right). Level-or module

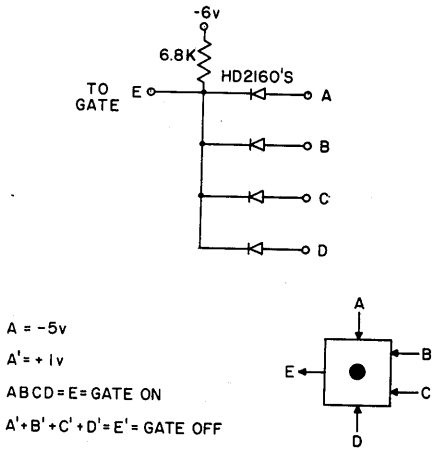


Fig. 5. Level-and module

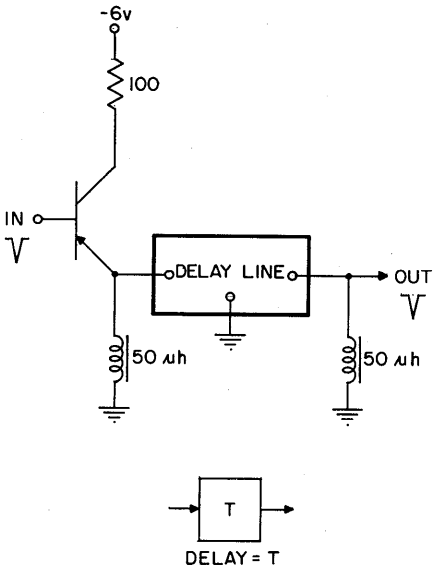


Fig. 6. Pulse delay module

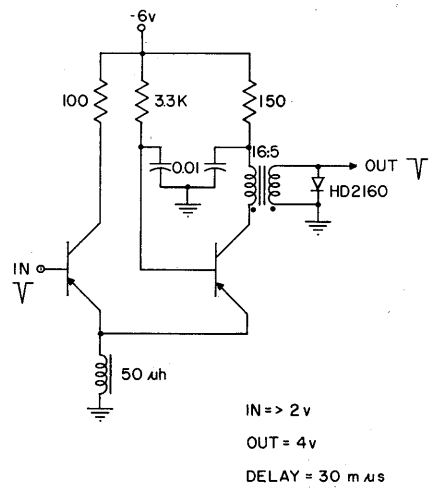


Fig. 7. Pulse amplifier module

switch a load impedance as low as 300 ohms. Normally, the load impedance is limited to that of six gates (1000 ohms).

Temperature range operation of up to 175 degrees Fahrenheit is achieved by providing an enable swing to the gates of +1 volt and biasing the flip-flop bases from +3 volts through 22 K.

Remaining Logic Modules

Conventional d-c level diode "or" and "and" circuits are well suited for use with the "and" gate. The schematics of the level "or" and level "and" modules are shown in Figs. 4 and 5. In connections involving "and"-to-"or" circuits tied to the gate base, the 10-kilohm resistor provided in the "or" circuit is used. For the opposite connection, "or"-to-"and", the 2-kilohm resistor in the "or" circuit is used.

Though introducing some loss as well, pulse delay is accomplished through use of a low-impedance delay line with a pulse emitter-follower input for impedance matching as shown in Figure 6. Lines with 150-ohms impedance were found to be suitable.

Since the gate is a highly-efficient passive element having about 5 ohms "on" resistance, and the input impedance to the flip-flop is maintained at 3,000 ohms, a 4-volt pulse from one gate can drive 14 flip-flop sets. Eventually, however, the pulse level deteriorates in passing through gates or delay lines and a pulse amplifier is required to re-establish the pulse level. The pulse-amplifier function was obtained by shutting off a high current in the primary of a pulse transformer as shown in Fig. 7. The pulse amplifier module has the following characteristics:

1. High input impedance (20 kilohms).
2. Low output impedance (10 ohms).
3. Small pulse delay (30 millimicroseconds).
4. No pulse polarity reversal.

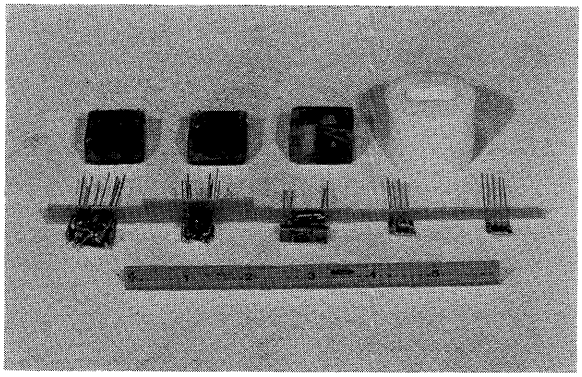


Fig. 8. Three-dimensional modules

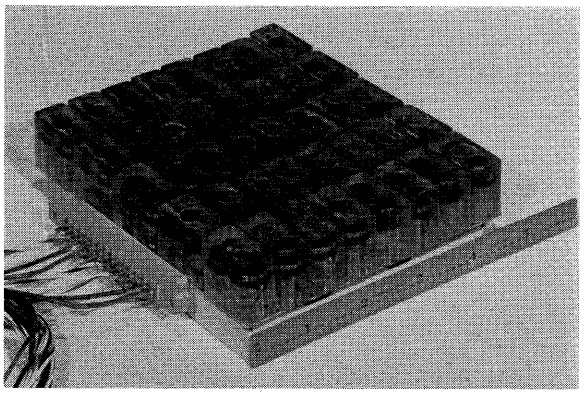


Fig. 9 (right). A 20-bit shift register using modular construction technique

Table I. Gate Switching Times

Set	Transistor Type	Switch-on (μs)	Switch-off (μs)
Medium Speed	2N136	0.50	0.04
	4JD1A73	0.40	0.04
	2N247	0.40	0.04
High Speed	T1166/2N393	0.08	0.03

Construction Technique

In contrast to scientific calculator applications, the air-borne applications require both small size through simple circuits and a rugged construction technique. An advanced construction technique has been developed involving potting of the modules previously described in reusable packages. Minneapolis-Honeywell has developed several

types of potting compounds, and type *6020A* which cures at room temperature has been used here. Several of the modules are shown in Fig. 8. The modules in the upper row are examples of the medium-speed set while the bottom row shows the high-speed set. A model shop mold is shown in the upper right. The 3-dimensional construction reduces interwiring capacity and pickup problems since all leads are short. These module packages are assembled on an etched circuit board just as transistors have been in the past, as shown in Fig. 9. All boards in a system are placed on top of each other and are clamped together for a secure package. This method expedites the construction of flyable engineering models of digital equipment.

A 20-bit shift register using the medium-speed set, as shown in Fig. 9, consumes a volume of 15 cubic inches. Since the high-speed set modules are more compact, a 20-bit shift register built with the high-speed set occupies a volume of 7 cubic inches. The technique provides the obvious advantage of being able to disconnect such a unit at any time and use the basic modules for other functions.

References

1. Transistor Circuitry in the Lincoln TX-2, **K. H. Olson.** *Proceedings of the Western Joint Computer Conference*, 1957, pp. 167–71.

2. Program Memory System for an Airborne Digital Computer, **A. Landy, Jr.** *M. S. Thesis*, Iowa State College, 1957.

Transistor-Magnetic Core Bilogical Computer Element

W. J. DUNNET
NONMEMBER AIEE

A. G. LEMACK
NONMEMBER AIEE

SQUARE-loop magnetic core materials such as the ferrites and thin tapes, when employed in a toroidal core configuration, can be used to perform common logical functions such as "or," "inhibit," and "invert."[1,2,3] It is possible to switch these core materials from negative to positive saturation flux density within several hundred millimicroseconds using less than 100 milliwatts of peak power. Cores themselves would then appear to be capable of operating in a sequential-type system at clock speeds in excess of one megacycle. However, in a practical system, realization of the fast switching time and corresponding low power consumption is difficult primarily due to the problems encountered in directly coupling one core with another.[4] Transistors and diodes are often used as coupling devices between cores.

A number of coupling schemes described in the literature employ semiconductor devices. In general those functions, aside from storage, which must be accomplished by the core and its associated semiconductors are: 1. regeneration, 2. isolation, and 3. logic implementation. Table I lists the respective functions of each.

This paper is concerned with introducing a relatively simply transistor-magnetic core type of circuit in which the core and transistor share in the performance of the logical function (bilogical) and the transistor provides coupling as well as regeneration.

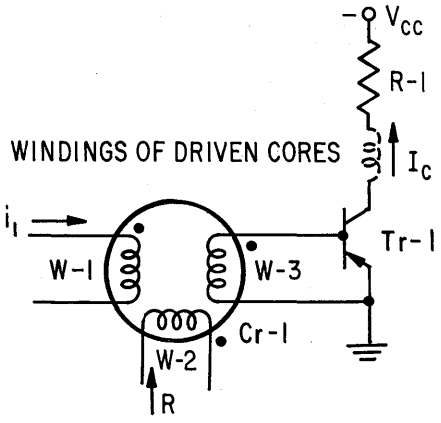
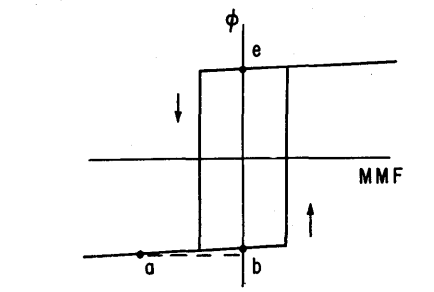
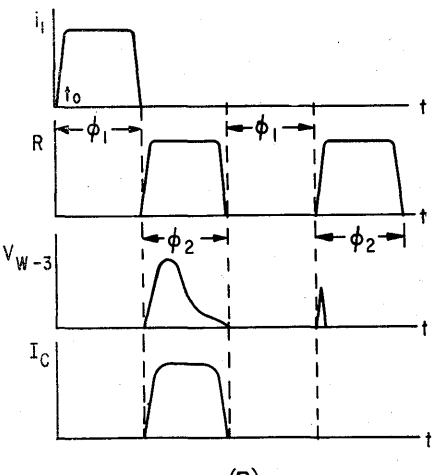


Fig. 1. The bilogical computer element



(A)



(B)

Fig. 2(A). An idealized hysteresis loop (B). Pulse timing diagram

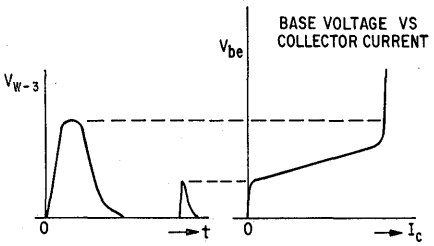


Fig. 3. Transistor transfer characteristic

The four advantages of the Bilogical Computer Element over other types of core-semiconductor circuitry are as follows:

1. Circuit speeds up to one megacycle can be achieved with approximately 250 milliwatts of average power per element. (The 250 milliwatts include power dissipated in transistorized-core driving circuits.)

2. Fewer system semiconductors are required due to a sharing of the logical function between cores and transistors.

3. Coding, control, and logical functions can be obtained with a single basic circuit.

4. The only circuit components are cores, transistors, and resistors. (The system master clock sources and transistorized-core driving circuits may employ other components such as capacitors, diodes, and pulse transformers.)

W. J. Dunnet and A. G. Lemack are with Sylvania Electric Products, Inc., Waltham, Mass.

This work was supported by United States Army Ordnance under the PLATO Contract DA-30-069-ORD-1166.

Table I. Circuit Functions of Semiconductors and Cores

Type of Circuit	Circuit Functions: Regeneration	Isolation	Logic
Core-Diode[5]	Cores	Diodes	Cores
Core-Diode[6,7]	Cores	Diodes	Diodes
Core[4]	Cores	Cores	Cores
Transistor-Core[8]	Transistors	Transistors	Cores

Table II. Switching Functions of Two, Three, and Four Variables That Can Be Obtained by Shunting and Stacking Two Bilogical Elements

i_1, i_2	i_1, i_2, i_3	i_1, i_2, i_3, i_4
$i_1 i_2$	$\bar{i}_1\bar{i}_2 i_3$	$(i_1+i_2)\ (i_3+i_4)$
$\bar{i}_1\bar{i}_2$	$\bar{i}_1 i_2 i_3$	$\bar{i}_1\bar{i}_2 i_3 i_4$
$i_1\bar{i}_2$	$\bar{i}_1\bar{i}_2\bar{i}_3$	$\bar{i}_1\bar{i}_2\bar{i}_3 i_4$
i_1+i_2	$i_1 i_3+i_2 i_3$	$\bar{i}_1\bar{i}_2\bar{i}_3\bar{i}_4$
$\bar{i}_1+i_2$	$i_1\bar{i}_3+i_2\bar{i}_3$	$i_1+i_2+i_3+i_4$
$\bar{i}_1+\bar{i}_2$	$i_1+i_2+i_3$	$\bar{i}_1+i_2+i_3+i_4$
$i_1+i_1\bar{i}_2$	$\bar{i}_1+i_2+i_3$	$i_1\bar{i}_2+i_3\bar{i}_4$
$i_1\bar{i}_2+\bar{i}_2 i_1$	$\bar{i}_1+\bar{i}_2 i_3$	$\bar{i}_1+\bar{i}_2\bar{i}_3 i_4$
	$\bar{i}_1+i_1\bar{i}_2\bar{i}_3$	$i_1+\bar{i}_2\bar{i}_3 i_4$
	$\bar{i}_1 i_2+i_1\bar{i}_2\bar{i}_3$	$i_1\bar{i}_2+i_2\bar{i}_3 i_4$
	$\bar{i}_1\bar{i}_2 i_3+i_1\bar{i}_2\bar{i}_3$	$i_1\bar{i}_2 i_3+\bar{i}_1 i_4$
	$i_1+\bar{i}_2 i_3$	$i_1\bar{i}_2+\bar{i}_3 i_4$
	$i_1+\bar{i}_1\bar{i}_2 i_3$	

Table III. Switching Functions of a 3-Bit Decoder

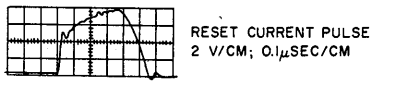
Output	Input Function
I_1	$\bar{i}_1\bar{i}_2\bar{i}_3$
I_2	$\bar{i}_1\bar{i}_2 i_3$
I_3	$\bar{i}_1 i_2\bar{i}_3$
I_4	$\bar{i}_1 i_2 i_3$
I_5	$i_1\bar{i}_2\bar{i}_3$
I_6	$i_1\bar{i}_2 i_3$
I_7	$i_1 i_2\bar{i}_3$
I_8	$i_1 i_2 i_3$

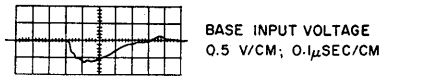
Operation of Transistor-Magnetic Core Bilogical Computer Element

Fig. 1 shows the bilogical computer element. As shown, the core performs storage and the logical function. The transistor is used to provide power gain and isolation between the driving and driven cores. Referring to the idealized hysteresis loop in Fig. 2(A), and the pulse timing diagram in Fig. 2(B), the operation of the circuit is as follows. At time t_0 the flux level in the core, *Cr-1*, is at point *b*, as shown in Fig. 2(A). During the interval ϕ_1, a current pulse i_1, is applied to input winding *W-1*. This pulse is of sufficient amplitude and duration to cause the flux level in *Cr-1* to be set to point *e*. When this switching occurs a voltage is induced, positive at the polarity markings, in windings *W-2* and *W-3*. Negligible current occurs in each of these windings during this interval. Reset winding *W-2* is terminated with the high impedance of an off-reset current supply, and output winding *W-3* is terminated with the high back impedance of the transistor emitter diode. During the interval, ϕ_2, the input pulse is removed and a reset current pulse, *R*, is applied to winding *W-2*. The direction of the reset pulse is such that the flux level in the core is restored to point *b*. This change in flux induces a voltage which is negative at the polarity markings, in windings *W-1* and *W-3*. Input winding *W-1* is terminated with the high impedance of an off input current source. The induced voltage in *W-3* is of proper polarity and sufficient magnitude to switch transistor *TR-1* into saturation. A path for current is then provided from ground through the transistor, the load resistor *R-1*, and the input windings of any logically connected cores, to the negative terminal of V_{cc}.

If an input pulse does not appear at *W-1* during interval ϕ_1, the core remains at point *b*. Upon application of a reset phase during interval ϕ_2, the core will be driven out to point *a*. The small change in flux that occurs produces a spike of voltage, negative at the polarity markings, in the various windings on the core. However, this spike of voltage is relatively ineffective in causing collector current, due to the nonlinear transfer characteristics of the transistor (as shown in Fig. 3) and magnetic core. A relatively large spike of current can be tolerated in the input windings of driven cores because narrow spikes (0.1 microsecond) for the most part cause elastic flux changes. Newhouse indicates that a 0.63-oersted field applied for 0.1 microsecond to a *4-79 mo-permalloy*, 1/8-mil core causes very little irreversible flux change.[9]

It is important to note that regeneration of the core output-pulse width as well as amplitude can be accomplished by the transistor. Width regeneration is obtained by controlling the transistor storage time. A small amount of energy is stored in the magnetic core as it is driven into saturation during reset. This energy is released during the fall time of the reset current pulse and aids in sweeping minority carriers out of the base region of *Tr-1*. Core output pulse shape and the maximum-transistor hole storage





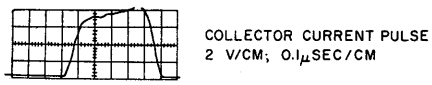


Fig. 4 (left). Waveshapes from a one-megacycle bilogical element

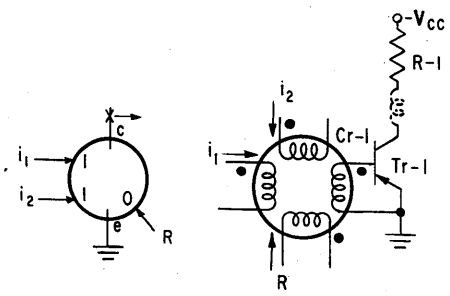


Fig. 5 (left). Symbolic representation of the bilogical element

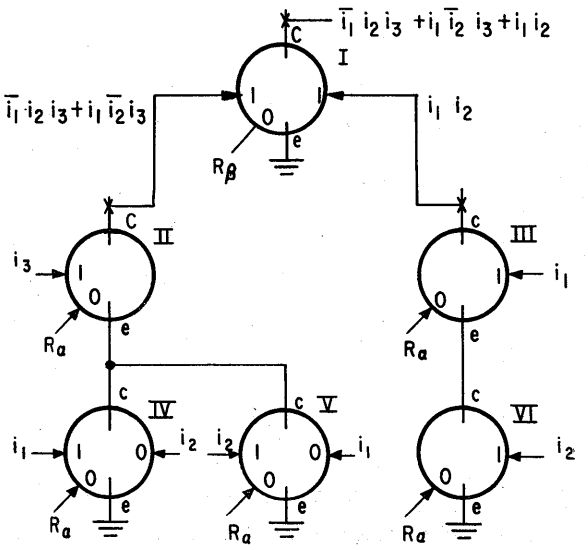


Fig. 6 (right). The carry for a full serial adder

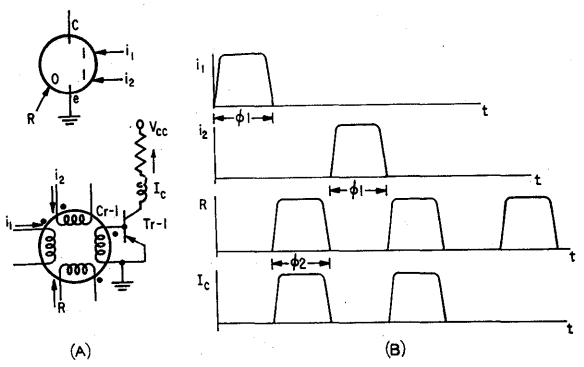


Fig. 7(A). i_1+i_2 with a bilogical element
B). Pulse timing diagram

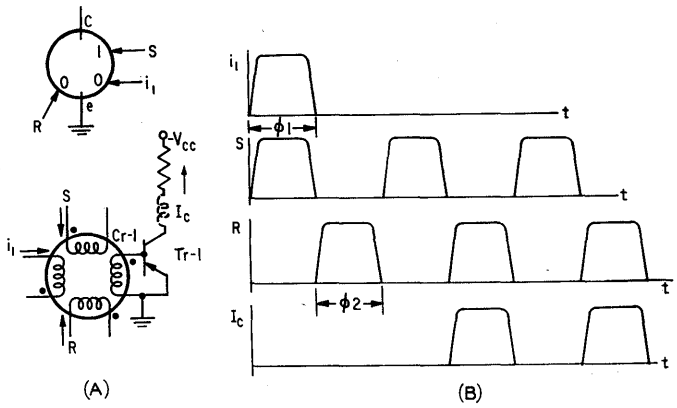


Fig. 8 (A). i with a bilogical element
(B). Pulse timing diagram

must also be considered in order to prevent excessive widening of the collector pulse. Fig. 4 shows the waveshapes of current reset, core output, and collector current pulses taken from a one-megacycle bilogical element.

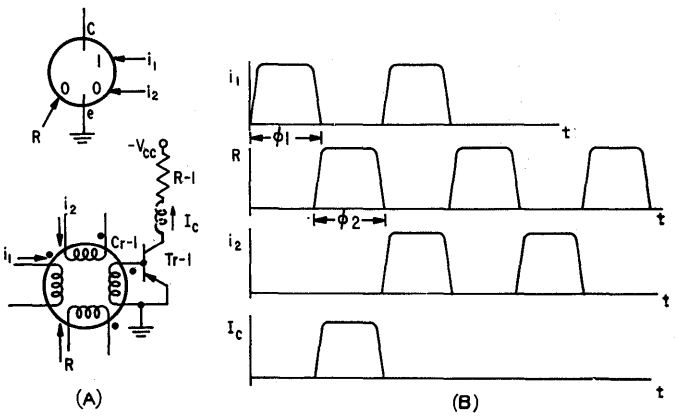


Fig. 9 (A). $i_1\bar{i}_2$ with a bilogical element
(B). Pulse timing diagram

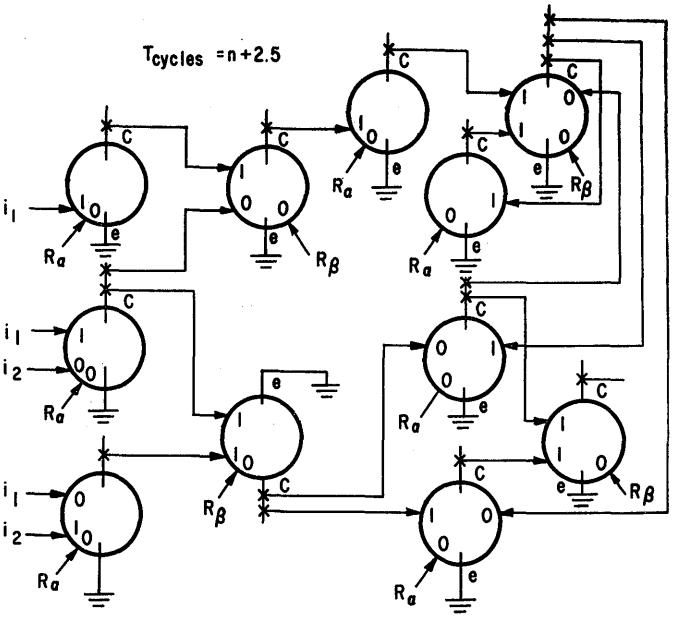


Fig. 10. A full adder using "inhibit-or" logic

Logic Design Symbolism

A description of the simple "or," "invert," and "inhibit" functions, as well as of the more complex functions, can be simplified by an explanation of input and output pulse timing and by the adoption of a symbolic representation of the bilogical element.

This paper is concerned only with the 2-phase type of synchronous system.[2] In this type of system two equal master clock intervals (phases) α and β, along with two equal local clock phases, ϕ_1 and ϕ_2, are employed. During its input phase ϕ_1, a core associated with a bilogical element, may or may not be set, depending upon which of the possible input pulses appear and which are absent. During its output phase ϕ_2, the core is reset, providing that the core was set during its previous input interval, and a pulse of current occurs in the collector of the associated transistor.

If ϕ_2 occurs during master clock phase α, the bilogical element is said to be in mode α. Similarly if ϕ_2 occurs during master clock phase β, the bilogical element is said to be in mode β. In 2-phase synchronous systems implemented with magnetic circuits, information is shifted from a circuit operating in one mode to another circuit operating in the other mode.

The schematic representation of the bilogical element, as shown in Fig. 1, can be simplified through the adoption of a symbolism.[5] The symbolism to be used here is taken in part from D. Loev et al.[5] Referring to Fig. 5, a circle represents the core, and an arrow pointing to the circle represents an input to the core, setting it to the binary state indicated just inside the circle. The symbol on the input arrow indicates the data designation. If an input signal appears, it will always be during local clock interval ϕ_1. R represents a reset function which always appears during local clock interval ϕ_2. The subscripts α and β are used to denote the master clock interval during which R occurs. Similarly, S represents a setting function which is also unconditional in its appearance and always appears during a local clock phase ϕ_1. In order to simplify the symbolic circuit, the transistor symbol has been omitted. However, the transistor collector and emitter are shown as follows. A c at the intersection of a line and a circle indicates a connection with the collector. Similarly, an e at the intersection of a line and a circle indicates a connection with the transistor emitter.

When symbolic circuits are interconnected, the appearance of an x at the junction of a line associated with the collector

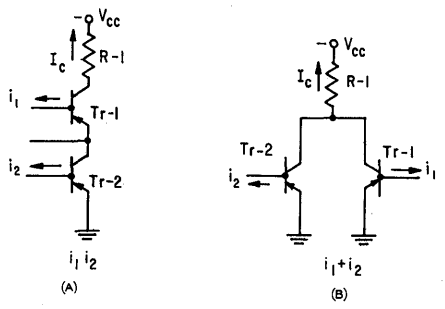


Fig. 11 (A). $i_1 i_2$ with stacked transistors (B). i_1+i_2 with shunted transistors

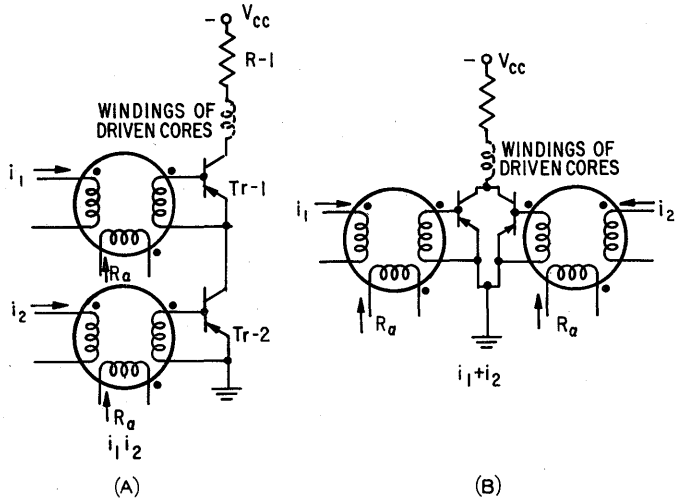


Fig. 12 (A). $i_1 i_2$ with stacked bilogical elements (B). i_1+i_2 with shunted bilogical elements

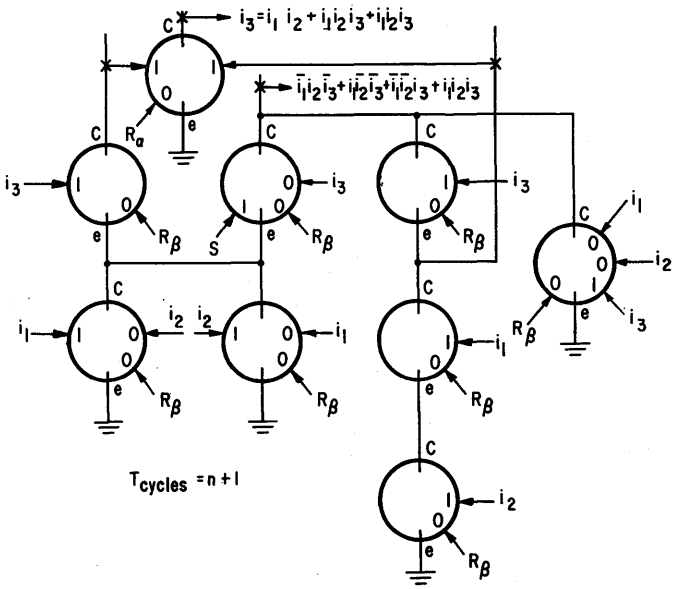


Fig. 13. A full adder using shunted and stacked bilogical elements

of a transistor and an input arrow indicates that the core winding associated with the input is in series with the collector. A dot at the intersection of lines associated with collectors indicates a voltage connection.

As an example, the symbolic circuit of Fig. 6 shows a number of bilogical elements arranged so that they develop the carry for a full serial adder. The transistors of elements *IV* and *V* are arranged in shunt. The collectors are tied directly together. The input windings of *I* are connected in series with the collectors of elements *II* and *III*.

Logical Functions Performed with the Magnetic Core of the Bilogical Computer Element

Through the use of two input windings, an output and a reset winding, the logical functions "or," "invert," and "inhibit" can be performed by the magnetic core of a bilogical element. Logic performed with cores has been described by others in the literature.[1-3] As a result it will be covered only briefly here.

Figs. 7(A) and (B) show the timing of input and output pulses, and arrangement of core windings required to perform an "or" function, i_1+i_2. If either input i_1 or i_2 is present, *Cr-1* will be set. During interval ϕ_2 the reset function, *R*, resets *Cr-1*. The resulting change in flux induces a voltage in the core output winding. This pulse of voltage is of sufficient amplitude to switch *Tr-1* into saturation and cause a pulse of current, I_c, to occur in the collector. If both i_1 and i_2 are absent, *Cr-1* will not be set. During interval ϕ_2 the reset function, *R*, effects little change of core flux. As a result *Tr-1* remains off and there is no collector current pulse.

The invert function $\bar{i}_1$ can be obtained as shown in Fig. 8(A). The setting function *S* appears every input clock interval ϕ_1. If input i_1 is present, *S* is prevented from setting the core, that is, the magnetomotive force due to i_1 and *S* oppose each other resulting in a zero net magnetomotive force and, therefore, no core flux is changed. As a result during interval ϕ_2, *Tr-1* remains off and there is no collector output pulse. If input i_1 is absent, the setting function *S* sets core *Cr-1*. During interval ϕ_2 the core is reset and an output pulse occurs in the collector of *Tr-1*.

Figs. 9(A) and (B) show the timing of input and output pulses and arrangement of core windings required to perform the "inhibit" function, $i_1\bar{i}_2$. If only input i_1 is present *Cr-1* will be set and an output pulse will occur in the collector of *Tr-1* during the next reset interval ϕ_2. If both i_1 and i_2 are present the core will not be set because the magnetomotive forces due to i_1 and i_2 cancel. As a result there will be no output collector pulse during interval ϕ_2. Finally, if i_2 alone is present, again the core will not be set because the magnetomotive force due to i_2 is in such a direction as to keep the core in negative saturation. There will be no output pulse during ϕ_2.

The circuits of Figs. 7, 8, and 9 can be cascaded in a system to perform more complex logical and control functions. Two types of logic "inhibit-or"[3] and "invert"[10] can be used to accomplish these more complex functions. For example, Fig. 10 shows bilogical elements arranged so that a full serial adder with "inhibit-or" logic is obtained.

It is important to note as before, that in the full adder shown in Fig. 10, the cores perform logical and storage functions; the transistors provide isolation and regeneration.

Logic Sharing

Systems developed by cascading bilogical elements with "inhibit-or" or

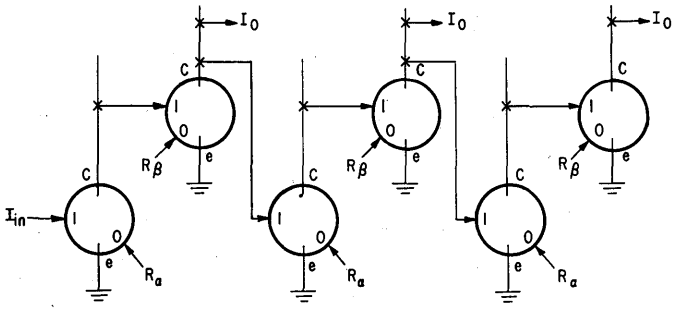


Fig. 14. A 3-bit shift register

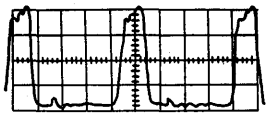


Fig. 15. 10101 pattern in a 5-bit one-megacycle shift register

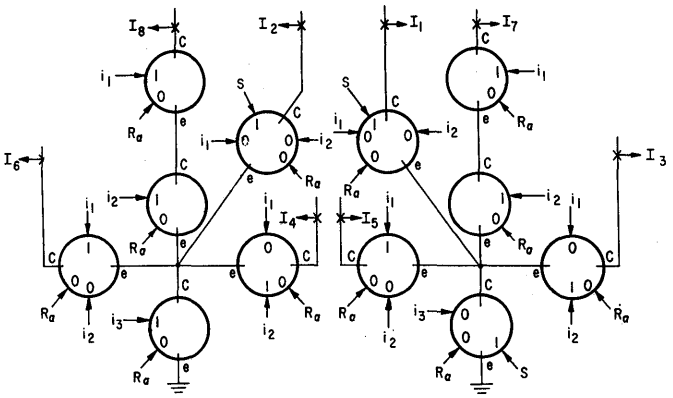


Fig. 16. A 3-bit decoder

"invert" types of logic as previously mentioned do not represent the ultimate in system reliability. Considerable savings in components and a reduction in system delay can be achieved by permitting the transistors to assist the cores in performing the logical function. This sharing of the logical function can be accomplished by stacking and shunting the bilogical element transistors.

Two transistors stacked as shown in Fig. 11(A) perform an "and" function, $i_1 i_2$. If inputs i_1 and i_2 are present both transistors will be switched on and a pulse of current will occur in their collector circuits. If either i_1 or i_2 is absent, one of the two transistors will remain off (nonconducting) and there will be no collector pulse.

Two transistors arranged in shunt, as shown in Fig. 11(B) perform an "or" function, i_1+i_2. If either i_1 or i_2 is present one of the two transistors will be switched on and a current pulse will occur in the output circuit. If both inputs are absent both transistors will remain in the off condition and there will be no output current pulse.

When the transistors of two bilogical elements are stacked as shown in Fig. 12(A) the transistors perform an "and" function and the cores perform either "or," "invert," or "inhibit" functions. Table II lists some of the switching functions of two, three, and four variables that can be obtained from two stacked bilogical elements.

When the transistors of two bilogical elements are arranged in shunt as shown in Fig. 12(B), the transistors perform an "or" function and the cores perform "or," "invert," or "inhibit" functions. Table II lists some of the switching functions that can be obtained from shunting two bilogical elements.

It becomes apparent then that a complex system can be constructed by stacking and shunting bilogical element transistors, For example, the Boolean expressions, defining a full adder are as follows:

$$\text{Sum} = i_1\bar{i}_2\bar{i}_3 + \bar{i}_1 i_2\bar{i}_3 + \bar{i}_1\bar{i}_2 i_3 + i_1 i_2 i_3$$

$$\text{Carry} = \bar{i}_1 i_2 i_3 + i_1\bar{i}_2 i_3 + i_1 i_2$$

These expressions can be implemented as shown in Fig. 13. Note that nine cores, nine transistors, and four resistors are required to obtain a full adder whereas the adder of Fig. 10 using "inhibit-or" logic requires 11 cores, 11 transistors and 11 resistors. From Fig. 13, the time in cycles of clock frequency required to add two n-bit binary words serially is

$$T_{\text{cycles}} = n+1$$

As shown in Fig. 10, if "inhibit-or" logic is used, the addition time becomes the following:

$$T_{\text{cycles}} = n+2.5$$

The use of the bilogical computer element is not restricted to the performance of logical functions. Shifting and coding can also be performed. Fig. 14 shows a 2-bilogical element per bit shift register. In this instance no logical function is being performed. The cores are used for storage, and the transistors are used both for isolation and for regeneration. Fig. 15 shows a collector current waveshape taken from a 1-megacycle shift register that is circulating the binary number 10101.

Fig. 16 shows 12 bilogical elements arranged so that they decode a 3-bit binary number. Table III lists the eight switching functions that describe a 3-bit decoder. Note again that the cores and transistors are sharing in the performance of the logical function.

Conclusions

The development of a transistor-magnetic core circuit, known as the bilogical computer element, permits both the cores and transistors in a sequential-type computer system to share in the performance of logical functions.

The transistor is an excellent core coupling device. It satisfactorily isolates driving from driven cores and provides pulse width as well as amplitude regeneration. The use of transistor coupling has permitted core logic to be performed, with moderate power consumption, above one megacycle.

Logic sharing effectively increases system reliability by decreasing the number of transistors required to perform a given function. System reliability is not only increased by lower transistor counts but also by the simplicity and flexibility of the basic circuit. Essentially a bilogical element system is made up of only cores, resistors and transistors. Coding and control as well as logical functions can be accomplished with the bilogical computer elements.

References

1. The Use of Magnetic Cores as Switching Devices, **R. C. Minnick.** *Ph.D. Thesis*, Harvard University, Cambridge, Mass., Apr. 1953.

2. Pulse-Switching Circuits Using Magnetic Cores, **M. Karnaugh.** *Proceedings*, Institute of Radio Engineers, New York, N. Y., vol. 43, May 1955, pp. 570–83.

3. Logical and Control Functions Performed with Magnetic Cores, **S. Guterman, R. D. Kodis.** *Ibid.*, Mar. 1955, pp. 291–98.

4. Diodeless Magnetic Core Logical Circuits, **L. A. Russell.** *National Convention Record*, Institute of Radio Engineers, pt. 4, Mar. 1957, pp. 106–14.

5. Magnetic Core Circuits for Digital Data-Processing Systems, **D. Loev, W. Miehle, J. Paivinen, J. Wylen.** *Proceedings*, Institute of Radio Engineers, vol. 44, Feb. 1956, pp. 154–62.

6. The Single-Core Magnetic Amplifier as a Computer Element, **R. A. Ramey.** *Report no.*

4030, Naval Research Laboratory, Aug. 8, 1952.

7. Series Pulse Magnetic Amplifiers, **R. D. Turney, A. L. Krell.** *Scientific Report AFCRC-TN-56-186, AD110190*, Feb. 1, 1957.

8. A Transistor-Magnetic Core Circuit, **S. Guterman, W. M. Carey.** *National Convention Record*, Institute of Radio Engineers, pt. 4, Mar. 1955, pp. 84–94.

9. The Utilization of Domain Wall Viscosity in Data-Handling Devices, **V. L. Newhouse.** *Proceedings, Ibid.*, vol. 45, Nov. 1957, pp. 1484–92.

10. Magnetic Computer Has High Speed, **T. H. Bonn.** *Electronics*, New York, N. Y., Aug. 1, 1957, pp. 156–70.

11. Arithmetic Operations in Digital Computers (book), **R. K. Richards.** D. Van Nostrand Company, Inc., Princeton, N. J., 1955, pp. 89–90.

High-Speed Circuit Techniques Utilizing Minority Carrier Storage to Enhance Transient Response

L. P. RETZINGER
NONMEMBER AIEE

SINCE the introduction of transistors to the digital computer field, many circuit techniques have been employed to mechanize computer logic, flip-flops, and in general, bilevel switching operations. Initial transistor applications to digital computers employed circuit techniques carried over from vacuum-tube computer technology and in many instances the vacuum tube has been replaced; base for grid, emitter for cathode, and collector for plate, with the configuration of the other components of the circuit remaining unchanged.

Because the transistor is basically a current amplifying device rather than a voltage amplifier like the vacuum tube, modifications in switching circuit techniques have continuously evolved to minimize undesirable characteristics of the transistor or, more fully, to employ its inherent characteristics. Notable examples of such evolution, which departs radically from techniques used in vacuum-tube switching circuits, include transistor antisaturation techniques, direct-coupled transistor logic (DCTL), current mode gating, and transistor avalanche circuit techniques.

This paper describes still another approach to semiconductor digital-computer switching circuitry. Here the normally undesired minority carrier storage effects of high-conductance silicon junction diodes are utilized in combination with transistors to produce very fast transient response in flip-flop and pulse circuits.

Minority Carrier Storage in Diodes

The minority carrier storage effect exhibited by junction diodes is well described in the literature.[1] This storage effect is illustrated in Fig. 1 by the voltage and current wave forms for a junction diode being switched from forward to reverse condition. In this illustration, the diode D_1 is driven by a constant current source in the forward direction and by a constant voltage source in the reverse direction. Reverse current is limited by the series resistance R_1. Diode D_2 decouples R_2 from the circuit during the reverse half-cycle so that all stored charge in D_1 is discharged through R_1.

The wave forms of Fig. 1 may best be understood by referring to the instantaneous diode volt-ampere characteristic shown in Fig. 2. The forward characteristics indicated in the positive quadrant account for the overshoot in forward voltage across the diode when a constant forward current is switched through the diode. The instantaneous volt-ampere characteristics shown in the negative quadrant account for the diode current and voltage wave form when the diode is reversed biased by E_s and the current limited by R_1. If it is assumed that diode D_3 has no loss, i.e., no forward voltage drop, the saturation reverse current will be given approximately by

$$i_{rs}=\frac{E_s+E_{ds}}{R_1}$$

where E_{ds} is the reverse saturation voltage (obtained from the inverse instantaneous volt-ampere characteristic of the diode). If the resistance of R_1 is reduced to a value R_1' the current-voltage waveforms for the diode will appear as Fig. 3. The reverse saturation current will increase as noted. As R_1 is reduced still further, the reverse saturation characteristics eventually will not be observed, unless the current in the forward half-cycle is increased. The total charge delivered through the load resistor R_1' will be approximately the same as in Fig. 1, provided that the time interval under consideration is short in comparison to the lifetime of the carriers, i.e., recombination of carriers need not be considered. If the value of R_2 is reduced, thus increasing the diode forward current, the total charge Q_s stored in the diode by minority carriers will increase and, in fact, for a given diode, Q_s will be a function of both the magnitude of forward current and its duration. It should be noted that the instantaneous reverse characteristics indicated for the diode are very similar to typical transistor characteristics. Excess stored minority carriers in the diode under reverse conditions act exactly as excess carriers injected by the emitter of a transistor. Since there is no emitter current, the reverse characteristic of a diode is, of necessity, instantaneous. Techniques have been described in which these characteristics may be employed to provide a diode amplifier with both current and voltage gain.

Having described briefly the diode characteristics of importance for purposes of this paper, attention is now given to a number of semiconductor switching, gating, and pulse circuit applications which employ these characteristics to provide efficient and fast transient performance.

Storage-Diode Coupled Flip-Flop

Factors of prime importance in the design of fast-response transistorized flip-flop trigger circuits include;[2] turn-on time t_0, storage time t_1, and turn-off time t_2. Total transient response time for a triggered flip-flop may be defined as the time required for the flip-flop, consisting of two cross-coupled amplifiers, A and B, to change from a steady state condition no. 1, i.e., amplifier A conducting and amplifier B cut-off, upon application of a trigger pulse, to a steady-state condition no. 2, i.e., amplifier A cut-off and amplifier B conducting. Here, all three delay times, t_0, t_1, and t_2, are involved. A reduc-

L. P. Retzinger is with Litton Industries, Beverly Hills, Calif.

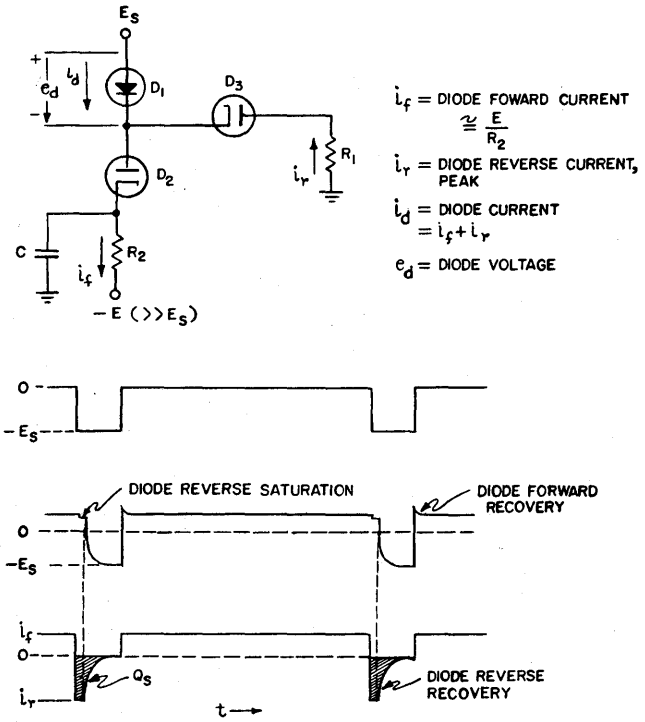


Fig. 1. Response of a silicon junction diode switched from forward to reverse bias

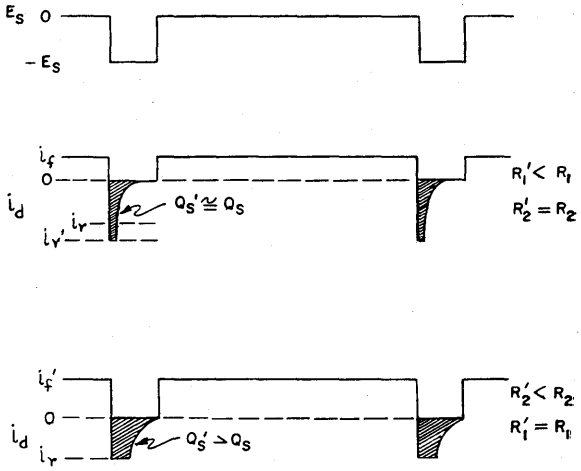


Fig. 3. Effect of lowering resistance in reverse diode discharge circuit and in forward charge circuit

tion of one or more of these time intervals will reduce the over-all transient response time.

Fig. 4 shows the schematic diagram of a base-triggered flip-flop circuit utilizing the storage diode cross-coupling technique to reduce delay times t_1 and t_2. The turn-on time t_0 of the triggered transistor, i.e., the initially "off" transistor, is reduced by a similar technique which is described later. In all cases, the end result using a storage diode will be to provide "over drive" for the base of the transistor during the transient period whether turning on or turning off a saturated amplifier.

Fig. 5 shows pertinent base and collector voltage wave forms for the triggered flip-flop during the transient period. The transient action of the circuit may be described as follows. Assume that transistor T_1 initially is in a cutoff condition and transistor T_2 is in saturation. At this point, the base of T_1 will be backbiased to about 0.25 volt, a value established by the current through R_3 in conjunction with conducting clamp diode D_5. The collector of T_1 will be a potential of approximately -1.5 volts, a level established by the sum of forward drops in conducting diode D_2, a silicon storage diode, and the base-emitter junction of transistor T_2. The current through D_2 will be the sum of the base current of T_2 plus the current i_{bb_2} through the back bias resistor R_4. Minority carriers will have been injected and stored in both diode D_2 and the base region of transistor T_2. The voltage at the base of T_2 will be approximately -0.5 volt established by the base characteristic of T_2; therefore, the base clamp diode, D_6, is in a nonconducting state. Transistor T_2 thus will be held in saturation so that its collector voltage e_{c_2}, will be approximately -0.1 volt for the transistor and current levels chosen. The voltage across storage diode

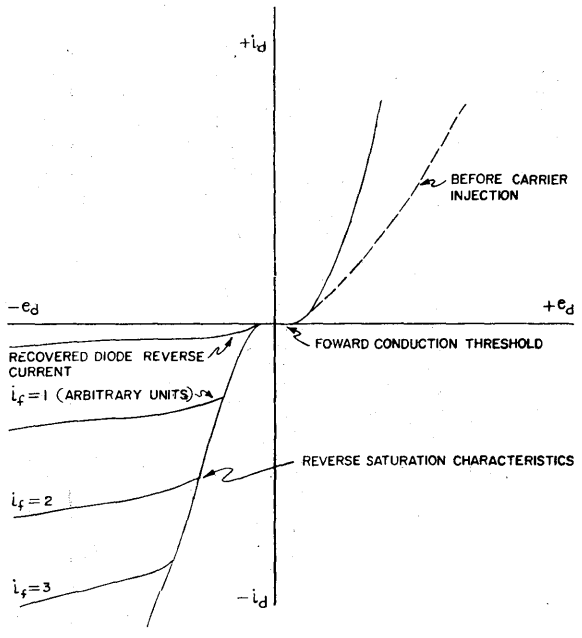


Fig. 2 (left). Instantaneous volt-ampere characteristic of a diode switched from forward to reverse

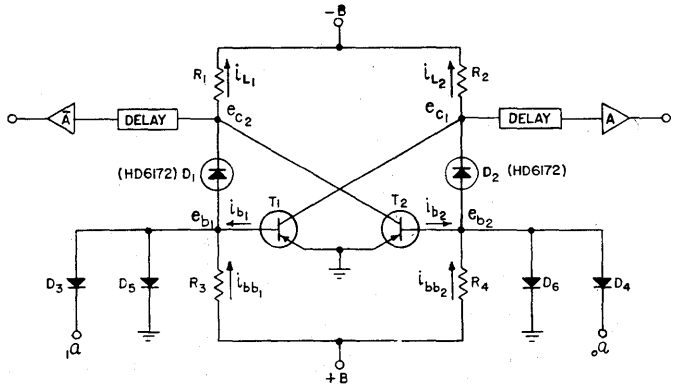


Fig. 4. Storage diode-coupled flip-flop

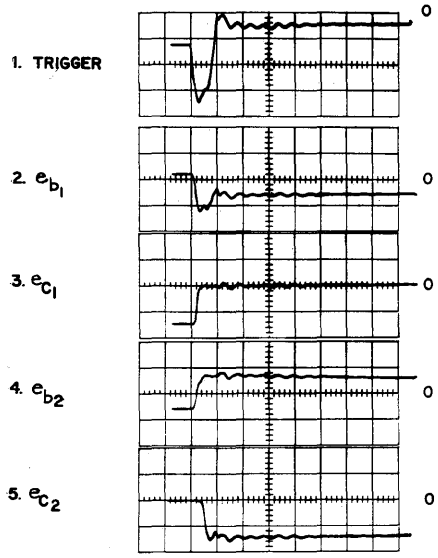


Fig. 5. Flip-flop waveforms. Vertical deflection = 1 volt per centimeter. Horizontal deflection = 0.1 μsec per cm. Trigger rate = 1 mc

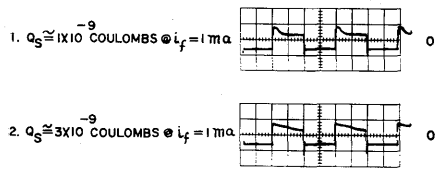


Fig. 6. Difference in storage diodes. Vertical deflection = 1 volt per cm. Horizontal deflection = 1 μsec per cm. Trigger rate = 500 kc

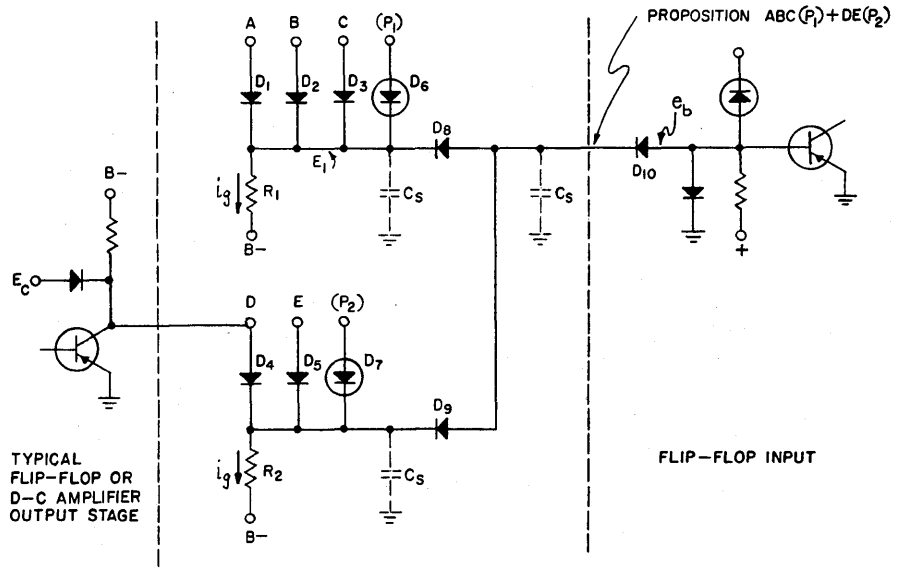


Fig. 7. Gated flip-flop using current-charge gating

D_1 is approximately 0.35 volt, a value below the forward conduction threshold for a silicon diode. Steady-state d-c stability for the circuit shown thus is assured.

Referring now to the wave forms of Fig. 5, trigger action of this circuit may be explained. Upon application of a negative current pulse of sufficient amplitude to the base of T_1, the collector of this transistor will be driven to a saturation potential of -0.1 volt in time t_0. For the case shown, measured t_0 is 25 millimicroseconds (mμ sec) as observed with an oscilloscope having a total amplifier rise time of 16 mμ sec. Diodes D_1 and D_2 have been chosen to have minority carrier storage capability, for a given forward current, several times in excess of the storage capacity of the transistors due to excess injected minority carriers, for the same base current. Since D_2, a storage diode as previously described, has a forward charge potential of approximately 0.7 volt due to carrier storage, the base of T_2 will be driven rapidly toward a positive potential of approximately 0.6 volt by the collector of T_1 going into saturation. During this period, part of the charge stored in D_2 provides a recombination charge for the minority carriers stored in the base region of T_2, thus reducing the storage time t_1 of the previously conducting transistor to a minimum. In the illustration shown, the measured value of t_1 is about 15 mμ sec. Minority carriers stored in diode D_2 in excess of those lost in discharging the carriers stored in the transistor and those due to natural recombination are discharged rapidly via high-conductance germanium diode D_6. Since the base of T_2 is rapidly switched to the back-biased state, the collector fall time of T_2 is also reduced to a minimum. (For the case shown, the measured value of t_2 is 25 mμ sec.) The swing at the collector of T_2 is limited by the forward conductance of D_1 and the base-emitter junction of T_1, thus terminating the transient condition of the transistors. The flip-flop circuit may be efficiently retriggered, this time by applying the current pulse to transistor T_2, as soon as the excess carriers in D_2 have been dissipated. Since the storage diodes have been selected to provide a minimum charge in excess of the worst case for carrier charge storage in the transistor for the same current, a designer must specify an upper limit for the minority carrier storage capabilities of a storage diode, as well as a lower limit if very high repetition rates are required. The Hughes type *HD6172* storage diode, indicated in Fig. 4, will have sufficiently recovered within 2 μ sec (microseconds), since the particular circuits under consideration had a maximum trigger frequency requirement of only 500 (kc) kilocycles. This specification includes diodes with a total charge storage for a forward current of 1 milliampere (ma) in the range of approximately 1 $\times 10^{-9}$ coulombs and provides a yield of about 80% when selected from the standard-type *IN456* high-conductance silicon junction diode stock. The maximum charge storage due to minority carriers injected at an excess base current of 1 ma for the transistor used (type *2N383*) is approximately 0.2×10^{-9} coulombs.

Fig. 6 shows the effect of the relative storage capability of two different cross-coupling diodes. In the case of the diode selected for lower storage, the diode sufficiently recovers in about 0.5 second discharging into the base clamp diode. In the case of the higher storage diode (total charge storage for a forward current of 1 ma being 3×10^{-9} coulombs) sufficient recovery occurs in approximately 1.5 seconds. By properly specifying diode and transistor storage characteristics, operation at trigger rates of 5 megacycles has been accomplished using type *2N393* transistors. Much higher rates are possible with a diffused base transistor (type *2N501*) and with so-called "fast" silicon diodes to provide the desired cross-coupling characteristics.

In terms of circuit performance, the storage diode coupling technique provides several significant advantages over and above the rapid transient response obtainable through this technique. Excellent d-c stability results since the "on" transistor may be operated very hard in saturation and the "off" transistor is back-biased by an effective current source clamped by the base clamp diodes. Current mode operation is achieved in

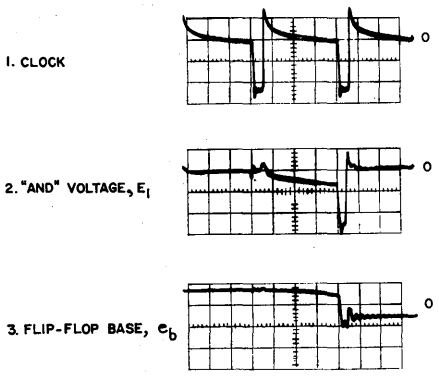


Fig. 8. Flip-flop gate wave forms (expanded). Vertical deflection = 1 volt per cm. Horizontal deflection = 0.2 μsec per cm. Trigger rate = 1.25 mc

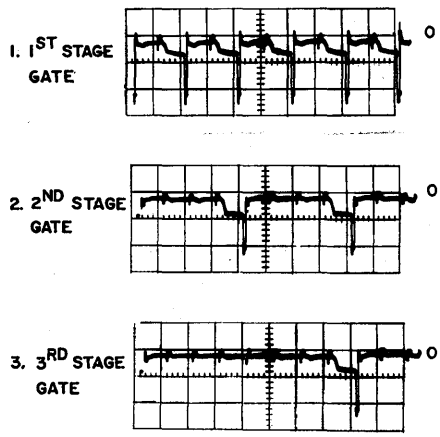


Fig. 9. Three-stage binary counter gate wave forms. Vertical deflection = 1 volt per cm. Horizontal deflection = 1 μsec per cm. Trigger rate = 1 mc

switching the current in the collector load resistors between the saturated transistor and the cross-coupling diode. Since the desired cross-coupling turn-off charge is developed as a function of forward current in the cross-coupling diode, low collector voltage swings in the order of 1.5 volts are possible. If low input capacitance is desired in the conventional resistance capacitance (RC)-coupled flip-flop, the voltage swing is generally in the order of 4 volts, using similar transistors. The cross-coupling diode also provides very efficient base triggering of the flip-flop, since the base input impedance of the "off" transistor does not reflect the collector characteristics of the saturated transistor to any extent. This is due to the fact that the cross-coupling diode in the triggered output will be recovered or "open" in the reverse direction. The only shunting of the input pulse will be due to the diode capacitance which is in the order of 20×10^{-12} farads.

Current-Charge Mode Gating

When considering a logical gating structure for a computing system, problems of compatibility between the gating and storage, or active, elements in the computer often arise. The gating structure within the system generally accepts the outputs of the storage elements, whether these be flip-flops, pulse amplifiers, or d-c amplifiers, and combines these, in accordance with the Boolean equations describing the system, to produce a logical output. This output then must be applied as the input of a storage or active element. When mechanizing a binary (2-state) logical structure, factors of importance in the gating techniques include:

1. Gating efficiency and the compatibility of the gates with storage and active element outputs.
2. Signal discrimination or the ability of the gates to detect the difference between a true and false proposition.
3. Compatibility of the gate output with the storage and active element input requirements.

When considering transistors for use in the storage and active elements, several facts are immediately evident concerning their input and output characteristics if they are to be operated as switches; i.e., in either a saturated or cut-off state with a required minimum transition time between these states:

1. To "turn on" a transistor in time t_0, the driving source must be capable of supplying a minimum current sufficient to charge the input capacitance and also to drive the transistor into current saturation.
2. Steady-state saturation requirements may be met by supplying a constant-current drive to the base.
3. The collector output characteristic of a grounded-emitter saturated stage provides an excellent constant voltage current sink.
4. To turn off a saturated transistor in time t_1+t_2, sufficient charge must be drawn from the base to reduce rapidly the minority carriers during t_1+t_2.

In the computer logical structure under consideration, a number of types of ac-

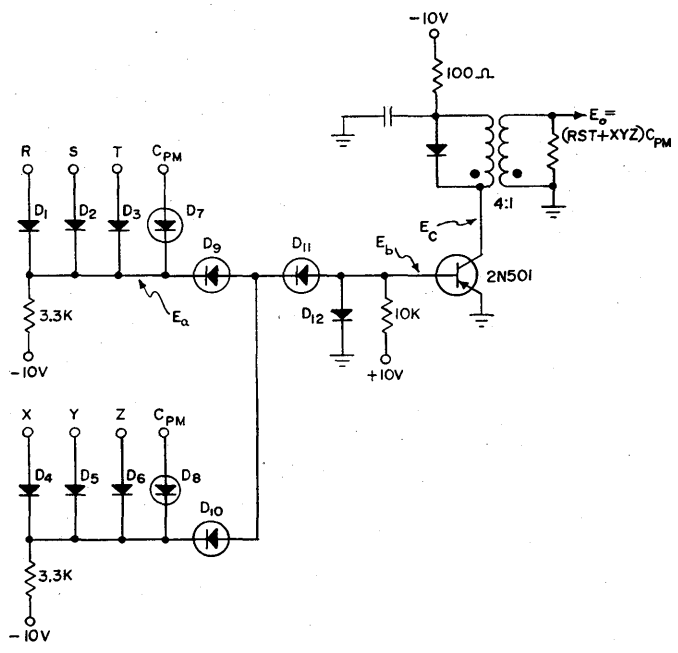


Fig. 10. Current-charge gated logical pulse amplifier

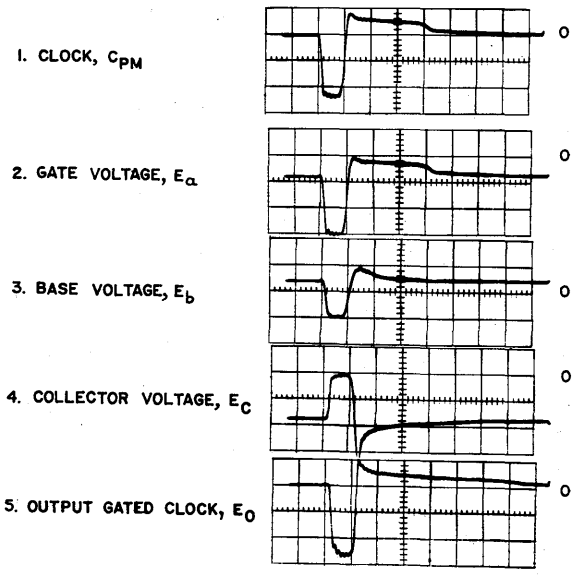


Fig. 11. Logical pulse amplifier wave forms. Vertical deflection = 1 volt per cm (1, 2, 3, 5) = 5 volts per cm (4). Horizontal deflection = 0.1 μsec per cm. Trigger rate = 1 mc

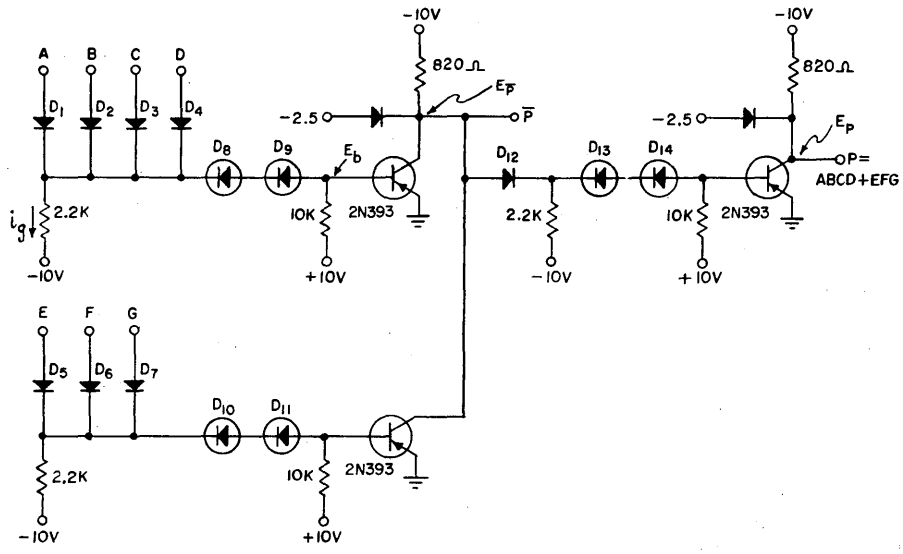


Fig. 12. Diode-transistor logic using storage diodes for fast turn-off

tive elements, or building blocks, are used. These building blocks differ in regard to their input requirements and output characteristics.

By employing high-storage silicon junction diodes in combination with fast diodes, a logical proposition may be gated in a manner compatible with the output characteristics and input requirements of the flip-flop previously described. Fig. 7 is the circuit diagram of a logical proposition mechanized in this manner.

As shown, the flip-flop trigger proposition is $ABC(P_1)+DE(P_2)$. Terms (P_1) and (P_2) represent timing clock pulses which may be the main clock and occur every period, or may be the amplified result of still other logical propositions gated with the main clock and occurring only as these propositions become "true." In this application, the clock signal swings from zero to -2.5 volts with a pulse width of 0.1 to 0.2 μ sec. Propositions A, B, C, D, and E are normally developed by saturated or cutoff output amplifiers of flip-flops or d-c proposition amplifiers and inverters, as indicated. The proposition signal swings from saturation (approximately -0.1 volt) to a clamped cutoff level chosen as -2.5 volts. The saturated level is chosen as the zero or "false" proposition state while the cutoff level is the true state of the proposition. In this logical mechanization, "and" proposition diodes D_1, D_2, D_3, D_4, and D_5 are high-conductance germanium units chosen for their low forward voltage drop. Clock diodes D_6 and D_7 are high-conductance silicon units chosen for their minority carrier storage characteristics, as previously described. The "or" proposition diodes D_8 and D_9, as well as flip-flop input diode D_{10}, are silicon units chosen for their forward conduction threshold and fast recovery characteristics.

It is noted that, with the exception of the clock diodes, the gate illustrated in Fig. 7 is a normal current mode gate with the current source supplied from B-minus through resistors R_1 and R_2. In the current mode gate, diodes D_6 and D_7 would be types similar to all proposition diodes. In the illustrated gating configuration, however, operation occurs in the following manner; first, consider only the upper "and" term of the "or" proposition when one or more propositions are in the saturated or false state. Under these circumstances, voltage E_1 will be at a level of approximately

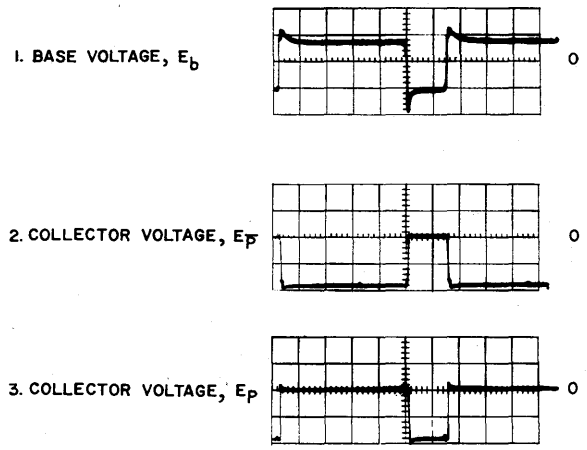


Fig. 13. Signal wave forms for diode-transistor logic. Vertical deflection = 0.5 volt per cm (1) = 2 volts per cm (2, 3). Horizontal deflection = 0.5 μsec per cm. Clock rate = 1.25 mc

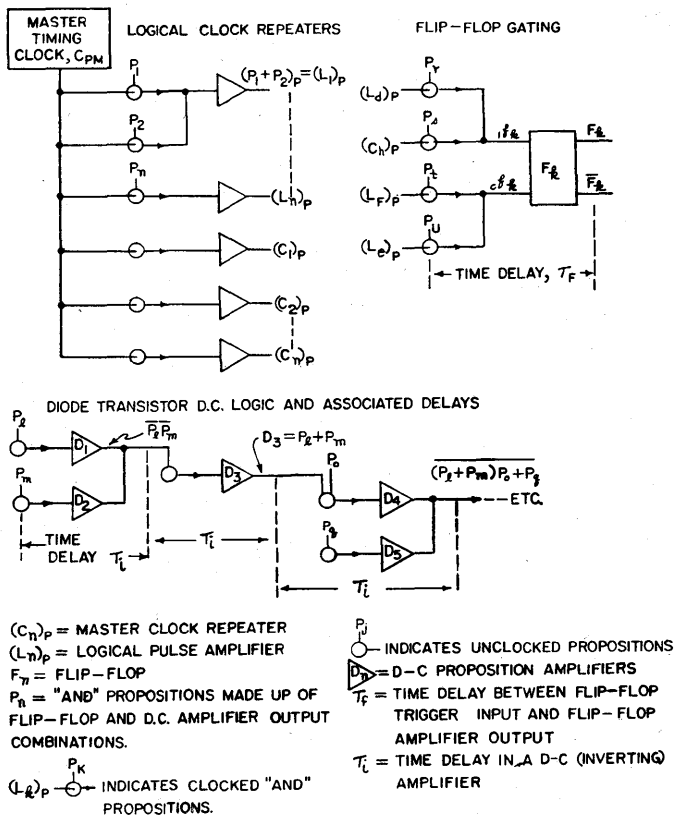


Fig. 14 (right). Computer logic circuit organization

-0.35 volt for a current source of 2 ma through R_1. This level is determined by the forward drop in the high-conductance proposition diodes and the saturation voltage of the false propositions. One or more of the saturated proposition drivers will be supplying the necessary current to maintain this false or inhibiting level at E_1. Diode D_6 will be nonconducting since it is a silicon unit with a forward conducting threshold of about -0.60 volt. If a clock pulse (P_1) now is applied, D_6 merely reverse biases rapidly, since no minority carriers are stored in this diode. The only charge coupled to the "and" junction of the gate (i.e., at E_1) will be due to the small amount of capacitance in D_6. This charge is quickly absorbed by the propositions in the false state. This effect is shown for the inhibited case in the oscillograms of Fig. 8.

Assume now, that all propositions (A, B, and C) are in the true state. In this condition, clock diode D_6 will be conducting the total gate current (2 ma for this application) and a minority carrier charge will be stored in the diode. Voltage E_1 will be at about -0.65 volt, a level below the conduction thresholds of approximately -1.35 volts for the series input diodes connecting the "and" gate to the flip-flop unit. Upon application of the clock pulse P_1, voltage E_1 will rapidly be driven negative until the input conduction threshold is reached. At this time the clock diode will discharge into the input of the flip-flop, since the clock diode is back-biased. The second pulse in Fig. 8 indicates the result of a true proposition. The total charge Q_T delivered into the flip-flop and associated stray capacitance is given by

$$Q_t = i_g \Delta t + Q_{st}$$

where:

i_g = the current in the gate resistor (considered constant)

Δt = the clock pulse width

Q_{st} = the charge transmitted by driving the clock diode in the reverse direction for a time Δt with the peak inverse current limited only by the amplitude of the clock pulse and the series impedance of the circuit. The series impedance includes the clock pulse generator impedance, the instantaneous reverse resistance of the storage diode plus the impedance looking into the flip-flop.

In practice, the charge transmitted into the flip-flop using the combination current-charge gating technique (as compared to a straight current gate) is about three times greater at the signal levels under consideration in this application. Upon termination of the clock pulse, any charge left in this clock diode will be discharged by "false" going propositions. Fig. 9 shows the voltage wave forms at the "and" junction of three current-charge gates operating at 1 (mc) megacycle as binary counter logic and using a 0.1-μ sec clock pulse.

Triggering a flip-flop using a combination current-charge gate offers a number of advantages over the straight current mode of operation. This is especially true when gating very narrow clock pulses using diode logic. For equal direct gate currents when using storage diodes, higher stray capacitance may be tolerated in the outputs of gates before malfunction occurs. Or, from another point of view, for a given input charge requirement, the gate direct current may be reduced, using the combination method. This enables a flip-flop output to drive more gates. Or, from still another point of view, flip-flop input trigger requirements may be made stiffer, thus reducing the susceptibility of the flip-flop to triggering on random transient noise. This is accomplished without reducing the number of gates which the flip-flop outputs may drive. The end result in each case is higher gating efficiency.

The added "booster charge" obtained by using this technique must be paid for, however. This additional charge must be supplied by the clock pulse driver. But the over-all power consumption of a computing system may still be considerably reduced, since clock power generally is made available at very high duty cycles, e.g., 10 to 1 or greater, when working at clock frequencies below 500 kc.

Current-Charge Gated Logical Clock Amplifiers

In large transistorized computers operating synchronously, total peak clock power requirements often are quite large. This demands either a vacuum-tube clock source, or multiple-transistorized clock amplifiers or repeaters enslaved to a master clock generator. The former often is undesirable because of the special requirements for power supplies, cooling, etc., not needed with transistor circuitry. Still another objection to supplying the total clock power, from one source is the problems encountered in transmitting very high-current pulses of very narrow width. The second approach is the commoner even though it is inefficient in terms of the total component count.

A more efficient technique, if over-all performance of a clock amplifier is considered as a measure of efficiency, is to provide clock repeaters in which the output pulse is controlled by logical propositions of flip-flops and d-c amplifiers, as well as providing clock amplifiers which repeat the master clock "one to one." These logical clock repeaters, generally amplifying clocked control propositions, then may be used to "and" with, and to clock flip-flop gates.

Fig. 10 is a diagram of such a logical clock repeater controlled by the proposition $(RST+XYZ)C_{pm}$. Here, C_{pm} is the master timing clock. As with the previously described flip-flop gates, all proposition diodes are high-conductance germanium while the clock diodes, D_7 and D_8, are chosen for their minority carrier storage capability. In this case, unlike the flip-flop gate, "or" diodes D_9 and D_{10}, along with input diode D_{11}, are also silicon junction storage diodes. In the case of the flip-flop gate, these were fast silicon diodes chosen for their input threshold characteristic and ability to "disconnect" rapidly upon termination of the gated clock pulse. Fast disconnect is not desired in the input to logical clock amplifiers.

The action in "turning on" the logical pulse amplifier of Fig. 10 is identical to the action described for the flip-flop gate. During this clock pulse duration, however, diode D_9 or D_{10} and diode D_{11} store minority carriers several times in excess of the carriers stored in the transistor base region. Upon termination of the gated clock pulse, therefore, the voltage $E_a - E_b$ (for the case where proposition RST is active) is about -1.3 volt, where E_b is equal to the base-emitter drop in the "on" transistor. Diodes D_9 and D_{11} are effectively charged to a potential of about 1.4 volts, a value dependent upon the instantaneous reverse characteristics of the diodes when subjected to the forward current during the clock width, Δt. Since the charge storage capabilities of these diodes greatly exceed that of the transistor used, E_b will be driven rapidly to a potential of about $+0.6$ volt during the positive transition of the clock pulse, rapidly discharging the carriers stored in the transistor and reducing its storage time (t_1) and fall time (t_2) to a minimum. Fig. 11 shows oscilloscope wave forms at points of interest in this circuit in which 0.1-μsec pulses are being logically gated. In the circuit shown, the output current pulse is in the order of 150 ma at 2.5 volts.

Diode-Transistor Logic Using Storage Diodes for Rapid Turn-Off

Still another useful application for storage diodes is in connection with

diode-transistor logic. Fig. 12 shows a typical gating arrangement in which "and" propositions are formed, using conventional current gating, and are combined using transistors for the "or" mechanization. Rapid turn-on time and assured saturation are obtained by providing sufficient current drive in the "and" gate. Excess base current in the "or" transistors, of course, causes increased storage time (t_1) and turn-off time (t_2). This problem is minimized in this circuit by the same technique previously described for turning off logical pulse amplifiers. The storage diodes indicated serve the dual purpose of signal discrimination and providing the transistor turn-off charge. Fig. 13 shows the base wave form of one of the "or" transistors resulting from a current drive of about 5 ma in the "on" direction. For the collector currents indicated, the transistor will be driven extremely hard into saturation. The base wave form, E_b, shows the effect of the base characteristic before injection of carriers and also the effect of the storage diodes in rapidly discharging carriers in the base region at turn-off time. The wave forms in Fig. 13 were taken at a clock rate of 1.25 mc and indicate a delay of about 30 mμ sec per amplifier (Type *2N393*).

Computer Circuit Organization

A few comments are in order concerning the organization of the gating technique described in a complex computing structure.

As noted in previous discussions, two basic types of logical "and" propositions are possible and generally are a requirement in computers. The clocked "and" proposition is used for triggering a flip-flop or logical clock amplifier. The trigger clock in the case of flip-flop logic may be any logical clock or repeated main clock. One clock, however, multiplies each "or" term of the flip-flop input. Fig. 14 illustrates a typical combination. "And" propositions gating the inputs to logical pulse amplifiers are always referenced back to the master timing source, C_{pm}. This prevents the accumulation of delays or the staggering of clock pulses which trigger flip-flops. The second type, or unclocked "and" proposition, is used for driving d-c amplifiers which may be combined as illustrated.

The techniques described have been applied to computers having both serial and parallel arithmetic structures. The complexity of the logic and clock repetition rate determine to a great extent the arrangement of the various building blocks. For example, in the case of a 20-bit parallel arithmetic unit, the propagation of the "carry" for the binary accumulator is readily accomplished at a 200-kc clock rate using the diode-transistor logic described. Here, assuming the worst case where the delay is 0.15 μsec per carry stage, all d-c amplifiers will have settled down in 3μsec, thus allowing ample safety factor in clocking the accumulator register to produce an addition in 5 μsec. Using the storage-diode flip-flop trigger technique, it is noted that a few tenths of a microsecond are required to discharge the clock diode into the proposition for the case of the proposition going false before clock time. Also, for the clock storage diode to be effective, a proposition must go true a few tenths of a microsecond before clock time so that the gate current may be integrated to provide sufficient carrier injection in the diode. For performing this current integration, the storage diode, unlike a capacitor, is extremely nonlinear in that it becomes saturated in the forward direction in a few tenths of a microsecond, operating at the gate current of 2 ma.

Conclusions

In this paper, techniques have been described which utilize the inherent minority carrier storage effect exhibited by high-conductance silicon junction diodes to improve transient response in digital switching circuits. Attempt is made to combine both slow and fast diodes with transistors in a manner which most effectively complements the fundamental characteristics of each semiconductor.

References

1. Handbook of Semiconductor Electronics, **L. P. Hunter.** McGraw-Hill Book Company, Inc., New York, N. Y., 1956.

2. Large Signal Behavior of Junction Transistors, **J. J. Ebers, J. L. Moll.** *Proceedings*, Institute of Radio Engineers, New York, N. Y., Dec. 1954.

3. Charge Storage in Junction Diodes, **Earl L. Steele.** *Journal of Applied Physics*, New York N. Y., vol. 25, no. 7, July 1954.

4. Large Signal Transient Response of Junction Transistors, **John L. Moll.** *Proceedings*, Institute of Radio Engineers, New York, N. Y., Dec. 1954.

Discussion

Chairman Ruhman: The first question for Dr. Poppellaum is from A. I. Gordon, Hughes Aircraft Corporation: "What would the necessary voltage swing be for a Schmitt-trigger?"

W. J. Poppelbaum: About 20 volts.

Chairman Ruhman: From J. H. Lane, ElectroData: "Just why do the transistors require such relatively high betas?"

W. J. Poppelbaum: The high beta is required so that you do not have too much loading between supply and the ground. It just limits you in the low rate because you have too much reflective on the base side.

I should point out, incidently, that these flip-flops at the University of Illinois are first open to mated, a certain relations method. Then you will find out that under the worse cores and conditions the thing still works.

Chairman Ruhman: from Mr. Klein, Magnavox: "You explained how the signal flow is gated; would you mind explaining how the desired drivers are selected? What kind of gating do you use for them?"

W. J. Poppelbaum: You use the output.

Chairman Ruhman: From H. Tweeden, General Electric Company: "What is the over-all delay of this circuit?"

W. J. Poppelbaum: The over-all delay is 15 micromilloseconds. This is ficticious, of course, as it takes time for this kind of a swing.

Chairman Ruhman: From F. W. Springe, Hughes Aircraft Corporation: "What are the space (packaging) considerations of low gating compared to gating with the regular Eccles Jordan?

W. J. Poppelbaum: I should actually say that this system was developed to provide very small buffer registers. We do have plug-in units. There were 15. One half unit by a quarter by a quarter.

Chairman Ruhman: From P. Writer, Litton Industries, Inc.: "What methods are used to connect leads to the modules? If you use connectors, what kind?"

Arney Landy, Jr.: We have used the flexible lead, just as in a transistor placed in the circuit board. With regard to connectors of the body to circuit, there is work going on at Honeywell regarding soldered connections. I can see by Mr. Writer's questions, that he too is concerned by the connections, number of them, and number of components and also getting a good connection.

Chairman Ruhman: From A. I. Gordon, Hughes Aircraft Corporation: "What is the dissipation derating factor for compound used? How are resistors derated? Does this element meet Mil-specifications?"

Arney Landy, Jr.: I believe your first question refers to billing compound or plotting. We intend to use the 10 wad or 8 wad.

Does the element meet the Mil-specifications? I am not sure, but I see no reason why it will not.

Chairman Ruhman: The first question for Mr. Dunnet comes from Mr. Fink, Hughes Aircraft Corporation: "What type of cores have you been using in this circuit?"

W. J. Dunnet: One-megacycle design that we built in the laboratory. We used 18,473 crips without much iron.

Chairman Ruhman: From V. W. Vorndran, Hughes Aircraft Corporation: "What type of core is used? What are number of turns? What are control and clock current amplitudes?"

W. J. Dunnet: I could use almost any number there. As a matter of fact, we have. The most might have 20 turns. About 10 on the input windings. We are planning to use something like 500 mils to reset the cores.

Chairman Ruhman: From W. E. Ross, Litton Industries Inc.: "Why not take advantage of regeneration by using a collector winding?"

W. J. Dunnet: I do not know if I can answer that question. You do not get much by regeneration of reverse write where they use regeneration. It might cause more problems, or we might gain by it. I do not know. Actually, we have not been too concerned with it. I would like to think about it for a while before I even consider it.

Chairman Ruhman: I would like to ask a question myself. "Could you, very briefly, indicate what you think the advantages are of this circuit over some of the other types of magnetic circuits as have been used for logic?"

W. J. Dunnet: The advantages would be the advantages I showed on the board. The simplicity of the thing.

Perhaps, in 6 months or a year we will be able to say more as to just what the advantages are, after we have actually built one.

A Chess Playing Program for the IBM 704

A. BERNSTEIN
NONMEMBER AIEE

M. DE V. ROBERTS
NONMEMER AIEE

T. ARBUCKLE
NONMEMBER AIEE

M. A. BELSKY
NONMEMBER AIEE

THIS paper describes the program which enables a human being to play a game of chess with the International Business Machines Corporation (IBM) *704* computer. The machine may play either white or black, and is capable of playing a complete game of chess, including such moves as castling, promoting, and capturing en passant.

The program is divided into five parts: (1). Input-output, (2). Table generation, (3). Evaluation, (4). Decision, and (5). Tree

Input-Output

The input-output section of the program allows the human player to state his moves to the machine in the normal English notation on punched cards, for example: P-K4 or B(Q4) × P(N7). The program then translates this statement into the machine notation for moves, which gives the "to" square and the "from" square with a code. A move in the machine notation may also be entered from the keys on the console.

The squares of the board are numbered octally from 00 in the upper right-hand corner of the board to 77 in the lower left. For a regular move, the tag field contains a 1, the decrement contains the "from square" co-ordinates, the address contains the "to square" co-ordinates, and the prefix contains a minus sign if the move is a capture. For a castling move, the tag field contains a 2, the decrement contains the "to square" of the king, the address contains 77 if castling up, 0 if castling down, and the prefix is 0. (Castling upwards means that the final position square of the king has a higher number than its initial square.) For an en passant capture, the tag field is 3, the decrement contains the "from square" of the capturing pawn, the address contains the "to square" of the capturing pawn, and the prefix contains a minus sign. For a promotion, the tag field contains a 4, the decrement contains the "from square" of the pawn that promotes, the address contains the "to square" on which promotion occurs, and the prefix contains a minus sign if the move is a capture.

At the start of a game the machine's pieces always reside on the squares 00 to 07, 10 to 17; the opponent's on the squares 60 to 67, 70 to 77.

If the machine were to play white and make P-K4 as its first move, in machine notation this would be the octal word, 000014100034.

When the machine has made a move, or accepted an opponent's move, it prints that move in the English notation, together with a picture of the board position, and makes a record of the move on tape. At the end of the game the full score is printed.

Table Generation

The table generating section accepts as its input a table of the board position which will be referred to as *TA*1. *TA*1 at the beginning of the game is arranged so that it represents the starting position; but any board position may be assembled into *TA*1 and entered as the starting position for the program.

*TA*1 is 64 words long, one word per square. The first word of *TA*1 refers to square 77, the last word to square 00. If the square is empty then the word is all zero. The word is negative if the square is occupied by a machine piece. The address contains a number indicating whether the piece is a king (5), queen (0), rook (1), knight (4), bishop (2), or pawn (3), and the decrement contains an indexing quantity to link *TA*1 with Table 2, which will be called *TA*2.

*TA*2 is now generated from *TA*1 in the following form. It is 32 words long, one per piece. The order is shown in Table I.

A word in *TA*2 is of the following kind: It is zero if the piece has been captured. Otherwise, the address contains the value of the piece, and the decrement contains the co-ordinate of the square where the piece is located. It can be seen that the decrement of *TA*2 is the indexing quantity of *TA*1.

Table 31 and Table 32, called *TA*31 and *TA*32 are also generated from *TA*1. *TA*31 refers to the machine and *TA*32 to the opponent. *TA*31 and *TA*32 are similar in nature and function. Thus it is sufficient to describe *TA*31.

*TA*31 is 512-words long, eight words per square. The first eight words refer to square 77 and the last eight words refer to square 00. Consider the eight words referring to square (*xy*). If the square *xy* is occupied, the eighth word is negative, and contains a bit in the first position if it is occupied by the machine but not if occupied by the opponent. The address of the eighth word contains:

1. The number of machine pieces which may move into square *xy* if *xy* is empty.
2. The number of machine pieces which may immediately capture an opponent's piece situated on *xy*.
3. The number of machine pieces immediately defending a machine piece situated on *xy*.

The location of the pieces described in the address of the eighth word are in the decrements of the words 8 through 1. The address of the seventh word stands for the number of doubled machine pieces bearing on square *xy*, e.g. a rook behind a rook bearing on *xy*, or a queen behind a bishop, etc. The location of these doubled pieces are to be found in the addresses of the words 6 through 1.

It will be seen that the location of the pieces bearing on *xy* is the indexing quantity needed to find that piece in *TA*1, while each square *xy* in *TA*3 is eight times the indexing quantity of *TA*1.

All possible moves are now listed in *TA*3, for each square has in it the number and locations of all pieces which may move into it.

The routine which generates *TA*3 also examines the position for any possible pieces pinned to the king, so that no illegal moves are listed in *TA*3.

*TA*2 and *TA*3 are generated in an average time of 89 milliseconds.

Evaluation

The evaluation routines calculate a score for the machine and the opponent based upon the following criteria:

1. Mobility
2. Area control
3. King defense
4. Material

A. BERNSTEIN and M. DEV. ROBERTS are with International Business Machines Corporation, New York, N. Y. T. ARBUCKLE and M. A. BELSKY are with The Service Bureau Corporation, New York, N. Y.

The program counts the number of available moves for each side; the number of squares completely controlled by each side; the number of controlled squares around each king; and the material count of each side.

These criteria are parametrized for easy variation, so that different values may be given to center squares, or to the squares around the king. At present criteria 1, 2, and 3 are added together while 4 is multiplied by a large factor before adding it to the total. This prevents the machine from sacrificing material, and encourages it to exchange when it is ahead, as it considers the ratio of its own and opponent's scores.

Decision Routines

The decision routines serve the function of producing a set of moves to be considered for further analysis. The idea behind these routines is to select a small number of moves which are potentially good (for strategic reasons) rather than to eliminate from all possible moves the much larger number which can be determined to be bad.

The decision routines examine the position and determine whether certain states exist; if the answer is yes, certain moves are generated. The questions asked are:

1. Is the king in check?
2. (*a*). Can material be gained?
   (*b*). Can material be lost?
   (*c*). Can material be exchanged?
3. Is castling possible?
4. Can minor pieces be developed?
5. Can key squares be occupied? (Key squares are those squares which are controlled by diagonally connected pawns.)
6. Can open files be occupied?
7. Can any pawns be moved?
8. Can any piece be moved?

If the answer to question 1 is yes, the machine asks itself whether it is checked by one or two pieces; this information is obtained by observing the square *xy* on which the king is located in *TA*32, the address of word 8 will be 2 if a double check exists and 1 if a single check. If it is a double check the only possible answer is a king move; the machine examines the empty squares around the king for legality of moves and if any of these moves are possible, the move is constructed and entered in the Plausible Move Table. The routine then returns control of the tree. If instead the king is in check by a single piece, interpositions of pieces, and the capture of the checking piece are considered, and if possible these moves as well as the king moves are entered in the Plausible Move Table. Control then returns to the tree.

Table I

1	Machine King
2	Machine Queen
3	Machine Bishop
4	Machine Bishop
5	Machine Knight
6	Machine Knight
7	Machine Rook
8	Machine Rook
9	Machine Pawn
10	Machine Pawn
11	Machine Pawn
12	Machine Pawn
13	Machine Pawn
14	Machine Pawn
15	Machine Pawn
16	Machine Pawn
17	Opponent's King
18	Opponent's Queen
19	Opponent's Bishop
20	Opponent's Bishop
21	Opponent's Knight
22	Opponent's Knight
23	Opponent's Rook
24	Opponent's Rook
25	Opponent's Pawn
26	Opponent's Pawn
27	Opponent's Pawn
28	Opponent's Pawn
29	Opponent's Pawn
30	Opponent's Pawn
31	Opponent's Pawn
32	Opponent's Pawn

If the answer to question 1 is no, the program goes down to question 2 and if the answer to 2(*a*) is yes, lists those moves which gain material in the Plausible Move Table; if 2(*b*) is yes, finds which moves will take the attacked piece to safety, enters them in the Plausible Move Table, and if 2(*c*) is yes, will enter those exchanges in the Plausible Move Table. Whenever the Plausible Move Table is filled, control goes back to the tree for further examination. At the end of question 2 if the Plausible Move Table is not filled, question 3 is asked. If yes, the castling move is constructed and entered in the Plausible Move Table. If the answer to question 3 was yes, then control goes back to the tree. If no, control goes to question 4, etc.

The reason for stopping the decision routine, if castling is possible, is that the castling move does very little to enhance the score of a position, but is nevertheless a very important element in bringing the king to safety. Therefore, whenever castling is possible no other alternatives except for material exchanges are given, and eventually, when there are no exchanges or pieces to be gotten out of attack, the program is forced to castle.

The ordering of these routines is important. At the beginning of the game questions 1, 2, and 3 are not applicable and questions 4 and 7 are the only ones that produce moves. In the middle game, questions 2, 5, and 6 bear the brunt of the work. In the end game, questions 5, 6, 7, and 8 are most employed. This ordering is arbitrary and will probably require adjustment as experience and more knowledge of the way the program plays is gained.

The Plausible Move Table is limited by time and coding considerations to seven.

The Tree

The tree is the master routine. It operates in the following manner: Having received a position, it asks for a score of the position, and for the set of plausible moves; executes the first move; gets a score and asks the decision routine for a set of plausible replies. It performs the first of these and repeats the procedure until it reaches the fourth level, where it performs each of the opponent's moves, obtaining a score for each. It now takes the maximum score of this last set of plausible moves and brings it down as the score of the move made on the third level, undoes that move on the third level, makes the second move of that set and requests plausible replies for the opponent. It performs each of these, gets their score, and brings back the maximum opponent score (its own minimum score) for the second move.

In this fashion it examines the full tree, and determines which move will maximize its score. With a plausible move set of seven, and depth of four half-moves, this examination takes an average of 8 minutes.

The tree has a selective mechanism which can cut some branches to be examined. This mechanism is based upon the relative scores of each of the plausible moves, and will function efficiently only after the scoring parameters are well adjusted. When this selective method is in operation only moves which are greater or equal in score to the position score before the move was made are allowed to be examined further (except when no moves are better, in which case it takes only the two highest scored moves).

At present this makes the total time for producing a move 2 minutes on the average. The program plays more conservatively in this mode, but does not allow certain moves which are definitely good, and which are otherwise considered and sometimes made.

Conclusions

The machine plays a passable amateur game at present during the opening and

middle game. It is weak toward the end, but the authors believe that additions to the decision routine will remedy the situation. There are several other routines that could be expanded, such as defending pieces rather than moving them away when attacked. All of these are, however, demonstrable, and add little to the entire scope of the program.

Self-adjustment routines are being considered, to change the value of parameters after the loss of a game, and to change the ordering in the decision routines, but these are still in the talking stage. While it is true that the machine makes the same move given in the same position, there are so many positions, that its play is not predictable in a new situation.

Applications of Digital Computers to Problems in the Study of Vehicular Traffic

WALTER HOFFMAN
NONMEMBER AIEE

RICHARD PAVLEY
NONMEMBER AIEE

THE study of vehicular traffic on a scientific basis is still in its infancy. This is due to the enormity of the job. Currently, research in this area is carried on along two distinct lines. First, there is the approach which attempts to study traffic conditions in the small and gradually builds up to larger systems. This is an approach which may accomplish the desired goals, but from all current indications success in this area is distant.

The second approach is a study of traffic problems in the large, meaning a major metropolitan area. Clearly, the methods which are being used today are relatively crude; but constant research for new methods and improvement of the old ones had led to results which have been verified in actual traffic situations, and the predictions made have agreed remarkably well with the observed data.

The research reported in this paper is based on data obtained by the Detroit Metropolitan Area Traffic Study in 1953. At that time a survey of existing traffic conditions was made under the direction of Dr. J. Douglas Carroll at a cost of one million dollars. The survey consisted of three parts: First, a suitably large sample of the population of the metropolitan area was interviewed at home and information was obtained as to the origins and destinations, nature, and frequency of the trips which were made. Second, the major truck and taxi companies were consulted and the same kind of information was obtained from them. Third, roadside interviewing stations were set up and vehicles were stopped and drivers were asked the origin and destination and nature of their trips.

The metropolitan area which was studied included the city of Detroit and its suburbs, covering three counties, and extended far enough beyond them to include those areas which could reasonably be expected to play any part in the metropolitan Detroit area traffic picture at the end of the forecast period, which was set at 1980. The area was divided into 265 zones, and the survey data were expanded to give the total number of trips between any pair of zones for an average 24-hour period. These results were then checked by comparing actual traffic counts across screen lines against computed traffic volumes. It was found that the data obtained from the traffic survey were uniformly about 10% low, and suitable adjustments were made. Thus, an accurate picture of the existing traffic flow had been obtained and the data were presented in the form of a matrix of order 265.

The assumption was made that any trip which originated in the metropolitan area would retrace itself within a 24-hour period so that the matrix mentioned was symmetric with nonnegative integers as entries. Thus, the element a_{ij} of the matrix represents the number of vehicles which make a trip between zones i and j.

The first problem which presented itself was the prediction of this matrix for some future date. It is well known that the trip generating characteristics of the given area depend on land use and on the population of the area. Thus, it was a relatively simple matter to predict the growth of the number of trips for each of the 265 zones. The difficult problem is the distribution of the total number of trips originating or terminating in a given zone over the remaining zones. There are a number of methods in use which attempt to cope with this problem, all of which are imperfect. The problem can be stated mathematically as follows: Given a real symmetric matrix with nonnegative integral entries, find a matrix of the same kind whose row sums are prescribed. Unfortunately, this mathematical problem does not have a unique solution. Its formulation thus must be revised to take into account the characteristics of traffic flow. All of the methods currently used operate in a similar fashion.

If the predicted trip volume between zones i and j is denoted by a'_{ij}, then this quantity must be influenced by the growth factors g_i and g_j of both zones which are involved. Thus a_{ij} must be multiplied by some mean value of the appropriate growth factors. In practice, a_{ij} is multiplied by both growth factors, after which an adjustment is made by dividing the product, which is thus formed by some normalization factor, K. The various methods employed differ in the choice of K. Regardless of how K is chosen, the new matrix obtained by this process will not have the correct row sums. To obtain a matrix of the proper kind the process is repeated, and after a few iterations convergence is obtained. The methods used vary in the number of iterations required for convergence. The method used at this laboratory was developed by Dr. Carroll's staff and has become known as the Detroit Method. The number K in this method is obtained by dividing the sum of all matrix elements at the end of any iteration into the predetermined sum of the elements of the final matrix. This method has been improved upon by choosing K as that factor which will give the total matrix sum as prescribed at the end of any iteration. It is clear that this will accelerate convergence.

Regardless of which method was used, this explains how the traffic picture for the entire metropolitan area is predicted.

A new problem is the presentation of this large number of individual data elements in an easily comprehensible form. The accepted means of doing this is the so-called trip-desire map. This is a map of the area divided into areas of equal traffic density by means of isolines. Production of such a map involves a great deal of labor. The usual procedure can

WALTER HOFFMAN and RICHARD PAVLEY are with Wayne University, Detroit, Mich.

be described as follows: The map of the area is overlaid with a square grid pattern. For each pair of zones a straight line segment is drawn which joins the centers of the two zones. The number of vehicles making the trip from zone i to zone j is counted for each square which is intersected by the line segment just described. This procedure must be repeated for each pair of zones before a trip-desire map can be constructed.

Trip-desire maps were constructed by this method in 1954 and 1955 by using the IBM (International Business Machines Corporation) *604* computer. The procedure used currently presents a vast improvement not only in that much more powerful computers such as the IBM *650* and *704* machines were used, but also in the basic method of obtaining the maps. The principal tool which has been adapted for this purpose is the so-called Moore algorithm, by which the shortest path joining any two points of a network can be found. This is accomplished within a computer in the following manner: One first chooses a node of origin and assigns to all other nodes an essentially infinite distance from the node of origin. Thereafter, the procedure employed attempts to reduce systematically these infinite distances by first replacing the distances of the nodes immediately adjacent to the node of origin by the actual distance and proceeding in an iterative fashion. When no distance can be reduced any further, there is a number associated with each node which represents the minimum distance from that node to the node of origin. To trace the shortest path from any node to the node of origin the following procedure is employed: The numbers assigned to the nodes adjacent to the one in question are compared with the number assigned to the node of destination to find which adjacent node has a value which differs from the value of the node of destination by exactly the length of the segment joining these two nodes. The node thus found yields the first link of the shortest path, and one proceeds in this manner toward the node of origin. It is for this reason that the final assignment of values to all nodes is referred to as a minimum tree. It will be recalled that a tree is a linear graph containing no cycles with the property that there is a unique path joining any two nodes of the graph. Clearly, to find the shortest path for all pairs of nodes it is necessary to construct a minimum tree for each node considered as a node of origin.

Computer programs have been written for both the IBM *650* and the IBM *704* to find all shortest paths joining all pairs of nodes for networks containing up to 99 and 1,000 nodes respectively. Trip-desire maps can be constructed using the Moore algorithm by replacing the square-mile grid mentioned before by a network whose nodes are the zone centers and whose segments are the lines joining the centers of adjacent zones. After all shortest paths are found, it is a relatively simple matter to assign the traffic volume making a trip between any two zones to the shortest path joining the respective zone centers, thus obtaining a trip-desire map which is essentially equivalent to the one obtained by the older methods. It took approximately three hours to obtain the minimum paths for a network of 265 nodes on an IBM *704*, and the assignment of traffic volumes to these minimum paths requires approximately two hours for each trip-desire map.

The Moore algorithm is also of extreme usefulness in the final problem reported in this paper, which is the assignment of traffic and the prediction of future traffic densities on networks of streets and expressways. Of particular interest is the prediction of street traffic on expressway facilities as yet in the planning stage. The procedure employed is the following: For each pair of zones the distance and time required to make the trip are computed for a surface street route as well as for a route utilizing the expressway network. The latter involves determination of the nearest ramp of entry of the expressway network as well as the nearest ramp of exit and proper speed estimates for the surface and expressway portions of the trip. Empirical evidence has been obtained which relates the percentage of the number of vehicles making a given trip which use the expressway system to the time and distance ratios. Thus, the number of vehicles using the expressways can be predicted, and computer programs have been written which simulate a traffic counter on each link of the expressway network. It is thus possible to obtain the future distribution and density of traffic on various proposed networks of expressways.

The first such projection which actually has been tested was performed at this laboratory in the summer of 1956. The problem concerned expected traffic loads on the John C. Lodge expressway, which was scheduled to be completed in the fall of 1957. The results indicated that this expressway would be overcrowded to such an extent that traffic would be slowed down considerably. Considering the nature of the methods employed in this prediction, the agreement of actual traffic conditions on this expressway with the predicted traffic densities is nothing short of amazing. It is therefore somewhat disturbing to look at the results of the latest prediction, which forecasts the traffic distribution on a much larger expressway network scheduled to be constructed within the next ten years. The predictions show that this network will be vastly overcrowded in 1980. What is more important is the fact that although these predictions cannot be viewed with the degree of confidence of earlier predictions due to the longer time involved, these predictions do point out the locations which will present the greatest bottlenecks.

Thus, it is possible to modify the network and reforecast future traffic distribution of several alternative networks until an optimal design is obtained. It must be pointed out that these investigations are expensive and require large amounts of expensive machine-time and personnel. However, when one considers that the cost of the projected expressway network is in the vicinity of one billion dollars, it does seem prudent to expend time and effort in an attempt to optimize such an expenditure.

The next problem to be solved is the determination and forecasting of traffic densities on the entire arterial network of the metropolitan area. The difficulties inherent in this problem are manifold. First, and most important, there is the magnitude of the system to be considered, which transcends the capacity of currently available commercial computers. Secondly, there are the difficulties caused by the lack of knowledge of driver behavior. It is clear that the method employed to predict expressway traffic densities is not suitable for the forecasting of traffic densities on the entire arterial network. The previous problem was solved by recognizing that only a fraction of the drivers making a given trip will use the expressway network, while the remaining portion will use the surface route. Thus, it is not sufficient to compute merely the shortest path joining any origin and destination, but a variety of paths must be computed and the traffic volume distributed amongst these paths according to some as yet undetermined criteria.

The problem of computing several best paths is considerably more difficult than that of merely computing the shortest path. Although this problem has been investigated by another organization which developed a computer program, the procedure employed at this laboratory is thought to be superior, primarily due to the application of the following theorem:

The $(N+1)$st best path is a deviation from some Kth best path where $K>N$.

A deviation from path P is defined to be a path which coincides with path P from the destination for a number of links which might be 0, which then contains exactly one link joining a node of path P to a node which is not on path P and which then proceeds to the origin via the minimum tree.

This theorem makes it possible to write a computer program which will economically find an arbitrary number of ranked best paths. Once these paths have been determined on the basis of distance alone, they will be examined for various other attributes which may have an effect on the frequency with which drivers will choose them. A partial list of such attributes is the following:

1. Travel time.
2. Number of intersections.
3. Number of turns.
4. Capacity of the links in the path (mean, maximum, minimum).

Clearly, some empirical tests will have to be made to determine a function of the criteria listed which could be used to determine route selection of drivers. Considerable numerical experimentation in comparison of computed results with the existing traffic conditions is being planned.

With the advent of better and faster computers, it seems highly possible that in the not-too-distant future, road planning will become an exact science.

The Role of the Digital Computer in Mechanical Translation of Languages

DAVID L. JOHNSON
ASSOCIATE MEMBER AIEE

THE PROBLEM of applying digital computers to the mechanical translation of languages is currently receiving world-wide attention. Projects are under way at three or four major universities and research centers in the United States; others in England, Italy, and Russia. Of particular interest to military intelligence, the problem is emphasized by the increase in communication speed relating one language to another. It becomes more and more difficult to find experienced translators with the specialized knowledge in various fields of science and technology that is required for adequate translation and to provide an interchange of knowledge regarding the progress made in different areas of the world. The vast quantity of this material to be translated lends urgency to the development of some high-speed means of translation. An obvious means of satisfying this need lies in the application of high-speed digital and logical equipment. The solution of the mechanical translation problem by means of digital computers is one of continuing importance and extreme interest to those who would follow unique and interesting digital computer applications.

As is true in many computer application problems, the most difficult and frustating element of machine translation is the definition of the means of approach and the structure of the problem itself. In resolving this aspect of the solution, skilled linguists must necessarily play an extremely important part. Language is formed by usage; formed organically developing a complex hierarchy of rules. Although the rules are inviolate, they are extremely difficult to specify and bring to conscious realization. At the University of Washington, the MT (Mechanical Translation) Project has progressed by means of closely coordinated efforts between the linguistic and engineering members of the staff. The work, in many respects, has followed the familiar research and development pattern; theoretical development has been followed to a certain point, the method is automatized and tested, then analysis of results yields further advances upon the theory.

The mechanical translation problem can be easily stated. It is necessary that a high-speed means be found to provide accurate and intelligible translation from one language to another. The MT project at the University of Washington deals with translation of general technical literature from Russian to English. The problem can be grossly divided into two parts. A computer memory or dictionary must be available for use in the translation process. The dictionary output must then be logically processed to provide a clear output translation.

The requirement of a dictionary or lexicon is obvious. The need for logical processing of the dictionary output is evident when the form of a normal dictionary output is considered. In human usage of a bilingual dictionary a single input word in the source language almost invariably yields multiple output words in the target language. Combining the effects of multiple target alternatives and other grammatical problems, the raw dictionary output is one that may be accurate; but the intelligibility is subject to considerable interpretation.

Dictionary Storage

Most groups working with the mechanical translation problem are approaching the computer application from the standpoint of using available general-purpose digital computers with their associated storage facilities as the translation dictionary. While this may be acceptable from the criteria of performance and output, the access to memory for a large-scale dictionary scanning process is one involving considerable time; and the magnitude of the dictionary itself may be seriously limited by available storage space. The University of Washington group has based their operation upon the use of the high-speed photoscopic memory developed specially for this application by the International Telemeter Corporation. This memory device has a permanent storage of 30×10^6 bits with a random access time on the order of 0.05 second. The magnitude and speed of the memory has several significant implications on the basis of dictionary content.

The contents of the translation dictionary is something that is subject to several approaches. If the dictionary memory is seriously limited in size, it is sometimes considered advisable to dissect words into stems, endings, and prefixes; using logical operations to assemble the related parts of a single word into the target language. In cases when memory limitations are not considered, a saving of both logical

---

DAVID L. JOHNSON is with the University of Washington, Seattle, Wash.

The author wishes particularly to acknowledge the help of Mr. R. E. Wall in organizing the content of this paper. Much of the specific work described was accomplished by Dr. L. R. Micklesen of the Department of Far Eastern Languages and Mr. R. E. Wall, assisted by Messrs. U. K. Neihaus and A. D. Stathacopoulos, all of the Electrical Engineering Department of the University of Washington.

circuitry and search time is effected by the storage of total words in the dictionary. This same general concept of complete storage can be extended to include many idioms and common phrases which may be unitized in both languages which are being considered. Thusly, in the University of Washington dictionary, stems are repeatedly stored with different endings and prefixes; many idioms are entered as single words, and dissection is minimized to conserve search time.

The determination of the contents of the dictionary is primarily a linguistic problem and one that need not be detailed herein. It is sufficient to state that over 110 scientific Russian texts from 40 fields of science and technology were sampled. When the word lists from the texts were expanded into various grammatical forms, the dictionary contained on the order of 150,000 Russian entries. The number of associated English equivalents for these entries was on the order of 500,000 words. Superficial examination of the lists of alternatives for given source entries gives evidences of redundancy. The equivalents however, were carefully analyzed by both linguists and scientists in the field to minimize the number without destroying possible meaning.

Ordering of the entries in the memory is in an inverse alphabetic means within given first letters of the Russian alphabet. Initial letter ordering is made on a frequency basis. The inverse alphabetization allows the search procedure to recognize complete words without unpremeditated dissection; i.e., in matching English, the procedure would find "teasing" before "tea" and then "sing."

In order to enter the logical processing operations following the dictionary output, it is necessary to include as a part of the dictionary entries information related to the grammatical usage of the particular entry and certain subroutines required for the processing of the word in question. This information is stored in "tags" which are considered as a part of the target equivalent in the dictionary. The dictionary output then yields both English equivalents and the data tags containing grammatical and related information. Further specification of the use of the processing tags will be made later in this paper.

Although it is premature to determine the number of words required for the translation of a language, preliminary investigation indicates that something on the order of 50×10^6 bits of storage will be required for the language alone with perhaps another 40×10^6 bits required for the related grammatical tags. This is an extremely controversial subject among MT workers and linguists in general; so any figures must be taken as the roughest of approximations.

Logical Processing

The most difficult part of the general mechanical translation problem from the computer application standpoint is that dealing with the improvement of the dictionary output. Among those who are working on actual output processing as opposed to pure linguistic research, something of a common approach seems to be evolving.

At the present phase of development. the dictionary output must be on a word-for-word basis. The "word-for-word" nomenclature is misleading, as has previously been stated, for a single word in the source language usually comes through in the form of several possible equivalents in the target language. It might be better stated that the dictionary output is in most cases obtained considering one word of the source text at a time. In order to reduce redundancy, select among multiple equivalents, specify ambiguities, and insert words which are included in the usage of one language and not another, the output must be considered in context in blocks of several words at a time. General-purpose computers or special-purpose processing computers must be used to accept the dictionary output, store portions of that output, and by consideration of certain linguistic and logical procedures modify the output to clarify the intent of the author in terms of the target language.

Specifications of Processing Operations

The techniques used in the specification of the required processing are of significance in the total logical processing operation. The first step in prescribing the processing is linguistic analysis of text material to determine where ambiguities exist and precisely what features of the sentence syntax uniquely specify the exact intended meaning. To facilitate the specification of the tests for computer programming, a system of notation has been developed at the University of Washington MT project. For example, the dictionary output of a given Russian verb may be in the form

(they) are-chosen/taken-out/get

in which parenthesis indicate that the word may or may not be used in a given context, slash bars indicate that one of the alternatives must be used. The linguist tells the engineer that in the given syntax of the text being examined this may be either a presonal or impersonal verb-form class in the third person plural with no preceding nominative expression, but with a following nominative expression; and in this case, the preceding (they) can always be omitted. The engineer will write:

$$I^V 3PL \cdot \{Ad \;\; U \;\; Pa\} \cdot \sim P_n E^n . f_n E^n = \sim,$$
(they)

where:

Ad = adverb
I^V = impersonal verb
V_3PL = verb in third person plural
Pa = particle
P_n = preceding in sentence by an undetermined number of words, but only words of form class { } may intervene.
E^n = nominative expression
f_n = following in sentence by an undetermined number of words, but only words of form class { } may intervene.
{ } = allowed intervening form class.
$\sim$ = negation
$\cdot$ = logical "and"
U = logical "or"

Other notation symbols are of course required and used before a language can be completely specified.

After these expressions have been written, similarities often become apparent. Many instances arise when several expressions can be combined into one which is only slightly more complex but much more general.

After a large body of the text material has been analyzed and the processing expressions have been written, the expressions are grouped according to the distributional class or grammatical specification of the individual words. A flow chart organizing the processing for each of the distributional classes is then written for the particular computer being used; i.e., separate flow charts are developed for the processing of verbs, nouns, prepositions, and other distributional classes.

There are two principal advantages related to the use of the symbolic notation of the necessary processing steps: first, the expressions allow easy determination of whether sufficient information is available to completely specify and process the step; second, the flow charts may be easily organized and written from the expressions.

The use of symbolic notation brings up an interesting possibility. The interpretive routines might be written for the symbolic expressions in such a way that the expressions themselves might be entered into the computer directly, allowing the computer to develop essentially its own program. This is essentially the

same type of computer technique which is currently being widely used in mathematical symbolical programming operations. The use of the IBM (International Business Machine Corporation) *650* precludes such an effort at this time, but investigations of the techniques which would be required are being made as the more mundane forms of programming develop.

Use of the IBM *650* Computer for Logical Processing

It has been indicated that the output of the word-for-word translation is quite vague and confusing. For the translation to be useful to the human reader, considerable improvement of the output material over the dictionary output is necessary. The types of improvements to be made can be classified into two major areas: first, those problems, such as word order, which arise only because of the difference in structure between the two languages in question; second, those problems which concern the intended meaning of the individual words. The syntactic or contextual rules which have been discussed are to permit the solution of certain representative intended meaning problems.

The IBM *650* computer has been used at the University of Washington to perform the tests stated in symbolic notation which have been previously discussed. These tests are essentially expressions of linguistic rules which determine the meaning of a given word in a specific syntax. Implementation of these rules by computer has not been completed at this time. The intention has been to develop a set of representative routines by which large quantities of text are then processed, finally re-evaluating and expanding the processes to resolve most of the remaining intended meaning problems. The routines which have been emphasized at this time are representative ones for resolving ambiguities of verb, substantive (noun), conjunction, and certain other classes. In addition, the rule of agreement of substantives with their modifying adjectives and prepositions is used to solve certain intended meaning problems.

The input of the processing operations has been made up of cards which duplicate the output of the word-for-word translation dictionary. The computer then is used only for testing of the logical processing procedures and not for the translation dictionary look-up. Because of the limited storage, it has been impossible to store all of the logical processing programs on the drum at one time. The program therefore has been divided into three sections: the first concerns the rule of agreement of the substantive with the modifying adjectives and prepositions; the second and third with substantive, conjunction, verb, and certain multiple class-processing patterns.

Generally, the text material is processed by loading the program and text deck into the computer. The computer punches out another card deck which is similar to the input text deck except that certain modifications are made to solve some of the intended meaning problems. This process is repeated through the three rounds of processing and then through a final interpret round of operation which provides for the insertion of certain English prepositions and the omission of certain word equivalents which the processing rounds have determined to be unnecessary. After the interpret routine, the output deck may be introduced into the accounting machine to give a printout of the processed translation.

The first round of processing considers the rule that Russian substantives must agree with the adjectives which modify them in gender, number, and case. Prepositions must agree with the substantives which they govern in case. Use of this rule defines many ambiguities of translation which may arise when a given word in isolation may have several possibilities of gender, number or case.

As has been previously noted, binary tags representing the grammatical specification of isolated words are carried in the translation dictionary output; i.e., each source word has a related grammatical tag in which ones or zeros in certain positions indicate the gender, number, case and other related information. In a noun clause, that information discussed which is common to all of the tags of the phrase or block is indicative of the use in the context of that particular sentence element; and all which are not common may be disregarded as not applicable to the context. By applying the intersection sum of Boolean algebra to the binary tags within the group, those grammatical specifications common to the group are developed. The general tags are then modified to represent the specific grammatical usage permitted in the partical context. In addition to the fact that this information is relayed for use in further processing, it immediately permits the resolution of certain grammaticial problems and yields a reduction of equivalents within the translation.

It will be noted that in this part of the processing, a relatively pure matching operation is used in the programming. A word is identified as a particular part of speech by matching of the tags. If the word is a preposition or adjective, the intersection sum is taken with successive words until the substantive is reached. The individual tags are modified as the information common to the group replaces the more general information related to the individual words in isolation.

In the later rounds of processing, more involved processing equations are explored. Many of these require the examination of several words in sequence and an analysis of words preceding and following the word being tested. The computer was used to store and examine six words at a time in a survey storage area; the word in question, two words following and three words preceding it. The text is then stepped through the survey storage area in one-word increments. The second and third rounds of processing are most efficiently carried out by what is essentially a combination between matching procedures and a programmed search process. For any given processing test, the computer examines the contents of the 6-word survey storage area to determine if the conditions of the test are satisfied.

Obviously, the grammatical processing tests cannot be efficiently made in isolation from each other. A group of tests considering the same grammatical class of word are related in a common operation by a generalized flow diagram and the program branches to determine which, if any, of the related processing equations are satisfied in a given context as it appears in the text. It would be possible to carry pure matching methods into this processing, but investigation indicates that optimal use of the computer can be best obtained by means of the combination matching and search methods.

In order to illustrate the clutter of the word-for-word dictionary output and the action of logical processing in clarification of the output, an example will be given showing the dictionary output and the output of the second round of processing. The dictionary output of a representative text sentence is:

Last/latter is attained/reached (to/for) (by/with/as) skillful (by/with/as) choice/election (of) private/particular/quotient/partial (of) case/chance/occurrence (of) general/common/total (of) laws (of) appearances/phenomena/symptoms, (of) tying/connecting/knitting (of) magnitude/quantity, (of) unit/one (of) which enter-into definition/determination/attribute (of) (to/for) (by/with/as) derivative (of) unit/one.

The output of the first and second rounds of processing is refined to:

Last/latter is attained/reached by/with/as

skillful choice/election of private/particular/quotient/partial case/chance/occurrence of general/common/total laws of appearances/phenomena/symptoms, (of) tying/connecting/knitting magnitude/quantity, units/one (of) which enters-into definition/determination/attribute of derivative unit/one.

Complete processing of the type described would yield:

Latter is attained by skillful choice of particular case of general laws of phenomena, of connecting quantity, units of which enter into definition of derivative unit.

It is obvious that the logical processing of the dictionary output does serve to clarify the text meaning. It is equally obvious that present processing does not completely satisfy the clarification. In the example specified, however, the first two rounds of processing act essentially to reduce the number of required reader decisions by fifty per cent.

There are certain multiple-meaning problems that cannot be directly resolved by grammatical testing. As an example, the same Russian word can represent either "matrix" as the word appears in mathematics or a tool maker's "die." To reduce this type of multiple equivalence, a scheme has been implemented which requires the identification of all dictionary words as to fields of application. The specification of field of application appears as a part of the grammatical specification tag, when a word has different meanings in different fields, this will be noted. As the text is processed, the common field or fields of application are determined and the unrelated equivalents are dropped. This matching procedure can, under certain conditions, yield results which are not completely satisfactory in translation, however the general concept and process is one which may be used in future investigations.

Word Arrangement and Block Analysis

A problem of translation which is related to the difference of structure between the two languages is that of word arrangement. The translation so far described has been essentially in the same order as the input text. In many languages, this could present an extremely serious problem. Fortunately, Russian technical literature uses word order which is usually acceptable in English. This relationship of word order, however, is not uniform, and the problem requires investigation. One approach to the general problem is the identification and separation of different blocks of the text sentences. After this has been accomplished, word rearrangement within the blocks can be accomplished by matching procedures; rearrangement of blocks within the sentence can be accomplished by a combination of matching and search operations.

At the University of Washington the identification and isolation of blocks within sentences was initially done using Greek as the source language, then extended to Russian. Noun phrases, prepositional phrases, verbal forms, adverbial clauses, and co-ordinating conjunction forms are isolated and identified. The word order within the blocks is not rigid, but follows a general set of forms which allows the use of matching procedures and Boolean intersection and union sums to identify and determine the limits of the blocks. Since the IBM *650* computer operates in biquinary arithmetic, it is not capable of performing direct intersection summations. Special subroutines are therefore used to transform normal arithmetic summation results to the corresponding intersection sums. As more complicated word blocks are studied, a combination of matching and search procedures are required. It should not be implied that methods of word rearrangement have been completely developed at the University of Washington. Specific programs and operations are currently in operation at the block analysis and isolation level, but little application work has been carried out in the rearrangement problem at this time.

Mechanical Translation Economics

Comparison of the costs of human translation with those for machine translation is a rather difficult one to make. Current machine-translation development is at the research phase rather than in production. Resultantly any estimates of the ultimate cost of a refined machine-translation process must necessarily be approximate and involve considerable extrapolation. There are many processes that are not currently being considered by mechanical translation techniques: the processing of mathematical developments, placing equations at the proper place on a printed page, graphical developments, flow diagrams, and other representations important to the meaning of technical literature have been neglected. One type of text material which might be used for comparison is scientific material which is not mathematical or highly specialized in nature; i.e., text that could be translated on a typewriter by a non-specialist with a general scientific background. This material can be translated for roughly one cent per word by human translators.

If the text input to the mechanical translator is to be either punched tape or cards, the experience at the University of Washington has been that the cost of punching text by employees of average ability is on the order of 0.85 cent per word. The cost of the dictionary search on the International Telemeter Corporation memory device and the print-out are combined to something on the order of 0.10 cent per word. The logical processing costs, even assuming a computer of the magnitude of the IBM *701*, would be on the order of 0.50 cent per word. This yields a comparison of 1.00 cent per word for human translation versus about 1.45 cents per word for mechanical translation.

In a comparison of translation times for the same sample, the punching of the text alone for MT requires nearly the same time as the complete translation process by human means. The actual computer time required is negligible.

Without the development of a high-speed electronic reader, machine translation of languages cannot be justified on the basis of cost alone. For many applications, however, it is likely that MT will not have to be justified on a cost-per-word basis. The electronic reader is of tremendous importance to MT; a really high-speed and efficient electronic reader could reduce costs well below the per word cost of human translation and increase the total speed of mechanical translation far beyond that of human translation.

Conclusions

Mechanical translation of languages by means of digital computers is an extremely diverse and interesting illustration of logical computer applications. Current development in the area is not so much oriented toward production practices as it is used as a research device in the development of the theory of methods of translation. The experience of using a medium size computer is invaluable in determining the final computer to be used in production operation. It is apparent that either a large, high-speed general-purpose digital computer or a high-speed special-purpose logically oriented digital computer can be used in conjunction with the dictionary storage device.

The purpose of the work at the University of Washington has not been to perform complete and perfect translation, but to choose representative operations that are significant in the translation process

and apply them to the computer at its disposal. The work has been carried out as a co-ordinated effort between the linguistic and engineering staff of the project. The specification of the logical procedures and programming have been largely the responsibility of the engineering staff, while the linguistic analysis of the text has rested with the linguists. The direction of the work has been shared by consideration of the over-all linguistic goals and the application methods and capabilities of computer operation.

The logical processing work currently developed reduces the total "clutter," or output unnecessary for meaning in a particular context in the results of the processing operation, to roughly fifty per cent of that of the dictionary output. Indications are that further work will lead to results in which the processing output is in the same or better form than would be obtained from a human translator.

Machine translation will probably be unable to compete with human translation on either a time or economic basis until a high-speed electronic reader has been developed that is satisfactory in a translation application.

The Application of a Large-Scale Electronic Computer to the Assignment of Telephone Facilities

MILTON DRANDELL
NONMEMBER AIEE

THE ASSIGNMENT of telephone facilities based on customer requests for service is today performed on a manual basis throughout the telephone industry. The General Telephone Company of California will be the first operating telephone company which will install a large electronic computer, the International Business Machine Corporation's (IBM) *705*, to process such service requests. At the present time there are 930,000 telephones installed in the General Telephone Company of California system and the company's growth in the past has been at an accelerated rate. This rapid growth and future potential has embarked the company on a large scale EDPM (Electric Data Processing Machine) program to handle assignment of telephone facilities. The machine is scheduled for delivery in late 1958.

First, some terms and concepts that will be used will be defined.

Central Office

A central office is a complex housing of the necessary equipment to serve a geographical area. At the present there are 66 such offices in Southern California. The cables serving these areas originate in the central office, together with such telephone equipment as X-blocks, line finders, and power and switching equipment. This paper is mainly concerned with the cables, and will just state that the other terms used are also necessary in supplying telephone service. In addition, telephone numbers are reserved and held in the central office for present customers and future assignment. A call originating in a subscriber's home is routed through its respective central office to its ultimate destination.

Terminal and Pairs

A terminal, or terminal box, contains the ends of the wires enclosed in the cable emanating from the central office. There are essentially two wires necessary for every telephone connection and this pair of wires is called, briefly, a pair. There are usually 2,121 pairs per cable, and terminal boxes come usually in 11-, 16-, or 26-pair complements. Larger terminal boxes are available for such buildings as apartment houses, office buildings, etc.

A pair is connected to the lugs in a terminal box and when service is desired by a subscriber, it is connected to the subscriber's phone, thus making a complete circuit from central office to terminal to subscriber's home and back again. A terminal will usually service several houses.

In order to give a wider distribution to pairs in a cable, they are spliced so that the same pair, each of which has a number assigned to it as do the terminals, may appear in several terminals, which in turn can appear in different locations. Therefore, a single pair is available to be assigned in a larger area than it would be if it terminated in only one terminal. Since one pair is used for a multiple party line, only the ringing frequency being different for each party, multiple subscribers living in scattered areas may share the same pair. There is another very important reason for such splicing of pairs and it will be discussed later. Those terminals having pairs in common are called associated terminals. Fig. 1 shows the inside of a terminal box with the lugs to which the pairs in the cable are attached.

Fig. 2 shows a schematic of a central office with a cable emanating from it and pairs from this cable extending down a street and ending in terminal boxes. Here 51 pairs of the cable, called *C*1, have been shown with pairs 1 to 16 in terminal *A*, pairs 11 to 26 in terminal *B* and 26 to 51 in terminal *C*. In addition these same 51 pairs could be spliced so as to terminate in other terminals on other streets. Here the terminals are depicted serving one house each, although usually this is not the case. The term "group" refers to a collection of terminals. From the picture it is obvious that terminals *A* and *B* are associated terminals, and the same analysis holds for terminals *B* and *C*.

Fig. 3 depicts a cable page from where the actual assignments of terminals, pairs, and other facilities such as ringing frequency, line finders, and X-blocks are made today. Fig. 3 shows three terminals for simplicity, although in practice there are usually 35 terminals per cable book page which is referred to simply as a group.

From Fig. 3 it can be seen that this group is from the West Los Angeles central office, referring to cable number 1234, and containing pairs 1 to 51 of that cable. Terminal 238, or briefly T-238, is a 16-pair terminal located at 2800 Butler Avenue. The assignment of pair 1 with a telephone number of

---

MILTON DRANDELL is with International Business Machines Corporation, Santa Monica, Calif.

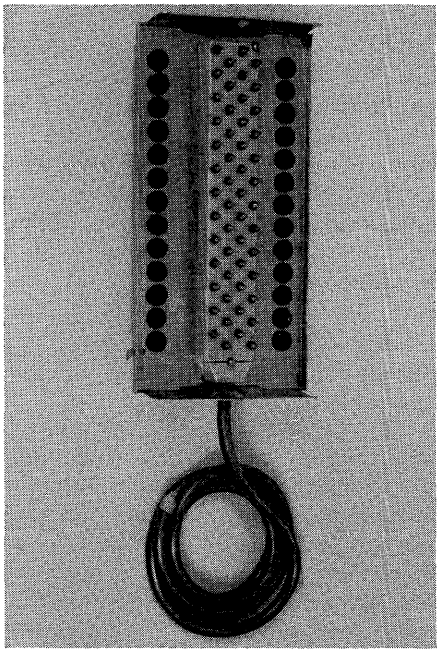


Fig. 1. Terminal box

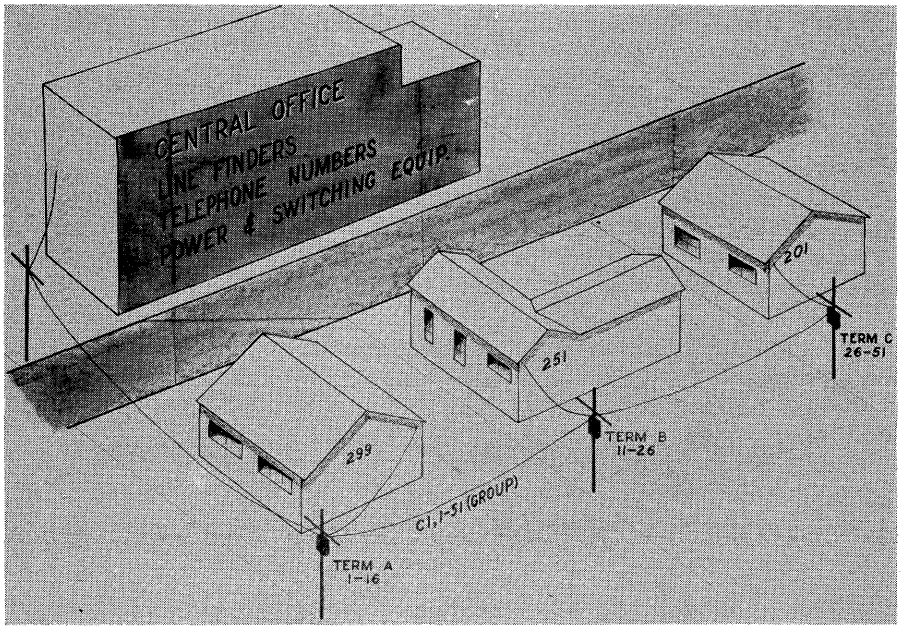


Fig. 2. Central office

4-3701 has been shown, the prefix being understood from the central office in which it occurs. It is a 2-party residential service, the "2" and the "*R*" under their respective headings denoting this, and the terms "X-block" and "line finder" indicate the columns which will contain the numbers of these facilities once they are assigned. It can be seen from the pair number column that T-238 contains pairs 1 to 16, T-1945 contains pairs 11 to 26, and T-1634 contains pairs 21 to 36. In addition the assignment has been shown of pairs 12 and 21 to T-1945 and T-1634 respectively, assuming the remaining pairs to be assigned in one way or another. Such a group as in Fig 3 is used by the company in assigning terminals and pairs on a manual basis, by visually determining those which are available and hence can be assigned.

Having defined the basic concepts involved, the EDPM processing of a customer's request for service will be discussed.

This request for service, called a service order, is taken at any one of the 20 commercial offices scattered throughout Southern California. This service order, one of approximately 4,000 per day, is transmitted via telephone lines to the central EDPM section in Santa Monica, California where the IBM *705* will be situated. In addition to service orders arising in the field, information as to new facilities which have been established and emergency assignments will also be sent to the EDPM section to update the master files. The entire physical facilities available to the company are stored on magnetic tapes, and the customers' requests are processed against these files to find the facilities peculiar to the service desired. This service requested may take the form of a 1-, 2-, or 4-party business or residence phone; a change from one type of service to another; the removal of a phone; a color telephone or an extension cord; or a host of other possibilities.

At the EDPM section the service orders are transcribed to punched cards, several cards per service order, and these cards are transcribed onto two magnetic tapes as follows:

1. The service order file, containing only that information necessary to find the proper physical facilities. This information consists basically of street name, house number, and type of service desired.

2. The "other information" file, containing information extraneous to the assignment of facilities, such as subscriber's name, color telephone and extension cord if desired, etc.

The service order file is the one processed through the system to assign the facilities.

The service order file is now matched against a magnetic tape file called the address file. This file contains the address of every house in the area served by the company which has physical plant facilities assigned to it. The first step is to find the terminal number of the terminal box serving the subscriber's house, extract this terminal number, together with other pertinent information, and write out a new service order magnetic tape.

This new service order file is now processed against the terminal file which contains all the terminals in the system, approximately 160,000 in number, together with the pair numbers contained in each terminal and information consisting of the present status of each pair. In addition, each terminal carries with it its associated terminal numbers.

The primary objective now is to find an available pair (namely, an unassigned pair) in the terminal serving the customer's house, which will fit exactly the service requested. For example, if the request is for a one-party service, a pair must be found among the 26 pairs which is completely free. If the request is for a 2-party service, a pair must be found having only one other customer on it (the same pair is used for multiple parties, only the ringing frequency for each varies). If this fails, then an available pair must be found to be designated as a 2-party pair.

Take the example of a request for a single party service and further, say that no available pair exists in the terminal involved. Other means are now introduced to try to satisfy the customer's request. In Fig. 3 let T-238 be the terminal serving the subscriber's home. The IBM *705* searches T-238 for an available pair and suppose there is none. The search does not end here. For, the next associated terminal of T-238, namely T-1945, is now searched for an available pair and say again the search is fruitless. The terminal associated now with T-1945, namely T-1634, is searched, and this

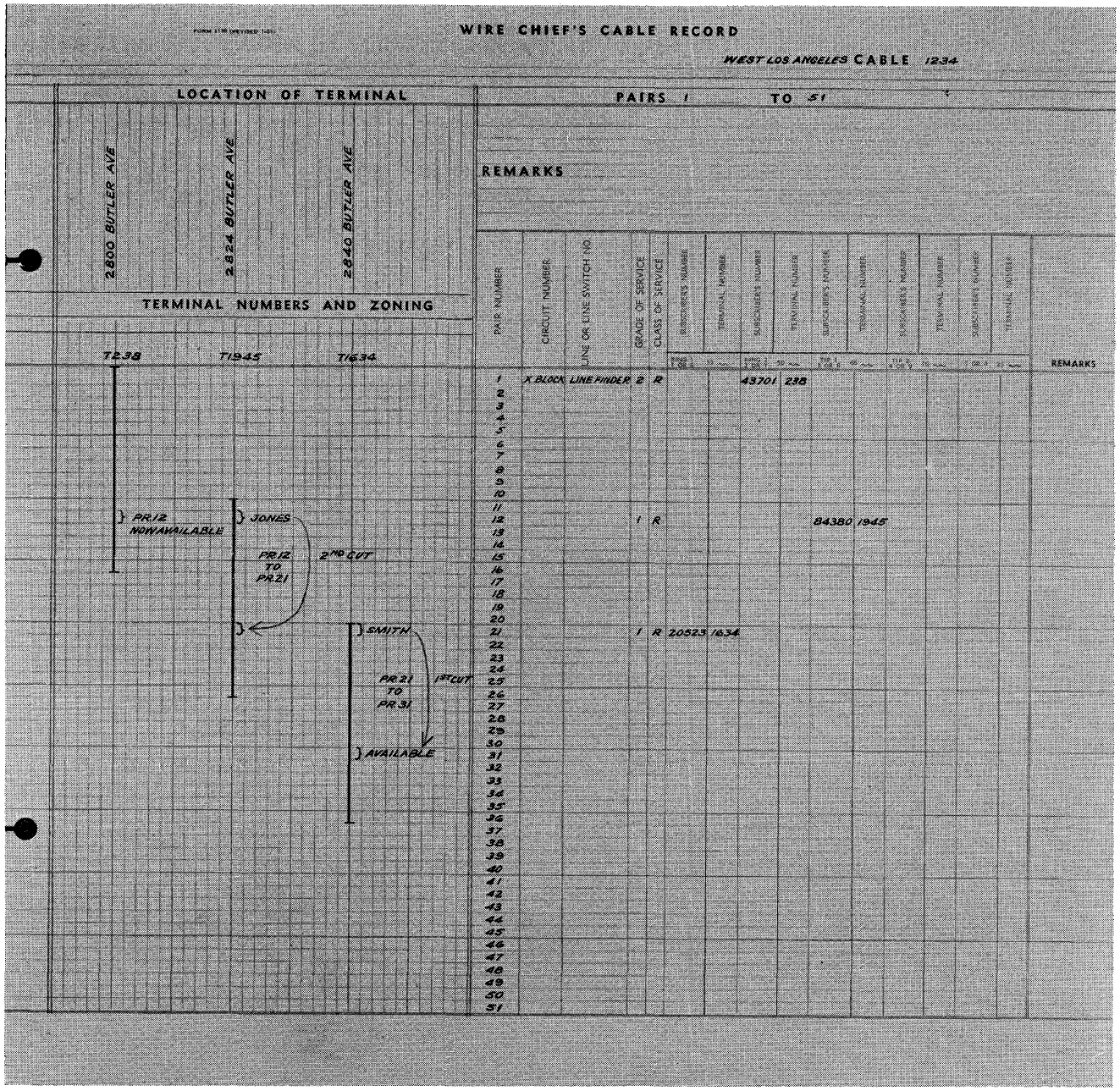


Fig. 3. Cable page

procedure is repeated throughout the group until either an available pair in the group is found or the search is unsuccessful. In the latter case the service order is printed out as being unable to be served, and held for the next day's processing, when the addition of new facilities or the removal of a pair from an existing subscriber may remedy the situation.

In the example, pair 31 in T-1634 is available and the search ends here. The problem now is to reflect this opening in T-238 and here another very important reason for the overlapping of pairs in terminals is evident.

A Mr. Smith has a one-party line in T-1634 and has been assigned pair 21 which is also common to T-1945. Mr. Smith is now moved to pair 31 in T-1634, this procedure called, in telephone parlance, a "cut". The mechanical procedure consists simply of moving pairs around in the terminal box. Mr. Smith still has his one-party service and same telephone number and couldn't care less about the switching of pairs. Now pair 21 is free in T-1634 and what is most important in T-1945.

Now a Mr. Jones has a one-party line using pair 12 in T-1945 which is also common to T-238, the terminal in which we are interested. Another cut is now performed which moves Mr. Jones from pair 12 to pair 21, thus freeing pair 12 in T-238 and giving the subscriber the service he wants. The same impartiality of Mr. Smith toward the procedure is also evidenced by Mr. Jones; however these changes must be reflected in the records stored on magnetic tape. There will be made at most two such cuts in order to assign a line, but the programming is such that any number within a group can be encompassed. Of the 4,000 service orders daily, approximately 1,000 will involve cuts of the type just described.

The IBM *705* will perform the above complicated analysis and will be programmed to do this so that the rear-

rangement of pairs is kept to a minimum. For example, if there is a request for a one-party service and no pair is available, but there are two 2-party pairs each having only one customer assigned, then both customers will be assigned to one 2-party pair, thus freeing a pair for the original assignment. This eliminates the cutting of pairs from terminal to terminal.

Once the terminal and pair numbers have been established, certain equipment in the central office must also be assigned and once again the magnetic tape files furnish this information. Finally the customer must be given a telephone number and a magnetic tape containing all available numbers supplies this. In case service cannot be assigned for various reasons such as temporary lack of facilities, these orders, called "held orders," are processed through the system on and subsequent days until service can be established.

The service orders, now containing all the necessary information to assign the specific telephone facilities are merged with the "other information" file and the merged tape is connected to a high-speed printer off line and the completed service orders printed out. These then go back to the areas from where they came and given to the telephone installer. Installation will usually take place on the third day after receipt of the service order. The time for the EDPM processing is within one shift of operation. There are in excess of 300 people engaged in this operation at the present time.

The IBM *705* will also be used in the areas of customer billing, inventory control, and eventually trunk assignment. The on-line *705* machine configuration to perform this procedure is as follows:

1. One *705* central processing unit
2. Two tape record co-ordinators (tape buffers)
3. Sixteen tape units
4. One card reader
5. One card punch

In addition there is various off-line equipment to handle the peripheral operations.

This procedure is a revolutionary concept in modern telephony as it automates, beyond anything existing today, a very complicated process.

An Experiment in Mechanical Searching of Research Literature with RAMAC

F. E. FIRTH
NONMEMBER AIEE

A LITERATURE searching system which can store for retrieval information on 25,000 documents has been developed at the International Business Machines Corporation (IBM) Research Laboratory, San Jose, California. This was accomplished on an IBM *305* Random Access Memory Accounting Computer (RAMAC) in a research project aimed at gaining experience in information retrieval while providing functional aid to the technical library at the laboratory.

This system is an outgrowth of a study conducted by J. J. Nolan of IBM Product Planning which demonstrated the practicability of the *305* as a tool for information searching.[1]

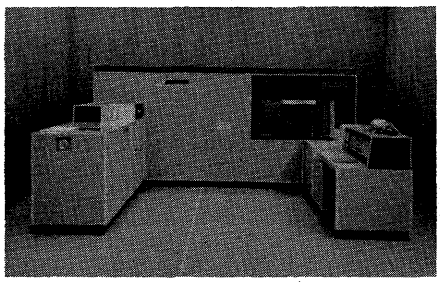


Fig. 1. IBM 305 RAMAC

When a researcher has a question that may be answered in the literature in the library, his inquiry is used to search through a dictionary of terms in the RAMAC, which then prints out a bibliography of titles with their library location numbers.

Description

The IBM *305* RAMAC shown in Fig. 1 contains a disk memory, a card reader and printer, punch, a drum for processing, a typewriter for manual input/output, necessary control panels, operator controls, and logical circuitry equipment.

The memory unit is composed of 50 magnetic disks and has a capacity of five million characters which are divided into 50,000 records of 100 characters, see Fig. 2. Any record is obtained by positioning the access arm at the desired address and reading its information onto the processing drum or into one of the output devices.

System Philosophy

As a result of experience from working with an IBM *101* card searching system, the following principles determined the approach to the operation:

1. The system was to be evolutionary in nature. Operating experience dictates possible improvements or necessary modifications.
2. As much of the work as practicable, such as the machine coding of input words and the direct bibliography print-out, would be done by the machine.
3. Human effort on input processing would be held to a minimum on the premise that a majority of the information put into the system will never be called for, so that the work in entering such information will be essentially wasted.
4. All effort spent on output is productive; therefore, work would be concentrated on that part of the program.

Information Preparation

Documents are normally sent to the retrieval office from the library immediately after the routine processing there is completed. A 5-digit serial number is assigned to identify the document within the system and two types of cards are punched.

The first is a bibliography card that will be used to identify the document in response to a search. This card contains the serial number, library location number, author, title, and date. The second is an input card used to describe the document within the machine system. It contains the serial number and as many words as necessary to describe the information content of the document. Word length is limited to ten characters. Combinations such as "spectrophotometer" are divided into logical word units, e.g., spectro. photometer.

F. E. Firth is with International Business Machines Corporation, San Jose, Calif.

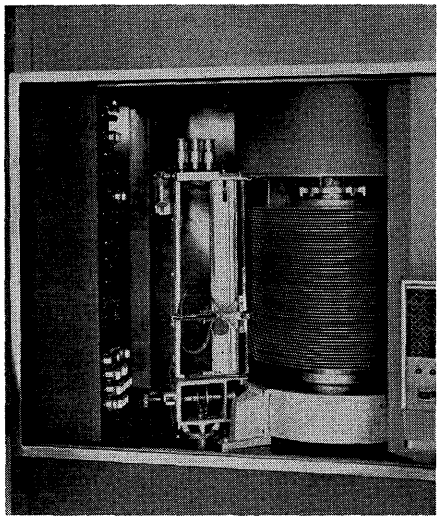


Fig. 2. 305 file unit

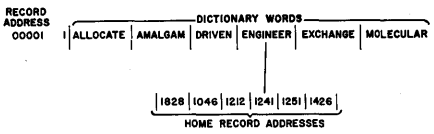


Fig. 3. Dictionary record

This record was divided for purpose of illustration. 1241 is the home record address for the word engineer. The 1 preceding the first word is an overflow indicator

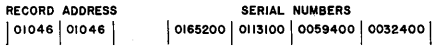


Fig. 4. Home record

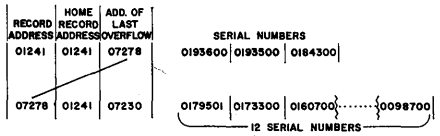


Fig. 5. Overflowed record

Note that the overflow record refers to a previous overflow

Dictionary

The initial machine dictionary was made up of the descriptive terms of the first 1,000 documents. Each different descriptive word is punched into a dictionary card together with the address of a file record assigned for recording the serial numbers of documents described by that word. This record is called the word's home record. Different forms of the same root word, as compute, computer, computation, are assigned the same home address. The word and its home-record address are posted at the file location so addressed, see Fig. 3.

When the dictionary cards are read into the RAMAC, an address is computed by the machine from the word itself to fix its location in the machine dictionary. To do this, the word is divided into two 5-character halves; these are added, the sum is squared, and the three central digits of this figure then become the address. This technique locates the word within a small area of the file and at the same time assures a fairly even distribution throughout the dictionary area.

When a dictionary record is filled with six words, subsequent words computing to the same address are written on the following record.

A printed alphabetical listing of the dictionary cards serves as a manual reference. The addition of new words subsequent to this initial loading will be described.

Machine Input

Each input descriptive word is processed by RAMAC in two steps. First, it locates the word in the dictionary to get the home-record address. At the home-record address, the document serial number is posted so that a list of document numbers is generated at the home address which will indicate all documents characterized by that particular descriptive word.

This is accomplished as follows: Document serial numbers establish the order of reading input cards into the machine. After computing the dictionary address (three center digits of the squared sums . . .) the words at this dictionary address are read into the RAMAC processing drum.

If the incoming word is one already in the system, this fact is established and the home-record address determined by comparing the entering word until it matches one already listed.

If the new word is not found in the dictionary, the word is automatically typed on the console typewriter where the operator assigns it the next unused home-record address manually. Step two repositions the access arm at this home address, where the home record is read out. The existing serial numbers are shifted to the right and the incoming serial number is posted. A home record is shown in Fig. 4.

As the home record is limited to 12 serial numbers, an overflow system is provided to extend the length of a home record when it becomes filled.

The address of the next unused record in the overflow section is indicated by a counter. After determining the overflow address from this counter, the program causes the entire home record to be transferred to the overflow (Fig. 5), erases the home record and records the incoming serial number along with the overflow address just used, which is to say that, as a home record successively acquires 12 document serials, these blocks are transferred to the next available overflow at the time and in such a way that a trail of addresses is left behind from location to location from the home record.

Searching

To illustrate this process, suppose an engineer requests a search on "magnetic memory devices." The home-record addresses of the words "magnetic," "memory," and "devices" are manually looked up in the dictionary, which insures the words existing in the dictionary; then the addresses are punched into the search card.

When the search card is read into RAMAC, all numbers listed at each home address are read onto the processing drum. Then a matching occurs to obtain serial numbers common to both "magnetic" and "memory"; each time a common number is located it is sought under "devices." The resulting serial numbers common to all these terms are the answers to the search, are used to address the bibliographical record, and stored in a separate part of the memory, which is then printed out as shown in Fig. 6. When all numbers on a home record have been compared, overflow records are automatically sought until all serial numbers for the word concerned have been exhausted.

When a search for alternate terms is desired, commas are punched in the search card between the alternate words; i.e. "magnetic drum, disk, tape memory." These cause selectors to modify the program accordingly. Obviously in this case the search continues until all alternate word records have been thoroughly examined.

Capacity

The 50,000 one-hundred-character records available in the *305* memory are divided as follows:

1,000 records-dictionary (6,000 words)
6,000 records-home records of dictionary words
18,000 records-overflow
24,900 records-bibliographies
100 records-program storage

With 12 serial numbers posted to each record, this organization permits the storage of 288,000 serial numbers, or 24,900 documents with an average of over 11 descriptors per document.

IBM SAN JOSE RESEARCH LABORATORY — MECHANIZED INFORMATION RETRIEVAL

SUBJECT REQUESTED	REQUESTED BY	DATE	REQUEST NO.
INFORMATION RETRIEVAL	JONES LA	03 11 58	0019
TITLE	**AUTHOR**	**DATE**	**CALL NO.**
INFO SYSTEMS IN DOCUMENTATION	SHERA JH	04 00 57	02197
LINGUISTIC PROB IN MECH OF PAT SEARCH	US PAT OFFICE	00 00 00	02362
ADVANCES IN PATENT OFFICE SEARCHING	US PAT OFFICE	04 15 57	02360
CHANGING STRUCTURE OF KNOWLEDGE	VICKERY BC	00 00 54	01499
CODING ORGANIC COMPOUND WITH A MECH ANIZED SEARCHING SYS	NORTON TR	05 27 53	02131

Fig. 6. Bibliography output

PROGRAM

Two basic programs and control panels are required for this application. The loading program which contains both dictionary and document loading requires 96 instructions. The search program is made up of 89 instructions and is applicable to any search without alteration.

Programs are permanently stored in the file. This permits a complete change from one operation to another merely by changing the control panel and reading in a single card to withdraw the desired program and put it into operation.

Results

To date 5,000 documents have been stored using a total of 25,059 descriptive words. These documents are company reports, articles and papers ordered by the library at the request of laboratory personnel, as well as the *Review of Scientific Instruments* since 1953, *Soviet Physics* since 1956, and selected material from all library technical periodicals since January 1, 1958.

Words are well scattered throughout the 1,000 records of the machine dictionary. With 2,394 words in the dictionary, 900 records contain at least one word.

Considerable delay was caused by the necessity of stopping the program each time a new word appeared as documents were being loaded. This has been eliminated by feeding the entire deck of new input cards two times. On the first feed an alteration converts the load program to a word look-up program which punches out words not found in the dictionary. Manual assignment of home-record addresses to these words creates dictionary cards which are loaded prior to the final run. With the new words already in the dictionary, actual document loading proceeds without interruption.

Search time varies with the number of terms searched and the number of postings to the records involved. After the search card has been prepared a normal search requires less than one minute.

As in other mechanical searching systems, the biggest problem encountered is selection of the correct descriptor to locate the information sought. The searches conducted have been completely successful in providing the requester with a bibliography within the limits of the terminology used by the searcher. The number of additional documents, described by words not provided by the requester, depends on the amount of human ingenuity applied in selecting other terms which will identify the desired information. That this effort should be applied at the time a search is conducted is in keeping with the basic philosophy of the system.

Reference

1. INFORMATION STORAGE AND RETRIEVAL USING LARGE SCALE RANDOM ACCESS MEMORY, **J. J. Nolan.** Presented before meeting of Division Chemical Literature, American Chemical Society, San Francisco, Calif., Apr. 15, 1958. (Not available.)

Discussion

Chairman Minnick: The first question is from Mr. R. A. Kirsch, National Bureau of Standards: "Do you have any learning feature built into the program?"

A. Bernstein: Not now. We have been thinking about it for some time, and we have two or three possible types of learning to put into the program. I do not know when we will do this, or if we will do this. The first guide is a self-adjusting of the parameters in the sense, that if the machine loses a game, it will then go through the set of moves the opponent makes, and will examine them to see if that move which the opponent made was in the plausible move table. Presumably, if the opponent beats the machine, some of the moves which were made will not be the same as the moves that the machine actually predicted for the opponent. I hope, however, that some of them may be in the set of plausible moves. If this is so, we will try to write a program to try to adjust the parameters in such a way that the moves which were predicted, which actually became the moves of the opponent, will be the ones which the machine will predict. We will try to weight the parameters in such a way as to take advantage of the strategy, or the weighing, which the opponent has just exhibited.

Another learning feature is that when the machine loses a game because the opponent has made a move which is not in the plausible move table, the machine will go through the entire plausible move table and generate all moves to see whether, indeed, it would have at least found this move in the plausible move set.

The third possibility, so my colleagues in New York tell me, is an alternate. Let us assume that the machine has a winning game, and that the opponent now makes a move which is not at all in the set of plausible moves, the machine will then say, "This is a strange move, why did he do that; let me try to examine it to see if it is in the full set?" If it is not, the machine will assume that the opponent made a mistake, some strategic concept is missing in the opponent's play. Then the idea would be that the machine would essentially know some characteristics about the opponent's play, and it might possibly be able to take advantage of it. This is certainly true in human chess. There are some people with whom you play that you know will never sacrifice a piece. If you know this much, then you can certainly take advantage of it.

Now, of course, this leads to a problem: What if the opponent is trying to fool the machine? This particular line of thought might be dangerous. Again, as I say, all that we have done is to think and to talk about it. We have not started really, but we may, if we get enough support.

Chairman Minnick: The second question is from Mr. S. H. Unger, Bell Telephone Laboratories: "Why did you decide against the idea of using different programs at different stages of the game? The objectives at the end game are so different from the opening game objectives that this would seem to be desirable."

A. Bernstein: I think that it was mostly through esthetic consideration; we wanted one program to play the entire game. In actual practice it seems to play all phases of the game equally well—or badly—depending on how strong a chess player you are. Essentially, the machine is geared to advance pieces, put in pawns and capture material if it can, and advance pieces into key squares; and as the end of the game progresses it tries to "queen the pawns." It realizes this will give it an increase in material, and once it has the necessary material it will just continue. I think that the thing should work mainly for esthetic consideration.

Chairman Minnick: From Mr. Paul C. Tiffany, SDC: "Has any consideration been given to storing up experience to modify the master program and speed up decision making?"

A. Bernstein: Dr. Samuel of IBM, who wrote the Checker Program, has such a program built into this machine. Essentially, the positions which occur are stored on tapes, and every time this position reoccurs the result of the game is also stored on this tape. When the machine takes a familiarized position, the first thing it does is to look on the tape and it says, "Have I met this in the past; how many times have I won; and how many times have I lost?" It says, "I have won 12 times, and lost 3 times, so my best move would be where I have won 12 times." It does that.

Now the result is that Dr. Samuel's machine now plays very nicely against strong players. Of course it requires a certain amount of time to go through this tape. In Dr. Samuel's case I think it takes two words to store position, and one word to store move and result. In our particular case it will take about 150 words to store all this information; 100 words at least.

Now the interesting thing about Dr. Samuel's machine is that it behaves this way against a strong player and it plays well. The machine has played quite often, and has by now discovered what the good moves are, and what the bad moves are. It meets position, and makes the correct move, and gets a good game, or at least an even game, and it does rather well. There was one occasion when Dr. Samuel said that the machine was playing both sides of the board; against a strong player it drew both games; against a weak player, what happened is this: It wasted a great deal of time looking up on tape for some positions that were not there. Having done this, it had resulted in a time limit, so it has very little time to calculate what move it should make. As a result it made a poor move and lost both games.

Chairman Minnick: Mr. David Lee, Rocketdyne, asks: "Is the game strictly defensive? Will the machine make a one move mate in lieu of a capture?"

A. Bernstein: Absolutely, the machine will make, rob, and capture. So far the machine has played with white only; it has not played with black, but it is certainly capable of doing it, and what happens is that it plays very aggressively. In fact it plays so aggressively that we have had to sort of bind it a little. It develops its bishop to knight 5, and it gets forced back. It is by no means a timid chess player. It definitely takes advantage of the mistakes. We are debugging it at Massachusetts Institute of Technology, and we were fortunate in having a machine operator who wanted to play the machine and did not know how to play very well, and by the fourth move he had lost one man, a knight, and about two moves later he lost a pawn. The machine will never leave a piece en pris, and will simply take it off if you leave it.

Chairman Minnick: The next question is from Edward M. Atone, Radiophone Company: "Would you give us some idea how far ahead the champions and near champions look? Whom can the machine 'beat now,' and whom will it be able to 'beat' with the 32,768 cores?"

A. Bernstein: Well, different champions have claimed different things. One claimed that it entered into sacrifice twenty moves before it would work; that is the sacrifice. I guess he really did not see the game through. Yet there was a world champion who claimed he could see 13 moves ahead. When I started this particular program I spoke to Arthur Bisguire and asked him how many moves he would like to be able to look ahead, and he said that he would be satisfied to see three moves ahead all of the time.

This is essentially what Dr. Lasker says, and he is one of the strongest players. He says that if you make a good move and have a good position, you will not lose the game. Strategically you are all right; there is no tactical trick that can be played upon you that will beat you.

As I said, we can now look two full moves ahead, and with 32,000 words we should be able to expand considerably to make the play more sophisticated. It plays a very nice amateur game now; it is a little aggressive in the openings, and a little weak in the endings, but even in the openings after the second move it is fine.

Chairman Minnick: From David Bennion, Stanford Research Institute: "If the machine prefers pawn move to knight move on first move, did this require special provision? Does machine have rating from any national chess organization?"

A. Bernstein: I think that I have answered the first question; there is nothing to influence the decision in any way.

We are thinking of entering the machine, in an anonymous fashion, in a Class D Postal Chess Tournament to see how it does. That is the lowest class of tournament there is. I suppose that the lowest range will be good for a beginner.

Chairman Minnick: G. D. Fisher, Convair asks, "Can a machine castle with a rooked pawn? Has a queen mate been tried?"

A. Bernstein: That is legally you can castle only when king and rook have not moved.

Chairman Minnick: The second part of the question?

A. Bernstein: We have tried it, but normally it will see mates, although not always. It depends upon the situation. I played a game with Mr. Lasker who said that 8 to 1 it is ignored, and sometimes it plays chess that is very complicated. There are millions of situations, special situations, and under most conditions, it will see a mate, but at certain times it does not. What would happen, I think, is that it would ask many more questions, or we should put in a lot of special situations.

Chairman Minnick: Mr. D. N. Young, Ramo-Wooldridge asks, "Roughly, what level of chess player can the machine regularly defeat? What are some of the outstanding weaknesses in the machine's play which frequently cause it to lose?"

A. Bernstein: We have not played enough games to tell you what level of player the machine will regularly beat. We expect that it will do well against a novice with not more than 2 or 3 month's experience, and who is not too bright.

As to the outstanding weaknesses, I think that one of them now is the fact that when under decent attack, the program only moves away a piece. Incidently, we have not put in the defence piece routine to defend moves, because it takes a certain amount of time; it takes a considerable amount of time as a matter of fact, the way our tables are formed. Moving pieces is no problem but moving a half level for a search, (this should be done eventually) we did not have space in our present machine.

Chairman Minnick: From John Neuman, Lockheed Aircraft: "Can the computer make the first move of a game? If so, is this move always the same?"

A. Bernstein: Yes, as long as it does not change, it will always be the same situation. There are always different situations, but they are always interesting.

Chairman Minnick: Dr. H. K. Flesch, I.T.T. Laboratories asks, "Could you not save move-time by introducing some continuity into the game (your program will play the same move regardless of who played the game up to that move); why not store branches, remember intended strategy, etc.?"

A. Bernstein: We thought of this and the only difficulty is position, evaluation, and information would have to be stored on tape. We have no other storage. Possibly when we get 32,000 we might allocate some of the 32,000 to it. Until we have more cores, and do not have to use them for other purposes.

Chairman Minnick: A question from Jerry Wiener, Magnavox: There is a chess problem marked on the card, so I will have Mr. Bernstein read it.

A. Bernstein: "How is a standard sequence like smothered mate from the discovered check, and Q sacrifice handled; deduced or prestored?"

R			K	
		On	P	P
OB	OQ			

He has a position here which is a classical position: The smothered mate that calls for the queen to be sacrificed, with the king in the corner. The queen must be captured by the rook because the queen depends upon the knight that moves around and checkmates the king since the king has not moved. This is a position that the machine will not see. It might possibly be able to make it, if the position were to arise, I would not guarantee it one way or the other. It would not aim for it, nor would it try to prevent the opponent from arriving at such a position, except that usually such a position involves the knight very closely to the king, and, as a result, the machine automatically tries to prevent such a position. It keeps all squares on the machine well protected.

Chairman Minnick: From J. H. Allen, General Electric Company, "What do you expect the reaction of the game's enthusiasts will be, if the machine should develop an unbeatable mate in 19 moves from the beginning?"

A. Bernstein: This is still speculative. If we are willing to work very hard for the next 25 years, this will occur, in which case I will be delighted to see it.

Chairman Minnick: Robert Shapiro, SDC wants to know, "Can the program also do chess problems as well as play complete games?"

A. Bernstein: Yes, with some modifications.

Chairman Minnick: From John McCarthy, MIT, "How can one learn from experiments how to improve programs of this kind?"

A. Bernstein: I think that if I get enough computer time that what I would like to do is to try the parameters, grind them by hand and see what sort of game they play. We can probably get information other than that. I can tell you that we put into the machine what we think a human being does, and if it does not do what we want it to do, all we do is to go back and say, "What have we learned about it; about the way we play chess?"

Chairman Minnick: I have another question here from Mr. Wiener, Magnavox. "Do you take account of P discovered and double checks in evaluation function?"

A. Bernstein: The evaluation function is not taken into account, however, the generation of moves takes it into account. Of course, if the king is in check something must be done about it. If the discovered check is possible, and if the machine sees it, certainly it will avoid it; if it does not see it, it will not.

Chairman Minnick: Mr. S. Naftaly, United States Air Force asks, "When the machine plays with itself, aren't there only 2 games?"

A. Bernstein: I should say that there would be only one game.

S. Naftaly: My question was based on which half the machine starts first?

A. Bernstein: They are identical. White always starts first by the way, but it makes no difference, since it plays black the same as white. It assumes when it calculates moves for the opponent, it does not actually flip the tables around, it just goes in the other direction.

Chairman Minnick: Mr. T. N. Hibbard, SDC asks, "Does your mobility calculation include the probability of the future movement of the pieces limiting the mobility?"

A. Bernstein: In the calculation of the tree, every time a move is made, whatever move has been made with the following cuts in mobility will be calculated, so there really is no probability of the question. The machine actually makes the move and generates the tables.

Chairman Minnick: This question is from N. Habibe, North American Aviation: "Please comment on a stalemate."

A. Bernstein: At present when the machine is in stalemate it says, "I see no moves," and resigns. This is something that has to be fixed.

Chairman Minnick: The second part of Mr. Habibe's question is, "Can you estimate how the program will perform in a tournament, that is, in terms of wins and losses?"

A. Bernstein: That depends upon the play, and it will probably not do too well, at least in its present state. I hope that future chess machines will be developed to do better, and I hope they will have gained something from our experiences as far as we have gone now.

Chairman Minnick: Mr. McMahon, SDC asks, "I am curious as to the computer time, the salaries of the programmers, and everything involved. Who paid the bills, and just what is the eventual purpose?"

A. Bernstein: The International Business Machines Corporation paid the bills. We also had a computer which was sitting idle a good deal of the time. As to the salaries, members of our department felt that there was sufficient worth in doing problems of this nature. There is no particular procedure that can be specified where the actual answer to a problem is merely a matter of opinion, and in many cases can come up with the solution. We feel that in time we may be able to put some learning into this, and there are certain groups of people who are interested in artificial intelligence.

Chairman Minnick: The first question for Mr. Hoffman is from Quentin Correll, IBM, "Has any work been done on using time instead of distance as the criterion for the most desirable route?"

Walter Hoffman: Yes, we use full time and distance in our determinations. As a matter of fact, in the current work we are doing, we have tentatively given recognition to weight factors.

Chairman Minnick: Mr. Correll also asks, "Is this just in distance?"

Walter Hoffman: The length of a street is according to its capacity (we figure that expressway travel is twice as fast as travel on the main streets), and the ratio of the

main street to the sidestreet is five to three. In testing, we do consider time.

Chairman Minnick: Mr. T. Stockebrand, MIT, wants to know, "Isn't the time of the day in which trips take place a very large factor in defining the matrix?"

Walter Hoffman: Not in the definition of the matrix. The people conducted a survey and decided to work on a 24-hour basis because we felt that the direction of traffic, according to time and the time of day was relatively fixed. Fifteen per cent of travel takes place in the rush hours. Of course, the freeways are perfectly fine in the off-rush time; it is during the rush hour that the tie-ups occur.

Chairman Minnick: Mr. Moser, Shell Oil Company asks, "Are you planning to make a detailed trip route survey of traffic patterns? This has been done for Contra Costa County, California, for study of desirability of alternate paths."

Walter Hoffman: I am not familiar with the study you mention. I must point out that I am merely a mathematician. There have been a number of surveys made, and as a result they have become available and we use them as much as possible. In the metropolitan areas we know of no such thing being done, and we doubt that it can be at the present time.

Chairman Minnick: From Mr. J. T. Lienhard, CEC: "What effect does the relocation of, or additional industry have on your predictions?"

Walter Hoffman: A great deal. This is one of the reasons for existence. For example, shortly after the initial survey was made, a large shopping center sprung up which naturally called for a renegotiation of the surrounding areas. This is why it is important to have programs available to adjust and update.

Chairman Minnick: A question from Mr. R. L. Cost, United States Air Force: "To what extent has the accident or casualty incidence been taken into account in your first recommendation?"

Walter Hoffman: Not at all. We do not have a satisfactory answer and we do not know how to do it as yet.

Chairman Minnick: From Dr. Amarel, RCA Laboratories: "Could you comment on the possibility of centralized travel control (control of flow through branches) using a central computer, and what is the probability of using additional features to this system? Is this clear to you?"

Walter Hoffman: I think that the first part is. It certainly is feasible to control traffic movement by means of computers, but that is the only thing that has been done that I know of. In Detroit, television gives better information on traffic tie-ups in the express freeways. We have not been concerned with the actual engineering design, only the optimal location in order to put them where they are most needed. The engineering design and control of these facilities are something with which we have not been concerned.

Chairman Minnick: Frank Heart, Lincoln Laboratories asks, "How was the traffic survey done? Did they use vehicle counters?"

Walter Hoffman: The traffic survey was done by three means, first by a statistician who took samples of population density, and the densely populated areas of the city were chosen at random. In addition, these people also contacted the truck companies, and in this way they got the best available figures for traffic movement from them. In the third method we used various stations where vehicles were stopped, and drivers were asked the nature and the origin of their trips. Through these statistics, and the other information that we have just discussed, we thought we had an accurate picture of traffic conditions in 1953.

Chairman Minnick: Mr. D. H. Young, Ramo-Wooldridge asks, "Are the studies which have been described concerned primarily with prediction traffic loads, or do they consider such things as setting up one way streets and other devices to facilitate traffic flow?"

Walter Hoffman: No, they do not. As a matter of fact in the last current problem we were working on, we imagined a combined one-way street with an imaginary street in parallel, that is, the pair combined into one because our data are nondirectional.

Chairman Minnick: Mr. Olman, SDC, asks: "Has any consideration been given to monorail for 1980? How did you originally choose your zones?"

Walter Hoffman: The zones were chosen in an attempt to equalize roughly the population and traffic with the generating capacity of each zone.

As far as the monorail is concerned, it has not been taken into account, it has not been projected as yet. However, at the moment there is quite a bit of discussion about a monorail system being built. We will put it into our projection if it is built.

Chairman Minnick: K. M. Snapp, SDC asks, "Mr. Moore's algorithm for the shortest path through a maze considers unidirectional flow only. How did you modify the algorithm for your application?"

Walter Hoffman: I do not know that I agree with the original statement. One can use it in unidirectional or bidirectional flow, and no particular modification was imparted.

Chairman Minnick: Mr. A. J. Unger, Lockheed MSO asks, "How did your trip-desire map compare with one done by the rectangular grid method?"

Walter Hoffman: It is quite equivalent. I wish I were able to show you the results of this, but due to computer delay we were unable to get the slides ready. The trip desire map is distinguished from the other in that the grid is a little more coarse. The method we chose compensates for other traffic, and is set along straight lines.

Chairman Minnick: Here is an interesting question from Walt Kopinsky, Computer Control Company: "Assume a new city called 'X,' has any traffic system been devised to serve best this city? This assuming the city presently has no streets."

Walter Hoffman: Well the first thing that we would do would be to decide where we were going to put a lot of people, then design the streets to go around that. This is what we would do if we were going to design a model city. I am sure it could be done.

Chairman Minnick: Mr. Craig Weingarten, Institute of Transportation and Traffic Engineering asks, "Has any work been done, or is any work planned, on doing comparisons between different freeway networks; i.e., do you have to accept a predetermined network?"

Walter Hoffman: No, that is the point of our work. We want to investigate and modify all of the networks in the light of prediction to make them more functional, or to function better than the old designs.

Chairman Minnick: The first question for Mr. Johnson is from Don Swanson, Ramo-Wooldridge Corporation: "What methods have you used to evaluate the quality of translations produced?"

David L. Johnson: We are currently comparing meanings against a given text, that is, a sample against a meaning which our translator gives us. We are developing physical methods for evaluations of the output text, and are trying to work information carry into this sample of output text.

Chairman Minnick: The next question is from Paul Armer, Rand Corporation: "What medium do you use for getting text into machine readable form, punched paper tapes or cards?

David L. Johnson: Cards.

Chairman Minnick: Part 2, "Is your photoscopic memory working? If so, what do you use for input and output from it?"

David L. Johnson: This is a delicate point. A photoscopic memory has been working. It was developed in Los Angeles and was shipped back to the United States Air Force at Rome, N. Y. I now hear that it is in workable condition.

I cannot answer that second part of the question. We primarily work with the dictionary as a device for processing.

Chairman Minnick: "From where does your financial support come?"

David L. Johnson: We have gotten $200,000+ in about 3-year's time from the United States Air Force Research Group in Rome, N. Y.

Chairman Minnick: "Do you do any editing of the text prior to keypunching?"

David L. Johnson: No.

Chairman Minnick: The next question is from W. Buchholz, IBM: "In your economic analysis did you take into account the scarcity of human translators as opposed to the probability of greater availability of transcribing clerks?"

David L. Johnson: We considered it only at the University where there are numerous linguists who are always willing and eager to translate for a penny a word. When we made our study we found a basic

need for mechanical translation which stems from the requirement that there simply are not enough people qualified to translate technology and science.

Chairman Minnick: The next question comes from Larry Grodman, IBM: "In the case of Russian how are the symbols converted so that the machine can handle them?"

David L. Johnson: We have a code whereby we represent Russian, and a Flexiwriter keyboard.

Chairman Minnick: Mr. Emory Franks, SDC asks, "Do you consider punctuation in analyzing blocks of words?"

David L. Johnson: Very definitely.

Chairman Minnick: From Mr. I. L. Wieslman, Telemeter Magnetics: "Concerning costs of translation: Did you consider the salary of the key punch operator as against that of a trained bilingual scientist? Also did you consider 20 words per second output as against 40 words per minute for the input typist."

David L. Johnson: Yes, we looked at the over-all problem, and we did it on a per word basis thereby obtaining a per word cost. The time was involved in that on all of the machine work.

Chairman Minnick: From J. Guiteras, SDC: "Is a foreign language phrase translated, or is it just repeated in the English translation?"

David L. Johnson: I said that certain very common word combinations, even beyond well-defined idioms in the dictionary came out in their English form as if the machine were processing a single word. Of course, the only words of that kind used in that method are those in which there is no question as to how they will appear, regardless of syntax of the given text. When that is not true we must handle one word at a time, and translate later in syntax.

Chairman Minnick: From Paul M. Rubin, SDC: "How are non-English Russian characters converted to the *650*'s biquinary configuration?"

David L. Johnson: I think that I have already covered that. We use other symbols besides alpha and as the Russian is longer than our alphabet we use innumerable other characters that appear.

Chairman Minnick: Mr. E. M. Richter, Hughes Aircraft asks, "Lacking mechanical translators, are there enough human translators at 1 cent a word to handle the desired volume?"

David L. Johnson: No, otherwise we would not be getting paid for it.

Chairman Minnick: Mr. A. A. Cohen, Remington Rand Univac asks, "Do I understand correctly that your dictionary has card input and tape output, and that the output tape is transferred to the input? How is the printer page prepared for inspection?"

David L. Johnson: This has not actually been unitized to specify the input and output as such, but we assume that as soon as we get to the point of logical processing the output will be directly connected to the processing equipment.

Chairman Minnick: From Mr. F. M. Verzuh, MIT: "How many 'field of application' grammatical tags do you now use?" How many multiple meaning tags are used? How many zeros and ones are thus used?"

David L. Johnson: As far as our field of science is concerned we have a nine-by-nine matrix.

Concerning the number of grammatical tag positions, we have a tremendous swarm of them, and we find as we go along that we need more and more of them, so on a research basis we left this open. At the present stage, therefore, we are carrying along about 80 binary positions for tags.

Chairman Minnick: M. Richins, Arizona State University asks, "What method do you use in triggering the idiomatic expression for the dictionary lookup? Also do you translate a sentence at a time, or a word at a time?"

David L. Johnson: As far as the idioms are concerned, they are entered into the dictionary as a word, and our process pulls out the meaning of the idiom rather than the specific word. The dictionary is so ordered that long words will be identified before short words. In other words, the inverse alphabetization allows the search procedure to recognize complete words without unpremeditated dissection: i.e., were we matching English, the procedure would find "teasing" before "tea" and then "sing."

As concerns output, the dictionary is word by word. We hope that as soon as our methods become more defined, we can start translating block by block, and perhaps that may be extended to sentence by sentence. The ideal is a very difficult thing to realize.

Chairman Minnick: Mr. Tien Chi Chen, IBM Research Center asks, "What is your opinion on translating Russian into an artificial, but easily understandable language? This artificial language can be read by humans, or be further translated into English."

David L. Johnson: That is something that practically all people working on mechanical translation have been considering. We think that as long as we are hunting for Russian to English translation exclusively that we would be better off continuing to work from Russian to English. If the machine were required to translate from Russian to English to French, etc., it might well be more economical to include an intermediate language, common to all concerned.

Chairman Minnick: Mr. H. Kantner, Armour Research Foundation asks, "What are your plans for incorporating learning capability in a future program?"

David L. Johnson: That is undefined.

Chairman Minnick: From Mr. R. Hagopian, General Electric Company: "Is it your opinion that the electronic reader must be capable of reading many different type fonts, or can this be restricted?"

David L. Johnson: It is a matter of what you want, and what we would accept. Of course it would be nice to put any technical literature on to the type reader. If the text must be retyped into a particular form, we might as well keypunch it.

Chairman Minnick: Mr. R. H. Hill, University of California, City of Los Angeles, asks, "Have you attempted the approach of separate dictionaries for specialized literature, thus cutting down the contextual problem? What difficulties led you away from this approach?"

David L. Johnson: Our source is one relating to general technical language. Our dictionaries will not operate on high-level technical papers. When it comes to working on some "souped-up" paper, we need a completely different dictionary to use than what we have.

Chairman Minnick: The first question for Dr. Drandell is from Mr. E. G. Andrews, Bell Telephone Laboratories: "Why does your processing not include line-finder terminal assignment?"

Milton Drandell: Line-finder terminal assignment? There are other things that I have not shown here, and this will be done.

Chairman Minnick: Mr. Charles K. Budd, United States Army Signals Corps asks, "Is there any estimate as to number of people saved by using the automated procedure described, or is there a cost comparison of manual versus automated methods?"

Milton Drandell: Of course, I would like to say this is the reason that the Telephone Company is getting the machine, so that it will do the work of the people they are unable to hire. This is a common problem in public utilities today. There are approximately 300 people employed in the process at the present time; the majority of them are doing what is described in the paper and they will be relieved for other duties within the company where their services are needed.

As to the cost estimate, we hope that there will be a cost saving, but I cannot give you a figure at the present time.

Chairman Minnick: Dr. R. A. Kudlich, Bell Telephone Laboratories asks, "What criteria are used for selecting among several available pairs in the primary terminal? What is the average processing time per service order?"

Milton Drandell: The pairs are selected in optimum fashion so that the telephone installer, as much as possible, has to work on the primary terminal.

Chairman Minnick: The next question is from Mr. G. E. McAllister, Douglas Aircraft: "What is the estimated time required each day on the *705* for this assignment? How do you incorporate new central offices?"

Milton Drandell: The time required is within one shift of the operation. In regards to new central offices, the information pertaining to it is punched in the punch card and updated in our master files.

Chairman Minnick: Mr. W. E. Naftaly, United States Air Force asks, "What is the over-all savings in time by use of the computer method over the manual method, that is, from the time the customer submits his application to the time he gets his phone?

Milton Drandell: That is a very good question. The installation of an average assignment takes 3 days from the receipt of the order. Another thing I might mention is that we will have the service order completed by the night they are to be installed, but all of this is dependent upon the facilities in the field.

Chairman Minnick: Mr. L. W. Cali, ElectroData asks, "What is the daily processing time?

Milton Drandell: Within one shift of the operation.

Chairman Minnick: Mr. Cali's next question is, "Is the TRC used in this application?"

Milton Drandell: Yes, cut-off point is five o'clock.

Chairman Minnick: Mr. Harold Wells, General Electric Company asks, "Has a computer application been made for the prediction of 10-to-20-year system growth, and the economic assignment of capital expenditure for plant and equipment?"

Milton Drandell: They are certainly taking a hard look at this, and they have people assigned to it. For instance, in the area of inventory control, they may solve the location of warehouses by linear programming aspects. They are also making statistical analyses to predict future growth. I say that this is a definite possibility of the machine at some future time.

Chairman Minnick: The first question for Mr. Firth comes from Mr. Ohlman, SDC: "How much does your average search cost?"

F. E. Firth: This would be based on the machine rental cost. Since our laboratory is the home of the Ramac, we have an engineering prototype machine which does not cost anything. Labor is our cost.

Chairman Minnick: The next part of the question is, "How do you handle connections between words, that is, you have the words completely separated. Suppose you put the words together in some fashion?"

F. E. Firth: You mean if the words are "magnetic" and "recording", and we want to tie these together, just for one search? We look up in the manual dictionary the address for the word "magnetic" and the word "recording," punch the two on the insert card; that would be the terms *A* and *B* as shown on the slide. We can take up to nine different words, with "and's" or "or's" between them, on one search.

We have also considered the possibility, although we have not done this permanently, of combining the two words in the dictionary. If "magnetic recording" always appeared together, this would be a wise thing to do. This decision that they will always appear together is a hard one, and we are steering shy of it by keeping them separate.

Chairman Minnick: The next question is from Frank Heart, MIT, Lincoln Laboratory: "How many people have been involved in this effort?"

F. E. Firth: Directly three; myself, and two keypunch operators who make up the cards. I would like to give a great deal of credit to the librarians, as well as to all of the personnel in our laboratory. This was not a one-man project by any means.

The way we get descriptive terms for record prints and pamphlets, which we believe is most valuable, together with the coming report, is by sending an index card to the requester of information when he receives the document. This means that the man who logically knows most about the subject is the man who defines the index terms for us, and we do not have to rely on guessing in some field. The specialist or his associates will probably be the one asking for information on this subject. It seems logical that they will be using the same terminology we were able to put into the machine.

Chairman Minnick: The next question is from Mr. R. Solomonoff, Zator Company: "How are terms assigned to documents? Are they simply words in titles of documents?"

F. E. Firth: I just described how we handle the documents requested specifically by engineers and company-coded reports. These words are selected by the requesting engineer, and whatever words he might give us are the ones we will use. They may appear in the title, but usually it contains the best words, however, this is not always true.

To give you an example, "Can the machines think?" We did not use this title at all, and we had quite a time deciding what to use. Now I would like you to know that for the periodical information we use titles alone. We put in Review of Scientific Instruments, and we enter the important words in titles.

Chairman Minnick: The next question is from Lauren Doyle, System Development Corporation: "How are the descriptors assigned? Has automatic assignment been considered? Is it feasible at all with your equipment?"

F. E. Firth: If we were willing to keypunch the entire text or abstracts perhaps, I believe it is feasible. We have not attempted this, although we have discussed it at great length. This, of course, is the crux of the whole problem.

Chairman Minnick: From Russell A. Kirsch, NBS: "Are there any advantages in using the Ramac over a drawer of 3 by 5 cards?"

F. E. Firth: This is a question for which we have been trying to find an answer. Why spend money to use a machine, when it might be less expensive to do it manually? I do not know of any solid facts to say one way or the other. But I believe this to be true, if you are searching for two terms, which only appear once in the system, with only one serial number posted after each word, I will not question that it would be faster, cheaper, etc., to do it manually. But with only 5,000 documents wherein the word "magnetic" appears more than 350 times, and the word "recording" appears 300 times (these are two that I happen to know offhand), when you get to 25,000 such a system could be envisioned, but at any more than that these numbers, serial posted numbers, skyrocket. I suggest that you would profit with a simpler machine, and Ramac would be one of them.

Chairman Minnick: From Mr. V. Weidemann, Physikal Techn. Bundesanstalt: "What amount of time was spent in assigning the dictionary words to the 5,000 documents?"

F. E. Firth: Let me try to divide this question as to assignment of terms: We send a little card out to the engineer. After reading the document, he writes down how much time it would take to assign the term. It does not involve a great deal of time; it is spread over a great number of people, so it does not really hurt us. As for the assignment of terms of a scientific periodical, we go through a list of titles and eliminate the "and's," "or's," and "but's," and put in the significant words. I would estimate that this thing might take a fraction of a minute per word.

Chairman Minnick: Mr. M. I. Rosenthal, SDC asks, "What would this type of retrieval cost those not fortunate enough to have an engineering prototype of the Ramac?"

F. E. Firth: I believe that the rental of the Ramac System is on the order of $3,200 a month. I am sure that our sales people would be very glad to give you better information.

Chairman Minnick: A question from Mr. Sidelford: "Have you considered using the new IBM electronic computer in your system?"

F. E. Firth: I believe it would be valuable in finding the terms. We do not have a *704*, unfortunately.

Logical Design Method

G. H. AMDAHL
NONMEMBER AIEE

THE LOGICAL Design Methods panel will be concerned with two different approaches to the problem of logical design, block diagrams vs. logical equations. In introducing this subject, it is necessary to define the region of the over-all computer design area which may be identified as logical design.

Logical design is sometimes used to describe both the systems design and the logical design functions involved in the over-all computer design area. For the purposes of this panel discussion the systems design of a computer is considered to be a separate function from the logical design. There does exist, however, a certain gray area which lies in both the areas of system and logical design. For purposes of this paper the systems design will be assumed to be the area of specification of the externally employable properties of the computer system, including the general organizational aspects of the internal functional assemblies of equipment. Logical design picks up at this point and consists of the detailed specification of the arrangement and interconnection of logical and storage components necessary to provide the required properties of the functional assemblies as determined by the systems design requirements.

In the process of performing this detailed structural design the logical designer will frequently determine that certain changes in the functional organization as specified by the systems designer may more readily or economically provide the desired external features for the computer, or alternatively, he may uncover the need for additional functional assemblies to execute some features of the required system specifications. It is these latter contributions by the logical designer to the systems design area that gives rise to the gray area between systems and logical design and which makes a clean distinction between the two areas difficult if not impossible.

A less frequent definition of the term logical design includes the design or fairly detailed specification of the logical and storage circuitry to be employed in the computer. Again there exists a gray area inasmuch as the logical designer must possess a reasonable knowledge of, and exert some influence upon, the engineering and logical properties of the components with which he must conceptually construct those portions of the computer for which he bears a responsibility.

For the purposes of this panel discussion the area designated logical design will include a modicum of the gray areas common with systems design and circuit design, in order to illuminate practical considerations involved in logical design methods.

The tasks and responsibilities of the logical designer are manifold. To him falls the problem of splitting and resplitting the specified operational requirements till they are reduced to their multitudinous individual elemental decisions. When reduced to the elemental decisions his problem is then to reconstruct the original system with the logical componentry available to him.

During the process of reducing the operational specifications to their elemental decisions he must co-ordinate his efforts with those of other logical designers who are performing this same operation on their portions of the over-all system. In this co-ordination process the team of logical designers maintain as their goal the identification of as many common denominators throughout the system as are recognizable.

During the reconstruction process the logical designer works toward the goal of achieving a relatively well minimized assembly of logical elements to achieve the desired functions in all their detail. During this process he must also stay within the bounds of the physical limitations imposed upon him by the properties of the circuit elements which he employs.

Upon completion of the reconstruction it is the task of the logical designer to provide the details of this design to the engineers and technicians in a form suitable for physical construction of the computer system. Along with the provision of this detailed structural information, the logical designer must also pass along at least a limited concept of the logical behavior of this creation.

This Logical Design Methods panel will be primarily concerned with two schools of thought in logical design methods. One of these schools of thought emphasizes the use of block diagram techniques employed for the process of reduction and reconstruction. This school of thought enjoys the distinction of being the oldest and most widely employed.

The other school of thought emphasizes the use of logical equations for the process of reduction and reconstruction. This school of thought arrived upon the computer scene at a considerably later date and has enjoyed a very rapid growth in its adherents.

The use of block diagrams for design does not imply complete avoidance of the application of Boolean algebra but rather the limitation of its application to the minimization of logical elements in restricted portions of the circuits. Similarily, the use of logical equations does not imply the complete avoidance of the use of block diagrams but assumes the restriction of their use to a relatively gross picture of the system operation.

Some of the practical points to be discussed on both sides of the panel may be classed in essentially three major categories.

The first is flexibility or ease of: (a). coordinating individual designer efforts, (b). introducing nonlogical components such as amplifiers, (c). adapting new sets of logical operators available in components, (d). checking logical designs, (e), introducing computing machine aids to design, and (f). designing to satisfy limitations imposed by circuit engineering specifications.

The second is education or ease of training new designers and maintenance personnel.

The third is utilization or ease of: (a). translating wiring diagrams or direct wiring, (b). performing maintenance, and (c). determining overly restrictive circuit design limitations.

G. H. AMDAHL is with Aeronutronic Systems, Inc., Van Nuys, Calif.

Block Diagrams in Logic Design

LOWELL S. BENSKY
NONMEMBER AIEE

THE THESIS proposed in this paper is: While logic equations and even Boolean algebra are not essential to the block diagram approach to logic design, considerable penetration of block diagrams into the logic equation approach is unavoidable. In support of this thesis, several aspects of logic design will be investigated in the body of this paper. Emphasis will be given to the creative nature of logic design.

The generation of basically new configurations and new logic systems will be discussed. Next, the question of accommodating nonlogical elements will be explored. Finally, some problems in the transition from logic design to physical equipment will be pointed out.

Logic design, as the term will be defined and used here, is in the last analysis an art. The individual logic designer, like the artist, must develop his techniques according to personal preference, training, and ability. A discussion on the basic approach and execution of details for a logic design must consider the individual and the specific problem; no universally optimum logic design procedure, block diagrams or logic equations, will be found.

Definition of Logic Design

For the purposes of this paper logic design will be taken to include the activities pursued by logic designers in conjunction with system analysts, programmers, circuit and equipment engineers, and operations-maintenance planners resulting in the detailed interconnection and synchronization of a set of logical elements which satisfy the requirements for a digital computing device. Working with system analysts, the logic designer will determine from the problem specification: computation rate, input-output rates, input-output data format, capacity and types of data storage, and word length. With the help of programmers, the logic designer will determine: word length, operations required, instruction execution times, and programming aids. Together with circuit and equipment engineers, the logic designer will specify physical and operating characteristics such as: volume, weight, power, reliability, acceptable environmental conditions, and available hardware techniques. The logic designer and operations-maintenance planners describe the level of field programming and operating and maintenance personnel.

Based on these specifications and requirements, the logic designer will take one or more cuts at the gross logic of the computing device. Sample programs, cost estimates, and manufacturing schedules are worked out for each cut. The most likely candidate will be logic designed in detail and checked thoroughly. Exhaustive programming will be performed to insure adequate problem satisfaction; logic changes are introduced as required. Working closely with circuit and equipment engineers, the logic is modified to accommodate circuit restrictions and to exploit hardware techniques. Then the logic is put in form convenient for establishing wiring diagrams. After the computing device is fabricated, the logic designer is a member of the debugging and checkout team. Errors and unwanted redundancies which show up in this phase are corrected by the logic designer. Finally, the logic designer plays a major role in preparing preventive and corrective maintenance procedures. All of these activities are essential parts of logic design. The arguments which will be presented have significance when interpreted through this definition of logic design.

Creative Aspects

To derive a logic design to solve a new computing problem or to substitute a more elegant design for a brute force approach requires a measure of creativity. An important ingredient in the creative process is unencumbered visualization. In the case of logic design, creative efforts depend on incisive visualization of both spatial arrangements, timing and the interaction between them. When the logic designer is faced with the problem of forced invention to solve a problem or when he attempts to invent in order to improve on an immediately obvious or known solution, the inventive process is facilitated by a complete comprehension of: 1. data flow paths, 2. the factors making up control information, 3. the timing relationships among computer units, and 4. the interdependence among these items. Further, since new ideas are very often fleeting phantoms, comprehension which measures practicality should be rapidly attainable. A block diagram of a computing device affords rapid and complete comprehension of: 1. the required major registers, counters, and memories, 2. the data transfer paths and associated control gates, 3. the role of control information, and 4. the required timing and synchronization. Creativity and invention in taking initial cuts on the gross logic of a machine or in detailing the final logic are enhanced by the over-all perspective characteristic of the block diagram approach. The logic designer using the logic equation approach is likely to bog down in symbolic representation and abstract equations diluting the opportunities to invent.

Since the early days of the Electronic Discrete Standard Analog Computer (Edsac), Harvard, Electronic Discrete Standard Analog Computer (Edvac), and Impulse Automatic System (IAS) computers, several basically new logic designs and new logic ideas have been created. Solidifying and testing these new ideas were facilitated by use of block diagrams.

Block diagrams paved the way for new logic ideas by showing functional feasibility and/or economic advantage: for integrating a new part into an existing structure, for replacing a part of a structure with a different implementation, or for assigning additional tasks to existing parts of the structure. It might be interesting to speculate on how many of these developments would not have emerged had logic designers depended exclusively on logic equations, restricting block diagrams to the simplified picture. The B-box, or instruction address modifier, would probably not have joined the ranks of such standard computer elements as the instruction counter.

The psuedo-parallel arithmetic using a phased clock for high-speed computation would not have been invented and used in computers like the Livermore Atomic Research Computer (LARC) and UDOFT. Variable word length and several of its important implications would not have been applied to digital computers like the International Business Machines Corporation (IBM) *705* and Radio Corporation of America BIZMAC.

For example, the freedom to rearrange the variable length items of a BIZMAC message during read-in from magnetic tape to high-speed storage (random composition on input) would not have been recognized as a feasible addition to the

LOWELL S. BENSKY is with Rese Engineering, Inc., Philadelphia, Pa.

BIZMAC order code. This operation allows the programmer to specify an arbitrary arrangement of the items of a message in the high-speed store, which arrangement may be different from the sequence of the items on the input tape. This rearrangement is accomplished with items of variable length even though each item may not be consistently present in a group of messages. In this example, several new ideas were generated during the gross and detailed logic design phases: 1. Integrating into a 3-address computer a variable length instruction having many more than three addresses; 2. Releasing memory locations holding this instruction and making them available for input message storage; 3. Changing the mode of operation under control of input message characteristics and allowing flexible cycling among the several modes. These ideas were to a large extent fostered by the complete comprehension of a block diagram showing previously established flow paths, timing, registers, and control flip-flops. The latent possibility of random composition on input would have been lost if this instruction had been designed by adding terms to the set and reset equations of flip-flops established for the other instructions and for a straightforward version of the read-in instruction.

Nonlogical Elements

The term "nonlogical element" might imply that in logic design these elements assume secondary importance. Yet, in practice nonlogical elements, which solve circuit restriction problems, marriage problems between separate units, and timing problems, are essential and often represent a major fraction of the equipment in a computing device. It is, therefore, important that these elements be introduced during the logic design process and not as an afterthought following the completion of the logic design. Only in this way can an over-all equipment optimization be approached, rather than just an optimum logic design.

A discussion on the accommodation of nonlogical elements can be divided into two parts: internal logic design and input-output device integration. Internally, digital computing devices require cathode or emitter followers, clamping circuits, pulse amplifiers, nonlogical delays, pulse stretchers, etc. Such nonlogical elements have a direct effect on: the number of components in a machine, the size and weight, power dissipated, cost, and operating reliability. Further, nonlogical elements have an effect on logic design minimization decisions. For example, one may choose central decoding of a function and use power distribution of the decoded levels or pulses to their points of application; or it may be more advantageous to distribute the function and decode separately at each point of application. A similar decision may be encountered between the use of separate corrective timing delays or a more complex time pulse distributor. Use of block diagrams will rapidly yield the number and physical location of inputs to be driven by each flip-flop, decoder, and pulse source output. Decisions on constraints or modifications to the logic can be made directly. Nonlogical elements can then be included on the diagram using simple and unobtrusive symbols, thus insuring that the elements are not overlooked.

Similar arguments apply to pulse standardizers, timing oscillators, one-shot multivibrators, and terminating or coupling filters as used in the marriage of input output devices and the central computing device. Decisions on the types of input output devices, their modes of operation, and associated modification and addition to the computer logic are facilitated by block diagrams. The use of block diagrams permits the evaluation of several eligible input-output devices. A decision to select a given device can be made with proper consideration to associated nonlogical elements. For example, the opportunity is minimized to neglect special matching filters required when a telephone line is used as an input to a data processor. Again, a block diagram will minimize the chances of deciding that a computer shall drive a group of tabular and graphical displays, forgetting to include the bank of buffer amplifiers required.

Space Allocation

The next argument is weak for computer development and construction programs which are allowed to proceed leisurely. Since this is usually not the case, the questions of what kind and how much rack, cabinet, or panel space cannot wait on a final logic design. A reasonable estimate can be made on the quantity of space required for each unit in a computer from a block diagram in process. This estimate can be used in selecting one of several logic approaches and it can also be used to give the mechanical and fabrication activities much needed lead time.

Another point in this connection is the use of the block diagram to come up with a physical configuration. The number of leads passing from unit to unit in the computer and special conditions imposed, such as lead length, separation from chassis and other leads, etc., are quite evident from a block diagram showing logical and nonlogical elements. These factors can then be used to locate each separate unit within the physical computer package.

Conclusions

It is the author's belief that the block diagram is the *sine qua non* for creative, efficient, and complete logic design. It is required not only in the conceptual phase, where several gross logic designs must be evaluated and creativity must be fostered, but also in the detailed logic design, in co-ordinating with circuit engineers, and in planning a construction program. Concession to the opposition is appropriate when a large, relatively pedestrian computer is called for and if: 1. the detailed logic design is a significant fraction of the total computer cost, and 2. the techniques and machine time are available for going directly from equation to wiring schedule with complete checking on a computer.

Except on these conditions, the author recommends employing a well-ordered block diagram and avoiding a more academic approach.

Logical Design Methods

R. K. RICHARDS
ASSOCIATE MEMBER AIEE

IT WAS recently reported in a magazine that Theodore Roosevelt, when a student at Harvard, objected to the idea of arbitrarily assigning a student to a topic to be defended in a debate regardless of the beliefs and convictions of the student. He felt that this practice was not in the best interests of sincerity and intensity of conviction. In the present instance the author has been asked to defend the use of block diagrams in computer design in contrast to the use of logical equations (Boolean algebra). Inasmuch as the author has made frequent use of logical equations, it would seem appropriate to, first of all, make a statement as to my true thoughts on the matter to allow this panel discussion to be maintained on a basis which is as sincere and objective as possible.

As the chairman, Dr. Amdahl, has already pointed out, for purposes of this discussion the use of block diagrams does not necessarily imply that Boolean algebra is not to be employed at all. It may still be used as an aid to design and as an aid to notation in portions of a computer, but it is implied that the main body of the engineering design, production, and maintenance effort is to be accomplished through the medium of block diagrams rather than logical equations. The question is whether or not it is advisable to go all the way and eliminate block diagrams completely insofar as the basic logical functions are concerned. Instead, complete reliance would be given to logical equations for designing the computer, for communicating the design to the manufacturing department, and for leading the maintenance engineer to potential and actual defects in the machine. The author's true thoughts, then, are that Boolean algebra is a very important tool for computer applications but that the advisability of attempting to rely on it entirely is open to question. As is discussed in more detail later, some types of computer components and circuits are such that it is substantially impossible, with the present state of the art, to get very far with an algebra in a manner that is useful. With other types of components and circuits it is quite possible to represent the entire computer by means of logical equations, and this has been done in some cases. Therefore, in a sense, a discussion of the relative merits of block diagrams and logical equations reduces to a discussion of the relative merits of the different physical realizations of logical functions. In addition to this factor in the comparison, there are other important factors, to be discussed later, which would tend to make it desirable to retain block diagrams even in instances where the circuits and components are of a type that allow a complete shift to Boolean algebra.

The first point is more of a warning than a concrete reason why block diagrams are superior to logical equations. Scientists, mathematicians in particular, are prone to use the concept of "elegance" in judging the merits of a solution of a problem. Many problems have more than one solution, with some solutions being more elegant than others. Individuals strive to find solutions which are as elegant as possible because such solutions command more prestige and a higher status in the scientific community. Elegance is a rather difficult property to define, but it seems that by almost any set of standards logical equations are much more elegant than block diagrams. However, one must be extremely careful in drawing conclusions from this fact because intellectual elegance is often a poor measure for weighing the merits of a proposal when something as down-to-earth and practical as a computer is involved. A good illustration of this point is in the very use of computers to solve mathematical problems. It has long been known that many types of mathematical problems, most forms of differential equations for example, can be solved readily by numerical techniques even when they do not yield to any known analytical approaches. However, numerical solutions have never been considered elegant; consequently, many mathematicians have shunned numerical approaches to problems in preference to the more elegant analytical approaches in spite of their limited application and usefulness. Now that computers are becoming so expensive and glamorous the interest in numerical solutions is increasing rapidly. It might be argued that in colleges a course in numerical analysis is now even more important than a conventional course in differential equations, but that is getting off the subject. The purpose here is to emphasize that the most elegant approach is not necessarily the most desirable approach, and promised advantages of an elegant approach are not necessarily always realized when the problem is set in a practical environment.

As is now well known, there are many different ways to go about the design of a digital computer. Even a casual inspection of a representative sample of the computers as built by different industrial, educational, and government organizations will reveal a vast difference in the nature of the components and circuits or in what is sometimes called the "design philosophy." It has often been noted that with each different set of components and circuits there are a relatively few basic logical functions being performed. These functions are "and," "or," "not," "delay," and "storage," or certain combinations of these functions such as "exclusive or" and "inhibit." However, in spite of the common denominator which the basic functions afford, it is not true that all of the different approaches are equally amenable to Boolean algebra.

When adapting Boolean algebra to a computer in its entirety, it appears to be the situation that the most success is achieved when the following concept is employed in both the logical design and the circuit design. The computer is viewed as consisting of a large number of bistable elements, and time is divided into a series of equal increments or steps. At the end of each step, the binary values stored in the bistable elements are functions of the binary values at the termination of the previous time step. The relationship between the binary values from one step to the next can be represented by logical equations.

A major limitation of logical equations is encountered when it is desirable for one reason or another to employ components and circuits which do not function in this straightforward step-by-step manner. Consider, for an example, an ordinary binary counter consisting of four binary elements capable of counting to 16. As is well known, the counter can be made to function by causing the pulses to be counted to be applied to one of the four binary elements connected in a complementing circuit. Whrn the binary value in this binary element changes from 1 to 0, a pulse is caused to be transmitted to a second binary element, also connected in a complementing circuit. Pulses derived in this manner are caused to be sent from one binary element to the next in a "ripple-carry" type of circuit. To assemble the counter, nothing more than

R. K. RICHARDS is a consulting engineer.

the four binary elements is needed. However, to employ logical equations to describe the counter it is generally necessary to perform a more complex analysis of the counting action. For example, the input pulse is caused to reach the second binary element when it arrives "and" when the first binary element contains the value 1. The input pulse is caused to reach the third binary element when it arrives "and" when both the first and second binary elements contain 1's. A straightforward extension of this logical arrangement with the "and" functions is used for the remainder of the elements in the counter. These "and" functions may have separate physical identities of their own, and with components and circuits where this is the case, logical equations have meaning and usefulness. However, as was mentioned, there need not be an "and" circuit physically identifiable between successive binary elements in the counter. When no separate "and" circuits are present, the use of logical equations becomes awkward and substantially meaningless. On the other hand, it is a simple matter to represent the counter with four rectangular blocks with a line leading from one block to the next to represent the carry.

If this analysis is correct, the question then becomes a matter of whether or not the simplified counter is ever better than the step-by-step "Boolean counter" in the first place. This aspect of the question can become quite complex because of the wide variety of applications in which binary counters might be used. Binary counters are useful not only as instruments by themselves but also as parts of general-purpose digital computers and other digital systems. It seems safe to say that there are applications where it is not only sufficient, but also preferable, to assemble a counter with nothing more than a series of bistable elements (flip-flops) where the output of one is used as the input to the next, a configuation for which logical equations are awkward and unnecessary.

The second example that will be cited where the use of logical equations becomes strained is in parallel operation. Consider a parallel accumulator circuit. In a parallel accumulator circuit the sum digit in any given order is a function of the digits of both factors being added in all lower orders of the number. In arithmetic units employing an end-around carry, all sum digits are functions of the digits of all orders in both factors. As a result of this situation, the logical equation expression for any given sum digit becomes lengthy and complex regardless of the nature of the components and circuits in use. Some simplification can be achieved through the use of subscripts or other identifying marks to represent the values of the signals corresponding to the "carry" signals that can be viewed as being transmitted from one order to the next. When such steps are taken it is important to keep in mind that the notation is the means and not the end. It has been observed that computer designers and mathematicians often become so engrossed in attempting to use their notation that the original problem becomes obscured.

Whether or not a complex notation is used for a parallel accumulator it is found that there are many physical realizations of the device for which the logical equation approach is not adaptable. A good example is an accumulator comprised of a set of complementing flip-flops wherein the flip-flops initially store one of the two numbers to be added, and the other number is applied as a set of simultaneous pulses to the complementing inputs of the respective flip-flops. The carries can be generated by any of several different means, but regardless of the carry generation scheme it has been found quite difficult to devise a meaningful logical equation notation for this type of circuit.

If logical equations are to be used in a parallel accumulator, it is found that the most natural correlation between the notation and the circuits is achieved if two static registers (sets of bistable elements) are used to hold the two numbers to be added with a pulse such as a clock pulse, used to set the elements in one register to the sum after this sum has been determined by logical elements. Even this type of accumulator contains the notation difficulties mentioned previously. Nevertheless the question does again become largely a matter of determining which types of circuits are the most desirable. In favor of the circuits not amenable to logical equations it will be mentioned that it is possible to devise a parallel accumulator with nothing more than the flip-flops required to hold the accumulated sum if a slight amount of delay is introduced between one flip-flop and the next for the carry, and this delay can be the inherent delay in the flip-flop itself with no extra components being required. So far as is known, all parallel accumulators derived by the logical equation approach have consumed many more components than this.

Before leaving the subject of parallel operation, the observation will be made that in Montgomery Phister's recent book, *Logical Design of Digital Computers*,[1] which employs the logical equation approach almost exclusively, the author, on page 260 recognizes the limitations of the approach and resorts to a block diagram to illustrate how a practical parallel adder functions. Also, Mr. Phister, on page 321, makes the following statements with regard to a parallel comparator: "The design of a parallel comparator for two numbers residing in two flip-flop registers is easily written. However, the resulting expression is a function of every memory element in the two registers, and is therefore expensive to mechanize." The conclusion about the expense of a parallel comparator is not unexpected when working exclusively with logical equations, but it happens that a parallel comparator is not particularly expensive to mechanize. Several different logical arrangements can be found quite easily by using block diagrams. The arrangements are similar to the carry circuits in a parallel adder.

A third example which will be cited to illustrate the power of block diagrams and the limitations of logical equations is a type of circuit rather than a particular application. The circuit has appeared in several forms, mostly in types of operation known as asynchronous. The "gated pulse amplifier" is one form of the circuit. An ordinary pentode tube may be used in the gated pulse amplifier. In this case a pulse is applied to the control grid and a steady-state signal is applied to the suppressor grid and thereby allows the pulse to pass or not to the output line in the anode circuit. The steady-state signal is said to "gate" the pulse. The pulse generally passes from one such circuit to the next, and in so doing it may pass through diode "or" circuits. (The pulse may also pass through "and" circuits, but a timing problem is introduced, so this particular variation is usually avoided.) The steady-state gating signal at each pulse amplifier may be generated through the use of arrays of "and" circuits, "or" circuits, and inverters. In actual practice, arrays of these circuits can become quite sophisticated with two or more pulses following each other along chains of the gated pulse amplifiers. Because of the timing relationships of the pulses and because some signals in the array are pulses whereas other signals are of the steady-state type, a logical equation approach to design and notation in a computer as a whole has not been found to be feasible when circuits of this kind

are involved. However, with block diagrams the flow of signals is readily visualized, and the nature (pulse or steady-state) of the signal at each point in the array is readily notated through the use of two different types of arrowheads. This pulse amplifier type of circuit should not be confused with the synchronous pulse amplifier technique, wherein a clock pulse is used to retime and reshape the pulse at each amplifier. Static flip-flops are used with the gated pulse amplifier.

The justification for using the gated pulse amplifier technique in contrast to a type of step-by-step circuit amenable to logical equations is in the high speed of operation that can be obtained in developing multipliers, dividers, and other devices for performing complex logical operations. Through the use of block diagrams it has been found quite easy to assemble the desired arrays in an economical manner with no particular need for logical equations being felt.

The next point in favor of block diagrams in contrast to logical equations is one considered so obvious that it will only be touched upon. Nevertheless, it is of great, perhaps even crucial importance. This point is that the use of block diagrams requires less training for the persons associated with a computer than does the use of logical equations. It has been observed that most people understand block diagrams with no instruction at all, whereas a substantial course of instruction is required for an average person to develop an understanding of the notation of logical equations. Further, to make good use of logical equations, a course of instruction is not sufficient; a substantial period of experience is necessary. It has been observed that some people who can work quite well with block diagrams never do develop an ability to employ logical equations in a useful way. In spite of this factor, a computer designer should probably be expected to have an understanding of logical equations, but in the manufacturing and maintenance aspects of creating a useful machine, it certainly seems a desirable objective to utilize personnel having a minimum of formal training. In this regard the block-diagram approach to notating a machine's structure appears to have a tremendous advantage in comparison with logical equations.

The last major point in favor of block diagrams concerns the relationship of a block diagram to the physical wiring in a machine. The physical wiring is, of necessity, a 2-dimensional array. Actually, the third dimension is usually involved also, but with few exceptions, the third dimension is used for stacking layers of the 2-dimensional patterns. For example, a computer can be assembled through the use of a multiplicity of printed wiring boards, each of which holds a circuit in a 2-dimensional pattern. The printed wiring boards are then mounted in a frame which holds the boards in a 2-dimensional array with "back panel" wiring interconnecting the boards. Countless variations are of course possible, but with regard to the comparison of block diagrams and logical equations it should be noted that block diagrams are basically two dimensional in nature whereas logical equations are basically one dimensional in nature. Block diagrams therefore, lend themselves quite naturally to a pictorial representation of the physical layout in the computer. This factor is of some importance in engineering design work in facilitating short-lead lengths and for other reasons, but its outstanding advantage arises in manufacturing and servicing where a close relationship between position in the logical array and position in the physical array can be readily established. Even in instances where a one-to-one relationship on the block diagram and in the computer is not convenient, a close correlation can be maintained through special notations as required on the blocks. For example, in the block notation for a flip-flop the Cartesian co-ordinate designation for the flip-flop position can be recorded in a specified corner of the block. In another corner of the block the precise terminals, either on a printed wiring card or on the back panel wiring, can be indicated. Further, it is possible to indicate on the block the distant terminals to which the input or output terminals are connected. It is difficult to visualize a logical equation notation that can compete with the completeness, flexibility, and simplicity of block diagrams in this regard.

It was originally intended to mention specific computers in which the block diagram approach has been used to advantage, but Dr. Amdahl has already pointed out that block diagrams are of the "Eastern school" with logical equations being of the "Western school." Therefore it is merely pointed out that, to the author's knowledge all computers designed in the East plus at least a portion of those designed in the West have had the benefit of the block diagram approach (except in isolated portions of a given computer, as allowed previously). While all of the computer designs will probably become obsolete in time, no particular competitive pressure has been noted on any of them as a result of fabulous or even moderate improvements made possible through the application of logical equations to the exclusion of block diagrams.

Conclusions

Although logical equations are a useful tool and every good computer designer should be expected to have a knowledge of their use, there are several factors which mitigate their exclusive use with a complete elimination of the more conventional block diagram approach. First, the elegance of the logical equation approach should not be allowed to blind one to the practical aspects of designing, building, and maintaining a computer in the most effective manner. Second, there are many different types of digital circuits, and the various types are not all equally amenable to the logical equation approach. The use of logical equations therefore imposes severe restrictions on the range of circuit types available to the designer. Third, although a sophisticated use of logical equations will produce an economical logical array when considered in terms of the circuit types to which logical equations are adaptable, still more economical logical arrays can often be found when the range of circuit types is not so restricted. Fourth, block diagrams allow the use of personnel with a lower level of ability, education, and experience, a factor which is particularly important in manufacturing and maintainance. Fifth, the 2-dimensional nature of block diagrams offers many possibilities for indicating the correlation between logical functions and the locations of the blocks, including specific terminals, in the computer.

Reference

1. Logical Design of Digital Computers (book), **Montgomery Phister.** John Wiley & Sons, Inc., New York, N. Y., 1958.

Machine Language in Digital Computer Design

H. L. ENGEL
NONMEMBER AIEE

IF THE description of the logical structure of a digital computer is in a form that can be processed by a digital computer, this description is said to be in a machine language. The notation of Boolean algebra is a particularly suitable machine language for the logical description of digital computers. The advantages of Boolean algebra notation in the logical design of digital computers has been amply pointed out in the companion paper.[1]

Machine language has three major advantages. First, it can be used to produce lists of all elements that use the output of a particular element. These are referred to as logic and usage tabulations. Second, machine language permits simple generation of a wiring tabulation indicating all signal wiring in the computer. Third, machine language allows simulation of the logic of one computer by another entirely different computer. Benefits that result from each of these capabilities are discussed in the remainder of this paper.

Logic and Usages

Table I shows the logical equations for the set and reset inputs to several flip-flops and for the gates that are used in these expressions. The set input to flip-flop *F019*, for example, is composed of three terms. The first term has three factors, namely the complement output of flip-flop *F013*, the normal output of flip-flop *F112*, and the complement output of flip-flop *F019*. In the second term there are four factors; three are outputs of flip-flops, and the fourth the output of a gate, *G007*. The final term in *F019S* has three factors, one of which is the output of a flip-flop and the other two the outputs of gates.

Below the equations for the flip-flop inputs in Table I are the equations for all of the gates used in these expressions. The gate expressions also can consist of one or more terms, with factors that may be either flip-flop outputs or gate outputs.

These gate and flip-flop input expressions may be recorded on punched cards to form a logic deck. Each card in the logic deck contains a single term and the name of the flip-flop input or gate input of which this term is a part.

A machine listing of the logic deck is called a logic tabulation. The logic tabulation corresponding to Table I appears in Table II. In this tabulation, as in the logic deck, parentheses and plus signs do not appear. They are unnecessary since the tabulation contains but one term per line and since the factors in each term are separated by spaces.

A new deck of cards known as the usage deck and a new tabulation known as the usage tabulation are manufactured by machine operations. In both the deck and the tabulation, each use of the output of a flip-flop or a gate is listed. A usage tabulation corresponding to the example of Table II is shown in Table III. Note that there is a line in Table III for each factor in Table II since each appearance of a factor in Table II represents one use of that quantity. For example, the first term in Table II, *F013' F112 F019'*, appears in Table III under the uses of *F013'*, *F112*, and *F019'*.

In the logic tabulation for an entire digital device, the usage tabulation indicates in a very convenient form all uses of all outputs of the digital elements. In this abridged example, for instance, Table III shows that the normal output of *F019* is used in two terms of the reset input to flip-flop *F019* and in one term of the set input to flip-flop *F020*. *F019* is used in no other place.

The logic and usage tabulations help the logical designer to determine that the system design rules imposed by circuit considerations have been observed. A casual examination of the logic tabulation will indicate if the allowable number of inputs to an "and" gate have been exceeded or if the number of terms that form the inputs to an "or" gate is greater than permitted. If the total number of gates that can be driven by the output of a flip-flop or a gate is limited, examination of the usage tabulation will quickly reveal whether this limit has been exceeded. If the total load on a gate or flip-flop is a function of the number of gates it drives and of the number of inputs to each of these gates, then the usage tabulation provides information in a convenient form for calculating an upper limit to the load that will be placed on any gate or flip-flop.

The logic and usage tabulations also assist the logical designer in redistributing gates to make the logical structure of the computer conform to circuit limitations. The tabulations indicate whether some gates or flip-flops are far more heavily loaded than others, and so demonstrate the desirability of reformulating logical expressions or duplicating some gates or flip-flops. When, as often occurs, there are several alternate descriptions of a logical function, the logic and usage tabulations aid in selecting the description that will most uniformly distribute gating loads.

The logic and usage tabulations are also useful in production design of a digital computer. If diode gating is used, an accurate count of the total number of diodes required for gating can easily, accurately, and rapidly be calculated from the logic tabulation. Further, these tabulations provide a good insight into the types and numbers of standard plug-in units that should be

Table I. Logical Equations

F019S = (F013')(F112)(F019')
+ (F013)(F112')(F019')(G007)
+ (F058')(G284)(G097)
F019R = (F013')(F112)(F019) + (G007)(F019)
+ (G003)(G007)(G097)
F020S = (F013)(F019)(G007) + (G003)(F013')
F020R = (F020)(G097)
F021S = (G007)(G097)
F021R = (F020')(F019') + (F021)(F019')(G284)
G003I = (F020')(F058) + (G284)
G007I = (F013)(F020')
G097I = (G284) + (F020')
G284I = (F112')(F113)(F114')

Table II. Logic Tabulation

Input	Term			
F019S	F013'	F112	F019'	
	F013	F112'	F019'	G007
	F058'	G284	G097	
F019R	F013'	F112	F019	
	G007	F019		
	G003	G007	G097	
F020S	F013	F019	G007	
	G003	F013'		
F020R	F020	G097		
F021S	G007	G097		
F021R	F020'	F019'		
	F021	F019'	G284	
G003I	F020'	F058		
	G284			
G007I	F013	F020'		
G097I	G284			
	F020'			
G284I	F112'	F113	F114'	

H. L. ENGEL is with Ramo-Wooldridge Corporation, Los Angeles, Calif.

Table III. Usage Tabulation

Name	Where Used	Term			
F013	F019S	F013	F012′	F019′	G007
	F020S	F013	F019	G007	
	G007I	F013	F020′		
F013′	F019S	F013′	F112	F019′	
	F019R	F013′	F112	F019	
	F020S	G003	F013′		
F019	F019R	F013′	F112	F019	
		G007	F019		
	F020S	F013	F019	G007	
F019′	F019S	F013′	F112	F019′	
		F013	F112′	F019′	G007
	F021R	F020′	F019′		
		F021	F019′	G284	
F020	F020R	F020	G097		
F020′	F021R	F020′	F019′		
	G003I	F020′	F058		
	G007I	F013	F020′		
	G097I	F020′			
F021	F021R	F021	F019′	G284	
F058	G003I	F020′	F058		
F058′	F019S	F058′	G284	G097	
F112	F019S	F013′	F112	F019′	
	F019R	F013′	F112	F019	
F112′	F019S	F013	F112′	F019′	G007
	G284I	F112′	F113	F114′	
F113	G284I	F112′	F113	F114′	
F114′	G284I	F112′	F113	F114′	
G003	F019R	G003	G007	G097	
	F020S	G003	F013′		
G007	F019S	F013	F112′	F019′	G007
	F019R	G007	F019		
		G003	G007	G097	
	F020S	F013	F019	G007	
	F021S	G007	G097		
G097	F019S	F058′	G284	G097	
	F019R	G003	G007	G097	
	F020R	F020	G097		
	F021S	G007	G097		
G284	F019S	F058′	G284	G097	
	F021R	F021	F019′	G284	
	G003I	G284			
	G097I	G284			

used. Still further, they indicate which flip-flops or gates should be grouped together to reduce wiring between inserts or chassis. In cases where wire length may be critical because of signal deterioration or noise pick-up, the logical designer and production engineer can work together using these tabulations to select a physical arrangement of inserts to minimize these difficulties.

When it is necessary to modify a computer to achieve additional capabilities, the information necessary for production can easily be brought up-to-date. A few cards can be substituted in the logic and usage decks and a complete, accurate, current description of the logic and usages tabulated in a very short time. Expensive alterations of logic schematics and expensive, difficult checking of such changes are not required.

The logic and usage tabulations are also helpful to the maintenance technician, for they provide in a very convenient form a list of all portions of the computer that will be directly affected by the failure of an element and a list of all elements whose failure will affect a particular element. This means that it is unnecessary for the technician to

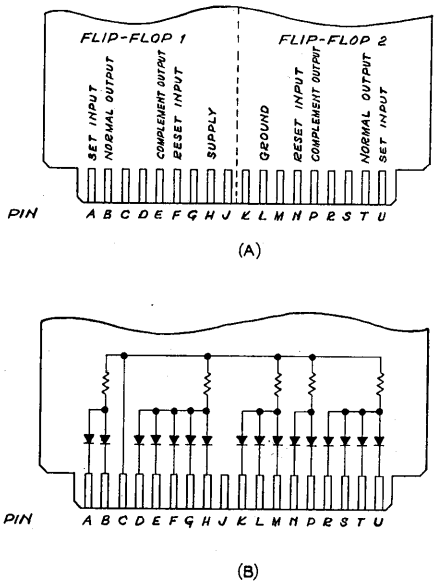


Fig. 1 (A). Flip-flop insert. (B). Gate insert

trace through many gates on large, complicated schematic diagrams to find this same information.

Wiring Tabulation

The wiring tabulation describes the interconnection of flip-flops, gates, and amplifiers in the computer, and connections to the outside world. In order to construct a wiring list, it is necessary to give each wire in the computer a name. In the example used thus far, all necessary interconnecting wires may be assigned the same name as the flip-flop or gate whose output they carry or the name of the flip-flop or gate input provided by the wire.

The usage deck cards indicate all points to which the output of a flip-flop or gate must be transmitted. It is only necessary to manufacture an additional card for each flip-flop or gate output utilized in order to have cards for all points on the wires carrying flip-flop or gate outputs. These additional cards are easily produced from the usage deck by machine methods, and are known as usage header cards. The usage header cards for the example used in this paper are tabulated in Table IV.

In like manner, and for the same reason, a logic header card corresponding to every flip-flop or gate input is generated. The logic header cards are tabulated in Table V.

The header cards, logic deck, and usage deck are collated by a machine so that all cards corresponding to the same wires are together, as indicated

Table IV. Usage Header Cards

Name
F013
F013′
F019
F019′
F020
F020′
F021
F058
F058′
F112
F112′
F113
F114′
G003
G007
G097
G284

Table V. Logic Header Cards

Name
F019S
F019R
F020S
F020R
F021S
F021R
G003I
G007I
G097I
G284I

on the left of Table VI. Having decided upon the arrangement of insert boards, it is then possible by manual or machine means to assign rack, card, and pin numbers to every signal wiring point. This is indicated on the right side of the preliminary wiring list, Table VI. In the example chosen here, the equipment is constructed in four racks, each rack having up to 20 insert cards. Each insert card may contain two flip-flops or a number of gates. The internal connections of these insert cards are indicated in Fig. 1. Note that any of the diodes in Fig. 1(B) may be used as an "or" diode; for this reason "or" gates are not tabulated separately. Instead, an "or" gate is created by wiring together the outputs of several "and" gates through these diodes. In this kind of gate it is necessary to return the "or" diode to a negative voltage through a resistor, although for purposes of simplicity this has not been included in the present example.

The rack, insert, and pin numbers are transcribed to the deck of header, logic, and usage cards. By sorting these cards the final wiring tabulation may be produced, as in Table VII. In this list the wiring starts at the lowest numbered rack, card, and pin for each wire, and the wires are arranged in order of increasing rack, card, and pin

throughout the equipment. This form is convenient for performing wiring.

For other purposes, other forms of the wiring tabulation are more useful; e.g., it may be desirable to separately list the wiring in each rack and the interconnecting wiring between racks. Such a tabulation is easily constructed by machine methods with very little effort. Another arrangement of the wiring list is also more useful for checking the wiring after it has been completed.

Table VI. Preliminary Wiring List

Name	Where Used	Term	Rack	Card	Pin
F019S	F019S		1	9	a
		F013′..F112..F019′	1	3	g
		F013..F112′..F019′..G007	1	5	h
		F058′..G284..G097	1	5	u
F019R	F019R		1	9	f
		F013′..F112..F019	1	8	g
		G007..F019	1	8	m
		G003..G007..G097	1	8	u
F020S	F020S		1	9	u
		F013..F019..G007	1	10	g
		G003..F013′	1	10	m
F020R	F020R		1	9	n
		F020..G097	1	11	m
F021S	F021S		1	13	a
		G007..G097	1	12	m
F021R	F021R		1	13	f
		F020′..F019′	1	14	m
		F021..F019′..G284	1	12	g
G003I	G003I		2	12	a
		F020′..F058	2	11	m
		G284	2	11	b
G007I	G007I		2	12	j
		F013..F020′	2	11	g
G097I	G097I		3	19	e
		G284	3	17	b
		F020′	3	17	p
G284I	G284I		4	15	e
		F112′..F113..F114′	4	17	g
F013	F013		1	4	b
	F019S	F013..F012′..F019′..G007	1	5	d
	F020S	F013..F019..G007	1	10	d
	G007I	F013..F020′	2	11	e
F013′	F013′		1	4	e
	F019S	F013′..F112..F019′	1	3	d
	F019R	F013′..F112..F019	1	8	d
	F020S	G003..F013′	1	10	l
F019	F019		1	9	b
	F019R	F013′..F112..F019	1	8	f
		G007..F019	1	8	l
	F020S	F013..F019..G007	1	10	e
F019′	F019′		1	9	e
	F019S	F013′..F112..F019′	1	3	f
		F013..F112′..F019′..G007	1	5	f
	F021R	F020′..F019′	1	14	l
		F021..F019′..G284	1	12	e
F020	F020		1	9	t
	F020R	F020..G097	1	11	k
F020′	F020′		1	9	p
	F021R	F020′..F019′	1	14	k
	G003I	F020′..F058	2	11	k
	G007I	F013..F020′	2	11	f
	G097I	F020′	3	17	n
F021	F021		1	13	b
	F021R	F021..F019′..G284	1	12	d
F058	F058		1	17	b
	G003I	F020′..F058	2	11	l
F058′	F058′		1	17	e
	F019S	F058′..G284..G097	1	5	r
F112	F112		2	6	t
	F019S	F013′..F112..F019′	1	3	e
	F019R	F013′..F112..F019	1	8	e
F112′	F112′		2	6	p
	F019S	F013..F112′..F019′..G007	1	5	e
	G284I	F112′..F113..F114′	4	17	d
F113	F113		2	7	b
	G284I	F112′..F113..F114′	4	17	e
F114′	F114′		2	7	p
	G284I	F112′..F113..F114′	4	17	f
G003	G003		2	12	b
	F019R	G003..G007..G097	1	8	r
	F020S	G003..F013′	1	10	k
G007	G007		2	12	k
	F019S	F013..F112′..F019′..G007	1	5	g
	F019R	G007..F019	1	8	k
		G003..G007..G097	1	8	s
	F020S	F013..F019..G007	1	10	f
	F021S	G007..G097	1	12	k
G097	G097		3	19	f
	F019S	F058′..G284..G097	1	5	t
	F019R	G003..G007..G097	1	8	t
	F020R	F020..G097	1	11	l
	F021S	G007..G097	1	12	l
G284	G284		4	15	f
	F019S	F058′..G284..G097	1	5	s
	F021R	F021..F019′..G284	1	12	f
	G003I	G284	2	11	a
	G097I	G284	3	17	a

Just a few man-days of work are required to generate a wiring tabulation for a computer having over 100 flip-flops. This means that it is far cheaper and faster to produce wiring tabulations than to produce wiring diagrams. Further, errors are more easily detected and corrected in the tabulations than they would be in the corresponding diagrams. It is also easier to train a production assembler to follow a wiring tabulation than to interpret a wiring diagram, and there is less likelihood of error.

The wiring tabulation completely describes the signal wiring of the computer in a few 8 $^1/_2$-by-11-inch printed pages. This puts the description of the computer in a form that is easily handled, rather than on a drawing requiring many square feet of floor or wall space to display and requiring a very considerable effort to interpret.

Another advantage of the wiring tabulation over the corresponding drawing is the ease with which the tabulation may be changed to allow modifications of the computer. A correct, up-to-date wiring tabulation can be produced on very short notice without days of drafting time to correct all drawings.

In maintenance the wiring tabulation is exceedingly helpful, for it provides the technician not only with the logic and usages but also with the locations of all flip-flops and gates. This greatly assists in trouble shooting. Another form of wiring tabulation not shown here is even more convenient for this purpose.

Logic Simulation

Use of machine language in the design of digital computers permits complete simulation of the logical structure of one computer on another computer. This results in considerable savings of time and money in the design, construction, and check-out of a new computer.

The general techniques used for simulating the logic of one computer with another computer may be described as follows. First, a master program that may be used for many different computers is written. The function of this program is to evaluate all gate outputs and all flip-flop inputs in terms of the current state of all flip-flops. Following this, all of the flip-flops are set to the state

that they would attain in the succeeding digit time and the process is repeated. This process is described for a synchronous computer but may easily be adapted to an asynchronous one.

In addition to this master program, other programs are also required. For example, there may be several types of print-out programs to show what is going on in the simulated computer. It is possible to indicate the state of every flip-flop in the simulated computer at every digit time. It is also possible to print out only the numerical results of each operation of the simulated computer.

Another program that might be written would enable the simulating computer to obtain the correct results without resorting to simulation of the logic. These results could then automatically be compared with the results obtained through simulation to determine if the simulated computer is performing as intended.

For each new computer to be simulated it is also necessary to write a program corresponding to the actual program that the simulated computer is to perform. That is, if the actual computer is to perform a sequence of additions, multiplications, divisions, transfers, etc., a program consisting of these commands must be inserted in the simulating computer. Considerable ingenuity is required in the design of this program so that the operations and operands will exercise all critical conditions in the simulated computer. These conditions might include additions and subtractions where all combinations of the signs of the operands and results occur, operations where overflow occurs, division by zero, etc. Ingenuity is required because time limitations make it absolutely impossible to check all combinations of operands and commands. The time required to check all operands and commands may well exceed the lifetime of the computer.

Once these programs have been written, the logic of the computer may be entered into the simulating computer at high speed from the logic deck, and the action of the computer simulated in less than an hour. If there are errors in the logical design of the computer, it is easy to locate these by means of the print-outs previously described. It is then a simple matter to correct the logic, produce a new logic deck by removing a few cards and adding new cards, and then to simulate the corrected computer. This process may be repeated several times if necessary.

Machine simulation of the logical design of a computer is useful to the logical designer in several ways. First, it checks the accuracy of the logical design. It enables the logical designer to determine that his design is both

Table VII. Wiring Tabulation

Name	Where Used	Term	Rack	Card	Pin
F013′	F019S	F013′ . F112 . F019′	1	3	d
	F013′		1	4	e
	F019R	F013′ . F112 . F019	1	8	d
	F020S	G003 . F013′	1	10	l
F112	F019S	F013′ . F112 . F019′	1	3	e
	F019R	F013′ . F112 . F019	1	8	e
	F112		2	6	t
F019′	F019S	F013′ . F112 . F019′	1	3	f
		F013 . F112′ . F019′ . G007	1	5	f
	F019′		1	9	e
	F021R	F021 . F019′ . G284	1	12	e
		F020′ . F019′	1	14	l
F019S		F013′ . F112 . F019′	1	3	g
		F013 . F112′ . F019′ . G007	1	5	h
		F058′ . G284 . G097	1	5	u
	F019S		1	9	a
F013	F013		1	4	b
	F019S	F013 . F012′ . F019′ . G007	1	5	d
	F020S	F013 . F019 . G007	1	10	d
	G007I	F013 . F020′	2	11	e
F112′	F109S	F013 . F112′ . F019′ . G007	1	5	e
	F112′		2	6	p
	G284I	F112′ . F113 . F114′	4	17	d
G007	F019S	F013 . F112′ . F019′ . G007	1	5	g
	F019R	G007 . F019	1	8	k
		G003 . G007 . G097	1	8	s
	F020S	F013 . F019 . G007	1	10	f
	F021S	G007 . G097	1	12	k
	G007		2	12	k
F058′	F019S	F058′ . G284 . G097	1	5	r
	F058′		1	17	e
G284	F019S	F058′ . G284 . G097	1	5	s
	F021R	F021 . F019′ . G284	1	12	f
	G003I	G284	2	11	a
	G097I	G284	3	17	a
	G284		4	15	f
G097	F019S	F058′ . G284 . G097	1	5	t
	F019R	G003 . G007 . G097	1	8	t
	F020R	F020 . G097	1	11	l
	F021S	G007 . G097	1	12	l
	G097		3	19	f
F019	F019R	F013′ . F112 . F019	1	8	f
		G007 . F019	1	8	l
	F019		1	9	b
	F020S	F013 . F019 . G007	1	10	e
F019R		F013′ . F112 . F019	1	8	g
		G007 . F019	1	8	m
		G003 . G007 . G097	1	8	u
	F019R		1	9	f
G003	F019R	G003 . G007 . G097	1	8	r
	F020S	G003 . F013′	1	10	k
	G003		2	12	b
F020R	F020R		1	9	n
		F020 . G097	1	11	m
F020′	F020′		1	9	p
	F021R	F020′ . F019′	1	14	k
	G003I	F013 . F020′	2	11	f
	G007I	F020′ . F058	2	11	k
	G097I	F020′	3	17	n
F020	F020		1	9	t
	F020R	F020 . G097	1	11	k
F020S	F020S		1	9	u
		F013 . F019 . G007	1	10	g
		G003 . F013′	1	10	m
F021	F021R	F021 . F019′ . G284	1	12	d
	F021		1	13	b
F021R		F021 . F019′ . G284	1	12	g
	F021R		1	13	f
		F020′ . F019′	1	14	m
F021S		G007 . G097	1	12	m
	F021S		1	13	a
F058	F058		1	17	b
	G003I	F020′ . F058	2	11	l
F113	F113		2	7	b
	G284I	F112′ . F113 . F114′	4	17	e
F114′	F114′		2	7	p
	G284I	F112′ . F113 . F114′	4	17	f
G003I		G284	2	11	b
		F020′ . F058	2	11	m
	G003I		2	12	a
G007I		F013 . F020′	2	11	g
	G007I		2	12	j
G097I		G284	3	17	b
		F020′	3	17	p
	G097I		3	19	e
G284I	G284I		4	15	e
		F112′ . F113 . F114′	4	17	g

complete and error-free. It assures that no combination of conditions has been overlooked, and that if the machine is constructed according to the logical design indicated, any malfunctions will not be due to mistakes in the logic. A second advantage of machine simulation to the logical designer is that it makes alterations in the design of the computer very simple. If errors have been found in the initial logical design through the simulation process, then the logical designer can make corrections and rerun the simulation to determine if these corrections are adequate. More important, however, is that if model changes are to be made in the computer, the logical designer can make the indicated changes in the design, quickly produce a new logic deck, and quickly determine by means of a simulation run that these modifications are correct and do not interfere with the previous operation of the computer.

Simulation is also useful to the development engineer. First, the span of time between inception of computer design, completion, and check-out of the first computer is reduced since any malfunctions can be relegated to circuits or wiring errors, eliminating extensive checking of both circuits and logic to find the fault. Second, knowledge that the logic is correct permits production design to commence earlier, since it is not necessary to wait until the prototype is checked out before doing the design required for production computers.

Simulation of the logic of the computer can also be of assistance to the maintenance technician. Not only can the logic of a properly functioning computer be simulated, but also the behavior of a computer with a faulty diode or flip-flop can be simulated. Thus, lists can be made describing the behavior of the computer according to the type of failure that has occurred. These lists can greatly reduce time required in locating and correcting troubles.

Future Uses

Still further advantages of machine language in the design of digital computers can be expected in the future. It is likely that computers will be programmed to find Boolean expressions that will require the smallest number of components to construct. Further, there is hope that computers will be programmed to derive logical designs with the minimum total component count that can perform a given job. This point requires further discussion. At present the technique used in designing digital computers is to assign a function or set of functions to each flip-flop and then to find the simplest gating that will enable the flip-flops of a computer to perform their assigned functions. These techniques permit some savings in the total equipment required in a computer, but there is room for further savings; if the functions of the flip-flops are not preassigned it is possible to find uses for the flip-flops that will result in a computer with fewer parts. At present, labor costs do not allow this. Furthermore, when this is done it will not be possible to pick a particular flip-flop and give it a name such as "carry flip-flop." Instead, because of the multiple time sharing of whole groups of flip-flops, physical meanings will not be attached to flip-flops. This in turn will make servicing more difficult, although it will reduce the number of failures that occur in the computer. For this reason it will be necessary to use machine aids in maintaining a computer. These machine aids may include tables of malfunctions produced by simulation on another computer.

Conclusions

It is evident from the foregoing that machine language is an extremely useful tool in both the design and production of digital computers. Although the examples given are based on no existing machine, the techniques described have been in use for several years and have been employed in the design of several computers, enabling considerable telescoping of design and production and have greatly reduced the time required to check out new computers.

Logical simulation in one instance was used to check the logical design of a computer, resulting in the discovery of two extremely sophisticated logical design errors that probably would not have been encountered in several months of actual computer operation. This particular computer, when wired as determined by the techniques described in this paper, had absolutely no logical design errors. This computer contained over 130 flip-flops.

Reference

1. The Advantages of Logical Equation Techniques in Designing Digital Computers, **V. L. Hesse.** *Proceedings of the Western Joint Computer Conference*, 1958.

The Advantages of Logical Equation Techniques in Designing Digital Computers

VICTOR L. HESSE
NONMEMBER AIEE

IN MEASURING the value of any system or process, it is necessary to determine how the process compares with other methods not only in terms of economy, but also in terms of time, over-all efficiency in accomplishing its purpose, and flexibility in handling variations in the basic problem. It is the purpose of this paper to point out some of the many advantages inherent in the logical equation method of digital computer design. Consistent with most problems, there are many approaches which can be used to reach the desired goal, equation technique is the most efficient in every case. One can always find examples and cases where some other technique is more efficient or more simple to execute. It is, however, the opinion of the author that the application of the logical equation technique to a general design problem usually produces a more nearly optimum design, and a final result is usually achieved more quickly and more efficiently than is possible with the block diagram technique.

The design of a digital computer has several ultimate objectives. Among these are simplicity, high reliability, low cost, flexibility, and ease of maintenance. Certainly the most direct step toward reliability, low cost, and ease of maintenance is the reduction of the total amount of equipment required to accomplish the

Victor L. Hesse is with Litton Industries, Beverly Hills, Calif.

purpose of the machine. Thus a method of design that allows an economy in equipment obviously must be of great value.

Since the logical design deals directly with the formulation of the gating networks, it is natural to look for economy in this design area first. The logical equation method is particularly adapted to the use of formalized reduction schemes. The Quine method, the Veitch diagram, and the laws of Boolean algebra are examples of aids that can be used in reducing a logical expression after it has been formulated.[1] It has been demonstrated that these processes have the great advantage of providing an analytical procedure that guarantees a minimal solution without depending upon the designer's skill or experience. In addition to the straight reduction of logical expressions, these methods also allow for the nonexistence of some of the logical states. By allowing for these nonexistent states, it is possible to reduce a logical expression further and thereby save equipment as well as simplify the machine. For instance, in the expression $AB'C+ACD+BCD+B'C'D'$ a restriction is possible that prevents the CD' condition from ever occurring. In that case, the expression would reduce to $AC+BC+B'D'$. The attendant saving is obvious.

Many computer engineers have the mistaken impression that the primary value in logical equations is to be found in the purely logical reduction of a proposition in the manner previously described. It should be noted, however, that there are two very distinct aspects or phases to logical design. These the author prefers to call strategic design phase and tactical design phase. Tactical design is concerned primarily with the determination of final form of an expression once the function and purpose of the expression has been ascertained. Thus, the purely logical reduction falls into the tactical design phase. Of much greater importance, however, is the strategic design phase, which determines the function of each building block, the configuration of signal flow paths, and the overall pattern for machine structure.

It is in the strategic design that the logical equations really prove their value, for it is in this phase that the designer fixes the number of each kind of building block; flip-flops, inverters, amplifiers, converters, and drivers. With logical equations set down in rough form for each of the blocks, it soon becomes obvious where weaknesses in the design are causing a buildup in the equipment, thus indicating where further effort should be directed to improve the design. The number of flip-flops in a machine is a fair indication of its potential reliability as well as its cost. With the logical equation techniques, it is a simple matter to determine suitable methods to time share a single flip-flop to accommodate functions that might otherwise require a greater number of flip-flops.

Table I

E	D	C	B	A	
0	0	0	0	0	
0	0	0	0	1	
0	0	0	1	0	
0	0	1	0	1	
0	1	0	1	0	
1	0	1	0	1	$SA=E'$
0	1	0	1	1	$RA=E'$
1	0	1	1	0	$SB=A$
0	1	1	0	0	$RB=A'$
1	1	0	0	1	$SC=B$
1	0	0	1	1	$RC=B'$
0	0	1	1	1	$SD=C$
0	1	1	1	0	$RD=C'$
1	1	1	0	1	$SE=D$
1	1	0	1	1	$RE=D'$
1	0	1	1	1	
0	1	1	1	1	
1	1	1	1	0	
1	1	1	0	0	
1	1	0	0	0	
1	0	0	0	0	
0	0	0	0	0	

One computer that was designed with logical equations made particularly efficient use of four flip-flops in a multiple time sharing configuration. These four flip-flops accomplished the prime functions of sign and carry in a full adder-subtractor, a half adder-subtractor, and two half adders, each of which used a different number system. As auxiliary functions, these flip-flops converted a number from one number system into another for use in one of the adders, removed switch bounce transients during front-panel fill operations, provided a path for signal flow to and from external equipment such as plotters, graph followers, and digital-to-analog converters, inserted a one bit delay to cause a precession in a short register, provided a path for intercommunication between two parts of the memory, and provided signals to indicated idle or computing modes of operation. All of these functions were accomplished by the simple use of only 184 diodes in the gates driving the four flip-flops.

Ultimately the time sharing philosophy would result in what is sometimes called an Optimum Computer. It can be shown that any particular problem has only a finite number of possible values of parameters and variables, and therefore it follows that only a finite number of solutions is possible. An optimum computer may be defined as one in which there are only as many possible states as there are possible solutions. This implies that no state of the computer is wasted. Since the problem to be solved defines the logical state of the optimum computer, any state of the computer that does not represent a solution, implies an excess of information that makes the computer suboptimal. In the present state of the art there are no known computers that even approach optimum conditions. However, the art is gradually advancing toward that goal through continued application and use of logical equation techniques.

As a direct result of the reduction of the machine logic, there are also other savings not to be overlooked. Already mentioned are the minimization of the number of diodes, gates, and flip-flops which results in a lower over-all component count. Furthermore, the number of connections is reduced, the number and length of wires is reduced and the loading on the various circuits in the machine is reduced.

In addition to minimizing the amount of equipment, the logical equation technique tends to improve the efficiency of the logical designer. Not only does it allow the designer to accomplish his task faster and easier, but it also grants to him advantages that block diagram techniques can not hope to offer. For instance, many problems in logical formulation have more than one solution, and frequently two or three forms can be considered of equal value. These alternate mechanizations are easily found with logical equation techniques so the designer can aptly choose from them on the basis of other criteria such as circuit loading, use of terms already available in another part of the machine, or repetition of gate form for use in repeated modules.

An example of a logical proposition that has two equivalent forms is $AB'C+ABD+A'BC'+A'B'D'$

An alternative form is $ACD+BC'D+A'C'D'+B'CD'$ These two expressions have no common terms, yet they are logically identical. Also the two expressions take identical amounts of equipment to mechanize. If the example was modified by declaring that the states $A'BCD$ and $AB'C'D'$ could never exist, it would be found that the resulting proposition has four equivalent forms. They are:

$BD+B'D'+AB'C+A'BC'$

$BD+B'D'+AB'C+A'C'D'$

$BD+B'D'+ACD+A'BC'$

$BD+B'D'+ACD+A'C'D'$

It is undoubtedly true that the block diagram technique would yield one or more of these minimal forms, but whether or not it would yield all of the forms would be dependent upon the skill and experience of the designer. With equation techniques; a purely analytical method would produce all of the forms, and the designer would be certain that there would be no other equivalent forms that he might overlook.

Calculation of circuit loads becomes a simple operation when the machine is defined in terms of logical equations. In fact, the job may be programmed on a computer very easily.[2] Equalization of loads on various flip-flops is also possible by re-arranging functions among the time shared flip-flops.

Other benefits of logical equations are obtained from the ability to make use of intentional redundancy. This type of approach will assist in improving reliability and at the same time will reduce the component count. A logical designer is usually confronted with the problem of insuring that no state in the computer can exist to the extent that the computer can not be cleared of that state. From the equations, such "hang-up" conditions can be readily spotted. Hang-up conditions are not always static. Sometimes an unallowed cycle will be possible, as in a 5-stage shifting counter. A computer might normally use such a counter to cycle through 21 steps using the sequence in Table I. The equations for the five flip-flops are included.

When the machine is first turned on, however, it is possible for the five flip-flops to assume the all 1's state, which is a static hang-up condition, or one of the states in Table II, which represent unallowed cycles or dynamic hang-up conditions.

Even after the computer is built the logical equations prove quite useful. In terms of sheer volume alone they are more easily dealt with than the block diagram. It is possible to separate equations to represent the various functions of the computing blocks and then to study them independently without

Table II

E	D	C	B	A	E	D	C	B	A
0	0	1	0	0	0	1	0	0	0
0	1	0	0	1	1	0	0	0	1
1	0	0	1	0	0	0	0	1	1
0	0	1	0	0	0	0	1	1	0
					0	1	1	0	1
					1	1	0	1	0
					1	0	1	0	0
					0	1	0	0	0

other functions confusing the issue. This factor allows much clearer explanations and easier understanding of the machine operation. This is very useful to a person just learning about a machine or to someone who is repairing it. In fact, two or three pages in a notebook can easily represent as much of the machine as several square yards of block diagrams.

Take for example a write flip-flop (W) for a circulating register that is acting in three different ways at three mutually exclusive times. At time $T1$, it is required to take the sum of the contents of the read flip-flop (R) and the contents of another flip-flop (A) using a carry flip-flop (C). At time $T2$ it is required to take the complement of the contents of A using C as the carry in a half-adder configuration to give the 2's complement rather than the 1's complement. At time $T3$, it is required to recirculate the contents of the register by reading R.

Assume that the computer circuits will allow the use of third-level logic, but will not allow the use of fourth level. The straightforward, unreduced equation for the set input of W that represents these actions is:

$$SW = T1(AR'C' + A'RC' + A'R'C + ARC) + T2(AC' + A'C) + T3R$$

or holding to the third level restriction:

$$SW = T1(A + R' + C')(A' + R + C')\times (A' + R' + C)(A + R + C) + T2(A + C)(A' + C') + T3R$$

The equations are divided up by the timing signals to indicate exactly what is being done at any one time. However a reduced form of the equation that might be mechanized is:

$$SW + (A + C)(A' + C')(T1 + T2) + T1R(A + C')(A' + C) + T3R$$

This form might be used because of its comparatively light load on the R flip-flop.

It is helpful to the reader when describing the operation of the machine in a manual to include two of the previous equations. The first (or second) provides the reader with an easily understandable form that isolates the time shared functions, whereas the third give the actual form of the mechanization.

Furthermore, to a person learning about a new machine, the value of readily recognizing new formulations is not to be overlooked, and rearrangment of the logic as written into a format with which the person is familiar is quite simple.

Conclusions

Thus, it has been seen that the concept of logical equations possesses distinct advantages in contrast to the use of block diagrams in resolving a digital computer design problem. In the strategic design phase, it is evident that logical equations can dictate improvements, acting as a feedback to the designer and helping him to use his equipment more efficiently. In the tactical design phase, the ability to make use of purely analytical techniques assures the optimum form for an expression. Although the understanding of logical equations requires somewhat more education and experience than the understanding of block diagrams, it can be seen that equations still have some advantages for maintenance personnel. Both systems have their place, and a good designer can make use of both as his particular requirements dictate.

References

1. The Logical Design of Digital Computing Machinery, **M. Phister, Jr.**, *Notes*.

2. Machine Language in Digital Computer Design, **H. L. Engel**. *Proceedings of the Western Joint Computer Conference*, 1958.

Discussion

Lowell S. Bensky: My analogy was Heaviside's operational calculus that I think can come up again, as it seems that it can be twisted several times. I think that most of us have had occasion to use Laplace transforms rather blindly, and get into trouble because we forget to take account of the location or number of poles, etc. I would say that in logic equations you may run into the same problems, or a similar one, that of putting your problems largely in the hands of a rather mechanical device. It certainly makes the design process apparently more efficient but I wonder whether we do not overlook logical possibilities in so doing.

Another point that I would like to make: Complete checking of a logic design is a question that has been beaten I believe, into the ground. It is my contention that when computer control is a function, not only of timing, but also of the characteristics of data, for example, in variable word length machines, all of the possible combinations of conditions must be enumerated for a complete check. I am not sure that such an enumeration is possible without a block diagram in a complete machine.

Another point in the area of maintenance: I would concede that you may be able to list some typical faults and their conditions but I think that you would have a considerably tougher problem either in using equations, or in listing conditions when you run into the problem of multiple defects such as usually occur when you have a transient that may effect more than one diode, or more than one vacuum tube in different parts of the machine. The conditions that exist after such a malfunction, or a group of malfunctions, I believe, are more easily comprehended from a diagram than from a set of pages of logical equations.

Finally, I think that we must be careful to blame our design procedure, and not put too much emphasis on the cost or speed of the actual logic design when, actually, they only represent a very minor percentage of the cost, or the time of propagation of the over-all computer.

Mr. Victor L. Hesse: I think that most of the arguments have been presented on on both sides, however, I would like to reiterate several here. Mr. Bensky made reference to the block diagram, saying that it is easier to visualize, and therefore prevents one from getting confused, but I think that this is probably due to the fact that he may not be as familiar with logical equations as some of the people here on the West Coast. Visualization of a logical equation and what it actually represents is not difficult. I think that I have already admitted the point on education; it does take a little education to learn logical equations, but once that education has been surpassed, then the technician, the designer, etc., who are concerned with the computer, can visualize just as easily, if not more easily, what is going on in the computer by looking at the equations.

He also mentions the thought that one would get bogged-down in location or in procedures, etc., with logical equations. I think that here the volume of data is enough to slow down a person with block diagrams.

Mr. Richards placed quite a bit of emphasis on parallel operations, and the ability of block diagrams to come up with so-called optimal solutions with parallel systems. From the computers with which I have come in contact, and have seen, I find that practically all parallel computers are of East Coast derivation; and practically all West Coast computers tend to be serial in nature. This might be quite significant but I think that the most significant part of it is the fact that the West Coast, with their logical equation techniques, has been able to solve the problems which are equivalent in complexity to those that have been solved on the East Coast. It has not had to resort to the large amount of equipment that is inherant with parallel-type operations.

R. K. Richards: I have a number of small points here that I would like to make; one of them had to do with loading. Both speakers mentioned the loading, and how easy it was to handle this problem with logical equations. Well, this expression is somewhat unusual because I always considered loading no problem at all; with respect to block diagrams it was obvious.

It was also mentioned by one of the speakers that it is difficult to use block diagrams for different types of inputs to a flip-flop. Well, again, I have in mind to point out that you can have several kinds of notations on your block, and the different lines of the block, and different kinds of positions and arrowheads indicate the nature of the input signal. It is quite simple.

Another point, which is more comprehensive, is the subject of wiring tabulations. This was dealt with at some length by Dr. Engel, I believe. Wiring tabulations do not seem to me to be an exclusive property of logical equations. There are some ways of getting at a wiring tabulation that may have applications in some cases, but you can make a wiring tabulation of exactly the same type from a block diagram as you can with logical equations. It seems to me that it would be wiring tabulations versus wiring diagrams. Wiring diagram approach is quite unrelated to the subject of Boolean algebra versus block diagram.

Another thing that should be considered, of course, is, as Dr. Engel pointed out, cost, but comparing the cost of the two methods is like comparing the cost of a vacation here at the Ambassador Hotel and a vacation fishing in the woods. On one hand the hotel bill is greater, and on the other hand the cost of travelling to get to the woods is greater. So you have to consider the whole thing, and this is difficult to do. The proof is in the pudding, you might say, as when some of the computers which are widely used, like the IBM's *700* machines and Sperry-Rand's Univac-type computers, are replaced by newer models which use only half the components, and where the reduction is obtained only through the use of logical equations instead of block diagrams. Somehow this competition with these existing computers, and others I might mention, does not seem to be coming from this quarter, it comes from other areas, faster tapes, faster transistors, etc.

Another thing I would like to mention as the last point, and which is somewhat related to this serial-parallel business, is that the relative merits of logical equations and block diagrams is somewhat dependent upon the nature of the computer and the application of the computer. If you are going to make only one of a model and you have to do it in 11 months, then you put in whatever effort is necessary to get the job done in 11 months to meet your deadline. And the considerations there are quite different from what you do sometimes if you plan to sell a 100 or a 1,000 of these, and if you spend only 11 months designing it, you would have an inefficient design. That has to be considered.

Also, it must be considered whether the machine is complex or simple. Some of the business machines which you see are complex. If you considered all the features that must go in to put the commas and periods in the right places, advance the paper and put headings on accounting forms, and get all the holes in the right places, you have a complex computer. And logical equations for a thorough job on that are quite difficult to use. Now, to follow the complexity that Mr. Hesse pointed out and used slides for, there is no question that you like logical equations to simplify problems of that complexity. And if you have a computer which is a serial-binary machine, you can get a little farther with logical equations than you can if its a more complex machine that has to do a complex function. A serial-binary machine for full-control applications does not compare in complexity with some of these others.

With regard to the serial machines of the West Coast to the parallel machines of the East Coast, they are still building and selling many parallel machines on the West Coast. I mean they are building them in the East and selling them in the West.

H. L. Engel: I have a couple of remarks about Mr. Bensky's presentation first. I will agree with him on several things. First I agree that logical design is an art. If you try to design a control unit for a computer, you discover that you make up your own general rules as you go along. Second, I will agree with him that all of the functions that he mentioned for a logical designer working in conjunction with systems engineers, programmers, manufacturing people, analysts, etc., are the proper functions of the logical designer.

Some of the speakers have said that logical equations are too hard to understand, to comprehend. On the other hand, think of trying to comprehend a block diagram of a large computer where just the square footage and the number of wires shown on it, makes it practically impossible to trace through all of the connections in a particular circuit.

I think that Mr. Bensky has confused a little bit the strategic and tactical forces of logical design, according to the definition. Mr. Hesse agrees that in the strategic forces it is foolish to try to use a notation of Boolean algebra.

Even in the strategic forces there are applications where notations of Boolean algebra will help you. For example, if you wanted to design a parallel adder, and

you had certain restrictions in your gatings, if you did not use the slope of the output in your flip-flop and instead of having a more reliable form of gating, you might be restricted in the number of level gatings that are permitted without exceeding the delay which is permitted to you. And here it is very convenient to write the expressions for the carry that must exist, using the notation of Boolean algebra. And if you are clever about using this, you can find that there are ways in reducing the number of levels of gating through which your carry signal has passed.

I was recently looking at the problem of a parallel adder and I determined that the total, the maximum number of levels that a carry signal would have to pass through was only nine. And this was done by proper manipulation of the logical equation.

Another point on which both of the speakers on the opposing side and I disagree is the application of the notation of Boolean algebra to gating systems other than direct, pulse traveling through. For example, a scale of 16 counters was mentioned where the slope of the change was used to trigger the next stage. This may also be described very simply by a notation of Boolean algebra, if you properly select the quantity described in the flip-flop. You use the change in the output as the description of the output in that signal rather than the state of the flip-flop and the description of the output signal.

I ask the rhetorical question, "How do you check the logic of a computer according to your block diagram form?" You must exhaustively go through truth tables or you must build a computer to see what happens.

I agree with Mr. Bensky that complete checking of the logical computer is not possible. For a complete check one has to go through all of the possible stages of the computer. If you have a 100-flip-flop computer with a two to the one hundred states, it would take you more than a lifetime to compute it into this check. Instead you must use some intelligence and a knowledge of what the critical conditions would be.

Another remark that was made was that use of logical design with Boolean algebra has not halved the size of computers. Well, it does not halve to have the size of the computers, it pays for itself. The cost of logical design in computers I have been associated with is less than 10% of the cost of the first computer. However, the big saving is in the cost of that first computer and subsequent computers. If you can reduce the number of components by 20% or even 10%, you have made a very substantial saving.

I have to attack my colleagues now. Mr. Hesse stated that smaller equipment is easier to understand. I think that I cannot defend this statement. We have made smaller equipment by using time sharing and this makes the equipment harder to understand, rather than easier. On the other hand, we have provided the maintenance man with a tool of logical description in terms of Boolean algebra with which to overcome this handicap.

Dr. Richards seems to imply that elegance and practicality are exclusive cohorts. I think that you are more likely to find that they are an end function.

Dr. Richards also made some remarks about Mr. Phister's using block diagrams in his book. If you wait a few months he can make a similar remark about a chapter that I am writing for a book to be published. However, I would like to rebutt this at the present time—in advance. The purpose of the book that I am writing, is that it is to be used as a handbook and is to be useful to logical designers whether they are trained in block diagram or logical equations.

Dr. Richards also described a digital computer as being two-dimensional. Well, I do not agree with him. It is not two-dimensional; it is not three-dimensional; it is four-dimensional. You have three dimensions in space, where you have to run your wiring and various components, and you also have time. So whether the computer is synchronous or asynchronous depends upon what happens next, and the notation of Boolean algebra has proved useful in the applications that I have seen for describing this time variant of the computer.

Another point that Dr. Richards stated was that the computer seemed to work well, and the loading had been obvious from the block diagram. I think that this is because of the fact that he worked with simple computers, simple in the sense that there is not extensive time sharing of units.

Dr. Richards also pointed out that you can obtain a wiring tabulation from a block diagram. I agree, but I ask the same question that everyone else asks.

Dr. Richards also pointed out that competition in the field of large-scale commercial and scientific computers has been more in the form of higher speed and higher speed elements, rather than from the reduction of the number of elements employing Boolean algebra in design. I think that this is a true statement, but I do not think that the truth of this statement is because block design takes precedence. It takes a certain amount of capital to get into the field of large-scale, scientific and commercial computers, and also you have to block those already established.

Another remark made was that this particular computer was not a one-of-a-kind item, it is the first of a number of items of the same or similar nature to be sold commercially.

Chairman Amdahl: The first of the questions is addressed to us by W. A. Clark, MIT: "Gentlemen: In view of the remarks made by the various speakers regarding areas of applicability, would it be unfair to rename the two schools the large, expensive machine school, and small, cheap machine school, rather than Eastern and Western respectively?"

R. K. Richards: It might be better to say the school of machines that do a complex job and the school of machines that do a simple job.

Chairman Amdahl: I have a suspicion that we want a conclusion to this subject, so we will proceed to the next question for Dr. Engel. R. Seidler, Northrup asks, "How does a wire tabulation inform a wiring technician as to wire routing?"

H. L. Engel: The items on the wiring tabulation corresponding to any one particular wire are listed in order in which they are to be wired-in. If there are several alternate routes that might be taken at varying lengths (we have not encountered this as yet) some other information would have to be added. Generally we have not had to worry about this point; the wiring technician wires directly, or goes out to either edge of the rack, and does not care very much which, then comes back in. We have not had to face that problem as yet.

Chairman Amdahl: We now have a question directed to Mr. Hesse from Samuel Litwin, National Cash Register Company: "Does optimization in terms of least number of states imply either minimum cost or maximum reliability? Or, is this criterion of optimization merely elegant?"

V. L. Hesse: I do not think that the definition of optimal computer that I used was meant to imply anything in regard to maximum reliability or minimum equipment. However, it does represent a little of both. The maximum reliability is more a result of the minimization of equipment. Actually the optimal computer would not be minimum equipment. If you were to plot a graph of the degree of optimization, according to the definition that I gave, versus the amount of equipment, you would probably find that there would be a minimum point, and it would not be at the optimum, under my definition. It would probably become quite large in equipment under the optimum condition because the optimum computer limit would be a single flip-flop, and obviously the gating involved in a single flip-flop itself is a complex problem and would be rather large. However, if in the present state of the art we are trying to optimize the computer in certain directions, we find the slope of the curve is definitely in the direction of less equipment. Therefore, I cannot say that optimization does it by less equipment, it does it by more reliability.

Chairman Amdahl: Now I have a question directed to Dr. Engel from W. N. Carroll, IBM: "Can Boolean algebra handle the possibility of feedback in the logical system?"

H. L. Engel: If you mean unwanted electrical feedback or parasitics, I do not think that it can handle this problem any better than the block diagrams can. If you mean intentional feedback, the output of one circuit has the input of the prior circuit or input of the same circuit and we have done this using Boolean expressions.

Chairman Amdahl: I have a question addressed to Mr. Hesse from F. F. Backman, IBM: "Do Western designers consider as optimum anything other than a logical expression containing a minimum number of literals. For example, of optimum timing can result from a slightly suboptimal expression, can a Westerner detect this fact and compromise?"

V. L. Hesse: I certainly think that any good designer will have many criteria in mind when he designs a computer. If he were to restrict himself by designing only in

terms of minimum expression in any given equation, I am sure that he would come out with a very suboptimal computer. So, as far as the West Coast variety is concerned, I am sure that we have a lot of good designers out here, and they are indeed considering the other factors.

Chairman Amdahl: We have another question from F. W. Springer, Hughes Aircraft Corporation for proponents of the block diagram: "Does anyone admit that in many cases a block diagram is very useful as a starting point prior to actual writing of equations?"

L. S. Bensky: I think that is the point where one should stop using the block diagram. As Dr. Hesse pointed out, there are two parts to the logical design: there is the strategic portion and the detail logical design.

Chairman Amdahl: Mr. Springer also asks, "Do you admit that a block diagram has some use in the design of a digital computer. How is it that when we design a computer in 11 months or 11 years, a block diagram is never used?"

L. S. Bensky: Well, here a block diagram is used as far as it is necessary to go in this particular computer. Other computers have gone further in the block diagram, have gone perhaps one stage in the arithmetic, where each of three registers are described, it is not necessary to go beyond that.

Chairman Amdahl: Here is a question directed to Mr. Bensky from H. L. Moody, Hughes Aircraft Corporation: "Do you include all gates on the block diagram, and are signals from one point to another represented by a complete line between both points?"

L. S. Bensky: I would say that the answer is not always yes. For instance, in gating between two registers, each set of y or $-y$ would not show 20 gates or 7 gates, it would show one gate.

As far as lines are concerned, it will not show a complete line, but it will indicate most cuts in the line. Where the other end line will be found. I would say, you do not always indicate all gates in both cases.

Chairman Amdahl: The next question is directed to Dr. Engel. It is from Ronald Compton, Librascope. "Is the logical equation method as optimum for transistor logic as it is for diode logic, where the Boolean operations have exact representations?"

H. L. Engel: I was afraid a question like that would come up. In gating systems, where you can have a number of gates coming together producing a number of wires, these being formed into other gates and coming out again, there is considerable sharing of the gating. There are difficulties in both methods of determining what is going on. If the gating is not too complicated, it can be shown pictorially, in a fairly clear manner. It can also be handled using techniques of Boolean algebra, but I think, as this man who asked the question, it is more difficult at the present time. I think that, however, Boolean algebra notations holds more promise for the future in handling gating problems of this kind than the block diagram techniques.

Chairman Amdahl: The next question is directed to Dr. Richards. It is from Paul Waller, Librascope. "How do you simplify logical expressions using block-diagram techniques?"

R. K. Richards: The answer to that question pertains to the point that, I would not attempt to do everything by block diagrams. I have stated all along that there are portions of the machine where you will want logical equations. For simplifying the machine as a whole, the problem is not always a matter of working with the physiological functions, it is a matter of working with the large blocks of the machine, such as the index registers, instruction counters, and special features. This is where you really get your gross improvement in the machine by binding ingenious over-all arrangements of the large box, not working with the individual box. This was one of Mr. Bensky's main points as I understand it.

For the restricted parts of the machine, you will want logical equations, but when you get to the really important parts of the machine, such as common index registers, etc., you get an effective machine.

Chairman Amdahl: The next question is directed to Dr. Engel. It is from B. Leitner, Lockheed. "How expensive are computer programs for simulation, etc., in terms of programming effort and machine time?"

H. E. Engel: It took us, I believe, 8 man-months to write a general-computer simulation program which could be applied to many families of computers. In addition, for each computer you must write a program that says what operation could that computer want to simulate; it is like writing an actual program for that computer.

Also, if new types of elements are introduced, you must modify your original program. It may take several weeks of effort, perhaps 2 or 3 man-weeks, in order to do this for an entirely new type of computer. Thereafter, it takes perhaps an hour to run through the program. The duration of this time depends on how thoroughly you want to go through the simulation.

In one other machine checked this way, we were working at roughly one one hundreth sixty thousandth of real time; in other words, using a large-scale digital computer took us one second to represent one digit time.

Chairman Amdahl: The next question is directed to Dr. Richards and/or Mr. Bensky. It is from Roy Keir, Bendix Computer. "What future developments in analytical treatment of block diagrams can be foreseen?"

L. S. Bensky: I do not think you will find much in the way of high-powered mathematics or analytical techniques in block diagrams; but the trend is more and more towards block diagrams.

Chairman Amdahl: The next two questions are directed to Dr. Richards. They are from Jay Wager, Hughes Aircraft: "In your example of the Mod *16* counter, how do block diagrams instruct the circuit layout man to wire the various input lines of a flip-flop?"

R. K. Richards: There are many different ways that one can do that and of course it depends upon the nature of the application of the counter. But the customary procedure would be to mark in one quarter of each block the wiring terminals to be used in the frames. As an intermediate step you could make up a wiring tabulation arrangement, and the actual assembly. In this way, you can then see the block diagram.

Chairman Amdahl: The second question from Jay Wager is: "How do block diagrams work with 'Nor' circuitry?"

R. K. Richards: There are many different kinds of circuits, and this is one of my favorite points. This question is related to another question which was asked Dr. Engel, that is: Some types of circuits work logically, but there can very well be a logical equation wherein sometimes they do not; also, in the case of a block diagram sometimes they work out, and sometimes they do not. But in the case of the "Nor" circuit there are a variety of ways to go about doing it. One way is by putting a little square marked "O," for "or," and subsequently a little square marked "I," for Inverter; or by marking an arm off. It is understood that at each one block there is an inversion in addition to the "or" function.

It has been interesting that it has been said at times that the logical blocks are easier to work with than other types, regardless of the notations that are used. In my opinion, that is one of the more difficult types.

Chairman Amdahl: The next question is directed to both Dr. Engel and Dr. Richards and is from Mr. W. H. Sturm, University of California, Los Angeles: "Having decided nothing else by our discussion, can we agree that elegance and practicality represent an inclusive or function?"

R. K. Richards: I believe that I have been slightly misquoted. I say that "elegance" is not necessarily the most practical solution.

Chairman Amdahl: I have another question for Dr. Richards from Mr. E. Younger, Bell Telephone Laboratories: "Please comment on the use of flow diagrams in conjunction with block diagrams to indicate time sequence."

R. K. Richards: That is a difficult question to comment on because there are so many different types of multiple circuits that differ from each other in their logical nature; and there are so many various component types that I am at somewhat of a loss to comment on this.

Chairman Amdahl: The next question is addressed to Mr. Bensky from Gordon Smith, Autonetics: "Do you really mean that the development of the B-box could not have resulted from the logical equation approach?"

L. S. Bensky: I do. I agree that it is not impossible to develop the "B-box" by logical equation, however, I do not think that we have seen the end of the development. Someone will come up with a logical equation system that allows invention of that sort.

R. K. Richards: I do not think that as far as the invention is concerned, either the block diagram or the Boolean algebra approach will invent things for you. This would be deduction rather than invention, and I feel that invention is a matter of who is doing the work.

Chairman Amdahl: The next question is for Dr. Richards from Richard Tanaka, Lockheed Missile Systems: "Assuming that the two approaches are probably of more or less equal merit today, which approach seems to have the greater future potential?"

R. K. Richards: As far as thinking development is concerned, the logical equation has more potential. There is no question about that.

For the large-scale computers that I have been designing, I have been using both logical equation and block diagram. As I see it now, and as the situation is at present, we will be using both for some time to come.

Chairman Amdahl: This question is addressed to Victor Hesse, and is from Verne Olson, Librascope Inc.: "Why are existing computers far from optimum?"

V. L. Hesse: If they were optimum, I am sure that we all would be out of jobs. The computer art is obviously in the early stages of its infancy, and I do not think that we have had anywhere near enough time to come close to optimum. It still remains, however, as I can picture it at least, if we will ever get there, it is just a matter of time and of going in the right direction.

Chairman Amdahl: The next question is also for Mr. Hesse, and is from R. G. Leitner, System Development Corporation: "Some of your remarks indicated that any computer designed using block diagram methods could be redesigned by your colleagues resulting in a cheaper, simpler, and better machine. Do you consider this to be true?"

V. L. Hesse: This is an all-engrossing point, because if you say any computer that was made from block diagrams would include a rather large number of such machines. I would hate to commit myself, other than to point out that any of them are able to be improved. However, I do believe that with a given problem and given specifications for a computer, that the two methods applied by two individual groups working in parallel, without communication between them, the people using the logical equation would achieve a better result, a result that would involve less equipment, and less time for the group itself.

Chairman Amdahl: The next question is addressed to Dr. Engel from A. Karson, G. C. Dewey and Company: "Can you give some idea of the ease of working with, and the complexity of the Boolean equations when system feedback is incorporated in the equations? Please compare this with a block diagram approach."

H. E. Engel: There have been other questions that other members of the panel have been able to give considerable answers to, and this is a general question that I do not think that I can answer shortly. I can not give a suitable example at this time.

Chairman Amdahl: The next question is directed to Mr. Hesse from Sherman Klein, Hughes Aircraft Corporation: "Have you designed in redundancy after minimizing the variables?"

V. L. Hesse: No. The computers that I have worked with have nearly all been for military projects where size, weight, volume, etc., were almost of prime consideration. Therefore, working intentional redundancies into the problem for the purpose of increasing reliabilities were not included. Actually the minimum form of the equation will really often conserve redundancies, and this is the purpose of making it smaller and with less equipment. But they are not for the purpose of reliability, that is as far as designing it in for the purpose of reliability.

Chairman Amdahl: The last question is directed to Dr. Richards, and is from Carl E. Jensen, General Electric Company: "Can automation be applied to your approach?"

R. K. Richards: Automation can be applied to logical equation, there is no question of that. However, I would like to make clear again, that when comparing the logical equation with block diagram, the logical equation is limited to certain types of computer components and circuits; and when you use other types it may be considered preferable to retain block diagrams. The relative merits of block diagrams and logical equations can be reduced to the relative merits of the different physical realizations of logical functions.

Contrasts in Large File Memories for Large-Scale Computers

JOHN A. POSTLEY
NONMEMBER AIEE

REQUIREMENTS for the use of very large files in digital computer systems for both scientific and business applications arise in a variety of circumstances. Certain characteristics of these applications tend to determine the type of file most suitable, where the file may contain well in excess of fifty million alphanumeric characters of information. These characteristics, some or all of which will be important in a particular application, include: 1. the upper and lower size limits of the file and the rate of change in total file size, 2. the volume of data which must be entered into and read from the file, either by other components of the mechanized system or by elements external to the mechanized system, 3. the speed with which each such reference must take place, 4. the average amount of data in one record and the variations in this amount, 5. the form of the original data and the desired form of information extracted from the file, 6. the control requirements dictated by particular kinds of files, especially the means of identification of records contained therein, and 7. the likelihood and implications of any unavailability, temporary or permanent, which may occur of information stored in the file.

Large files have been divided historically into two classes, called serial access and random access files. It now appears that the spectrum between serial and random access files is essentially continuous and, hence, that the attributes of one can be traded off for more advantageous attributes of another, in a particular application. The relationships between the engineering attributes of very large files and the characteristics of particular large file applications are subject to a certain amount of interpretation with respect to the previous list. It is the purpose of this panel to establish the nature of these relationships from the various points of view represented.

General Characteristics of Problems with Large Files

In the short period comprising their history, large files have been classified as sequential or random access equipment. Sequential access files, typified by magnetic tapes, actually are sequential within each group (tape), totaling approximately five to ten million characters each, but can be treated as random among groups. Access time to unpreselected items in sequential access files ranges from several minutes to several hours. Random access files are typified by magnetic drums or discs. They are characterized by access times of from several milliseconds up to about one second.

It is important to note that there are two types of processing, not necessarily corresponding to these memory classifications, namely: "on-line processing" and "batch processing." On-line processing involves the execution of all of the appropriate operations performed by the system on each transaction that has entered the system before proceeding to the next transaction. Batch processing involves the execution of one or more, but not all, of the appropriate operations on the entire batch of transactions before proceeding to the next operation; hence, it involves at least two passes of a transaction through the computer system.

Observe that the decision as to which operations are in fact the "appropriate" ones is often extremely important and complex, in spite of the typical operations research assumption to the contrary. Both problem and computer considerations are highly relevant. It is not necessarily true that the need to "interrogate" the file dictates either random access equipment or on-line processing, since 1. a form of random access may be possible with sequential access equipment and 2. only access and the corresponding printing may be required, and hence processing by the computer may not be involved at all. Conversely, random access capability may yet provide significant advantages over sequential access in a system in which only low-activity sequential processing is involved.

The main advantage of the on-line type of processing is its ability to deal only with the file information relevant to the transactions at hand, completely bypassing the file information irrelevant to these transactions. In some problems the irrelevant information may comprise about 98% of the file. The main advantage of batch processing is that, for a given problem complexity, less logical capacity of the central processing unit is necessary than in on-line processing, due to the more complex job (discussed previously) done on each transaction in the random case.

Information: Type and Capacity Considerations

The total information capacity of a file which might be called "large" in the present context is assumed to be in excess of fifty million characters. Fortunately, in most problems involving such files a relatively simple analysis does indicate to a sufficient degree of accuracy the actual size of file which will be required. Whereas the typical magnetic tape file can be expanded to contain more and more information at trivial cost (the cost of the necessary number of reels of magnetic tape), the capacity of the corresponding random access file is increased in steps of relatively large cost and may, for a particular machine, reach an absolute maximum.

The volume and speed of input and output required is basically of two kinds: that between the entire machine system, including the file, and the people operating the system; and that, within the machine system, between the computer or control unit and the file. The most stringent requirements for fast access are a matter of seconds in the former case but may be a matter of microseconds in the latter.

Several factors tend to determine the optimum volume and speed of input and output between the machine system and its environment, but the most important of these are the problem requirements themselves. The true problem requirements are not necessarily the stated ones, and hence, if the latter impose a significant burden on the procedures or the equipment, they must be modified to produce an efficient system over-all. Does the inventory manager really need stock balance information on any item in seconds? Must the status of every record in the file be printed out in daily or weekly reports? Can a potential customer for an airplane or train seat or a hotel room be allowed to wait several minutes for availability information?

The volume and speed of communication within the computer system between the file and other elements of the system is a different matter. Overlapping of different system functions such as file

JOHN A. POSTLEY is with the Rand Corporation, Santa Monica, Calif.

searching and computing, and paralleling of such similar functions as searching for information in the file, are among the more important considerations which can completely alter the apparent internal volume and speed requirements of very large files.

While some magnetic tape systems provide the means to record variable amounts of information in any single record, no known random access system does this. If the size of such records in a particular problem is highly variable, fixed record size may result in a substantial increase in unused storage space in the file or complexity in assembling variable working records stored piecemeal in fixed records of the file.

Logical and Physical Control

Another important contrast generally evident is that of control; control of random access devices is considerably more complex than that of sequential access devices in terms of the hardware or the program or perhaps both. One of the primary reasons for this greater complexity in the case of random access memory is the need to place the identification key into one-to-one correspondence with the location of the item in the random access file when no mathematical relationship exists between the key and the location. This operation is essentially "indexing"; it is an integral and patently simple part of a sequential processing operation, but is entirely exogenous to the processing operations when the random access file is employed and significantly increases the complexity of the operation.

Physical control of the search operation is likely to be far more complex, i.e., costly, with random access files than with sequential files since the sequential method of searching is the simplest, most straightforward one possible.

Information Unavailability

In direct opposition to the whole concept of random access to file information is the fact that information so stored is unduplicated elsewhere and, therefore, highly vulnerable to permanent loss or not available on a special basis when the equipment in which it is stored is in temporary disrepair. These difficulties are far less serious in the case of magnetic tape files.

The major significance of this fact is that relatively elaborate and costly measures are required in the random access case to reduce the incidence of temporary or permanent information unavailability. Very few indeed are the applications where such unavailability does not present a serious problem.

A Research Conjecture

Although the spectrum from sequential to random access is essentially continuous, the middle area is often particularly difficult to discuss without a specific application in mind. On the basis of the previous discussion, however, it appears useful to think of sequential access as a special case of random access (or vice-versa) as regards their quantitative aspects. This is possible since access among sequential groups is truly random and these groups can, at least in theory, be made as small as necessary to reduce access within them to any desired time. By doing this one may be in a position to answer questions relating to the "ideal" size of each access group; *e.g.*, magnetic tape or magnetic disc file, and the optimum control configuration with respect to these groups.

It might be interesting to consider the entire large file problem stated symbolically as follows: suppose that the computer system consists of k levels of storage and control, with s_k and c_k respectively designating the storage and control at each level. The computer is then the control c_1, and the working store the storage unit s_1. By associating access times and logical capacities of these levels of storage and control as parameters and under the constraints of a particular problem context represented by the equations of the system, it may be possible actually to solve for a preferred hardware configuration. Perhaps unfortunately, the author has not developed such a set of equations and parameters, but the conjecture appears to be very interesting nonetheless.

Conclusions

The increasing requirement for very large files in digital computer systems has led to the identification of several important characteristics of these files, and to the development of files which exhibit these characteristics in varying degrees. As a result, a new situation has been created wherein a detailed study of these characteristics will now be necessary in some applications to determine the particular file most suitable for the problem or problems to be dealt with.

Methods of File Organization for Efficient Use of IBM RAMAC Files

W. P. HEISING
NONMEMBER AIEE

THE International Business Machines Corporation Random Access Memory Accounting Computer (IBM RAMAC) is a disk file storage unit with superior characteristics for use in processing a large class of commercial applications.

The RAMAC unit is produced in two different versions for use in *305* and *650* data-processing systems. The two types of RAMAC's have slightly different characteristics which reflect differences in the central processing units with which they operate. The principal factors to be considered in their use are sufficiently similar for purposes of this paper, however, that only the *355* RAMAC which is used in IBM *650* systems will be discussed.

The type *355* RAMAC contains 6 million digits of file storage. The unit contains 50 disks mounted on a vertical shaft rotating at 1,200 rpm. The disks are coated on both sides with magnetic material and are separated to permit entry of read-write heads mounted on horizontal arms.

An arm may be positioned in one of 100 positions radially in order to read or record serially on one of 100 tracks on each disk face. The arms also move from one disk to another vertically in a retracted position. Each of the 100 disk

W. P. HEISING is with International Business Machines Corporation, New York, N. Y.

faces has 100 tracks, which in turn each contain 60 words (10 decimal digits plus sign).

Each RAMAC has three arms, which may be operating simultaneously. However, since the three arms transmit information between the file and a single 600-digit core buffer, only one arm may be transmitting information at a time. Transmission always occurs in blocks of 600 digits (a full track).

Up to four RAMAC's with a total capacity of 24 million digits may be attached to the *650*, which may also have tape units.

The basic *650* instructions for using the file are "seek," "read," and "write." "Seek" directs a designated arm of a designated RAMAC unit to locate itself at a specified one of the 10,000 tracks. "Read" and "write" initiate information transfer through a particular arm between the file and the core buffer. An interlock delays transmission if the arm has not completed its last "seek" instruction.

The "seek" time will be dependent on the number of disks and tracks over which the arm must move. The average time for "seek" between arbitrary file locations is 0.600 second. The average time between locations on the same disk is 0.160 second. The adjacent track can be found in 0.080 second.

Writing requires 135 milliseconds (ms) and reading 110 ms. It is evident from these speeds that it is not at all necessary in system planning to order the incoming transactions in a sequence corresponding to the file ordering. This has many consequences that are not immediately obvious.

Rather than plunge into a discussion of specific techniques for RAMAC use at this point, several different philosophical approaches to the file problem will first be discussed.

The essential purpose that file organization must serve is to facilitate information retrieval. The simplest method of file organization is to have no organization. The author must confess to using this method himself. To find a given record in the desk, he keeps going through the pile until he finds it.

As the file becomes larger, the method becomes less and less efficient. Although there is some prospect that future developments, the cryotron in particular, might use this method, it is not generally useful in present file systems.

Other methods of using the file all depend on some method of file ordering. A telephone directory has records ordered alphabetically within town or city, for example. In this instance, as with most but not all methods of file organization, it is easily seen that an artificiality has been introduced for no other reason than to aid information retrieval. There is generally no logical connection between successive entries in a telephone directory.

The situation is similar with manual filing systems. Most manual filing systems have a very important characteristic that is so obvious that it is often overlooked; additions, changes, and deletions to the file are simple operations. Records can be inserted, replaced, or thrown away without affecting the rest of the file. This characteristic is extremely desirable in any system of file organization. Manual and punched card systems readily permit additions and deletions, sequential file systems like telephone directories and conventional tape systems must be completely rewritten to incorporate a single addition directly into the file.

Like manual and punched card systems, the RAMAC disk file units have this important ability to accept additions, deletions, and changes with minimum disturbance to the rest of the file. Although this ability is related to the random access characteristics of the file, it does not automatically come to mind when the term "random access" is used. For business data processing, this ability to make changes individually is as important as the efficiency of the random access itself.

Techniques for using the RAMAC in this way will now be mentioned. As in all file systems with any type of organization, the information retrieval problem reduces to the following: Given part of a record, the item key, find the entire record.

The "item key-RAMAC address" relation therefore becomes the crux of the problem. If the item keys form a single group of consecutive numbers, the obvious solution is to use the item key as the RAMAC address and no searching is involved. In many applications, this direct method cannot be used, however, since the item keys are not consecutive.

At first thought, one might consider using an index, each entry gives the item key and the RAMAC address containing the complete record, but this does not solve the problem; it merely transforms it. Finding the index entry corresponding to a given item in the index file is exactly the same problem.

A description of a method developed for business files follows. To find an item in the file, the item key is operated upon by an arithmetic algorithm to generate an address. For example, suppose that the file addresses 5,000 through 6,986 have been set aside for a particular file. A possible algorithm would be to divide the item key by 1,987 (the number of tracks) and to add 5,000 to the remainder, resulting from the division. The record at that address is examined to determine if it is the record sought. If not, a special field in the record called the "chaining" field contains another RAMAC address or no address. If the special field contains an address, the record found at that address is in turn examined. If the second record is not the item sought, its chaining field is examined. The process is repeated until the record is found, or until a chaining field containing no address is reached. The latter case is taken to mean that there is no record in the file corresponding to the given item key.

An arithmetic algorithm is chosen which generates a "pseudo-random" number from the item key. The initial loading of the file is performed in two steps. During the first stage, every record is loaded into the address given by the algorithm provided there is space. Otherwise the record is not entered. For those records which are entered into the file during the first stage the chaining field is left blank.

During the second stage, all of the records which were not entered during stage one are now loaded. The procedure for entering each record in this stage involves extending the chain beginning at the address given by the algorithm. The record itself is entered at an empty file location, and its address is put in the chaining field at the end of the chain beginning at the address generated by the algorithm.

Mathematically, an algorithm can be found which will generate addresses such that the number of addresses having given numbers of duplicates obeys the Poisson distribution. Using this, it is possible to determine the expected number of "seek's" to find a typical record. This is found to be 1 plus $f/2$ where f is the fraction of the file space containing records. Thus, if 9,000 out of the 10,000 tracks are used for active information, 1.45 records, or "seek's," must be examined on the average to find the one sought.

The system permits additions to or deletions from the file very readily and thus is analogous to manual or punched card systems in this respect.

Strictly speaking, random access is not a property of hardware, but a method of use of equipment. The random access method has certain other advan-

tages not always recognized as being derived from it.

One has only to consider, for example, the large amount of time devoted to sorting, merging, and collating in tape systems to realize it is a very real economic and programming headache. In virtually all cases these operations are required solely to facilitate information retrieval, e.g., to order the transactions in the same way the file is organized.

Random access systems tremendously reduce the complexity of many business file applications because sorting is done in the best way conceivable; it is entirely avoided.

With reference to Dr. Postley's list of application characteristics, the RAMAC characteristics are as follows:

1. Data flow rates of 600 digits per second between the file and the processor would be typical.

2. Effective access rates of one second per record.

3. For direct addressing systems, no identification need be in the file, although for programmed verification, it is frequently desirable to retain the item key as part of the record.

For chaining systems the central processor must utilize a simple address-generating algorithm. In addition, the item key must be explicitly retained in the record, plus a chain address field of five digits to enable the information retrieval method to operate.

4. 100- to 300-digit, or character, records are typical. Applications requiring records larger than 600 digits can be handled by program control, however, the system planning is slightly more complex.

5. A built-in file-protect device can be activated (but not by program control) to prevent writing in the file. For equipment or program error correction, periodic emptying of the file to provide restart points is frequently used. These are often desired or audit purposes anyway. Upon restart, the entire file must be reloaded and all transactions reprocessed. This is a slow process, and every effort has been made to insure equipment reliability. Guards against errors arising from faulty input data, however, are often employed just as in tape systems.

6. 10 million characters are the present upper limit for the *305* RAMAC and 24 million digits is the upper limit of the *650* RAMAC system.

7. Punched card or tape input of alphabetical and decimal information is handled. A typewriter can be used as an inquiry station for handling urgent requests. The speed of handling special inquiries is of the order of one second per request, not including the time for typing the request and the time for the response to be typed back.

The Design and System Aspects of the HD File Drum

H. W. FULLER
NONMEMBER AIEE

S. P. WOODSUM
NONMEMBER AIEE

R. R. EVANS
NONMEMBER AIEE

THE HD File Drum was developed by the Computer Products Division of Laboratory for Electronics, Inc., to meet the ever-growing demands for fast, inexpensive and large-capacity random access memories. The initial application of the HD File Drum was its use in a multidrum file in the DIANA system, which is a general-purpose business machine designed and built by Laboratory for Electronics, Inc. The File Drum is now finding application in various computer systems that require the reliability and rapid random access that it affords. The ability to form arbitrarily large multidrum files without effecting the random access time is frequently valuable. The self-clocked reading method, moreover, allows the use of wholly variable word and block organization.

A picture of the HD File Drum cabinet is shown in Fig. 1, and a partial list of characteristics for one drum is given here by way of introduction to the technical account that follows:

Total storage capacity: 15 million bits
Random access time: 180 milliseconds (average)
Drum speed: 180 rpm
Drum cylinder diameter: 15 inches
Drum cylinder length: 14 inches
Number of tracks: 300
Packing density: 1,040 bits per inch
Size: 48 by 29 by 45 inches

Drum Design

A set of initial performance objectives originally dictated the general form of the present File Drum. The requirement for rapid random access, for example, suggested the use of stationary heads to avoid time-consuming mechanical positioning. The prospect of one head and one head-selection device per track argued for a large number of bits per track in the interest of economy. This latter led to the preference for a relatively large drum diameter and high linear bit density, and resulted in the use of hydrodynamically maintained head separation and self-clocked reading to eliminate the need for tight mechanical tolerances. Other objectives which similarly effected the consequent design will be pointed out.

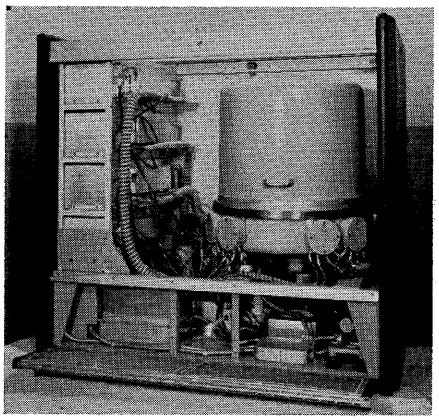


Fig. 1. Complete file drum cabinet, front panels removed

Requirements of High Density Recording

It is well established that high-density (short wave-length) magnetic recording requires, in general, a small effective head-medium separation, a small effective head gap (for longitudinal recording) and a high coercive-force recording medium.[1-5] A small physical head-medium separation of 180 μinches (microinches) is achieved in the HD File Drum by continuously supplying the drum surface with a film of oil and causing the head surface to act as a hydrodynamically lubricated slider bearing. A simple spring-loaded pivot mount allows the head to contact the drum during the starting and stopping of the drum. Ferrite is accordingly used as a head pole piece material since permalloy develops a low permeability layer with mechanical working which results in an increased effective separation between head and medium.

Start-stop contact between head and medium demands a durable recording surface which is given by the use of the material, Cunife I. In addition to providing a tough, machinable surface, Cunife posesses a relatively high coercivity of 500 oersteds required for high density recording.

The short head gap required for high density recording is achieved by optically lapping the mating faces of head pole pieces, and by using precision-rolled berrylium-copper strip of 250 μ inch thickness to form the gap spacer. The bearing surfaces of assembled heads are also optically lapped. The edges of the resulting gaps must be well defined to achieve the required small gap lengths, and Ferroxcube 3C material is accordingly used since it possesses relatively low porosity. The final lapping operation is equally important in this respect. The *3C* material has a high-saturation flux density, furthermore, which is important in connection with the high coercive force of the Cunife recording medium, and also posesses low losses at the drum operating frequency of 150 kc.

A section of a typical head gap is shown in the micrograph of Fig. 2. The gap shown has a gap length of 340 μ inches. By quality control of head fabrication, gap lengths are maintained within a tolerance of $\pm 10\%$ about a nominal length of 340 μ inches. Heads are inspected also for a so-called gap-loss figure which establishes that the accumulated edge imperfections are not excessive.

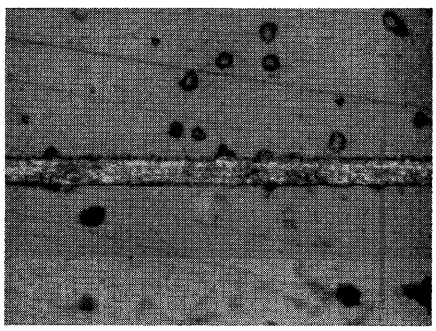


Fig. 2. Micrograph of head gap

Mechanical Design

The principle of hydrodynamic lubrication[6] applied to head support in the HD File Drum is indicated schematically in Fig. 3. The upper drawing shows the planar head surface resting in line contact against the nonrotating cylindrical drum surface, where the relative size of the drum is much reduced to emphasize the

H. W. Fuller, S. P. Woodsum, and R. R. Evans are with The Computer Products Div. of Laboratory for Electronics, Inc., Boston, Mass.

The authors wish to thank the Laboratory for Electronics, Inc., for permission to publish this material.

The authors wish to acknowledge the contributions by many people to the work reported here, in particular those of Robert C. Kelner, General Manager of the Computer Products Division. Additional contributors, in part, are Charles W. Gardiner, Harold E. Lerner, Robert T. Pearson, Stanley O. Cheney, William J. Gorman, and Walter Loyte.

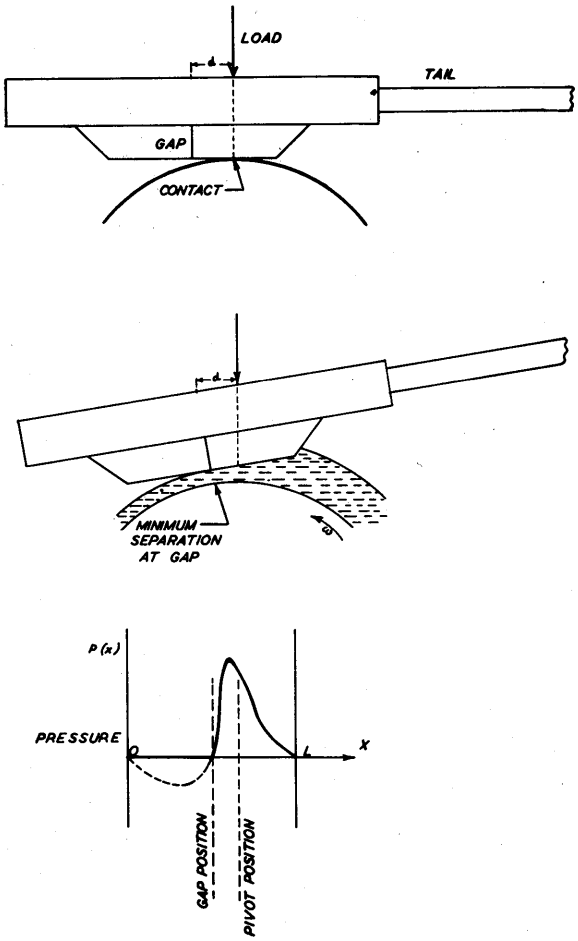


Fig. 3. Schematic illustration of hydrodynamic head lubrication

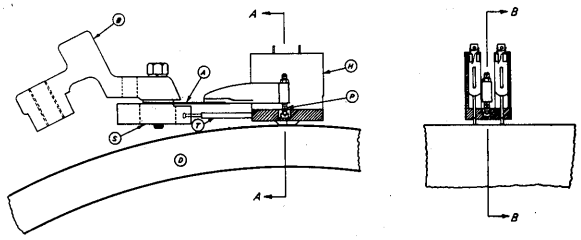


Fig. 4. Dual head and head-mounting arrangement

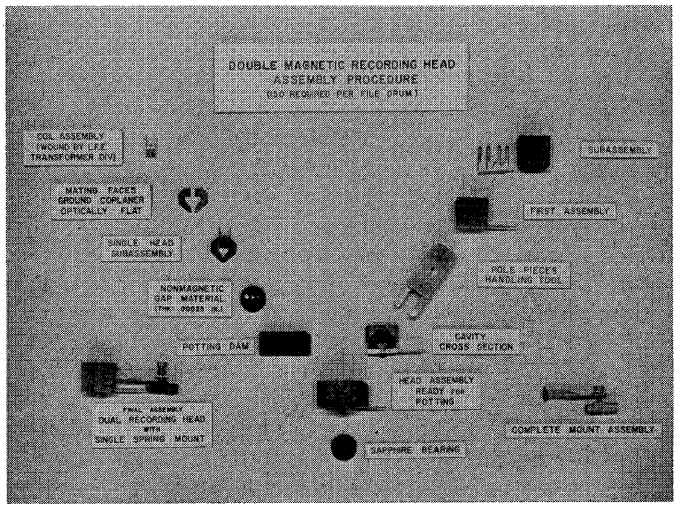


Fig. 5. Head parts and assembly procedure

contour of the separation between head and drum. The bearing surface of the head is actually 200 mils long and 30 mils wide, while the drum diameter is 15 inches. Note that the head gap is centrally located on the head bearing surface, while a load force is applied at a pivot point located a distance d toward the leading edge of the bearing. This results in head-medium contact directly below the pivot point, and distant from the gap. When the drum rotates counterclockwise (center drawing) carrying a surface film of oil, a positive pressure distribution, $p(x)$ in the lower drawing, arises between the drum and the head surface which lifts the head against the applied load. Since the integrated pressure under the bearing decreases with increasing separation, an equilibrium separation is reached. An equilibrium tilt angle is also reached, as shown, corresponding to a vanishing moment about the pivot point of the pressure distribution $p(x)$.

For the nominal surface velocity of the drum surface, and for the stated head-medium geometry mentioned, a pivot point location, bearing load and oil viscosity were chosen to give the desired minimum separation of 180 μ inches, and to give a tilt angle causing the central gap to be at the minimum separation position. Solutions of Reynold's equation describing the behavior of hydrodynamically lubricated bearings have been given for narrow plane slider bearings and for very wide convex slider bearings,[8] but since solutions have not been obtained for the present case of a narrow convex slider bearing, the determination of the appropriate parameter values were obtained empirically.

The pressure distribution drawn in Fig. 3 includes a dotted region of negative pressure on the trailing side of the minimum separation point that is predicted analytically, but this does not occur since the oil film ruptures under tension. The distribution is also given as a function of distance along the bearing length only, while in fact the pressure is a function of the distance along the width as well because of the large side leakage that occurs in such a narrow bearing.

The actual head package and mount are shown in Fig. 4 where it is seen that a head package H contains two heads with coplanar bearing surfaces. The heads are separated in the dual head package to allow a spring load to be applied between them. The leaf spring A, which should be shown in an upward-deflected position, applies a normal force through the pivot pin P to a sapphire cone jewel mounted in a cavity in the head package. A tail T is integral with the dual head package which engages a slot in the head mount part labeled S. The tail centerline passes through the pivot point, and so allows the head package the desired degrees of freedom in pitch and roll, but restricts unwanted yaw.

The head bar B of Fig. 4 holds 20 such mounts, and is attached to a drum cage that is not shown. The end view along section AA shows the ferrite wafers and coil bobbins. Note that the pivot point is offset in this view as well. This is done to compensate for the gravity moment acting on the dual head package which would otherwise unbalance the forces applied to the two heads. The hydrodynamic offset is visible in the side view, although it is much smaller than in the exaggerated drawings of Fig 3. The hole in B and the second slot in S, finally, afford the means for preadjusting the

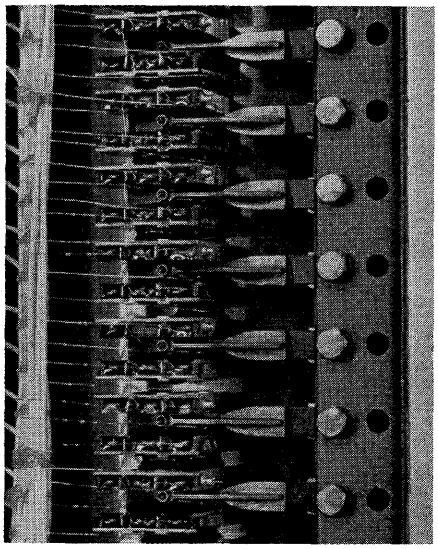


Fig. 6. Close-up of heads mounted on drum

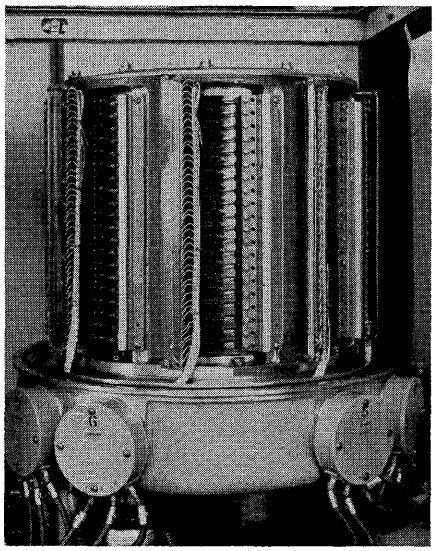


Fig. 7. Basic drum assembly with cover removed

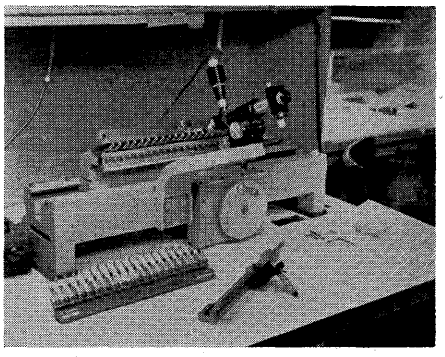


Fig. 8. Pivot aligning fixture, and bar holding 20 head mounts

pivot point position by means of an eccentric tool.

Fig. 5 illustrates the dual head and mount and the assembly procedure. Ferrite pole piece pairs with gap spacers and bobbins in place are inserted through the open tops of the magnesium cavity and through precision punched slots in the bottom from which they protrude. Springs in the sides of the cavity ensure firm mating of pole piece pairs until the heads are potted. The epoxy filling is done with a rubber dam in the middle region between heads, and is removed after curing. The protruding ferrites are ground and lapped thereafter, where the head surfaces in this operation are made coplanar to within one wavelength of helium light. Production is facilitated by mass grinding and lapping in a special fixture.

A section of a bar of heads is shown mounted on a drum in Fig. 6. The fine connecting wires were chosen to eliminate any effects of wire stiffness on the force system supporting the head during drum operation.

Fig. 7 shows the basic drum assembly, with the cover removed. There are eight bars of heads placed in eight windows about the drum circumference, and the accurate interlacing of all 320 tracks depends on several things. First, the protruding ferrites of a dual head package must be accurately eight tracks apart which is assured by the slots in the head cavity. Second, the cone jewel must be located accurately in each package with respect to the head bearing surfaces, including the hydrodynamic and gravitational offsets. This is accomplished by positioning the jewel in the jewel cavity with metal-filled epoxy on a jewel-setting fixture. Third, the pivot points of a bar must be accurately spaced 16 tracks apart and positioned with respect to a reference point on the bar. This is done optically with the pivot aligning fixture shown in Fig. 8, where the pivot points are adjusted to ruled lines on the polished aluminum standard, and where the reference is obtained by a dowel pin in the fixture and dowel-pin hole in the bar. The fourth and final step is the alignment of each of the eight bars on the drum cage to be vertically staggered by one track distance (40 mils) between bars. Provision for this is made at the time the cage casting is machined by properly indexing dowel-pin holes in the bar-mounting slots around the drum. Note that all bars and heads become dimensionally identical and interchangeable by these means.

The precision alignments guaranteeing track interlacing are thus either alignments that occur automatically, or that are carried out before assembly of the drum on convenient fixtures.

Referring again to Fig. 7, it is apparent that a cable of head leads is brought down the drum cage and out through shielded glass feed through headers in the base casting to multiple-lead shielded cables. The headers, which are invisible behind shield cans, are *O*-ring sealed since the base casting forms the sump for the oil system. Referring to Fig. 1 it will be seen that the multiple-lead shielded cables travel to the boxes arrayed along the left-hand end of the cabinet which contain the relay head-selection matrix. The single output of the selection matrix runs to the preamp-write unit that rests on the base plate of the cabinet which is located underneath the selector.

The hydraulic pump motor is seen in the lower right-hand corner of the cabinet. The pump delivers oil under pressure through high- and low-pressure cut-off safety valves, through two fine-particle filters to the base casting of the drum. A pipe carries the oil to one hollowed vertical member of the drum cage, which is covered on the inside with a plate containing small discharge holes. The oil thus sprays continuously on the drum surface and returns to the sump for recirculation. The oil not only serves as a lubricant, therefore, but serves also to wash continuously the surfaces of the drum and heads.

The 180-rpm, 1/8th-horsepower hysteresis-synchronous drum-drive motor can be seen in Fig. 7 to protrude from the bottom of the base casting. The motor is flexibly coupled to a shaft passing into the base casting which in turn drives the drum through pulleys and paracril timing belts yielding a 10 to 1 speed reduction. The final gear is mounted on the drum cylinder which turns on Timken tapered roller bearings against a nonrotating spindle. The driving arrangement permits the surface speed variation, relative to the nominal 60 cycles per second line, to be held well below 0.5%.

A final note on mechanical design concerns the preparation of the Cunife surface. The high H_c of Cunife I is obtained by successive cold working and heat treating and occurs uniaxially along the direction of rolling or drawing.[9] The recording surface is accordingly formed by wrapping a long length of Cunife wire in a helix around the aluminum drum cylinder. A final-stage die was made for a chosen rectangular wire cross-section of 30 mils by 100 mils. The wire is fastened to the drum at the beginning of the helix, is wound, under tension, and is fastened again at the end. The Cunife is bonded

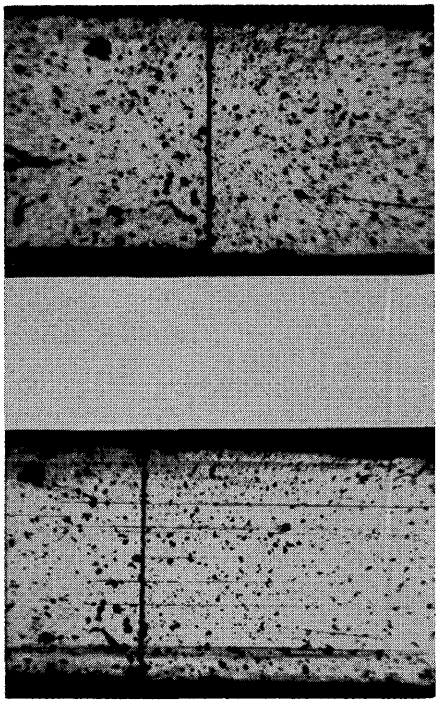
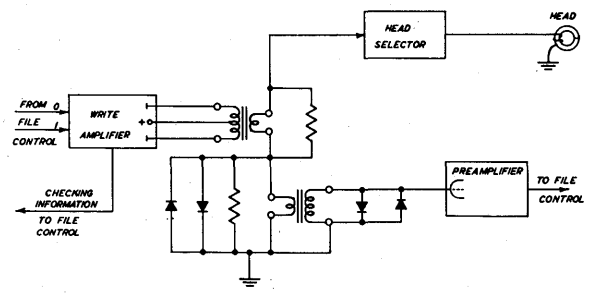


Fig. 9. Micrographs of head bearing surface before and after 520 start-stop operations

Fig. 10. Electronic circuits within drum cabinet



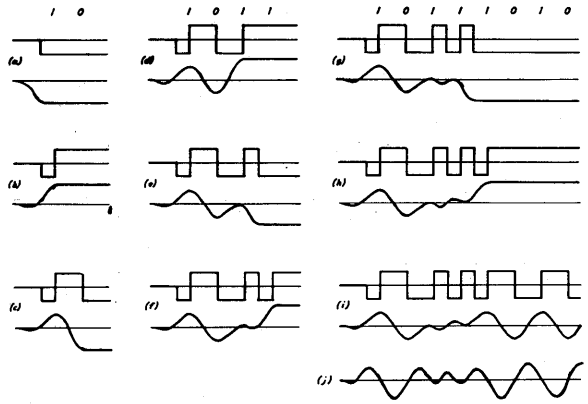


Fig. 11. Successive pairs of writing current and recorded flux waveforms, together with head voltage waveform during reading (j)

to the drum surface by the use of an epoxy resin that is applied before the winding operation. The wire is forced turn against turn as the winding proceeds in order to minimize the separation between successive turns of the helix.

The wound drum is heat cured and is thereafter cylindrically ground. The surface is then optically lapped to achieve a surface finish in the order of 1 μ inch rms. The final surface finishing is, at present densities, much less important to the magnetic behavior of the drum surface than it is to the physical effect of the drum surface on the heads during the contact periods of starting and stopping. The heads are out of contact during steady operation, but Fig. 9 shows a comparison of the typical appearance of a head bearing surface before and after 520 start-stop cycles. The surface appearance levels off after this number of cycles, with little change occurring between 200 and 520 cycles. The effect of these scratches on the head performance is undetectable. Note that the scratch marks center under the pivot point rather than at the gap in accordance with the afore-mentioned pivot offset.

Magnetic Recording Aspects

It will be seen in the next section that the self-clocked circuits used in the HD File Drum require rapid switching from reading to writing so the final writing stage must be electronically connected to the preamplifier input. The manner in which this is done is shown in Fig. 10 which indicates all the electronic circuits that are housed within the File Drum cabinet. The double-ended final writing stage is connected by a 3-to-1 transformer to the head through the selector, and the secondary of the writing transformer is placed in series with the primary of a 1-to-15 reading transformer. Silicon diodes of type *1N251* shunt the reading transformer to remove effectively the reading transformer from the circuit during writing when a short current rise-time is required. Separate resistors provide correct damping for both writing and reading. The presence of the writing transformer reduces the effective turns ratio to 10 to 1 during reading. Under these conditions the reading level at the preamplifier input is nominally 45 millivolts peak-to-peak. The writing current in the head is 300 ma peak, using a 30-turn bobbin and an inductance of 100 microhenry.

The head selector matrix within the cabinet contains 109 Clare relays with mercury-wetted contacts. These were chosen for their long life, and for their ability to switch small signal levels dependably.

The writing method used in the HD File Drum has been previously referred to as modified nonreturn to zero (NRZ) recording[10] where this is the same as double-pulse return to zero (RZ) for one-half bit pulse widths, and where both of these are identical with William's phase-modulation method.

The low-frequency B-H loop for Cunife saturates quite slowly even for long samples which is disadvantageous in short wavelength recording. More serious is the thickness of the medium which is infinite for practical purposes, 20 mils compared with the 1- and 2-mil wavelengths that occur in the HD File Drum. It is to be noted as well, that the present application demands erasure of previously written information by a single asynchronous over-writing of new information. The problems presented by these shortcomings of the medium are overcome by the use of modified NRZ recording since the frequent reversals of recording field act to maintain the lower depths of the medium in a demagnetized state, and to maintain the surface layer balanced about zero. It is possible to verify these remarks by recording a pulse at twice the nominal current level, whereupon a disturbance in the reading waveform base line is observed that quite slowly disappears after many rerecordings with normal writing waveforms. This latter recovery is a good indication of the stability of the normal recording process.

An attempt is made in Fig. 11 to illustrate the effect of the considerable overlap of the recording head field on the previously recorded bit cell. The bit sequence is built up slowly to show the result in the recorded flux distribution of terminating the writing current reversals at

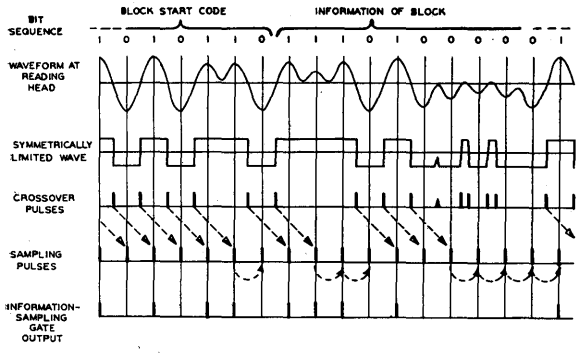


Fig. 12. Key waveforms occurring in self-clocked reading circuits

successively later times in the bit sequence. The lower waveform of each pair is thought of as the effective flux seen by the windings on the recording head. It is seen in the waveform at (*i*) that in spite of the overlapping of bit cells by the recording field during writing, a representation of the bit sequence is apparent. The time, or space, derivative of flux shown in (*j*) is easily interpretable by associating positive peaks with ones and negative peaks with zeros at one-bit intervals, where the phase of the peaks has shifted to the left by about one-half bit cell with respect to the writing waveform. The waveform of (*j*) is that of the voltage across an idealized reading head.

Note the positive monotonic drift of the flux pattern in (*i*) in the region of three ones, giving rise to the symmetrical lifting of the short-term average level of the waveform in (*j*). This observation makes plausible the waveforms to be considered next.

The previous remarks serve to indicate the purpose of one test procedure used to establish the reliability of tracks on production HD File Drums. The test consists in part of recording blocks of random information on a track at a writing current that is 15% greater than the nominal, followed by a single asynchronous over-writing of random information using a writing current that is 15% less than nominal, followed by an appropriate interval of errorless reading. The writing amplifier is designed for a maximum long-term current fluctuation of 5%, but the additional margin established by the afore-mentioned test allows for the variation of other parameters that are interpretable as effective writing current changes.

Self-Clocked Reading Method

Some key waveforms and pulses that occur in the present self-clocked reading circuit are shown in Fig. 12. An earlier method of self-clocked reading was given[10] that required sufficient short-wavelength resolution to produce crossovers of the reading waveform at the center of each bit cell. The self-clocked timing phase derived from bit n in that case was responsible only for the reading of bit $n+1$, whereupon a new timing phase became available. The requirements on the short-wavelength resolution of the drum and head are relaxed in the present method as evidenced by the typical reading waveform of Fig. 12 in a region of like bits. Advantage is taken of this improvement to increase the bit density and the writing current.

Since no angular rotational reference is available in a completely self-clocked system, it is therefore necessary to mark the beginning of a block of information with a distinctive code so that serial bits may be appropriately grouped into characters.

A block start code consisting of a sequence of alternating ones and zeros was chosen for this purpose, where the length of the sequence is made longer than the maximum length (11 bits) of a similar pattern that can ever occur within a block of valid characters. This detection operation is performed by using base-line crossover pulses derived from the symmetrically limited reading waveform. These are shown in Fig. 12 for a sample bit sequence.

The distinctive property of the crossover pulses associated with an alternate one and zero region is the one-bit time interval between their occurrence. Detection proceeds, therefore, by commencing to count the minimum number of crossover pulses P_{co}, indicative of a block start

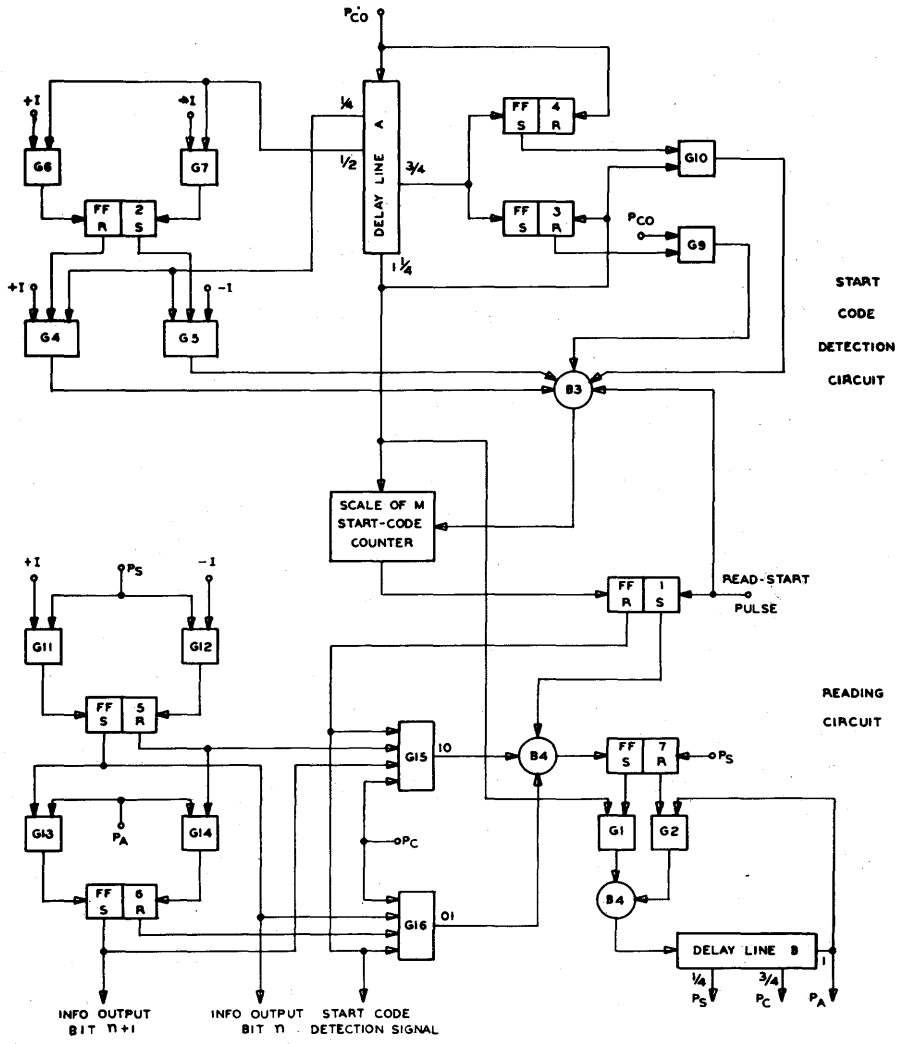


Fig. 13. Logical diagram of the basic start-code detection and self-clocked reading circuit

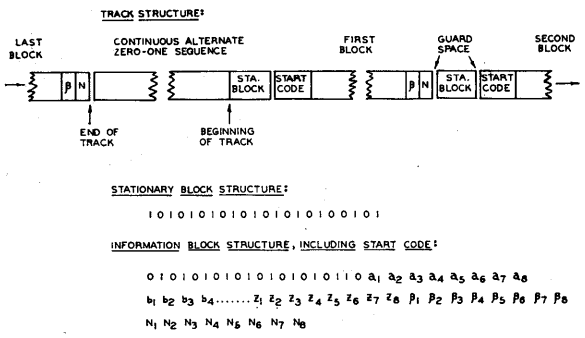


Fig. 14. Track structure on file drum as used in the DIANA system

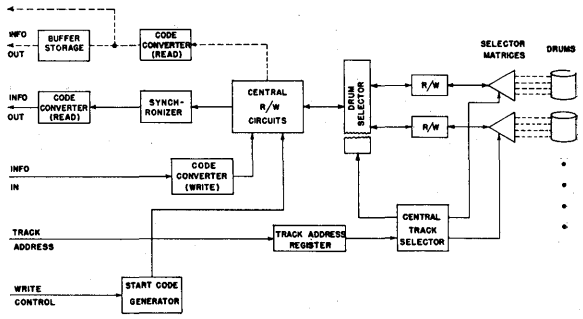


Fig. 15. Block diagram of file system

code (15 in practice), as shown in the logical diagram of Fig. 13. The start code search phase is initiated by a read-start pulse that sets *FF1*. Logical circuits operating on crossover pulses are then employed to produce counter reset pulses whenever the interval between successive pulses differs greatly ($5/4\ t_b < T < 3/4\ t_b$) from one bit time, t_b. In the region of three ones in Fig. 12, for example, the time between two successive P_{co} pulses is $T > 5/4\ t_b$. The last P_{co} before the long interval therefore sets *FF4* in Fig. 13 when it emerges from delay line *A* at the $3/4\ t_b$ tap. A new P_{co} has not yet occurred to reset *FF4* by the time the same P_{co} arrives at the $5/4\ t_b$ tap, so it passes gate *G10* and buffer *B3* to reset the counter.

The circuit in Fig. 13, consisting of *FF3* and *G9* similarly operates to reset the counter in the event of two distinct P_{co} pulses occurring less than $3/4\ t_b$ apart. This would occur, for example, in the middle of the five zeros region of the waveform of Fig. 12. The circuits to the left of delay line *A* act to reset the counter when a P_{co} is formed that is not connected with an alternation in the bit sequence, as between the first two of the five successive zeros in Fig. 12. The circuit accomplishes this by sensing for a reversal in the sign of the limited-information waveform, $\pm I$, after a P_{co} has occurred.

When the counter reaches the pre-established minimum count, it emits a pulse to reset *FF1* and the circuit enters the reading phase. Reading is accomplished by sensing the polarity of the limited information waveform in *G11* and *G12* with a sampling pulse P_s that emerges from delay line B. The P_s pulses initially are entirely due to P_{co} pulses, delayed by $6/4\ t_b$ to position them at the peaks of the information wave. This is true until the second of two ones occurring toward the end of the block start code, after which there is no P_{co} to use for the next information sampling. Note that information is being advanced at bit intervals from *FF5* to *FF6* by *G13* and *G14* at P_A time. A comparison of the nth and (n+1)st bits is continually being made for disparity by *G15* and *G16* at P_{co} time setting *FF7* and keeping *G1* open for delayed P_{co} pulses to enter delay line B. The situation of a double one is thereby saved because *FF7* remains reset from the P_s input, and allows P_A to pass *G2* and recirculate in the one-bit delay line, B.

The same regeneration of sampling pulses occurs in the case of the three one and five zero regions of Fig. 12, except that a longer period ensues before a fresh crossover phase is found to substitute for the recirculating pulse. The origin of sampling pulses is indicated schematically by arrows in Fig. 12. The last line of pulses corresponds to the output of *G11*.

It is seen that a double one heralds the end of the block-start code, where in addition, an extraneous zero occurs before information begins. The appropriate logical decisions necessary to reading can be derived from the three outputs at the bottom left of Fig. 13. It is also clear that there is merit in choosing a character code that minimizes, insofar as possible, the maximum number of like bits that can occur in sequence. This was accomplished in the application of the HD File Drum in the DIANA system by using an eight-bit alphanumeric code consisting of combinations of four zeros and four ones. By eliminating 2 of 70 possible codes the maximum number of successive ones or zeros is thereby reduced to 6 which is a practical interval over which to recirculate a self-clocked timing pulse.

By eliminating the two alternate one and zero codes, the maximum number of alternating bits is reduced to 11. Note that the eight-bit code has the potentiality of a redundancy check that exceeds somewhat the value of an ordinary parity bit.

Drum File System

Fig. 14 illustrates a track structure that was established for use in the DIANA system. In the initial formation of a track, alternating ones and zeros are written for something longer than one drum revolution time. A "stationary" block and the first information block, complete with start code, are then written in a continued fashion without losing the recording phase established by the alternating zeros and ones. Successive pairs of stationary and information blocks are written on subsequent drum revolutions where information concerning the time to commence writing is furnished by reading the previously written block and noting its termination. The structure of the stationary block is shown in Fig. 14 and it differs from the start code of an information block largely in that it ends soon after a double zero instead of a double one. The stationary block is so named because it is never rewritten unless the entire track following it is rewritten as well.

It will be noticed that a stationary block is placed between each information block. Stationary blocks are used in rewriting information blocks to indicate when the block to be rewritten is about to begin. Fig. 14 shows the last two characters of an information block to be comprised of a special β code indicating the end of a block, followed by a digit character indicating the number of the following block. The tagging of a block by the last character of a preceeding block is done to eliminate the necessity for reading any portion of a block prior to rewriting, since the reading waveform transient introduced by commencing to record within a block would render the rewritten block unreadable.

Stationary blocks are used in a completely self-clocked system to prevent the slow positional migration, after much rewriting, of blocks on the track. This would occur in the absence of stationary blocks since they would be positionally

dependent in being rewritten with random contributions from speed variations of the drum surface. Even using stationary blocks, a guard space must be left between blocks to allow for the case of initially writing a block at a time of minimum surface speed, and rewriting at a time of maximum surface speed. The guard space depends not only on the maximum fractional speed variations but also on the maximum block length that is allowed. A similar space allowance must be left between the last block on a track and the first.

It is the use of stationary blocks that enables the HD File Drum to be adapted to variable word and block length organization without wasted space, except for that consumed by stationary blocks. Note that while wholly variable block lengths are possible in the initial formation of a track, a block must not be increased in length thereafter without reforming the entire portion of the track that follows. A system using fixed block markers on an auxiliary track has been worked out, however, that dispenses with stationary blocks. The system is particularly valuable for fixed block organization.

Fig. 15 shows the major elements of a file system which uses HD File Drums. Output and input arrows at the left run to and from the controlling computer. Code converters are used to transfer between the balanced eight bit code of the file, and the system code if it is different. The output information is shown on two alternate lines. One method uses a buffer store that posesses the capacity of the maximum block length to be transferred and that is operable from either the system clock phase or the self-clocked phase depending on the operation being performed. The alternative method uses a smaller capacity synchronizer that accepts information at the self-clocked phase and passes it on at the system clock phase.

The size of the synchronizer in the latter case is dependent upon the allowed maximum variation in the phases of the two clocks, as well as on the maximum block length to be transferred. One form of such a synchronizer was previously described.[10]

References

1. Some Factors Influencing the Choice of a Medium for Magnetic Recording, **L. C. Holmes.** *Journal of the Accoustical Society*, New York, N. Y., May 1947.

2. Frequency Response of Magnetic Recorders, **O. Kornei.** *Electronics*, New York, N. Y., Aug. 1947, p. 124.

3. The Reproduction of Magnetically Recorded Signals, **R. L. Wallace.** *Bell System Technical Journal*, New York, N. Y., vol. 30, Oct. 1951, p. 1145.

4. The Reproduction of Signals Recorded on Magnetic Tape, **E. O. Daniel, P. E. Axon.** *Proceedings*, Institution of Electrical Engineers, London, England, vol. 100, pt. III, 1953, p. 157.

5. Magnetic Data Recording Theory, Head Design, **A. S. Hoagland.** *AIEE Transactions*, vol. 75, pt. I, 1956, p. 506.

6. Analysis and Lubrication of Bearings (book), **M. C. Shaw, F. Macks.** McGraw-Hill Book Company, Inc., New York, N. Y., 1949.

7. The Lubrication of Plane Sliders of Finite Width, **M. Muskat, F. Morgan, M. W. Meres.** *Journal of Applied Physics*, New York, N. Y., vol. 11, 1940, p. 208.

8. Theory for a Slider Bearing with a Convex Pad Surface; Side Flow Neglected, **S. Abramovitz.** *Journal*, Franklin Institute, Philadelphia, Pa., vol. 259, 1944, p. 221.

9. Ferromagnetism (book), **R. M. Bozorth.** D. Van Nostrand Company, Inc., New York, N. Y., 1951, p. 396.

10. Techniques for Increasing Storage Density of Magnetic Drum Systems, **H. W. Fuller, P. A. Husman, R. C. Kelner.** *Proceedings of Eastern Joint Computer Conference*, Special Publication T-70, Dec. 1954.

Transistorized Modular Power Supplies for Digital Computers

THEODORE C. GAMS
NONMEMBER AIEE

POWER supplies for digital computers are characterized by unusually rigid reliability requirements. After reliability, size and cost are the next most important considerations. Simplicity of design and ease of servicing are considered as part of the reliability problem, since computer down-time, particularly in large installations, may cost from $10.00 to $100.00 per minute. In military digital computers, reliability may be of incalculable value.

The design of high-reliability digital-computer power supplies is made practical by the relatively poor regulation performance that may be tolerated, of the order of $\pm 1\%$ to $\pm 10\%$, in contrast to analog computer and telemetering system power supplies, which require regulation performance on the order of $\pm 0.01\%$ to $\pm 0.1\%$.

In order to evaluate a power supply meaningfully in terms of regulation, however, it is necessary to construct a definition of "total regulation." The total regulation of a power supply includes the following factors:

1. Static load regulation.
2. Static line regulation.
3. Dynamic load regulation.
4. Dynamic line regulation.
5. Peak-to-peak ripple.
6. Thermal drift.
7. Interaction.
8. Long-term stability.

When total regulation is so expressed, the circuit designer may then assume that, under the worst possible simultaneous combination of all these factors, the power supply voltage can never fall below a predetermined minimum or above a predetermined maximum value. Fig. 1 readily illustrates the concept of total regulation.

The transistorization of digital computer circuitry has led to three unusual complications of the power supply design problem.

1. Transistor circuits demand very much lower voltages, and relatively higher currents, than vacuum-tube circuits performing the same functions. The resultant lower impedance level required of the power supplies, therefore, reduces the advantages gained by the lower total power requirements of transistorized equipment, and may actually increase the power supply size, weight, and cost over those formerly required. Table I contains a typical comparison example for reference.

2. Since transistorized circuitry is very much more compact than equivalent vacuum-tube circuitry, the power supply is often the largest element in the system, and can often be several times as large as the computing element it powers. This leads to attempts to miniaturize the power supply, often at the expense of reliability and economy.

3. The efficiency of the power supply is now of much greater concern than it was previously, since it is often responsible for more heat dissipation than all of the actual computer circuitry. This condition was seldom encountered before transistors and magnetic-core storage were adopted for digital computer use.

Theodore C. Gams is with the NJE Corporation, Kenilworth, N. J.

Table I. Comparison of Typical Digital Computer Power Supply Requirements: Vacuum-Tube Versus Transistor Circuitry

Vacuum-Tube Computer Circuitry				Transistorized Computer Circuitry			
Voltage Volts	Current Amperes	Approximate d-c Internal Impedance Ohms	Total Regulation ± %	Voltage Volts	Current Amperes	Approximate d-c Internal Impedance Ohms	Total Regulation ± %
+200	4.0	1.5	3	+30	12.0	0.075	3
+150	6.0	0.75	3	+12	24.0	0.015	3
+30	5.0	0.3	5	+6.0	12.0	0.025	5
+10	5.0	0.1	5	+2.0	10.0	0.02	10
−3.0	7.0	0.043	10	−2.5	6.5	0.038	10
−30	3.0	0.5	5	−6.0	7.0	0.043	5
−150	5.0	0.9	3	−18	4.0	0.135	3
Total Power = 2,761 watts (Heater Power = 2,200 watts)				Total Power = 870 watts (No Heater Power)			
Relative Power Ratio: 3.2:1 Relative Size Ratio: 2.2:1 Relative Cost Ratio: 1.1:1							

Table II. Comparison of Typical Digital Computer Power Supply Designs: Cost, Size, Weight, Versus Power-Line Frequency

Supply Under Consideration Has 16 Output Levels, Total Output of 6,200 Watts, Average d-c Impedance Level of 0.02 Ohms. Average Load Dynamic "Step" Is ±20%. Line Dynamic Step Is ±10%. Transistorized Circuitry Throughout

	Cost Dollars	Size Cubic Feet	Weight Pounds
At 60 cycles per second	11,000	40	2,000
At 400 cycles per second	6,900	19	800
At 800 cycles per second	6,600	17	700
At 2,000 cycles per second	6,700	15.5	610

Three Modern Power Supply Techniques

The general acceptance of germanium transistors as reliable devices, when properly derated and pretested, has permitted the use of efficient, compact, series-regulated and shunt-regulated power supplies for voltages as high as several hundred volts, currents as much as 200 amperes, and power levels up to and above 2 kilowatts (kw). Recent circuit improvements have realized complete freedom from overload and short-circuit failures. Fig. 2 illustrates a typical series-regulated transistorized power supply.

In many applications, particularly when a multiplicity of closely-spaced voltage levels must be furnished to a variety of loads, the transistorized shunt-regulator technique may be more attractive than the series-regulator technique. This configuration eliminates the necessity for protecting the transistors against short-circuit or overload stress, and permits operation from a common unregulated supply. Fig. 3 shows a typical shunt-regulated power supply, and Fig. 4 illustrates a typical array of several shunt regulators operating from a common unregulated supply.

When the regulation allowance is sufficiently broad, and/or the size allowance permits, brute-force circuitry still offers practical solutions to power supply requirements in the 5-to-500-volt region, particularly when line transients and line frequency variations are not severe. Modern ferro-resonant line regulators are available with greatly improved transient characteristics and "constant-average" rather than "constant rms" adjustment, which greatly facilitates brute-force design practice. Fig. 5 shows a typical brute-force supply.

Recent developments indicate that there is a promising future in certain digital computer power supply systems for the practice of "floating" nickel-cadmium batteries across the output of brute-force supplies in place of massive capacitor banks, to achieve low dynamic impedance, freedom from line-transisent regulation, and, under ideal conditions, the ability to complete a problem after total line failure.

When neither brute-force nor a pure transistorized technique is applicable, a third circuit has found favor. This employs the same kind of transistor correction amplifier employed in the pure transistor circuitry, but substitutes magnetic-amplifier gates for the power transistor series or shunt regulators. By designing the rectifier plate transformer appropriately, the short-circuit current can be limited to protect the rectifiers, permitting dependence on primary circuit breakers or fuses. Is some cases, protection may be omitted entirely.

Effect of Input Frequency—Motor Alternator

Freedom from severe line voltage fluctuations, and relief from the effects of momentary line failures, can be achieved by the introduction of a motor-alternator set of adequate rotational inertia between the power line and the power supply system. In addition, the power frequency may then be raised to

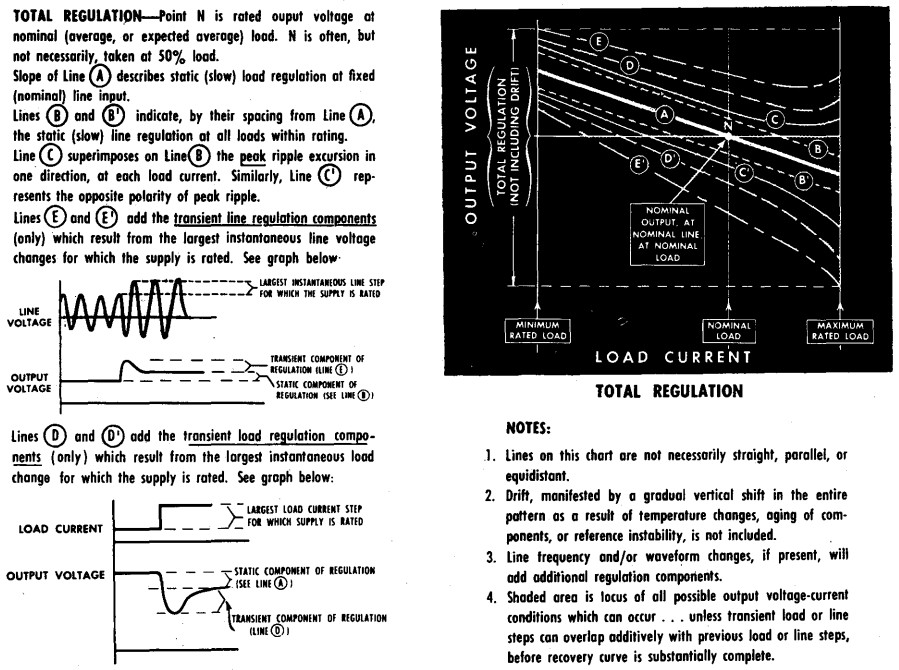


Fig. 1. Concept of total regulation

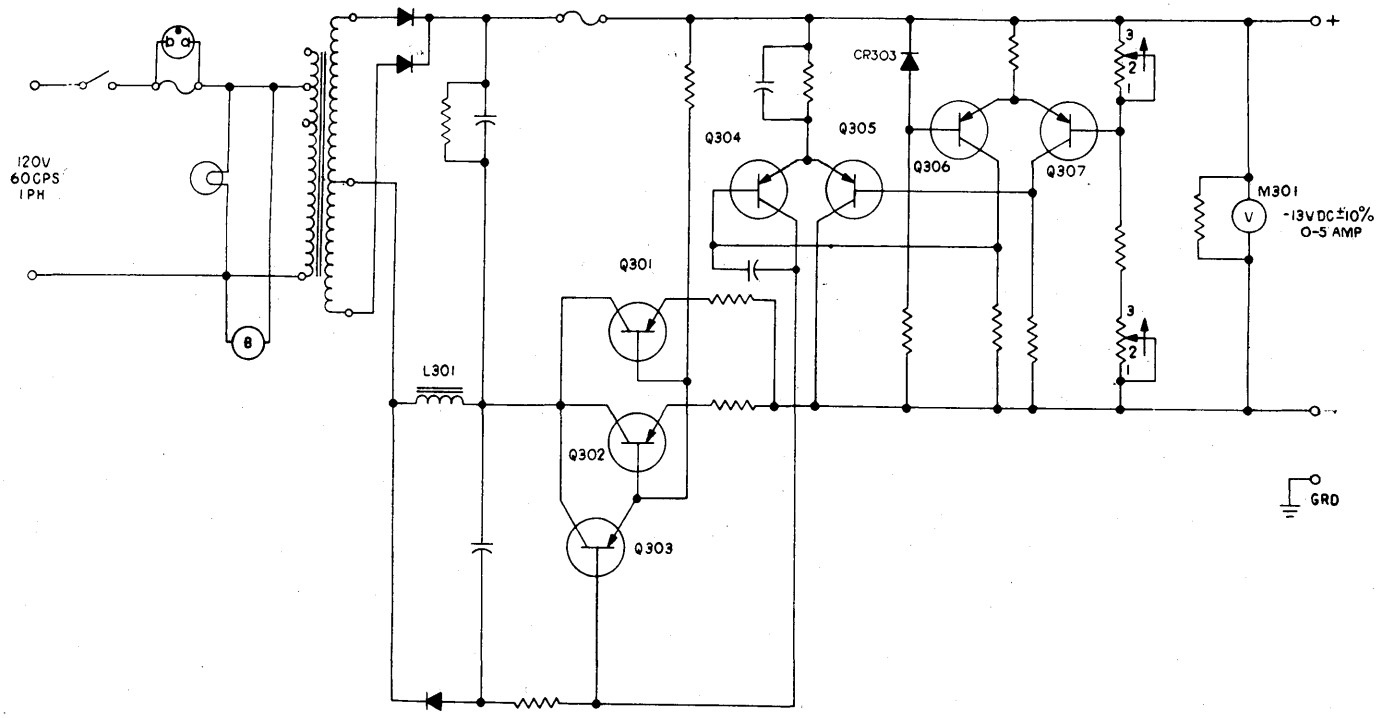


Fig. 2. Typical series-regulated transistorized power supply

400, 800, or even 2,000 cycles per second (cps), greatly reducing the size and cost of the rectifier, filter, and regulator elements. Modern motor-alternators do not require the elaborate maintenance techniques normally considered necessary for older machines, and the acoustical problem has been successfully solved. Over 5 kw, the motor alternator almost always pays for itself in reduced power supply cost, and should be considered for its many other advantages. Modern tubeless voltage-regulating circuits are available to hold the alternator frequency and output voltage within close limits, over wide ranges of load and input voltage, and the response characteristics of these regulators is often better than the response of static line regulators of comparable power capacity. Table II illustrates some of the advantages of motor-alternator applications.

Modulization

Analysis of almost any set of required voltage and current levels for a computer load will suggest to the designer that certain basic circuits and groups of components are repeated with little or no variation from design to design.

It will further be noticed that the number of unique designs can be reduced by producing flexible modules, each covering a number of different desired voltage levels, at current ratings and with regulation characteristics which will satisfy all the requirements of each voltage level.

The individual ranges may be made manually selectable by taps or switches, or automatically selectable by proper wiring of the socket into which a module plugs. Ranges may be extended by adding additional pluggable (series or shunt) regulator elements.

The process of selecting the minimum number of economical, reliable, easily

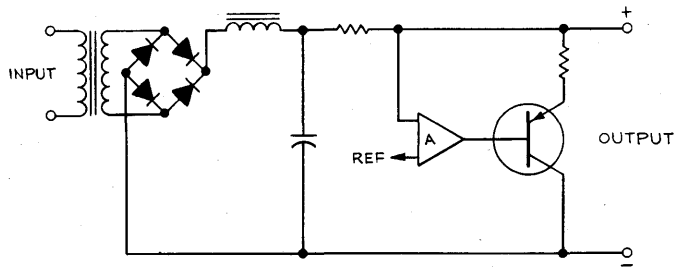


Fig. 3 (left). Typical shunt-regulated power supply

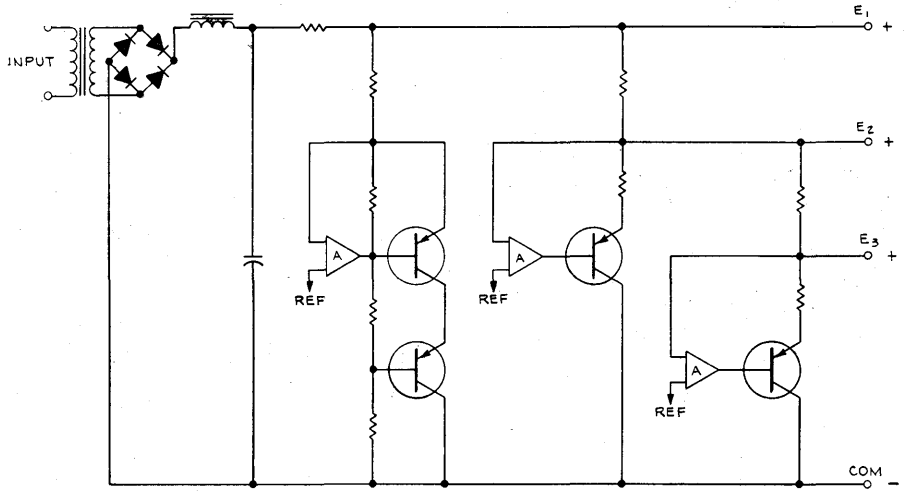


Fig. 4. Typical array of several shunt regulators operating from a common unregulated supply

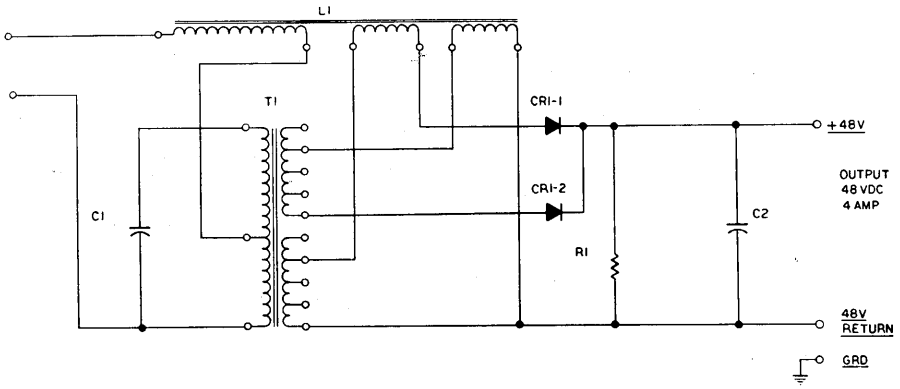


Fig. 5. Typical brute force power supply

maintained modules to cover a set of required outputs involves a very delicate balancing of many factors. It can be influenced by as abstruse a consideration as the cost of preparing elaborate instruction-manual literature on the smallest number of circuit configurations.

In most regulated systems, it is practical to design one universal correction amplifier, incorporating a suitable reference power supply, as a standardized plug-in submodule for all regulating circuits in the system. Another common plug-in sub-module is a power transistor array mounted on a pluggable heat-sink. A typical "universal" amplifier is illustrated in Fig. 6. In this instance, it was considered desirable to mount the amplifier power supply, as well as a universal-output control circuit, on a separate (pluggable) chassis, shown in Fig. 7. This chassis provides the choice of locally or remotely controllable marginal-checking facilities, as well as remote sensing, etc.

Through the use of modern modulization techniques, one recent large transistorized computer power supply system provided more than 150 outputs from only 11 basic designs. Analysis of the expected component life of that system indicates that 85% of all anticipated failures can be corrected by substitution of one or more of only three of the 13 submodules.

Distribution, Circuit Blocks, Load Protection

Certain precautions (e.g., protection against reversal of voltage due to inadvertent short-circuitry of a positive output lead to a negative output lead), are necessary in connecting the power supply system to the complex load geometry common in modern high-speed digital computers. If these precautions are not observed, catastrophic failures of thousands of transistors, diodes, or core assemblies can occur.

Further, the low internal impedance that has been painstakingly designed into the power supplies can be multiplied many times by the distribution system. Transmission lines having characteristic impedances of the order of 100 micro-ohms are not uncommonly required.

Intolerable interaction can occur if the distribution system is not properly designed

If the distribution problems become too difficult to handle with large "central" power supplies, a common solution is found in the (noncritical) distribution of unregulated d-c power, with smaller modulized regulators located at each of the several racks of load equipment. In this instance, a system such as that outlined in Fig. 4 is often preferable.

Proposed Definitions of Computer Power Supply Electrical Parameters

These definitions were prepared by the NJE Corporation for consideration by the National Electrical Manufacturers Association Subcommittee on Computer Power Supplies, Semiconductor Section.

Regulation

The regulation of a power supply is defined as the algebraic sum of all the components listed, measured individually, with all uninvolved parameters held at nominal. For convenience, a "norm value" of output voltage is selected at nominal input conditions, with all other supplies in the system inactive, at a nominal load current, arbitrarily selected as the average of the maximum and minimum load rating, at nominal ambient temperature. Positive-going voltage variations from the norm are considered as positive regulation components, and negative-going voltage variations from the norm are considered as negative regulation components. The total regulation is then stated as "plus or minus" a given percentage, and it is understood that the arithmetic sum of all positive regulation components must be less than the given percentage, and the arithmetic sum of all negative regulation components must be less than the given percentage. Definitions of the individual components follow:

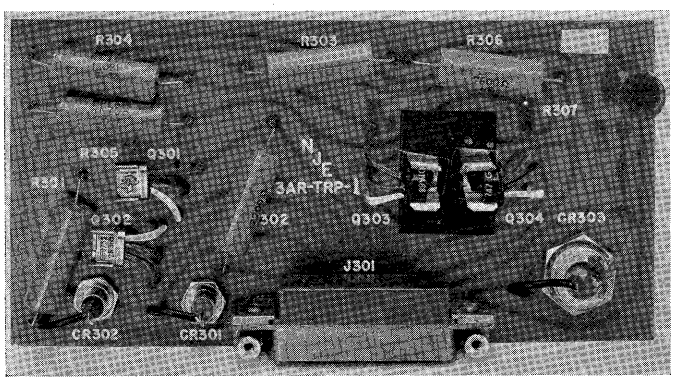


Fig. 6. Typical "universal" amplifier

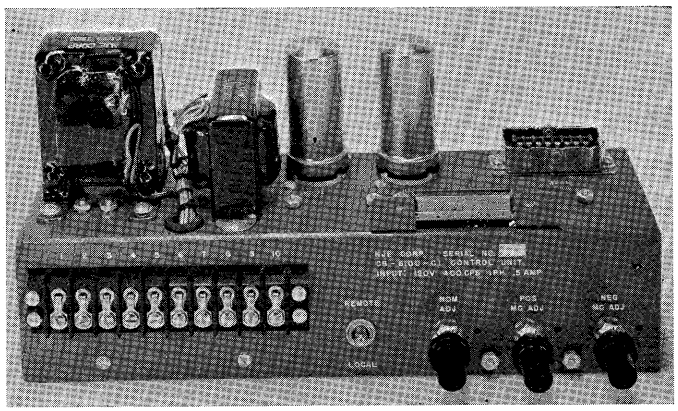


Fig. 7. This control chassis provides the amplifier and reference power for "universal" amplifier of Fig. 6, as well as remote and local marginal checking facilities

1. *Static Load Regulation* is the variation from norm of the output voltage as the load current is varied slowly from a rated minimum to rated maximum, expressed as a percentage of the norm.

2. *Static Line Regulation* is the variation from norm of the output voltage, as the line supply voltage (in 3-phase systems, line-to-line, all three phases varied simultaneously with no significant phase-to-phase unbalance) is slowly varied from rated minimum to rated maximum, expressed as a percentage of the norm.

3. *Peak-to-Peak Ripple* is the maximum periodically-occurring excursion of the output voltage under a passive (resistive) unvarying rated maximum load, expressed as a percentage of the norm. To avoid confusion with induced noise (from sources external to the power supply) the periodicity referred to above is restricted to those frequencies which are integral harmonic multiples of the line frequency.

4. *Dynamic Load Regulation* is the difference between the maximum instantaneous excursion of the output voltage in response to a unit-step change of output current (indefinitely maintained) between the rated minimum value and the rated maximum value and the static excursion which remains after all transient recovery behavior has effectively disappeared, expressed as a percentage of the norm. Unit-step changes are taken in both directions (minimum to maximum and maximum to minimum) to obtain both positive and negative components of dynamic load regulation.

5. *Dynamic Line Regulation* is the difference between the maximum instantaneous excursion of the output voltage in response to a unit-step change of line voltage current, indefinitely maintained, between the rated minimum value and the rated maximum value and the static excursion which remains after all transient recovery behavior has effectively disappeared, expressed as a percentage of the norm. Unit-step changes are taken in both directions, minimum to maximum and maximum to minimum, to obtain both positive and negative components of dynamic line regulation.

6. *Instability* is the variation in the output voltage which occurs under "norm" conditions over a period of _ _ consecutive days, not including at least _ _ minutes of warmup time under norm conditions, expressed as a percentage of norm.

7. *Thermal Drift* is the variation in the output voltage which occurs under "norm" conditions as the ambient temperature, measured at least 4 inches from the coolest surface of the power supply, is slowly varied from rated minimum to rated maximum, expressed as a percentage of norm.

8. *Interaction Regulation* is the variation in output voltage of any particular supply of a system of supplies, under "norm" conditions, as all the other supplies normally connected to the same power source, motor-alternator or line, simultaneously experience a unit-step change in load current from rated minimum to rated maximum, expressed as a percentage of the norm. As previously, unit-step changes are taken in both directions.

The Shiftrix-Machine Organization for High-Speed Digital Computation

GERALD ESTRIN
NONMEMBER AIEE

THE NEED for higher speed computation demands continuous scrutiny of machine organization concurrent with the search for higher speed switching and storage elements.

Every digital computer operation may be decomposed into a set of micro-operations whose decision points are conditional upon the information in the operands and instructions. Between the extremes of strictly serial and extravagantly parallel methods, the machine designer is guided by some combination of formal synthesis, intuition, and learned weighting factors.

This paper focusses its attention on an arithmetic organization which utilizes a shifting matrix between an operand and the accumulator. Part I considers the multiplication process. Part II describes the use of the "Shiftrix" in other machine operations. Part III discusses possible mechanizations.

A block diagram representation of a conventional parallel multiplier is shown in Fig. 1.

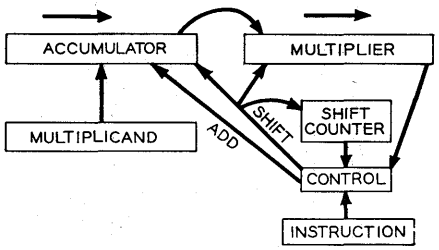


Fig. 1. Conventional parallel multiplier organization

In a binary multiplication involving two positive n-bit operands, the least significant bit of the multiplier is observed; the multiplicand is added to the existing partial product if the multiplier bit is a "one"; and the resulting partial product is shifted one place to the right, independent of the state of the multiplier bit. The multiplier is shifted right at the same time permitting the partial product bit to enter the left end of the multiplier register and bringing the next multiplier bit into control position. This process is iterated and the counter tallies the number of shifts. The control stops the iteration process when a preset count is reached.

In multipliers using a fixed operation time an addition of zero or the multiplicand occurs at every step. The multiplication process, exclusive of memory access time and negative operand corrections, requires a time interval

$$t_x = n(t_a + t_s) \qquad (1)$$

where

t_a = the time interval required to perform an addition
t_s = the time interval required to shift the partial product and multiplier

Multipliers using a variable operation time conventionally expend the addition time only when required by the state of the multiplier digit. The resulting operation time is

---

GERALD ESTRIN is with the University of California, Los Angeles, Calif.

The preparation of this paper was sponsored in part by the Office of Naval Research.

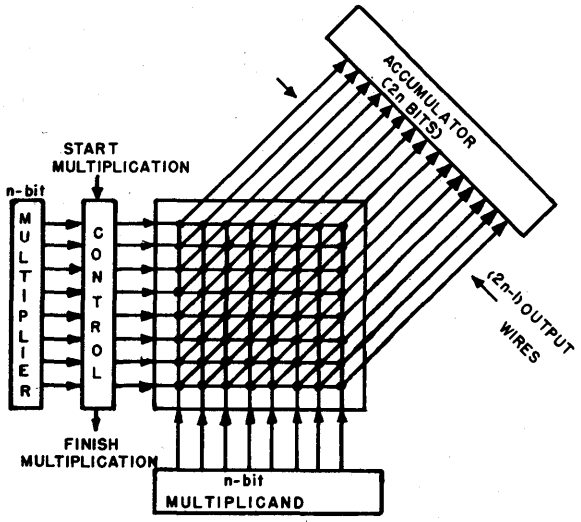


Fig. 2 (left). Shiftrix multiplier

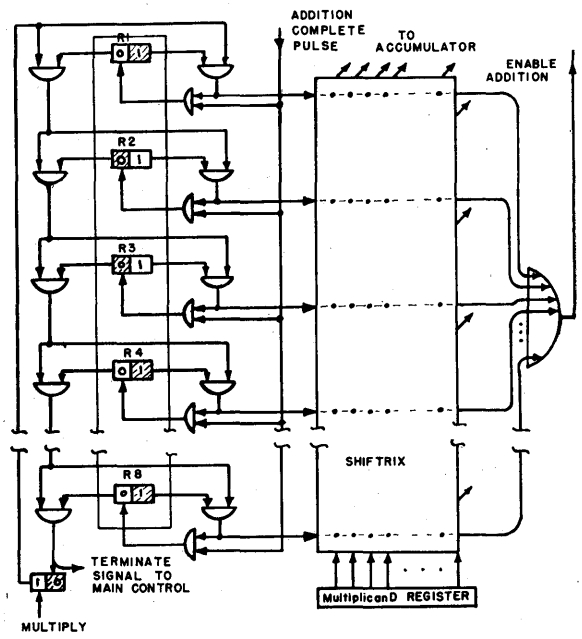


Fig. 3 (right). Shiftrix multiplier control

$$t_x = n_m t_a + n t_s \tag{2}$$

where n_m = the number of nonzero bits in the multiplier. The longest and shortest multiplication operation times are the following:

$$t_{xl} = n(t_a + t_s) \tag{3}$$

$$t_{xs} = n t_s \tag{4}$$

The average multiplier contains an equal number of ones and zeros requiring

$$\bar{t}_x = \frac{n}{2} t_a + n t_s \tag{5}$$

The time required for each addition may be reduced by parallel initiation of carries and parallel sensing of their completion;[1] by parallel propagation of carries,[2-4] by elimination of carry propagation as a result of separate storage[5-7] of the carry states until a final extra step. Both the maximum and the average number of additions may be reduced by appropriate transformation of the multiplier to minimize the number of nonzero bits.[7-11] The number of additions and the number of shifts may be reduced by forming multiples of the multiplicand and using a number base greater than two with corresponding multiple shifts of the partial product.[4,7]

Part I. The Shiftrix

The configuration of Fig. 2 readily illustrates a multiple place shifting structure.

The switching matrix or shiftrix, S, receives the digits of the multiplicand on the vertical set of inputs at the start of multiplication operation. The horizontal set of inputs is derived from the multiplier. The $2n-1$ outputs enter the double-length accumulator. The sequence in which the horizontal inputs are energized is determined by the box marked "control."

One form of the control is logically detailed in Fig. 3. The multiplier, R, is represented in the vertical column of flip-flops. The multiplicand, D, acts on the vertical inputs to the shiftrix and therefore, relative to the accumulator is preassembled in all of the shifted positions. For clarity, a particular multiplier, $x = 0.1001110110$, is discussed. The operation is initiated by enabling the input gate at the top. Since the most significant bit of the multiplier is a "1", matrix driver number one is enabled, energizing the top row and adding $D/2$ into the accumulator. The accumulator produces a pulse, announcing completion of its task. That pulse combined with the driven state of the top row resets the most significant multiplier flip-flop to "0". As a result the top row of the shiftrix is disabled and the enabling level is permitted to proceed down the column of multiplier flip-flops. The enabling level propagates past the second and third flip-flops since they are in the zero state. However, it is stopped by the fourth flip-flop, enabling the driver of the fourth row and causing $D/16$ to be added into the accumulator. The process continues until the enabling signal emerges at the bottom of the column announcing the termination of the multiplication operation and setting the termination flip-flop which in turn disables the input gate at the top of the column.

Transformation of the Multiplier

It is evident from the foregoing description that any reduction in the number of nonzero bits in the multiplier directly reduces the operation time. Such transformations have been proposed[7-11] for other arithmetic organ structures but are particularly effective with the shiftrix organization.

Lehman's[10] formulation of the transformation produces a ternary coding of the multiplier with a resultant minimization of the number of nonzero bits in the transformed multiplier. It is expressed by

$$2^{-n}\sum_{0}^{m} b_t 2^t = 2^{-n}\sum_{0}^{m+1} (-1)^{s_t} c_t 2^t \tag{6}$$

where

b_t = the original binary coefficients and
$c_t = [b_t \oplus b_{t-1}] \cdot \bar{c}_{t-1}$ is a binary variable determining the magnitude of the ternary coefficient
$s_t = b_{t+1}$ is a binary variable which determines the sign of the ternary coefficient

The transformation defined in equation 6 requires a propagation of the c_t from the least to the most significant end; the decision to subtract or not is governed by $s_t = b_{t+1}$; the decision to execute an arithmetic operation at all is governed by c_t. If it is found undesirable to expend the c_t propagation time implied in equation 6 a less optimum transformation is given by equation 7.

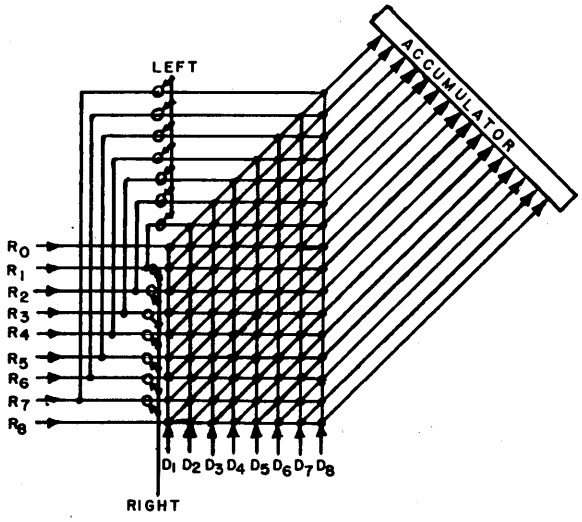


Fig. 4 (left). Left or right shiftrix

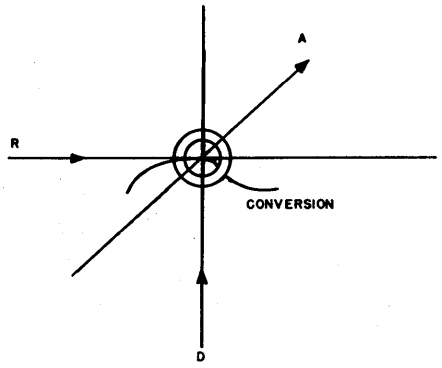


Fig. 6 (right). Magnetic core intersection

$$c_t=\bar{b}_t b_{t-1} b_{t-2}+b_t \bar{b}_{t-1} \qquad (7)$$

$$s_t=b_{t+1} b_t$$

Maximum, minimum and average non-zero bits produced by equation 7 are

$$N(n)_{\max}=\frac{2n+3}{3}$$

$$\overline{N(n)}=\frac{3n-2}{8} \qquad (8)$$

$$N(n)_{\min}=0$$

Using the conventional multiplier organization enhanced by the ternary coding the maximum minimum and average operation times become

$$t_{x3}=\frac{n+2}{2}\,t_A+nt_s$$

$$t_{xs}=nt_s \qquad (9)$$

$$\bar{t}_x=\frac{n}{3}\,t_A+nt_s$$

Since two arithmetic operations may not occur in sequence, facilities for two place shifting of the partial product and transformed multiplier lead to

$$t_{x1}=\frac{n+2}{2}\,t_A+\frac{n}{2}\,t_s$$

$$t_{xs}=\frac{n}{2}\,t_s \qquad (10)$$

$$\bar{t}_x=\frac{n}{3}\,t_A+\frac{n}{2}\,t_s$$

If the ternary coded multiplier is applied to the shiftrix structure the following is obtained

$$t_{x1}=\frac{n+2}{2}\,t_A+{}_p$$

$$t_{xs}=t_p \qquad (11)$$

$$\bar{t}_x=\frac{n}{3}\,t_A+t_p$$

where

t_p = the time required for the enabling signal to propagate down the multiplier control chain

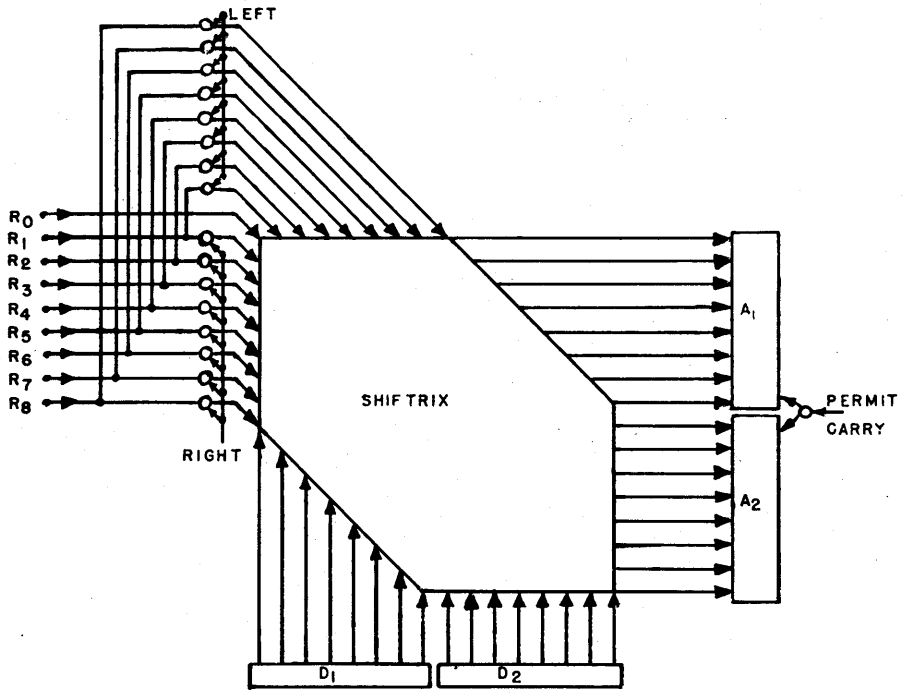


Fig. 5. Double length shiftrix

Part II. Other Machine Operations

Figs. 4 and 5 show expanded forms of the shiftrix-providing facilities for shifting in both directions. Fig. 5 permits operation on a double-length word. The following descriptions refer to the larger matrix of Fig. 5.

Shifting

The number to be shifted is brought into position to drive the vertical lines of the matrix. The direction of shift is established on the busses labelled "Left" and "Right." If the number of shifts is stated in binary representation, a binary to base n decoding before insertion in the R register permits a one step shift. If the binary number is inserted directly into the R register, a number of binary power shifts alternating with transfers from the accumulator register to the D register must be sequenced.

In the first case the operation is executed simply by scanning the R register and energizing the appropriate row of the matrix. No feedback is provided to reset the R flip-flop, and the operation terminates when the signal is obtained announcing delivery of the shifted number to the accumulator.

If combined with some extract logic, rather complicated decomposition and composition of "packed" words may be easily accomplished. This is particularly significant with the previously described multiplication logic in which there is little penalty for short word multiplications.

Normalization

The number to be normalized is brought into both the R and D registers. The direction bus is set to "Left." The R register is scanned and, if sign magnitude representation is used, the first "one" energizes the appropriate row of the matrix. An all-zero quantity is signified by the emergence of the scanning level at the bottom of the R register control. The number may be introduced in the R register in such fashion as to limit the magnitude of the normalized quantity to either $<1/2$ or <1.

Division

There are several modes of division which may be tailored to the shiftrix structure. Among these are:

Mode 1: Divisor>Dividend

The dividend is brought to the accumulator
The divisor is put into D
All "ones" are inserted in R
R is scanned and a difference is formed at each step. Any of the restoring or nonrestoring schemes may be executed. The excited row defines the quotient digit to be determined and the polarity of the remainder establishes the value of the quotient digit. The operation terminates when the scanning signal emerges.

Mode 2 Divisor>Dividend

The divisor is normalized
The dividend is shifted left by the same amount as the divisor
At each step the remainder is inserted in R.
An algorithm is defined which guarantees the reduction of the significance of the most significant remainder digit at every step. The R register scanner always operates on a row corresponding to the most significant digit of the remainder. In this way leading sequences of zero bits in the remainder are ignored.

Mode 3

The divisor and the dividend are normalized with proper correction of exponents. The division process is carried out in the same fashion as Mode 2.

Conversion

Binary Coded Decimal to Binary Conversion

The matrix is forced into a state such that it contains the binary equivalents of appropriate decimal powers in each row.

The binary coded decimal, (BCD) quantity is inserted in R. The operation then proceeds as in a normal multiplication.

The binary result will be in the accumulator.

Binary to BCD Conversion

The matrix is forced into a state such that it contains the binary equivalents of the appropriate decimal powers in each row.

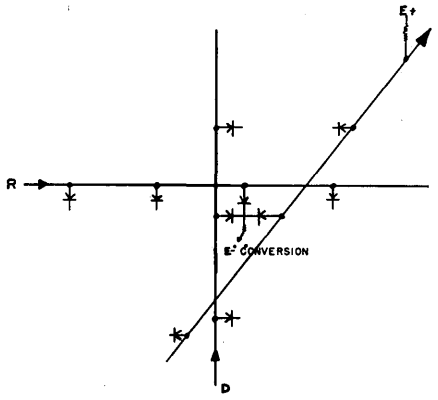


Fig. 7. Diode intersection

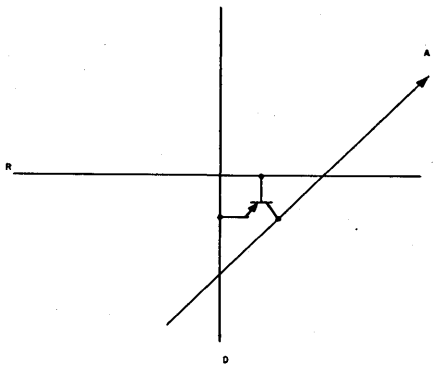


Fig. 8. Transistor intersection

The binary quantity is inserted in A.
All "ones" are inserted in R.
The division proceeds as in Mode 1 with a modified restoring process.
The BCD result goes to a quotient register.

Floating Point Operations, Multiplication and Division

These operations are executed in the same fashion as the previously described fixed point mode except for the independent addition or subtraction of the exponents. Normalization may be automatic or a programmed operation. Note that it is reasonable to define a conditional normalizing instruction in which normalization would take place only if the number of leading zeros exceeds a preset criterion. The scanning character of the R-register control facilitates such an operation.

Addition and Subtraction

These are the more complex floating point operations and make use of the shiftrix and the double-length accumulator to reduce the number of steps.

The exponents are handled independently of the mantissae. The comparison of the two exponents determines which mantissa is to be shifted before summation. The operands are entered into both the accumulator and D registers in a manner such that either mantissa in the D register may be shifted relative to the other mantissa in the accumulator, conditional on the sign of the exponent difference. The magnitude of the exponent difference is entered into the R-register via decoding logic (as described for the simple shift operations) and determines the amount of shift. Conditional upon the sign of the exponent difference, the result may appear in either half of the accumulator and, therefore, require a final half-accumulator shift.

Part III. Mechanization of the Shiftrix

The choice of an elementary component for use within the matrix structure depends upon the properties of the full system within which it acts. The primary purpose of the added set of components is the acceleration of the sequence of operations in any significant computation. The power of the shiftrix lies in the enhanced ability of the computer to exploit the variations in the numbers being processed rather than penalizing every operation with the time required by a worse case. The shiftrix structure is, however, wasteful if the access time for a quantity within it takes as long as an average number of shifts would if they were done sequentially in the accumulator.

If magnetic memory elements are used at the intersections, they are set initially to represent the state of one of the operands and then read sequentially in rows conditional on the other operand. In both cases there are no coincident current considerations to limit the amount of drive current used and they may therefore be switched rapidly. The necessary accessing of operands, scanning of the R-register and the arithmetic operation concommitent with the selection of any row aid in the definition of a maximum duty cycle.

The conversion technique described in Part II may be accomplished by threading a winding through the cores in such fashion that the binary equivalents of the decimal powers may inserted into the matrix in a single step.

If a diode matrix is chosen, a cluster of three diodes may be used at each intersection as in Fig. 6. The row and column biases are arranged such that a column driver must supply the load current of only a single intersection. The row driver must be able to supply the sum of the intersection load current in its row.

When the state of the D-register is

initially established on the column lines, each column driver looks into a load consisting of the parallel back resistances of the set of column diodes. When the scanning of the *R*-register proceeds, the energized row driver reverse biases all of its row diodes and causes each intersection current to switch either into the output line or into the column line depending upon the state of the *D*-register digit.

The conversion technique with the diode matrix may be accomplished by operating on the common return of the set of intersection resistors such that the binary equivalents of the decimal powers are forced in each row.

For application to those situations where speed is crucial and the number of intersections is not too large, Fig. 7 displays a typical intersection utilizing a transistor switching element. (Diodes may be required to separate groups of collectors which converge on each output line).

In this configuration each vertical line driver must supply a single intersection current and the row drivers must control the set of bases. The conversion technique is most readily accomplished in this case by breaking each row into two pieces such that only those intersections representing the binary equivalents of the decimal powers are enabled. In this case the column drivers would be forced into the enabled state.

Conclusion

This paper has presented some of the results of investigations into a machine organization based on a large switching matrix. This form has not received very much attention in the past because of the implied large number of components.

The expanding application of digital techniques continually unearths problems which are beyond their present power. The most striking and gratifying solutions of these problems come from their creative reformulation. However a component with that capability is not yet mass produced. Therefore digital technology requires concentration in three areas which may permit the solution of larger problems. These are developments of higher speed components, reorganization of existing components to permit faster basic operations, and reorganization of the control of the various parts of a computer to enhance the efficiency of sequences of operations.

The switching matrix has many intersections. However, due to its regular structure, it becomes amenable to automatic production techniques which may lead to lower cost. It has been considered in a restricted type of operation in this paper. In the more general case one would like independent insertion into each of its intersections and distribution to various parts of a machine under control of a second set of inputs such that complex subroutines might be executed.

In a less futuristic sense the switching matrix is of interest because it may be used to delay the day when a given machine is declared obsolete. It can directly speed up the arithmetic operations. If that is accomplished sufficiently, it may become efficient to utilize modern memory components and achieve further increase in speed.

In still another sense, if the speed of a new machine is prescribed it may be desirable, as a result of other considerations, to use a slower basic component enhanced by the parallel organizing attributes of the shifting matrix. In this way for example the area of application of magnetic computers may be enlarged.

There are certainly a number of problems to be evaluated relative to the switching matrix organization. There are however enough positive indications to induce experimental evaluation of existing components and alterness to new physical phenomena which may suit the described structure.

References

1. Fast Carry Logic for Digital Computers, **B. Gilchrist, J. H. Pomerene, S. Y. Wong.** *Transactions*, Professional Group on Electronic Computers, Institute of Radio Engineers, New York, N. Y., vol. EC-4, Dec. 1955, pp. 133–36.

2. *Internal Report.* Electronic Computer Project, Institute for Advanced Study, Princeton, N. J., 1951.

3. A One-Microsecond Adder Using One-Megacycle Circuitry, **A. Weinberger, J. Smith.** *Transactions*, Professional Group on Electronic Computers, Institute of Radio Engineers, vol. EC-5, no. 2, June 1956, pp. 65–73.

4. Faster, Faster (book), **W. J. Eckert, R. Jones.** McGraw-Hill Book Company, Inc., New York, N. Y., 1955.

5. Preliminary Discussion of the Logical Design of an Electronic Computing Instrument (book), **A. W. Burks, H. H. Goldstine, J. von Neumann.** Institute for Advanced Study, Princeton, N. J., pt. I, second edition, 1947.

6. A Note on High-Speed Multiplication, **G. Estrin, B. Gilchrist, J. H. Pomerene.** *Transactions*, Professional Group on Electronic Computers, Institute of Radio Engineers, vol. EC-5, no. 3, Sept. 1956, p. 140.

7. On the Design of a Very-High Speed Computer. *Report No. 80*, University of Illinois Digital Computer Laboratory, Urbana, Ill., Oct. 1957.

8. Automatic Digital Calculators (book), **A. D. Booth, K. H. V. Booth.** Academic Press, Inc., New York, N. Y., 1953.

9. On Performing Binary Multiplication with the Fewest Possible Additions, **G. W. Reitwiesner** (*B.R.L. Technical Note 1113*, Feb. 1957). 12th National Meeting, Association for Computing Machinery, Houston, Texas, June 1957.

10. High-Speed Digital Multiplication, **M. Lehman.** *Transactions*, Professional Group on Electronic Computers, Institute of Radio Engineers, vol. EC-6, no. 3, Sept. 1957, pp. 204–05.

11. A High-Speed Electronic Arithmetic Unit for Automatic Computing Machines, **M. Nadler.** *Acta Technica*, Czechoslovak Academy of Science, Prague, Czechoslovakia, vol. 1, no. 6, 1956.

A Device to Facilitate Combined Analog-Digital Computation

B. L. SCHWARTZ
NONMEMBER AIEE

G. JENKINSON
NONMEMBER AIEE

L. WINSLOW
NONMEMBER AIEE

B. GORDON
NONMEMBER AIEE

J. SOLOMON
NONMEMBER AIEE

MANY computing problems present characteristics which appear to call for both digital and analog-type computers. To the operator of a small computing facility, such problems present a dilemma, for the analog-digital conversion equipment available is generally expensive, highly specialized, or both. This paper describes a device developed at Battelle Memorial Institute which can resolve this problem for a certain (admittedly rather limited) class of problems. Some applications are given.

The class of problems covered is that which admits complete simulation within the capacity of the available analog computer, but whose results must be processed subsequently on a digital computer. That is, the values of the traces of the analog output during the course of the run, not just a final single number, must be processed digitally.

The Data Collection Device

Need for the Device

It is common knowledge that there exist classes of problems which lend themselves to combined analog-digital computation of the sort just described. It follows that a method for communication between the two types of computers is desirable.

However, since these problems are rather special in nature, their occurrence is not too frequent in a medium to small size computing installation. This means that any special equipment developed for use in these problems should be sufficiently general in nature that it could fill other data-collection needs. Investigation into other possible applications indicated several areas where a flexible device for data collection could be efficiently employed.

Choice of the Design

After some consideration, the picture of this device became clear. It should be a very flexible sequencing device which was portable and could conveniently utilize several types of data-recording components which are already available.

This conclusion led directly to the design of a programmer. The block diagram appears in Fig. 1, and a general view of the actual machine in Fig. 2. It contains no permanent circuitry. All electrical circuits must be externally connected on the control panel, and are thus completely flexible, and there is plenty of space for addition of the components.

Programmer Description

The Control Panel. The control panel is the heart of the device, and is responsible for a large share of the total cost. It is a standard unit with 816 pluggable positions, and an assorted selection of jumper wires. By use of spare plugboards, the programmer can be quickly converted from one use to another simply by exchanging the plugboards, and changing the connectors to external measuring and recording instruments. The location of connections to the internal components are shown in Fig. 3.

Stepping Switches. The "program steps" connections provide access to three levels of a 4-level 22-position stepping switch. The other level is connected to neon indicators which are on the front panel for convenience in observing operation during panel wiring.

There are three single-level 10-position stepping switches for counting or selection purposes. There is also a 2-level 10-position stepping switch which is usually used for selection purposes.

Relays. For logic operations, there are 20 relays. Ten of these relays have two transfer points, and the other ten have two transfer points and one normally open point. In addition, the second group have indicator lights for use in checking the logic of panel wiring.

Diodes. There are ten diodes for isolation when the same relay is being actuated from more than one possible source.

Toggle Switches. Five double-pole double-throw toggle switches provide flexibility and on-off start-stop control.

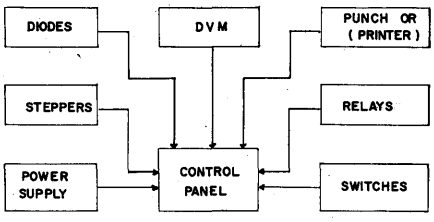


Fig. 1. Block diagram of programmer

External Circuit Connectors. The programmer presently has three external circuit connectors: one for an International Business Machines Corporation (IBM) type *16*, *26*, or *526* punch, one for a digital voltmeter, and one 20-pin connector for any miscellaneous circuitry required in interconnections.

Power Supply. A 48-volt d-c power supply provides the necessary power to operate the relays and stepping switches and provides a source for external signals when required.

Busses. Eleven 5-hub rows of common connections are provided to eliminate the necessity of using split wires for common connections.

Indicator Lights. Twenty-two neon indicators provide visual access to show which step the program stepper is on. Also, as previously mentioned, ten of the relays have neon indicators.

Mode of Operation

When operated as an analog-digital converter, the action of the device would be as follows: The analog computer would be wired for computation as usual. Certain additional connections which are indicated would be made from amplifier outputs to the external connector of the programmer, but these would in no way affect the problem computation wiring, nor would they affect the computed results. In addition, a timing integrator in the computer would be wired to a relay amplifier, with the gains so adjusted that the relay would operate at the time interval at which digital readout was desired. This relay amplifier would control the computer "hold" circuit, and transfer system control to the programmer. When the analog computer was manually switched into "operate" mode, computation would commence, and simultaneously the timing integrator would begin to build up a voltage. When this voltage reached the preset level and caused the relay to transfer, the computer

B. L. Schwartz, G. Jenkinson, L. Winslow, B. Gordon, and J. Solomon are with Battelle Memorial Institute, Columbus, Ohio.

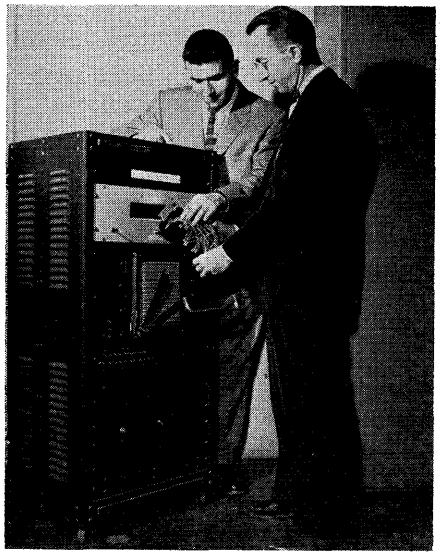


Fig. 2 (left). General view of programmer

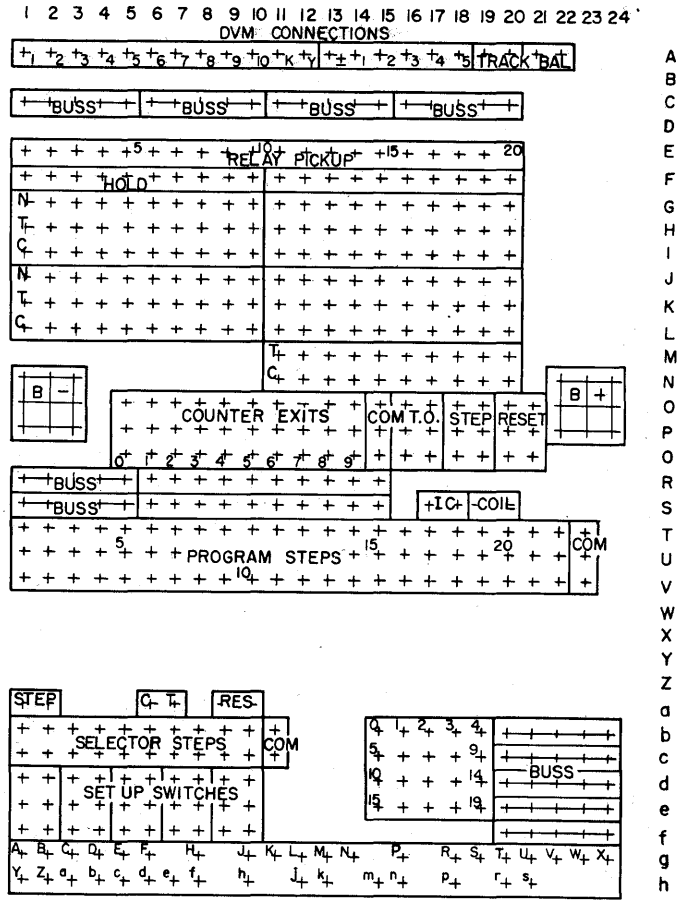


Fig. 3 (right). Front view of control panel

would go into "hold" and the programmer would take command.

The programmer would then connect the first voltage to be digitized from the plugboard to the digital voltmeter. Balancing of the voltmeter would produce a pulse to advance the program stepper switch to the next position; the voltmeter reading would then be connected to the card punch cable. Completion of the punching operation would again make a pulse available from the punch for advancing the stepper. Each voltage of interest would in turn be punched, in like manner, from its connection on the programmer plugboard through the connector. Finally, the program stepper would switch a small resistor across the capacitor of the timing integrator, causing the voltage to decay rapidly to effectively zero. With the timer so reset, the programmer would relinquish control of the computer, and the programmer would reset itself to be ready for the next cycle. The analog would then proceed with the problem solution until the timing integrator again built up to the preset voltage, at which time the punch cycle would repeat.

Special Considerations

Since the computer had been in "hold" during the punch operation, the computation would continue rather than reset. A great deal of experience with the system shows that this presumption is quite valid. Many problems have been run with hundreds of interruptions for punching and the results compared with the same problem run without interruption. Results agree to within the normal reproducible ability of any analog computer calculation.

If it were desired to control punching on the basis of some other criterion than fixed time increment, the voltage controlling the relay amplifier could be selected from anywhere in the problem, with only minor change in operating procedure.

If the punching device were an IBM-type *26* or *526* card punch, as was the case in Battelle's application, punching format could be controlled by a program card. In one application, a modified IBM-type *16* punch was also employed. There is in principle no complication associated with activating a tape punch or a data-logging device producing hard copy in lieu of the card punch. Such choices might be preferred if the available digital computer were suited for tape or keyboard input, instead of cards.

A distinctive feature of the programmer is its relatively low cost and ease of construction. No critical components are employed and construction and maintenance are straightforward. The cost of the programmer cabinet built by Battelle was under $3,000. To this must be added the cost of the digital voltmeter and the punch mechanism; however, these instruments are normally used independently and need not be charged to the programmer.

Applications

Ejection Seat

One interesting application for the device was in the design of an aircraft ejection seat. The parameters of the ejection capsule were to be selected to meet several requirements, one of which was that the stress on the occupant was to be minimized. Capsules pitch violently when ejection occurs under certain circumstances. Very high short-term accelerations can occur at the extremities, where the pilot's head may be located.

In the analysis of this problem, the aerodynamic equations of the capsule were written, treating it as a two dimensional free body in the airstream. These were mechanized for the analog computer. The direction of the gravity vector and the magnitude of the effective wind force vector varied with assumed changes in aircraft altitude and attitude. The objective of obtaining a design minimizing the acceleration of the pilot's head was complicated by the fact that the actual desired acceleration was not obtained on the analog computer. Translational and rotational acceleration components were separately obtained, and used in the simulation, but the net acceleration was a highly nonlinear combination of these.

To resolve this difficulty, it was possible to employ the programmer to interrupt computation every 5 seconds and punch into cards the necessary values from which to compute the desired accelerations. Immediately following each analog run, the cards were processed on a digital computer to produce a table of acceleration values. These were then used to guide revision of design parameter values. Ultimately, a design was obtained which maximized the pilot's survival chances under all conditions.

Satellite

Another application was the calculation of availability of solar energy to an artificial satellite. Consider a satellite part of whose instrumentation is powered by solar batteries. These batteries can accumulate energy only when the vehicle is on the sunny side of the earth, and this energy may be attenuated by layers of atmosphere, depending on the altitude and orientation of the satellite with respect to the earth. The problem considered here is that of obtaining the history of energy level available during a few cycles of the satellite.

The elevation of the satellite is readily computed on the analog computer. Previous experience of analog-computational laboratories has indicated that trajectory problems, including orbital problems, cannot be readily solved directly on analog computers because the force of gravity is so large in comparison to air resistance, etc., that these quantities must be scaled until they are down in the noise level and, therefore, are highly inaccurate. When a preliminary investigation indicated that this was also true of this problem, it was decided to work from perturbation equations.

An initial orbit with a known elliptical shape was assumed and then equations were set up to give the perturbation from this orbit due to air resistance (all other forces were assumed minor in comparison to this one during this particular period in the lifetime of the satellite).

However, since the perturbed elliptical orbit has additional motions superimposed, a further calculation was then required. The axes of the ellipse rotate in their plane, and the plane itself regresses about the earth's axis, these effects resulting from the earth's spheroidal mass distribution. To determine the satellite's position relative to the earth and sun, the orbital calculations were interrupted by the programmer at increments of 45 degrees around the track, and time, angular position, and altitude were punched out. A subsequent digital calculation from these cards located the sun with respect to the orbit plane, in right ascension and declination. These astronomical coordinates were then converted to azimuth and elevation by the computer. A digital integration in space was then carried out from the satellite toward the sun to determine the degradation in solar energy level occasioned by atmosphere. If the satellite was eclipsed, the degradation was automatically set at 100%. The resulting energy intensity values were then punched out by the computer, and a listing of them gave the solar energy history of the satellite at 45-degree intervals as it pursued its rotating, regressing, perturbed elliptical orbit around the earth.

Conclusions

The need for simple inexpensive analog-digital conversion equipment arises many times in the operation of almost any computing center. Existing commercial equipment does not appear to meet this need adequately, but a general purpose plugboard-controlled programmer such as the one described can readily be applied. Experience with the preceding examples and with other data recording uses of the programmer have shown it to be a valuable and highly flexible tool to augment a computer installation.

Discussion

Chairman Rogers: Mr. W. A. Farrand, Autonetics asks Mr. Fuller: "How do you close when writing 1010 initially, when admittedly, self-clocking is necessary? That is, initially you wrote a one zero pattern all the way around the drum. If you wrote it all the way around, there was some reason for it."

H. W. Fuller: Yes.

Chairman Rogers: Mr. Farrand asks, "You should, to some degree, close on yourself so you would be writing a one where a one was, and a zero where a zero was when you came back around, then you started writing your specific pattern. Admittedly, you could not close because you said you had to use self-clocking. Now what trouble does this cause and how do you do it?"

H. W. Fuller: When you close on yourself, it is, as you say, not a good closure. You must print zero over zero and one over one. We do. We record these synchronously on the drum. We erase these synchronously by rerecording. Simply by recording your information you erase what you have written initially, so there is no need for closure.

Chairman Rogers: Mr. Farrand also asks, "Then was there any need for writing the 1010 pattern?"

H. W. Fuller: Yes, there is. If we have a transient occurring as a result of beginning or stopping the recording in the middle of this open zero-one region which affects or governs the tract in the early stages or formation of the track, this transient could erroneously be read as the end of the block start code. We do not want that, so we just coast along as though this were a very long start code until the first block appears.

Chairman Rogers: Mr. Reese, Hughes Aircraft Corporation asks, "What is the gap space between the head and the drum when it is running?"

H. W. Fuller: 180 microinches, I forgot to mention that.

Chairman Rogers: Mr. Reese also asks, "How much runout can you take on the drum surface to keep the heads within their operating position?"

H. W. Fuller: Within 5 or 10 mils at the present spring design, which is a pretty competent spring design.

Chairman Rogers: Mr. Farrand asks, "It would be a dynamic problem, would it not; that is, with respect to suspending the head?"

H. W. Fuller: Yes. Eventually it would become a dynamic problem, but in terms of the sensitivity of separation, the results of the spring force turn out to be not much more expensive.

Chairman Rogers: Mr. Thorpe, Corning Glass Works asks Mr. Gams: "I would like to know about the difficulties in temperature variation in transistors?"

Theodore C. Gams: The amplifier that I showed when used as the input stage uses a pair of transistors such the *4JD*, *A168G*, or other similar RCA transistors for long term stability in bistable flip-flop circuits. These transistors are designed, although no one who makes them will tell you this, to have a low-base collector current, and a low variation of that base, and that is the parameter that determines the temperature stability of the input pair. We have obtained stabilities on the order of $\pm 0.03\%$ to ± 40 degrees centigrade change in ambient temperature. Below -55 C, I do not recommend long term operation of any germanium devices; above 71 C, which is the usual military breaking point, it is necessary for us to consider silicon if you want to have good stability, because any two transistors of the same type and manufacture will leak differently, varying from 50 to 1, between the degrees of 70 and 85 C.

Chairman Rogers: Mr. W. Brooks, Ramo-Wooldridge asks, "At what point in voltage do you attempt to go, let us say, from series to shunt regulators?"

Theodore C. Gams: It depends upon the money available. We like to use series regulators as high as we can because for rather subtle, and often certain reasons, one can get a somewhat better response from the circuit. However, when one begins to protect those series transistors, they begin to take more voltage than the output takes at times. In order to protect the transistor, I would say that the general crossover is between 50 and a 100 volts for relatively high currents; and for low currents, where you can afford to waste power, it is perhaps between 300 and 400. However, let me say that the shunt regulator, all by itself, has certain other advantages. For example, its ability to operate over wider ranges of adjustment without having to store much energy while waiting on the shunt regulator.

Chairman Rogers: The first question for Mr. Schwartz is from Mr. Gams, NJE: "How long it takes for this box of assorted surplus parts to take one reading in relationship to the reading on the curve trace in a manually operated input?"

B. L. Schwartz: It depends upon the balancing time of the digital instrumentation you are using, that is, the principle amount of time involved is balance time.

Now we are using the electronic digital-volt meter, which is fairly subtle. With that one, I would say the time is about the time that it would take to run it off on our curve volt meter. We have now acquired a semi-automatic curve reader. If we had this 6 years ago, when the problem came up, we probably never would have considered it. Taking a photographic trace, we would have gone directly to our curve reader.

Chairman Rogers: Mr. Gams also adds, "I am suggesting just manually. The balance of time for a digital-computer voltmeter is on the order of 2 seconds of the sort you are depicting there, three times as fast as reading it manually and punching it on a card."

B. L. Schwartz: That is very, very true. The main advantage here is not time. We are obviously not doing any high-speed machine work here. In automatic operation you eliminate verifying of the key punch, you eliminate manual transcription, you eliminate basically the delay in taking something off the computer, manually processing it, and taking the card punch. The time saving element is probably in its favor but it is not very significant.

Chairman Rogers: The first question for Mr. Estrin is from Mr. S. Littler, National Cash Register Company: "Why is there any necessity for the shifting operation to be definitely slower than the addition operation?"

General Estrin: It is not slower; it is usually faster than the addition operation. The shifting operation implies a simple transfer from one set of flip-flops to another; the addition operation implies a certain number of logical stages within this transfer.

As I pointed out you are not talking about the carry storage in which you do not propagate the carry. At each step store the state of a one-stage carry; that is, you actually represent the sum, not by one-ended binary quantity but by two-ended binary quantity.

Chairman Rogers: Mr. Lettler asks, "Do you imply that an addition takes at least twice as long as it should?"

Gerald Estrin: No, this is just a mode of carrying out a set of additions in which you would not need to have a computation of the carriage set in one extra step. There has been a great deal of amplification of this in some of the reports that have come out of the University of Illinois.

But I still make the very simple statement that the addition process must take longer even if it is epsilon, with the simple transfer of one flip-flop to another.

Communication Between Computers

WILLIAM S. KNOWLES
NONMEMBER AIEE

IRVING L. WIESELMAN
NONMEMBER AIEE

RAYMOND STUART-WILLIAMS
NONMEMBER AIEE

THE USE of electronic computers to solve large-scale data processing problems has created a number of serious communication problems. Not the least of these problems is the necessity that machines be capable of communicating with each other. To achieve complete communication between machines requires that three conditions be satisfied. First there must be compatibility of the medium used for transference. Second, there must be understanding of the coded language used; and third, there must be the ability to interpret the coded language as structured by the program.

Unfortunately, present machines generate and receive information in different languages and structures. Even identical machines can and do change their language by program means. These differences are not attributable to the ignorance or to the whims of machine designers but instead are due to the operational requirements for which the machines have been designed. It seems clear that for some time, future computers will necessarily follow this pattern of incompatibility.

There are two basic methods which may be used to resolve the communication problem. The first is to legislate that a common medium, code, and language be used, and the second, is to employ other machines to solve some or all of the compatibility problems involved in machine communication. Legislation has the disadvantage that it would involve a tremendous loss of investment, in terms of existing programs, and would also stifle the future development of computers. The second approach is to use machines which may be programmed to interpret the languages of different machines. This does not introduce these disadvantages and, therefore, has been adopted throughout the industry as the preferred approach.

Communication through a transfer medium is almost always employed. Some possibilities for this are printed forms, paper tape, punched cards, and magnetic tape. Magnetic tape, because of its high packing density and high rates of access, is becoming the accepted medium for communication. The use of a transfer medium always introduces both physical and linguistic differences. Two classes of machine will be defined to handle these differences. The first class is termed "media translators." These are used to overcome the physical differences. They transfer information from one medium to another without radical changes of the language employed. The second class is termed "data converters," and these are used to overcome both physical and linguistic differences. They effect both transference and translation of information between media.

In the future a third class of machine will be introduced. This is simply an adapter attached to the existing data processing computer and represents the minimal form of media translator. The media translator will continue to exist as a special-purpose instrument. The data converter will evolve into a machine which uses stored program computing techniques to solve the linguistic problem.

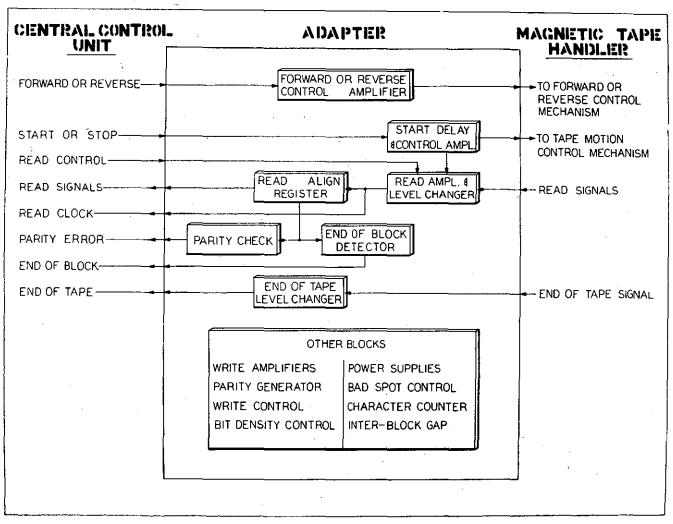


Fig. 1. Typical magnetic tape adapter block diagram

The Compatibility Problem

Historically, computers were designed as either scientific or data processing machines. Each machine was considered from a system point of view, as a separate entity. As the number of data processing machines in everyday use have increased, the results obtained from one machine are used as the input data to a second machine, which is often geographically remote from the first, and which is frequently of different type and manufacture. The direct approach used to solve this problem is to employ either printed paper, punched tape, or punched cards as the transfer medium. Thus, either written human language, teletype code, or punched card code are used.

The use of magnetic tape, as a means of mass data storage, has become more common and this is also being widely adopted as a high-speed transfer medium. It is apparent that future machines will need an increasing ability to intercommunicate. This fact has been recognized to a small degree by the manufacturers who are building new machines with capacious buffered magnetic-tape storage systems.

However, it seems clear that near term machines of the future will not utilize a common language or common magnetic tape equipment. Indeed, it is perhaps not in the best interest of machine data processing progress to permit stasis of this type. Therefore, it may be safely concluded that problems in communication will exist to an increasing degree in the foreseeable future.

In communicating from one machine to another there are both physical and linguistic differences which together create what is generally called "the compatibility problem." These problems can be defined as follows:

Physical-Mechanical

These are the differences: In the tape material; the width of tape; the number of heads; and the head spacing. Further problems arise due to the fact that the rate of tape motion, the acceleration time, and deceleration time vary in the different types of transport. These physi-

William S. Knowles, Irving L. Wieselman, and Raymond Stuart-Williams are with Telemeter Magnetics, Inc., West Los Angeles, Calif.

cal-mechanical differences are so gross that there is very little possibility of devising a tape transport which can accept all types of magnetic tape.

Physical-Electrical

These are the differences: The system of recording employed, such as return to zero or nonreturn to zero; the information packing density and the rate of tape motion which define the character rate. Certain systems have means whereby the condition of the tape can be checked and bad spot areas rejected. Obviously, this operation involves special circuits and delays peculiar to each type of transport.

The foregoing difficulties are resolvable by the use of relatively unintelligent electronic equipment. In essence, units defined as "adapters" are used to convert the signals in and out of a particular type of tape transport to a standard from, acceptable to the computer or to a media translator or to a data converter.

Linguistic-Formal

These are the differences found between tapes produced from different data processing machines utilizing various formats. These differences exist even when tapes are produced on the same tape transport but fed from different machines. Typical differences are found in:

1. Characters: 4-, 5-, 6-, 7-, or 8-bit codes are employed to represent characters.
2. Words: Fixed or variable word length is employed.
3. Blocks: Fixed or variable block length is employed.
4. Sign: Special bits or special characters are employed, either at the beginning or end of words.
5. Checking: Odd or even lateral parity, longitudinal parity, or count of characters for fixed block length.

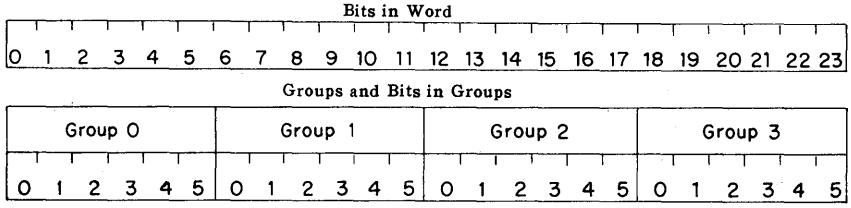
These problems can be solved in most cases by the use of straightforward computing-type circuitry but without requiring programmed operations to perform conversions.

Linguistic-Variable

The major linguistic problem associated with data conversion is the variation from program to program. Linguistic formal differences can be variable in some circumstances. For example, a binary machine may be programmed to produce a coded output. This code is entirely under the control and at the convenience of the programmer. Under these circumstances code problems cannot be defined as a formal difference. As another

Table I. Generalized Central Control Unit Command List

Word Structure



Command Codes, Structure and Function

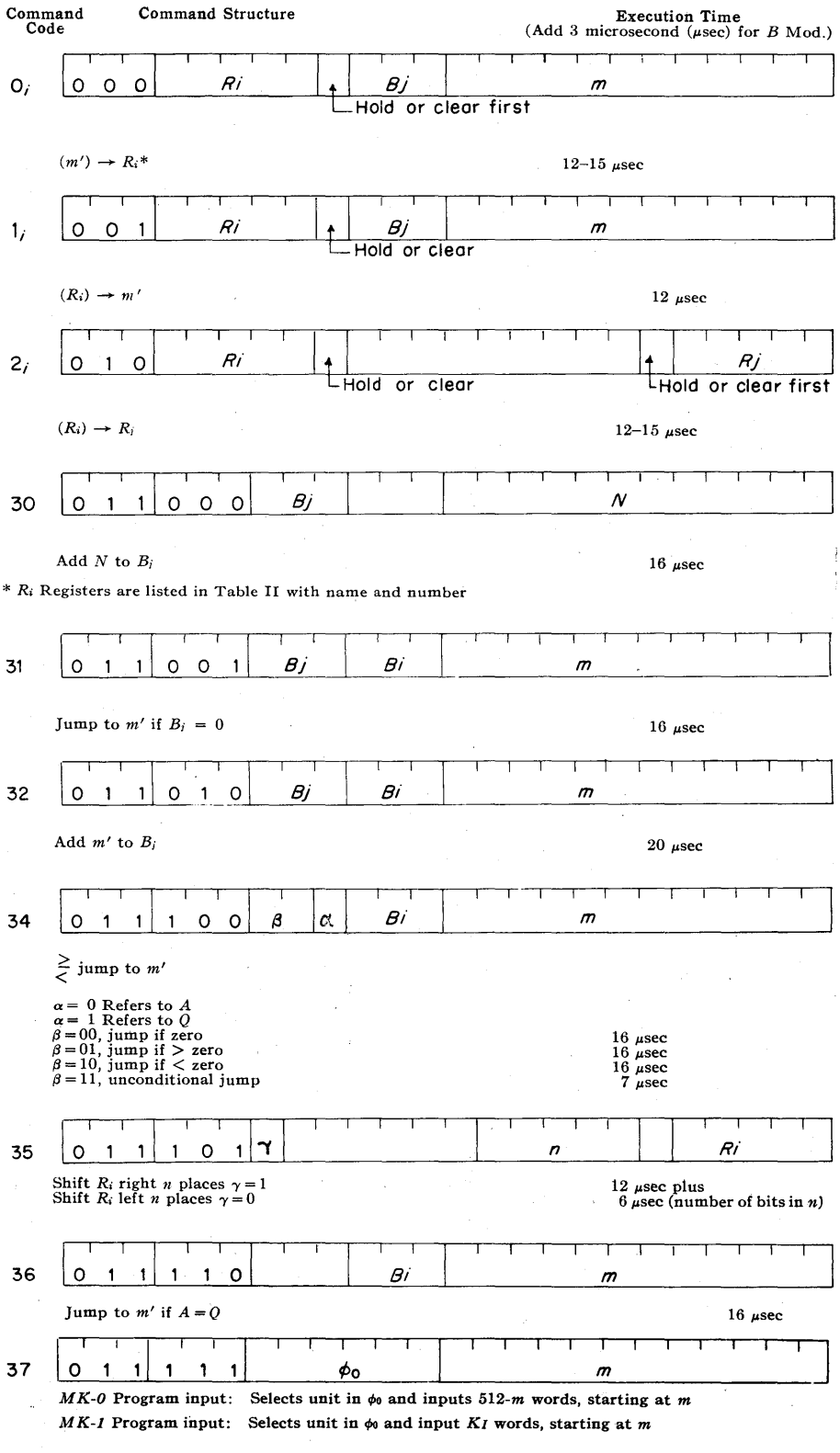


Table I. (Continued)

40 | 1 0 0 | 0 0 0 | ϵ | α | Bi | m | δ

Select group δ of m' and place in group ϵ of $A(\alpha = 0)$ $[Q(\alpha = 1)]$, replacing ϵ — 30 μsec

41 | 1 0 0 | 0 0 1 | ϵ | α | Bi | m | δ

Select group ϵ of $A(\alpha = 0)$ $[Q(\alpha = 1)]$ and place in group δ of m', replacing δ — 33 μsec

42 | 1 0 0 | 0 1 0 | | Bi | m

Stuff address of NIA register plus one in m' — 16 μsec

43 | 1 0 0 | 0 1 1 | ν | α | Bi | m

Form logical product of $E\nu(\nu = 00, 01, 10, 11)$ with (m') and then logically sum with $A(\alpha = 0$ $[Q(\alpha = 1)]$ — 30 μsec

44 | 1 0 0 | 1 0 0 | λ | α | Bi | m

Add (m') to $A(\alpha = 0)$ $[Q(\alpha = 1)]$, Sum in $A[Q]$

$\lambda = 00$, full add	44–70 μsec
$\lambda = 10$, add groups 2 and 3	20–28 μsec
$\lambda = 11$, add group 3	20–28 μsec

45 | 1 0 0 | 1 0 1 | λ | α | Bi | m

Subtract (m') from $A(\alpha = 0)$ $[Q(\alpha = 1)]$, Difference in $A[Q]$

$\lambda = 00$, full subtract	50–78 μsec
$\lambda = 10$, subtract groups 2 and 3	26–34 μsec
$\lambda = 11$, subtract group 3	26–34 μsec

46 | 1 0 0 | 1 1 0 | λ | | Bi | m

Multiply A by m', most significant portion of product in A, least significant portion of product in Q

$\lambda = 00$, full multiply	0.6–1.8 millisec
$\lambda = 10$, multiply groups 2 and 3	0.3–0.8 millisec
$\lambda = 11$, multiply group 3	0.14–0.3 millisec

47 | 1 0 0 | 1 1 1 | λ | | Bi | m

Divide A by m', quotient in Q, remainder in A

$\lambda = 00$, full divide	1.7–3.2 millisec
$\lambda = 10$, divide groups 2 and 3	0.6–1.1 millisec
$\lambda = 11$, divide group 3	0.25–0.4 millisec

5i | 1 0 1 | σ | ν | Bi | m

Jump to m' if flip-flop $F\sigma\nu$ is set
$\sigma = 0\,0$ to $1\,1$; $\nu = 0\,0\,0\,0$ to $1\,1\,1\,1$ — 8 μsec

6σ | 1 1 0 | 0 | σ | W | 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Set $(W = 1)$ [Reset $(W = 0)$] Selected flip-flops $F_{\sigma\nu}$
$\sigma = 0\,0$ to $1\,1$; ν represented by bit position — 8 μsec

64 | 1 1 0 | 1 0 0 | W | | σ | ν

Set $(W = 1)$ [Reset $(W = 0)$] selected flip-flop $F_{\sigma\nu}$
$\sigma = 0\,0$ to $1\,1$; $\nu = 0\,0\,0\,0$ to $1\,1\,1\,1$ — 8 μsec

70 | 1 1 1 | 0 0 0 | ϕ_0 | ϕ_1 | ϕ_2 | ϕ_3 | ϕ_4

ϕ_0 Selects one of 63 units for flow path 0
ϕ_1 Selects one of 63 units for flow path 1
ϕ_2 Selects buffer 1 or 2 for flow path 2
ϕ_3 Selects buffer 1, 2 or R register for flow path 3
ϕ_4 Selects buffer 1, 2 or R register for flow path 4 — 8 μsec

74+θ | 1 1 1 | 1 0 θ | μ_0 | μ_1 | μ_2 | μ_3

Input-Output mode conditioning
$\theta = 0$, flow path 0 selected
$\theta = 1$, flow path 1 selected
$\mu 0$ selects fixed mode of operation for selected unit
$\mu 1$ selects variable modes of operation which may be changed conditional on bits selected in μ_2 and μ_3
$\mu 2$ selects which bits of μ_2 will be changed
$\mu 3$ selects the conditions for change — 8 μsec

77 Stop

example, it is possible to define formal rules whereby the word length or the block length is changed during the process of a data conversion. However, in many programs the word may, in fact, have an implied sign which may be defined by its position in a block and, therefore, change of either a word length or a block length can destroy this implicit meaning so that formal methods are not usable in this conversion.

The problem of translating computer languages is similar to that of translating spoken languages. Each spoken language has its own alphabet, words and grammatical structure. The meaning of a word is a function of the grammatical indicators associated with its common usage, and the surrounding context according to the rules of syntax. The words in the output language are chosen by its structure so as to represent the same meaning as the input. The difficulty in translating spoken languages is that the language structure is so complex that it is difficult to define the rules of translation. However, in the case of translating computer languages no matter how complex the structure is, and how it varies from sequence to sequence, the rules which governed the original program may be stated. Translation may be accomplished provided the computer is capable of performing the rules of operation, and this, in general, can only be done by a fully programmed controlled conversion.

Solution to the Problem

The general problem of machine compatibility is extremely complex. However, solutions are possible using data-processing computers. If adapters or media translators are employed, then either the generating or receiving computer is programmed to perform the actual translation of information. If, on the other hand, a data converter is used, then this must be defined as a special-purpose off-line computer which is used to perform the translation.

Adapters

An adapter is a device which allows a computer to communicate with input or output equipment. Fig. 1 shows some of the functions which must be performed in a typical magnetic tape adapter so that information on the tape may be read or written. Information signals are received and transmitted in a form compatible with the computer. Control signals are interpreted and cause prescribed actions in the tape handler. Signals are

also generated in the adapter which indicate the state of the tape or control information such as parity errors, end of block, and end of tape. The timing for writing information is generated as well as logic and delays for special functions such as tape bad spot controls.

All data-processing computer designers face the problem of building adapters for their own input or output devices. The designer of data conversion computers must build adapters so that alien input-output units can communicate information to the conversion equipment. The design problems differ only in that the data processing designer may simplify the adapter by considering the total system.

It is possible for machines of newer design which have adequate buffering and sufficient input-output tape positions to add adapters so that alien tapes could be processed. The adapters can also be employed for producing alien tapes for use on other machines.

Media Translators

A media translator consists of two or more adapters, plus a central control unit which performs those conversions necessary to introduce compatibility to the physical and formal linguistic level, but not at the linguistic variable level. Fig. 2 shows a media translator built by Telemeter Magnetics, Inc. to perform translations between Remington Rand Univac *1103A* and International Business Machines Corporation (IBM) *704* magnetic tapes in either direction. See functions of Remington Rand Univac *1103A* and IBM Media Translator. In addition to solving the physical compatibility problem, it checks for parity and count errors; it can be used for editing by translating selected blocks; and it solves a formal linguistic problem by inputting continuous data and outputting data in block form.

The central control unit uses a standard Telemeter Magnetics, Inc. *1092-BQ-8* magnetic core buffer to store information. It also has presettable decade counters and transistorized logical and control circuits.

The reasons for existence of media translators are quite logical. When dealing with alien tapes, adapters of some form are required in any case. It is often desired to produce a copy of an original tape for transmission which wastes time for most modern machines. An off-line machine which performs a copying operation is almost as complex as a complete media translator. Media translators are also used to solve compatibility problems by translating information between dissimilar media such as between paper tape, magnetic tape, punched cards, or from unformatted real-time data. For these reasons, and other economic considerations, the media translator is a logical and useful development. It is, however, severely restricted in application as it is incapable of performing any translations other than those defined as formal linguistic differences.

Data Converter

The problems involved in data translation can be solved by using stored program computing methods. The computation required is trivial and repetitive. It is an accepted principle that under these conditions a special-purpose computer is always faster and less expensive than a general-purpose machine. It is on this basis that the data converter exists as a class of equipment. Data converters are fundamentally media translators, to which have been added a special-purpose computer to perform the translation functions. Present data converters are rudimentary in that they are sequence-controlled machines in which the sequence is controlled only by means of a plug-board. Machines of this type are not sufficiently flexible to perform many significant translation operations. They can, however, perform quite complex changes in word and block structure. They can also make simple code changes and can perform some editing functions. The methods of error detection that are employed are quite sophisticated and it is possible to program them to make simple error corrections.

A data converter, built by Telemeter Magnetics, Inc., for the Air Force Missile Test Center (AFMTC) is shown in Fig. 3. It was built so that missile test data produced at AFMTC could be made compatible with formats of various commercial computers in operation at participating missile development organizations in scattered locations.

The present input-output equipment allows for conversion between the AFMTC, Florida Automatic Computer (FLAC) computer paper and magnetic tapes, the IBM *704* and IBM *650* magnetic tapes, and the Remington Rand Univac *1103A* paper tape. (See list of conversions and functions of AFMTC data converter) It has plug-boards and switches which are used to control editing and conversion functions such as selecting given blocks and given words within blocks for conversion, translating from variable word length to fixed word length, changing the position of code for

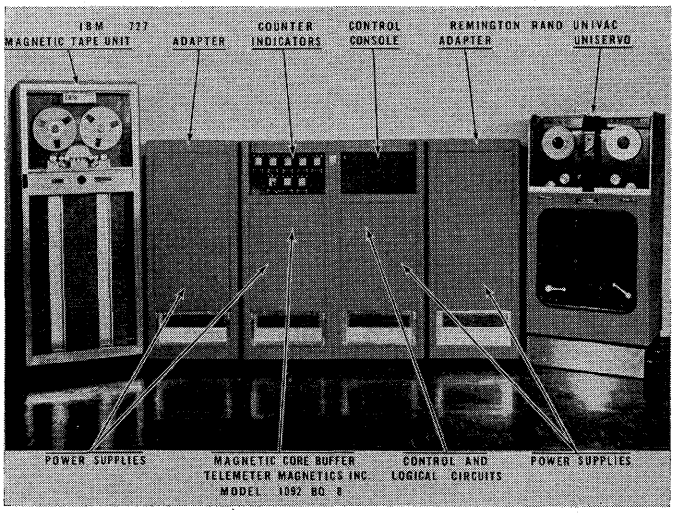


Fig. 2. Remington Rand Univac 1103A-IBM 704 media translator

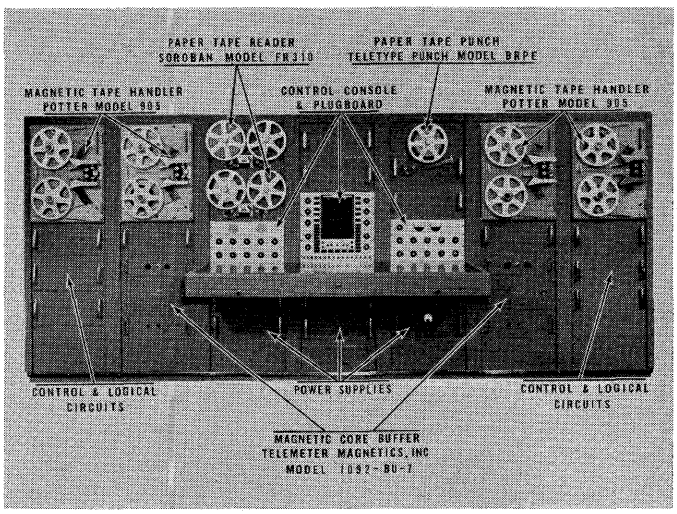


Fig. 3. AFMTC data converter

Table II. Generalized Central Control Unit Addressable Registers, Bit Transfers, Function, and Type

Name	Register No.	Write	Read	Function	Type
B_k	$0k$ ($k=1,2,3$)	$R_j \rightarrow B_{kj}$ ($j=2,3$)	$B_{kj} \rightarrow R_j$ ($j=2,3$)	index register	core storage
B_m	$1k$ ($k=0,1,2,3$) ($m=4,5,6,7$)	$R_j \rightarrow B_{mj}$ ($j=2,3$)	$B_{mj} \rightarrow R_j$ ($j=2,3$)	index register	core storage
A	20	$R_j \rightarrow A_j$ ($j=0,1,2,3$)	$A_j \rightarrow R_j$ ($j=0,1,2,3$)	accumulator	core storage
Q	21	$R_j \rightarrow Q_j$ ($j=0,1,2,3$)	$Q_j \rightarrow R_j$ ($j=0,1,2,3$)	Quotient, LSP of product, or accumulator	core storage
N	22	$R_j \rightarrow \bar{N}_j$ ($j=0,1,2,3$)	$\bar{N}_j \rightarrow R_j$ ($j=0,1,2,3$)	complementor	core storage
L	23	$R_j \rightarrow L_j$ ($j=0,1,2,3$)	$L_j \rightarrow R_j$ ($j=0,1,2,3$)	stores location of initial address when filling or emptying internal memory	core storage
E_k	$3k$ ($k=0,1,2,3$)	$R_j \rightarrow E_{kj}$ ($j=0,1,2,3$)	$E_{kj} \rightarrow R_j$ ($j=0,1,2,3$)	extractors	core storage
W_L	40	$R_j \rightarrow W_{Lj}$ ($j=0,1$)	$W_{Lj} \rightarrow R_j$ ($j=0,1$)	stores left half of word	core storage
W_{LS}	41	$R_j \rightarrow W_{LSj-2}$ ($j=2,3$)	$W_{LSq} \rightarrow R_q$ ($q=0,1$)	shifts right half of word into left half of word	core storage
W_R	42	$R_j \rightarrow W_{Rj}$ ($j=2,3$)	$W_{Rj} \rightarrow R_j$ ($j=2,3$)	stores right half of word	core storage
W_{RS}	43	$R_j \rightarrow W_{RSj+2}$ ($j=0,1$)	$W_{RSq} \rightarrow R_q$ ($q=2,3$)	shifts left half of word into right half of word	core storage
F_k	$5k$	$R^i \rightarrow F_k{}^i$ ($i=0,1,..16$)	$F_k{}^i \rightarrow R^i$ ($i=0,1,..16$)	flip-flop used in input-output, store state of flip-flop register	core storage
S_I	60	$R_3 \rightarrow S_I$		symbol recognizer on input register	flip-flops MK-1 core storage MK-0
D_I	61	$R_3 \rightarrow D_I$		bit mask for S_I	flip-flops MK-1 core storage MK-0
S_O	62	$R_3 \rightarrow S_O$		symbol recognizer on output register	flip-flops
D_O	63	$R_3 \rightarrow D_O$		bit mask for S_O	flip-flops
K_I	70	$R_j \rightarrow K_I$ ($j=2,3$)	$K_I \rightarrow R_j$ ($j=2,3$)	character counter for input register	flip-flops MK-1 core storage MK-0
K_O	71	$R_O \rightarrow K_1$ ($j=2,3$)	$K_O \rightarrow R_j$ ($j=2,3$)	character counter for output register	flip-flops
RS_k (Not addressable) ($k=1,2,4,8,16$)		$R^i \rightarrow RSk^{i+k}$ ($i=0,1,..23$)	$RSk^{i+k} \rightarrow R^{i+k}$ ($i=0,1,..23$)	right shift	core storage
LS_k (Not addressable) ($k=1,2,4,8,16$)		$R^i \rightarrow LSk^{i-k}$ ($i=0,1,..23$)	$LSk^{i-k} \rightarrow R^{i-k}$ ($i=0,1,..23$)	left shift	core storage
W_k (Not addressable) ($k=0,1,2,3$)		$R_3 \rightarrow W_k$ ($k=0,1,2,3$)	$W_k \rightarrow Rk$ ($k=0,1,2,3$)	word assembly	core storage
G_k (Not addressable) ($k=0,1,2,3$)		$Rk \rightarrow G_k$ ($k=0,1,2,3$)	$G_k \rightarrow R_3$ ($k=0,1,2,3$)	group assembly	core storage
H_k (Not addressable) ($k=0,1,2,3$)		$R_j \rightarrow Hkj$ ($j=0,1,2,3$ but $j \neq k$)	$Hkj \rightarrow R_j$ ($j=0,1,2,3$ but $j \neq k$)	stores word without storing group k	core storage
J_0 (Not addressable)		$R_j \rightarrow J_{oj}$ ($j=0,1,2,3$)	$J_o \rightarrow R_{Jo}$	wired to set F_{JO} if all 1's in J_O	core storage
J_1 (Not addressable)		$R_j \rightarrow \bar{J}_1 j$ ($j=0,1,2,3$)	$\bar{J}_1 \rightarrow F_{J1}$	wired to set F_{J1} if all 1's in J_1	core storage

sign, and changing between 4-bit to 6-bit per character binary representation. The central control unit contains two Telemeter Magnetics, Inc., *1092-BU-7* magnetic core buffers for storage and uses transistorized logical and control circuits. Four magnetic tape handlers are provided to permit flexibility so that conversions may proceed while changing reels. Sufficient flexibility was provided so that new conversions could be handled with a minimum of modification.

The sequence-controlled calculator developed into the automatic data processing machine, and in the same manner it is expected that the sequence-controlled data converter will evolve into a stored program, selective sequence-controlled, special-purpose data-processing machine. This will perform complex code changes such as those required in binary to decimal conversions. Frequently, it will be used to reduce the amount of information fed to other machines by using curve fitting techniques and, in the process, may perform sophisticated forms of error detection and correction. It will handle highly variable field structure within the information and will be capable of performing simple sorting operations. This approach will allow the operator to use data-processing machines to their full efficiency without undue restriction in the form of legislation. This concept is compatible with the present development of the computing art in which different machines perform steps in the flow of data processing. The simple intermediate steps will be performed by a data conversion computer.

Proposed Design of Special-Purpose Computer for Data Conversion

This design adopts the philosophy that a central control unit of a data converter which may be programmed like a stored program general-purpose computer, is more useful and flexible than a sequence-controlled plug-board unit. As the translation problem becomes more complex, it is only necessary to change the program. This is different from the philosophy of adding special-purpose computing circuits as new translations occur.

The generalized central control unit (GCCU) has the facility of controlling adapters and internal operations by stored program commands and a plugboard for changing the interpretation of input-output controls to adapters. It contains a high-speed random access core memory called internal memory (IM) for internal storage, (512 24-bit words), two sequential buffer core memories (*TMI 1092-BU-7*) *BU1* and *BU2*, a specially wired core memory called core storage registers (CS) used for most internal registers required for performing commands, a set of flip-flops for controlling input-output functions, a plug-board to define the relation between these flip-flops and the adapters, internal control circuits, and miscellaneous registers. Basic functions are:

1. To select and control input-output adapters and their mode of operation, such as read or write, forward or backward, rewind, etc.

2. To transfer information between adapters, buffers, IM registers, and the CS registers.

3. To interpret information from the adapters or internally generated so as to change the mode of operation by changing the state of the adapter or by internal transfers. For example, the tape unit may be stopped when an end of block is reached, a certain symbol recognized, or a given count reached.

4. To do stored programmed computing on data entering the system so that a translation may be made from an input language, given its structure, to an output language, given its structure. Typical operations are changing word length and structure, code transformation by table look-up or algorithms, changing block length, editing, collating, sorting, and searching.

Information Flow in the Generalized Central Control Unit

The structure of the GCCU excluding the input-output controls is displayed in

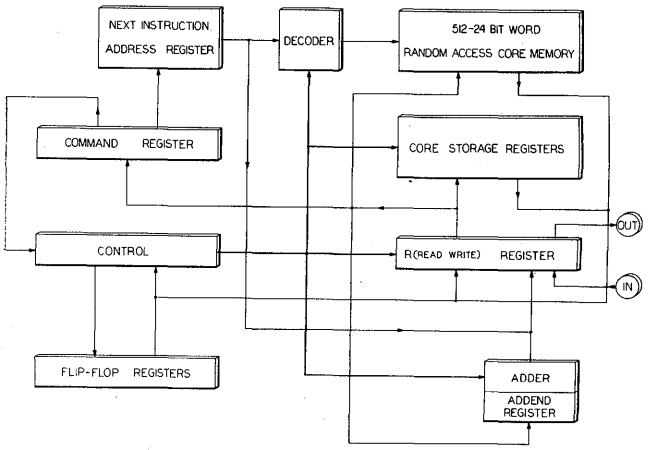


Fig. 4. Generalized central control unit block diagram

Fig. 4. All information flows through the 24-bit R (read-write) register. It serves as the static register for reading, or writing, from, or to, the IM or the CS registers.

The address of the next instruction is stored in the next instruction address register (NIA) which is decoded to select the instruction from the internal memory. The word in the selected address is read into the R register and then transferred to the command (C) register.

Most commands are executed by transfers between information in the CS and the IM. The read-write time for random access to a word in the IM is 6 microseconds. The read and write times for selected CSR are 3 microseconds each. Any commands involving arithmetic operations utilize the 12-bit parallel adder. Individual flip-flops in the flip-flop registers which control input-output operations may be set and reset either by commands or from other information sources. Information from input-output enters and leaves the IM and CS register through the last six bits of the R register.

A list of the commands, their bit structure, the functions performed, and execution times may be found in Table I. The 24-bit word may be considered as four 6-bit groups. The first group usually contains the two octal digits of the command code. The second group controls index registers and miscellaneous operations. The last two groups usually contain the address involved in the operation with the last two bits of the address referring to one of the four groups of a word when pertinent.

Structure of the Core Storage Registers

One of the unique features of the GCCU is the way the CS registers are utilized in performing instructions. A complete list of functions of the registers and their wiring may be found in Table II. Their full use in the commands will be discussed later. Some typical registers will be discussed.

Fig. 5 shows schematic wiring of the LS_1 (left shift 1), A (accumulator) and RS_1 (right shift 1) CS registers. Information is transferred between the R register and the selected CS register by writing into the selected register from R and reading the selected register into R. The wiring of the cores for each register determines into which bit positions the information is to go. The wires with arrows pointing upward refer to writing in the cores. The wires with arrows pointing downward refer to reading from the cores.

When writing into the A register, bit R_j^i (i is a bit in group j) is written into a core with the same bit position when A is selected. When reading, this core is wired back to bit R_j^i. The A register merely stores information without change. When the RS_1 register is selected, information written from bit R_j^i is read into bit R_j^{i+1}. Thus a shift of one bit to the right is accomplished.

The series of transfers required to shift the A register one bit to the right would consist of:

$A \rightarrow R$

$R \rightarrow RS_1$

$RS_1 \rightarrow R$

$R \rightarrow A$

The shift commands are mechanized by transfers of this type to registers RS_1, RS_2, RS_4, RS_8, and RS_{16}. A shift of binary 10110 (22) to the right utilizes registers RS_2, RS_4, and RS_{16} in executing the command.

Fig. 6 shows the wiring of the word assembly (W_i) and group assembly (G_i) registers. Since information is to enter the IM from the buffers in 6-bit characters, it is necessary that these characters be assembled into 24-bit words. Words consist of 24 bits or four groups of six bits. The figure shows the wiring of one bit from each group. The W_i registers each contain six consecutive cores representing the six bits of the selected group. The groups entering will be

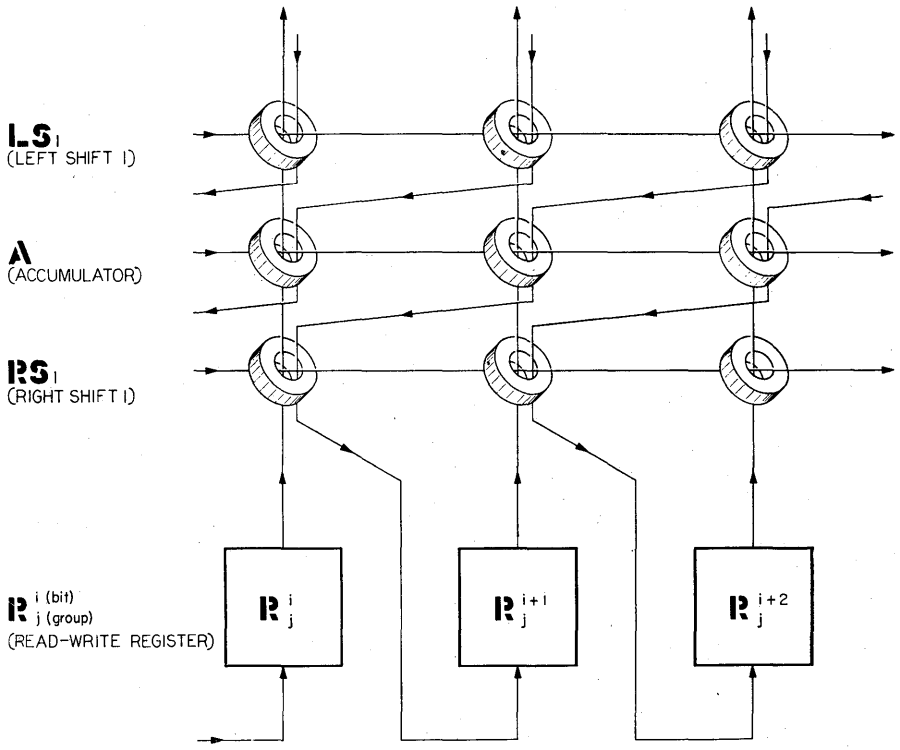


Fig. 5. Input-output selection and information flow (GCCU MK-O)

TABLE III. EXAMPLE OF MODE CONDITIONING COMMAND

7	4	μ_0	μ_1	μ_2	μ_3
1 1 1	1 0 0	1 1 0 1 1 0	1 0 0	1 0 0	1 1 0 1 1 0

μ_0: Selects Fixed Mode of Operation for Selected Unit

- μ_0^0 = Tape Motion Forward or Backward [Fwd.]
- μ_0^1 = Read or Write Tape [Read]
- μ_0^2 = No Function
- μ_0^3 = Write Density 1 or Density 2 [Density 1]
- μ_0^4, μ_0^5 = Control Interblock Spacing

μ_1: Selects Variable Mode of Operation for Selected Unit

- μ_1^0 = Start or Stop [Start]
- μ_1^1 = Fast Rewind
- μ_1^2 = No Function

μ_2: Selects μ_1 Modes to Change if any Flip-Flops Selected in μ_3 Set

- μ_2^0 = 1 (μ_1^0 changes state if any Flip-flops in μ_3 set)
- μ_2^1 = 0 (μ_1^1 does not change state)
- μ_2^2 = 0

μ_3: Indicates a Change in Mode of Information Concerning Adapter

- μ_3^0 = End of Block
- μ_3^1 = Selected Count of Characters Exceeded
- μ_3^2 = Symbol Recognized
- μ_3^3 = End of Tape
- μ_3^4 = End of File
- μ_3^5 = No Function

called γ_0, γ_1, γ_2, and γ_3, and the desired word is the string $\gamma_1\ \gamma_2\ \gamma_3\ \gamma_4$. Each group enters group 3 of R (R_3) in sequence with the bits of each group transferred in parallel. The sequence of transfers necessary to assemble a word consist of $\gamma_0 \rightarrow R_3$, $R_3 \rightarrow W_0$; $\gamma_1 \rightarrow R_3$, $R_3 \rightarrow W_1$; $\gamma_2 \rightarrow R_3$, $R_3 \rightarrow W_2$; $\gamma_3 \rightarrow R_3$, $R_3 \rightarrow W_0$; and finally the simultaneous transfer of $W_i \rightarrow R_i$ (i = 0,1,2,3,). Thus the assembled word ends up in the R register and may be written into memory.

When a word is to be read from memory to a buffer a character at a time, the CS registers G_i (i = 0,1,2,3) are used. The simultaneous transfer $R_i \rightarrow G_i$ (i = 0,1,2,3) occurs first, then the sequential group transfers $G_i \rightarrow R_3$, $R_3 \rightarrow$ buffer (i increasing from 0 to 3).

Command codes which have their first octal digits a 0,1, or 2 control transfers between certain addressable numbered CS registers R_i (see Table II) (i = 1,2,. . .73) and IM and among themselves. Typical of these addressable registers are those which serve the function of index registers, accumulator, quotient, extractors, and complementor.

Other CS registers, which are not explicitly addressable in these commands, such as the registers used for shifting and assembly are used in executing other commands. Commands 40 and 41 are typical in that they use special core registers to select any group from the contents of an index-modified address and place it in a selected group of A or Q (quotient register) and vice versa.

Structure of the Input-Output Controls

The structure of these controls allow the GCCU to achieve the versatility required in performing data conversions. In the process of designing the input-output controls, two systems of information flow emerged. *MKO*, Fig. 7, the simple system is less versatile, and less expensive since it uses one input-output register. *MK-1*, Fig. 8, uses two registers, one for input and one for output allowing a very versatile system, but more expensive.

Command code 70 is used in selecting information flow paths. It is divided into five parts controlling five information paths denoted by ϕ_i (i = 0,1,2,3,4). The command is executed by setting flip-flops which select the chosen paths. These paths remain until they are changed by another 70 command with different ϕ_i's. The amount of information flow, when it starts and stops, is controlled by commands 74 and 75.

MK-0, Fig. 7, only uses four ϕ's. ϕ_0 controls information flow from the adapters, buffers, or R register to the $I/0$ register. The superscript on ϕ_0 denotes which particular device is selected for transmission. ϕ_1 controls information from $I/0$ to a selected adapter ϕ_1^j (j = 4 to 63). ϕ_2 controls information from $I/0$ to a selected buffer ϕ_2^k, (k = 1,2). ϕ_3 controls information from $I/0$ to the R register.

Two or more ϕ's in command 70 set up an information flow path. If a numerical zero is in a given ϕ field, that ϕ is not selected. Information normally flows from a selected adapter i to a buffer. Thus if one block is to be transferred into buffer 1, ϕ_0^i and ϕ_2^1 would be selected. The information is processed by taking information from buffer 1 to IM by selecting ϕ_0^1 and ϕ_3^3. If it is desired to recirculate the information in buffer 1, then $\phi^1{}_2$ is also selected. The processed information may then be sent to buffer 2 by selecting ϕ_0^3 and ϕ_2^2. It probably would then be sent out to adapter j by selecting ϕ_0^2 and ϕ_1^j.

MK-0 thus has the limitation that only one process such as inputting, outputting, or computing is going on at any one time. Except for the loss in time in getting things done this isn't too serious. However, it does limit the ability to input from a real time source which is generating data at a constant rate and cannot be stopped without losing data.

MK-1, Fig. 8, uses five flow path selectors ϕ_i. ϕ_0, ϕ_1, and ϕ_2 control information flow similar to *MK-0* except that the input and output registers are separated. ϕ_3 controls information from the input register to the R register or a selected buffer. ϕ_4 controls information from a selected buffer or the R register to the output register.

Considerable flexibility is possible now. Information may be flowing to buffer 1 from adapter i, and information may be going to the R register from buffer 2 by

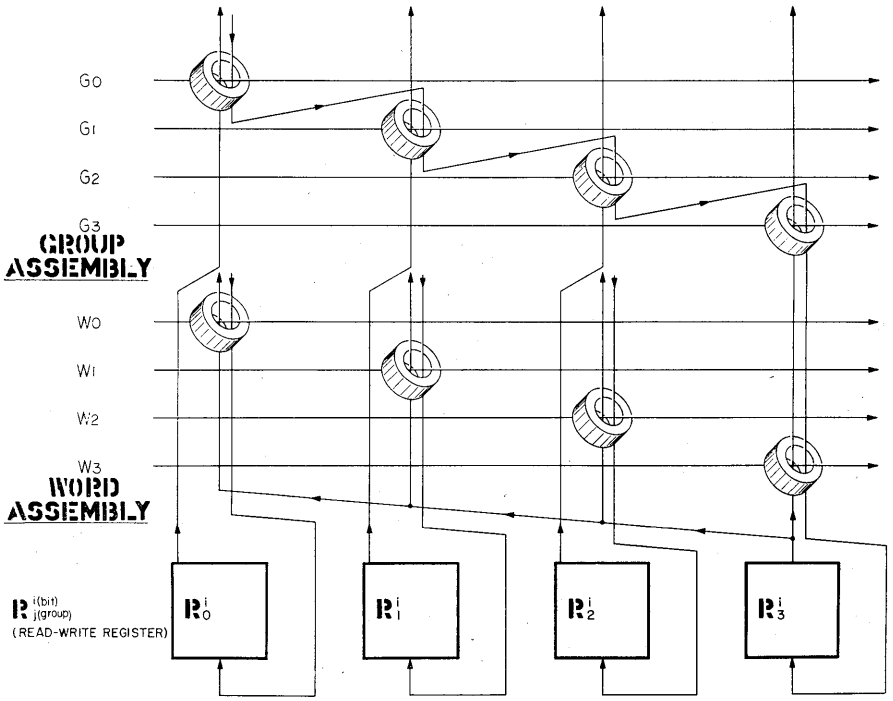


Fig. 6. Input-output selection and information flow (GCCU MK-1)

selecting ϕ_0^i, ϕ_3^1, ϕ_4^2, and ϕ_1^3. Inputting and outputting, while recirculating a buffer, may be done by selecting ϕ_0^i, ϕ_3^2, ϕ_4^1, ϕ_2^1, and ϕ_1^i. It is possible to output from buffer 1 and transfer information back and forth from buffer 2 to the R register by selecting ϕ_4^1, ϕ_1^i, ϕ_0^3, ϕ_3^2, and then selecting ϕ_4^1, ϕ_1^i, ϕ_0^2, ϕ_3^3.

All the combinations of paths will not be enumerated here, since they are too numerous. It is interesting to note that the roles of adapters, buffers, and the R register are similar in being transmitters or receivers through the input and output registers. Thus, transfers between internal units, buffers and R register, are controlled like transfers between internal and external units. The limitations imposed by using the *MK-0* scheme no longer exist. Real-time continuous inputs may be used provided that there is time to process and output the data. Computing may be going on simultaneously with inputting or outputting. There is a gain not only in being able to do more functions, but a real gain in getting the conversion job done much faster.

The valves of input-output information are controlled by the mode conditioning commands 74 and 75. These commands select a mode of operation for an input-output unit, and determine how the mode will change dependent upon other selected conditions. The input-output plug-board determines specific relations between selected bits of the command and properties of the input-output units. The second octal digit 4 or 5 of the command code determines which flow path is being controlled. 74 controls ϕ_0 selected paths and 75 controls ϕ_1 selected paths. Only these two paths need to be controlled. It is only necessary to establish that either the generator or the receiver controls the basic timing.

The command consists of four parts μ_i (i = 0,1,2,3). An example of a command is given in Table III. μ_0 selects the fixed mode of operation for the selected units; for example read or write, forward or backward, and bit density. Modes selected here remain fixed until a new command is given changing them. μ_1 selects variable modes of operation which may be changed conditional on bits selected in μ_2 and μ_3. μ_2 selects which bits of μ_1 will be changed, and μ_3 selects the conditions for change. For example a bit in μ_1 may control start or stop. If the corresponding bit in μ_2 is selected, then the unit will change from start to stop when conditions selected in μ_3 occur. The μ_3 condition might be that an interblock gap occurs, or a certain number of characters has entered, or a symbol has been recognized, etc.

The plug-board wiring determines which particular characteristics of the selected unit are to be connected with bit positions of the command, and which flip-flops of the flip-flop registers are to be used for the bit positions. For example, when adapter i is used, its forward or backward control is connected to μ_0^0, and the flip-flop which controls might be F_{012}. Adapter j might have its forward or backward control connected to μ_0^0 also, but the flip-flop which controls might be F_{013}. Thus adapter i could be inputting, while adapter j is outputting.

The number of characters entering the I register will be counted in the K_I counter, and also those leaving the O register will be counted in the K_O counter. Fixed counts may be selected by presetting the counters since the counters are addressable through the register transfer commands (first octal digit 0,1, or 2).

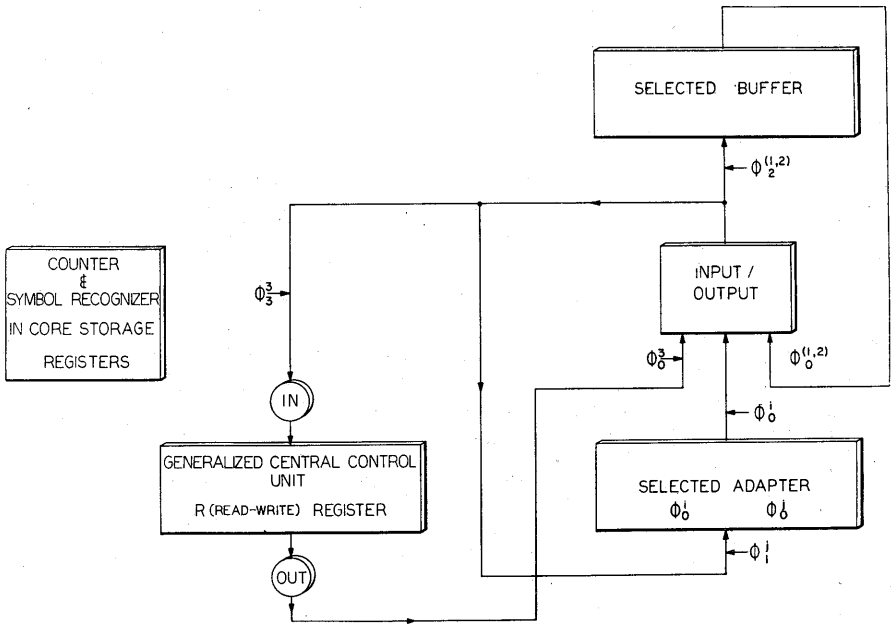


Fig. 7. Core storage re gisters LS_1, A, and RS_1, wiring

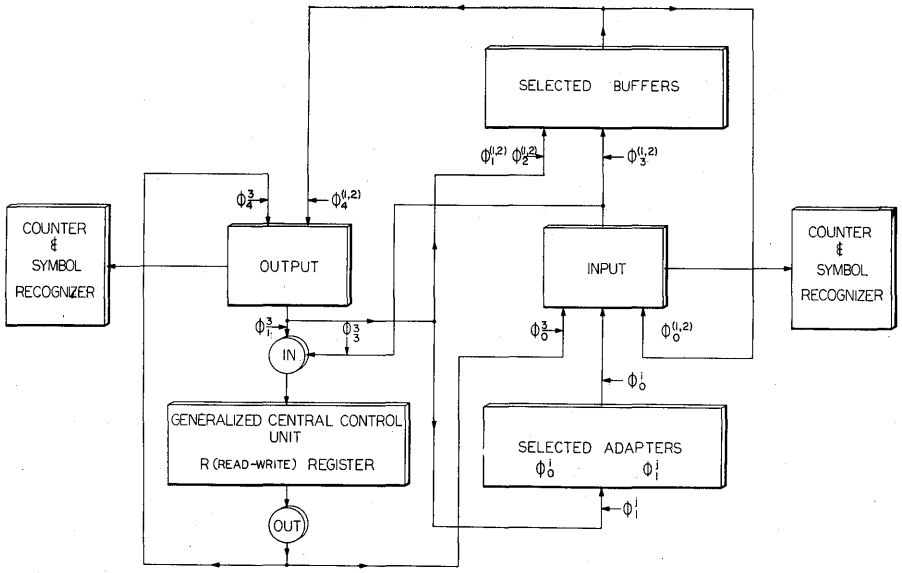


Fig. 8. Core storage registers G_i and W_i wiring

Selected flip-flops will be set when the counter goes through zero.

Arbitrary symbols may be recognized when passing through the I or O register. This is accomplished with the addressable registers S_I, D_I and S_O, D_O. S_I and S_O contain the symbol and D_I and D_O contain those bits of the symbol which are to be compared. This was included since it is not always desirable to compare all six bits of a character for recognition.

The Telemeter Magnetics, Inc. *1092-BU-7* sequential buffers have the property that a pulse is emitted when the buffer is full or empty. These pulses may be used to set flip-flops by using the plug-board position associated with the two internal buffers. If block sizes greater than *1092* are required, standard *2184-BU-7* buffers may be used instead of the *1092-BU-7*. It should be pointed out that additional memory may be added to the GCCU by considering additional buffers as input-output equipment. Similarly storage devices of large capacity such as magnetic drums may be added as input-output devices if needed for some particular applications.

Other Commands

The other commands not previously discussed relate to index registers, jumps, flip-flop controls, arithmetic operations, and extracts. These commands are listed in Table I; however, they will be discussed briefly to complete the picture of the GCCU.

There are seven index registers B_i (i = 1,2,. . .,7) contained in the CS registers. Commands which may be index-modified contain a B_i in the last three bits of group 1. Transfers to and from B_i's are controlled by the transfer commands. The contents of an index register may be modified by adding N to it or by adding the contents of an index-modified address. A jump instruction is also provided so that a jump to an index-modified address is made if the contents of the selected index register is zero.

There are conditional jumps made on the contents of the A or Q registers being zero, and an unconditional jump. Index-modified address jumps may also be made on the set condition of any of the 64 addressable flip-flops. Groups of flip-flops selected by bit position in a word or by address may be set or reset. The use of these flip-flop commands allows the programmer to program break-points, since flip-flops can be set by switches. When trouble-shooting programs, the contents of IM or the CS register may be examined by programming the GCCU to read out the information of interest on an automatic typewriter.

The arithmetic operations provided are add, subtract, multiply, and divide. Addition and subtraction may be done by adding and subtracting the contents of an index-modified address to either the A or Q register, with the sum remaining in the register. Multiplication and division are performed with the multiplicand (or dividend) in A, and the multiplier (or divisor) in an index-modified address. The most significant part of the product appears in A, and the least significant part in Q. The quotient appears in Q and the remainder in A. These operations may be performed on the whole word, the last 12 bits or the last 6 bits (group 3). The representation of negative numbers is by 1's complement.

There is an extract command which forms the logical product of the selected extract register with the contents of an index-modified address, and which then forms the logical sum of this product with A or Q. The extract registers E_i (i = 0,1,2,3) are addressable CS registers.

Applications of the Generalized Central Control Unit

The GCCU will do as fast a job as a media translator or a data converter with a conventional CCU will do. It will do much more since programs can be written which will take care of any translation problem which can be programmed. If new adapters need to be added to take care of new physical format problems, they are added by means of the plug-board. But the internal machine does not change, even if new linguistic problems arise.

The time required to do a conversion of course depends on the complexity of the operations, and the capabilities of the programmer. An IBM *705* to Remington Rand Univac *1103A* conversion program was written. The conversion rules were that variable IBM *705* words were to be made into fixed length *1103A* words by chopping long words and adding zero to short words. The binary coded decimal word was to be converted to a binary word. The variable IBM record was to be made into a fixed Remington Rand Univac block.

All the housekeeping, counting, and rearranging took 75 words in IM, including instructions and storage of constants. The binary-coded decimal to binary conversion subroutine utilized a table of the double-precision binary equivalents of the *b c d* digits for the appropriate powers of ten. This takes 9 (*b c d* character/power of 10) times [7 (powers of 10) times 2 (double precision binary equivalents) + 4 (powers of 10) times 1 (single precision binary equivalents)] or 162 words in IM. The double-precision binary equivalent is found by summing the equivalents of each *b c d* digit found in the table. In addition to the table, 56 words were required to program the subroutine. This method of table look-up is the most efficient timewise technique to use for the GCCU, since the machine was designed to handle manipulation of data much more rapidly than its arithmetic operations.

Not much has been said of other adapters beside those used for magnetic tape units. Some important adapters which have been considered are these which allow for punched card and punched paper tape inputs, and for punched card, punched paper tape, and high-speed

printer output. In the punched card and printer cases, special magnetic core buffers would be used in the adapters for the special conversions required. In the case of punched card input, a Telemeter Magnetics, Inc. *80-CB-7* magnetic core buffer would be used. This takes information presented in 80 columns, 12 rows and presents it with some decoding into 6 bit serial characters. For punched card output, a Telemeter Magnetics, Inc. *80-CB-12* magnetic core buffer would be used. This takes 6-bit serial character information and presents 12-row 80-bit parallel output. A typical printer might use a *120-BA-56* which takes 6-bit serial character information and forms a print wheel core image for printing.

Besides doing conversions with difficult variable linguistic format transformations, this equipment is well suited to do complex editing searches, collating and sorting. Simple editing, like selecting given characters out of given blocks, is done at speeds comparable with pluggable data converters with unlimited selection capabilities.

Complex editing used when selecting only significant data by criteria such as selecting significant points for curve fitting take more time, but can be done efficiently if not too many arithmetic operations are required. Searches, collating and sorting are done rapidly since the equipment was designed to handle table look-ups, comparisons, and manipulation of data rapidly.

Remington Rand Univac *1103A*—IBM *704* Media Translator Functions

The functions of the Remington Rand Univac *1103A*—IBM *704* media, translator follow:

A. Conversion
1. Remington Rand Univac 1103A magnetic tape and IBM 704 magnetic tape.

B. Functions
1. Checks Remington Rand Univac *1103A* magnetic tape for lateral parity and block length.
2. Checks IBM *704* magnetic tape for lateral and longitudinal parity.
3. Fixed block length conversions from blocks of 60 characters to 1,000.
4. Input may be continuous up to 5 kc (without inter-block gaps) and output in block format.
5. Bad spot logic and control for Remington Rand tape.
6. Block counters to position tape or translate a preset number of blocks.
7. IBM end of file mark stops conversion optionally.
8. Speed limited only by tape handlers.
9. Newly written tapes may be checked on rewind.
10. End of tape mark can be written manually.

Air Force Missile Test Center Data Converter Functions

A. Conversions
1. FLAC magnetic tape⇄FLAC paper tape.
2. IBM (*704, 650*) magnetic tape⇄FLAC paper tape.
3. Remington Rand Univac *1103A* paper tape⇄FLAC paper tape.
4. FLAC paper tape⇄FLAC paper tape.
5. IBM (*704, 650*) magnetic tape⇄FLAC magnetic tape.
6. Remington Rand Univac *1103A* paper tape⇄FLAC magnetic tape.
7. FLAC magnetic tape⇄FLAC magnetic tape.

B. Pluggable or Switch Operations
1. Odd or even lateral parity check and generation.
2. Ten character codes may be recognized and used for control.
3. Parts of blocks may be converted by counter selection:
   (a) Select any four words in block.
   (b) Select all words between counter selected boundaries.
4. Block counter used to convert every *n*th selected block.
5. Fixed word length from variable word length by clipping or adding zeros.
6. Automatic complementing and sign code conversion from *1103A* paper tape.
7. FLAC to IBM *650* causes a change of position of sign in word.
8. A word of six 6-bit characters may be converted to a word of ten 4-bit characters and vice versa.

The Universal Data Transcriber. A New Approach to Data Conversion Equipment

MARVIN S. MAXWELL
NONMEMBER AIEE

THE Universal Data Transcriber is a device to convert digital data from one form to another. The input to the system can be cards, paper tape, magnetic tape, or the output of almost any source of digital data that does not exceed 500,000 bits per second. The output of the converter can be cards, magnetic tape, paper tape, low- or high-speed printers, point plotters, or almost any other digital output device. The conversion process allows for very flexible code and format conversion as the data is transcribed from one medium to another.

The Universal Data Transcriber consists of appropriate input-output equipment and a microprogrammed, semivariable-structure, stored-programmed, single-address binary computer. The computer has a high-speed random access memory of 8,192 characters of 8 data bits plus one check bit each, and will average 70,000 instructions per second. The operation of the system depends upon the microprogram of the computer to generate special orders which will transfer data from the particular external input device currently in use to the computer memory, and from the memory to the external output device currently in use. The use of microprogramming, which is accomplished by use of a plugboard, allows an efficient transfer of data between the computer memory and the external devices with a minimum of special equipment. Conversion of the data within the memory from one form to another is accomplished by the use of an appropriate stored program. This gives a very flexible system, since all that is required to change the system from one job to another is to change the connections to the external equipment, insert a different plugboard, and load a new program into the computer memory. This system was conceived, designed, and is under construction by the Computer Research and Development Branch of the Computation and Exterior Ballistics Laboratory of the United States Naval Proving Ground, Dahlgren, Virginia.

Description of the Computer

The basic structure of the computer is shown in the diagram of Fig. 1. It con-

Marvin S. Maxwell is with the United States Naval Proving Ground, Dahlgren, Va.

A project of this nature is the work of a number of people. The author wishes to thank P. H. Burkhart for his assistance in writing this report, W. P. Warner for his assistance in preparing preliminary programs, and J. Pinker, Jr., G. V. Olds, A. O. Danello, J. C. Carlock, E. J. Culhan, and J. R. Gros for their work in designing the system.

sists of an input register, output register, two computing registers (*R*-1 and *R*-2), six *B*-registers (address modifiers), instruction register, instruction counter, indicator latches (single-bit registers), and other special registers. External devices communicate with the computer via the input and output registers under control of the computer. The input register can select at high speed from either of two different external devices. The output register is normally connected to only one unit. Indicator latches are used both to control the external devices and to signal the condition of the external devices to the computer. Special electronic signal-generating equipment tailored to each type of external device is used to facilitate communication with the input register, output register, indicator latches, and the external device.

The computer has a high-speed random access core memory with a capacity of 8,192 characters, each consisting of eight bits plus a check bit. A character is treated as a pure binary number by the arithmetic section of the computer and a single instruction generally affects only one character. Since there are no multiply or divide orders, the operating binary point may be considered to be in any convenient location. The carry (borrow) bit may be propagated from character to character in addition (subtraction) with use of double precision orders. A single reference to the memory brings out four characters designated as *M*0, *M*1, *M*2, and *M*3 into the memory register. Addresses evenly divisible by four always correspond to the character read out as *M*0. Instruction words consist of the four characters *M*0, *M*1, *M*2, and *M*3. Instruction words are logically divided into 4 fields as shown, namely: operation code, *B*-register specification, address specification of reference to memory, and the limit value of *Bx*.

The time required to execute orders which require only one reference to memory such as program transfers is approximately 11 microseconds. Orders which require two references to memory, such as read out or store orders, require approximately 21 microseconds. A complete description of the order codes is available in Appendix I.

Each of the six *B*-registers has eight binary columns plus a check column. Address modification is performed by overlaying the contents of the eight least significant columns of the 13 binary-column address field with the contents of the specified *B*-register. The overlaying process is a logical addition in which the contents of a binary column in the address field is changed only when a zero is overlaid with a one-bit.

For example:

Address field	1	1011	1001	0110
B-register			0101	0011
Result	1	1011	1101	0111

The contents of *B*-registers may be incremented by one either automatically after use if so desired or on command. If it is desired to increase the contents of a *B*-register by any value other than one or to decrease its contents by any value, the *B*-register may be operated on by the use of appropriate order codes.

One of the *B*-registers, *B*6, is normally connected as a real-time counter. It may be preset to any value up to its limit of eight bits and a one is added to the contents of the register every 0.504 millisecond. The counting ceases when the contents of the register reaches 377 octal. With use of this feature and instructions for read out and limit sensing on the *B*-registers it is possible to program measurement of real-time for controlling external devices.

	Bits	
Operation code	8 7 6 5 4 3 2 1	*M*0
B-register specifications	8 7 6 5 4 3 2 1	*M*1
Address specification of reference to memory	8 7 6 5 4 3 2 1	*M*2
Limit value of *Bx*	8 7 6 5 4 3 2 1	*M*3

Checking Procedures

The computer has automatic circuitry built into the system to check the accuracy of its operation. This check adds a parity bit to the 8 bits in each character so that the modulo two sum of the binary one's of these nine bits is always odd. This check bit is generated after data enters the input register, is corrected as the characters are modified by various orders, and is stored in the memory along with the character. An automatic check is made for the presence of the proper parity count as the data are transferred from the memory into the working registers or the instruction register. The values in the *B*-registers are checked automatically as they are used and there are checks on the execution of the overlay and shifting operations in the computing registers.

Whenever possible, checks will be made on the accuracy of data transmission between the computer and the external devices. For example, in card reading, data will be loaded into two independent shift registers from two reading stations, and after the card images are assembled in memory they will be checked against each other. In punching data into cards, the card will be read back into the computer after being punched, and this card image will be checked against the card image sent out to the punch. When magnetic tapes are written, the data will be read back into the computer and a check will be made on the correctness of the data.

Special Features of the UDT

The most outstanding difference between the computer of the Universal Data Transcriber (UDT) and any other single-address binary computer is the availability of the plugboard and the plugboard instructions. The plugboard is divided into three regions. The first region consists of information coming from equipment in the computer to the plugboard. This includes all of the registers, such as register 1, register 2, input register, output register, instruction register, instruction counter, *B*7, and the indicator latches, plugboard instruction specification, and the internal clock. Also in this region are external inputs from the various input and output devices which have been converted to the proper signal levels. The second region of the plugboard consists of a set of approximately 75 logical packages. These packages are identical to those used in the construction of the rest of the computer. In the third region of the plugboard are exits from the plugboard to the control lines in the computer. These lines control the transfer of data from "register to register," use of the *B*-registers, controlling memory cycles, setting of indicator latches, shifting various registers, etc. Thus, by using all

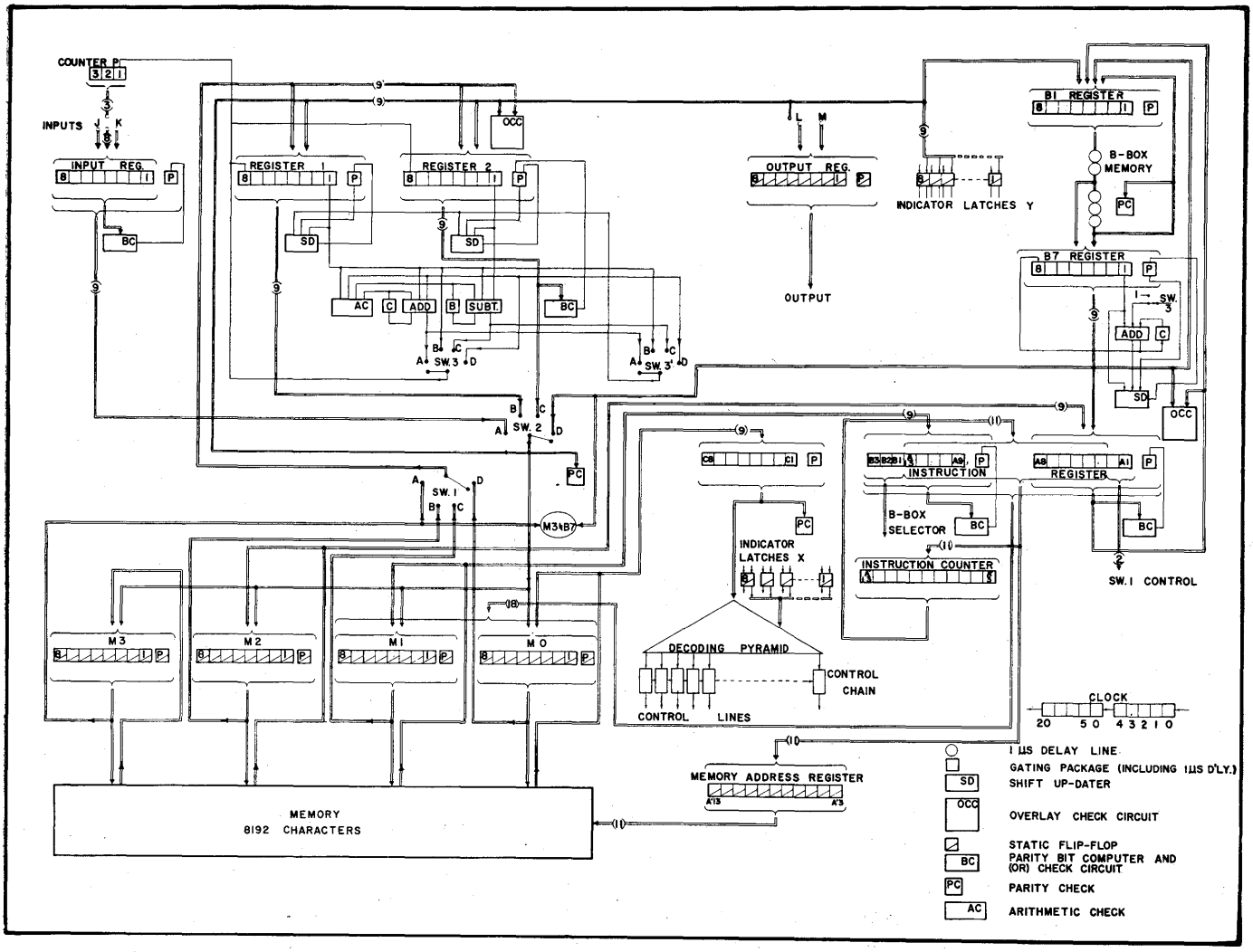


Fig. 1. Block diagram of Universal Data Transcriber

three regions of the plugboard almost any conceivable, or desirable, cycle of actions can be controlled from the plugboard. This feature is primarily for use with external devices to get data to or from them and the memory of the UDT.

The indicator latches in the computer are used primarily for communication between the UDT and external devices. For example, some of the indicator latches could be wired, via the plugboard, to control the stopping, starting, or reading or writing of a tape unit. Other indicator latches could be used to indicate to the UDT that an external device is in certain conditions; for example, that a card reader is moving cards, or ready to scan one row of information, or that it is out of cards, etc. Thus, the program can control external devices, and external devices can be sensed by the program by use of the indicator latches.

Another feature of the UDT is the "program interrupt" ability. If a particular exit on the plugboard is energized the computer will go into a program interrupt cycle. This exit can be energized from an indicator latch, or combinations of indicator latches and various conditions by wiring on the plugboard. When this condition occurs the computer will automatically make a program transfer to instruction location 4 at the end of the current instruction. The address, Y, of the instruction which would have normally been executed next, if the program interrupt condition had not occurred, will be automatically stored in character locations 1 and 2 in a form so that if the character in location 0 is the code for a program transfer, jump, command and the instruction at location 0 were to be executed, the computer would jump to the proper address, Y. When this feature is used the program, starting at location 4, must be suitable to take the appropriate action for the condition which caused the jump. After this is done, the program would normally remake the appropriate registers, and then jump to location 0, which would cause the jump back to the main program at the proper place. By using this feature the computer can react rapidly to external control information without requiring repeated sensing on the condition.

Advantages and Limitations

The major advantage of the UDT is its flexibility. It is not tailored to any specific computer or type of data conversion and is therefore not likely to become obsolete as fast as many specialized converters. The microprogramming and stored program features makes it easy to implement almost any desired conversion with a minimum of engineering effort and special equipment. The major disadvantage of this approach is that it is more expensive than any single specialized converter.

To establish the capabilities of the UDT, several preliminary programs have been prepared. One program for converting 80-column alphanumeric International Business Machines Corporation (IBM) cards to NORC magnetic tape

provides for arbitrary code and format conversion, specified by header cards, and converts data to magnetic tape at a rate of 450 cards per minute. Similar programs have been developed for conversion from one magnetic tape system to another. If there is a conversion in both the code representation of the data and in the format, but not in the number base, the system can convert 4-, 5-, 6-, 7-, or 8-bit characters from one form to another at a rate of approximately 3,000 characters per second. Conversion can be made from 48-bit binary words to decimal digit words at a rate of approximately 16 words per second. Conversion can be made from 13-digit decimal words to binary words at rates in excess of 50 words per second.

Engineering Considerations

The Universal Data Transcriber is being designed and constructed at the United States Naval Proving Ground, Dahlgren, Virginia. Subcontractors are providing the memory, logical building blocks, and various specialized input and output circuitry. The system which is currently being assembled is scheduled to become operational in the first quarter of 1959 and will consist of the central computer, a magnetic tape transport, an IBM *514* used as a card punch, an IBM *407* used as a printer, a modified IBM *101* used as a card reader, and a Flexowriter. The computer with memory, level conversion circuitry, plugboard, power supplies, and console will occupy 5 relay racks.

The logical building blocks are all transistorized megacycle SEAC type circuitry built by Computer Control Company. Some of these are being modified to provide 2-phase operation where the extra speed is required. The memory is an all-transistorized magnetic core memory with a full read-write cycle time of 10 microseconds, and operates in parallel on a 36-bit word or 4 characters of 9 bits each. The 80-brush reading station of the IBM *101*, used as a 450-card per minute reader, will load the data from a row in the card in parallel into a magnetic shift register which will be shifted into the computer on four wires in 600 microseconds. A similar circuit will be used on the second reading station so as to provide a check on the reading. Data are punched into IBM cards at 100 cards per minute by serially shifting, one bit at a time, at a 100,000 cycle shift rate, the 80 bits in the row to be punched. This shift register will pick up relays which will control the punch magnets in an IBM *514*. The reading station which follows the punching station will be equipped with magnetic shift register for reading back the data from the punched card for a check. The same shift register and relays which are used in punching is 120-bits long so that it can be used to control the printing on an IBM *407*. A Flexowriter is permanently attached to the system to provide communication between the computer and the operator and is used as an input for the program tapes, and as an input or output of 5-, 6-, 7-, or 8-channel paper tape.

Conclusions

The Universal Data Transcriber is a very flexible data conversion system. This system not only allows conversion of data from almost any digital source into an appropriate form for a general-purpose computer but also reduces the computing time required in the general-purpose computer, since most of the format conversion and editing will be performed in the Universal Data Transcriber.

Appendix I. Order Codes

An instruction word consists of 32 bits. 13 bits are used to designate the address of the operand; this is called the address field and is designated by A. Three bits specify which of the six B-registers, $B1$, $B2$, $B3$, $B4$, $B5$, or $B6$, (also called B box, index register, or address modifier) is to be used. In all cases the contents of the selected B-register (8 bits) is overlaid, forming the logical sum, with the low-order 8 bits of the A field before the address is used. Eight bits are used to designate the limit value of the contents of the selected B-register to control some conditional transfers. Eight bits, divided into 3 digits, are the command field. The high-order digit specifies the code group 0, 1, 2, or 3. An instruction word is assembled from 4 characters as indicated.

The prime symbol "′" is used to designate "the contents of." For example, a reference to A refers to the 13-bit address field itself, but a reference to A' refers to the character stored at address A. In all cases, unless specifically stated to the contrary, when data is transferred from one register to another the data remain in the source register and the receiving register is cleared to all zeros before, or as, the data are received.

In writing the orders in the following groups the subject matter is consistent with the concepts expressed in the following definitions.

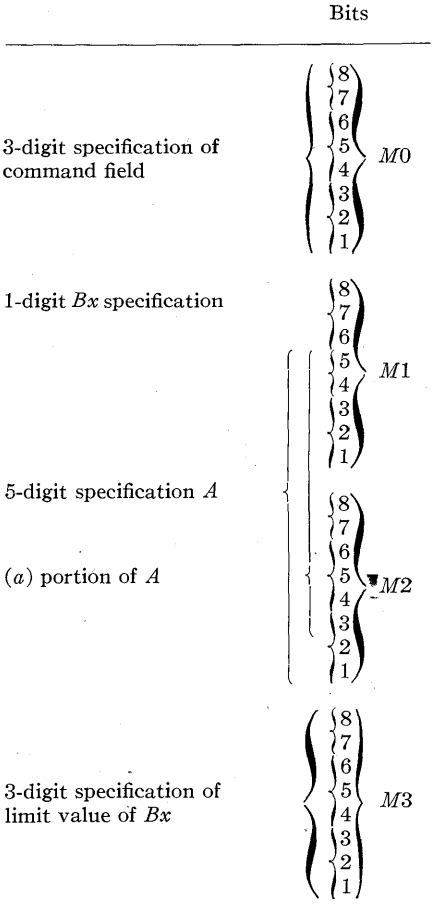
(a) = The 11 high-order bits in a 13-bit address in the instruction register in which the 2 low-order bits are always zero. (a) identifies the location of a specific instruction stored in memory.

A = The 13-bit address in the instruction register which identifies a specific location in the memory.

A' = The 8-bit character made available by access to the memory at address A.

Bx' = The 8-bit character made available by access to that B-box which is designated as Bx.

The address A may be modified by overlaying it with Bx'. A' will then identify the 8-bit character made available by access to the memory at the modified address A.



Group 0

The execution time of orders in this group is 20 microseconds. All A addresses may be overlaid with the contents of any one of six B registers (in the recirculating ring) by specifying Bx in the B-field of the instruction. The value of x may be: 1, 2, 3, 4, 5, or 6 to overlay A with the contents of $B1$, $B2$, $B3$, $B4$, $B5$, or $B6$ respectively.

If $B0$ is specified, there will be no overlay. If $B7$ is specified, the contents of Register $B7$ will not be overlaid on A. Subtract orders do not leave a record of a possible negative sign in the result. Propagate add and subtract orders are used in multiple precision arithmetic.

041 A' to $R1$: clear register 1, transfer A' into register 1.

051 A' to $R1$: add 1 to Bx after modifying A.

021 A' to $R2$: clear register 2, transfer A' into register 2.

031 A' to $R2$; add 1 to Bx after modifying A.

025 Add A' to $R1$: clear register 2, transfer A' into register 2, clear carry storage, add A' to contents of register 1, put sum in register 1, store overflow bit, if any, in carry storage, reset register 2 to all zeros.

035 Add A' to $R1$; add 1 to Bx' after modifying A.

024 Propagate add A' to $R1'$: clear register 2, transfer A' into register 2, add A' to contents of register 1 without previously resetting carry storage, put sum in register 1, store overflow bit, if any, in carry storage, reset register 2 to all zeros.

034 Propagate add A' to $R1'$; add 1 to Bx' after modifying A.

Note: With reference to orders 023, 033, 022 and 032, if the contents of register 1 are greater than or equal to the contents of register 2 (minuend greater than or equal to subtrahend) the result will be a binary difference with final contents of borrow storage equal to zero. If, however, the contents of register 2 are greater than the contents of register 1 (minuend less than subtrahend) there will finally be a one in borrow storage and the true difference will be equal to the contents of register 1 minus 2^8. In this latter case, by subtracting the final contents of register 1 from zero, the absolute value of the difference between the minuend and the subtrahend can be obtained.

For multiple-precision arithmetic the nonreset of the borrow in orders 022 and 032 is related to preceding orders so that the final presence or absence of borrow refers to the entire sequence of operations. The true difference may be equal to the contents of register 1 minus 2^{16} or 2^{24}, etc.

023 Subtract A' from $R1'$: clear register 2, transfer A' into register 2, clear borrow storage, subtract A' from contents of register 1, store the borrow bit, if any, in borrow storage, store the answer in register 1, reset register 2 to all zeros. (See note.)

033 Subtract A' from $R1'$; add 1 to Bx' after modifying A. (See note.)

022 Propagate subtract A' from $R1'$: clear register 2, transfer A' into register 2, subtract A' from contents of register 1 without resetting borrow storage, store the final borrow bit, if any, in borrow storage, store the answer in register 1, reset register 2 to all zeros. (See note).

032 Propagate subtract A' from $R1'$; add 1 to Bx' after modifying A. (See note.)

026 Logical product (extract) A' and $R1'$: clear register 2, transfer A' into register 2, form the logical product, bit by bit, of A' and contents of register 1, store the result in register 1, clear register 2.

Example:

Contents of register $1 = R1' = 10101010$
Contents of register $2 = A' = 00001111$

Logical product $= 00001010$

036 Logical product (extract) A' and $R1'$; add 1 to Bx' after modifying A.

027 Add A' to $R1'$ with no internal carry: reset carry storage, clear register 2, transfer A' into register 2, add A' to contents of register 1 with no internal carry propagation during adding, store the result in register 1, clear register 2. The presence of a final result in register 1 is an indication that the original operands were not equal.

Example:

Original $R1' = 01011101$
$R2' = A' = 10111101$

Final $R1' = 11100000$

037 Add A' to $R1'$ with no internal carry; add 1 to Bx' after modifying A.

020 Overlay $R1'$ with A' and put the result in $R2$: reset register 2, overlay $R1'$ with A' and put the result in register 2.

030 Overlay $R1'$ with A' and put the result in $R2$; add 1 to Bx' after modifying A.

040 Store $R1'$ in A: reset the memory at the address A, transfer the contents of register 1 into memory at address A.

050 Store $R1'$ in A; add 1 to Bx' after modifying A.

060 Store $R2'$ in A: reset the memory at the address A transfer the contents of register 2 into memory at address A.

070 Store $R2'$ in A; add 1 to Bx' after modifying A.

011 Add 1 to Bx' and program transfer to (a) if $A \neq 0$ and final $Bx' \neq M3'$: add 1 to the contents of Bx and store the result in Bx. The program will proceed to the next instruction unless $A \neq 0$ and the final value of $Bx' \neq M3'$. For this condition there will be a program transfer to the instruction in address (a).

010 Add $R1'$ to Bx' and program transfer to (a) if $A \neq 0$ and final $Bx' = M3'$; add the contents of register 1 to the contents of Bx and store the result in Bx. The program will proceed to the next instruction unless $A \neq 0$ and the final value of $Bx' \neq M3'$. For this condition there will be a program transfer to the instruction at address (a).

044 Shift the 4 low-order bits of $R1'$ into the relay register and program transfer to (a) if $A \neq 0$ and $Bx' \neq M3'$: right shift the 4 low-order bits of Register 1, bit by bit, into the high end of the 120 bit relay shift register at a rate of 5 microseconds per bit. The original four high-order bits of register 1 will remain in register 1 as the four low order bits. The program will proceed to the next instruction unless $A \neq 0$ and $Bx' \neq M3'$. For this condition there will be a program transfer to the instruction at address (a).

Group 1

The low order digit used in codes in this group is designated by the octal digit x. (x may have the value 0, 1, 2, 3, 4, 5, 6 or 7.) In all cases the number of shifts specified by x and the number of bits which may be thereby counted is equal to $x+1$. All shifts are to the right, that is: high order bits are shifted toward lower order positions. For all orders in this group the program will proceed to the next instruction unless $A \neq 0$ and $Bx' \neq M3'$. For this condition there will be a program transfer to the instruction at address (a). The execution time for orders in this group is 10 microseconds if $A = 0$ and 11 microseconds if $A \neq 0$.

$11x$ Right shift register 1, $x+1$ places: the bits shifted out of register 1 are lost.

$12x$ Right shift register 2, $x+1$ places: the bits shifted out of Register 2 are lost.

$13x$ Right shift the contents of register 1 into register 2: right shift the contents of both register 1 and register 2 so that $x+1$ low-order bits of register 1 are moved into the high-order end of register 2. The $x+1$ low-order digits of register 2 are shifted out and lost.

$14x$ Shift the contents of register 2 into register 1: right shift the contents of both register 2 and register 1 such that $x+1$ low-order bits of register 2 are moved into the high-order end of register 1. The $x+1$ low order bits of register 1 are shifted out and lost.

$15x$ Count bits: right shift the contents of register 1, $x+1$ places, inceasing the contents of counter P by the number of binary ones shifted out of register 1. Counter P counts modulo 8.

Group 2

224 Transfer the 8-low-order bits of A to Bx: reset Bx, transfer the 8-low-order bits of A in the instruction register into Bx. The execution time for this order is 14 microseconds.

225 Transfer the contents of register 1 into Bx: reset Bx, transfer $R1'$ into Bx. The execution time for this order is 10 microseconds.

213 Program transfer to (a) if $Bx' \neq M3'$: the program will proceed to the next instruction unless $Bx' \neq M3'$. For this condition there will be a program transfer to the instruction at address (a) regardless of whether $A = 0$ or $A \neq 0$. In this program transfer, (a) is not overlaid with the contents of Bx. The execution time for this order is 11 microseconds.

Note: For all of the following orders in Group 2, the program will proceed to the next instruction unless $A \neq 0$ and $Bx' \neq M3$. For this condition there will be a program transfer to the instruction at address (a). The execution time for the following orders in Group 2 is 10 microseconds when $A = 0$ and 11 microseconds when $A \neq 0$.

221 Set Iy: set the condition of all 8 Iy latches in conformity with the information in register 1. There is a latch for each bit in register 1 and this latch is turned on for a "one" condition in register 1 and turned off for a "zero" condition in register 1.

241 Set counter P to zero.

223 Transfer contents of input register to register 1: reset register 1, and then transfer the contents of the input register into register 1.

226 Transfer contents of Bx to register 1: reset register 1, transfer the contents of Bx into register 1.

232 Transfer the contents of counter P into the input register: reset the input register, transfer the contents of counter A by way of the plugboard into the input register. The plugboard allows flexibility in the location, the bits transferred and the sense of the bits transferred into the input register.

230 Input K to input register: reset the input register and transfer 8 bits by way of input K into the input register. Input K is connected to its signal source through the plugboard.

231 Input J to input register: reset the

input register and transfer 8 bits by way of input J into the input register. Input J is connected to signal source by plugboard.

260 Transfer the contents of register 2 to the output register by way of L: reset the output register, transfer the contents of register 2 to the output register by way of the plugboard and entry terminal L of the output register.

261 Transfer contents of register 2 to the output register by way of M: reset the output register, transfer the contents of register 2 to the output register by way of the plugboard and entry terminal M of the output register.

Group 3

The execution time for orders in this group is 10 microseconds except for order 34x. In the jump orders the eight low-order bits of A may be overlaid with the contents of Bx. When the low-order digit of the code is designated by x, x can have the value of 0, 1, 2, 3, 4, 5, 6, or 7.

310 Jump to (a): unconditionql program transfer to the address (a) specified in the instruction register.

317 Jump to (a) and stop: unconditional program transfer to the address (a) specified in the instruction register and stop before the execution the new instruction.

311 Jump if contents of register 1=0: conditional program transfer to the address (a) specified in the instruction register if the contents of register 1=0. If contents of register 1≠0, go to the next instruction.

312 Jump if contents of register 2=0: conditional program transfer to the address (a) specified in the instruction register if the contents of register 2=0. If the contents of register 2≠0, proceed to the next instruction.

313 Jump if contents of counter P=0: conditional program transfer to address (a) specified in the instruction register if the contents of counter P=0. If contents of counter P≠0, proceed to the next instruction.

314 Jump if contents of carry storage=1: conditional program transfer to the address (a) specified in the instruction register if the contents of carry storage=1. If contents of carry storage=0, proceed to next instruction.

315 Jump if contents of borrow storage=1: conditional program transfer to address (a) specified in the instruction register if contents of borrow storage=1. If contents of borrow storage=0, proceed to the next instruction.

32x Jump if indicator latch Ix is in "one" condition: conditional program transfer to address (a) in the instruction register if the indicator latch is in condition "one." If indicator latch is in condition "zero" proceed to next instruction.

33x Set indicator latch x to "one" condition.

37x Reset indicator latch x to "zero" condition.

34x Execute plugboard controlled instruction x.

35x Sense on breakpoint switch x. The breakpoint switch is not shown in the circuit diagram. If breakpoint switch x is in "off" position, proceed to the next instruction. If the breakpoint switch x is in "transfer" position use the address (a) specified in the instruction register for the address of the next instruction. If the breakpoint switch x is in "stop" position, prepare to use the address (a) specified in the instruction register for the address of the next instruction and stop.

A Universal Computer Language Translator

R. B. BONNEY
NONMEMBER AIEE

CAN Univac talk to Bizmac? As more and more electronic data-processing systems are put into operation, the need for rapid and efficient information interchange becomes important. Many organizations have evolved data-processing systems using a combination of different computer types, together with a variety of input and output equipment. The use of different computers in a single organization does not necessarily represent poor planning, since the requirements of one group may dictate the use of a particular type of computer which is inadequate for the requirements of a different group.

To achieve efficient utilization of an electronic data-processing system, it is often desirable to transfer output data from one computer to a different computer for additional operations.

The need for interchange between different organizations is also beginning to develop, and can be expected to increase as electronic data-processing spreads. Acceptance of payroll deduction reports on International Business Machines Corporation (IBM) *705* magnetic tapes by the Treasury Department and the Social Security Agency are typical examples.

Transformation from one form of data to another is another common requirement. Conversion from magnetic tape to punched paper Teletype tape is representative of a wide variety of such requirements.

Unfortunately, the designers of electronic computers have made very little attempt at standardization. In nearly all cases, the recording format and coding used by the various computer manufacturers are sufficiently different to prevent direct interchange of data. Undoubtedly, the very rapid development and growth of the computer industry have been major factors in the lack of standardization.

The commonly used method of accomplishing data translation between computers up to now has been by means of punched cards, and in some cases punched paper tape. Translation by these methods is seriously hampered by the speeds of operation. Some special-purpose high-speed translation equipment has been built for direct translation of data on magnetic tape. However, because of the wide variation in requirements, the equipment developed for a specific application ordinarily cannot be used in a different application without modification or complete redesign.

Realization of the need for "universal" translation capability resulted from work on data-handling systems for scientific test data. For this purpose, a computer format control buffer was developed to accept unsynchronized, continuous digital input data which are to be converted to magnetic tape, in a format directly usable by a computer.

In discussing other applications of this equipment, it soon became evident that no two applications were alike, and that too much re-engineering would be required to adapt the equipment to each new situation. It was, therefore, decided that an attempt should be made to develop a system with the flexibility necessary to meet all commonly encountered translation requirements without becoming overcomplex and economically impractical. The Computer Language Translator is the result.

In considering all possible applications, there is a wide variety of requirements

R. B. BONNEY is with Electronic Engineering Company of California, Santa Ana, Calif.

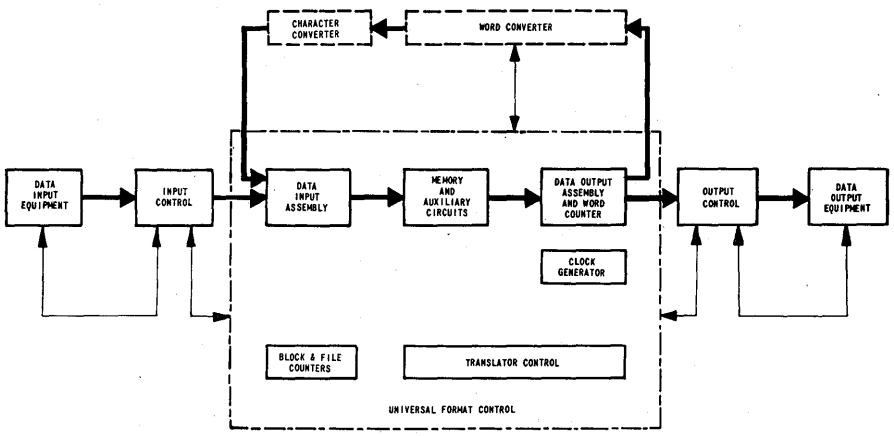


Fig. 1. Basic block diagram of Computer Language Translator (ZA-100)

which must be fulfilled by the translator equipment. A major undertaking in connection with this development has been the assembly of the myriad of details required for each computer format and for the numerous input and output devices which are expected to operate with the Translator. Although the equipment manufacturers have generally cooperated admirably, the problems in sorting out and verifying the necessary details have been considerable. The available literature is helpful but ordinarily does not give sufficient information, and it is difficult to locate the individual who can provide all of the answers.

Another problem has been to incorporate the necessary intelligence into the equipment without the Translator growing into a full-fledged digital computer.

The reader may ask, "Why not use one of the smaller general-purpose digital computers rather than develop new equipment specifically for translation?" There are a number of answers to this question:

1. All of the smaller computers currently available use magnetic drum memory, and as a result their speed of conversion would be much too low.

2. A magnetic drum is usually not suitable for delivering synchronous output data at a rate different from the input. This is one of the most fundamental translation requirements.

3. A large part of the translation process is involved with control of the data input and output devices. A general-purpose computer cannot provide these control features.

The required flexibility has been provided by means of the "building block" modular technique. Thus, by providing the proper combination of modules, the needs of a particular data processing system can be fulfilled without providing an overcomplicated system, containing excess features and equipment.

The design has been worked out around a foundation building block called the universal format control. Tliis portion of the system is common to all applications. It contains a buffer memory, operator's controls, counters, and power supplies required for most operations.

Functions which are peculiar to a particular type of data input or output device are treated separately through the use of input and output control units. Ordinarily a separate control unit is required for each different type of data to be handled. This works out to be feasible from an equipment cost standpoint, since most input or output controls represent less than five % of the cost of the central universal format control.

Fig. 1 is a block diagram of the basic system. For multiple function systems, the universal format control can be shared between several input and output controls through the use of patchboards as shown in Fig. 2.

The operation of the Computer Language Translator can be divided into the two general categories of format conversion and code conversion.

Format conversion is concerned with the transformation of the physical characteristics of the input data to the required output form. Format conversion may involve radically different forms of input and output such as the translation of punched cards or punched tape to magnetic tape. On the other hand, the format conversion may require the translation of one magnetic tape to another magnetic tape with different recording speed and geometry.

Code conversion may or may not be required, depending on the application. Both character and word coding can be handled. Following the building block concept, the code converters are provided as auxiliary equipment when required, rather than as a part of the basic system.

In character conversion, the coding of each character is changed, as for example binary coded decimal to excess-three.

Word conversion involves reordering the characters within a word and transposition of the sign character where necessary.

Modification of the word order within a data block would be helpful in some cases. However, this feature has not been included in the interest of simplification. Deletion of certain words or characters from each block can be handled so long as the order remains fixed for any one operation.

In the normal operating mode, data is transferred from the input device to the buffer memory with the output equipment stopped. The amount of data entered can be arbitary, up to the capacity of the buffer memory (1,092 or 2,184 characters). On completion of loading, the input device is stopped, the output device is started, and the information is transferred from the buffer memory to the output.

In general, it is not practical to operate with the input and output devices running

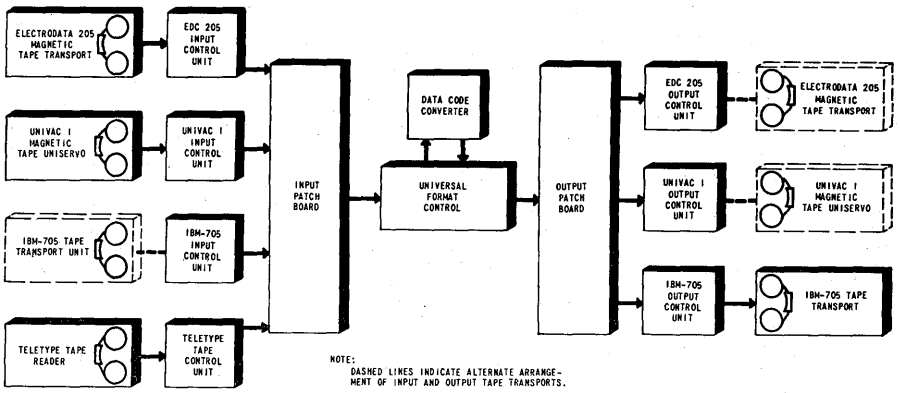


Fig. 2. Typical multiple function Computer Language Translator system

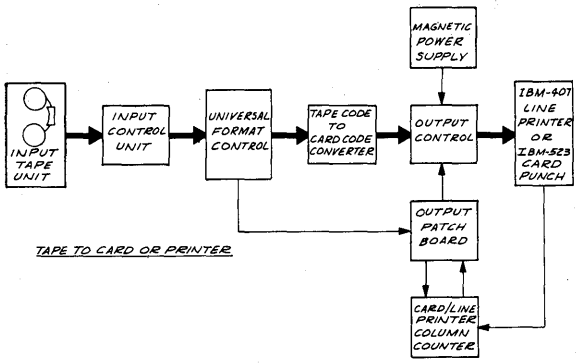


Fig. 3 (left). Translator system for magnetic tape to punched card or line printer

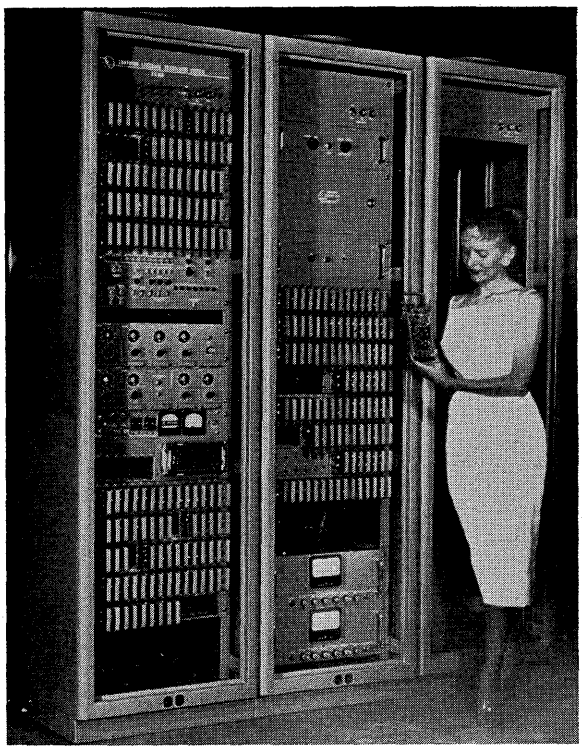


Fig. 4 (right). Computer - language Translator equipment

simultaneously. Unless the input and output rates are synchronized, the buffer memory will eventually overflow or empty. Since the input and output rates are normally different, the control logic is greatly simplified by using separate load and unload cycles.

In cases involving word conversion and for some other conversion operations, a recirculation cycle is introduced between the input and output cycles. This is illustrated in Fig. 1. On completion of transferring the input data to the memory, the recirculation cycle is initiated. During this cycle, the data is unloaded to the word converter one word at a time. The word converter is in turn unloaded, through the character converter, back into the buffer memory. At the completion of the recirculation cycle, all of the data will be in the required output coding. The memory can then be unloaded as a complete block at the rate required by the output device.

Insofar as possible, the equipment has been designed to eliminate the need for correlation of the input and output data formats.

The output word structure and block length can be entirely different from the word and block structure of the input data. In cases where fixed word length or block length output is required, zeros or other fill characters can be inserted into each word, and zeros or fill characters will automatically fill out to the proper block length. Several short input blocks can be compressed into a single output block.

Elimination of dependence between input and output format is also achieved by entering control characters into the memory. For example, when operating with variable input block length, an "end of data" character is entered into the buffer memory at the end of the input cycle. This special character is detected at the memory output and used to control the output cycle. The use of control characters thus eliminates the need for counting circuits to determine the output block length. The special characters, in most cases, do not appear in the output data.

It is doubtful that a detailed functional block diagram of the Computer Language Translator would be of interest to most readers. The function of each component part may not be obvious from the block diagram without a step-by-step detailed description which is beyond the scope of this paper. It is felt rather that a brief listing of the functions and characteristics of the system will assist in illustrating the design philosophy.

As stated previously, all requirements peculiar to a particular input or output device are handled in the associated input or output control units. This results in much greater flexibility, since the designs for new types of control units can be completed individually without involving the majority of the system.

Input to the system can be in the form of: magnetic tape, punched paper tape, punched cards, analog to digital converter register, or other "on-line" source.

Output can be in the form of: magnetic tape, punched paper tape, punched cards, line printers, "on-line" registers for use by digital to analog converters, or other similar applications.

A rather formidable list of requirements for input and output control units results if all likely translator applications are considered. The situation is further complicated by various optional equipment combinations which may be desirable for various reasons. As an example, magnetic-tape data can be accepted directly from an IBM *727* tape unit, or *727* tapes can be played into the system using an Ampex *FR-307* or *FR-407* digital tape transport which is an integral part of the Translator. A somewhat different control unit is required for each case.

To date approximately 25 input and output control units have been designed or are in process. These include control units to handle magnetic tapes for IBM, ElectroData, and Remington Rand computers. Also included are control units for punched paper tape, IBM punched cards, and line printers.

By using the central portion of the system to fulfill more than one operational requirement, the cost per function decreases as additional operations are added. The central universal format control represents between 50 and 75% of the total equipment for most systems. The universal format control is designed to operate with any input or output control unit. Thus, the cost of adding an additional function to the system is essentially the cost of the input or output control unit plus any associated patchboard wiring.

To date Electronic Engineering Company has been reasonably successful in isolating the universal format control from special input or output data requirements. Anticipating the need for

considerable flexibility, circuits needed to adapt the universal format control to different translation applications have been brought out to external terminals. By providing the necessary control signals to these terminals, any requirement within the basic system capability can be handled. In some cases, auxiliary equipment is designed to provide the necessary controls when they are not directly obtainable from the data input or output equipment.

All normal operator's controls are located on a panel housed in the universal format control. For magnetic tape input, provision is made for positioning the tape both automatically and manually in either forward or reverse direction. Indicating counters keep track of the block or file position of the tape.

The actual translation operation can also be controlled automatically or manually.

Other controls and indicators are associated with marginal testing and parity-error control circuits. All controls are interlocked to prevent improper operation.

Character parity is checked at the input before entering the buffer memory and also at the output of the memory. A third parity-check circuit is also provided for output parity checking when possible, as for example with the IBM *729* tape unit. At the operator's option, the controls can be set to automatically stop the operation if a parity error is detected.

Longitudinal (row) parity, if used, can also be checked by means of an optional module.

Marginal operation checks are provided for both the buffer memory and the d-c supplies for the balance of the system.

A common translation requirement is for off-line conversion from magnetic tape to punched cards or line printer and from punched cards to magnetic tape. The multipurpose features of the Computer Language Translator work to advantage here, since the basic system can be adapted to handle these off-line functions in addition to tape-to-tape or other requirements.

As anyone knows who has considered the problem, existing card-handling equipment and line printers are not easily adapted to magnetic tape. This is because the most commonly used equipment handles the data on a row-by-row rather than a column-by-column basis. Some column-by-column card-handling equipment is available, but it is too slow in most cases. A number of high-speed column-by-column readers are under development, but as yet are not generally available. The straight-forward approach to handling row-by-row cards is to provide a complete card size or printer line buffer memory which can be unloaded or loaded on either a row or column basis. Because of the number of parallel outputs involved (120 in the case of a line printer), this type of memory would be a relatively expensive addition to the translator system.

It is possible to utilize the buffer memory in the basic translator to handle cards and line printing by operating the memory in a manner equivalent to a magnetic drum. Using this approach, the stored information is recirculated in the same manner as discussed previously for word conversion.

Fig. 3 is a block diagram illustrating the method used for punching cards or controlling a line printer such as the IBM *407*.

Data corresponding to one card row or one line is first loaded into the memory from the input equipment. The memory is then recirculated 12 times in synchronism with the 12 card row positions. During each circulation cycle, a decoder detects characters which correspond to the row position and are to be punched or printed.

A column counter, and associated set-up patchboard, sets up the proper punch magnet or print magnet. Since this same counter controls unloading and reloading the memory, the required synchronization is maintained between the memory and the punch or printer columns. A similar approach is used for reading punched cards. Each row of the card is scanned sequentially by the column counter. A card-to-tape coder enters the proper character into the memory when a punch is detected.

Card reader verification can be handled by double reading. This necessitates a reader with two reading stations. Verification is done during the second reading by comparing the reader output against the decoded data entered into the memory during the first reading.

Card punching can be verified in a similar manner. In this case, the punched card is read following the punching cycle. The information from the reader is compared to the decoded output from the memory.

The translator is designed around a combination of plug-in vacuum tube circuits, diode logic circuits, transistor-driven ferrite cores, and in some cases tape wound cores. Some consideration was given to eliminating vacuum tubes entirely. However, the idea was not carried out because of the considerable amount of design knowledge and proven reliability of the vacuum tube plug-ins. These same components have been thoroughly proven by use in many other similar applications. On the other hand the transistor-driven ferrite core memory was used since this unit was originally developed using transistors. Tape wound cores are being used for the card and line printer column counter because of the relatively large linear count and number of output circuits involved.

In general, d-c logic is employed. Conservative design policies have been established in the interest of achieving high reliability. Active circuits are isolated from external loading. Definite rules for interconnected logic circuits are followed.

Standards are maintained for component parts and workmanship which are somewhat higher than are often found in commercial electronic equipment. Transformers, for example, are hermetically sealed. Other components are of top quality. It is sincerely believed, however, that the modest cost for high-quality equipment is more than offset by higher reliability and decreased maintenance requirements.

The basic system is housed in two equipment racks which are each 2 feet square and 7 feet high. One or more additional racks house input and output control units, patch-boards, auxiliary power supplies and, in some cases, magnetic tape transports.

Fig. 4 shows a typical translator system. This system was demonstrated at the Western Joint Computer Conference. The input to the system comes from punched paper Teletype tape or from ElectroData *205* magnetic tape. The output is in IBM *727* format. The input tape unit is the ElectroData transport used with the *205* Datatron computer. The output tape unit is an Ampex *FR-307* tape recorder with IBM head dimensions.

Until more varied application experience has been obtained, it will be difficult to assess the success or failure which has been achieved in developing a truly universal Computer Language Translator. Experience to date, based on actual applications and discussions with many potential users, has been very encouraging.

A Computer Oriented Toward Spatial Problems

S. H. UNGER
NONMEMBER AIEE

THE STORED-program digital computer has been in existence for about a decade and has proven itself to be a powerful and versatile instrument. Roughly speaking, any solvable problem can be solved by a digital computer. At many tasks, such as the solution of systems of linear equations, these machines are thousands of times as fast as human beings.

However, there are certain tasks, which might be termed spatial problems, at which digital computers are relatively inept. For example, it seems to be extremely difficult to write chess playing programs that will enable a computer to compete successfully with a capable human opponent.[1] Pattern recognition is another area in which present-day machines cannot match the performance of their designers. References 2, 3, and 4 describe some important and interesting work done in this field.

The difficulty in such cases appears to be that these machines can actively cope with only a small amount of information at any one time. This circumstance is aptly illustrated by the title of an article by A. L. Samuel, "Computing Bit by Bit," which appeared in the computer issue of the *Proceedings*, Institute of Radio Engineers. It appears that efficient handling of problems of the type previously mentioned cannot be accomplished without some form of parallel action.

An important implication of this argument is that machines of greater complexity are needed. This, however should not be regarded as an insurmountable barrier, since it is likely that great advances will be made in the components field during the next decade.

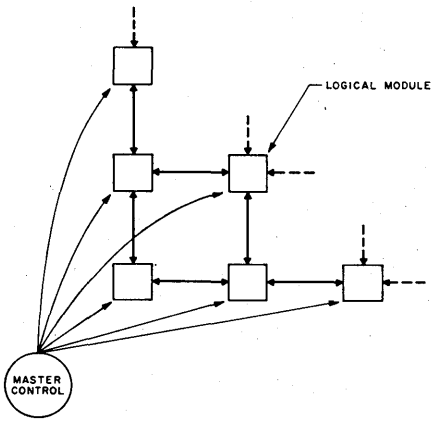


Fig. 1. Organization of spatial computer

The principal objective of the research reported on here is to learn how to build machines that can efficiently solve problems not well suited to solution by conventional digital computers.

The approach in this paper is to attempt to describe a computer which is inherently well matched to one such problem (pattern detection), hoping that such a computer, probably in an improved version, will then perform well when faced with a much larger class of problems.

General Structure of the Computer

Fig. 1 illustrates the form of the proposed machine. It consists of a master control and a rectangular array of modules. Each module communicates with its four immediate neighbors and receives orders from the master control. The master control cannot address the modules individually, but issues general orders which go to all of the modules.

A module consists of a one-bit accumulator, a small amount of random access memory, perhaps six bits in one-bit words, and some associated logic. Inputs to each module come from the master control and from the accumulators of the modules above, below, to the left, and to the right of it. It will also be assumed that an input signal may be fed directly to each accumulator from outside the machine.

The master control includes a random access memory for storing instructions, a clock, and decoding circuits. It acts like the operation decoding section of a conventional digital computer, reading out instructions from memory in sequence, decoding them, and sending appropriate control voltages out on a set of buses feeding the modules.

A logical adder (or-gate) with an input from each module accumulator tells the master control if all of the accumulators have zeros in them, and thereby makes possible the use of a transfer on zero order. This instruction, analogous to the conditional transfer orders used in ordinary digital computers, tells the master control to skip to the instruction addressed by the transfer zero order if there are no ones in any of the module accumulators. It is the only decision order used in this computer.

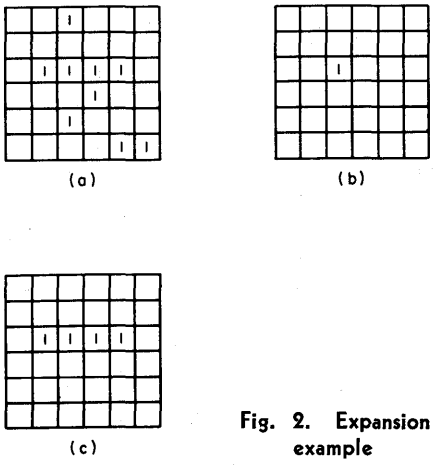


Fig. 2. Expansion example

Order Structure

The order structure is summarized in Table I. The transfer on zero order (trz x) has just been described. In addition there is an unconditional transfer (tr x) which simply orders the execution of the instruction at address x.

As was mentioned before, each module has a small number of individually addressable one-bit memory elements. Within a module these registers will be referred to as a, b, c, etc. The instruction store b (st b) tells each module to store the contents of its accumulator in the bth memory register without altering the accumulator contents. Similarly write b (wr b) orders the contents of each b register to be written into the accumulator (without changing the contents of b).

Information can be transmitted directly from one module to another by means of a shift instruction. A shift right (sR) order for example, causes the contents of each accumulator to be transferred to the accumulator of the module to the right of it. Shift left (sL), shift up (sU) and shift down (sD) have analogous meanings. When shift orders are used assume the matrix is bordered by modules having zeros in their accumulators.

The invert order (in) causes the contents of each accumulator to be comple-

S. H. UNGER is with Bell Telephone Laboratories, Inc., Whippany, N. J.

The idea of considering switching circuits in n-dimensions came to the author through the work of Professor D. A. Huffman of M.I.T. Thanks are due to C. Y. Lee and M. C. Paull of the Bell Telephone Laboratories for many interesting conversations which contributed directly and indirectly to the work presented here, and for their advice regarding the preparation of this report.

Table I. Order Structure

Order	Abbreviation	Meaning
Transfer to instruction x.	tr x	Execute instruction x next.
Transfer on zero to instruction x	trz x	Execute instruction x next if there are no ones in the field. Otherwise continue in the normal sequence.
Invert	in	Change the contents of all the accumulators.
Add	add	Add logically to the contents of the accumulator the contents of the specified memory cells and neighboring accumulators. Leave memory cell contents unchanged.
Multiply	mpy	Multiply logically the contents of the accumulator and the specified memory cells and adjacent accumulators. Result is placed in accumulator and memory cell contents are left alone.
Add in memory	adm	Add logically to the contents of each of the specified memory cells the contents of the accumulator. Do not change accumulator contents.
Multiply in memory	mpm	Multiply logically the contents of each specified memory cell by the contents of the accumulator without disturbing the accumulator.
Store	st	Store the accumulator contents in each of the specified memory cells without disturbing the accumulator.
Write	wr	Write the contents of the indicated memory cell into the accumulator without altering the contents of the memory cell.
Shift left (or right or up, or down).	sL (sR, sU, sD)	Write the contents of each accumulator into the accumulator to the left (or right or above or below).
Add reference	Add ref	Add a one to accumulator of module in lower left corner of matrix.
Link	Link	Record in memory elements located between adjacent pairs of modules, whether or not ones are contained in both accumulators. (Diagonally touching modules are considered as adjacent for this order and the next one.)
Expand	exp	Add ones to the accumulators of those modules connected through a chain of active intermodule memory elements of the kind specified in the order (horizontal vertical, positive diagonal, or negative diagonal) to some module with a one in its accumulator.
Shift around	sA	Same as shift right except that rightmost modules of each row send contents to leftmost module of row above, lower left module gets bit from input register, and upper right module sends contents to output register.

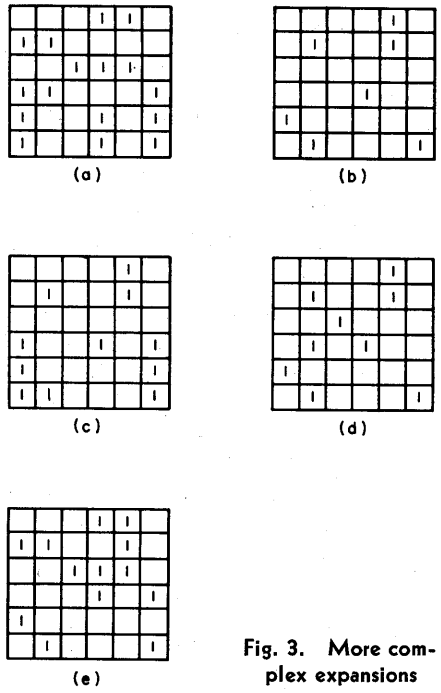


Fig. 3. More complex expansions

mented. Logical addition and multiplication can be carried out among the contents of the accumulator, memory registers, and neighboring accumulators. (Logical addition is defined by the equations: $1+0=0+1=1+1=1$ and $0+0=0$. Logical multiplication is defined by the equations: $1\times0=0\times1=0\times0=0$ and $1\times1=1$.) Add x causes the contents of memory register x to be added logically to the contents of the accumulator. The result is stored in the accumulator and the contents of x are left unaltered. The add left (add L) order causes the contents of each accumulator to become the logical sum of the original contents of that accumulator and the original contents of the accumulator of the module to the left. Add right (add R), add up (add U), and add down (add D) have similar meanings. Any combination of the contents of neighboring accumulators and memory registers may be added to the accumulator contents in the previously mentioned manner using a single order. For example, if it is desired to add to the contents of each accumulator the contents of memory registers a and c, of the same module, and the contents of the accumulator above, then the order add a, c, U would be used.

A completely analogous set of instructions exists for a logical multiplication. Thus (mpy b, D) would cause the original accumulator contents, to be multiplied by the contents of memory registers b and the contents of the accumulator below. The result appears in the accumulator, and the contents of the memory registers remain unchanged.

The order add in memory (adm b) cause the logical sum of the accumulator contents and the contents of register b to appear in b, without altering the accumulator contents. The accumulator contents can be added simultaneously to any number of memory registers by an order (adm, a, b, d) for example. Note that separate additions are made into each memory register, each sum involving the accumulator contents and the contents of one memory register. A similar set of orders exists for multiplication, for example (mpm b, d, etc.).

The add reference order (add ref) causes a one to be added to the accumulator of the module in the lower left corner of the matrix. In contrast to the previously described orders, add reference does not treat all modules alike.

A few new definitions at this point may help to facilitate the discussion. Module accumulators will be referred to as cells; sometimes one-cells or zero-cells, depending on their contents, when their contents are discussed as parts of an overall pattern. The arrays of ones and zeros contained in the accumulators or in a particular set of memory elements, say the b-registers, will be termed fields.

Inputs to the matrix can be made in parallel by associating an input device, a photocell perhaps, with each module. The shift-around order (SA) provides a simple method for serially loading and unloading the matrix. A single bit is transmitted to the accumulator of the lower left module from an external source. At the same time the contents of each cell is transmitted to the cell to the right, as in the shift right order, except for the right-most cell of each row, which sends its contents to the left most cell in the row above. The upper right corner module transmits to an output register. Thus, a new pattern can be loaded into the matrix while the old contents are simultaneously taken out.

In certain pattern-processing operations it is desirable to obtain an output from the master control instead of from the network. For example, in a recognition problem it would be desirable for the machine to identify the input pattern as a particular member of some

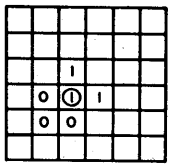


Fig. 4. Lower left corner point

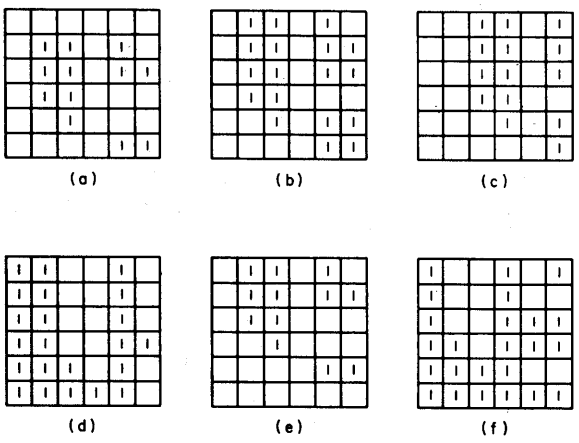
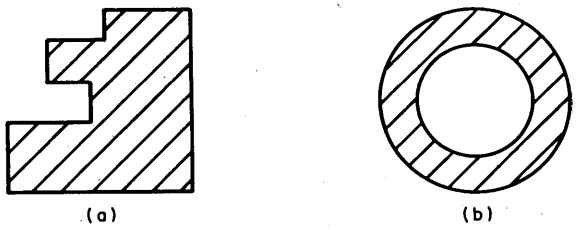
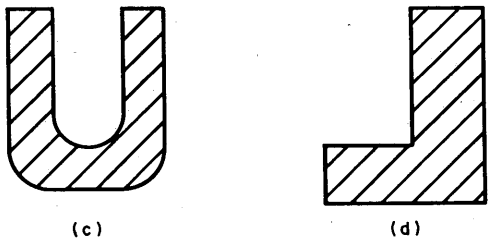


Fig. 5 (left). Detection of lower left corner point

Fig. 6 (right). Vertical concavity





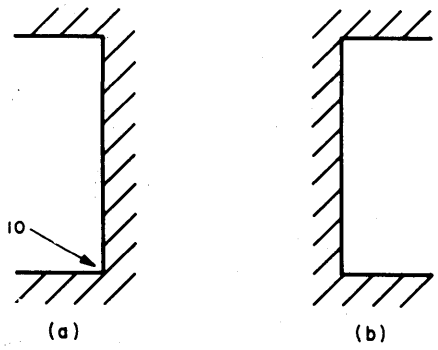


Fig. 7. Characteristic features of vertically concave figures

alphabet, by means of the conditional transfer order and an order enabling the machine to send out pulses on one or more members of a set of output buses.

This paper will not go into further detail regarding input-output problems, since these questions are initimately related to the specific tasks being performed.

The link and expand orders are used together and require a somewhat more elaborate explanation due to the fact that, unlike the previously described orders, they permit the contents of a cell to be directly affected by the contents of arbitrarily distant cells.

Connectivity is an important concept in spatial operations. For example: Given a particular cell of the matrix, called a basis point, it might be desirable to find all points in a given field connected to that cell through a chain of horizontally linked one-cells. See Fig. 2. Part (b) shows the location of the basis point which is to be expanded horizontally (H) over the field shown in (a). The result of this expansion operation appears in (c).

To accomplish this with the link-expand orders one would write the a-field in the accumulators and give the order "link." Then one would write the b-field in the accumulators and give the order "exp H." The same idea can be used to expand a set of basis points over a field. In Fig. 3 part (b) is expanded vertically over part (a) to obtain part (c).

Expansions along diagonal lines can also be defined for either positive or negative slopes (D_P and D_N respectively). The results of a D_P expansion of part (b) over part (a) is shown in part (d) of Fig. 3.

Expansions simultaneously involving several types of connectivity are also possible. For example, it might be desired to consider two points as being connected if it is possible to move from one to the other in single steps which are either diagonal with negative slope or vertical and which pass through one-cells. Thus figure 3(e) shows the results of an H, D_N expansion of part (b) over part (a). Note that the results of an exp H order followed by an exp D_N order is not, in general, the same as the result of an exp H, D_N order. Subroutines using simpler instructions can be written for any expansion, but the importance of this process probably justifies the added circuitry required to shorten their execution times by a substantial amount.

Some Simple Programs

Some elementary programs will now be presented in order to illustrate the use of the orders.

EXAMPLE 1: DETECTING LOWER LEFT CORNERS

Locate all lower left corners in a given field, where a lower left corner is defined as a one-cell with one-cells both above and to the right, and with zero-cells to the left, below, and diagonally (lower left) adjacent, as typified by the circled one in Fig. 4. (In this figure the unmarked cells can contain either zeros or ones.)

The program will be illustrated by referring to Fig. 5, where part (a) is assumed to be the initial field in the accumulator. (In the following sections unmarked cells in the diagrams will be understood to be 0-cells.)

1.	st a	
2.	add D	Result shown in part (b) of Fig. 5.
3.	sR	See part (c).
4.	in	See part (d).
5.	st b	The field shown in part (d) is stored in the b-registers.
6.	wr a	The original field Fig. 5(a) is written into the accumulators.
7.	sU	See part (e).
8.	in	See part (f).

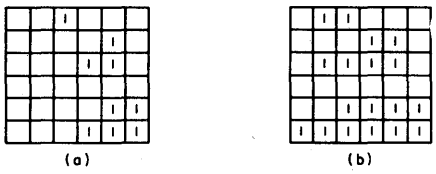


Fig. 8. Line doubling

9. mpm *b* — The product of fields, parts (d) and (f), shown in part (g) appears in the *b*-registers. The ones in this result designate those points in the original field which are oriented with respect to three zeros in the same manner as the circled one in Fig. 4.
10. wr *a* — Again field (a) is written into the accumulators.
11. mpy *U, R, b* — The resulting field, shown in Fig. 5(h) has ones at the two locations which are lower left corners in the original field.

Example 2: Detecting Vertical Concavity

A figure will be said to be vertically concave if it is possible to connect two points on the figure with a vertical line which falls outside the figure at some point between the given points. Parts (a) and (b) of Fig. 6 depict vertically concave figures, whereas parts (c) and (d) depict vertically convex figures.

In this example, a program will be presented for selecting from an arbitrary field all vertically concave figures. Every vertically concave figure has a portion of the form of Fig. 7(a) or of Fig. 7(b) (the vertical part might be only one unit long) and so the problem can be solved by selecting all figures having one or both of these features.

Beginning with the given field in the accumulators, the program is as follows:

1. st *a*
2. sR
3. in
4. mpy *a* — Locates all left edge points
5. link
6. st *b*
7. wr *a*
8. mpy *L, U*
9. sU
10. mpy *b* — Yields lower inside corner points on left edges. See point labelled (10) in Fig. 7(a).
11. exp *V* — At this point the accumulators contain all vertical edges facing left which start from inside corner points as described in 10.

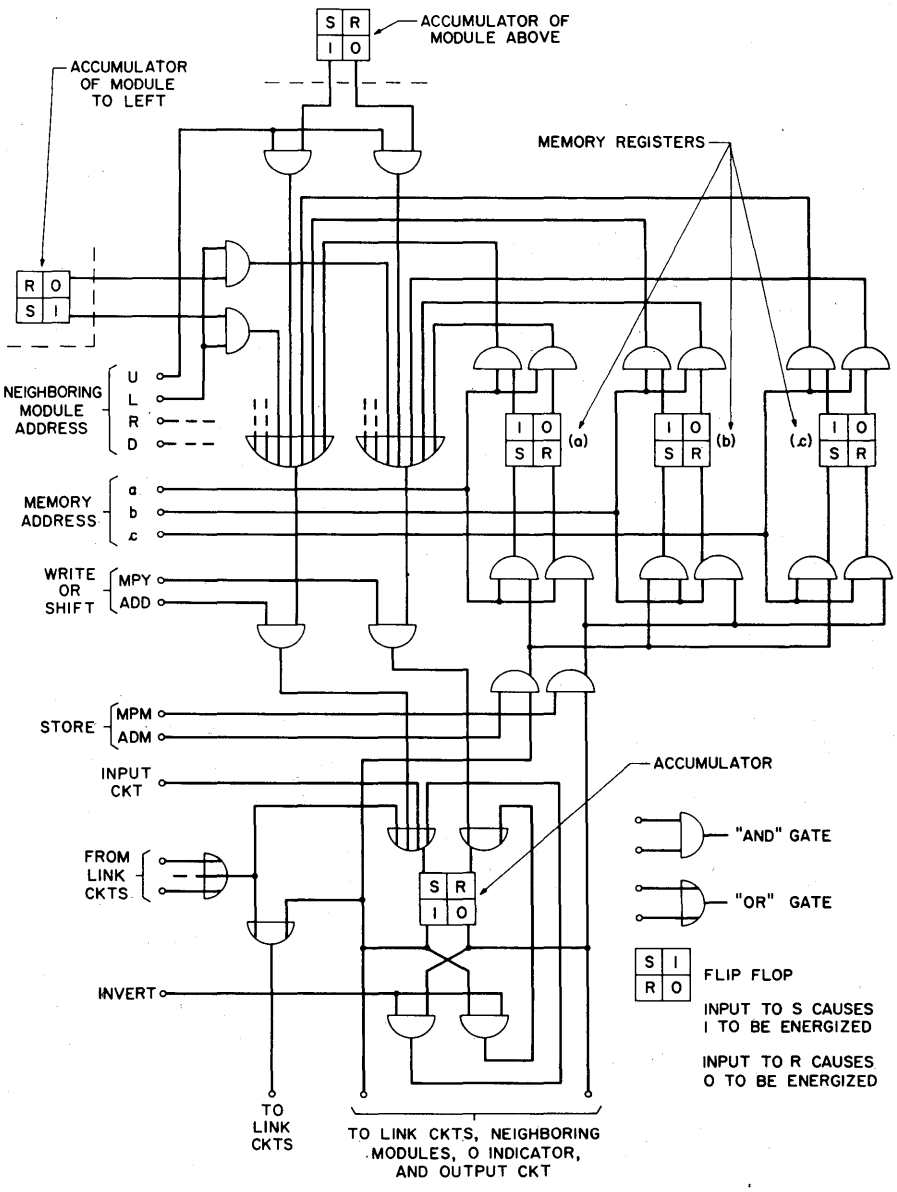


Fig. 9. Module logic

12. st *b*
13. wr *a*
14. mpy *L, D*
15. sD
16. mpy *b* — The accumulators now contain upper inside corner points on left edges that also have lower inside corner points; i.e., points have been found on all figures having the feature described in Fig. 7(a).
17. st *c*
18. wr *a*
19–32. — Repeat steps 2 to 16, interchanging *L* and *R* in all places so as to obtain points in the accumulator corresponding to the feature in Fig. 7(b).
33. adm *c* — In the *c*-registers one now has at least one point on every vertically concave figure.
34. wr *a*
35. link
36. wr *c*
37. exp *V, H, D_N, D_P* — Only vertically concave figures remain in the accumulators.

Example 3: Line Doubling Program

The object of this program is to extend every horizontal line of ones to the left, by its own length. Thus, if the initial field is as shown in Fig. 8(a), the resulting field will be the one in Fig. 8(b).

The program (one of the few which requires a loop) is as follows (the initial field is in the accumulators):

1. st *a*
2. sL
3. adm *a*
4. mpy *R* — Eliminates a point from the right end of each line.

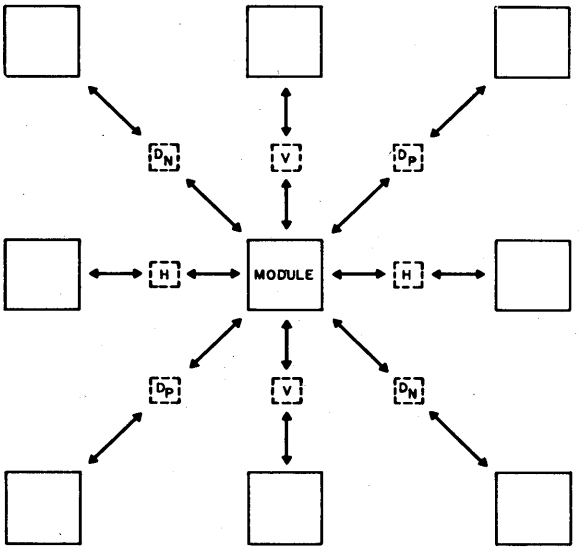
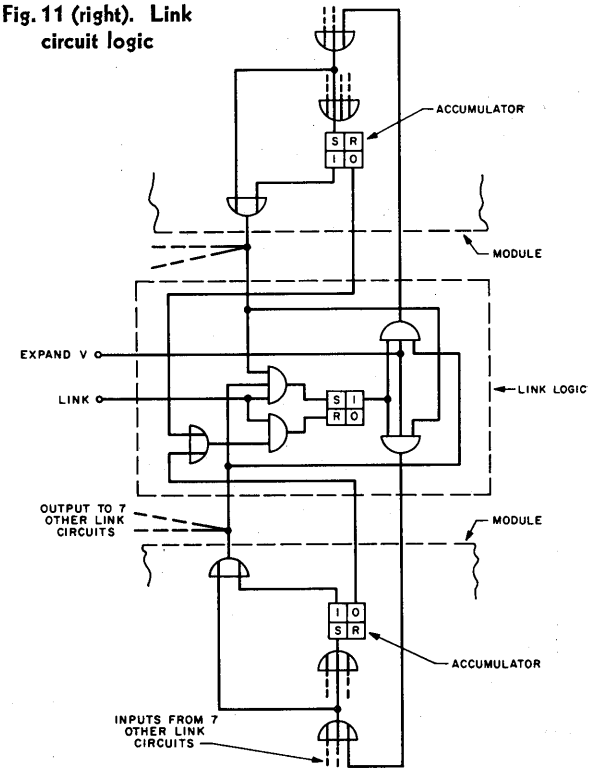


Fig. 10. Arrangement of link circuits

V = vertical link circuits
H = horizontal link circuits
D_P = link circuits on positively sloped diagonal
D_N = link circuits on negatively sloped diagonal

Fig. 11 (right). Link circuit logic



5. trz 7
6. tr 2 — One point has been added to the left end of each line in *a*.
7. wr *a* — The desired field is now in the accumulators.

Logic for the Distributed Computer

A logical diagram for a typical module of a computer realizing the order structure of Table 1 is shown in Fig. 9. Flip-flops are used for the accumulators and for the memory registers. Only three memory registers are shown, but it is easy to see how others can be added.

Five order leads enter the module from the master control. They are add, multiply, add in memory, multiply in memory, and invert. An additional lead from the master control refers to each of the four neighboring modules, and there is also a lead corresponding to each memory register. The instructions (add), (mpy), (mpm), (adm), and invert are given by pulsing the appropriate order lead and the desired set of address leads: these may be memory register addresses or references to adjacent modules. The instructions store or shift are given by pulsing both the (adm) and the (mpm) leads, and the write order is given by pulsing both the add and the (mpy) leads.

Note that with this system of logic the (mpm) and (adm) instructions require no additional logic, and only one additional order lead once the store order has been mechanized. The (mpy) and (add) orders are similarly related to the write order.

In order to carry out the link and expand orders, the direct intermodule connections must be supplemented by special circuitry which will be termed link circuits. These are placed between every adjacent pair of modules. Fig. 10 shows the eight link circuits connecting a particular module to its neighbors. Note that whereas diagonally adjacent modules are not connected by direct lines, they are connected, however, through link circuits.

The logic necessary in the link circuits is shown in Fig. 11. When a pulse is applied to the link terminal from the central control, the flip-flop will be placed in the one-state, if, and only if, ones were present in both of the cells connected by the circuit. (This setting remains unchanged until another link order is given.) If the flip-flop is in the one-state, then a pulse appearing on the expand *V* lead, (consider a vertical link circuit) will cause a one appearing at the output of either of the accumulators of the linked modules to appear at both the input and output terminals of the other accumulator. (Strictly speaking, the ones appear not at the outputs of the accumulator but at the outputs of or-gates driven by the accumulators.) Ones can thus be propagated through the matrix as fast as the logic permits since they can pass through modules without waiting for the accumulators to be activated.

The central control unit consists of a random access memory, a clock, an order decoder, and some input-output control circuitry. No significant departures from the arrangements used in conventional computers are necessary.

Conclusions

The spatial computer described here is a versatile tool for solving problems in which the data to be manipulated occurs in large blocks which are locally correlated. In the field of visual pattern detection, a prime example of this sort of problem, some powerful techniques have been developed for using this machine. These will be reported on at a later date. Other applications of the spatial computer remain to be explored.

The logical scheme, previously presented, for realizing the machine calls for about 170 gate inputs and 11 memory elements per module. This includes the link circuits but not the central control circuitry, and provides for six memory registers per module, a number which has been found adequate for all programs written thus far. Since the number of modules should be at least of the order of several hundred, this means thousands of memory elements and tens of thousands of gate inputs in the matrix alone.

These are alarming figures, and it would probably be difficult to find applications that would justify the cost involved if present day fabrication techniques, requiring individual physical devices for each element, had to be employed. However, progress in the components field is such that it is reasonable

to hope that within a few years there may be available manufacturing processes whereby entire blocks of logical circuitry could be constructed in one unit. Etching, plating, thin film components, multihole ferrite plates, and other techniques now in the development stage may be used to take advantage of the repetitive nature of the modules to sharply reduce costs.

While the author has specifically discussed a computer based on a rectangular matrix of modules using one-bit registers, it might be worthwile to consider other arrangements. Possible variations include matrices in three or more dimensions, registers enlarged to handle multibit words, and polar co-ordinate arrays. Investigations of such variations have not yet been made and would probably occur naturally in conjunction with studies concerning applications of the machine.

This machine has recently been simulated on an International Business Machine *704* computer at Bell Laboratories.

References

1. EXPERIMENTS IN CHESS, **J. Kister et al.** *Journal of the Association of Computing Machinery*, New York, N. Y., vol. 4, 1957, pp. 174-77.

2. PATTERN RECOGNITION AND LEARNING, **O. G. Selfridge.** *London Symposium on Information Theory*, 1955.

3. PATTERN RECOGNITION AND MODERN COMPUTERS, **O. G. Selfridge.** *Proceedings of the Western Joint Computer Conference*, Institute of Radio Engineers, New York, N. Y., Mar. 1955.

4. PROGRAMMING PATTERN RECOGNITION, **G. P. Dinneen.** *Ibid.*

The Magnetic Ledger-Card Computer

THOMAS P. HOLLORAN
NONMEMBER AIEE

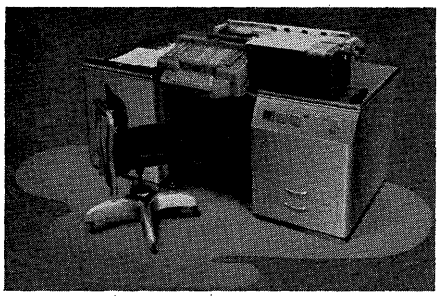


Fig. 1. The magnetic ledger-card computer

BUSINESS and government have a need for mechanization of record-keeping and accounting. Both big business organizations and government organizations seem to be approaching a solution to this problem by means of large-scale computing equipment and large-scale random-access electronic memory devices.

It is suggested here that much of this approach to the solution of the problem has been caused by the power and speed of the vacuum tube.

A computer is a device which requires many separate elements. These must perform functions of memory, switching, and amplification. The first digital computers utilized vacuum tubes to perform these functions. The vacuum tube is a device with a very-high power amplification, excellent frequency response, and, when all power supply and air conditioning requirements are considered, rather high cost. The result of building computers with such a device was that the machines tended to become costly and very fast. This led to a need for complete automation of the record-keeping facilities for the computer. Such fast and expensive machines could not tolerate human entry or semiautomatic entry of the data they require for record-keeping and accounting.

Complete automation of the record-keeping facilities of a computer has often meant magnetic tape files of various types. One problem encountered with this approach is that it makes difficult the necessary random access to such files of information.

It is possible to arbitrarily classify random-access problems into two categories. One type of such random access takes place when some record-keeping or accounting function is being performed by the computer which function necessitates access to various portions of a file on a random basis. That is, the data must not only be searched out and found, but operations must be performed upon the data by the computer at this time.

Another category of random access consists of those situations in which personnel in the organization desire to learn something from the master files but do not require any actual machine processing (alteration) of the master files.

Various completely automatic devices have been devised to make possible the first category of random access, but the second type of action requires programs or devices which permit an interruption of the basic computer operation or a means of temporarily detaching the file from the computer when random inquiries by personnel are to be made. Several simultaneous inquiries make the system even more complicated.

Since this random access to the file is a major problem no matter what the size of the business, a small business which would undertake a large-scale computer-type mechanization program would also be required to install whatever random-access equipment was necessary. In addition to being expensive, it could well be complex enough to be beyond the understanding of many of the concerned employees of such an organization.

The advent of semiconductors of little cost and low power supply or heat dissipation requirements made reasonable the visualization of machines complex enough to do the work of, and be classed as computers, but far lower in speed and cost than vacuum-tube machines.

Estimates showed that such a machine could be inexpensive enough to consider only a partial mechanization of the main file of the system and to further consider feasible some direct human input.

The present systems used in small business often involve the use of a ledger card as the unit of the master file of information for accounting and record-keeping purposes. The ledger card is a combination of historical data, current data, and relatively fixed data grouped by some common important characteristic; and it is in human-readable form. Ledger cards can be filed compactly and offer rapid access without machine aid or tie-up. The ledger card concept is a very powerful business tool which is widely used. The principal disadvantage of the ledger card is that it is not machine-readable.

This paper describes the system of making the ledger card readable by a small computer. This results in mechanization of an already powerful tool at a cost which

THOMAS P. HOLLORAN is with the National Cash Register Company, Dayton, Ohio

The author wishes to thank Patrick B. Close, James H. Randall, Robert F. Goldammer, Charles S. Jenkins, Lawrence D. Kilheffer, and many others who have contributed their talents to this development in the fields of logical design, circuit design, mechanical design, and magnetic design. The rapid successful development of this machine has been the product of their diligent and intelligent effort.

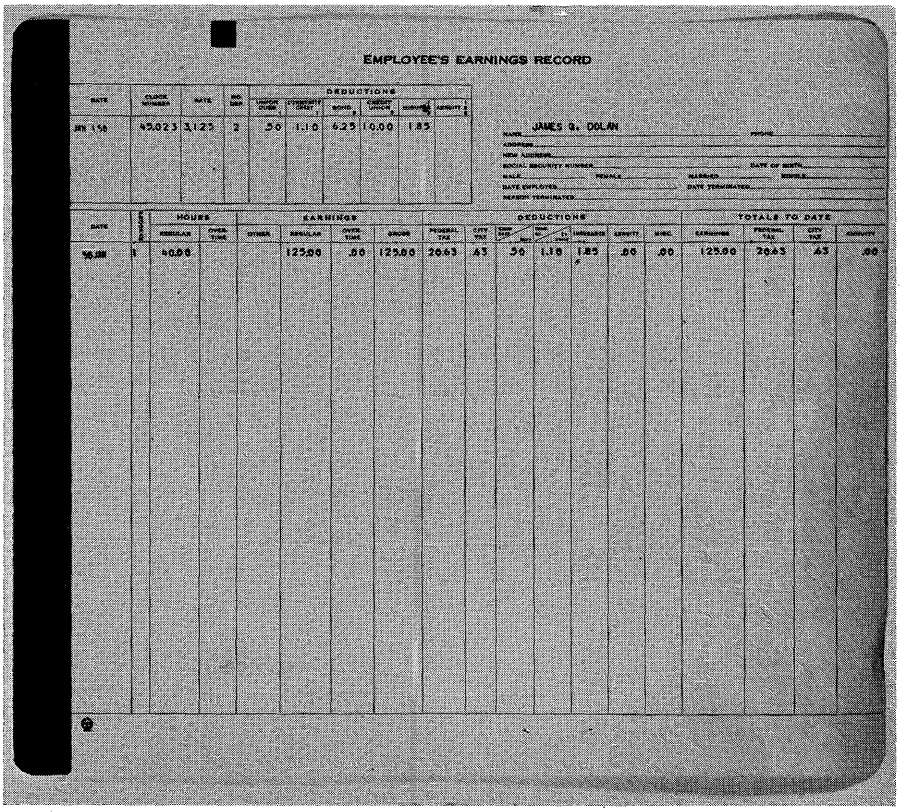


Fig. 2. A magnetic ledger card

can be borne by the small business organization. To avoid excessively detailed description of a particular accounting application, a general statement covering accounting work will be made.

In general, an accounting operation is made necessary by either the arrival of new data which must be processed in combination with the main file, to up-date the main file and produce auxiliary new documents, or by the elapsing of a specified period of time which makes necessary a processing of the main file documents.

The new information usually arrives in the form of some document. Input documents considered in this system are, for example, requisitions, time tickets, production reports, sales checks, etc.

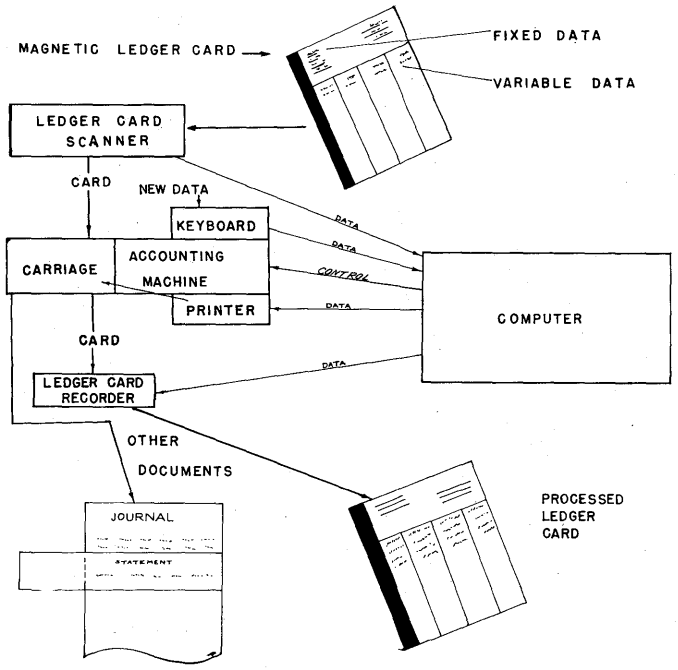


Fig. 3 (left). Processing a magnetic ledger card

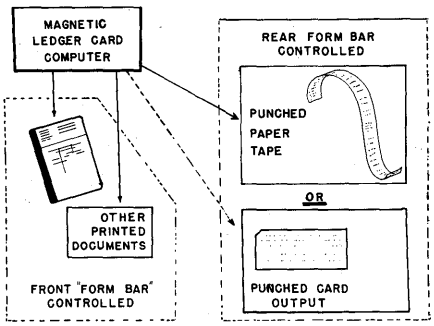


Fig. 4. A posting operation

It is assumed that the original input documents may continue to be manually prepared. This input media arrives in its original form at the small computer to be described. Either the media is hand-sorted to match the ledger file (a great amount of data naturally arrives in this sequence), or magnetic ledger cards are hand-selected to match them to the input media. This action amounts to an on-line random-access operation to the file for processing purposes. The machine-readable ledger card is inserted in the computer, followed by keyboard entry of the new data from the input document. The new data are utilized by the computer together with the data from the ledger card to perform all calculations involved in the accounting operation. The computing capabilities, the memory size, and the input-output capabilities of the magnetic ledger card computer are such that many operations are now grouped and speeded-up which were previously handled separately and sequentially.

The foregoing action results in production of several outputs simultaneously. The machine-readable ledger card is altered in both its human-readable and machine-readable sections. A separate journal sheet can be produced which provides a printed record of each operation.

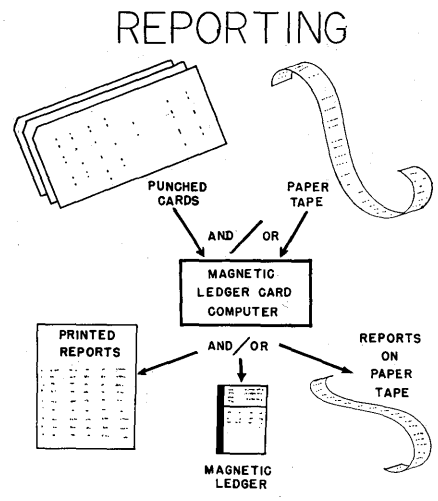
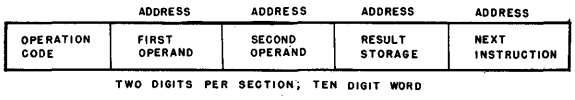


Fig. 5 (right). A reporting operation



ADD

SUBTRACT

MULTIPLY ("DOLLAR" DECIMAL)

DIVIDE

Fig. 6. Four-address instruction format

KEYBOARD ENTRY

PRINT

READ MAGNETIC LEDGER CARD

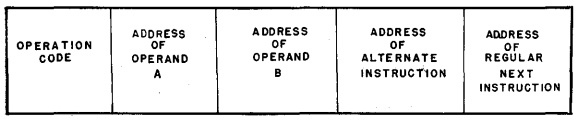
RECORD ON MAGNETIC LEDGER CARD

READ PUNCHED CARDS

CLEAR

SUM

Fig. 7. String-of-addresses instruction format



COMPARE FOR: A ≠ B

COMPARE FOR: A = B

Fig. 8. Branching instruction format

Printed auxiliary documents can be produced, such as, for example, a check and earnings statement, or, for another example, a sales audit report. Where necessary, an original document can be inserted into the carriage of the computer to verify the correct entry of the data into the accounting system by printing the keyboard-entered data upon the input document.

The prototype of the magnetic ledger card computer which produces all of the just-described output is a machine which can be described in the following paragraph.

It is an internally programmed 100-word computer, using all solid-state elements in its computing sections, and having as its principal input-output device an accounting machine. The computer is numeric-only and has a fixed word length of ten digits. It is all-decimal and has a core memory. It utilizes magnetic ledger cards as the main file or memory of the system, and the basic computer will be in the price range which has come to be known as that of the low-priced computer.

Fig. 1 shows one of the prototypes of the computer. The entire computer is contained in a very small desk, the tops of the side sections of the desk are only about 27 inches above the floor. A conventional accounting machine is seen to be the principal input-output device of the computer. The unit to the right of the accounting machine is the magnetic ledger-card reading and recording device. Below the ledger card reading station is the control panel of the computer, which is quite simple. The data indicated by the lighted numbers on the display section of the control panel include the operation code of the current instruction and the address of the next instruction. The very large carriage is seen to hold many documents which are processed at one posting operation of the machine. Across the front of the carriage is a mechanical programming device known as the form bar. A somewhat similar device is on the rear side of the carriage.

Fig. 2 shows a typical magnetic ledger card. The strip down the side of the ledger card consists of a magnetic oxide quite similar to that used for making magnetic tape. The printed portion of the particular sample ledger card is divided into two sections. The upper section contains semipermanent data which are seldom changed, the lower section contains variable data.

The capacity of the ledger card is ten decimal digits per usable inch of magnetic strip. A typical card such as the one shown would have a capacity of about 100 digits with variable word length and with the end-of-word marker requiring the equivalent of one digit space. Ledger cards having widths from approximately 5 to 16 inches can be utilized. The limit for length is set primarily by the depth of available file cabinets suited for storage. The aforementioned capacity is sufficient for the magnetic storage of the relatively fixed data found on such cards and for a portion of the last line of variable data. Thus, historical data are contained primarily in printed form, the assumption being that historical data will be referred to only by human beings rather than by the machines. This has been found to be an accurate assumption in practice.

The heart of this magnetic ledger-card computer system is seen to be groups of magnetic ledger cards which are serviced by one or several computers which can talk many machine languages and, in essence, can say a great deal without doing much thinking. The unusual aspect of this is that a relatively low priced and simple machine is so fluent.

Fig. 3 is a block description of a posting operation involving a magnetic ledger card, new data, and the production of other documents. The flow is from top to bottom in the diagram. The magnetic ledger card is inserted in the ledger-card scanner as indicated. The data from the magnetic strip of the card flow to the core memory in the computer while the card itself is transferred to the carriage of the accounting machine. The new data are entered into the keyboard of the accounting machine, from whence they go to the core memory of the computer.

The computer accomplishes processing on a time shared basis with the printing of the output, so that the entire operation is essentially proceeding at the printing speed of the accounting machine, computer processing being rapid enough that no appreciable delay occurs. Data produced in the computer flow to the printer as shown and to the ledger card recorder. The card, at the completion of the operation, moves from the carriage of the accounting machine out through the ledger card recorder which in fact is physically identical to the ledger-card scanner. It is shown separately here for clarity. The results are shown at the bottom: the processed ledger card and other documents produced at the same time.

The new data, whose arrival has brought about the necessity of the posting operation, together with some of the generated data, are often needed for further operations which are usually known as reporting.

Fig. 4 shows how this is accomplished. The magnetic ledger-card computer block in the upper left, essentially encompasses

all of Fig. 3. In the lower left are shown the outputs described previously, and on the right side are shown the other possible outputs, which can be obtained simultaneously. Punched paper tape or punched-card output can be obtained during the posting operation. The choice of which is to be obtained will depend upon the sorting and accumulating requirements for the report-making operation.

For those cases in which the report consists primarily of accumulations and does not include too many categories, paper tape can be utilized quite successfully. When the number of categories becomes quite large, the punched card can show advantages. The actual report-making operation is accomplished by reading the sorted punched cards or the original paper tape into the magnetic ledger-card computer and producing printed reports, magnetic ledger cards and/or reports on paper tape. When the number of categories on paper tape are more than the number which can be accumulated in a single pass of the tape, multiple passes of the tape are made through the tape reader of the magnetic ledger-card computer to accomplish production of the report. This operation is shown in Fig. 5.

During one program it is possible to read punched paper tape, punched cards, and magnetic ledger cards. Punched paper tape and punched cards are both machine fed, the magnetic ledger card is human-fed to the computer.

Some of the techniques which permit relatively inexpensive control of so many input-output devices can be outlined as follows:

1. Control of punched output, that is, paper tapes or punched cards, is by means of plugboards, stepping switches and a format-related device known as the rear-form bar of the accounting machine. So the control of these devices is not done directly by the computer. The computer need only control the accounting machine, which in turn controls these other functions.

2. Format control and mechanical storage control for the accounting machine is done by a mechanical-programming device on the accounting machine known as the form bar. This removes such problems as column spacing, nonsignificant-zero suppression, and storage of some data from the high-speed computer memory.

3. Another technique is that the core memory, rather than being run on a fixed timing basis, is, in effect, run in fits and starts as necessary to keep the core memory synchronized to the external device to which data are being fed or from which data are being received. This synchronizing technique avoids the need for extensive buffers between the external devices and the computer memory.

4. The computer itself is a strictly serial machine which keeps down its cost. The appearance of parallel operations is obtained, in many cases, by taking advantage of the self-completing characteristics of the actions of some of the electromechanical peripheral devices. As an example, the printing cycle of the accounting machine can be divided into two approximately equal portions. During the first of these portions the computer must control and feed data to the accounting machine. During the last portion of the cycle the accounting machine is essentially self-controlled and the computer is free at this time to perform mathematical operations. The approximately 250 milliseconds available at this time makes possible a time-sharing of computation and printing such that the machine appears to be printing at full speed in most posting operations.

In order to avoid the need for full time professional programmers for applying this machine, a number of features which are not common to many of the present-day large-scale computers were included in the programming capabilities. Some of the unusual features of the instruction of the computer are the following.

Fig. 6 shows one of the instruction formats utilized. This is sometimes known as the four-address format, sometimes as the three-address plus one. The basic arithmetic instructions utilize this format.

At first there seems to be a great deal of redundancy in the use of the address of the next instruction as a portion of each instruction. However, this has proved to be very useful for patching programs, and if a programmer is not too proud to leave his program in a patched form, a fairly large amount of programming time can be saved. In addition, a certain type of sorting by accumulation program benefits greatly from the ability to specify arbitrarily the address at which the next instruction is found. This particular example involves a program in which the instructions are scattered between certain desired accumulation cells.

Another instruction format used in the machine is shown in Fig. 7. For want of a better term it has been called the "string of addresses" format. Most of the input-output instructions utilize this format. Its principal feature is that many words of data can be inserted in or taken from a sequential series of cells merely by specifying the first and last addresses of this string. The operation code and modifier sections are needed to accomplish control of the various input-output devices. The instructions for which this format is used are listed in the figure.

Branching instructions have an easy-to-handle format in which both the regularly encountered next instruction and the alternate instruction addresses can be arbitrarily specified by the programmer. See Fig. 8.

An unusual feature of the core memory design utilized in this computer is the fact that it can operate over a fairly wide temperature range. Because of the temperature tolerance of the core memory and of all of the other computer circuits, no office modifications are necessary when placing this computer in service. The power requirements of the computer are less than 500 watts and a conventional 115-volt service is utilized.

The internal construction techniques for the computer involve the use of diode logic and resistor-logic techniques. The active elements in all cases are transistors. No vacuum tubes are used in the machine.

The philosophy of the control portion of the computer permits expansion to and alteration of the instruction list of the machine quite frequently.

Prototypes have been in operation for some time, and indications are strong that the basic system philosophy is workable and useful and has a very wide field of application.

Discussion

Chairman Levonian: The first question for Mr. Weiselman comes from Mr. G. H. Ottaway, IBM: "When will this device be commercially available?"

Irving L. Wieselman: I guess that I did not make it clear that this device is still a paper device. We have built media translators, media converters, and adapters for the device, and we have built memories, but we have not built this particular system, however. We would like to do so.

Chairman Levonian: From Mr. C. K. Budd, United States Army Signal Corps: "In converting codes or media, how are requirements for different machine programming handled?"

Irving L. Wieselman: If you know what the requirements of the machine are, you can write a program for them. This last device is a stored-program computer, and you merely have to write a program capable of handling many programs of the other computer.

Chairman Levonian: A question for Mr. Maxwell from Jerry Svigals of IBM: "Are you passing megacycle pulses through the microprogram control panel? If so, do you have a severe alteration problem?"

Marvin S. Maxwell: Yes, we will be operating on megacycle pulses. The pulse is impeded left. The cross problems will be more severe.

Chairman Levonian: I have a question for Mr. Bonney from Mr. H. G. Cragen, ARO, Inc.: "Can milliSADIC tape be used as input media?"

R. B. Bonney: Yes. We have tackled the design of input and output control units in the order of possible usage. As you can imagine, if you are considering all possible sources of data handling to feed a machine of this type, it would become a monumental list. Not having enough man power to handle it in the order of importance, the milliSADIC was considered, but the design is not yet set up to a point where we are certain of process, although it should be shortly.

Chairman Levonian: I have a question from J. I. Wieselman, Telemeter Magnetics, for Mr. Unger: "In what way do you get information in and out of the matrix if all operations affect all points of the matrix?"

S. H. Unger: Briefly, one way would be to have parallel input and have a photocell with each module. I have an instruction, which I mentioned in the paper, that is called a shift-around instruction which makes one big shift operation out of the whole matrix. It works like the shift instruction described here; i.e., when you get to the right-hand sign, the information shifts to the left-hand module, and in this manner it shifts information into the matrix in a serial manner.

Chairman Levonian: The next question is from Mr. J. Ameril, RCA Laboratories: "How could you use the matrix in a problem of patterns changing with time?"

S. H. Unger: One way is to think that the patterns are not changing too fast. For example, if the rate of the pattern changing were not greater than the operating time of the machine, we could simply consider the discrete array of the field in time just as we have considered the quad time. If the pattern were of a finite quantity, it changes the pattern into a series of sampled intervals, and operate our pattern then. Although we have not considered any application, I do not see any particular obstacle if the rate of change is not too great.

Chairman Levonian: Here is a question from J. M. McCarthy, MIT: "Have you considered whether your operations are universal with respect to local calculations?"

S. H. Unger: I am not sure what you mean by, "local calculations," so it is difficult to answer.

You could make the contents take a single point on the matrix, and convert that point to any desired function from the neighboring point extended out to any distance. If that is what you mean, any logical neighboring function in infinity could be transferred into that point.

Chairman Levonian: Mr. McCarthy asks, "Can you prove it?"

S. H. Unger: It is fairly obvious. You can shift since you have memory on the context of the neighboring module accumulator. I feel that you could do it within the limits of finiteness and that you could take care of any problem. I have never found a case where I have been stymied as yet.

Chairman Levonian: The next question is from Mr. A. A. Cohen, Remington Rand: "Can you achieve a carry equivalent, and thereby simulate various types of parallel arithmetic?"

S. H. Unger: I take it that you mean that you would represent the number by a row of ones and zeros on the matrix?

Chairman Levonian: Mr. Cohen adds: "Yes, as an experimental tool for playing with logic, that being on the parallel."

Mr. Unger: I think that it could be done, but I have not worked out the details.

I should mention that I left out one minor order which is the add reference. This means you can add a one into the lower left corner of the module.

Chairman Levonian: The next question is from Dr. H. K. Flesch, I.T.T. Laboratories: "Have you considered simulation of your machine on a conventional one and time sharing of one arithmetic unit for the whole array (simulation would do that)?"

S. H. Unger: Yes, I am considering simulating the machine on a conventional computer. We have just reached that stage, consideration. It seems to me that the other point, if you simulate the machine on a conventional computer, you are sharing one arithmetic unit among all the modules. This, in general, is a very good way to check out the programs, and I hope to do this on one machine or another.

Chairman Levonian: From Mr. F. Heart, Lincoln Laboratories: "What other pattern-type problems beside letter recognition have you considered?"

S. H. Unger: This is quite interesting. First of all we are considering a variety of miscellaneous-type problems that defy description. There is one problem perhaps as significant sounding as any, as it is the closest to the counting problem, where you have a wierd configuration of ones and zeros originating from implied figures, and you wish to count the number as isolated blocks, even blocks within blocks, multipulse, etc. The way we have adopted the problem to this machine is that we have attempted to find a program which will result in a single point being selected on every independent figure, if you can do this after doing the shift out of the machine, and count the points. I have been able to do this. In fact, as I mentioned before, there were two important steps in this particular program which had me stymied for a while, but it can be done, although it is a fairly complicated operation if the figure is to be considered in its most general sense without putting constriction on the number that is, the wheels within wheels, so to speak.

Chairman Levonian: Mr. R. Solomonoff, Zator Company asks, "Could you give any idea of how the machine recognizes rotational and dilational invariances?"

S. H. Unger: As far as rotational invariance is concerned, it is not too well adapted to this, and we can recognize a figure if it is rotated within a reasonable amount, 25 degrees once turned upside down. I feel that this is something new, recognizing a new figure, and the machine can be programmed to do just this, but it is hard for me to go into a description of just how it is done.

As for dilation invariances, I gather that you mean distortion in size and relative proportion. If there is no strain at all here, the machine does not count things, it works in a qualitative way.

Chairman Levonian: The first question for Mr. Holloran is from Mr. E. S. McCollister, ElectroData: "The machine described presumably would rent for about $1,000 per month or more. Why do you think that it would compete favorably with a punched-card installation of approximately the same cost and presumably of greater versatility? Are you assuming the need for a ledger card in business record keeping?"

Thomas P. Holloran: I am sure that the sales organization of our companies will be arguing this for some time. I am assuming that many people use ledger cards, and I would like to find some means of mechanizing them. This machine will appeal to some people, but as to whether it is economically preferable to punched-card operation in one particular operation or another, I think would require analysis. I am sure that our sales people will be doing this. I cannot give a more accurate answer to such a question.

Chairman Levonian: Here is a question

from Jerry Svigals, IBM, who asks, "How is the magnetic surface protected from potential damage resulting from manual handling? For example, creasing, dirt pickup, and scratching of the oxide coat, etc.?"

Thomas P. Holloran: The actual coating is a secret that is almost kept from me by our research department. We have been using these cards for quite some time, not only in the laboratory, but for commercial use, that is, actual commercial work cards. We have found that they do stand up well under the conditions encountered in offices.

Chairman Levonian: The next question is from J. Rogers, ElectroData: "What about speeds, arithmetic transfers, and actual cost?"

Thomas P. Holloran: As to the cost, I will have to give you the usual answer, see our sales force.

Concerning speed, during the last portion of the cycle the accounting machine is essentially self-controlled and the computer is free at this time to perform mathematical operations. The approximately 250 milliseconds available at this time makes possible a time-sharing of computation and printing such that the machine appears to be printing at full speed in most posting operations.

Chairman Levonian: The next question is from George Spangler, ElectroData: "How are the data represented on the magnetic strip?"

Thomas P. Holloran: The data on the magnetic strip are in serial form. Both the digits and the coding of the digits are serial, and variable word length is used, so a 2-digit word is recorded as a 2-digit plus word symbol. Therefore many data can be recorded on this strip.

Chairman Levonian: Mr. R. W. McClendon, Librascope Inc., asks, "How do alphabetic data get on to output if memory and ledger card handles only numeric data?"

Thomas P. Holloran: The alphabetic is only on output if in printed form. On the typewriter that you saw, the typewriter keyboard mechanically prints on the documents that are in the carriage at that time. The alphabet is not machine processed.

Chairman Levonian: The next question is from A. A. Cohen, Remington Rand Univac: "Did you state how programs are inserted into the machine? By what approximate factor does this machine multiply the output ability of one operator relative to the conventional accounting machine?"

Thomas P. Holloran: They can be by any means that we can use for getting data into a machine from the keyboard, magnetic ledger cards, paper tapes, or punch cards.

Concerning the second portion of the question about the approximate factor of multibility, one operator, and this does not imply that the machine is now no longer doing the work that the previous accounting machine could have done; for example, the accounting machine does not multiply or make decisions that are very complex and it does not divide. The answer appears to be that the one operator appears to be approximately the same, but doing so much more efficiently, thus the reduction in mistakes, as well as the simplicity of the operation is actually making money for the company.

Chairman Levonian: P. Smaller, Ampex Corporation asks, "Is the read head on the magnetic signal of standard design?"

Thomas P. Holloran: If you mean, is this a commercially available item, no. It is a head of our own design, using ferrites, as we did not need high density or wear resistance. It is quite conventional in structure.

Chairman Levonian: Mr. M. L. Lesser IBM asks, "Do the relatively permanent data on the magnetic strip stay throughout the operation, or are they all taken into the internal memory and rewritten on output?"

Thomas P. Holloran: Your second assumption was correct.

Chairman Levonian: Mr. S. H. Dinsmore, Bendix Computers asks, "What is the speed of reading the magnetic record?"

Thomas P. Holloran: This is approximately ten digits per second. On one of the prototypes we are considering changing this to approximately 100 digits per second.

Chairman Levonian: Mr. H. K. Flesch, I.T.T. Laboratories asks: "Did you mean 600-digit-per-second punched tape input? If so, why so fast, compared to 18-digit-per-second punched card input?"

Thomas P. Holloran: This is an interesting question. Yes, I did mean 600 per second. It does seem strange, but when using paper tape, a characteristic of the tape is that data on the tape remains in the same sequence as if it were punched. When we wish to make a record and form accumulative operations by certain categories, and in order to obtain a report, we scan data in successive passes, and it is necessary to read the tape extremely fast. In order to have useful output speed, we perform this type of work.

Chairman Levonian: Mr. McCollister, ElectroData asks, "Are multiple documents prepared in one operation, i.e., check payroll register and ledger?"

Thomas P. Holloran: Yes, they are. All of the documents in the carriage of the machine at one time by a carbonless reproducing process, developed by our company, causes the imprinting of several documents at once.