Abstracts of Current Computer Literature

These abstracts and the associated subject and author indexes were prepared on a commercial basis under the direction of Dr. Geoffrey Knight, Jr., Consultant to Cambridge Communications Corp., 1612 K Street, Washington, D. C., who publish Information Processing Journal, Solid State Abstracts Journal, Electronics Abstracts Journal, Cumulative Computer Abstracts, Cumulative Solid State Abstracts, and Cumulative Electronics Abstracts.

CONTENTS

ABSTRACTS................................................................. Pages 285–294

Abstract Numbers

0) GENERAL; PEOPLE AND SOURCES; EDUCATION.... 5531–5533
1) LOGIC AND SWITCHING THEORY; SEQUENTIAL MACHINES.................................................. 5534–5552
2) DIGITAL COMPUTERS AND SYSTEMS.................... 5553–5564
3) LOGIC DEVICES AND CIRCUITS (HARDWARE)...... 5565–5567
4) DIGITAL STORAGE AND INPUT-OUTPUT EQUIPMENT 5568–5570
5) PROGRAMMING AND CODING OF DIGITAL MACHINES.................................................. 5571–5582
6) HUMAN COMMUNICATION, DOCUMENTATION, AND HUMANITIES........................................ 5583–5584
7) BEHAVIORAL SCIENCE, PATTERN RECOGNITION, AND ARTIFICIAL INTELLIGENCE............... 5585–5586
8) MATHEMATICS........................................................... 5587–5594
9) PROBABILITY, MATHEMATICAL PROGRAMMING, DIGITAL SIMULATION, INFORMATION THEORY, AND COMMUNICATION SYSTEMS.................................................. 5595–5597
10) SCIENCE, ENGINEERING, AND MEDICINE......... 5598–5601
11) ANALOG AND HYBRID COMPUTERS..................... 5602
12) REAL-TIME SYSTEMS AND AUTOMATIC CONTROL; INDUSTRIAL APPLICATIONS......................... 5603
13) GOVERNMENT, MILITARY, AND, TRANSPORTATION APPLICATIONS..................................... —
14) BUSINESS APPLICATIONS OF INFORMATION PROCESSING............................................... 5604

SUBJECT INDEX............................................................ Pages 294–297

AUTHOR INDEX.......................................................... Page 297
0) GENERAL; PEOPLE AND SOURCES; EDUCATION


Contents: Interactive information processing; Optimization, adaptation, and learning in automatic systems; Recent work on theorectical models of biological memory; Some approaches to optimum feature extraction; Evaluation and selection of variables in pattern recognition; Some topics on non-supervised adaptive detection for multivariate normal distributions; Nonlinear environments permitting efficient adaptation; Recognition of order and evolutionary systems; Stochastic automata as models of learning systems; Adaptive systems with a variable structure; Fundamentals of principal and blanket controls; Preliminary design of an intelligent robot; Iterative storage of multidimensional functions in discrete distributed memories; A command language for visualization of articulated movements; Some theoretical aspects of automatic indexing.

The nature of syntactic redundancy; On communicating with machines in natural language; A generalization of the linear threshold decision algorithm to multiple classes.


The purpose of the Defense Intelligence Agency Manual of Data Processing Standards is to provide guidance, to establish standard methods and procedures for languages, design, programming, and operation of the Automatic Data Processing Systems (ADPS) Center, and to define personnel and equipment performance standards and methods of control and evaluation of performance standards to insure maximum utilization of the ADPS Center's resources.


Digital computers have had a significant effect on engineering practice, but they have not yet had a comparable effect on educational methods, primarily because of a lack of convenient man-machine communication capabilities. However, during recent years substantial strides have been taken in making the digital computer more responsive and easier to use with natural "problem-oriented" languages and almost instantaneous response. This emerging generation of rapid-access computer systems promises to have a profound impact on both engineering practice and education. The purpose of this paper is to describe an educational experiment that was undertaken to explore the use of "computer laboratory" projects as part of a graduate course on "Electronic Circuit Theory: Nonlinear and Time-Varying."

1) LOGIC AND SWITCHING THEORY; SEQUENTIAL MACHINES


The theory of J. A. Robinson's resolution principle, an inference rule for first-order predicate calculus, is unified and extended. A theorem-proving computer program based on the new theory is proposed and the proposed semantic resolution program is compared with hyperresolution and set-of-support resolution programs. Renamable and semantic resolution are presented as a method and shown to be identical. Given a model M, semantic resolution is the resolution of a latent clash in which each "electron" is at least sometimes false under M; the nucleus is at least sometimes true under M. The completeness theorem for semantic resolution and all previous completeness theorems for resolution (including ordinary, hyper-, and set-of-support resolution) can be derived from a slightly more general form of the following theorem. If U is a finite, truth-functionally unsatisfiable set of nonempty clauses and if M is a ground model, then there exists an unresolved maximal semantic clash {E1, E2, ..., En, C} with nucleus C such that any set containing C and one or more of the electrons E1, E2, ..., En is an unresolved semantic clash in U.


In many fields of mathematics the richness of the underlying axiom set leads to the establishment of a number of very general equalities. For example, it is easy to prove that in groups (x^2)^n = x and that in rings -x -y = x -y. In the presence of such an equality, each new inference made during a proof search by a theorem-proving program may immediately yield a set of closely related inferences. If, for example, b -a = c is inferred in the presence of (x^2)^n = x, substitution immediately yields obviously related inferences such as (b^2 -a) = c. Retention of many members of each such set of inferences has seriously impeded the effectiveness of automatic theorem proving. Similar to the gain made by discarding inferences of inferences already present is that made by discarding instances of repeated application of a given equality. The latter is achieved by use of demodulation. Its definition, evidence of its value, and a related rule of inference are given. In addition a number of concepts are defined the implementation of which reduces both the number and sensitivity to choice of parameters governing the theorem-proving procedures.


The structure of the algorithms implementing Quine's method for prime implicant determination is analyzed, and a new class of algorithms—partitioned list algorithms—for Quine's method is derived. Such algorithms are optimal in the sense that, in actual computation because they permit (1) avoiding repetitions while generating clauses, (2) representing each clause by only one binary configuration, (3) reducing memory capacity requirements, and (4) applying the basic operations by means of nonexhaustive techniques.


Design of flip-flop networks, realization of incompletely specified state tables, design of asynchronous sequential networks, state assignment, and other logic design problems can lead to Boolean functions which contain arbitrary parameters. These parameters are a generalization of don't care conditions and may be assigned arbitrary values so as to minimize the cost of realizing the functions. A modification of the Quine-McCluskey procedure permits minimization of arbitrary parameter functions. A prime implicant list is developed in terms of the parameters and is used to derive a conditional prime implicant chart. Minimum solutions are obtained from this chart by a modified Petrick method or by branching. A second method for minimizing arbitrary-parameter functions treats a function of m parameters and n variables as an (m+n)-variable function. The prime implicants of this function are derived by iterated consensus and then modified to obtain the conditional prime implicant chart. Both methods have been generalized to the multiple-output case.


The minimal number of terms required for representing any switching function as a modulo 2 sum of products is investigated, and an algorithm for obtaining economical realization is described. The main result is the following: every symmetric function of 2m+1 variables has a modulo 2 sum of products realization with at most 3^{m+1} terms; but there are functions of n variables which require at least 2^n/log_2 n terms for sufficiently large n.


The correspondence defined in this paper was used to convert Slepian's tabulation of the number of equivalence classes of (m, r) group codes into a hitherto unpublished data of relevance to switching theory. Specifically,
Table II lists the number of equivalence classes of switching functions of weight $m < 20$ or $m > 2^{2n} - 20$ in nine or fewer arguments under the group of linear transformations on its argument variables. The correspondence is established by means of symmetric functions. The rows of this matrix define $m$ points at which a switching function of $n$ arguments takes on unit value. If the rank of this matrix (over the 2-element field) is $r$, then there is an $r$-dimensional subspace of binary $m$-tuples which, by definition, is the message set of an $(m, r)$ group code.

### 5540

**Symmetric Ternary Switching Functions:**


A simple, systematic procedure for detecting symmetric ternary switching functions is developed. The detection procedure is essentially based on a derived counting theorem. Other important properties and close approximations of symmetric functions are developed. A design method is given whereby symmetric functions can be readily synthesized with switching networks which are economical, and, in certain instances, minimal in the number of ternary threshold devices required. The greatest lower bound on the number of devices required for a given symmetric function is developed.

### 5541


Signal switching networks with transient blocking are defined. The number of $2 \times 2$ crossbars necessary to synthesize a signal switching network with transient blocking capable of performing all one-to-one connections of $N$ inputs to $N$ outputs is shown to be at least $N \log N - N \log \log N + (1 + (1/2) \log \log N + 2 + o(1))$ as $N \to \infty$. It is shown that this lower bound can never be attained for $N > 2$. An algorithm for building a network using at most $2 \log N N 2 \times 2$ crossbars is described. If $N$ is a power of 2, $N = 2^m$, then the algorithm described requires $N \log N = N N + 122 \times 2$ crossbars, which is close to the theoretical minimum. Generalizations of this work to networks performing an arbitrary permutation group of connections of inputs to outputs are indicated. Explicit results are obtained in the case of Abelian groups.

### 5542


An algorithm is described for constructing a set of input patterns able to detect every given failure of a combinational network. The result may not be a minimal set of tests, but it is surely a complete test. The procedure is designed for large networks with high number of independent inputs. It can deal with networks including any elementary logical gates such as AND, OR, NAND, NOR, and the function of fault out are considered. The method can treat any class of single or multiple logical failures which can be described by means of a transformation of the logic equations of the network. The conventional description of the failures on one wire stuck at one or stuck at zero is not required. This can be useful in detection of wiring errors in equipment testing. The basis of the procedure is a "choice technique" which avoids, in most cases, the two-level expansion of the mapping realized by the network. The procedure is illustrated in detail and two application examples to the detection of a single wire stuck at zero failure and of a wiring error are given.

### Cellular Threshold Arrays Realizing Combinational or Sequential Logic—see 5567.

### 5543

**Quasilinear Automata (in German),** G. Hotz (Univ. des Saarlandes, Saarbrucken); Computing, vol. 2, no. 2, pp. 139–152, 1967.

The results of the theory of linear automata which concern the reduction, the cycle lengths of the autonomous factor, and the isomorphism between the autonomous and other factors of the automata are generalized to quasilinear automata. As the linear automata gives rise to the cyclic codes, so the quasilinear automaton defines a corresponding class of codes.

### 5544


The techniques of automatic programming are used for constructive proofs in automata theory. A formal definition of an elementary programming language for a stack automation is given, and it is shown how this may be readily adapted to other classes of automata. In the second part of this paper shows how this programming language can be applied to automata theory, as the authors prove their non-context-sensitive languages accepted by a stack automation.

### 5545


The closure properties of the class of languages defined by real-time, on-line, multi-tape Turing machines are proved. The results obtained are, for the most part, negative and, as one would expect, asymmetric. It is shown that the results remain valid for a broad class of real-time devices. Finally, the position of the class of real-time definable languages in the "classical" linguistic hierarchy is established.

### 5546


The purpose of this note is to show that it is recursively undecidable how much tape is required by a Turing machine to recognize nonregular context-free languages.

### 5547


A Post machine is a Turing machine which cannot both write and move on the same machine step. It is shown that the halting problem for the class of 2-state Post machines is solvable. Thus, there can be no universal 2-state Post machine. This is in contrast with the results of Shannon, that there exist universal 2-state Turing machines when the machines are capable of both writing and moving on the same step.

### 5548


The analysis of the Krohn-Rhodes theory of finite state machines is developed for linear positive machines with inputs from a commutative Banach algebra with involution. The notion of prime machine in this context is shown to correspond to maximal ideal in the input algebra. Further, every positive machine is shown to lie in the closed convex hull of its primes.

### 5549


In previous papers, Krohn and Rhodes exploited the natural connection between finite semigroups and finite-state sequential machines to prove a prime decomposition theorem for finite semigroups and machines. The present paper uses the methods of this algebraic approach to obtain a prime decomposition theorem for generalized machines together with the determination of properties preserved under series and parallel (or cascade) composition of sequential machines.

### 5550


If, in a sequential machine, the output upon application of an input character depends only upon the current state of the machine and upon the input character, the machine is called input-independent. Two states, $p$ and $q$, of an input-independent machine are compatible if and only if the output strings from the machine starting in initial state, $p$, are the same as the output strings from the machine with initial state, $q$, for all input strings to the machine in both states. The minimal length of tape which tests compatibility of states in an input-in-
dependent sequential machine with \( n \) states is \((n^2 - 2n)/4 + 1\) if \( n \) is even, and \((n^2 - 2n + 1)/4\) if \( n \) is odd. There are machines with incompatible states such that no tapes of length less than the given bounds will detect the incompatibility.


Given a minimal sequential machine \( M \) and a positive integer \( T \), it is desired to partition the state set of \( M \) into \( T \) classes, say \( S_1, S_2, \ldots, S_r \), such that all states in \( S_i \) under all possible inputs, pass into states in \( S_{i\pm T} \). If such a \( T \)-partition exists, \( M \) can be realized by means of periodically varying logic, which often results in the saving of memory elements. The period of \( M \) is defined as the greatest common divisor of all cycle lengths of \( M \)—a quantity which can be readily evaluated since it depends only on a finite set of independent loops exhibited by the state graph. The main result is that a \( T \)-partition exists for \( M \) if and only if \( T \) is a divisor of the period of \( M \). For every such \( T \), an algorithm is given for constructing the corresponding partition. If \( M \) is not required to be minimal, it is shown (constructively) that a \( T \)-partition exists for every \( T \).


In most studies of the structure of sequential machines there has been a tacit assumption that the machine was to be realized with unit delay memory elements. This report considers sequential machines that are realized with either trigger or set-reset flip-flop memory elements. It is shown that the relation called a partition pair which predicts the dependence of the input functions to unit delay memory elements does not predict the dependence of the input functions to trigger or set-reset flip-flop memory elements. Relations called input-pairs and output-pairs are defined which characterize the dependence of the input functions to trigger and set-reset flip-flop memory elements, respectively. It is found that these relations do not have all the algebraic properties that partition pairs possess. Feedback in sequential machines that are realized with trigger or set-reset flip-flop memory elements is also studied. A method is given for determining when a machine can be realized with either trigger or set-reset flip-flop memory elements using function \( f \) for feedback. It is shown that if a sequential machine can be realized with unit delay memory elements using a function \( f \) for feedback then it can be realized with set-reset flip-flops using \( f \) for feedback. It is also shown that for completely specified machines that if a machine can be realized without feedback using unit delay memory elements then it cannot be realized without feedback using trigger flip-flop memory elements. The converse statement is also true.

2) DIGITAL COMPUTERS AND SYSTEMS


Real numbers can be represented in a binary computer by the form \( i.B \) where \( i \) is the integer part, \( B \) the base, and \( e \) the exponent. The accuracy of the representation depends upon the number of bits allocated to the integer part and exponent part as well as what base is chosen. If \( L(i) \) and \( L(e) \) are the number of bits allocated to the magnitudes of the integer and exponent parts and \( I = 2^{2L(i)} \) and \( E = 2^{2L(e)} \), the exponent range is given by \( B^{2E} \), the maximum relative representation error is given by \( B/21 \), and the average relative representation error is given by \( (B-1)/(4I \in B) \). The formulas provide quantitative comparison for the effectiveness of alternative formats for real number representations.


An extension of a previously reported interval arithmetic is described. It is one of the essential features of this new interval arithmetic that in the case of interval functions the exact range of interval values is obtained. A machine interval arithmetic corresponding to the new interval arithmetic is also given and the properties of these arithmetic are studied.


The time required to perform multiplication is investigated. A lower bound on the time required to perform multiplication, as well as an algorithm modulo \( N \), is derived, and it is shown that these lower bounds can be approached. Then a lower bound on the amount of time required to perform the most significant part of multiplication \( (x/y) \mod N \) is derived.


A functional device with the logic operation of a half adder using optical coupling in a single unit of the injection laser was fabricated. The mechanism of the operation is described and the results at 77 K are shown.


Two algorithms are presented: one, DALG II, computes a test to detect a failure in acyclic logic circuits; the other, TEST-DETECT, ascertains all failures detected by a test. Both are based on the utilization of a “calculus of D-cubes” that provides the means for effectively performing the necessary computations for very large logic circuits. Strategies for combining the two algorithms into an efficient diagnostic test generation procedure are given. APL specifications of the algorithms are given in an appendix.

Diagnosis of Failures in Logic Networks—see 5542.


This paper is intended to show how the unique features of the Variable Instruction Computer (VIC) make it valuable for applications requiring high reliability. By careful choice of components, use of error-checking circuits, and selected application of redundant hardware, the basic unextended reliability of the VIC is comparable with the state-of-the-art. This is verified by a standard MIL-HDBK-217 type of analysis. The variable instruction technique is briefly described by use of a block diagram. The method for extending reliability by use of variable instructions is explained and an example is given. The concept of algorithm change to achieve controlled graceful degradation is discussed. An analysis of a typical application of this technique is given and extensions of the variable instruction concept to more advanced reliability requirements are discussed.


This report describes a set of techniques which can be used to design and construct self-repairs digital systems with spare parts switching. Both the problem of fault detection and location and the problem of spare switching are discussed. Residues coding for arithmetic unit fault detection and location is discussed and rejected. Then, duplication techniques for both fault location and repair are discussed. A design plan is given for a demonstration model to demonstrate self-repair techniques. The design plan includes logic equations, timing diagrams, and a detailed operational description.

The steadily increasing complexity of spaceborne digital systems tends to lower the reliability of those systems that operate in an environment where the cost of failure is extremely high. The reliability of digital systems can be increased by redundancy techniques. The majority of work in redundancy has concentrated on the development of synthesis techniques and initial reliability estimation procedures. Relatively little effort has been spent on the development of procedures for testing redundant systems and estimating their reliability when some components may be failed. This paper describes 1) a procedure for allocating a limited number of test points within a redundant digital system, and 2) a compatible procedure for estimating the probability of successfully completing a mission, using information obtained from the allotted test points.


A new approach to a computer organization promises very effective utilization of the LSI technology. Functional partitioning of both the data path and control is employed. A dramatic reduction in array pin requirements by a factor of two or more is achieved. Arrays as small as a few dozen gates can be effectively utilized. The total system is exceedingly flexible in both performance and instruction set.


The continued development of Large Scale Integration (LSI) presages the advent of a fourth generation of computers, and is causing all levels of computer technology—both technical and managerial. Today’s computers use integrated circuit components containing at most ten gates per component; however, LSI is introducing hundreds of gates per component and will eventually evolve into thousands of gates per component. To the aerospace planner, this technological breakthrough of LSI means tremendous reductions in cost, size, weight, and power consumption of logic components, together with increased speed and reliability. However, for the aerospace planner to successfully implement aerospace computers with LSI, the computer designers and managers must reorient their methodological and goals. A multitude of new design and cost considerations must be carefully scrutinized, and out of this must come the new techniques that will permit effective incorporation of LSI in aerospace computers.


The problem of predicting the performance of modern computer systems is formidable. One general technique which can ease this problem is macroscopic simulation. This paper reports on the applicability of that technique to System/360. The paper describes an experimental model of System/360—its hardware, software, and its environment. The measures of system performance produced by the model consist of statistics relating to turn-around time, throughput, hardware utilization, software utilization, and queuing processes. The model is mechanized in SWSCRIPT and consists of some 1750 statements. An auxiliary program, the Job Generator, creates automatically the properties of System/360 jobs that get simulated.

Bulk Core Storage in a 360/67 Time-Sharing System—see 5568.

GEORGE 3 General Purpose Time-Sharing and Operating System—see 5582.


FAMOUS is an on-line system for the manipulation of linguistic forms. Although these forms can have quite arbitrary interpretations, the standard interpretation is that they are algebraic expressions. FAMOUS allows its "algebraic expressions" to include arbitrary functions which may or may not be defined. In this way, regular non-algebraic constructions may be concealed as arguments of ad hoc functions. Rules of local change are the heart of FAMOUS, and supplied by the user. Using these rules, FAMOUS looks at an algebraic manipulation as a series of local changes. The centrality of proximity in FAMOUS was originally prompted by G-theory, which might be called the study of proximity.

3) LOGIC DEVICES AND CIRCUITS (HARDWARE)

Diagnosis of Failures in Logic Networks—see 5543.

Memory Array Permitting Logic Manipulations—see 5569.


A systematic and general method of computing ac and dc characteristics of double-diffused junction transistors using major process parameters such as dimensions of the device, surface concentration, junction depth, diffusion time, temperature, and diffusion coefficient as a function of temperature and impurity concentration is described. These parameters can be checked during the process and, therefore, can aid the process control problem by predicting the expected values of junction depth and sheet resistivity. If the specified control parameters are met during the fabrication, ac and dc characteristics of the device will be realized. The time and/or frequency response of a circuit can be computed using the ac and dc characteristics of the diffused devices of the circuit on the basis of a distributed or an equivalent lumped model. The measurement of important ac parameters on the basis of these models has also been simulated on the computer, thus aiding the characterization problem of the device in the integrated circuit environment. Also, the switching speed of a loaded logic net can be computed and optimized by trading off interacting parameters and relating them back to the original diffusion process parameters and dimensions of the components. Finally, experimental verification of the computed results has been accomplished and found to be satisfactory.


The variational approach to the optimal design of high-speed switching circuits is explored. The approach implements the variational calculus to obtain an expression for the vector sensitivity of a scalar performance function (e.g., delay or switching time) to changes in the vector of design parameters. Gradient methods are established for using the vector sensitivity to iteratively update the parameter vector and obtain an optimal design. It is shown that the variational approach retains, typically, an M-fold computational advantage over conventional step-and-repeat methods in determining the sensitivity of a scalar performance function to M design parameters. The approach is shown to be well adapted for incorporation into package analysis programs with matrix formulations, and vested with sufficient generality to be applicable to a wide range of switching circuit problems (e.g., low-power or large-scale integrated circuits). It is further shown that subsumed in the general class of nonlinear parameter-value synthesis problems is the class of delay-minimization problems, and that the switching time minimization problem is a special case of the classical "time-optimal" problem.


The purpose of this paper is to describe the design of an all-digital cellular threshold array that is well adapted to realization by large-scale integrated semiconductor technology. A set of these arrays may be inter-
connected to realize arbitrary combinational or sequential logic, or may be stacked to
form a multilevel adaptive pattern classification machine.

Injection Laser Half Adder—see 5556.

4) DIGITAL STORAGE AND
INPUT-OUTPUT EQUIPMENT

Memory Allocation for Multiprocessors—
see 5572.

Buffer Storage Required for Random Input
Word Rate—see 5573.

5560 Bulk Core in a 360/67 Time-Sharing Sys-
tem, H. C. Lauer (Carnegie Inst. Tech.,
Pittsburgh); 1967, 34 pp., AFOSR-67-1968;
U. S. Gov't R & D Rept., vol. 67, p. 88(A)
November 10, 1967. AD 657 782 CFSTI
$3.00.

In time-sharing systems where programs
and data move frequently between storage
media, performance measured in terms of
response time, availability, capacity, and
generality depends on the ability of the sys-
tem to move information quickly and,
promptly upon demand. Analysis of and
early experiences with TSS/360 reveal that
a drum-oriented system cannot meet the
demands imposed by user tasks. Conse-
quenty, Carnegie Institute of Technology
has replaced the drum on its 360/67 with
Large Capacity Core Storage. A model of the
drum system was constructed, and it
was discovered that it could not support its
maximum paging rate except under condi-
tions which impose high system costs. It was
also found that because of its rotating na-
ture, it actually withdraws significant por-
tions of memory from the usable main mem-
ory of the system. Bulk core, when operated
with a single core-to-core channel, has
neither of these faults. It provides the added
advantage that not all pages need be swaped—those which are not heavily used
may be referenced directly by the CP. By
operating selectively in both modes nearly
an order of magnitude better performance
than is possible with a drum is expected.

5569 Plated Wire Content-Addressable Mem-
ories with Bit-Steering Technique, W. F.
Chow (Bell Tel. Labs, Murray Hill); IEEE

This report describes a new concept of
content-addressable memory (CAM) im-
plemented with plated wires and the bit-
steering technique. The 5-mil plated wires
are insulated with a thin (0.2 to 0.3 mil) coat
of polyurethane, and imbedded in a copper
plane. A memory array is formed by means
of a simple orthogonal arrangement of plated
wires and an overlying two layers of copper
stripes. Because of the low loop impedance (2 to 5 ohms
depending on the design but uniform to
within ±5 percent for a given design)
formed between an insulated plated wire
and the copper plane, a loop current of the
current of 30 mA can be generated by pulsing
a pair of straps. This current is used to trans-
fer information from one bit position to an-
other bit along the same plated wire. Conse-
quently, logic manipulations can be per-
formed within the memory array. The res-
olved operation is executed in the array
where a binary address tree is stored. The
addressed words are obtained via the sieving action of the tree. With a com-
plementary address tree added to this res-
olved area, the addressed READ/WRITE
operations are executed by first searching for the
given address and then read/write on the
matched word. The search cycle time of a
CAM of 4096 words, 40 bits per word, is
estimated to be about 1 to 2 μs, based upon
experimental results. The resolve time is
about 1.5μs for any one of the 4096
addresses. Because the 8-bit words, the
resolve logic are implemented with a portion
of the memory array, this bit-steering
plated-wire content addressable memory
has a minimum of semiconductor electronics
which is proportional to the number of bits
per word but, to first order, independent of
the number of words of the memory.

5570 Design Considerations for a Parallel Bit-
Organized MOS Memory, L. Pasqualini
(Phillips Field Corp, Clark); IEEE Trans.
557, October 1967.

This paper discusses the design trade-offs
for a parallel bit-organized MOS memory.
A memory capacity of 40K bits can be
achieved using LSI techniques. Memory
storage capability is expandable in both word
length and number of words stored. The
physical dimensions of the memory should
be considerably smaller than those of a com-
parable core design. Power consumption per
bit should likewise be less than that achiev-
able with cores. A full cycle time of 1μs or
less can be achieved. Cost per bit should
compare very favorably with that of a core
design.

5) PROGRAMMING AND CODING
OF DIGITAL MACHINES

5571 Conversion of Limited-Entry Decision Tables to Optimal Machine Programs. II;
Minimum Storage Requirement, L. T.
Reinwald and R. M. Soland (Research
Analysis, McLean); J. ACM, vol. 14, pp.
742–756, October 1967.

Given the number of words of computer
storage required by the individual tests in a
limited-entry decision table, it is sometimes
desirable to find an equivalent computer pro-
gram with minimum total storage require-
ment. In this paper an algorithm is de-
developed to do this. The rules in the decision
table are grouped into action sets, so that
several rules with the same actions need not
be distinguished. Moreover, if certain com-
binations of conditions can be excluded from
consideration, the algorithm will take ad-
vantage of this extra information. The algo-
rithm is initially developed for computer
programs possessing a treelike form and then
extended to a wider class of programs. The
algorithm can be combined with one which
finds an equivalent computer program with
minimum average processing time, and thus
used to find an equivalent computer pro-
gram which minimizes a cost function which
is nondecreasing in both average processing
time and total storage requirement.

5572 Memory Allocation for Multiprocessors,
A. F. Rosene (Sylvania, Needham); IEEE

In multiprocessor systems it is desirable
to look for and allocate storage without ex-
tensive data moving. Two techniques for
accomplishing this, associative memory
structures and indirect addressing tech-
niques, are described. It is concluded that
the two methods are similar in performance,
but that indirect addressing is more eco-
nomical. An indirect addressing method is
described in detail and various methods of
implementation compared. The memory
overhead and the time penalty (in memory
accesses) are given as a function of memory
size and method of implementation. It is
concluded that a computed address table
look-up technique should be used and that
an block size should be either 356 or 512
words.

5573 Guide to the Length of Buffer Storage Re-
quired for Random (Poison) Input and
Output Operations. N. M. Dor (U. of
California, L. A.); IEEE Trans. Electronic
Computers (Short Notes), vol. EC-16, pp.

The required buffer size for a random
(Poisson) input word rate with constant rate
removal in the same order as arrival is con-
sidered. The method of computation rests
on analytical study. Results are tabulated for
a range of values and may be used as a design
guide. Applications may be found in both
the partitioning of computer stores and in
the communications field of data compression.

5574 Microprogrammed Control in Problem-
Oriented Languages, I. T. Hawryszykiewicz
(Postmaster General's Dept. Res. Labs.,
Australia); IEEE Trans. Electronic Comput-

The application of microprogramming to
problem-oriented languages is described in
terms of a simulated analog system on a
digital computer. Microprogramming facili-
tates the development of a problem-oriented
machine code, which can then be genera-
ted from the problem-oriented source lan-
guage by simple translation. In the system
the problem defined by differential
expressions is expressed as a set of
analog diagrams. The diagram is coded into
an analog-oriented source language which
is converted by translation into the analog
machine code. The system also allows the
machine assembly code to be freely used
with the analog input in coding the super-
visory and interrupt facilities that are in-
corporated in the overall system design.

5575 A Computer System for Inference Execution
and Data Retrieval, R. E. Levien (RAND,
Santa Monica) and M. E. Maron (U. of
California, Berkeley); Commun. ACM, vol.

This paper presents a RAND project con-
cerned with the use of computers as assis-
tants in the logical analysis of large collec-
tions of factual data. A system called the
Relational Data File was developed for this purpose. The Relational Data File is briefly detailed and problems arising from its implementation are discussed.

5576

A file adjustment procedure based on maximizing the Bayes expected gain is proposed for matched term retrieval systems. The expected gain and its probability distribution are derived as a function of 1) the prior proportion of omitted terms, and 2) the coefficient of separation between two distributions corresponding to values of an adjustment statistic. An example evaluates the gain parameters for a typical information retrieval system.

Programs for Solving Combinatorial Search Problems—see 5585.

5577

The many faces of programming and systems development demand an immense amount of mechanical routine work. This paper tries to explain some areas where automation of many tasks may be of great help. One special area where progress seems to lag behind unduly can be found in debugging, testing, and diagnosing systems. The generation of programs automatically from a definition of a problem and the characteristics of programs for its solution by a software system, specially designed for this purpose, has been attempted. It is indicated how the ideas underlying this project may be applied successfully to other areas.

5578

The Automatic Checkout Systems (ACS) were designed and used for the Titan III and Apollo Projects. This article describes the logical design of new approaches to program checkout and the results achieved. These checkout systems are a series of programs written for a large airborne commercial computer. When a program for the airborne computer of the inertial guidance system is operated on by the ACS, it performs a symbolic simulation and generates the symbolic equations \( A = B + C \times D \) that are performed by the airborne computer program. The ACS performs many checks for program errors in this process, indicates all areas where program analysis is required, and furnishes the program information required for analysis in optimum form. The equations generated by the ACS are manually checked against the input specification. This check is absolute in nature compared to the conventional method of analyzing the results of numerical simulations. Time consumed on the large scale commercial computer is much less than for numerical simulation.

Programming Languages for Automation—see 5544.

5579

LINGO is a special-purpose programming language developed for computer-aided analysis of linear networks at a remote terminal of a time-sharing computer system. A special compiler (written in FORTRAN) enables the computer to read and execute LINGO programs, and the complete system can be implemented on very small computers. (The General Electric Time-Sharing System, on which LINGO was developed, allows the user only 5300 locations in core memory.) Taking full advantage of the conversational interaction with the computer, the engineer can proceed from the rough idea of the circuit diagram, through writing the LINGO program, to execution and tabulated results within 20 minutes or less. Having access to such a tool encourages the use of the computer as a real desk-side aid to linear network analysis.

5580

A compiler generation system is described which is rigorously based and which allows formal specification both of the source (procedure oriented) languages and of the object (machine oriented) languages. An intermediate or "buffer" language, BASE, is interspersed, reducing the required transformation techniques described. The system, so far, includes those elements in BASE necessary to produce ALOG, FORTRAN, and JOVIAL compilers.

5581

This report describes the organization of compilers to realize increased system efficiency in the areas of better hardware utilization, reduced compilation overhead, and improved man-machine communication. The principal emphasis is directed to the following topics: incremental translation, reentrant and re usable compilers, execution using interpreters and emulation techniques as well as conventional object code processing, and conversational source language techniques for program testing and alteration.

5582

An operating system is described which will run on a wide variety of configurations of the I.C.T. 1900, and can handle a large number of online console users while at the same time running several offline (background) jobs. The system is not oriented towards either mode and can be either a batch processing system (such as the ATLAS supervisor, IBSYS, or GECOS), or a multi-access system (resembling, to the user, CTSS or MULTICS), or both simultaneously, depending on the installation, which can adjust the schedulers. Both online users and offline jobs use a common command language. The system includes a multilevel device-independent file store.

6) HUMAN COMMUNICATION, DOCUMENTATION, AND HUMANITIES

5583

A bracketed grammar is a context-free grammar in which indexed brackets are inserted around the right-hand sides of the rules. The language generated by a bracketed grammar is a bracketed language. An algebraic condition is given for one bracketed language to be a subset of another. The intersection and the difference of two bracketed languages with the same brackets and terminals are context-free (although not necessarily bracketed) languages. Whether \( L(G_1) \subseteq L(G_2) \) and whether \( L(G_1) \cap L(G_2) \) is empty are solvable problems for arbitrary bracketed grammars \( G_1 \) and \( G_2 \) with the same brackets and same terminals. Finally, bracketed languages are shown to be codes with strong properties.

Languages Defined by Real-Time On-Line Multitape Turing Machines—see 5545.

Turing Machine Memory Requirements for Context-Free Language Recognition—see 5546.

5584

The "information explosion" noted in recent years makes it essential that storage requirements for all information be kept to a minimum. A fully automatic and rapid three-part compressor which can be used with "any" body of information to greatly reduce slow external storage requirements and to increase the rate of information transmission through a computer is described in this paper. The system will also automatically decode the compressed in-
formation on an item-by-item basis when it is required. The three component compressors, which can be used separately to accomplish their specific tasks, are discussed: NUPAK for the automatic compression of numerical data, ANPAK for the automatic compression of “any” information, and IOPAK for further compression of information to be stored on tape or cards.

Computer System for Inference Execution and Data Retrieval—see 5575.

Gain Parameters for a Typical Information Retrieval System—see 5576.

7) BEHAVIORAL SCIENCE, PATTERN RECOGNITION, AND ARTIFICIAL INTELLIGENCE

Proceedings of a Symposium on Computer and Information Sciences—see 5531.

Multilevel Adaptive Pattern Classification Machine—see 5567.

Statistical Impulse Generator for a Neuron—see 5596.

Automatic Theorem Proving with Renamable and Semantic Resolution—see 5534.

Concept of Demodulation in Theorem Proving—see 5585.


Programs to solve combinatorial search problems may often be simply written by using multiple-valued functions. Such programs, although impossible to execute directly on conventional computers, may be converted in a mechanical way into conventional backtracking programs. The process is illustrated with algorithms to find all solutions to the eight queens problem on the chessboard, and to find all simple cycles in a network.


This report describes progress toward an “intelligent question-answering system”—a system that can accept facts, retrieve items from memory, and perform logical deductions necessary to answer questions. Two versions of such a system have been implemented, and the authors expect these to be the first in an evolving series of question-answering systems. The first system, QA1, is based on a list-structured memory that can be used to store the memory organization is simpler than that of QA1, the sophisticated logical abilities of QA2 result in greater question-answering power. The report gives examples of the performance of QA1 and QA2 on typical problems that have been done by previous question-answering systems, and describes plans for extending the capabilities of QA2.

8) MATHEMATICS

On-Line Computer System for Algebraic Manipulation—see 5564.

Transcendental Equation Analysis—see 5600.


Two general methods of matrix inversion, Gauss’s algorithm and the method of bordering, are analyzed from the viewpoint of their adaptability for parallel computation. The analysis is not based on any specific type of parallel processor; its purpose is to show if parallel capabilities could be used effectively in matrix inversion. It is shown that both methods are indeed able to make effective use of parallel capability. With reasonable assumptions on the parallelism that is available, the speeds of the two methods are roughly comparable. The two methods, however, make use of different kinds of parallelism. To implement Gauss’s algorithm one would like to have 1) parallel transfer capability for $n$ numbers, if the matrix is $n \times n$, 2) the capability for parallel multiplication of the accessed numbers by a common multiplier, and 3) parallel additive read-in capability. For the method of bordering, one needs, primarily, the capability of forming the Euclidean inner product of two $n$-dimensional real vectors. The latter seems somewhat harder to implement, but, because it is an operation that is fundamental to linear algebra in general, it is one that might be made available for other purposes. If so, then the method of bordering becomes of interest.


A technique for computing the fixed-point probability vector of an ergodic cyclic transition matrix $P$ is developed. The technique utilizes generalized matrix inversion in a scheme which only necessitates calculation of the probability fixed point of a transition matrix having smaller dimensions than $P$.


The general problem considered is that of solving a linear system of equations which is singular or almost singular. A method is described which obtains a “solution” to the system which is stable with respect to small changes in the matrix elements. This method will solve an overdetermined system in $m$ variables and $n$ equations ($m < n$) even when the system rank is less than $m$, and should therefore be very useful in many statistical applications. In this case the error of the system is minimized in the Chebyshev norm using a linear programming formulation and solution. A numerical example using the Hilbert matrix is described in detail.

Fourier Transform Analog Computer—see 5602.

Instability in the Numerical Integration of Differential Equations—see 5598.


In a previous paper the authors suggested that the accurate correctors proposed by Gregg and Stetter for solving ordinary differential equations should be accompanied by similar generalizations of each method in that paper. They state that the predictors use one “nonstep” point between the interval of integration. In the present paper a corrector is dealt with in which two “nonstep” points are used, and in which, to some degree, the authors have “balanced” the contributions of the errors in the predictors to the total local truncation error, a technique due to Butcher.


This paper deals with the approximation of weak solutions of nonlinear elliptic equations of the form

$$\sum_{i=1}^{n} (\partial^2 f(x))/\partial x_i \partial x_i + c(x)u = f,$$

where either $f=f(x, u)$ or $f=f(x, u, \nabla u)$. The differential equation is replaced by a difference equation and convergence of the solutions of the difference equations to the solution of the differential equation is proven by functional analytic means. A unified treatment of the convergence of solutions of elliptic difference equations to the solution of the elliptic differential equation is then given.


The analog-hybrid computer Monte Carlo technique for solving elliptic and parabolic partial differential equations has been implemented on a new hybrid computer capable of taking statistics over 1000 two- or three-dimensional random walks per second. This exceptional computing speed and flexible digital control permit direct plotting of partial differential equation solutions; of
perhaps even greater interest is the incorporation of such Monte Carlo routines in real-time analog computer setups in process control applications. In this connection, the Monte Carlo method has been extended to a wider class of problems, and is especially applicable to heat conduction/diffusion problems.


Theoretical methods, based on a priori pointwise bounds, for approximating solutions of many elliptic and parabolic initial and/or boundary value problems, have been developed in recent years. These methods, however, are relatively unknown to potential users since applications of the methods have not appeared in the literature. In this paper their usefulness is illustrated by employing some of the author’s theoretical results as a basis for the construction of a digital program to compute an approximate solution of an initial boundary value problem for the heat equation.


This note considers the graphs of linear and affine transformations on modules over Boolean rings. It is shown that such graphs are similar to those of linear forms on vector spaces over finite fields.

Topological Properties of Linear Graphs—see 5598.

9) PROBABILITY, MATHEMATICAL PROGRAMMING, DIGITAL SIMULATION, INFORMATION THEORY, AND COMMUNICATION SYSTEMS


Pseudo-random number generators of the power residue (sometimes called congruential or multiplicative) type are discussed and results of statistical tests performed on specific examples of this type are presented. Tests were patterned after the methods of MacLaren and Marsaglia (M&M). The main result presented is the discovery of several power residue generators which performed well in these tests. This is important because of all the generators using standard methods (including power residue) that were tested by M&M none gave satisfactory results. The overall results give further evidence for their conclusion that the types of tests usually encountered in the literature do not provide an adequate index of the behavior of n-tuples of consecutively generated numbers. In any Monte Carlo simulation problem where n supposedly independent random numbers are required at each step, this behavior is likely to be important. Finally, since the tests presented differ in certain details from those of M&M, some of their generators were retested as a check. A cross-check shows that results are compatible; in particular, if a generator failed one of their tests badly, it also failed the present author’s corresponding test badly.

Analysis of Data from Computer Simulation of Economic Systems—see 5604.

Queuing Model for an Economic System—see 5604.


Following a review of terms used in information theory, conventional communications systems are examined. Functional models that lead to a statistical impulse generator are developed for a neuron and a fish. The problems of information generation, reduction, and storage in man are considered. The concept of bidirectional communication is advanced where the receiver as well as the transmitter is active and the theory of bidirectional communication is shown to be a generalization of Shannon’s information theory.


Such phrases as “information flow” may be purely metaphorical, or may refer to portage and storage of physical documents, transmission of signals, power required for signaling, Shannon’s selective information, changes in the state of one’s personal knowledge, propagation of announcements concerning messages, social increase of awareness, propagation of or reaction to imperatives, and so on. These matters are distinct and must be distinguished. Then conditions must be stated under which one can validly speak of and measure the appropriate flow. In this paper it is shown that within the field of notification (mention and delivery of recorded messages to users) there are twenty basic activities formed by choosing triads from the six variables, message, code, channel, source, destination, and designation. “Flow” has meaning only when two such triads have two variables in common, forming a tetrad. Then flow or correspondence between any pair of variables is inextricable from a conjugate flow or correspondence between the other pair. Between any pair of endpoints there are six possible distinct types of flow, according to which two of the remaining four variables are directly used to achieve the flow.

Correspondence Between Equivalence Classes of Switching Functions and Group Codes—see 5539.

Quasilinear Automata and the Codes Defined by Them—see 5543.

10) SCIENCE, ENGINEERING, AND MEDICINE


In this tutorial paper, the influence of the computer not only on the modus operandi of circuit design, but also on network theory itself, is discussed. The topological properties of linear graphs are reviewed and a matrix-topological formulation of the network problem is described. In addition to the graphical, nodal, and cutset methods, a mixed method of analysis is described which is applicable to dc, ac, and transient problems. Numerical methods of solving linear and nonlinear dc network problems are discussed and a new approach to ac analysis, using the mixed method and a numerical solution of the matrix eigenvalue problem, is described. The extension of this method to the transient analysis of linear networks is also explained. Finally, the problem of instability in the numerical integration of differential equations is discussed and several means of solving the problem are outlined.


Computer-aided analysis of electronic circuits is examined from the standpoint of the user. A general discussion of program requirements is presented, and several numerical and practical aspects of computer-aided electronic circuit design are given. The engineer’s role is discussed in terms of the information and insight needed by the user, and some of the methods utilized in developing computer analysis capabilities at a particular laboratory are given. On-line design is reviewed, with particular emphasis on graphical input and output systems. A design supervisor program is described that would render on-line circuit design a practical reality. Finally, the impact of on-line systems upon computer-aided analysis is examined with reference to the time-sharing system, OCTOPUS, at Lawrence Radiation Laboratory, Livermore, California.

LINGO Programming Language for Network Analysis at a Remote Teletype Terminal—see 5579.

Algorithm for Finding Cycles in a Network—see 5585.


The report describes the analysis and programming required in the development of eight computer programs which cover a wide variety of scientific research. Some of these investigations are interrelated, such as the geomagnetic co-ordinate conversion and magnetic field determinations, while others range from radar-trajectory analysis to electromagnetic theory. For each case, this report describes the mathematical development as well as the programming
Aspects in order to more completely define the problems and increase their applicability. While the problem descriptions of this document are mathematically oriented, the sections explaining the programming aspects of each problem are designed to facilitate the use of the computer programs by programmer and scientist alike. The specific topics covered in this report are: Orbiting Geophysical Observatory (OGO), Scattering Integration Analysis (SCATT), Electromagnetic Integration Analysis (INTE), Trajectory Analysis for Single Station Radar (TRAJ-1), Proton and Electron Ionosphere Models (ION), Magnetic Field and B and L Determination (FIELD), Geomagnetic Coordinate Conversion (GEMAG), Transcendental Equation Analysis (OPTIC).

Automatic Checkout Systems for Space Guidance Programs—see 5578.


An algorithm for deriving the primary sequence of a protein or RNA is presented. The data is in the form of short sequences of letters which must be fitted together to form the unknown complete sequence. A computer program for carrying out the steps is described, with an example. It is shown that the algorithm cannot make an error and empirical results are given which illustrate the successful use of the algorithm in reconstructing complete sequences known to be solvable.

11) ANALOG AND HYBRID COMPUTERS


The report describes the modifications of the Idealab Fourier Transform analog computer and the results obtained from these modifications. Indicated in the report is the final figure of a resolution for an input of 2249 sin wave cycles within one frequency increment (14 percent of the (sin x)/x crossover). Also indicated in the report are the methods and circuitry used to obtain this accuracy and operator instructions for the new controls. Emphasis is placed on the electronic and engineering considerations of the monitor panel, white light actuator, velocity servo, precision clamping, and oven control. A complete set of blue prints covering the modifications is provided at the end of the report.

Digital Simulation of an Analog System—see 5574.

Hybrid Computer Solution of Optimal Control Problems by the Maximum Principle—see 5603.

12) REAL-TIME SYSTEMS AND AUTOMATIC CONTROL; INDUSTRIAL APPLICATIONS


When the maximum principle is applied to optimal control problems, a two-point boundary value problem must be necessarily solved. This paper discusses a solution method for general problems in which the initial and terminal states of the system are given, but in which the time of arrival at the terminal state is unknown. Various difficulties accompanying the conventional methods used in the past are pointed out, and a new method for solving these difficulties is proposed. The hybrid computing system is suitable for carrying out this method, and hybrid computers have been applied to the solution of numerous concrete problems. As a result, it becomes possible to seek a solution completely automatically, and it has been proven that this method is extremely practical.

13) GOVERNMENT, MILITARY, AND TRANSPORTATION APPLICATIONS


14) BUSINESS APPLICATIONS OF INFORMATION PROCESSING


This paper addresses itself to the problem of analyzing data generated by computer simulations of economic systems. The authors first introduce a hypothetical firm, whose operation is represented by a single-channel, multistation queueing model. The firm seeks to maximize total expected profit for the coming period by selecting one of five operating plans, where each plan incorporates a certain marketing strategy, an allocation of productive inputs, and a total cost. The results of the simulated activity under each plan are subjected to an F-test, two multiple comparison methods, and a multiple ranking method. These techniques are illustrated, compared, and evaluated. The paper adopts the position that the particular technique of analysis (possibly not any one of the above) chosen by the experimenter should be an expression of his experimental objective; the F-test tests the homogeneity of the plans; multiple comparison methods quantify their differences; and multiple ranking methods directly identify the one best plan or best plans.

SUBJECT INDEX

A

Adaptive
Proceedings of a Symposium on Computer and Information Sciences 5531
Multilevel Adaptive Pattern Classification Machine 5567
Addition; Adders
Injection Laser Half Adder 5556
Algebra; Algebraic
Linear Positive Machines and a Commutative Banach Algebra 5548
Algebraic Theory of Sequential Machines 5549
On-Line Computer System for Algebraic Manipulation 5564
Chebyshev Solution of Large Linear Systems 5589
Transcendental Equation Analysis 5600

Algorithms
Partitioned List Algorithms 5536
Nondeterministic Algorithms 5585
Algorithms for:
Prime Implicant Determination from Canonical Forms 5536
Periodic Decomposition of Sequential Machines 5551
Computing Tests to Detect Failures in Logic Circuits 5557
Reconstructing Protein and RNA Sequences 5601
Analog Computers
Digital Simulation of an Analog System 5574
Monte Carlo Routines for Real-Time Analog Computer Setups in Process Control Applications 5592
Fourier Transform Analog Computer 5602

Approximation
Continuous Method of Approximating Solutions of the Heat Equation 5593
Arithmetic
Representation Error for Real Numbers in Binary Computer Arithmetic 5535
Machine Interval Arithmetic 5554
—see also Addition, Multiplication
Arrays
Large-Scale Integrated Cellular Threshold Array 5567
Memory Array Permitting Logic Manipulations 5569
Artificial Intelligence
Proceedings of a Symposium on Computer and Information Sciences 5531
Intelligent Question-Answering Systems 5586
—see also Patterns, Theorem Proving
Associative
Plated Wire Content-Addressable Memories with Bit Steering 5569

Automata; Turing Machines
Proceedings of a Symposium on Computer and Information Sciences 5531
Quasilinear Automata and the Codes Defined by Them 5543
Programming Languages for Automata 5544
Languages Defined by Real-Time On-Line Multitape Turing Machines 5545
Turing Machine Memory Requirements for Context-Free Language Recognition 5546
Solvability of the Halting Problem for 2-State Post Machines 5547
—see also Sequential

C

Circuits
Computer-Aided Design of Electronic Circuits 5599
—see also Integrated Circuits, Logic, Networks, Sequential

Codes; Coding; Decoding
Correspondence Between Equivalence Classes of Switching Functions and Group Codes 5539

Combinational
Diagnosis of Failures in Large Combinational Networks 5542
Cellular Threshold Arrays Realizing Combinational or Sequential Logic 5567

Combinatorial
Solving Combinatorial Search Problems 5585

Communications
Theory of Bidirectional Communication 5596
Morphology of Information Flow 5597

Compilers
Compiler for Reading and Executing LINGO Programs 5579
Compiler Generation Using Formal Specification of Procedure-Oriented and Machine Languages 5580
Compiler Organization Studies

Computer Applications
Defense Intelligence Agency Manual of Data Processing Standards 5532
Reliability of Redundant Spaceborne Digital Systems 5560
Aerospace Computers Based on Large-Scale Integration 5562
Automatic Checkout Systems for Space Guidance Programs 5578
Monte Carlo Routines for Real-Time Analog Computer Setups in Process Control Applications 5592
Computer Methods of Network Analysis 5598
Computer-Aided Design of Electronic Circuits 5599
Programs for Solving Space Science Problems 5600
Reconstructing Protein and RNA Sequences 5601
Analysis of Data from Computer Simulation of Economic Systems 5604
—see also Control, Education, Mathematics, Patterns, Retrieval

Computer Systems
Reliability of Redundant Spaceborne Digital Systems 5560
Computer System Organization Utilizing Large-Scale Integration 5561
Aerospace Computers Based on Large-Scale Integration
Experimental Model of System/360 5563
Predicting the Performance of Computer Systems 5563
On-Line Computer System for Algebraic Manipulation 5564
Bulk Core Storage in a 360/67 Time-Sharing System 5568
Computer System for Inference Execution and Data Retrieval 5575
—see also Digital Computers, Remote, Time Sharing

Computers
Proceedings of a Symposium on Computer and Information Sciences 5531
—see also Analog Computers, Computer Systems, Digital Computers, Time Sharing

Control
Hybrid Computer Solution of Optimal Control Problems by the Maximum Principle 5603

Cybernetics
Information Theory and Cybernetics 5596

D

Data
Computer System for Inference Execution and Data Retrieval 5575
Relational Data File 5575
Automatic Data Compression 5584

Data Processing
Defense Intelligence Agency Manual of Data Processing Standards 5532
—see also Computer Systems, Digital Computers, Education, Multiprocessing, Processing

Debugging; Checkout
Debugging, Testing, and Diagnosing Systems 5577
Automatic Checkout Systems for Space Guidance Programs 5578

Decision Tables
Conversion of Limited-Entry Decision Tables to Optimal Computer Programs 5571

Diagnosis
Diagnosis of Failures in Large Combinational Networks 5542
Debugging, Testing, and Diagnosing Systems 5577
—see also Error, Failures

Differential Equations (Ordinary)
Solution of Ordinary Differential Equations Using Two Off-Step Points 5590
Instability in the Numerical Integration of Differential Equations 5598

Differential Equations (Partial)
Functional Analytic Approach to Solving Nonlinear Elliptic Equations 5591
Monte Carlo Solution of Partial Differential Equations Using a Hybrid Computer 5592
Continuous Method of Approximating Solutions of the Heat Equation 5593

Digital Computers
Reliability Aspects of the Variable Instruction Computer 5558
Reliability of Redundant Spaceborne Digital Systems 5560
—see also Computer Systems, Processing

E

Education
Experiment in Computer-Aided Education 5533
Error; Ambiguity

Representation Error for Real Numbers in Binary Computer Arithmetic 5553
Linear Programming Minimization of Error in Solving Large Linear Systems 5589
—see also Codes, Failures

F

Failures; Faults
Computing Tests to Detect Failures in Logic Circuits 5557
Fault Detection and Location in Self-Repairing Digital Systems 5559
—see also Diagnosis, Reliability

Files
Relational Data File 5575
Bayes File Adjustment Procedure for Matched Term Retrieval Systems 5576

Games
Algorithm for Solving Chessboard Problems 5585

Graphs
Graphs of Linear Forms on Modules over Boolean Rings 5594
Topological Properties of Linear Graphs 5598
—see also Flow Graphs

H

Hybrid Computers
Monte Carlo Solution of Partial Differential Equations Using a Hybrid Computer 5592
Hybrid Computer Solution of Optimal Control Problems by the Maximum Principle 5603

I

Information
Proceedings of a Symposium on Computer and Information Sciences 5531
Gain Parameters for a Typical Information Retrieval System 5576
Automatic Information Compression 5584
Information Theory and Cybernetics 5596
Morphology of Information Flow 5597
—see also Retrieval

Integrated Circuits
Computer System Organization Utilizing Large-Scale Integration 5561
Aerospace Computers Based on Large-Scale Integration 5562
Large-Scale Integrated Cellular Threshold Array 5567
Design of a Parallel Bit-Organized Large-Scale Integrated MOS Memory 5570

L

Languages
Languages Defined by Real-Time On-Line Multitape Turing Machines 5545
Bracketed Context-Free Languages 5583
—see also Programming Languages

Lasers
Injection Laser Half Adder 5556

Lists
Partitioned List Algorithms 5536

Logic
Inference Rule for First-Order Predicate Calculus 5534
Inferences in Theorem Proving Programs 5535
Diagnosis of Failures in Logic Networks 5542
Computing Tests to Detect Failures in Logic Circuits 5557
ABSTRACTS OF CURRENT COMPUTER LITERATURE

Simulation; Simulators
Simulation of System/360 5563
Digital Simulation of an Analog System 5574
Analysis of Data from Computer Simulation of Economic Systems 5604
Standards
Defense Intelligence Agency Manual of Data Processing Standards 5532
Statistics
Bayes File Adjustment Procedure for Matched Term Retrieval Systems 5576
Solution of Large Linear Systems for Statistical Applications 5589
Statistical Impulse Generator for a Neuron 5596
Analysis of Data from Computer Simulation of Economic Systems 5604
—see also Monte Carlo, Probability
Storage; Memory
Bulk Core Storage in a 360/67 Time-Sharing System 5568
Plated Wire Content-Addressable Memories with Bit-Steering 5569
Design of a Parallel Bit-Organized Large-Scale Integrated MOS Memory 5570
Memory Allocation for Multiprocessors 5572
Buffer Storage Required for Random Input Word Rate 5573
Switching
Synthesis of Signal Switching Networks with Transient Blocking 5541
Optimal Design and Synthesis of Switching Circuits 5566
Switching Functions
Prime Implicant Determination from Canonical Forms 5536
Minimization of Boolean Functions Containing Arbitrary Parameters 5537
Minimal Modulo 2 Sums of Products for Switching Functions 5538
Correspondence Between Equivalence Classes of Switching Functions and Group Codes 5539
Symmetric Ternary Threshold Switching Functions 5540
—see also Automata, Logic, Sequential Machines
Systems
Reliability of Redundant Spaceborne Digital Systems 5560
Debugging, Testing, and Diagnosing Systems 5577
—see also Communications, Computer Systems, Operating Systems

Teletype
LINGO Programming Language for Network Analysis at a Remote Teletype Terminal 5579

Terminals
LINGO Programming Language for Network Analysis at a Remote Teletype Terminal 5579

Tests
Computing Tests to Detect Failures in Logic Circuits 5557
Redundant System Test Point Allocation Procedure 5560
Generating Test Programs from Syntax 5577

Theorem Proving
Automatic Theorem Proving with Renamable and Semantic Resolution 5534
Concept of Demodulation in Theorem Proving 5535
Question-Answering System Based on Theorem Proving Techniques 5586

Threshold
Symmetric Ternary Threshold Switching Functions 5540
Large-Scale Integrated Cellular Threshold Array 5567

Time Sharing
Bulk Core Storage in a 360/67 Time-Sharing System 5568
GEORGE 3 General Purpose Time-Sharing and Operating System 5582
On-Line Time-Sharing Design of Electronic Circuits 5599
—see also Remote

Transistors
Computer-Aided Transistor Design, Characterization, and Optimization 5565
—see also Integrated Circuits

Turing Machines
—see Automata

AUTHOR INDEX

A
Anderaa, S. 5547
Alligower, E. 5591
Anar, V. 5542
Apostolatos, N. 5554

B
Bailey, M. J. 5582
Barbieri, R. S. 5581
Barnes, B. H. 5550
Bigelow, R. H. 5544
Bouricius, W. G. 5557
Branin, Jr., F. H. 5598
Brush, D. G. 5590
Bucy, R. S. 5548
Bullard, A. H. 5602
Burkhardt, W. H. 5577

C
Carson, D. F. 5535
Chow, W. F. 5569
Clark, C. O. 5578
Coates, C. L. 5552
Cole, F. B. 5539
Condulmari, N. 5542
Coury, L. 5560

D
Dawson, D. F. 5599
Decell, H. P. 5588
De La Morea, P. H. 5565
de Maine, P. A. D. 5584
Dono, N. R. 5565
Dor, N. M. 5573

E
Even, S. 5538
Fairthorne, R. A. 5597
Fenichel, R. R. 5564
Fitzgerald, J. M. 5550
Flexer, J. R. 5531
Floyd, R. W. 5585

G
Ghosh, H. N. 5565
Gilbert, P. 5580
Gill, A. 5551
Ginsburg, S. 5583
Goldstein, L. J. 5541
Green, C. C. 5586
Guenter, R. 5591

H
Hachtel, G. D. 5566
Handler, H. 5592
Hannigan, J. M. 5560
Harlow, C. A. 5552
Harrison, M. A. 5583
Hartmanis, J. 5546
Hawryszkiewycz, I. T. 5574
Hotz, G. 5543

I
Iwata, J. 5603

J
Joseph, E. C. 5562

K
Katz, J. H. 5563
Kautz, W. H. 5667
Kawaji, M. 5586
Kim, K. S. 5600
Kirsh, E. K. 5537
Knuth, D. E. 5544
Kohavi, I. 5538
Kohfeld, J. J. 5590
Kolod, R. M. 5541
Kulisch, U. 5554
Kuo, F. F. 5599

L
Lauer, H. C. 5568
Lechner, R. J. 5539
Lee, H. B. 5533
Leipholz, S. W. 5541
Levin, R. E. 5575
Levy, S. Y. 5561
Linhardt, R. J. 5561

M
Magnuson, Jr., W. G. 5599
Marko, H. 5596
Marno, M. E. 5575
Marron, B. A. 5584
Masters, Jr., C. G. 5560
Matesosian, R. 5549
McKeeman, W. M. 5553
McLelland, W. G. 5580
Meredith, P. H. 5578
Merrill, Jr., R. D. 5540

N
Naylor, T. H. 5564
Nemoto, T. 5556

O
Odell, P. L. 5588
Oestreich, M. D. 5582

P
Pasqualini, R. 5570
Paz, A. 5538
Pease, M. C. 5587
Peeler, D. L. 5578

R
Raphael, B. 5586
Reinwald, L. T. 5571
Rhodes, J. 5549
Richards, L. M. 5578
Robinson, G. A. 5535
Rohrer, R. A. 5566
Rosen, J. B. 5589
Rosenberg, A. L. 5545
Rosen, A. F. 5572
Roth, Jr., C. H. 5537
Roth, J. P. 5557

S
Schneider, P. R. 5557
Shalla, L. 5535
Shapiro, M. 5601
Shekel, J. 5579
Shumway, R. H. 5576
Sidman, R. D. 5561
Sigillito, V. G. 5593
Slagle, J. R. 5534
Soland, R. M. 5571
Strauss, J. I. 5582

T
Thompson, G. T. 5590
Thornton, R. D. 5533
Tou, J. T. 5531
Tsuda, J. 5603

V
Van Gelder, A. 5595

W
Wang, K. C. 5594
Wertz, K. 5604
Winograd, S. 5555
Wonnacott, T. H. 5604
Wos, L. 5535

Y
Yonezu, H. 5566

Z
Zimmerman, S. E. 5559