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

0) GENERAL; PEOPLE AND SOURCES; EDUCATION

5771
Computers Then and Now, M. V. Wilkes (Cambridge U., England); *J. ACM*, vol. 15, pp. 1–7, January 1968.

Reminiscences on the early developments leading to large-scale electronic computers show that it took much longer than was expected for the first of the more ambitious and fully engineered computers to be completed and prove themselves in practical operation. Comments on the present computer field assess the needs for future development. The author feels that revolutionary advances can come only by the exploitation of the high degree of parallelism that the use of integrated circuits will make possible. One area in which a high degree of parallelism is promising is that of pattern recognition in two dimensions. Many problems in symbol manipulation have a large element of pattern recognition in them, a good example being syntax analysis. The author does not exclude the possibility that there may be some big conceptual breakthrough in pattern recognition which will revolutionize the whole subject of computing.

5772
Rules of Ethics in Information Processing, D. B. Parker, Chairman, ACM Professional Standards and Practices Committee (Control Data, Palo Alto); *Commun. ACM*, vol. 11, pp. 198–201, March 1968.

The background and motivation for the adoption by the ACM Council on November 11, 1966, of a set of Guidelines for Professional Conduct in Information Processing are described. A brief history is given of ethical codes in other professions. Some reasons for and against adoption of ethical rules are considered, and several sections of the ACM Guidelines are analyzed. The purpose is to inform about this important aspect of our profession, as well as to stimulate thought and interest.

5773
Proposed USA Standard—General-Purpose Alphanumeric Keyboard Arrangement for Information Interchange; *Commun. ACM*, vol. 11, pp. 126–129, February 1968.

This USA Standard arrangement is intended principally for general-purpose typewriter-like alphanumeric keyboards implementing the USA Standard Code for Information Interchange (ASCII) (USAS ×3.4-1967). This keyboard arrangement was developed from a study of keyboards already in use by millions of trained operators in the teleprinter, typewriter, and related fields. Also included in this study were the USA Standard Typewriter Keyboards (USAS ×4.7-1966) and the applicable standards work being carried out internationally.

5774
Proposed USA Standard—USASCSOCR Dual Case Keyboard Arrangement; *Commun. ACM*, vol. 11, pp. 130–132, February 1968.

This USA Standard presents a standard dual case keyboard arrangement to implement the USA Standard Character Set for Optical Character Recognition (USASCSOCR×3.17-1966), and is one of a proposed series of standards on Optical Character Recognition (OCR) arrangements. The greatest benefit to the user will be in the areas of operator training, purchasing, and interchangeability of operators and machines. Present user demands dictated that there be an OCR keyboard capable of preparing dual case correspondence in addition to OCR input. This standard is applicable to both electric and manual machines in this environment.

5775
Marking and Evaluating Class Tests and Examinations by Computer, P. D. Groves (U. of Aston, Birmingham, England); *Computer J.*, vol. 10, pp. 365–367, February 1968.

A computer is capable of producing more information from a series of tests than just a set of marks and an ALGOL program has been written for an Elliott 803 computer which will 1) mark multiple-choice type questionaires, 2) comment on the progress of each student, 3) summarize the results obtained by the class, question by question, and 4) provide information on the ability of individual questions to discriminate between able and less able students. The procedures used are outlined and discussed. A print-out of the program is available from the author as is also a set of operating instructions for its use on the Elliott 803.

5776
Extensions to the Heuristic Algorithm for University Timetables, A. P. Yule (CERN, Geneva, Switzerland); *Computer J.*, vol. 10, pp. 360–364, February 1968.

An alternative method of data handling in the heuristic algorithm for University timetables is suggested and methods of including various constraints on the final solution are given. The extended algorithm includes the allocation of suitable rooms to lectures, the consideration of lecturers' preferences, dynamic allocation of a lecturer's free day, and the spreading of similar lectures over the week.

5777
Curriculum 68: Recommendations for Academic Programs in Computer Science; *Commun. ACM*, vol. 11, pp. 151–197, March 1968.

This report contains recommendations on academic programs in computer science which were developed by the ACM Curriculum Committee on Computer Science. A classification of the subject areas contained in computer science is presented and twenty-two courses in these areas are described. Prerequisites, catalog descriptions, detailed outlines, and annotated bibliographies for these courses are included. Specific recommendations are given for undergraduate programs. Graduate programs in computer science are discussed, and some recommendations are presented for the development of master's degree programs. Ways of developing guidelines for doctoral programs are discussed, but no specific recommendations are made. The importance of service courses, minors, and continuing education in computer science is emphasized. Attention is given to the organization, staff requirements, computer resources, and other facilities needed to implement computer science educational programs.

1) LOGIC AND SWITCHING THEORY; SEQUENTIAL MACHINES

5778
Logic, Logical Design, and Digital Circuits, H. K. Sherer (Vitro, Florida); December 1967, 103 pp., APGC-TR-67-141; *U. S. Gov't R&D Repts.*, vol. 68, p. 61(A), February 25, 1968. AD 662 878 CFSTI $3.00.

The report begins with a discussion of logical propositions and their mathematical extension—Boolean algebra. This is followed by various representations of Boolean functions including the graphical method. The concept of designation numbers as a unique circuit description is developed followed by simplification methods and reduction by maps. Application of these methods to actual circuit design is demonstrated by various examples. In the final portion the more common digital building blocks are presented and discussed.

5779
Symmetry Types in Threshold Logic, R. O. Winder (RCA Labs., Princeton); *IEEE Trans. Computers*, vol. C-17, pp. 75–78, January 1968.

The idea of "similarity" between threshold functions is developed in a unified fashion and illustrated geometrically. Formulas are given to transform the corresponding threshold gate realizations.

5780
An Adaptive Threshold Logic Gate Using Capacitive Analog Weights, J. R. Smith, Jr. (U. of Texas, Austin) and C. O. Harbourt (U. of Missouri, Columbia); *IEEE Trans. Computers*, vol. C-17, pp. 78–81, January 1968.

A physical realization of an adaptive threshold logic gate which can be used to realize linearly separable switching functions is presented. The system is completely electronic and is easily implemented in the sense that standard components may be used throughout. The memory circuit used for each weight of the device stabilizes the voltage across a capacitor by means of a sampling technique. Over the range that the capacitor voltage is stabilized, the net average capacitor current is zero at a discrete number of stable voltages which can be made numerous enough to approximate analog memory. The device exhibits both long-term stability and ease in weight adjustment. Training results are given for several linearly separable switching functions. This circuit shows that a practical adaptive threshold logic gate can be realized. The incorporation of integrated circuitry in the device presented would greatly enhance the feasibility of building more complex trainable machines.

5781
An Approach for the Realization of Multithreshold Elements, C. W. Mow (Litton Industries, Woodland Hills) and K. S. Fu (U. of California, Berkeley); *IEEE Trans. Computers*, vol. C-17, pp. 32–46, January 1968.

An algorithm for the realization of k-threshold realizable functions is presented. Instead of solving the set of linear inequalities, where the unknowns are the weights corresponding to the input variables, incremental weights are sought. The procedure reduces to that of resolving contradicting pairs of vertices by the incremental weights. The minimum number of thresholds are sought for each complementation and permutation of input variables. A definition of an optimal multithreshold weight threshold vector is derived from the reliability viewpoint. The desired solution is obtained through a search of all possible obtainable realization vectors of the function. For single-threshold realizable functions, permutation and complementation of input variables need not be considered if the input variables of the function are ordered and positivized. The procedure is systematic and has been programmed in FORTRAN IV. As a comparison with Haring and Ohori's tabulation on the 221 equivalence classes of four-variable Boolean functions under the NPN operation, it can be seen that 42 of the 221 equivalence classes need fewer numbers of thresholds for their realization. For the same number of thresholds, 58 equivalence classes have less absolute sum of weights. Finally, with the number of thresholds and absolute sum of weights being equal, 36 equivalence classes have lower threshold values.

5782
Input Tolerance Considerations for Multithreshold Threshold Elements, C. W. Mow (Litton Industries, Woodland Hills) and K. S. Fu (U. of California, Berkeley); *IEEE Trans. Computers*, vol. C-17, pp. 46–54, January 1968.

This paper deals with input tolerance considerations of a multithreshold threshold element. The concept of a multilevel minus-plus-one model of a threshold element is introduced. It is shown that the multilevel minus-plus-one model is equivalent to the zero-one model of a multithreshold element. The equivalence is established through a set of defining equations similar to that of a single-threshold threshold element. For a nonlinearly separable Boolean function, more than one multithreshold weight threshold vector may exist for its realization, even though the number of thresholds and the sum of the absolute magnitude of all the input weights are the same; i.e., $\sum_i^n=1|w_i|$. From reliability considerations, the optimal synthesis of the multithreshold threshold elements is shown to depend on the magnitudes of max $(|T_i|, |T_k|)$ in addition to the minimum number of thresholds needed to realize the Boolean function and the min $\sum_i^n=1|w_i|$. An example is given for illustrative purposes of choosing an optimal realization. For a given margin of operation with the error model assumed, it is shown that one need only work from the normalized set of inequalities. The solution with margins specified is obtained from the optimal solutions of the normalized system of inequalities.

5783
Generation of Self-Dual and Self-Complementary Dual Functions, C. W. Mow (Litton Industries, Woodland Hills) and K. S. Fu (U. of California, Berkeley); *IEEE Trans. Computers*, vol. C-17, pp. 57–66, January 1968.

Properties of self-dual and self-complementary dual functions are discussed. Necessary and sufficient conditions of self-dual and self-complementary dual functions are obtained in terms of the multithreshold weight threshold vector. In particular, self-dual and self-complementary dual functions are shown to be realizable only by an odd and even number of effective thresholds, respectively. A threshold T_i is effective if $E_{\min}<T_i<E_{\max}$. It is shown that $(n+1)$-variable self-dual and self-complementary dual functions can always be generated from 1- and 2-effective-threshold weight threshold vectors of n-variable Boolean functions, respectively. If the number of effective thresholds exceeds 2, constraints on the thresholds must be met in order to generate $(n+1)$-variable self-dual and self-complementary dual functions. Such generations of self-dual and self-complementary dual functions are shown to correspond to the functional forms of self-dualization and self-complementary dualization of an n-variable Boolean function. Moreover, they are realized by the same threshold vector T. Furthermore, it is shown that if an n-variable Boolean function $F_n(X)$ is self-dual or self-complementary dual with weight threshold vector $[W_n; T]$, then an $(n+m)$ variable self-dual or self-complementary dual Boolean function $F_{n+m}(N)$, where m is any positive integer, can be realized by a weight-threshold vector $[W_{n+m}; T]$.

5784
Ternary Cellular Cascades, M. Yoeli (Technion-Israel Inst. Tech., Haifa); *IEEE Trans. Computers*, vol. C-17, pp. 66–67, January 1968.

This note discusses ternary cellular cascades, i.e., one-dimensional arrays of ternary two-input one-output cells. It is shown that every ternary combinational switching function can be realized by such a cascade.

5785
A Partitioning Method for Combinational Synthesis, D. R. Smith (State U. of New York, Stony Brook); *IEEE Trans. Computers*, vol. C-17, pp. 72–75, January 1968.

A method of network synthesis for general combinational functions that uses a number of fixed threshold logic units not greatly in excess of the theoretical minimum is described. It can be used with adaptive logic in which case a minority of the interconnections are simply adapted once according to the rule originally suggested by Hebb. This network on its own has no capacity for "generalization," and if used for pattern recognition must therefore be operated in conjunction with a suitable property filter, such as that of the author's previous paper. In this case the present network could perform as a trainable "categorizer."

5786
Algebraic Automata and Context-Free Sets, J. Mezei and J. B. Wright (IBM, Yorktown Heights); *Information & Control*, vol. 11, pp. 3–29, July–August 1967.

The concepts of "equational" and "recognizable" are defined for sets of elements of an arbitrary abstract algebra. Context-free languages and finite-state languages become realizations of the same general concept (of equational sets) when the proper algebra is specified. A principal objective of the paper is to establish in the context of abstract algebra, relationships between such concepts as equational and recognizable sets.

5787
Two-Way Pushdown Automata, J. N. Gray, M. A. Harrison, and O. H. Ibarra (U. of California, Berkeley); *Information & Control*, vol. 11, pp. 30–70, July–August 1967.

In this paper, a new type of automaton, called a two-way pushdown automaton, is defined and studied. The model is a generalization of a pushdown automaton in that two-way motion is allowed on the input tape which is assumed to have endmarkers. The model is investigated in both the nondeterministic and deterministic cases. A number of basic results are obtained which include relationships with other families, closure and nonclosure results, and decidability properties. Certain special cases are studied such as the cases when the input alphabet has one letter and the device has no endmarkers.

5788
Turing Machines with a Schedule to Keep, P. C. Fischer (Cornell U., Ithaca); *Information & Control*, vol. 11, pp. 138–146, July–August 1967.

It is shown that if a sequence of digits $\alpha=\alpha_1\alpha_2\alpha_3\ldots$ is generated by a multitape Turing machine so that the nth digit α_n is produced no later than time $T(n)$, where T is a real-time-countable function, then there exists a multitape Turing machine which generates α and produces α_n at *exactly* time $T(n)$. This result makes it possible to strengthen some theorems of Hartmanis and Stearns and to answer a question of Yamada. Other applications yield a more general approach to countability and a new technique for dealing with time-limited computational complexity.

Graph Model of a Self-Producing Universal Turing Machine—*see* 5820.

5789
A Mathematical Model of Finite Random Sequential Automata, F. Stănciulescu (Computation Lab. of the Elec. Res. Inst., Bucharest, Rumania) and M. F. A. Oprescu (Ctr. Economic Computation and Economic Cybernetics, Bucharest, Rumania); *IEEE Trans. Computers*, vol. C-17, pp. 27–31, January 1968.

The sequential logic theory previously advanced is generalized to operations between variables of maximum word length n. Based on the generalized sequential logic operations, a new kind of probability, termed *sequential probability*, is introduced. By means of sequential probability, a mathematical model can be obtained for finite ran-

dom sequential automata. This model can be used for analysis and synthesis of random sequential automata.

5790
On Expediency and Convergence in Variable-Structure Automata, B. Chandrasekaran (Philco-Ford, Blue Bell) and D. W. C. Shen (U. of Pennsylvania, Philadelphia); *IEEE Trans. Systems Science and Cybernetics*, vol. SSC-4, pp. 52–60, March 1968.

A stochastic automaton responds to the penalties from a random environment through a reinforcement scheme by changing its state probability distribution in such a way as to reduce the average penalty received. In this manner the automaton is said to possess a variable structure and the ability to learn. This paper discusses the efficiency of learning for an m-state automaton in terms of expedience and convergence, under two distinct types of reinforcement schemes: one based on penalty probabilities and the other on penalty strengths. The functional relationship between the successive probabilities in the reinforcement scheme may be either linear or nonlinear. The stability of the asymptotic expected values of the state probability is discussed in detail. The conditions for optimal and expedient behavior of the automaton are derived. Reduction of the probability of suboptimal performance by adopting the Beta model of the mathematical learning theory is discussed. Convergence is discussed in the light of variance analysis. The initial learning rate is used as a measure of the overall convergence rate. Learning curves can be obtained by solving nonlinear difference equations relating the successive expected values. An analytic expression concerning the convergence behavior of the linear case is derived. It is shown that by a suitable choice of the reinforcement scheme it is possible to increase the separation of asymptotic state probabilities.

5791
Minimization Theorems and Techniques for Sequential Stochastic Machines, A. Paz (U. of California, Berkeley); *Information & Control*, vol. 11, pp. 155–166, July–August 1967.

This lecture contains a survey of most known theorems related to the problem of reducing the number of states of a given sequential stochastic machine to a minimum. Some known results are presented from a new point of view and a new technique for reducing the number of states of a given machine with given initial distribution is introduced. Some new related concepts are introduced and discussed.

5792
Maximal Memory Binary Input–Binary Output Finite-Memory Sequential Machines, M. M. Newborn (Columbia U., New York); *IEEE Trans. Computers*, vol. C-17, pp. 67–71, January 1968.

Gill has shown that if there exists a finite-memory n-state sequential machine with finite memory μ, then μ cannot exceed $\frac{1}{2}n(n-1) \triangleq N_n$. He has further shown that there exists an n-state N_n input–binary output machine with memory $\mu = N_n$ for every n. The question of whether a tighter upper bound might be placed on μ by the order of the input alphabet was raised by Gill. Massey recently has shown that there exists a ternary input–binary output finite-memory machine with memory $\mu = N_n$ for every n. The primary purpose of this note is to show that for every n there exists an n-state binary input–binary output finite-memory machine with memory $\mu = N_n$, and thus μ is shown not to be limited by the order of the input alphabet. It is shown that for every n there are actually at least two different machines with memory $\mu = N_n$. It will also be shown that for every n there exists a binary input–binary output n-state finite-memory machine with $\mu = N_n - 1$.

5793
On Equivalence of State Assignments, M. A. Harrison (U. of California, Berkeley); *IEEE Trans. Computers*, vol. C-17, pp. 55–57, January 1968.

Using a new definition for the equivalent of state assignments, the number of nonequivalent state assignments is derived. The exact number of nondegenerate state assignments is also computed.

5794
Definite Asynchronous Sequential Circuits, J. A. Brzozowski (U. of Waterloo, Canada) and S. Singh (U. of Ottawa, Canada); *IEEE Trans. Computers*, vol. C-17, pp. 18–26, January 1968.

An asynchronous unit delay is an n-input n-output asynchronous sequential circuit in which the present value of the output n-tuple is equal to the value of the input n-tuple prior to the last input change. This paper considers the problem of determining when a fundamental mode flow table is realizable as a feedback-free connection of asynchronous unit delays. It is shown that such a realization exists if and only if the flow table is asynchronous definite, where the asynchronous definite property is a modification of the definite property of synchronous sequential machines. A straightforward method of realizing asynchronous definite flow tables without critical races by feedback-free circuits of asynchronous unit delays and combinational gates is developed. The use of asynchronous unit delays for definite tables avoids complicated secondary assignment problems, results in circuits with very simple structure, and brings closer the theories of synchronous and asynchronous sequential machines.

Iterative Array of Sequential Machines Similar to Distributed Logic Associative Memory—*see* 5795.

2) DIGITAL COMPUTERS AND SYSTEMS

Software Package for a Floating Point Arithmetic Unit—*see* 5810.

5795
An Iteratively Structured General-Purpose Digital Computer, J. N. Sturman (Bell Tel. Labs., Murray Hill); *IEEE Trans. Computers*, vol. C-17, pp. 2–9, January 1968.

A general-purpose synchronous stored-program digital computer is described. The computer is composed solely of many small, identically structured sequential machines, each machine having fewer than 2^{11} states. The one-dimensional iterative array of sequential machines, or cells, which constitutes this computer is similar to the distributed logic associative memory originally proposed by Lee. A significant difference in the machine's structure, however, arises from the fact that instructions are stored within, and are under the control of, other cells, rather than a central processing unit. No specific area of this machine acts solely as a memory, an arithmetic unit, or a control unit, but rather, one or more cells within the iterative array perform all of the above tasks.

5796
Asynchronous Operation of an Iteratively Structured General-Purpose Digital Computer, J. N. Sturman (Bell Tel. Labs., Murray Hill); *IEEE Trans. Computers*, vol. C-17, pp. 10–17, January 1968.

The preceding paper proposed a general-purpose digital computer structure consisting of many small iteratively connected sequential machines. A set of machine instructions was presented and a programming example of synchronous machine operation was given. This paper examines some of the problems associated with asynchronous operation of this iteratively structured digital computer. A solution is presented enabling the many small sequential machines which constitute the large iterative computer to communicate asynchronously. The instruction set outlined in the previous paper is maintained and a pictorial example of asynchronous operation is presented.

Logical Design of Digital Circuits and Building Blocks—*see* 5778.

Intelligent Machine with Application to Logic Design of Switching Systems—*see* 5820.

5797
Computers with Core-Diode Elements, Y. A. Makhmudov; June 1967, 14 pp., Rept. FTD-HT-67-48; *U. S. Gov't R&D Repts.*, vol. 68, p. 60(A), February 25, 1968. AD 662 793 CFSTI $3.00.

The author describes a digital computer developed at the Computation Center of the Azerbaydzhan Academy of Sciences, using sequential and in some cases parallel-sequential units. The computer employs commercially produced ferrite-diode elements, operating at a low timing frequency (30 kHz). The article describes the principles underlying individual units of the computer, which can be used either independently or as parts of other computer systems. All the computer units operate with 24-digit binary numbers (including the sign digit) with fixed radix. The units described are a universal arithmetic unit of sequential action, the information input unit, the external magnetic-tape memory, and the output unit. The computer employs a total of 88 simple and 50 logic elements.

5798
DDA Scaling Graph, R. J. Leake and H. L. Althaus (U. of Notre Dame, Ind.); *IEEE Trans. Computers*, vol. C-17, pp. 81–84, January 1968.

Finite graph theory is applied to the problem of scaling fixed point Digital Differential Analyzers. The DDA scaling graph is introduced as a representation of the essential scaling information rather than the conventional integrator servo program network. Using the scaling graph, optimal scales are obtained simply, by constructing a maximum distance tree.

3) LOGIC DEVICES AND CIRCUITS (HARDWARE)

5799
Pulse Noise Immunity in Saturated Logic Gates, S. K. Ghandi and F. L. Thiel (Rensselaer Polytech. Inst., Troy); *IEEE J. Solid-State Circuits*, vol. SC-2, pp. 81–86, September 1967.

This paper develops techniques for assessing the inherent pulse noise immunity of saturated logic gates, with a view towards determining their ability to operate reliably in a pulse noise environment. A test method has been outlined for specifying and measuring this noise immunity. Although this method is applicable to all forms of saturated logic gates, the low-level T^2L gate has been singled out for experiment because of its high-speed capability. Using both discrete component and microcircuit gates of this type, close correlation was obtained between experimental results and calculations based on internal parameters of the individual devices.

Adaptive Threshold Logic Gate—*see* 5780.

Russian Digital Computer Using Ferrite Core-Diode Elements—*see* 5797.

4) DIGITAL STORAGE AND INPUT-OUTPUT EQUIPMENT

5800
A Simple Probability Model Yielding Performance Bounds for Modular Memory Systems, E. G. Coffman, Jr. (Princeton U.); *IEEE Trans. Computers*, vol. C-17, pp. 86–89, January 1968.

A simple probability model is defined that represents modular memory systems under saturation demand for storage. An analysis of the model based on the theory of finite Markov chains leads to results that demonstrate the effects of changes in loading times and the number of modules on system performance.

Iterative Array of Sequential Machines Similar to Distributed Logic Associative Memory—*see* 5795.

Magnetic-Tape Store for a Russian Computer—*see* 5797.

5801
Method for Estimation and Optimization of Printer Speed Based on Character Usage Statistics, E. B. Eichelberger, W. C. Rodgers, and E. W. Stacy; *IBM J. R&D*, vol. 12, pp. 130–139, March 1968.

Many high-speed printers now in the field and under development use a constantly moving chain or train containing the characters required in the printing process. Generally they skip to the next line whenever the last character on a given line is printed. Since individual character usage varies widely, it may be possible to increase the printing speed by repeating high-usage characters more frequently on the chain than low-usage ones. This paper presents an analytic method of accurately estimating the printing speed of a chain printer for any character arrangement and describes a technique for determining the number of copies each character should have on the chain so that the printer will operate at or near maximum speed. Using these methods, significant increases in printing speeds have been obtained for actual applications.

Standard for Alphanumeric Keyboard Arrangement for Information Interchange—*see* 5773.

Standard for Dual Case Keyboard Arrangement—*see* 5774.

Input/Output Units for a Russian Computer—*see* 5797.

FORTRAN IV Free Field Input/Output Subroutine Package for the IBM System/360 Operating System—*see* 5811.

5802
An Analog Comparator as a Pseudo-Light Pen for Computer Displays, K. H. Konkle (M.I.T. Lincoln Lab., Lexington); *IEEE Trans. Computers*, vol. C-17, pp. 54–55, January 1968.

An Analog Comparator Interrupt Device signals the computer central processor whenever the computer display unit deflection voltages cause the CRT beam to fall within a small square surrounding a previously specified display coordinate. This provides the display with the sensing capability of a light pen that is necessary if a RAND tablet or other position input device is to be used to point at display picture elements in graphics applications. The X and Y deflection voltages corresponding to the cursor or pointer are stored and then compared with subsequent X and Y input voltages. An interrupt output is produced when a match occurs within certain limits. Operational amplifiers and field-effect transistor switches are connected in a simple feedback loop to provide storage and comparison. A diode net and threshold detector produce the output signal. This unit is now being used with the Lincoln Laboratory TX-2 computer curve-drawing display unit.

5803
Steering Circuitry for an Electroluminescent Panel, T. Katoh (Illinois U., Urbana); May 1967, 51 pp., Rept. 236; *U. S. Gov't R&D Repts.*, vol. 68, p. 95(A), February 10, 1968. AD 662 649 CFSTI $3.00.

In many cases it is desirable to display the output of information processors by graphical means. One of the top contenders in the display area is the electroluminescent panel in which matrix steering of all display points is possible. Up to now the highly nonlinear and time-variable characteristics of electroluminescent panels have led to considerable difficulties in building practical display systems. The system described in this thesis is the outcome of the theoretical investigations concerning the mechanism of electroluminescence and uses a very high voltage pulse superposition method. It is felt that the uniformity of a 28×28 panel is sufficient and that the access circuitry is of not more than average complication.

Graphic Software and Display Programs—*see* 5810.

5) PROGRAMMING AND CODING OF DIGITAL MACHINES

5804
Automatic Detection of Parallelism in Computer Programs, H. W. Bingham, D. A. Fisher, and E. W. Reigel (Burroughs, Paoli); November 1967, 90 pp., Rept. TR-67-4; *U. S. Gov't R&D Repts.*, vol. 68, p. 93(A), February 10, 1968. AD 662 274 CFSTI $3.00.

The object of the first year of this research has been to establish the desirability and feasibility of automatically recognizing parallelism in computer programs written in compiler languages, given suitable machine organizations capable of parallel or concurrent execution. An algorithm for formal analysis of programs is described. This algorithm detects implicit parallelism that exists between parts of a given sequentially ordered program from the input–output set intersections and any initially known essential serial order. The algorithm can be applied to program structures including subroutines, loops, conditionals, recursive subroutines, arrays, and serial input–output calls. To demonstrate that the automatic recognition of parallelism is feasible, a simulation based on the algorithm has been programmed on the Burroughs B5500 for analysis of compiler language programs. The source language for the programs to be analyzed (a subset of ALGOL), a description of the simulation, example programs for analysis, and the results of parallelism detection are given. Desirability is dependent on having machine organizations capable of using the parallelism effectively. The relation between parallelism in programs and the machine organizations will be investigated in future reports.

5805
Automatic Monitoring of the Correct Recording of Algorithms in the ALGOL-60 Language, V. A. Vasilev and N. N. Lozinskii; June 1967, 33 pp., Rept. FTD-MT-67-78; *U. S. Gov't R&D Repts.*, vol. 68, p. 72(A), January 25, 1968. AD 661 773 CFSTI $3.00.

A semantic method for checking the accuracy of ALGOL algebraic problems is proposed. The content and organization of the semantic program are discussed as well as various additional problems associated with freeing the information from errors. The proposed method verifies the program with respect to the following points: 1) the rules established for description of the programs should be observed, 2) the quantities appearing in the program should be used in positions corresponding to their "nature," and 3) the actual parameters of the procedure operator and the formal parameters for description of this procedure should correspond to one another in the sense that the procedure field is modified according to the

rules for syntactic and semantic sense. That is, these three points should be fulfilled in the operator. A general program is described for carrying out this checking method. This verification system is self-contained with respect to the translator and may be used on machines with less complex coding. Some of the general limitations of the system are pointed out.

5806
Electronic Computer Program for Renumbering of FORTRAN Statement Numbers, M. Marques (Bureau of Public Roads, Washington, D. C.); October 1967, 44 pp., BPR Program M-2; *U. S. Gov't R&D Repts.*, vol. 68, p. 74(A), January 25, 1968. PB 176 538 CFSTI $3.00.

The function of this program is to replace old statement numbers in an original FORTRAN source deck with new sequential statement numbers. All statement numbers appearing at the beginning, middle, or end of FORTRAN statements are converted to a new system. The result is a new FORTRAN listing and a reproduced source deck containing the new sequential statement numbers. This report contains two listings of the Renumbering Program. One listing shows the statement numbers in an irregular pattern of sequencing while the other listing shows the statement numbers in a sequential pattern after the source deck was processed with the Renumbering Program.

5807
A Method of Minimizing Microprograms, T. F. Slobodyanyuk; July 1967, 17 pp., Rept. FTD-HT-23-600-67; *U. S. Gov't R&D Repts.*, vol. 68, p. 60(A), February 25, 1968. AD 662 821 CFSTI $3.00.

The author proposes an algorithm for minimizing individual edges, i.e., for reducing the number of microcommands which correspond to individual edges. This reduction is achieved by adding some type of coupling in the operational registers, and introducing new registers or new signals, which enables the device to use less cycles to execute an operation, which, in turn, reduces the microprogram of the operation. The construction of an algorithm is described in detail. The algorithm serves to minimize the following subprograms: 1) evolution, 2) conversion from a decimal system of notation to a binary system, and 3) conversion from a binary system to a decimal system. The results obtained during the minimization of microprograms using the algorithm discussed are tabulated. An analysis of the results shows that the application of the algorithms a) reduced the time for the execution of the microprogram by 6 to 12 percent, b) reduced the dimensions of the microprogram by 23 to 39 percent, and c) for some microprograms it reduced the amount of equipment required in the operational section.

5808
Microprogram Control for System/360, S. G. Tucker; *IBM Sys. J.*, vol. 6, no. 4, pp. 222–241, 1967.

This paper describes the kind of microprogram control that has been used in several models of System/360. A microprogramming language, as well as some of the main techniques used in "assembling" and testing microprograms, are discussed. Applications of microprogram control to the design of emulators, to compatibility features, and to special modifications are summarized.

5809
A Combination Hardware–Software Debugging System, K. C. Knowlton (Bell Tel. Labs., Murray Hill); *IEEE Trans. Computers*, vol. C-17, pp. 84–86, January 1968.

A scheme is proposed for automatically detecting many programming errors; in particular, those errors which can cause a program to misbehave in different ways, depending upon how the faulty program and its data are mapped into storage. Error detection is accomplished by simultaneously running two versions of a program which purport to be logically identical, with appropriate hardware checking between them.

5810
Special Utility Programs to Enhance the Performance of an On-Line Medium Size Processor Used for Statistical Information Extraction and Evaluation, R. J. Arsenault, V. M. Flynn, L. B. Metrick, N. V. Pitblado, and J. R. Wright (Wolf R&D, West Concord); September 1967, 76 pp., AFCRL-67-0605; *U. S. Gov't R&D Repts.*, vol. 68, p. 61(A), February 25, 1968. AD 662 872 CFSTI $3.00.

A new method for determining eigenvectors and eigenvalues in the Attribute Extraction Process is described. A software package for the new floating point arithmetic unit of the Experimental Dynamic Processor, a visual documentation system for DEXTER programs, and a general data display are described. Several debugging programs, graphic software, special display programs, and utility routines are presented.

5811
360 O. S. FORTRAN IV Free Field Input/Output Subroutine Package, R. W. Doran (Stanford U.); October 1967, 22 pp., Rept. CS-79; *U. S. Gov't R&D Repts.*, vol. 68, p. 61(A), February 25, 1968. AD 662 884 CSFTI $3.00.

Programmers dealing with aspects of natural language processing have a difficult task in choosing a computer language which enables them to program easily, produce efficient code, and accept as data freely written sentences with words of arbitrary length. List processing languages such as LISP are reasonably easy to program in but do not execute very quickly. Other, formula oriented, languages like FORTRAN are not provided with free-field input. The Computational Linguistics group at Stanford University Computer Science Department is writing a system for testing transformational grammars. As these grammars are generally large and complicated it is important to make the system as efficient as possible; therefore, the group is using FORTRAN IV (O. S. on IBM 360-65) as the language. The group has developed a subroutine package which is described in the hope that it will be useful to others embarking on natural language tasks. The package consists of two main programs, free-field reader, free-field writer, with a number of utility routines and constant COMMON blocks.

5812
A Language for Real-Time Systems, BCS Specialist Group—On-Line Computers and Their Languages; *Computer Bull.*, vol. 11, pp. 202–212, December 1967.

This paper considers particular problems involved in the programming of computers which are to operate in a real-time environment. Detailed recommendations for extensions and some alterations to an existing high-level scientific programming language are given. These changes and additions should provide the essential facilities required in real-time systems, and also include other features, not specifically related to real-time systems, which are considered to be generally desirable and useful. The report also discusses more general problems of real-time system operation which are not covered by the formal language structure but for which a real-time language compiling system must make provision. It includes, in outline, a description of one method of implementation which, although in no sense necessarily the best method, indicates that the generation of suitable compilers will be practicable.

User-Oriented Languages for Digital Simulation of Continuous Systems—*see* 5833.

Heuristic LISP Programs for Symbolic Integration—*see* 5826.

Use of the PL/1 Language in the OPS-4 Simulation System—*see* 5834.

5813
Application of Finite Geometry in File Organization for Records with Multiple-Valued Attributes, S. P. Ghosh and C. T. Abraham; *IBM J. R&D*, vol. 12, pp. 180–187, March 1968.

The schemes for organizing binary-valued records using finite geometries have been extended to the situation in which the attributes of the records can take multiple values. Some new schemes for organizing records have been proposed which are based on deleted finite geometries. These new schemes permit the organization of records into buckets in such a manner that, by solving certain algebraic linear equations over a finite field, it is possible to determine the bucket in which records, pertaining to two given values of two different attributes, are stored. Since the bucket identification required for the storage of record accession numbers is based on the combination of attribute values, the file does not require any reorganization as new records are added. This is a definite advantage of the proposed schemes over many key-address transformation procedures wherein the addition of new records may lead to either a drastic revision of the file organization or significant reduction of retrieval effectiveness. The search time for the new schemes are very small in comparison to other existing methods.

5814
JOSS: Central Processing Routines, J. W. Smith (RAND, Santa Monica); August

1967, 188 pp., Rept. RM-5270-PR; *U. S. Gov't R&D Repts.*, vol. 68, p. 71(A), January 25, 1968. AD 661 539 CFSTI $3.00.

This is a reference guide for JOSS users covering 1) the language used for couching instructions to JOSS, 2) JOSS's responses to instructions, 3) the collection of machine-language routines (in JOSS's central computer) responsible for interpreting and responding to instructions, and 4) the details and decisions that bilaterally influenced the language and the design and implementation of the routines. The myriad details of total system design are given constant exposure, and particular emphasis is placed on the delicate balance and symbiosis that must exist among system, language, computer, and routines and on the pervasive effects of each component on the others. The material is presented in a narrative form, augmented by flow-chart representations of most of the principal routines, and is in part designed to serve as prolegomena to the annotated machine-language listings of the routines (copies of which are obtainable from RAND).

6) HUMAN COMMUNICATION, DOCUMENTATION, AND HUMANITIES

5815

A Note on Backus Naur Form, J. S. Rohl (U. of Manchester, England); *Computer J.*, vol. 10, pp. 336–337, February 1968.

Backus Naur Form has gained acceptance as a language for formally describing the syntax of programming languages. However, BNF suffers from the disadvantage that it cannot describe itself. This is because the metalinguistic formulas involve special metalinguistic symbols. BNF is being accepted as an input by some compilers and there is a strong case for embedding the special-purpose facilities in a common language rather than producing special-purpose languages. Thus it can be expected that it might be useful to allow some facilities of BNF to be added to ALGOL. (They have already been added to Atlas Autocode). As it stands ALGOL with BNF facilities would be capable of describing a host of languages including normal ALGOL but excluding itself. There are at least two solutions. One, adopted in the Compiler Compiler, is to create special pseudo-metalinguistic variables which implicitly describe the special symbols. The second solution is to use brackets to surround not the metalinguistic variables but the basic symbols. Both these modifications are described.

Algebraic Automata and Context-Free Language—*see* 5786.

Subroutine Package for Computational Linguistics—*see* 5811.

5816

Computer Evaluation of Indexing and Text Processing, G. Salton (Cornell U., Ithaca) and M. E. Lesk (Harvard U., Cambridge); *J. ACM*, vol. 15, pp. 8–36, January 1968.

The SMART document retrieval system, which has been operating on an IBM 7094 for over three years, has been used extensively to test a large variety of automatic retrieval procedures, including fully automatic information analysis methods, automatic procedures for dictionary construction, and iterative search techniques based on user interaction with the system. The present study summarizes the results obtained with the SMART system over a two year period starting in 1964, and presents evaluation output based on the processing of three document collections in three different subject fields. Conclusions are drawn concerning the most likely analysis methods to be implemented in an operation environment. The emphasis throughout is on test analysis procedures, since they form an important part of a document handling system. The basic features of the SMART system are first described, and the design of the main experiments is outlined, including the statistical procedures used to test the significance of the evaluation output obtained. The principal evaluation results are then presented, and tentative conclusions are reached concerning the effectiveness of automatic test analysis procedures as part of future information systems. The results derived from the present experiments are also briefly compared with the output obtained with several other testing systems.

7) BEHAVIORAL SCIENCE, PATTERN RECOGNITION, AND ARTIFICIAL INTELLIGENCE

Future of Pattern Recognition—*see* 5771.

5817

A Finite-Memory Adaptive Pattern Recognizer, K. B. Irani (U. of Michigan, Ann Arbor); *IEEE Trans. Systems Science and Cybernetics*, vol. SSC-4, pp. 2–11, March 1968.

This paper gives an adaptive procedure for selecting a discriminant for a pattern recognizer. The optimum discriminant is selected from a given finite set of discriminants. The selection of this set itself is not considered here. At any stage the optimum discriminant is selected on the basis of the past observations. Since the storage space for these observations is assumed to be limited, and hence the qualifier *finite memory*, the information stored about these past observations is judiciously selected. No other a priori knowledge is assumed. A mathematical model of the problem of pattern recognition is constructed and several theorems are proved. With the help of these theorems, the adaptive procedure is developed. This adaptive procedure is, in effect, a method of using the finite memory efficiently in "training" the pattern recognizer.

5818

Learning Games Through Pattern Recognition, E. B. Koffman (Dept. of Defense, Washington, D. C.); *IEEE Trans. Systems Science and Cybernetics*, vol. SSC-4, pp. 12–16, March 1968.

The objective of this research was to investigate a technique for machine learning that would be useful in solving problems involving forcing states. In games or control problems a forcing state is one from which the final goal can always be reached, regardless of what disturbances may arise. A program that learned forcing states in a class of games (in a game-independent format) by working backwards from a previous loss has been written. The class of positions that ultimately results in the opponent's win is learned by the program (using a specially designed description language) and stored in its memory together with the correct move to be made when this pattern reoccurs. These patterns are searched for during future plays of the game. If they are formed by the opponent, the learning program blocks them before the opponent's win sequence can begin. If it forms the patterns first, the learning program initiates the win sequence. The class of games for which the program is effective includes Qubic, Go-Moku, Hex, and the Shannon network games, including Bridge-it. The description language enables the learning program to generalize from one example of a forcing state to all other configurations that are strategically equivalent.

5819

A Feature-Detection Program for Patterns with Overlapping Cells, W. M. Rintala (Philco-Ford, Palo Alto) and C. C. Hsu (U. of Washington, Seattle); *IEEE Trans. Systems Science and Cybernetics*, vol. SSC-4, pp. 16–23, March 1968.

An attempt is made to extract feature informations automatically from patterns which may consist of open lines, partially overlapping cells, and cells that may lie entirely inside another cell. The usual pattern-recognition techniques, such as the linear threshold logic technique and the masking or template technique, are not practical here, if not entirely impossible. In this paper, a direct-search computer program using a heuristic approach is described. A test pattern is used to illustrate the capability of the program. The subject should be of general interest to those in the field of automation and cybernetics.

Adaptive Threshold Logic and Pattern Recognition—*see* 5785.

Standard Character Set for Optical Character Recognition—*see* 5774.

5820

Mathematical Studies for Self-Organizing Systems, M. Rubinoff, A. Eliasoff, and G. Ingargiola (Pennsylvania U., Philadelphia); November 1967, 134 pp., Rept. 68-09; *U. S. Gov't R&D Repts.*, vol. 68, p. 63(A), February 25, 1968. AD 663 239 CFSTI $3.00.

The final report describes progress made in studies of mathematical models for self-organizing systems. Two models have received detailed investigation. The first is an intelligent machine with specific application to logic design of switching systems, and the second is a graph model of Turing machines and the data on which they operate. The latter model has been successfully applied to the description of a self-producing universal Turing machine. The latter has been programmed on the 7040 and a simulation has been run successfully.

Efficiency of Learning for Stochastic Automata—*see* 5790.

Extensions to the Heuristic Algorithm for University Timetables—*see* 5776.

8) MATHEMATICS

5821
A Computational Method for Evaluating Generalized Inverses, R. P. Tewarson (State U. of New York, Stony Brook); *Computer J.*, vol. 10, pp. 411–413, February 1968.

A method for computing the generalized inverse of a matrix is described, which makes use of the well-known Gauss–Jordan elimination method and the orthogonal triangularizations.

5822
Numerical Solution of the Inverse Algebraic Eigenvalue Problem, Z. Bohte (U. of Surrey, London); *Computer J.*, vol. 10, pp. 385–388, February 1968.

The inverse algebraic eigenvalue problem is formulated and a numerical method for its solution explained. The convergence of the method is considered and a numerical example given.

5823
Computer Algorithm for Spectral Factorization of Rational Matrices, W. G. Tuel, Jr.; *IBM J. R&D*, vol. 12, pp. 163–170, March 1968.

An algorithm is derived for the numerical spectral factorization of matrices arising in optimal filter design. The method uses a bilinear transformation to convert the factorization problem into a stable nonlinear difference equation of the Riccati type. The computer solution of several examples is presented to illustrate the technique.

5824
The Convergence of Fourier–Bessel Expansions, S. H. Storey (U. of Liverpool, England); *Computer J.*, vol. 10, pp. 402–405, February 1968.

A technique is described which allows Lanczos technique for increasing the convergence of Fourier expansions to be extended to apply to expansions in terms of Bessel functions. The effectiveness of the technique is examined by means of worked examples.

5825
The Numerical Evaluation of Principal Value Integrals, L. M. Delves (U. of Sussex, England); *Computer J.*, vol. 10, pp. 389–391, February 1968.

This paper considers the numerical evaluation of the principal value of the improper integral $P\int_a^b (f(t)/t-q)dt$ over a region $[a, b]$ which may include q. Methods of Newton–Cotes and modified Gaussian type are considered. Some of these require the evaluation of $f(q)$, while others do not. The methods are illustrated by some numerical examples.

5826
Symbolic Integration, J. Moses (M.I.T., Cambridge); December 1967, 271 pp., Rept. MAC-TR-47; *U. S. Gov't R&D Repts.*, vol. 68, p. 95(A), February 10, 1968. AD 662 666 CFSTI $3.00.

SIN and SOLDIER are heuristic programs written in LISP which solve symbolic integration problems. SIN (symbolic integrator) solves indefinite integration problems at the difficulty approaching those in the larger integral tables. SIN contains several more methods than are used in the previous symbolic integration program SAINT, and solves most of the problems attempted by SAINT in less than one second. SOLDIER (solution of ordinary differential equations routine) solves first-order, first-degree, ordinary differential equations at the level of a good college sophomore and at an average of about five seconds per problem attempted. The differences in philosophy and operation between SAINT and SIN are described, and suggestions are made for extending this work.

5827
Note on Two Methods of Solving Ordinary Linear Differential Eauations, M. R. Osborne and G. A. Watson (Australian Nat'l U., Canberra); *Computer J.*, vol. 10, pp. 383–384, February 1968.

An important class of methods for finding global solutions to ordinary linear differential equations involves assuming a trial solution containing free parameters, and determining these by some strategy. Attention has recently focussed on two methods of this kind: a) methods equating coefficients of the independent variable, and b) collocation methods, where the residual is made equal to zero at certain values of the independent variable. In this note two assumptions based on recent results regarding these methods are considered. They are shown to be false by means of counter examples.

5828
The Acceleration of the Peaceman–Rachford Method by Chebyshev Polynomials, A. R. Gourlay (U. of Dundee, Scotland); *Computer J.*, vol. 10, pp. 378–382, February 1968.

Considerable interest centers at the present time on the application of alternating direction implicit (ADI) procedures to the numerical solution of elliptic systems of equations. Under model problem conditions such methods have very rapid convergence rates. It is the purpose of this note to show how such convergence rates may be improved on by building into the ADI method an acceleration procedure based on the use of Chebyshev polynomials. The Chebyshev semi-iterative procedure is summarized and the ADI process is defined. This new process is applied to an ADI procedure with a constant acceleration parameter and is generalized to the important case when the ADI procedure has a cycle of acceleration parameters. The process is also applied to solving Laplace's equation by two well-known methods.

5829
A Note on the Estimation of the Optimum Successive Overrelaxation Parameter for Laplace's Equation, T. J. Randall (J. Dalton College of Tech., Manchester, England); *Computer J.*, vol. 10, pp. 400–401, February 1968.

The optimum parameter for any given region is estimated by finding an equivalent rectangle intuitively. Experimental evidence is produced in support of the method for two dimensions and for three dimensions with axial symmetry.

Application of Graph Theory to the Scaling of Fixed Point Digital Differential Analyzers—*see* 5798.

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

Sequential Probability—*see* 5789.

5830
Variance Algorithm for Minimization, W. C. Davidon (U. of Arhus, Denmark); *Computer J.*, vol. 10, pp. 406–413, February 1968.

An algorithm is presented for minimizing real valued differentiable functions on an N-dimensional manifold. In each iteration, the value of the function and its gradient are computed just once, and used to form new estimates for the location of the minimum and the variance matrix (i.e., the inverse of the matrix of second derivatives). A proof is given for convergence within N-iterations to the exact minimum and variance matrix for quadratic functions. Whether or not the function is quadratic, each iteration begins at the point where the function has the least of all past computed values.

Markov Chain Model of Modular Memory Systems—*see* 5800.

Statistical Information Extraction and Evaluation—*see* 5810.

5831
Spanning Tree Manipulation and the Traveling Salesman Problem, A. K. Obruca (U. of Strathclyde, Glasgow); *Computer J.*, vol. 10, pp. 374–377, February 1968.

The reasonable assumption is made that the majority of lines appearing in a minimal spanning tree for any network also appear in an optimal solution to the corresponding traveling salesman problem. A technique is described of manipulating the tree, by means of deletions and additions of lines, into a chain and hence obtain a feasible solution. An extension is considered with regard to the minimal wiring problem.

Optimizing the Sites and Heights of Transmission Line Towers by Dynamic Programming—*see* 5836.

5832
Digital Computer Simulation: The Allocation of Computer Time in Comparing Simulation Experiments, G. S. Fishman (RAND, Santa Monica); October 1967, 31 pp., Rept. RM-5288-1-PR; *U. S. Gov't R&D Repts.*, vol. 68, p. 95(A), February 10, 1968. AD 662 592 CFSTI $3.00.

The report describes an improved step-by-step procedure for minimizing the computer time needed to obtain a specified statistical precision in the comparison of simulation experiments. The simulations are

considered as covariance stationary stochastic processes. Well-known time-saving methods for reducing variance by inducing positive correlations between replications are included in the procedure. Results show that for a given level of accuracy, significantly less computer time is required when sample sizes are determined by the method suggested in this study than when they are equal. Also, small differences in the autocorrelation functions are important when each process is highly correlated.

5833
Two Continuous System Modeling Programs, R. D. Brennan and M. Y. Silberberg; *IBM Sys. J.*, vol. 6, no. 4, pp. 242–266, 1967.

The motivation, history, and basic concepts of user-oriented languages for digital simulation of continuous systems are presented. Reference is made to two illustrative programs, the IBM 1130 and System/360 Continuous System Modeling Programs (CSMP). Both programs accept user-oriented input statements for constructing simulation models and controlling simulation runs. The 1130 CSMP also allows online interaction by the user. An engineer or scientist at the console can alter the model or change run conditions based on direct observation of simulation outputs. The System/360 CSMP is intended for batch-mode operation. It has extended facilities for describing the model and and obtaining automatic program control of successive simulation runs.

5834
Incremental Simulation on a Time-Shared Computer, M. M. Jones (M.I.T., Cambridge); 1967, 253 pp., Rept. MAC-TR-48; *U. S. Gov't R&D Repts.*, vol. 68, p. 93(A), February 10, 1968. AD 662 225 CFSTI $3.00.

The thesis describes a system which allows simulation models to be built and tested incrementally. It is called OPS-4 and is specifically designed to operate in the environment of the MULTICS system. It represents a major expansion and improvement of the OPS-3 system implemented in CTSS and also includes many features adapted from other current simulation systems. The PL/1 language, augmented by many additional statements and new data objects, provides the basis for defining models in OPS-4. A list of desirable features for an incremental simulation system is presented and it is shown how OPS-4 incorporates these features, whereas other current simulation systems satisfy only some of them and are not suitable for use in time-shared environment. A simplified model of page and segment fault handling in MULTICS illustrates some of the features OPS-4 provides to allow the user to continuously interact with a model during its construction, testing and running phases. It also illustrates how the user himself may portray portions of a model that are not yet defined.

5835
Conventions for Digital Data Communication Link Design, J. L. Eisenbies; *IBM Sys. J.*, vol. 6, no. 4, pp. 267–302, 1967.

A definitive set of conventions has been established for the automatic, synchronous transmission of digital data over half-duplex (nonsimultaneous) communication links. Provision has been made for communication between different device types and between computer processing units. Although one transmission code must be used on a given data communication link, a special feature permits digital data in any form to be transmitted, including encrypted data and complied computer programs. This paper describes the Binary Synchronous Communication (BSC) conventions, which prescribe the encoding of data, the procedures for synchronizing stations, the methods for controlling the data links, transmission message formats, and error detection and correction methods. The presentation is sufficiently detailed to indicate the kinds of design decisions that are involved in setting up automatic data communication links.

10) SCIENCE, ENGINEERING, AND MEDICINE

5836
A Computer Technique for Optimizing the Sites and Heights of Transmission Line Towers—A Dynamic Programming Approach, G. Mitra and K. Wolfenden (U. of London); *Computer J.*, vol. 10, pp. 347–351, February 1968.

Given the survey data of a transmission line route and the choice of available towers of suspension type and of angle towers, a dynamic programming algorithm is described which chooses and sites the towers (the location and angle of the angle towers being prescribed) in such a way that the overall cost of running the line from one end of the route to the other, subject to all the established design constraints, is a minimum. The EMA program has been successfully run at the University of London Atlas Computer using exacting test data supplied by C.E.G.B.

Numerical Spectral Factorization of Optimal Filter Matrices—*see* 5823.

Analog Computer Simulation of Wave Filters Having Polynomial Transfer Functions—*see* 5837.

11) ANALOG AND HYBRID COMPUTERS

5837
The Simulation of Wave Filters Having Polynomial Transfer Functions on an Analogue Computer, K. G. Beauchamp (College of Aeronautics, Cranfield, England); *Computer J.*, vol. 10, pp. 352–359, February 1968.

The simulation of filter characteristics is often required to form part of a signal processing operation carried out on an analog computer. By adopting a simplified approach to the problem of direct mechanization of the polynomial transfer function involved, many of the required filter characteristics can be realized by the use of standard analog elements. Groups of potentiometer settings can be calculated and made available in tabular form to cover a wide range of requirements. Matching of desired filter characteristics with analog circuit configuration and gain/potentiometer setting may be simplified by reference to these tables.

5838
The Simulation of Variable Delay, J. P. Seddon (IBM, Winnipeg) and R. A. Johnson (U. of Manitoba, Winnipeg); *IEEE Trans. Computers*, vol. C-17, pp. 89–94, January 1968.

A dual classification of variable delays 1) by their explicit dependence on time or through a state variable and 2) through their origins as "pipe" or "nonpipe" problems is presented. Techniques of simulation through standard analog components and, more particularly, transport devices are analyzed and further classification made of the latter to distinguish those suitable for state-variable delay systems from those that are not. The construction and testing of a simple variable delay are described. An application to a system in which the delay is a sinusoidal function of time is discussed in detail. A theory is presented that predicts what frequencies will be present in the output of such a device when excited by a sinusoid of commensurate frequency; experimental results corroborating this are presented. A scheme for converting a problem with variable delay to one with fixed delay is further analyzed and its range of application delimited. Conclusions include a discussion of the best type of simulator to be used for variable delay problems.

SUBJECT INDEX

AUTHOR INDEX