

Listed are abstracts from recent papers by IBM authors. Inquiries should be directed to the publications cited.

An Analysis of High-Speed, Linear-Passive Binary, Read-Only Stores, R. L. Gamblin, *IEEE Transactions on Computers* C-17, No. 9, 889-893 (September 1968). A general analysis of conventional linear-passive, high-speed, binary, read-only stores is performed by the application of the scattering matrix formalism. It is found that for a broad class of cases, which include most read-only stores that have been discussed in the literature, the maximum transmission of power from a word driver to a sense amplifier is one divided by the number of bits stored in the memory. Other types of stores are considered that do not suffer from such a severe power degradation, but these latter appear to be pattern sensitive.

Abstracts

Combinatorial Information Retrieval Systems for Files, D. K. Ray-Chaudhuri, *SIAM Journal on Applied Mathematics* 16, No. 5, 973-992 (September 1968). This paper develops a theory of combinatorial information retrieval systems for file organization. Geometric and algebraic methods are employed to construct some combinatorial configurations. These configurations are used for constructing combinatorial filing systems—for files with n binary-valued attributes. These systems use some redundancy in storage and allow for efficient retrieval of records relevant to a query involving t or fewer attributes ($t < n$).

FORMAC Language and its Application to Celestial Mechanics, A. R. LeSchack and P. Sconzo, *Astronomical Journal* 73, No. 3, 217-220 (April, 1968). The IBM FORMAC language, a computer system for the symbolic manipulation of formulas and expressions, is described and several applications to problems of celestial mechanics are outlined. These applications include the derivation of power-series solutions to the equations of motion, the generation and application of high-order time derivatives of arbitrary powers of the radius vector, and the extension of Cayley's tables.

A Formula for Logical Network Cost, E. Kellerman, *IEEE Transactions on Computers* C-17, No. 9, 881-884 (September 1968). This paper presents an experimentally derived formula for the average cost of a one-output combinatorial logical network. The formula is a function of the number of independent variables, the number of "one" vertices, and the number of "zero" vertices of the Boolean function realized by the network.

A Generalized Partial Pass Block Sort, A. Bayes, *Communications of the ACM* 11, No. 7, 491-492 (July 1968). The design of a partial pass block sort with arbitrary range of key and number of work files is described. The design is a generalization of the Partial Pass Column Sort by Ashenburst and the Amphisbaenic Sort by Nagler. The power of the sort is tabulated for various sizes of input file and number of work files. Consideration is given to the problem of combining a block sort with internal sorts, and to the best use of direct access storage devices.

A General-Purpose Display Processing and Tutorial System, K. J. Engvold and J. L. Hughes, *Communications of the ACM* 11, No. 10, 697-702 (October 1968). ADEPT (A Display-Expedited Processing and Tutorial) system is described. This system was designed to improve man-computer communications by employing a display unit to interleave tutoring with other computer operations such as simulation, programming, and information retrieval. It is written in FORTRAN IV (G) for the IBM System/360, Model 40, and the IBM 2250 Display Unit under Operating System/360. ADEPT is a cataloged program that controls the standard operating system by terminating and rescheduling itself automatically, relinquishing computer resources allocated to it, and surrendering control to the operating system to perform other jobs. It expands the power and flexibility of computer-assisted instruction by making immediately available to students, teachers, and other users, the full resources (system-cataloged programs) of the operating system. Language processors and compilers, simulation models, mathematical solution techniques stored data, and all other library and user programs can be incorporated into instructional material without reprogramming. Illustrations of the various applications are presented and their implications are discussed.

Graphic Computer-assisted Design of Optical Filters, F. Gracer and R. A. Myers, *IBM Journal of Research and Development* 13, No. 2, 172-178 (March 1969). An interactive graphic system is described in which a designer using a light pen creates an optical filter design at an IBM 2250 Graphic Console controlled by an IBM 1130 computer. The designer can observe the reflectivity (or some other property) of the filter as a function of wave number plotted on the CRT. The application was developed using an experimental graphic version of the 1130 Continuous System Modeling Program (CSMP), a general-purpose, block-oriented simulation language in which the functional blocks represent the elements and organization of an analog computer. The designer has available a full set of operators for further analyzing the behavior or modifying the design of the filter. In addition to providing a highly flexible analytic tool, the system is intended to explore means for making the interactive computer terminal an important element in the inventive process.

A High-Speed Threshold Memory Element, R. L. Gamblin and C. J. Tunis, *IEEE Transactions on Computers* C-17, No. 9, 893-894 (September 1968). This note describes a new analog-memory or variable-weight device that can be set to any of 100 stable states and switched between states in 100 nanoseconds or less. The device uses the coupling between a circularly polarized microwave signal (as generated by a helical structure) and a partially switched ferrite core.

On the Independence of Real-Time Definability and Certain Structural Properties of Context-Free Languages, Arnold L. Rosenberg, *Journal of the Association for Computing Machinery* 15, No. 4, 672-679 (October 1968). An operator precedence language which is not real-time definable by any multiple Turing machine is constructed. This strengthens previous results about the existence of unambiguous context-free languages (CFL's) which are not real-time definable. In contrast, a family of CFL's of increasing degree of inherent ambiguity, each real-time definable by a one-tape Turing machine is exhibited. In fact, an unboundedly ambiguous CFL, also real-time definable by a one-tape Turing machine, is presented. These results are the basis for the assertion that real-time definability of a CFL is independent from each of the structural properties considered.

Memory Core Design and Drive Pulse Optimization using a Charge Model of the Storage Core and its Computer Implementation, H. H. El-Sherif and H. F. Koehler, *IEEE Transactions on Magnetics* **MAG-4**, No. 2, 193-196 (June 1968). This paper describes the representation of the switching characteristics of the ferrite memory core by flux-charge curves and the derivation of significant design or application parameters for a new, undeveloped ferrite memory core. The technique used is that of appropriately scaling flux and charge by a computer program. User options permit computation of signal and noise amplitudes and waveforms, read current and percent flux switched, core height and threshold, or outer and inner diameters, depending on the information available as input data. Effects of parameter tolerances can be evaluated and technology advances effecting device improvement can be indicated and accommodated.

Object Code Optimization, E. S. Lowry and C. W. Medlock, * *Communications of the ACM* **12**, No. 1, 13-22 (January 1969). Methods of analyzing the control flow and data flow of programs during compilation are applied to transforming the program to improve object time efficiency. Dominance relationships, indicating which statements are necessarily executed before others, are used to do global common expression elimination and loop identification. Implementation of these and other optimizations in OS/360 FORTRAN H are described.

*Sun Oil Company, Tulsa, Oklahoma.

A Procedure for Detecting Intersections of Three-Dimensional Objects, P. G. Comba, *Journal of the Association for Computing Machinery* **15**, No. 3, 354-366 (July 1968). As a step toward the solution of the placement problem in engineering design, a procedure has been developed for detecting intersections of convex regions in 3-space by means of a pseudocharacteristic function. The mathematical techniques underlying the procedure are discussed, and a system of programs embodying these techniques is described. As a special case a solution is given for the hidden-line problem in graphic display.

Regulation of Incompletely Identified Linear Systems, A. Chang and J. Rissanen, *SIAM Journal Control* **6**, No. 3, 327-348 (August 1968). This paper is concerned with regulation of linear systems disturbed by stationary Gaussian processes. Neither the noise nor system characteristic are assumed to be known a priori, but the input and output signals can be observed for the purpose of identification. The problem is to find a feedback law, a linear function of finitely many past observations, which minimizes an appropriate objective function measuring the goodness of the regulation. Precise conditions under which the problem can be solved with a stable control law are stated, and an algorithm for finding the solution with arbitrary accuracy is given.

Single-Channel Error Correction in a f-Channel System, M. Y. Hsiao, *IEEE Transactions on Computers* **C-17**, No. 10, 935-943 (October 1968). This paper discusses a method of correcting almost unlimited errors in a single channel of an f -channel system. The method uses a two-dimensional variable-length code (2-DVL). The encoding and decoding systems are implemented by a parallel linear-feedback shift register (LFSR). This type of error correction can be easily applied to data-recording devices such as tape, disk, or drum. The method places no restriction on the length or type (random or burst) of the error pattern. The method requires only that errors occur in a single channel among f channels within each block. This single-channel error-correction system provides five advantages: (1) The parallel operation allows fast data processing; (2) The system corrects unlimited errors in a single channel; (3) The encoder and decoder are highly economical, e.g., the decoder needs no buffer register to store the input code word for error correction; (4) Redundancy is extremely low; and (5) The system is very flexible—it can be applied to a variable-length block as well as to a fixed-length block.

Toward A General Processor for Programming Languages, M. I. Halpern, *Communications of the ACM* 11, No. 1, 15-25 (January 1968). Many efforts have been made to develop a better way of implementing a higher level programming language than by the construction of a whole new compiler, but so far none has proved generally satisfactory. In this paper, it is contended that a programming language is best described functionally as a body of macroinstructions, and that the macro call constitutes a canonical form in terms of which a programming notation may be described. A supporting discussion of the logical and historical role of the macroinstruction is presented. Also discussed are the conflict between machine independence and object program efficiency and the question of where the greatest difficulties lie in compiler construction.