REPOSITORY

technical papers
covering the full range
of computer system design

The Repository, a collection of over 2000 technical papers and documents relating to computer science and engineering, is maintained by the Computer Society as a service to the information processing community. Some of the papers have been refereed; others have not.

If you have a paper of interest to the computer field, you are invited to submit two copies, including abstract and index terms, to Dr. Warren L. Semon, IEEE Computer Society Editor-in-Chief, Division of Systems and Information Sciences, 313 Link Hall, Syracuse University, Syracuse, NY 13210. Be sure to include a cover letter giving permission to enter the paper in the Repository. Entry in the Repository does not constitute publication.

Photocopy prices are $1.00 per page. Add $1.00 service charge to all orders under 50 pages. Microfiche copies are available for $2.50 for each manuscript under 50 pages, plus $2.50 for each additional 30 pages or fraction thereof.

Be sure to state the R number, listed before the author's last name, of each paper you order. All Repository items must be prepaid except for companies or institutions with established accounts. A $2.00 invoice charge will be added to all non-prepaid orders. Please make your check or money order payable to the IEEE Computer Society.

R76-251—Liebowitz, Burt H., "Application Review" (44 pp., International Computing Company, Bethesda, Maryland)
The last five years have seen the emergence of new computer applications prompted primarily by the drop in computer hardware costs, and to a lesser degree by increased computing power. Several examples of new applications are presented. These include point-of-sale systems, small business computers, distributed processing systems, and systems for weather forecasting. It is concluded that the trend towards new applications will continue, particularly as low-end computing costs drop. The future should bring the computer into the home, the automobile, and into more business functions.

R76-252—Reynolds, Dennis A. and Garnot Metze, "Fault Detection Capabilities of Alternating Logic" (30 pp., Sandia Laboratories, Albuquerque, New Mexico)
This paper details the fault detection capability of a design technique named "alternating logic design." The technique achieves its fault detection capability by utilizing a redundancy in time instead of the conventional space redundancy and is based on the successive execution of a required function and its dual. In combinational networks the method involves the utilization of a self-dual function to represent the required function and the realization of the self-dual function in a network with structural properties which are sufficient to guarantee the detection of all single faults. One network structure (Yamamoto, Watanabe, and Urano) with sufficient structural properties is the standard AND/OR or OR/AND two-level network. However, other more general combinational logic structures also possess sufficient structural properties. Necessary and sufficient structural properties for any alternating network to be capable of detecting all single faults are derived.

R76-253—Korn, Granino A., "Analog/hybrid Computation and Interactive Digital Simulation" (28 pp., University of Arizona, Tucson, Arizona)
This survey article traces the development of analog/hybrid and digital continuous-system simulation since 1950, describes a modern interactive mini-computer system, and discusses future trends.

This thesis discusses the design, specification, and verification of computer operating systems. The operating system problem considered, the many-process problem, is the design of an operating system that can support a large number of concurrent processes. This design problem is a vehicle to investigate the use of a design methodology, the hierarchical levels of abstraction methodology; the use of structured programming techniques in the specification of the system; and the development of techniques for the verification of concurrent programs, particularly operating system programs. A solution to the many-process problem is obtained, and it is shown that the hierarchical levels of abstraction methodology simplifies the conception of the solution and helps avoid potential deadlocks in the system. A Pascal specification of the four levels of the system is given demonstrating the usefulness of structured programming techniques for specifying operating system programs. A detailed description of the development of the simple memory manager, a complex and large segment of the system, is given to show the use of step-by-step refinement for improving the efficiency of the program and as an aid in understanding its final specification. The specifications for the first two levels, simple scheduler and simple memory manager, are formally verified. The notion of exclusive access of a resource has been formalized and used in the verification of concurrent program. Sufficient conditions for verifying the absence of deadlocks in a system of mailboxes are also developed.

R76-255—Semsarzadeh, Gholam A., "Access Methods in Hierarchic Data Bases" (20 pp., Arya-Mehr University of Technology, Tehran, Iran)
While the notion of hierarchic relationships among group occurrences and the possibilities for their implementation are well understood, there has been no precise definition on how the group occurrences having such relationships should be accessed. The question of access in a data base is a central issue because the capability of a data base depends on the
power of the commands that a user may employ to specify the appropriate relationships to access the relevant information. This is especially important in the hierarchic (as opposed to the network) data bases which allow for unique methods of traversing the group occurrences in the hierarchy. This paper
gives a precise definition for accessing group occurrences in a hierarchic data base. Such a data base is described in terms of hierarchic relationships among a set of group occurrences without considering the physical organization of data.

In the view presented here, a file is regarded as a hierarchic data base where the hierarchy consists of the root only.


It is shown that the two-dimensional Fourier transform of a function bounded by a parallelogram can be approximated by one-dimensional discrete Fourier transforms of the rows and then the columns of the function sampled on the appropriate non-orthogonal Cartesian grid in much the same way as for the usual rectangular grid bounded by a rectangle.

R76-257—Paquet, Jean-Paul, "Secondary State Assignment with Assignment Covers" (44 pp., Université du Québec à Chicoutimi, Quebec, Canada)

Preserved covers on the set of internal states of synchronous sequential machines are used here to find secondary state assignments when D-type flip-flops are used as memory elements. A secondary state assignment is represented as a family of covers called assignment covers. A set of rules is established to determine if a given cover is an assignment cover. For that purpose, extensive use of the adjacency graph and the state map is made. The set of preserved covers of a given machine is generated with the help of the implication graph. Each preserved cover is tested to determine whether or not it is an assignment cover. Whenever a preserved cover is also an assignment cover, the secondary state assignment is determined and the multiple output cost of the flip-flop input equations is evaluated for each possible combination of column complementation. Numerous examples taken from the literature show that the preserved cover approach is often faster than other known methods of state assignment, and the costs that are obtained are comparable with the best published costs.

R76-258—Kulkarni, Ashok V., "On the Mean Accuracy of Hierarchical Classifiers" (16 pp., University of Maryland, College Park, Maryland)

The existence of an optimal measurement complexity for a given sample size used for training a two-class recognizer is shown for a multi-class non-hierarchic classification scheme. The optimum quantization increases with sample size and the number of classes to be distinguished.

R76-259—Rao, C. V. Kameswara and Kenneth Balck, "Finding the Core Point in a Fingerprint" (25 pp., Linkoping University, Linkoping, Sweden)

A procedure for finding the core point in a fingerprint is presented. The search for the core point is done in steps with increasing resolution, each step indicating for the next step in which subarea of the print to search for the core. This results in a sequence of diminishing areas where the core can be found. The present implementation uses three levels.

R76-260—Kwaternik, Ratinik, "An APL Terminal Approach to Computer Mapping" (189 pp., University of Toronto, Toronto, Canada)

This thesis demonstrates a remote terminal approach to computer mapping and suggests that the terminal should be utilized in future work in computer cartography. A package was developed in APL primarily to help landscape architects and urban planners in representing combinations of various environmental factors by maps. Factors such as rivers, forests, and slopes are represented in the computer as matrices called base maps. The X_i-th element of each matrix corresponds to the (i, j)-th point of a rectangular grid. If a factor is present at the (i, j)-th point, the corresponding element of the area is set to 1; otherwise, it is set to 0. Various logical and arithmetic operations are available to combine base maps into new composite maps. These show meaningful results that could not be easily produced by hand. The use of the package can be extended to other similar problems and fields wherever the information can be displayed in the form of a map.


The goal of Project SUE at the University of Toronto is to construct a reliable and understandable operating system based on ideas drawn from other systems and from operating systems literature. In this paper, we present a discussion of some of the choices and decisions that have been encountered during the design of the SUE system. Besides selecting an overall system structure, we have had to develop a compatible set of mechanisms for communication, control, protection, resource allocation, and accounting. We have attempted to evaluate the practical worth of some conceptual ideas, to test in a full-scale system some ideas proven useful in small systems, and to assess the mutual compatibility of some individually useful mechanisms.

R76-262—Schumacher, Helmut, "The Synthesis of Optimal Decision Trees from Decision Tables" (128 pp., University of Toronto, Toronto, Canada)

For years there has been a steady interest in establishing methods for automatically translating decision tables to programs of minimum average execution time or minimum storage requirement. We review a branch-and-bound method which has been, to our knowledge, the only published algorithm which guarantees the optimality of the derived decision tree. We discuss some of the heuristics proposed for finding near-optimal decision trees with respect to average execution time. All these methods generate the decision trees proceeding from the root to the leaves and thus considering subtrees of decreasing length. The method proposed synthesizes decision trees by considering optimal subtrees of increasing length, applying the dynamic programming principle. Whereas the branch-and-bound method is feasible only for decision tables with up to five conditions, the proposed method can convert tables with up to eight conditions within a time comparable to that of heuristic methods. If virtual memory is available, tables with up to twelve conditions can be converted. The proposed algorithm handles extended entries as well as limited entries. The method can be applied for minimizing the total storage requirement of condition tests and actions in a decision tree, taking advantage of hoisting actions in the tree. An implementation of the algorithm is presented in the appendices.

R76-263—Tischritzis, D., "LSL: A Link and Selector Language" (35 pp., University of Toronto, Toronto, Canada)

This paper presents the main ideas behind the language LSL. The purpose of LSL is to provide a compromise between different, data base approaches and attitudes. The hierarchical, network, and relational models of data can coexist within the LSL environment. LSL is implemented on a PDP-11/45 running the UNIX operating system. It is interfaced both to a host language and a conversational graphical facility.

R76-264—Barnard, David and David Thompson, "An Annotated Bibliography on Computer Program Engineering" (100 pp., University of Toronto, Toronto, Canada)

This bibliography is an outgrowth of a graduate course on computer program engineering and is an attempt to provide some guidance to the literature in this area. References annotated by students in the course are provided, along with a cross reference list by topic area. This report, CSRG-69, updates and supercedes CSRG-54.


A new set of partitions on the states of a sequential machine are defined in terms of transition behavior. This approach gives a unified set of partitions which are independent of the eventual flip-flop to be used and require only three generation rules. With the knowledge of the FF type, candidate partitions are selected from the transition set and scored.
The new partitions generate all useful members, and their relation to the $\pi$, $\rho$, $\tau$ partitions previously used is given.

R76-266 Papachristou, Chrisostos A., "An Algorithm for Optimal NAND Array Logic Synthesis" (44 pp., Drexel University, Philadelphia, Pennsylvania)

This paper is concerned with optimal synthesis of switching logic by a limited depth tree-like network, the NAND array. This array consists of a number of complete three-level, fan-in restricted, NAND trees feeding a NAND collector. The goal of the proposed synthesis is to minimize the number of NAND trees of the array, which in turn will minimize its overall depth, i.e., the delay time of the array. An algorithm is developed for optimal realization of an arbitrary switching function $f$ by the NAND array. This algorithm generates NAND tree formulas implying $f$ and distinguishes the formulas in primes and unprimes in the same sense prime implicants are distinguished among implicants of $f$. The algorithm generates all prime tree formulas of $f$ and then produces an irredundant cover. Experimental results for 5 and 6 variable functions are presented.

R76-267—Cordy, James R., "A Diagrammatic Approach to Programming Language Semantics" (84 pp., University of Toronto, Toronto, Canada)

This thesis describes a concise, easy to understand diagrammatic notation for programming language semantics. The notation utilizes semantic charts which provide a model for practical table driven semantic processing. This technique has been used successfully to implement a semantic phase for a production PL/I subset (SP/I) compiler. The scheme provides facilities for symbol table management, exception handling, control structure handling, as well as semantic error detection, diagnosis, and repair. Its output is a machine language program.

R76-268—Shadlethesky, John J., "A Rollback Interval for Networks with an Imperfect Self-Checking Property" (7 pp., Digital Systems Laboratory, Stanford University, Stanford, California)

Dynamic self-checking is a technique used in computers to detect a fault quickly. It is attractive to those computers. The recovery procedure may be required to roll back program execution to a point prior to the first undetected error caused by the detected fault. This paper presents a method by which the rollback distance required to achieve a given probability of successful data restoration may be calculated. To facilitate this method, operational interpretations are given to familiar network properties such as the self-testing, secureness, and self-checking properties.

An arithmetic and logic unit with imperfect self-checking capability is analyzed to determine the minimum required rollback distance for the recovery procedure.

R76-269—Luccio, Fabrizio and Linda Pagli, "Rebalancing Height Balanced Trees" (31 pp., University of Pisa, Pisa, Italy)

A new balancing technique for binary search trees is presented, based on the repositioning of $k$-1 nodes (k-rotation). Some properties of k-rotation are shown, and bounds to $k$ are derived. The performance of such a technique is discussed on the basis of the length of node search and the frequency of tree rebalancing.

R76-270—Brown, Richard D., "A Recursive Algorithm for Sequence-Ordered Fast Walsh Transforms" (13 pp., Computer Sciences Corporation, Silver Spring, Maryland)

An algorithm has been developed for calculating sequence-ordered fast Walsh-Fourier transforms (FWT) using an additive recursive rule. Sequence-ordered FWT's of an N-dimensional sampled data set are generated by a summation recursion of FWT's on subintervals of the date set. The algorithm is fast (N log $N$ summations), computer efficient, and can be applied to time dependent spectral analysis of nonstationary phenomena such as speech.


An algorithm for generating uniformly distributed random deviates with microprocessors or minicomputers utilizing memories with a small word length is presented. Other alternative solutions are discussed, and the statistical quality of sample streams of generated numbers is evaluated and compared with that of other widely used generators.

R76-272—Trivedi, Kishor S., "Preparing and Applications to the STAR-100 Computer" (29 pp., NASA Langley Research Center, Hampton, Virginia)

The use of prepping is described for the CDC STAR-100 system. A feature known as ADVISE is provided by the system for this purpose. A timing analysis of a matrix multiplication routine is carried out to evaluate the effect of prepping. Finally, a suggestion for a controlled prepping algorithm is made.

R76-273—Moore, Roger K., "Evaluating Speech Recognizers" (34 pp., University of Essex, Colchester, Essex, England)

Although automatic word recognition systems have existed for some twenty-five years there is still no suitable standard for evaluating their relative performances. Currently the merits of two systems cannot be meaningfully compared unless they have been tested with at least the same vocabulary or, preferably, with the same acoustic samples. This paper develops a standard for comparing the performance of different recognizers on arbitrary vocabularies based on a new word recognition model. This standard allows recognition results to be normalized for comparison according to two intuitively meaningful figures of merit: (1) the noise-level necessary to achieve comparable human performance and (2) the degree of confusion from human performance. Examples are given of recognizers evaluated in this way, and the role of these performance measures in automatic speech recognition and other related areas is discussed.

R76-274—Gorrie, J. D., "A Processor Generator System" (143 pp., University of Toronto, Toronto, Canada)

The implementation of a processor generator for student languages is presented. Included are schemes for the formal specification of a language by its syntax and semantics, for the automatic construction of a symbol table, and for the production of the driving arrays for a table-driven scanner, parser, and code emitter. In addition, a pseudo-machine designed to facilitate semantics specifications and to provide a high diagnostic capability is examined.

R76-275—Klebanoff, Jack, Fred Lochovsky, Arnis Rozitis, and Dennis Tschritzis, "Educational Data Base System User's Manual" (109 pp., University of Toronto, Toronto, Canada)

The Educational Data Base System (EDBS) is an interactive data base management system designed for educational use. The system implements both the functional and hierarchical views of data. EDBS is written in and accessed from APL PLUS. A student-user oriented description of the EDBS information structures and data manipulation language is presented. This is followed by a number of suggested exercises and games to be used with EDBS in a course.

R76-276—Wortman, David B., "Notes from a Workshop on the Attainment of Reliable Software" (51 pp., University of Toronto, Toronto, Canada)

In June 1974 a Workshop on the Attainment of Reliable Software, jointly sponsored by the ACM Special Interest Group on Programming Languages, the IEEE Computer Society Technical Committee on Fault-Tolerant Computing, and the University of Toronto Computer Systems Research Group, was held in Toronto. This report contains summaries of the discussion sessions held at the workshop.
R76-277—Tilson, Michael D., "Editing Computer Animated Film" (119 pp., University of Toronto, Toronto, Canada)

Principles involved in the production of computer animated film are discussed. Two paradigms of animation are proposed: "demonstrative" and "algorithmic." It is argued that algorithmic animation requires an interactive editing phase and that demonstrative animation can be implemented as an editing sub-system. Requirements for an interactive editing system are proposed, and an experimental implementation is described.

R76-278—Poppendieck, Mary and Edouard J. Desautels, "Memory Extension Techniques for Minicomputers" (14 pp., University of Wisconsin, Madison, Wisconsin)

The address range of minicomputers purchased two to three years ago was typically 32K words. Many owners of these computers are realizing that this is no longer adequate for their application. This paper discusses and compares the various methods of addressing main memory which are becoming available for minicomputers.

R76-279—Rau, B. Ramakrishna, "The Stack Working Set: A Characterization of Spatial Locality" (67 pp., Digital Systems Laboratory, Stanford University, Stanford, California)

Multilevel memory hierarchies are attractive from the point of view of cost-performance. However, they present greater problems than two-level hierarchies when it comes to analytic performance evaluation. This may be attributed to two factors: (1) the page size (or the unit of information transfer between two levels) varies with the level in the hierarchy and (2) the request streams that the lower (slower) levels see are the fault streams out of the immediately higher levels. Therefore, the request stream seen by each level is not necessarily the same as the one generated by the processor. Since the performance depends directly upon the properties of the request stream, this poses a problem. A model for program behavior, which explicitly characterizes the spatial locality of the programs, is proposed and validated. It is shown that the spatial locality of a program is an invariant of the hierarchy when characterized in this manner. This invariance is used to solve the first problem stated—that of the varying page sizes. An approximate technique is advanced for the characterization of the fault stream as a function of the request stream and the capacity of the level. A procedure is then outlined for evaluating the performance of a multilevel hierarchy analytically.


As part of the research effort investigating the properties and applications of Burst processing, the question of compatibility with the more usual weighted-binary data representations has been considered. The MICROBURST system is a microprocessor-controlled, general-purpose interface system capable of operating in a multichannel Burst environment. System structure, programming details, and support software are discussed.

R76-281—Bilodeau, Mark B. and G. Michael Schneider, "An Analytic Study of Message Delays in Distributed Computer Networks" (48 pp., University of Minnesota, Minneapolis, Minnesota)

This paper is an analytic investigation into the operation of a distributed, heterogeneous, resource-sharing, computer network. The network is decomposed into a series of single-server models. Each independent process in the network is modeled as a server, and each server is related to its neighbors by the network routing table. The object of this analysis is to investigate the relationship between network architecture, protocols, traffic pattern, and the response of the network as measured by mean logical message delay.

Hard-to-find specials from the Computer Society Repository


This report summarizes the research work which has been performed and is currently active in the Center for Reliable Computing in the Digital Systems Laboratory, Stanford University. Price, photocopy—$10.30


Technological advances have made possible the development of advanced hybrid computing systems (AHCS) with cost and speed advantages of at least 30:1 over pure digital systems for solving dynamic problems. This proceedings contains 27 papers examining various aspects of AHCS. Price, photocopy—$20.80

R76-179—Sunshine, Carl A., "Interprocess Communication Protocols for Computer Networks" (279 pp., Stanford University, Stanford, California)

This report focuses on the design and analysis of interprocess communication protocols for networks of computers. Previous research has emphasized system performance at lower levels, within the communication medium itself. This work examines requirements and performance of protocols for communication between processes in the host computers attached to the communication system. Both the reliability and the efficiency of protocols are discussed. Reliability involves overcoming unreliable network transmission facilities to avoid loss, duplication, or out-of-order delivery of data. Reliability performance goals are defined, and the correctness of different protocol mechanisms in achieving these goals is demonstrated. Consequences of protocol failures (Host crashes) and problems of initializing control mechanisms required for reliable communication are also considered. Efficiency primarily concerns throughput and delay achievable for communication between remote processes. The performance of successively more powerful protocols including error detection, retransmission, flow control, limited buffering, and sequencing is analyzed. Protocol parameters such as retransmission interval, window size, buffer allocation, packet size, and acknowledgement strategy emerge as important factors in determining efficiency. Several graphs showing quantitative performance results for representative situations are included. An additional section of the report considers the problems of interconnecting heterogeneous computer networks to allow communication between processes in different networks. Topics discussed include global addressing and routing techniques, level of network interconnection, extent of changes required in individual nets, and functions performed by the interface or gateway between the networks.