To submit papers: Send two copies of a paper of interest to the computer field to IEEE Computer Society Publications Office, 5855 Naples Plaza, Suite 301, Long Beach, CA 90803. Include a 50-100 word abstract, a list of index terms (four maximum), and a cover letter giving permission to enter the paper in the Repository (entry in the Repository does not constitute publication).

All submitted material should be unbound, printed one side only in black on 8½" × 11" white paper.

To order papers: State the R number, listed before the author’s last name, of each paper you order. Prices are 10 cents per page, plus $1.00 service charge for orders under 50 pages.

Microfiche copies are available for $2.50 for manuscripts under 50 pages, plus $2.50 for each additional 50 pages or fraction thereof.

All Repository items must be prepaid except for companies or institutions with established accounts. A $2.00 invoice charge is added to all non-prepaid orders. Make your check or money order payable to the IEEE Computer Society.


Denoting by T1, P1, and S1 an edge tree, path or star, respectively, we show that the complete graph Kn can be decomposed into trees T1, T2, . . . , Tn, where Ti is either P1 or S1 for i = 1, 2, . . . , n. We also show that the same result holds for decomposing the complete bipartite graph Knn−1 (K0, n−1) for an odd (even) n. We then treat the cases of decomposing Kn into P1, P2, . . . , P2n−1 and K0,n+1 (with an odd n) into P2, P4, . . . , P2n. All of these results are being proved by decomposing the adjacency matrix of the appropriate graph.

A different kind of result is shown for decomposing a full m-ary tree.


Let c and d be two integers such that c > d ≥ 1. Let t denote the binomial coefficient (c). Given c distinct colors, we order the t subsets of c colors in some arbitrary manner. Let G1, G2, . . . , Gt be graphs. The d-chromatic Ramsey number, denoted by r(d)(G1, G2, . . . , Gt), is defined to be the least number p such that if the edges of complete graph Kn are colored in any fashion with c colors, then for some i the subgraph whose edges are colored with the ith subset of colors contains a Gi.

In this report, we study the case c = 3, d = 2. Results are obtained for cases where G1, G2, G3 are complete graphs or stars.


This thesis is concerned with the arithmetic and logic design of an arithmetic operation unit to be used in a computer environment in which the basic arithmetic operations satisfy the on-line property; that is, to generate the jth digit of a result (where a digit consists of a bits for base 2j), it is necessary and sufficient to have the operands available only up to the jth digit plus, in the case of division, a predetermined number of extra digits which correspond to an “on-line” delay. Since there is no on-line delay for addition, subtraction, and multiplication, the unit can begin generating result digits as soon as one digit of each operand has been input. The delay for division is shown to be a small, positive, radian dependent constant. To fulfill the on-line requirements, a set of left-to-right (most-to-least significant), digit-by-digit algorithms has been derived. The existence of such algorithms is contingent upon the use of a redundant representation for the result digits. These algorithms and a block diagram level implementation of the basic arithmetic unit are developed in the thesis.


BURSTLOCK is a digital phase-locked loop implemented using Burst Processing. It is used in a receiver to perform FM demodulation of commercial broadcast signals. It is also shown that BURSTLOCK has some theoretical advantages over conventional phase-locked loops.


This report contains an analysis of the TUTOR programming language used on the PLATO computer-aided instruction system. The analysis centers on the requirements for implementation on a small-to-medium scale computer system. Implementation choices and their reasons are outlined, and the results of one such implementation are presented. Special emphasis is placed on the desire for program transportability to and from the large PLATO system. Appendices contain descriptions and specifications of the critical software components of the implementation.


The program INDUCE-1 is a Pascal program which generalizes VL1j decision rules to form consistent, complete, and in near optimal VL1j decision rules under a user specified criterion. The paper supplies the list of parameters and commands for operating the program, the I/O files which the program uses, description of the program structure, and a complete program listing.
R77-290—Michalski, Ryszard S., “Toward Computer-Aided Induction: A Brief Review of Currently Implemented AQVAL Programs” (18 pp., Report No. UIUCDCS-R-77-874, University of Illinois, Urbana, Illinois). The paper reviews a set of computer programs whose purpose is to aid in solving certain classes of inductive tasks. The programs can be used, in particular, for determining most economical (according to a user-specified criterion), generalized, and discriminant description of given sets of data. The inferred (and initial) descriptions are constructed of various logical and set-theoretic operators (such as “not,” “and,” “or,” set membership, “climbing a generalization tree,” quantifiers, equivalence), multiple-valued variables (originally defined or new ones generated by the program), k-place predicates, and k-place functions. The underlying formalism for expressing descriptions is variable valued logic systems VLI (a form of a first-order predicate logic with additional operators) and VLI, a form of a multiple-valued propositional calculus.

R77-291—Yelow, Edward, “YAQ: A 360 Assembler Version of the Algorithm Aa and Comparison with Other PL/1 Programs” (54 pp., Report No. UIUCDCS-R-77-840, University of Illinois, Urbana, Illinois). This paper contains a user’s guide and program description of the program YAQ. Included are some examples of VLI formulas synthesized by the program and a comparison between the assembler version and PL/I versions of the covering algorithm AQ.

R77-292—Liu, Jane W. W., and Chung L. Liu, “Performance Analysis of Multiprocessor Systems Containing Functional Dedicated Processors” (24 pp., Report No. UIUCDCS-R-77-625, University of Illinois, Urbana, Illinois). General models of multiprocessor systems in which processors are functionally dedicated are described. In these models, processors are divided into different types. Clearly, multiprocessor systems with identical processors is a special case of our models. These models also include the job shop problem in which there is exactly one processor of each type. Worst case performance bounds of priority-driven schedules are obtained.

R77-293—Roman, Gruia-Catalin, “An Argument in Favor of Mechanized Software Production” (39 pp., Washington University, St. Louis, Missouri). A new software engineering methodology called program control structuring is proposed. The philosophy behind program control structuring is that of minimizing the probability of error in the design and implementation process by structuring the language in a manner oriented towards the construction of maintainable software. The particular standardization approach used by program control structuring is shown to provide (1) a program structure that is simple and flexible, readily understood, easily developed, and inexpensively maintained; and (2) a high quality and partially mechanizable structure-oriented documentation scheme. Subsequently, it is suggested that many of the standards are machinizable as a necessary condition for achieving high productivity and improved program quality. Finally, the mechanism of software production is proposed as a step included to precede the use of generation of automatic program generation systems.

R77-294—Liu, Ming T., “The Distributed Loop Computer Network (DLCN)” (74 pp., Ohio State University, Columbus, Ohio). Considered as a means of investigating fundamental problems in distributed processing and local networking, the Distributed Loop Computer Network (DLCN) is envisioned as a powerful distributed computing system which interconnects micro, mini and micro-computers, terminals, and other peripheral devices through careful integration of hardware, software, and a communication channel. The network is constructed so that its users will see a single, integrated computing facility with great power and many available resources without being aware of the system’s actual organization and method of operation. System design of DLCN has been mainly in four areas: (1) a novel message transmission mechanism has been developed; (2) a bit-oriented, distributed message communication protocol (DLMCP) has been proposed; (3) the network operating system (DLOS) has been designed; and (4) a network command language is provided.

R77-295—Niemeegeers, I. G., and F. J. Mowle, “Data Rate and Time Delay Performance Modelling for Bit Oriented Computer Communication Protocol” (44 pp., Purdue University, W Lafayette, Indiana). Reliable communication between computers in a network requires protocols to control an error-free operation in an error-prone environment and set rules for the control and sharing of the communication resources. The SDLC line protocol for a point-to-point full duplex link is considered. Two performance measures are examined: the observed maximum effective data rates for batch transfers, and the time delay experienced by a single message. A renewal process model is used to determine the relationship between maximum data rate and packet length. An E-net model of an error-free link is presented to serve as a basis for a simulation analysis of the time delays.

R77-296—Murthy, K. R. Srinivasava, “Quadrant Conversion for Sine-Cosine Function Generators” (6 pp., Isro Satellite Centre, Peenya, Bangalore, India). This paper describes a simple method for converting angles in BCD ranging from 0° to 360° to the corresponding angles in the first quadrant (0° to 90°). The circuit to achieve this has been given. This interface circuit enables a digital sine function generator which can accept BCD angles from 0° to 90° to accept angles ranging from 0° to 360°. With suitable modification, this circuit can be used with cosine and other trigonometric function generators.

R77-297—Tront, Joseph G., and Donald D. Givone, “Multiple-Value Logic Gates Using MESFET’s” (25 pp., SUNY at Buffalo, Amherst, New York). The majority of the work in the multiple-valued logic area has been devoted to the study of various algebras and their properties. One such algebra was proposed by Allen and Givone. This paper is concerned with circuit realizations for this algebra using GaAs MESFET’s.

R77-298—Amzen, Robert J., and Robert A. Ellis, “A Comparative Study of Several Logic Family Implementations of Microprogrammed Processors” (26 pp., Washington University, St. Louis, Missouri). A comparative study is made of some microprogrammed processors. Generally microprogram controllers, arithmetic logic unit, and register file. Implementation of each system in standard 7900 TTL, Schottky TTL, ECL 10K, and the generation of input test data and is considered. A major goal was to gain a general understanding of relative and absolute cycle times achievable with each implementation in addition to isolating critical paths and bottlenecks. Comparisons of speed, area, and functional capability are included.

The authors found that little time penalty is paid for using 2900-bit slice parts as compared to standard 7400 implementations; however, Schottky systems are always faster than either, and in addition, are generally as fast as 10800 implementations.

R77-299—Moranda, Paul B., “Quantitative Model for Software Reliability Measurements” (201 pp., McDonald Douglas, Huntington Beach, California). This research in software reliability is primarily based on probability: as applied to error detection rates; as applied to the generation of input test data; and is applied to the economics of random versus constructed test cases. Random numbers are employed as input to programs instrumented to detect the use of the program’s segments. An algorithm is employed to establish the total number of test sequences which are likely to be eventually driven by random inputs. Tools for determining counts of these execution sequences are described and tables which facilitate estimation of the number of logical paths and related parameters are provided.

R77-300—Moranda, Paul B., “A Comparison of Software Error-Rate Models” (30 pp., McDonald Douglas, Huntington Beach, California). Five detection (failure) rate models for the software error process are compared: the Jelinski-Moranda model; a variation of that model; a geometric-Poisson model; the Shooman model; and the Schick-Poisson model. The models are applied to the same data (a daily record of the number of errors found and the CPU time used), which favors the Shooman model. Maximum likelihood estimates of the total error content, mean time to next error, and the degree of testing completeness are developed from a small time segment of the data, and estimates are compared where possible. The estimates.
of total error content are compared to the total number eventually found. All models produce reasonable estimates.

R77-301—Jubin, John C. III., and Vyta B. Gylys, “Executive and Software Architecture for an Actual Real-Time System of Distributed Microprocessors” (52 pp., Texas Instruments, Dallas, Texas)
The design of an actual executive (operating system) for a distributed system of minicomputers is described in this paper. The real-time computer and sensor system controlled by this executive will provide high-precision, satellite aided navigation capabilities for an aircraft. The implied data processing functions range from extremely fast-access, high-speed interactive receiver control processes to relatively long-running floating point navigation computational processes. The choice of a microprocessor as the basic processing element, coupled with the complexity and real-time constraints of the problem, required a distributed network of these processors. Partitioning of the computational workload over the network was one of the critical issues of design.

R77-302—Harte, D. H., W. H. Sterling, and J. E. Shen, “Design of a Raster Display Processor for Office Applications” (36 pp., Xerox Corporation, El Segundo, California)In recent years there has been a pronounced emphasis to develop alphanumeric display systems for document creation, filing, and text editing applications. This paper describes the design of a raster scan Display Processor which was specifically implemented to produce full-page, high-quality images (8½” x 11” or 11” x 8½”) which emulate typewriter output. A dual data path architecture is presented in which a Character Generator produces video images of text data, while a FAX Generator generates imaged video images corresponding to graphic forms used in the office. The design approach chosen provides degrees of functional richness that are not commonly present in current systems. Furthermore, a modular approach allows for a choice of whether or not to include forms overlay capability based on application requirements.

R77-303—Beale, G. O., and Gerald Cook, “Optimal Digital Simulation of Aircraft Via Random Search Techniques” (28 pp., University of Virginia, Charlottesville, Virginia)This paper discusses a technique for the discrete-time simulation of an aircraft to be used in the simulation process. The integration operator can be optimized for a particular system subjected to a set of specified inputs. The class of systems being investigated are those which are described by a set of state equations. A discrete time integration operator with certain free parameters is hypothesized. An adaptive random search optimization (ARSO) technique is used to find the optimum values for these parameters. Examples are presented to show the effectiveness of this technique.

Order by R-Number. Use the Repository order form on page 122.

R77-304—Goldstein, Lawrence H., “A Probabilistic Analysis of Multiple Faults in LSI Circuits” (32 pp. Sandia Laboratories, Albuquerque, New Mexico)This paper presents a method for identifying the most probable faults in a complex LSI circuits by assessing statistical information relating to physical defects that develop during wafer processing with chip layout data. Equations are developed that translate probability distributions on physical defect size and location into probability of single and multiple faults. By generating tests for only those faults that are likely to occur, the test sequence generation problem for LSI circuits becomes computationally feasible. The sacrifice required to achieve feasibility is a confidence level of less than 100% that a circuit which passes the test is actually fault free.

R77-305—Marcott, Michael, and Frederick G. Sayward, “The Definition Mechanism for Standard PLI” (79 pp., General Motors Research Laboratories, Warren, Michigan)The mechanism used to define the programming language PLI in the recently adopted American National Standard is presented. This method provides a rigorous though semi-formal specification of the language. It uses the model of translation of programs into an abstract form to define the context-free and context-sensitive syntax. The semantics are defined by the interpretation of the abstract form of the program on a hypothetical machine. The method and metalanguage are presented along with several small examples to illustrate the definition techniques. The completeness of the definition process is shown by the definition of a small example language.

R77-306—Misra, Jayadev, “A Technique of Algorithm Construction on Sequences” (16 pp., University of Texas-Austin, Austin, Texas)A technique is presented which is shown to be useful in designing algorithms that operate on sequences (strings). A generalization of the principle is presented for more general data structures.

R77-307—Cooper, David B., “Maximum Likelihood Estimation of Markov Process Blob Boundaries in Noisy Images” (16 pp., Brown University, Providence, Rhode Island)Effective and elegant procedures have recently been described for determining by computer a highly variable blob boundary in a noisy image [1,2,3]. In this paper the author investigates the case if the blob boundary is modeled as a Markov process and the additive noise is modeled as a white Gaussian noise field, then maximization of the joint likelihood of the hypothesized blob boundary and all of the image data results in roughly the same blob boundary as its boundary as does one of the aforementioned deterministic formulations [2]. More generally, the author argues that maximization of the joint likelihood of the hypothesized blob boundary and the entire picture function is a fundamental approach to boundary estimation or the estimation of image features (roads, rivers, etc.) in images.

R77-308—Mitra, Sanjit K., and Gregory K. Sorknes “On the Implementation of a 2-D FIR Filter Using a Single Multiplier” (10 pp., University of California)A general scheme for implementing a two-dimensional FIR filter using only one multiplier is outlined. The realization is based on sequentially multiplexing the multiplier inputs and is illustrated with the aid of a block diagram and the sequence of operations for a 3 x 3 filter.

R77-309—Murthy, K. R. Srinivasu, “Quantitative Analysis of Priority Failure Logic Functions” (8 pp., Isro Satellite Centre, Peenya, Bangalore, India)A method is given for calculating the probability of occurrence of the output event from priority NAND gate, priority-NOR gate, priority-mCh gate priority-exclusive OR gate. All these priority logic functions are assumed to be fed by non-reparable, exponentially distributed, S-independent events as inputs.

R77-310—Belal, A. A., “Decimation in Frequency for p-Dimensional Fast Fourier Transforms” (Control Data Corporation, Minneapolis, Minnesota) p-Dimensional Discrete Fourier Transforms are obtained by applying one-dimensional fast algorithms along each dimension [1,2,3]. The authors show that if instead the processing is done in p-dimensional blocks using a p-dimensional version of the one-dimensional decimation in frequency FFT algorithm, then a substantial amount of savings in the number of arithmetic operations is achieved. In particular, for a p-dimensional cube with side length 2 p the complex multiplications is reduced by the factor 2 p-1 while the number of complex additions remains approximately the same.

R77-311—Tanenbaum, Andrew S. “Design and Implementation of an Algol 68 Virtual Machine” (86 pp., Mathematical Centre, Boerhaavestraat, Amsterdam)A virtual machine specifically designed for running Algol 68 programs is proposed. The instructions and implementation of this machine are discussed in detail. A method of implementing the run time organization for this machine, based upon use of descriptors, is given. Memory organization, garbage collection, procedure and range entry and exit, and parallel processing are among the topics covered. The machine has been designed so that a hardware implementation of it could have a single instruction for all assignments, and a single instruction for garbage collection.

R77-312—Blue, Richard B., Sr., and Gerald E. Short, “Computer System Security: Technology and Operational Experience” (37 pp., TRW, Redondo Beach, California)This publication presents two papers, “Computer System Security: Technology,” and “Computer System Security: Operational Experience,” which review the state-of-the-art in computer security in the light of recent work done by TRW on contract to IBM. Discussed are the issues faced by a computer center manager in balancing security considerations with other performance considerations.
R77-313—Dreyfus, J. M., and P. J. Karcsony, “The Preliminary Design as a Key to Successful Software Development” (27 pp., TRW, Redondo Beach, California)
This paper discusses a preliminary design methodology developed by TRW which has successfully been applied to large-scale software development and the benefits obtained therefrom.

R77-314—Dreyfus, J. M., “Use of Simulation in the BMD Systems Technology Program Software Development” (27 pp., TRW, Redondo Beach, California)
This paper discusses a brief overview of the Ballistic Missile Defense (BMD) Systems Technology Program and identifies several specific software development issues related to moving the development risk forward in time. Four different classes of simulation techniques are identified, and the specifics of their implementation and the resulting benefits on this project are discussed.

R77-315—Short, G. E., “Threats and Vulnerabilities in Computer Systems” (50 pp., TRW, Redondo Beach, California)
This report covers the vulnerabilities in a computer environment for which safeguards must be established and, subsequently, for which costs must be determined. It is not meant to be a total definitive catalog for all possible vulnerabilities on a computer system; rather it is meant to be an analysis of that set of vulnerabilities that are likely to be encountered across a general set of user environments.

R77-316—Stepczyk, F. M., “Requirements for Secure Operating Systems” (133 pp., TRW, Redondo Beach, California)
The purpose of this study is to develop a comprehensive set of requirements, together with techniques for meeting these requirements, which, when applied to a computer operating system, will serve as a basis for certifying that system as secure.

This report identifies several approaches to obtaining verified programs: programming methods, program analysis, and testing. The organization of this paper closely follows the chronology of the software development cycle. Verification considerations presented are meant to improve efficiency and the thoroughness of verification at reductions in cost.

R77-318—Pinchuk, P. L., “TRW’s Evaluation of a Secure Operating System” (85 pp., TRW, Redondo Beach, California)
This report presents a chronology of events related to the Resource Security System (RSS) implementation at TRW, together with the results obtained. The report also presents an evaluation of RSS in terms of the security features included in the system, plus evaluations of RSS in terms of usability, reliability, maintenance requirements, and performance costs.

R77-319—White, R. C., “Issues in Secure Computer System Certification” (41 pp., TRW, Redondo Beach, California)
In this report, the results and conclusions of a study of computer system certification and its associated issue are summarized. Emphasis is on a comprehensive explanation of computer system certification and its key problems, rather than on their exhaustive resolution.

This paper describes some of the important lessons about software management that have been learned during the past two decades. Mr. Royce has experienced different degrees of success with respect to arriving at an operational state, on-time, and within costs; in doing so he has become prejudiced by his experiences and relates some of these prejudices in this paper.

R77-321—Kastelein, J. E., “Flight Software Quality Assurance” (42 pp., TRW, Redondo Beach, California)
This report documents the results of the third phase of study of techniques for reducing flight software verification costs. The objective of this phase has been to define the requirements for a system to reduce costs in development and verification of man-rated flight software and to establish a conceptual design of a specific system configuration that meets these requirements.

R77-322—Kastelein, J. E., “Quality Assurance Requirements During Flight Software Development” (41 pp., TRW, Redondo Beach, California)
This report documents the results of the second phase of study of techniques for reducing flight software verification costs. Phase 2 of this study analyzes the flight software development process and tries to identify and analyze the quality assurance requirements which should be satisfied at each phase of the development process.

R77-323—Gibson, C. G., and H. L. Widdifield, “Flight Software Development and Verification System Requirements” (84 pp., TRW, Redondo Beach, California)
This report documents the results of the first phase of study of techniques for reducing flight software verification costs. This phase consists principally of a review of current techniques and current literature on all facets of software quality assurance, and a study of basic problems and concepts in software quality assurance. The activity has as its objective the development of a basic understanding of the overall software quality assurance problem in order that the more specific objectives of subsequent phases can be placed in proper perspective. Therefore, the discussions of the basic quality assurance concepts and problems in this report are not exhaustive in scope or detail.

R77-324—Wolverton, R. W., and G. J. Schick, “Assessment of Software Reliability” (30 pp., TRW, Redondo Beach, California)
This paper discusses methods for and problems in achieving reliability of large-scale software systems. It also analyzes software development and test management procedures which lead to software reliability.

This paper discusses the TRW Product Assurance Confidence Evaluator (PACE) system, an evolving collection of automated tools which provides support in various phases of software testing.

R77-326—Boehm, B. W., “Keynote Address: The High Cost of Software” (13 pp., TRW, Redondo Beach, California)
This paper addresses three main questions: (1) How high is the cost of software? (2) Where do the costs go? (3) What factors influence the costs? (or, what can we do about them?). For reference, “software production” here includes all the effort involved in producing and maintaining the necessary executive, support, and applications programs and their documentation, starting from a reasonably well-defined functional specification.

R77-327—Gearhart, Jack B., “Keynote Address: Data and Configuration Management of Computer Programs” (10 pp., TRW, Redondo Beach, California)

This paper was presented at the Sixth Annual Workshop of the Electronic Industries Association, Phoenix, Arizona, in November 1972. It touches upon three topics: trends in software configuration and data management versus hardware C&DM; some do's and don'ts in the design, development and implementation of software C&DM techniques; and some suggested guidelines.

R77-328—Boehm, Barry W., “Structured Programming: Problems, Pitfalls, and Payoffs” (14 pp., TRW, Redondo Beach, California)

This paper analyzes recent experience in using structured programming techniques at TRW and elsewhere. The first section discusses some recent quantitative results and their interpretation. The second section presents 13 of the most significant problems and pitfalls encountered to date using structured programming techniques and discusses some enhancements which help avoid the pitfalls.

R77-329—Boehm, Barry W., “Software and Its Impact: A Quantitative Assessment” (54 pp., TRW, Redondo Beach, California)

This paper discusses quantitative data which demonstrates the impact of software on operational performance and to provide perspective on R&D priorities. The main purpose of this article is to make these scanty but important data and their implications better known, and to convince people to collect more data of the same type.

R77-330—Mullin, Frank J., “Considerations for a Successful Software Test Program” (37 pp., TRW, Redondo Beach, California)

This paper discusses software testing and software test activities as they should occur during the software development process. It notes that a well-planned test program starts when software requirements are being produced and continues on through acceptance testing. It describes the activities and responsibilities of the group assigned to formally test the software and discusses their interactions with the people preparing the software requirements and the people developing the software. Three levels of testing are defined, and suggested test support software is identified.

R77-331—Ingrassia, Frank S., “The Unit Development Folder (UDF): An Effective Management Tool for Software Development” (18 pp., TRW, Redondo Beach, California)

This paper describes the content and application of the Unit Development Folder, a structured mechanism for organizing and controlling software development products (requirements, design, code, test plans/data) as they become available. Properly applied, the Unit Development Folder is an important part of an orderly development environment in which unit-level schedules and responsibilities are clearly delineated and their step-by-step accomplishment made visible to management.