A collection of over 2600 technical papers (some refereed, some not), covering the full range of computer system design and maintained by the Computer Society as a service to the information processing community.

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, unstapled, printed one side only in black on 8½" x 11" white paper.

Material not conforming to the above requirements will not be accepted.

To order papers: State the R-number, listed before the author's last name, of each paper you order. Prices are 12 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.


R78-34—Reiss, Steven P., "A Theoretic Approach to Automatic Programming" (58pp., Brown University, Providence, Rhode Island)

This paper gives bounds and directions for practical automatic programming. The author first demonstrates that automatic programming is a problem of inverting a semantics-specifying mapping. Then, using a series of simplifications and formalisms, he shows that a specific problem is defined by a mapping that takes the target language into the source, and that automatic programming simply involves inverting this mapping. This concise model is used to determine the bounds of practical automatic programming and those types of mappings that can be used in a practical system. The complexity of the problem depends on the type of mapping to be inverted. The author provides examples of practical problems that can be defined using the appropriate mappings, and discusses new directions for automatic programming.

R78-35—Shrivastava, S. K., and J. P. Banatre, "Reliable Resource Allocation Between Unreliable Processes" (38pp., University of Newcastle upon Tyne, Newcastle upon Tyne, England)

Basic error recovery problems between interacting processes are discussed, and the desirability of having separate recovery mechanisms for cooperation and competition is demonstrated. The paper then concentrates on recovery mechanisms for processes competing for the use of the shared resources of a computer system. Appropriate programming language features are developed based on the class and inner features of Simula and on the structuring concepts of recovery blocks and monitors.

R78-36—Julien, G. A., "Residue Number Scaling and Other Operations Using ROM Arrays" (43 pp., University of Windsor, Ontario, Canada)

Over the last two decades there has been considerable interest in the implementation of digital computer elements using hardware based on the residue number system. This paper considers implementing such systems with arrays of look-up tables placed in high density read-only memories. The type of system discussed is restricted to one in which the only operations are addition, subtraction, multiplication, and scaling by a predetermined constant. Special attention is given to the scaling algorithm, and two different algorithms are developed. Two applications are discussed: one involving the implementation of a multiplier with scaling and the other involving the realization of a second-order recursive digital filter section.


The author shows a one to one correspondence between all the regular k-ary trees with n internal nodes and certain integer sequences. After generating these sequences lexicographically, the paper discusses the ranking function and the unranking procedure. Relations to existing algorithms are also discussed.

R78-38—Perl, Yehoshua, "Average Analysis of Simple Path Algorithms" (27 pp., Report No. UIUCDCS-R-77-905, University of Illinois, Urbana, Illinois)

Given a graph of n vertices and e edges, the average complexity of several known simple path algorithms is analyzed. The average and the standard deviation of the number of the edges scanned to find the target vertex in both breadth first search and depth first search are shown to be of order n. Both the average and the variance of Prim's minimum spanning tree algorithm are shown to require \(O(n \lg n + (e/n))\) time. The same result holds for Dijkstra's shortest path algorithm. Kruskal's minimum spanning tree algorithm, which competes with Prim's algorithm, requires \(O(n \ln \ln n + e)\) on the average.

R78-39—Redinbo, G. Robert, "Finite Field Arithmetic on an Array Processor" (49 pp., Remselaer Polytechnic Institute, Troy, New York)

This paper develops algorithms for implementing finite field arithmetic on general
signal-processing machines. The emphasis is on achieving practical approaches compatible with a wide range of computer systems. The techniques easily mix with normal signal-processing algorithms, obviating the need for special finite field hardware. After outlining the basic principles of finite fields and the general architecture of array processors, the author focuses on multiplication and inversion as the major problem areas. The results employ the fast Fourier transform (FFT) and vector multiplication, both a forte of array processors. Only real arithmetic machine operations are used. The FFT size is proportional to the extension degree of the field, and relatively short transforms can handle very large fields. A logarithmic relationship exists between the field size and the transform length. These algorithms can be abbreviated when the multiplications or calculations use intermediate results. Several multiplicands can be treated simultaneously by extending the techniques for two operands. The corresponding FFT length still only increases linearly with the number of multiplicands. The author presents an iterative procedure for computing element inverses using multiplication algorithms.


For arbitrary \( k \geq 1 \) and \( \alpha \in (0, \pi/2) \), \( A(\alpha) \)-stable \( k \)-th order \( k \)-step formulas exist. Thus in an ODE solver, \( \alpha \) can be an extra parameter used to identify among a family of methods of order \( k \) the \( A(\alpha) \)-stable method that should be used for the particular problem. Two measures for assessing the accuracy of \( k \)-th order \( k \)-step formulas are proposed. The problem of finding the upper bound on the angle of absolute stability for the \( k \)-th order \( k \)-step formulas having the same accuracy (with respect to one of the measures) is considered. Analytical results are obtained for \( k = 1, 2, 3 \) whereas a numerical search is used for the cases when \( k = 4, 5, 6, 7 \).

R78-41—Bruening, James T., "Inverses of Transfer Function Matrices" (38 pp., Baker University, Baldwin City, Kansas) 

This paper investigates a general form for inverse systems of linear sequential circuits. The form for the transfer-function matrix is a linear combination of matrix extensions of the adjoints of square submatrices of the transfer-function matrix of the original circuit. Any inverse system can be written in the form presented here. This result is then utilized to generate polynomial inverses and yield the minimal delay. Finally, upper bounds on the minimal degree of all polynomial inverses of a matrix are established. The results of this paper are limited to transfer-function matrices of full rank.

April 1978

Software Professionals:

If you have a background in Systems Analysis, Systems Programming, Operating Systems Support, Data Base Management or Software Course Development and Teaching, we would like you to consider a unique opportunity to broaden your technical background while imparting your expertise to Digital employees and customers.

Responsible for providing technical training to Digital customers and employees, the Educational Services group is seeking professionals with a solid technical knowledge and a desire to develop or teach a broad range of advanced software instructional course materials required for the support of both current and future products.

As a member of our Educational Services group, you will be involved in the following areas of technology.

- Data Base Management
- Data Communications
- Commercial Applications
- Computer Networks
- Operating Systems (DOS, IAS, RSTS, RSX-11, RSX-20F, RT-11, TOPS-10, TOPS-20)
- Programming Languages (ALGOL, APL, ASSEMBLY, BASIC, BLISS, COBOL, DIBOL, FOCAL, FORTRAN, RPG)
- Real-Time Applications

If you are a highly motivated self-starter who can conceptualize what is required to solve specific problems and then develop the resources to solve the problems within a dynamic, unstructured environment, we would like to talk to you.

Forward your resume outlining salary requirements to Mary Ann Joyce, Digital Equipment Corporation, Dept. H4 3816, 129 Parker Street, Maynard, Massachusetts 01754. We are an equal opportunity employer m/f.