The Repository, a collection of over 1200 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 to Dr. Robert A. Short, IEEE Computer Society Editor-in-Chief, Department of Electrical Engineering, Oregon State University, Corvallis, OR 97331. 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 $.10 per page. A $1.00 shipping and handling charge will be added to all orders under 50 pages. Microfiche copies are available for $2.50 for each 50 pages of copy. Minimum incremental charge: $2.50.

All Repository items must be prepaid. Please make your check or money order payable to the IEEE Computer Society.

R75-28—Su, Stephen Y. H. and Edgar DuCasse, "Reconfiguration Scheme for Multiple Fault Tolerance" (23 pp., Su: Dept. of Electrical Engineering, City College, City University of New York; DuCasse: Dept. of Computer & Information Science, Brooklyn College, City University of New York).

A scheme and its realization are proposed for reconfiguring a multiple-valued NMR (N modular redundancy or N-input majority voting system) into a (N−2) - input majority voting system under single or double module failures. Since the NMR with binary variables is just a special case of a multiple-valued system, the scheme is good for both multiple-valued and binary digital systems. The proposed technique allows an N-module redundant system to tolerate N−2 failures instead of (N−1)/2 failures in NMR. Some failures intolerable by the hybrid redundancy system can be tolerated by the proposed scheme.


Bell, et al., introduced a set of Register Transfer Modules (RTM's) capable of implementing sequential as well as concurrent processing. This paper shows the insufficiency of the control module set in implementing concurrency. An n-synchronizer control module is introduced, and the modified control module set is shown to be sufficient in implementing the necessary controls.


The authors have developed a scan type color graphic display device which generates the standard NTSC TV signal according to picture data sent from a computer. The device produces picture and character signals independently and mixes them at the output point of the device. The picture display mechanism is based on a simple idea: a picture consisting of many colored polygons can be defined by vectors which include both the edges of the polygons and the colors on the right hand side of the edges. The data represented in vector format are converted into a picture signal in the device, so that the amount of data can be substantially reduced compared with that in the general scanning display, which requires the data from many picture cells. 228 characters, each of which consists of 32×32 picture cells, are displayed in a frame. Several thousand character patterns may be displayed by replacing the stored patterns in the pattern buffer. Polygons and characters are painted by colors selected from 32,768 color combinations.

R75-31—Howden, W. E., "Methodology for the Generation of Program Test Data" (31 pp., University of California, La Jolla, California).

A methodology for generating program test data is described. The methodology is a model of the test data generation process and can be used to characterize the basic problems of test data generation. It is well defined and can be used to build an automatic test data generation system. The methodology decomposes a program into a finite set of classes of paths in such a way that an intuitively complete set of test cases would cause the execution of one path in each class. The test data generation problem is theoretically unsolvable: there is no algorithm which, given any class of paths, will either generate a test case that causes
some path in that class to be followed or determine that no such data exists. The methodology attempts to generate test data for as many of the classes of paths as possible. It operates by constructing descriptions of the input data subsets which cause the classes of paths to be followed. It transforms these descriptions into systems of predicates which it attempts to solve.

R75-32—Yang, C. C. and M. A. Tarpy, "New Algorithms for Deriving All Pairs of Compatible States and All Maximal Compatible States and All Maximal Completions" (29 pp., University of Alabama, Birmingham, April 1975).

This paper reviews thoroughly all major contributions of previous work concerning the derivation of all compatible states up to the maximal level by various techniques. Then two new algorithms are proposed: one is used for deriving all pairs of compatible and incompatible states of any incompletely specified sequential machine by simply checking the contents of the closure classes with respect to each pair of states, and the other is for deriving all maximal completions of state compatible with respect to an incompatibility graph by manipulating some extended incompatibility relations. Separate working examples are given for illustrating both methods by hand computation. In addition, both algorithms are implemented by PL/I programs and executed with the examples as input data in an IBM System 370 Model 158. These demonstrations assure that both algorithms are feasible and efficient not only for hand computation but also for computer execution.

R75-33—Wakimura, Y., and N. Yoshida, "Design of Checking Experiments for Finite State Machines" (48 pp., Hiroshima University, Hiroshima, Japan).

This paper presents a new method for the design of the checking experiment for sequential machines which possess distinguishing sequences. Although the procedure described here is based on previous work, and as many as the classes of paths as possible, the procedure is such as to be utilized by hand computation. The method is to be applied to the design of checking experiments for sequential machines which possess distinguishing sequences.


It is shown that for any given 0 < N < M, the number of nodes in a random 3-2 tree, satisfies the equality 0.70N ≤ eN(1-N/2)N/2 < 0.79N. A similar analysis is done for general B-trees. It is shown that storage utilization is essentially 8N/9 for B-trees of high orders.


A brief overview is presented on the theory and philosophy of testing a sequence of digitally generated samples for the property of uniformity. Random sequences from a uniform distribution are a valuable starting point for the generation of random sequences from other probability density functions which may be required in computer simulations or Monte Carlo studies. The discrete Fourier transform is used to test the uniformity of the random sequence. The results of these tests are compared with conventional method of testing.


Present day computers have been designed with processing elements and memories in a one-to-one correspondence. For many problems this architecture limits the speed of solution. In this paper a machine architecture is presented in which processing elements and memories are considered independent resources. This architecture provides a technique for increasing the logical bandwidth of the memory without increasing the physical bandwidth. The scheme for interconnecting processing elements and memories is based on the mathematical formulation of a matrix-matrix product. Of interest in determining the usefulness of a particular computer architecture are the problem classes which it is able to solve efficiently. For this machine we consider several problems. On a serial processor the multiplication of two n x n matrices requires O(n^3) steps when using the classical algorithm or O(n log^2 n) steps when using Strassen's algorithm. We present an algorithm for our machine which performs this multiplication in O(log n) steps. This can easily be shown to be the minimum time possible. We also consider the solution of linear triangular systems of equations. This problem requires O(n^2) steps for a sequential processor, but O(n log n) steps for a parallel processor of the ILLIAC IV-type. We present a parallel algorithm suited to execution on our machine which solves these systems in O(log^2 n). This algorithm is based on an extension of the principle of recursive doubling. In this type of algorithm, we present algorithms for several combinatorial type problems. In particular, we give methods for performing permutations and sorting. These algorithms require O(log n) steps when operating on n items.

R75-38—Houser, T. J., "Manual of System Development Procedures" (22 pp., Computer Services Center Bulletin No. 12, Millersville State College, Millersville, Pennsylvania).

The system development process is often properly described as a series of successive approximations. One first conceives of a system in a general way and, after gathering additional information, a much better conception of the system is obtained. Finally, during the period of manual procedure development, a close approximation to the desired system is reached. Sometimes patience and almost evangelistic faith in the long term advantages of the system are required at all levels, and poor system development efforts are often the result of management not meaningfully involving itself in the development of its own system. Although the overall structure of the planned Millersville information systems will impose certain guidelines on the development of other computer systems, an orderly process for the development of all systems is indispensable to informed management decisions. Since the computer services center cannot itself make all of the priority decisions, the support and awareness of top management and the participation of middle management must be obtained throughout the process. It is the purpose of this manual to delineate the various responsibilities which must be assumed in the system development process if such support and participation is to be effective. The format and content of this manual is based on a similar tool used at Mankato State College (Minnesota) and on
numerous books on business systems analysis and data processing management.


The concept of a semiautomatic directed vision processor has been presented in an earlier paper. The current paper outlines the progress in developing the stages of early visual processing in that system. It describes a parallel structure, a compiler of finding "preprocessing" configurations which transforms and reduces large amounts of visual data in a layered fashion. The cone is a structure of layered spatial arrays with level 0 containing the original image and a classification system. To 4, square arrays consisting of 128², 64², 32², and 16² points. Information flows up, down, and laterally within the cone via a sequence of local parallel functions. Any particular function applied to some level is unitarily in parallel and applied across the array. As an example of the flexibility of the structure, a top-down parallel line finder is presented. Information is reduced upward to level 4, lines are found on this level, and subsequently projected downward (level by level) to the next level, refinements are made in the lines by utilizing information at the current level and from above. In addition, we describe parallel texture analysis, region growing, and shape extraction. The techniques are applied to a color image of a natural outdoor scene and presented in a variety of examples. The goal of this system is to develop functions which extract visual features useful to perception while providing a suitable interface between parallel processing and sequential processing between processed visual data and symbolic information. The utility of this structure for higher level analyses is discussed.

R75-40—Hanson, A. R., E. M. Riseman, and E. G. Fisher, "Context in Word Recognition" (25 pp., COINS Technical Report 74C-6, University of Massachusetts, Amherst, Massachusetts). Relatively low character error rates can often lead to prohibitive levels of word error rates. This paper is an examination of several techniques for integrating an independent contextual postprocessor (CPP) into a full recognition system. The CPP detects errors and is the control structure for directing additional processing for correction of errors. The contextual postprocessor can correct many errors directly. In those cases where the correction process leads to ambiguity, this paper compares the relative payoff of various types of additional processing. These include reclassification under the supervision of the CPP and requests for further specific measurements under the supervision of either the CPP or the classifier. Most of the payoff is obtained by CPP-directed reclassification. This only requires the CPP to have the classifier likelihoods fed forward to it. Therefore, the CPP can be built independently of the rest of the classifier and standardized. Experimental results demonstrate the effectiveness of the CPP by analyzing a system with a weak set of features and the simplest of classifiers. The techniques for additional processing result in more than an order of magnitude improvement. A 45% word error rate is reduced to about 2% word error rate and 1% reject rate. Presence of a dictionary allows these figures to be reduced even further.

R75-41—Liu, J. J. and L. D. Rudolph, "A Direct Method of Computing the GNAF of an Integer" (6 pp., Syracuse University, Syracuse, New York).

This correspondence presents a simple, direct method, for nonadjacent form and arithmetic weight of an integer with respect to an arbitrary radix.


A lot of programs have been written dealing with texture recognition and other areas of artificial intelligence. The large amount of core these programs take up makes it impractical to create an entire program in core before execution. Therefore a MONITOR program that loads each subprogram at the time it is needed is desirable. Also, the fact that the flow of control between these subprograms may not be easily predetermined makes it necessary that the MONITOR be flexible enough to allow programmers to easily modify the flow of control during execution. A MONITOR which fulfills the above needs in research in artificial intelligence was implemented on the IBM 360-75 under OS MVT release .20.6 at the University of Illinois Urbana-Champaign. In addition to general design principles of the MONITOR, the paper also discusses methods of setting up a supervisor-slave type system using the MONITOR, as well as possible ways of interfacing interactive console support to the MONITOR.


The reconfiguration of service requests with memory resource availabilities presents a formidable problem for many computer service centers. When the sum of the service requests for a given resource exceeds the amount of the available resource, then that resource is constrained and scheduling techniques must be used to maximize the utilization of that resource. Two types of memory resource are considered: fragment-ed resources which can be allocated in pieces and contiguous resources which may only be allocated in one piece. An algorithm is developed which attempts to minimize the maximum flowtime for a set of multi-programmed jobs while minimizing the idle resources.


This is a revised manual for NOR network transduction programs NETTRA-P1, NETTRA-P2, and NETTRA-P2 which prunes connections in a given network, without creating new connections, according to the principles discussed in a previous paper by the authors and Y. Kambayashi. Networks produced by these programs are always subnetworks of the originally given networks.

R75-45—Dempyrot, M., "Coassignment on Universal Macromodules for Diode-Free LSI Programming" (26 pp., IRIA, Rocquencourt, France).

A class of universal macromodules for programmable sequential logic is described together with a state-assignment technique which offers a substitute to firmware or chartware for LSI circuits. A generalized model of sequential machines—called polymaton and previously applied to real-time processing—together with a fixed-record data representation of its dynamics proves to be the adequate tool for design automation.


A few years ago, after a careful examination of many existing automated test generation systems, Grumman Aerospace decided to develop a completely new system aimed primarily at sequential networks, and flexible enough to be applicable to the next generation logic. This system, called LOGOS, embodied the results of past research and new concepts from the viewpoints of logic and system design of digital systems. The results were formulated into several algorithms easy to understand and convenient to implement.


In Part I, the essential descriptive features of a new tabular-graphical algorithm that closely approximates the effort required in Karnaugh map minimization of Boolean functions are presented. The directed search tree approach is used to find a minimal cost covering of the function. Only necessary prime implicants are identified using techniques of essentiality and a form of row dominance. The algorithm is not limited by the number of variables of the function, and functions of up to 15 variables have been solved by a very fast computer implementation of the algorithm. Part II of this paper discusses the implementation of this minimization algorithm in both manual and computer programmed form.


The available page replacement strategies assume that either behavior of various programs is constant or approximated on the basis of some function. The authors here try to describe a page replacement algorithm which takes exact values of parameters of the program behavior.