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 $1.10 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 50 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.

In order to provide facilities for solving problems of a practical size, the special procedure proposed here considerably reduces computer storage requirements. For the case of identical processors two approaches for solving the problem have been compared.

R76-283—Cornish, Merrill, “The d-Algorithm for Sequential Circuits: An Extension and its Application” (5 pp., Texas Instruments Inc., Austin, Texas)

Both recently announced an R-Design Form which extends his d-algorithm to sequential circuits. This correspondence offers TI’s Advanced Scientific Computer as a practical application of this method. This design form offers automatic test generation for sequential logic plus an excellent checkout and maintenance tool.


The article describes a variable high-speed incremental computer that adopts the basic concept underlying a digital differential analyzer (DDA). Its structure closely resembles that of a microprocessor and includes the following features: floating point arithmetic, a word length transfer with the elimination of a residue (R) register, multibit multiplication, and a flexible software scheme of interconnections that includes the use of a stack to avoid redundant operations. The proposed structure has been simulated by solving a variety of differential equations with distinctly accurate results.

R76-285—Lalonde, W. R., “An Efficient LALR Parser Generator” (70 pp., University of Toronto, Toronto, Canada)

The implementation of an efficient parsing table generator for LALR(k) grammars input in BNF format is described. The full generality of LALR(k) context free grammars including empty right parts is processible.

R76-286—Potvin, John N. T., “The Star-Ring System of Loosely Coupled Digital Devices” (49 pp., University of Toronto, Toronto, Canada)

The design of a general system to loosely couple digital devices at high speed is discussed. This system is designed to be easily interfaced and expandable up to eight high-speed ports. Each device connected to the system is aware of the only relatively simple interface of the system, not the numerous complexities of the other devices also connected to the system. The system has been constructed and is capable of sustaining eight simultaneous transmissions of over 2.5 Mbytes/sec. each. Some considerations relating to present day digital computers and digital devices which have been connected to the system are also presented.

R76-287—Evans, Kenneth B., “Design Study for a Two-Dimensional Computer-Assisted Animation System” (110 pp., University of Toronto, Toronto, Canada)

This is a proposal and design study for a system of programs which will create and animate two-dimensional line drawings interactively. The system allows the user to create, manipulate, and animate both simple and hierarchically structured pictures, to create his own special purpose programs to extend the system, and to store large quantities of data in a compressed form in picture, scene, and film libraries. The thesis concentrates on two major system design aspects: data structures and man/machine interface procedures.


Project SUE is designing and building an extensible timesharing operating system for the IBM System/360 series of computers. This report is a preliminary description of its general structure and of the design of particular components in varying detail. Functional specifications
are given for the kernel and the system primitives. The structure of the I/O system is outlined. General functional specifications are presented for the basic file system and the bank. A detailed description is given of the system language developed in the project to facilitate the writing of the operating system.

R76-289—James, Lewis R., “A Syntax Directed Error Recovery Method” (64 pp., University of Toronto, Toronto, Canada)

A table-driven, syntax-directed scheme for automatic error recovery during program compilation is described. It is designed to be used in a table-driven LALR parser which can form the nucleus of a language translator. After the underlying theory is reviewed, the implementation of the error recovery scheme is discussed. Examples of its performance are given and comparisons are made with some other methods. It is concluded that it would be worth including in a compiler writing system for LALR languages.

R76-289—Sevcik, Kenneth C., “The Use of Service Time Distributions in Scheduling” (138 pp., University of Toronto, Toronto, Canada)

Analytical solutions are sought for various special cases of a general scheduling model based on the following assumptions: (1) work is scheduled on a single processor, (2) the scheduling objective is to maximize service rendered, (3) a loss function associated with each request reflects the decline in utility caused by its delayed completion, (4) each request is associated with a service requirement for which the distribution of service times is known, (5) the process by which new requests arrive is known for each request class, and (6) preemptive of a request not yet completed requires a fixed amount of processor time. After a detailed examination of the model and a discussion of past work on related models, various special cases are studied. First, free preemption, linear losses, and no arrivals are assumed. A new scheduling discipline called smallest rank, or SR, is defined and its optimality is proven. Next, arrivals are considered. With free preemption, linear losses, known service times, and Poisson arrivals, a well-known rule, SRPT/c, is shown to be optimal within a broad class of practical scheduling rules. An argument is made that the class necessarily encompasses the globally optimal rule. Finally, significant preemption costs are considered. An optimal preemption strategy is derived and proven for a simple system. Consideration is also given to a system with a hyperexponential service time distribution, a distribution found empirically in many time-sharing computer systems. Some other cases are discussed briefly, then generalizations of the model suitable for future study are proposed.

R76-289—Turnbull, C. J. M., “A Comparative Analysis of Several Disk Scheduling Algorithms” (81 pp., University of Toronto, Toronto, Canada)

An analytical approach is taken to the comparison of different disk scheduling policies in terms of average waiting time. The results are combined with similar ones due to Coffman et al. The comparisons confirm conclusions reached using simulation techniques; in particular, it is shown that SCAN and CSCAN are the “best” of the algorithms considered for the shortest seek-time-first policy are also presented.

R76-289—Wortman, David B., “A Study of Language Directed Computer Design” (220 pp., University of Toronto, Toronto, Canada)

Language directed computer design is an approach to designing computers which implements, in the machine hardware, operations and data structures normally associated with high-level programming languages. This dissertation examines the use of scientific experimentation as a tool for the design and evaluation of language directed computers. Experimental techniques are illustrated with a case study of the design, improvement, and evaluation of a language-directed computer which implements a dialect of PL/I called Student PL. The improved Student PL machine is compared with a contemporary general purpose computer, the IBM System/360. In terms of traditional measures of computer performance (number of instructions executed, number of bits of information accessed, number of memory references required) the Student PL machine is shown to be considerably more efficient for the execution of Student PL programs.

R76-289—Lazowska, E. D., “Scheduling Multiple Resource Computer Systems” (124 pp., University of Toronto, Toronto, Canada)

The goal of this thesis is to find “good” scheduling rules for computer systems that have more than one scarce resource, and to investigate program interactions in such systems. The approach involves the analysis of yet another queueing network model. The basic model includes two processors, which may be thought of as a CPU and an I/O channel, and two program classes with distinct time characteristics. Three types of objective functions are considered: total processor utilization, program run time expansion factor, and various rewards for program completion. A scheduling discipline optimal with respect to the first two criteria is found. It is hoped that a solid understanding of the interaction effects that arise in this relatively simple system will be useful for the formal and intuitive comprehension of more complex environments. The thesis includes a survey of various heuristic approaches to the scheduling of multiple-resource computer systems, a discussion of more sophisticated queueing network models in the literature, and an annotated bibliography.

R76-289—Tschritzis, D., “A Network Framework for Relation Implementation” (39 pp., University of Toronto, Toronto, Canada)

A network model is defined in terms of simple relationships among data. The hierarchical and network approaches to DBMS architecture are outlined using the model. A network language is proposed to declare and manipulate access paths between data. It is claimed that the proposed framework can be appropriate as a basis for relation implementation. Finally, some implementation problems for building access paths are discussed.

R76-289—Farley, J. H. Gilles and Stewart A. Schuster, “Query Execution and Index Selection for Relational Data Bases” (81 pp., University of Toronto, Toronto, Canada)

An algorithm to evaluate queries is presented, and it will be argued that the algorithm is minimal with respect to the number of relational accesses and with respect to the merging of partial results. The algorithm’s unique quality is its efficiency in evaluating partially inverted relations. A simple cost function which can be used to reflect the complexity of the query and to predict its response time is derived from the algorithm. The cost function forms the basis of a procedure to suboptimally the selection of the domains to be inverted. The domains to be inverted are selected by analyzing a typical set of queries submitted to the system. Such a method does away with usual methods of updating usage counters for every domain and relation in the system. In this approach, the selection of a good set of inverted lists is based on the algorithm which uses those lists.

R76-289—Holt, Richard C. and David B. Wortman, “Structured Subsets of the PL/I Language” (41 pp., University of Toronto, Toronto, Canada)

This report describes a sequence of subsets of the PL/I language that has been designed for the purpose of teaching introductory computer programming. The
eight subsets restrict PL/I to those language features which the authors feel encourage good programming habits. The subsets are called SP/1, SP/2, ..., SP/8. Each subset introduces new language features while retaining features introduced in earlier subsets. A second purpose of this report was to describe the compiler developed to support these PL/I subsets. Versions of the SP/k compiler that run on either the IBM 360/370 computers or the DEC PDP-11 computers are available from the university’s Computer Systems Research Group. (This report supersedes the report that described the first six subsets.)

R76-298—Tsichritzis, D., “Features of a Conceptual Schema” (31 pp., University of Toronto, Toronto, Canada)

A conceptual schema has often been proposed as a common basis for support of many data base views. We outline a set of proposed functions and facilities which can support hierarchical, network, and relational organizations of data. Common integrity constraints for the data base can be enforced at this level irrespective of user’s view.


This paper describes an adaptable, modular data acquisition system suitable for the collection of experimental data in a form readily adapted to subsequent digital computer entry and analysis. The system utilizes voltage-to-frequency conversion of an input analog signal over predetermined timing intervals. The digitized data is initially stored in shift-registers and later digitally formatted and digitally tape-recorded. The recorded data is read by a complimentary tape reading system and entered into a minicomputer for subsequent analysis and long-term storage. The digital formatting technique permits unambiguous definition and recovery of each data word as the tape is read. Adaptations of this system necessary to accommodate variations in the length of each data-word and the number of words per message block are discussed. Commerically available TTL integrated circuits and modular components are used in the design of this system.

R76-300—Russell, David L., “State Restoration Among Communicating Processes” (74 pp., Stanford University, Stanford, California)

In systems of asynchronous processes using message lists with SEND-RECEIVE primitives for interprocess communication, recovery primitives are defined to perform state restoration: MARK saves a particular point in the execution of the program; RESTORE resets the system state to an earlier point (saved by MARK); and PURGE discards redundant information when it is no longer needed for possible state restoration. Errors may be propagated through the system, requiring state restoration also to be propagated. Different types of propagation of state restoration are identified. Data structures and procedures are proposed that implement recovery primitives. In ill-structured systems the domino effect can occur, resulting in a catastrophic avalanche of backup activity and causing many message list operations to be undone. Sufficient conditions are developed for a system to be domino-free. Explicit bounds on the amount of unnecessary restoration are determined for certain classes of systems, including producer-consumer systems, k-producer systems (cyclic systems of k producers and k message lists), and MRS systems (where the sequence of recovery primitives is described by the regular expression (MARK; RECEIVE*; SEND*)).


The SAM76 language combines into a single interpretable processor characteristics of twoapps an general purpose macro generators and one (or more) infix operator mathematical systems.

Order by R-number. Use the Repository order form on page 108.

R76-302—Hansen, Eldon and Patrick Mérrel, “Estimating the Multiplicity of a Root” (84 pp., Duke University, Durham, North Carolina)

One point iteration functions for approximating roots of a function of a single variable and sequences generated by them are considered. Methods of estimating the multiplicity of the root are given which such sequences are converging are given. In particular, multiplicity estimators are given for Newton’s method, the modified Newton’s method, and Laguerre’s method.

R76-303—Liu, Peter C. and Robert C. Geldmacher, “A Q(n) Graph Reducibility Algorithm Using Depth-First Search” (16 pp., Stevens Institute of Technology, Hoboken, New Jersey)

An algorithm is presented for determining reducibility of a graph. A reducible graph is defined as one with the property that reduction of a dangling edge or a loop into a single vertex and reduction of a set of series edges into a single edge followed by the reduction of consequent parallel edges into a single edge, and so on, results ultimately in a single vertex. The algorithm has time and space complexities proportional to the number of vertices of the graph and is based upon establishing an order in which the graph contains a subgraph homomorphic to Kk. The algorithm consists of two depth-first searches and a radix sort.

R76-304—Luccio, Fabrizio and Linda Pagli, “Rebalancing Heights Balanced Trees” (30 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-rebalancing). Properties of k-rebalancing 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.

R76-305—Srin, Vason P., “Test for Adjacent Pattern Interference Fault in RAM Chips” (10 pp., Virginia Polytechnic Institute and State University, Blacksburg, Virginia)

The change in the content of a storage element in a RAM chip when only the contents of its neighbors are altered is stuck-at-1, stuck-at-0, or multifault (API-fault). The detection of API-fault when the neighborhood is restricted to four storage elements, one in each sense in the dimensional directions, is considered. A set of patterns which assigns every element of [0,1] to every storage element and its four neighbors detects the static part of API-fault. It is shown that by using 32 patterns, which is the minimal number, the above fault can be detected. An efficient method for generating the test patterns is also shown.

R76-306—Knaizuk, John, Jr., and Carlos R. P. Hartmann, “An Optimal Algorithm for Testing Stuck-at Faults in Random Access Memories” (15 pp., Syracuse University, Syracuse, New York)

This paper presents an optimum algorithm to detect any single stuck-at-1, stuck-at-0 fault and any combination of stuck-at-1, stuck-at-0 multifaults in a RAM using only the n-bit memory address register input and m-bit memory buffer register input and output lines. It is shown that this algorithm requires 4 × 2n memory accesses.

R76-307—Pavlidis, Theodosios, “Polynomial Approximations by Newton’s Method” (26 pp., Princeton University, Princeton, New Jersey)

The problem of locating optimally the breakpoints in a continuous piecewise linear approximation is examined. The integral square error E of the approximation is used as the cost function. Its first and second derivatives are evaluated, and this allows the application of Newton’s method for solving the problem. Initialization is performed with the help of the split-and-merge method. The evaluation of the derivatives is performed for both waveforms and contours. Examples of implementation of both cases are shown.

R76-308—Neuhauser, Charles, “Functional Description of the EMMY Main Memory System” (10 pp., Stanford University, Stanford, California)

This document gives the functional description of an emulation-oriented main memory system for use on the EMMY
bus system. The main memory system consists of a byte-addressable core memory system and a memory controller which performs elementary transformations on address and data under CPU control.

R75-309—Neuhauser, Charles, “An Emulation Oriented, Dynamic Microprogrammable Processor (Version 3)” (68 pp., Stanford University, Stanford, California)

This report describes the CPU of the Stanford Emulation Laboratory, known as the EMMY system. The EMMY CPU is a 32-bit microprogrammable processor designed specifically for the task of emulation research. The control store is dynamic, that is, it is writable by the CPU and thus serves for data storage as well as for microinstruction storage. Previous reports provided a design specification; this report describes the system as it is now implemented. Specifically, this report provides an EMMY system user with the basic information necessary to microprogram the EMMY CPU and to design hardware and software interfaces to the system bus.

R75-310—Hoelw, Lee W. and Walter A. Wallach, Jr., “Proposed Enhancements to EMMY” (8 pp., Stanford University, Stanford, California)

Current and projected versions of the Stanford EMMY are outlined. A number of changes to the organization of EMMY are suggested, based on work done to date. The changes improve the utility of the processor as an emulation vehicle.

R75-311—Neuhauser, Charles, “EMMY System Peripherals—Principles of Operation” (44 pp., Stanford University, Stanford, California)

Bus system peripheral units currently available in the EMMY laboratory are discussed from a functional point of view. The main memory system consists of a core memory system and an associated memory controller designed to provide elementary data transformations useful in the emulation environment. The Data-point interface provides a means of communicating between the 8-bit Datapoint 2200 bus system and the 32-bit EMMY bus; direct status indication and control of the CPU are also available. The maintenance console provides the user with a direct display of the information being handled by the EMMY bus system; information may be “trapped” and held for examination when a user select event occurs on the bus system.

R75-312—Hoelw, Lee W. and Walter A. Wallach, Jr., “Emulation Oriented Software First Development” (17 pp., Stanford University, Stanford, California)

“Software first” is the design philosophy whereby applications software is developed to solve specific problems prior to the availability of application hardware. We propose the use of an interpretive computing facility, designed around a high performance microprogrammable host machine, to support and enhance software first in the following manner: (1) Applications programs are initially converted into a high-level intermediate text (DEL) by a straightforward “one-plus” pass compiler. The intermediate text so generated is executed interactively via a microcoded interpreter. (2) The intermediate text surrogate for applications programs, having been verified by interactive debugging, are then processed by a simple generator to produce applications hardware compatible code. This “hard” code is then checked out on the development system by redefining the microcode running in the host machine so that it becomes an image of the projected applications hardware. Advantages of this approach, as compared to the conventional approach, accrue from the directness with which the source language and applications hardware are mapped into the development facility.

R75-313—Hoelw, Lee W. and Walter A. Wallach, Jr., “A Tale of Three Emulators” (17 pp., Stanford University, Stanford, California)

This is a preliminary report on the development of emulator code for the Stanford EMMY. Emulation is introduced as an interpretive computing technique. Various classes of emulation and their correlation to the image machine are presented. Functional and structural overviews of several emulators for the Stanford EMMY are presented. These are IBM System/360, CRIL, and DELTRAN. Performance estimates are included for each of these systems.

R75-314—Flynn, Michael J., Lee W. Hoelw, and Charles J. Neuhauser, “The Stanford Emulation Laboratory” (43 pp., Stanford University, Stanford, California)

The Stanford Emulation Laboratory is designed to support general research in the area of emulation. Central to the laboratory is a universal host machine, the EMMY, which has been designed specifically to support high efficiency and cost effective host for a wide range of target machine architectures. Microstore in the EMMY is dynamically microprogrammable and thus is used as the primary data storage resource of the emulator. Other laboratory equipment includes a reconfigurable main memory system and an independent control processor to monitor emulation experiments. Laboratory software, including two microassemblers, is briefly described. Three laboratory applications are described: (1) a conventional target machine emulation (a system 360), (2) “microscopic” examination of emulated target machine I-streams, and (3) direct execution of a high level language (FORTRAN II).

R75-315—Rau, B. Ramakrishna, “An ‘Almost-Exact’ Solution to the N-Processor, M-Memory Bandwidth Problem” (29 pp., Stanford University, Stanford, California)

A closed-form expression is derived for the memory bandwidth obtained when N processors are permitted to generate requests to M memory modules. Use of generating functions is made, in a rather unusual fashion, to obtain this expression. The one approximation involved is shown to result in only a very small error—and that, too, only for small values of M and N. This expression, which is asymptotically exact, is shown to be more accurate than existing closed-form solutions.

Lastly, a family of asymptotically exact solutions are presented which are easier to evaluate than is the first one. Although these expressions are less accurate than the previously derived closed-form solution, they are, nevertheless, better than existing solutions. This family of solutions is shown to include a couple of existing solutions.

R76-316—Mori, R., F. Roese, and P. Schmidlin, “Pattern Recognition in Noisy Pictures” (23 pp., IBM Zurich Research Laboratory, Zurich, Switzerland)

A method for processing noisy digital images using factor analysis is proposed. The method leads to separation of noise and true object structure. The image is reconstructed by a subset of orthogonal factors which are calculated by means of an iterative procedure. A criterion for optimal image restoration is given in terms of an X' test.

R76-317—Milgram, David L., “Adaptive Techniques for Photomosaicking” (16 pp., University of Maryland College Park, Maryland)

A previous report described techniques for creating photomosaics: first, the two overlapping images are brought into register; second, a seam from the top of the overlap region to its bottom is tracked one row at a time; finally, the resultant artificial edge introduced by the seam is smoothed by a ramp function. This report describes improved methods for grayscale registration and for choosing a best seam path with specified endpoints using dynamic programmig.

R76-318—Tateoka, Hitoshi and Tomozo Furukawa, “A Ridge-Sensitive Nonlinear Laplacian Operator” (9 pp., Hokkaido University, Sapporo-shi, Japan)

A nonlinear Laplacian operator with 3 X 3 mesh window is proposed to extract the ridge line from the digitized picture. The operator shows better performance that enhances the ridge line and suppresses noise compared with the general Laplacian operator.

R76-319—Wang, Paul P. and Richard C. Burns, “Classification and Machine Recognition of Severe Weather Patterns” (23 pp., Duke University, Durham, North Carolina)

Atmospheric activity, electromagnetic fields radiated from electrical disturbances in the atmosphere, increases during severe weather conditions. The atmospheric rate data are that data resulting from the measurement of the rate of occurrence of these atmospheres. These time-varying patterns represent various stages of development in the life cycle of a
severe storm. In this paper, the computerized severe weather forecasting and detecting system is characterized as a pattern recognition model, adopting complementarity decision-theoretical and mathematical linguistic methodologies. (For a longer version of this title, see R76-129, Computer, July 1976, p. 77.)

R76-320—Pohl, Ira, "Minimize Optimality in Sorting Algorithms" (16 pp., University of California, Santa Cruz, California)

We prove that an algorithm for selecting both the minimum and maximum element from an unordered set is minimally optimal. This algorithm had already been shown (Pohl 1972) to be minimally optimal. The circumstances in which the same algorithm is optimum for both norms leads to a conjecture relating both norms. The method of proof utilizes a new idea in the theory of sorting optimality, namely degree information of the Hasse digraphs.

R76-321—Wessellkamper, T.C., "Divided Difference Methods for Finite Fields" (23 pp., Virginia Polytechnic Institute and State University, Blacksburg, Virginia)

An alternative is provided to a recently published method of Benjaudrit and Reed for calculating the coefficients of the polynomial expansion of a given function. The method herein is an adaptation to finite fields of a method of Newton. The method is exhibited for functions of one and two variables. The relative advantages and disadvantages of the two methods are discussed. Finally, some empirical results are given.

R76-322—Agui, Takesi and Masayuki Nakajima, "Picture Processing by Shape Countour Functions" (25 pp., Tokyo Institute of Technology, Yokohama, Japan)

In this paper we describe a one-variable argument function defined over the boundary contour of arbitrary figures and describe its properties and applications in pattern recognition. This argument function, transforming planar closed curve into a function of one-variable, has many features for recognition of figures. First, this is invariant under parallel-translation and magnification operations as well as under the change of the starting point defined on the boundary of figures. Second, by this function we can easily detect and measure the angle of a rotation operation performed on input figures.

R76-323—Stewart, William J., "MARCA: Markov Chain Analyzer" (130 pp., Universite de Rennes, France)

This report presents a software package for the numerical analysis of Markovian models. The technique upon which the analysis is based is that of lumped simultaneous iteration—a new method which produces the correct result even under extremely difficult circumstances. A further advantage of the package is the ease with which it may be used, since only the minimum amount of information concerning the basic configuration and behaviour of the model is required. Two rather different examples are analyzed in detail.

R76-324—Fernandez, Eduardo, Rita C. Summers, Tomas Lang, and Charles D. Coleman, "Architectural Support for System Protection and Data Base Security" (38 pp., IBM, Los Angeles, California)

A set of architectural extensions to a machine of the type of IBM System/370 is proposed. The proposal involves hardware/software interaction to constrain the execution-time behavior of application and higher authority programs. The extensions consist of the addition of a new state to the previous supervisor and problem states, enforcement of discipline in transition between states, hardware distinction of five information types, and a mechanism to ensure correct I/O data transfers. Application of the extensions to a shared data base system shows that the protection of the operating system is enhanced significantly, with respect to errors or deliberate attacks, and that a high level of protection against unauthorized access can be obtained for the data base itself, without excessive cost.

R76-325—Wallach, Walter, A., Jr., "System/360 Emulator Performance Estimate" (14 pp., Stanford University, Stanford, California)

This note describes the performance and instruction timing of the System/360 emulator for EMMY. The Stanford EMMY will emulate typical 360 instruction streams at about 97KIPS. A production (Model II Control Store) EMXY will achieve 143KIPS on the same instruction stream. A 360 Model 50 processes this stream at about 141KIPS. Minor modifications to the Stanford machine should enable it to achieve 120KIPS.

R76-326—Wallach, Walter, A., Jr., "EMMYXL User's Guide" (40 pp., Stanford University, Stanford, California)

This document is intended as a guide to the use of EMXYXL, the expression-oriented line-by-line assembler developed by Hedges for the Stanford Emulation Lab. It is intended to be used along with two other technical notes from Stanford's Digital Systems Laboratory: "Principles of Operation for the Stanford EMXY" and "EMMY/360 Assembler.", Various IBM OS/370 and VSII documents may also prove useful.

R76-327—Wallach, Walter, A., Jr., "EMMY/Unibus Interface—Preliminary Specification" (17 pp., Stanford University, Stanford, California)

An interface unit to allow communication between the EMXY system bus and a DEC PDP-11 Unibus is described. The interface is to the functionally independent elements—an EMXY master/Unibus slave unit, a Unibus master/EMMY slave unit, and a memory management unit.

R76-328—Wallach, Walter, A., Jr., "Virtual Addressing for the EMMY/360" (16 pp., Stanford University, Stanford, California)

A virtual addressing technique for expanding the address space of the EMXY/360 2 million bytes. The scheme uses a translation table of 512 words of EMXY control store, addressed using bits 20:12 of the logical (virtual) program address. The page table entry reflects status of the referenced page, i.e., available or written. The semantic pointer for each instruction controls whether a logical address is to be translated or the execution routine is to use the logical address.

R76-329—Wallach, Walter A., Jr., "High Performance Emulation" (18 pp., Stanford University, Stanford, California)

The Stanford EMXY is examined as an emulation engine. Using the 360 emulator and the DELtran interpreter as examples, the performance of the current EMXY architecture is examined as a high performance emulation vehicle. The problems of using a sequential, vertically organized processor for high speed emulation are developed and discussed. A flexible control structure for high speed emulation studies is derived from an existing high performance processor. This structure issues a stream of microinstructions to a central command bus, allowing user-defined execution resources to execute them in overlapped fashion. The execution resources may be added or deleted with little or no processor rewiring.

R76-330—Wallach, Walter A., Jr., "EMMY/360 Functional Characteristics" (28 pp., Stanford University, Stanford, California)

An emulation of the IBM System/360 architecture is presented—the EMXY/360. Problem state code which executes correctly on an IBM 360 will also execute correctly on the EMXY/360. Code producing execution exceptions will, in most cases, produce the same results on the two systems. Certain exceptions occurring on IBM 360 cannot occur on the EMXY/360, such as address specification exceptions for main store operands, and certain precise interrupts on IBM 360 will be imprecise on the EMXY/360, such as address exceptions. The EMXY/360 supports the standard 360 instruction set with single precision floating point. The 360 input/output structure is not supported; I/O on the EMXY system is done by function call instruction, rather than by channel program and start-test I/O.

R76-331—Deguchi, Koichiro and Iwao Inamshito, "Texture Characterization and Texture-Based Image Analysis: Using the Two-Dimensional Linear Estimation Techniques" (43 pp., University of Tokyo, Tokyo, Japan)

A new approach to the problem of texture characterization and texture-based image partitioning is presented. The scheme approaches the gray level of a pixel as estimated from a linearly weighted sum of gray levels of its neighbor pixels. The weight coefficients are determined so that
the mean square estimation error is mini-
mized. With such a two-dimensional linear
estimator a class of textures are charac-
terized. Two procedures based on this tex-
ture characterization are developed for
texture-based image partitioning; the
first procedure is applicable to the case
where the presentation of an image forms
one region and the remainder is only a
small fraction of the whole image, and
the second procedure can be used even when
an image consists of several regions of
comparable sizes.

R76-332—Wilken, Edward J., “Realiza-
tions of Sequential Machines Using
Random Access Memory” (68 pp., Rut-
gers University, New Brunswick, New
Jersey)

Modern large scale integration tech-
niques, microcomputers, and programming
techniques have made the classical reali-
zations of sequential machines obsolete.
The recurrences all occur in environments
where random access memory, whether read-only or
writable, is an extremely cost effective
device. This paper presents a realization of a sequential machine which preserves
the multipport branching characteristic of
a sequential machine. A structure theory
and a design technique are presented
which allow an optimal memory size reali-
zation to be found.

R76-333—Ben-Bassat, Moshe, “Myopic
Policies in Sequential Classification”
(28 pp., University of Southern California,
Los Angeles, California)

Several rules for feature selection in
myopic policy are examined for selecting
the sequential finite classification prob-
lem with conditionally independent
binary features. The main finding is
that no rule is consistently superior to the
others. Likewise no specific strategy for
the alternation of rules seems to be sig-
ificantly more efficient.

R76-334—Maritas, D.G., A.C. Avrillias,
and A. Bounas, “Phase Shift Analysis
of Linear Feedback Shift Register Struc-
tures Generating Pseudorandom Se-
quences” (33 pp., NRC “Democrates,”
Computing Center, Athens, Greece)

We propose a systematic way for ana-
lyzing split-up feedback shift register
structures which generate parallel p-n
sequences. The analysis aims towards de-
termining the relative phase shifts among
the various realizations of the same p-n
sequence. The main design criterion is
derived when such systems are used as
pseudorandom number generators. The
results of this analysis are incorporated
in the procedure for constructing pseudo-
random number sequences. A case study
is presented to demonstrate the construc-
tion procedure, and in the light of our
analysis a criteria for phase shift is given of
a class of generators proposed by W.J.
Hurd. Two polynomials of degree n=20
contained in the table reported by W.J.
Hurd are checked as being non-primitive.
The study aims towards “safe” designs
where the greatest part of the machine
is given of a class of generators proposed by W.J.
Hurd. Two polynomials of degree n=20
contained in the table reported by W.J.
Hurd are checked as being non-primitive.
The study aims towards “safe” designs
where the greatest part of the machine
is given of a class of generators proposed by W.J.
Hurd. Two polynomials of degree n=20
contained in the table reported by W.J.
Hurd are checked as being non-primitive.
The study aims towards “safe” designs
where the greatest part of the machine
is given of a class of generators proposed by W.J.
Hurd. Two polynomials of degree n=20
contained in the table reported by W.J.
Hurd are checked as being non-primitive.
The study aims towards “safe” designs
where the greatest part of the machine
is given of a class of generators proposed by W.J.
Hurd. Two polynomials of degree n=20
contained in the table reported by W.J.
Hurd are checked as being non-primitive.
The study aims towards “safe” designs
where the greatest part of the machine
is given of a class of generators proposed by W.J.
Hurd. Two polynomials of degree n=20
contained in the table reported by W.J.
Hurd are checked as being non-primitive.
The study aims towards “safe” designs
where the greatest part of the machine
is given of a class of generators proposed by W.J.
Hurd. Two polynomials of degree n=20
contained in the table reported by W.J.
Hurd are checked as being non-primitive.
The study aims towards “safe” designs
where the greatest part of the machine
is given of a class of generators proposed by W.J.
Hurd. Two polynomials of degree n=20
contained in the table reported by W.J.
Hurd are checked as being non-primitive.
The study aims towards “safe” designs
where the greatest part of the machine
is given of a class of generators proposed by W.J.
Hurd. Two polynomials of degree n=20
contained in the table reported by W.J.
Hurd are checked as being non-primitive.
The study aims towards “safe” designs
where the greatest part of the machine
is given of a class of generators proposed by W.J.
Hurd. Two polynomials of degree n=20
contained in the table reported by W.J.
Hurd are checked as being non-primitive.
The study aims towards “safe” designs
where the greatest part of the machine
is given of a class of generators proposed by W.J.
Hurd. Two polynomials of degree n=20
contained in the table reported by W.J.
Hurd are checked as being non-primitive.
The study aims towards “safe” designs
where the greatest part of the machine
is given of a class of generators proposed by W.J.
Hurd. Two polynomials of degree n=20
contained in the table reported by W.J.
Hurd are checked as being non-primitive.
The study aims towards “safe” designs
where the greatest part of the machine
is given of a class of generators proposed by W.J.
Hurd. Two polynomials of degree n=20
contained in the table reported by W.J.
Hurd are checked as being non-primitive.
The study aims towards “safe” designs
where the greatest part of the machine
is given of a class of generators proposed by W.J.
Hurd. Two polynomials of degree n=20
contained in the table reported by W.J.
Hurd are checked as being non-primitive.
The study aims towards “safe” designs
where the greatest part of the machine
is given of a class of generators proposed by W.J.
Hurd. Two polynomials of degree n=20
contained in the table reported by W.J.
Hurd are checked as being non-primitive.
The study aims towards “safe” designs
where the greatest part of the machine
is given of a class of generators proposed by W.J.
Hurd. Two polynomials of degree n=20
contained in the table reported by W.J.
Hurd are checked as being non-primitive.
University of California Santa Barbara: Regular faculty openings in the Computer Science Program are anticipated beginning Fall 1977. Expertise in programming languages, operating systems, information storage and retrieval/data base systems, or architecture is desirable. Excellence in research and graduate and undergraduate teaching, consistent with the candidate’s professional level, is required. Rank and salary are dependent on qualifications. Candidates should send resumes and names of four references to Computer Science Department, Santa Barbara, CA 93106, by February 15, 1977. An Equal Opportunity/Affirmative Action Employer.

University of Illinois at Chicago Circle Department of Information Engineering: Applications are invited for an Assistant or Associate Professor position for Spring/Fall 1977. Applicants should have a Ph.D. in computer science and demonstrated research capabilities in one of the following areas: information systems, computer vision, software engineering or operating systems. Send resume and selected publications to: Professor Bruce H. McCormick, Head Department of Information Engineering University of Illinois at Chicago Circle Box 4348, Chicago, Ill. 60680

An equal opportunity and affirmative action employer.

Dean, School of Advanced Technology which is a graduate school offering M.S. and Ph.D. Emphasis in Computer Systems, Applied Management Systems, General Systems and Systems Sciences/Technology. School provides continuing education for working professionals in local industries. Dean is expected to provide leadership role in strengthening interdisciplinary programs and relationships with other units on campus. A broadened role of School by merger with Department of Math Sciences is under consideration. Applicants should ideally have continued research interests, a strong background in Computer Sciences and demonstrated success in obtaining research funding. Send resumes and nominations by December 15, 1976 to: Professor Narendra Goel, AD 229, SUNY at Binghamton, Binghamton, New York 13901. An equal opportunity/affirmative action employer.

Our East Coast-based clients are seeking highly motivated BSEE's with 2 to 10 years analog/digital design and software experience. Send background information in confidence to Career Advisors, 125 Wolfe Rd., Suite 408, Albany, NY 12205.

Faculty Positions in Computer Science: The expanding Computer Science department at Wayne State University has several assistant professorships available. A Ph.D. degree is required. Applications begin with the Fall Quarter, September, 1977. Applications are welcome from those interested in teaching graduate and undergraduate courses and pursuing research in any of the following areas of Computer Science: Artificial Intelligence, Computer Architecture, Data Base Management, Discrete or Continuous Simulation, Operating Systems, Pattern Recognition, Programming Languages, Systems Analysis.

Send letter of interest and resume to: Professor Seymour J. Wolfson Computer Science Section Wayne State University Detroit, MI 48202 or call (313) 577-2477.

Wayne State University is an affirmative action equal opportunity employer.

Ph.D. (77 expected), computer hardware, control theory. Computer software as minor. Four years experience designing and repairing all analog and digital circuits. Research in developing a centrifugal analyzer system and control mechanism using minicomputer, PDP-8 in real time mode. Yong H. Lee, 483 Jones Tower, 101 Curl Drive, Columbus, OH 43210.

Faculty Positions in Computer Science, University of Minnesota: The University of Minnesota invites applications for faculty positions as assistant (or associate) professors of Computer Science available January 1, 1977. Applicants should have the Ph.D. degree and be qualified to teach and perform research. Applicants with specialties in computer architecture (including micro-programming and microcomputers), operating systems, or operating systems with emphasis to data base management aspects are encouraged to apply. Applicants with operating systems specialties are expected to have theoretical as well as practical experience.

Forward resume together with names of at least three references to W. K. Giloi, Chairman Faculty Search Committee, Department of Computer Science, 114 Lind Hall, 207 Church Street S.E., Minneapolis, Minnesota 55455. An equal opportunity/affirmative action employer.

University of Illinois at Chicago Circle Department of Information Engineering: Applications are invited for an Assistant or Associate Professor position for Spring/Fall 1977. Applicants should have a Ph.D. in computer science and demonstrated research capabilities in one of the following areas: information systems, computer vision, software engineering or operating systems. Send resume and selected publications to: Professor Bruce H. McCormick, Head Department of Information Engineering University of Illinois at Chicago Circle Box 4348, Chicago, Ill. 60680

An equal opportunity and affirmative action employer.

Dean, School of Advanced Technology which is a graduate school offering M.S. and Ph.D. Emphasis in Computer Systems, Applied Management Systems, General Systems and Systems Sciences/Technology. School provides continuing education for working professionals in local industries. Dean is expected to provide leadership role in strengthening interdisciplinary programs and relationships with other units on campus. A broadened role of School by merger with Department of Math Sciences is under consideration. Applicants should ideally have continued research interests, a strong background in Computer Sciences and demonstrated success in obtaining research funding. Send resumes and nominations by December 15, 1976 to: Professor Narendra Goel, AD 229, SUNY at Binghamton, Binghamton, New York 13901. An equal opportunity/affirmative action employer.

Our East Coast-based clients are seeking highly motivated BSEE's with 2 to 10 years analog/digital design and software experience. Send background information in confidence to Career Advisors, 125 Wolfe Rd., Suite 408, Albany, NY 12205.

Assistant Professors: The University of Iowa has openings at the assistant professor level in the Division of Information Engineering. Candidates would teach undergraduate and graduate level courses in the Electrical Engineering Program and conduct research in the following areas: computer architecture, software and hardware; computer communications; computation theory; mini and microcomputer systems; operating systems. There are modern facilities within the Division for research in these areas and close relationships exist with the College of Medicine and with Physics and Mathematics.

Send resume by January 15 to: George M. Lance, Associate Dean, College of Engineering, The University of Iowa, Iowa City, Iowa 52242. The University of Iowa is an equal opportunity employer.