A Digital System Modeling Philosophy and Design Language

Mehmet B. Baray*, Stephen Y. H. Su**

*Middle East Technical University, Ankara, Turkey
**University of California, Berkeley, California

ABSTRACT

With the increasing complexity of digital systems, a new approach to system decomposition and modelling must be developed. Such an approach must not only permit a system to be described in a modular fashion, but the basic system modules must be able to be decomposed to allow a precise description of the module function. In this paper, a modelling philosophy and language are presented which permit the system designer to: (1) specify the system design in a hierarchical, modular fashion; (2) describe the function of each module by independently specifying the structure and the control function of the module; (3) build arbitrarily complex structures and control functions through the use of macro-like definitions; and (4) handle the problem of specifying complex concurrent control functions in a straightforward and unambiguous manner. Examples are used throughout the paper to clarify the semantics of the language. A complete unit design and a Backus Normal Form (BNF) specification of the language syntax are presented.

REFERENCES

The Structure and Operation of a Design Language Compatible Simulator

Memhet B. Baray, Stephen Y. H. Su, Robert L. Carberry
University of California Berkeley, California

ABSTRACT
Logic design language provides a powerful tool for digital system specification. When coupled with automated logic design techniques the result is an extremely efficient mechanism for system realization. However, before any design process can be considered complete, the design integrity must be verified and some measure of system performance must be obtained. It is the function of the design language simulator to provide the capability for design verification and performance evaluation. This paper describes the structure and operation of a simulator compatible with the design language of Baray and Su. The overall structure of the simulator is discussed and the data base required by the simulator is specified. An algorithm is presented for extracting the data base information from the design language specification. Finally, the simulator run time environment is examined closely so as to identify the mechanisms which give rise to efficient operation in terms of both execution time and memory space requirements.

REFERENCES
A System Modeling Language Translator

Stephen Y. H. Su, Mehmet B. Baray, Robert L. Carberry
University of California, Berkeley, California

ABSTRACT
The structure of the design automation language is examined and the sections of a system specification which can be translated independently of one another are identified. The ability to do incremental translation of changes in a system specification is demonstrated. The tables required for translator operation are defined and an algorithm for generating the required tables from a design language system specification is presented. The procedure for minimization of the system specification prior to logic circuit realization and a method for realization of the various sections of the system specification are described.

REFERENCES
A Summary of Architectural Involvement With Computers

C. James Olsten
California State Polytechnic College, San Luis Obispo, California

Abstract
The architectural profession is generally considered by its members to be engaged in the practice of an art and as such has resisted for a time the application of computer aids to the design process. This process is concerned with the organization and specification of the built environment which houses man for work or pleasure. For those in architecture familiar with the computer, the pace of the profession's involvement has been slow. This paper covers some of the past developments that have produced the current state of the art. Questions concerning the methods and costs of utilizing technology in design are now being answered. Conferences and workshops such as this one are restructing the profession's concern; how can the designer integrate available techniques into his personal design approach? This is the major question remaining for architecture.

Bibliography
The Future of Computer Applications in the Architectural Profession

Gifford H. Albright
The Pennsylvania State University

A commentary on the future of computer application in architecture.

Paper not received in time for publication.
Some Computer - Aided Approaches to Housing

Anthony Schnarsky
University of Wisconsin, Milwaukee

THE NEXT LEVITT TOWN
It is happening. The fragmentation in the building industry is beginning to congeal. The mobile home industry has grown from 60,000 units per year in 1947 to over 400,000 units per year in 1970. National Homes Corporation produced 28,000 houses in 1970, a bad year in building. Levitt Technology is building an automated plant to produce houses. The concept of the house as a product of industrialized processes is a reality……

REFERENCES
[7.] Miller, "Computer-Aided Space Planning", a working paper for A. C. Martin and Assoc., Los Angeles, 1969
[8.] Rapoport, Amos, House Form and Culture, Prentice-Hall, 1969
[9.] Alexander, Hirshen, Ishikawa, Coffin, Angel, Houses Generated by Patterns, Center for Environmental Structure, Berkeley, 1970
Micro-programming has been defined as an orderly approach to the design of a control section of a computer using control signals arranged in fixed-length words. The control section is the part of a computer which controls the activities of the memories, the central processing unit, the arithmetic unit and the peripheral units. The most elementary operation is called a micro-operation. Such an operation could be a comparison of two registers or a register to register transfer. Some micro-operations have implicit operands which always are used. Others use operands which are expressed explicitly within the control word. Each control word, or micro-instruction, contains one or more micro-operations which are performed in a fixed time interval. The set of micro-instructions required to execute one machine language instruction is called a micro-program…
ABSTRACT
Critical timing races (hazards) are a potential problem in digital systems consisting of a large number of interchangeable modules and replaceable integrated circuits. These hazards can be minimized by careful design procedures and reviews but will still exist due to system complexities. Systems have been proposed for simulating digital networks based on a time-sequenced logical tracing, but these systems require much host computer time and input stimuli to thoroughly exercise the system. The basis of the proposed race analysis system is that all critical races result in an improper flip-flop state. Each flip-flop input is traced, from logic element output to input, accumulating delays. Two basic algorithms are required and described to locate the racing of parallel paths on flip-flop inputs. This system requires minimal user inputs and can perform either a statistical or worst case race analysis.

REFERENCES
ABSTRACT

AIDS is an interactive computer aided design system developed principally for architects and designers. An integrated system consisting of both hardware and software, AIDS provides powerful graphics capabilities at a sufficiently low cost to make its use feasible in a commercial environment. Although AIDS is continuously being extended and refined, it is an operational system in daily use at the Boston design firm of Perry, Dean and Stewart, and not merely an experimental system or a research tool.

Although presented here as a complete interactive, graphic system for architects, it is in fact highly modular and its individual find uses in many other fields.
Image: An Interactive Computer System for Multi-Constrained Spatial Synthesis

Guy Weinzapfel, Timothy E. Johnson, John Perkins
Massachusetts Institute Of Technology

SYNOPSIS
IMAGE is a system of computer programs which generates three dimensional spatial arrangements to satisfy spatial requirements specified by a designer. A problem specification is composed of two elements: geometric descriptions of the spaces to be configured; and relationships between those spaces which the designer wishes to achieve. Given a problem description, IMAGE generates arrangements until a configuration is achieved which satisfies all specified relationships. If no such configuration is possible, IMAGE produces arrangements which minimize dissatisfaction of the specified relationships.

REFERENCES
Revisiting an Operational Graphic Design System

S. P. Krosner, W. H. Sass
IBM, Kingston, New York

In 1968 we described an experimental multitasking monitor for the IBM 1130 system. [1] In 1970 we described a logic design and analysis application using the multitask monitor as a base. [2] This application supports interactive graphic design (Figure 1) on the IBM 1130/2250 system. The emphasis in both of these papers was placed on what we had done. This paper will instead attempt to answer the question "how was it done?"

References
ALMS: Automated Logic Mapping System

R. L. Russo, P. K. Wolff, Sr.
IBM, Thomas J. Watson Research Center, Yorktown Heights, New York

ABSTRACT
ALMS is a set of design automation computer programs which accepts as input a description of a logic design, specifications of modules (e.g., chips, cards, etc.) into which the blocks of the design are to be partitioned or mapped, and some constraints that must be satisfied. It produces as output a documented assignment of the blocks to the modules satisfying the specified constraints. The system algorithms are presented, system features are discussed, program execution times are given and results are presented and compared to manual solutions for the same tasks. Three conclusions are reached. First is that computer programs make it possible to perform partitioning and mapping experiments which were not possible before. Second, for one-level partitions (e.g., logic gates on chips), highly automatic solutions obtained by the program are at least as good as manual solutions and are less costly to obtain. Third, for multi-level partitions (e.g., logic gates on chips on cards) or for mappings, the solutions obtained with the program are again at least as good as manual solutions; furthermore, ALMS allows a designer to try more alternatives than he could manually, so that he can trade-off the time and cost of trying additional alternatives against the value of a better solution.

REFERENCES
ABSTRACT
The concepts underlying a versatile tiered layout design scheme for MOS LSI logic devices is presented. A series of three computer-aided design programs is discussed which provide designers with information permitting rapid organization of minimum area layout designs.

REFERENCES
Automated Placement of Multi-Terminal Components

F. Taylor Scanlon
Honeywell Information Systems, Phoenix, Arizona

Abstract
An algorithm for placing multi-terminal electronic components in a printed circuit board environment is presented. A heuristic technique of applying vector forces to component models and iteratively relocating these modules has been developed. The methods used to compute these forces are what make this algorithm unique and improve its ability to find solutions that are independent of initial conditions and near optimal. A flexible modeling scheme has been devised to allow for any general class of components to be placed. The algorithm will place modules of arbitrary size with any number of terminals onto boards of irregular shape, including placement on two sides of a board. While most powerful in a printed circuit environment, the algorithm may be used in wire-wrap technologies and hybrid designs as well. The technique has been implemented in a Fortran program and exhibits the high degree of flexibility required in a design automation system. A summary of the results obtained in practical applications at Honeywell is included.

References
Wire Routing by Optimizing Channel Assignment within Large Apertures

Akihiro Hashimoto, James Stevens
University of Illinois, Urbana, Illinois

ABSTRACT
The purpose of this paper is to introduce a new wire routing method for two layer printed circuit boards. This technique has been developed at the University of Illinois Center for Advanced Computation and has been programmed in ALGOL for a B5500 computer. The routing method is based on the newly developed channel assignment algorithm and requires many via holes. The primary goals of the method are short execution time and high wireability. Actual design specifications for ILLIAC IV Control Unit boards have been used to test the feasibility of the routing technique. Tests have shown that this algorithm is very fast and can handle large boards.

REFERENCES
Introduction

One technique in space planning that has received a lot of attention in the last five years is the use of linear graphs for representing floor plans [Levin] [Casalaina] [Krejcirik] [Grason70a] [Grason70b] [Grason70c] [Teague]. The fact that this paper describes one such technique will probably provide its chief attraction for space planners. However, the paper is also intended as a case study in computer-implemented design. One of its purposes is to illustrate the relationship between the representation chosen for a design problem and the methods developed for solving that problem. This aspect will be introduced in more detail in the next section.

The paper treats computerized space planning by discussing methods for the solution of a formal class of floor plan design problems. These methods have been implemented in an experimental computer program called GRAMPA (for GRAph Manipulating PAcKage). The methods of solution depend on a special linear graph representation for floor plans called the dual graph representation and a pseudo-grammar called PGG for specifying these graphs.

This paper is a condensation of a much larger document [Grason]. Because of this, the primary emphasis will be placed on results, rather than the reasoning behind the results. The interested reader can consult the larger document for more detailed descriptions of the material presented.

Bibliography

INTRODUCTION
ACD is an approach for space allocation which uses the digital computer for analysis of large quantities of subjective data. It allows direct participation by an increased number of people in the solution generating process, and facilitates the investigation of a greater number of alternate spatial concepts than traditional procedures would normally permit…
The Automated Generation of Architectural Form

William J. Mitchell
University of California, Los Angeles

ABSTRACT
A systematic framework for discussion of automation of the solution of architectural problems is established, based on an examination of the concepts of solution representation, generation, and testing. Some of the more important implications of various different techniques and principles of representation, generation, and testing are then illustrated by a discussion of procedures for solution of some simple spatial arrangement games and puzzles. It is shown that when we attempt to write procedures for solution of the rather larger and more complex problems encountered in practical architectural design, we discover some quite severe limitations on our current ideas about architectural design automation. The major limitations are outlined, and progress made towards overcoming them recounted.

NOTES
[6] The game is manufactured by Parker Brothers, Inc., P.O. Box 900, Salem, Mass.


[26] Eastman, ibid.

[27] Eastman, ibid, and Grason, ibid.


[29] Similar problems at the urban design and planning scale are discussed in: Britton Harris, Computation is Not Enough, in Proceedings: Second World Congress of, Engineers and Architects, Tel-Aviv, December 1970.


[34] Thomas A. Markus, Optimization by Evaluation in the Appraisal of Buildings, in EDRA 2, ibid.


Introduction
This paper focuses on computer augmentation of some simple design tasks found in every architectural office - the design and contract specification of arrangements of objects and equipment within a space. Examples of such tasks include stairwells, restrooms, mechanical rooms, and kitchens. This type of design is not considered creative, for it consists of a variation of a standard design within a particular context. Rather, it is an example of the "dogwork" involved in realizing initial design conceptions. A major portion of an architectural firm's time and effort is currently expended on the production aspects of design, of translating schematic design into construction documents [1].

General Space Planner (FOOTNOTE: GSP is written in FORTRAN IV, is made up of approximately 1600 card images, and occupies about 16K words of core storage. Credit for a significant portion of the programming of this version of GSP goes to C. Yessios. It would not be in operation without his help.) (GSP) is an automated design, drafting, and problem solving system recently implemented on Carnegie-Mellon University's IBM 360/67 time-shared computer that begins to explore how arrangement problems and other production aspects of design can be either automated or augmented by the computer. This paper describes the current user interface of the GSP system, its required inputs, modes of interaction, and its outputs, and describes how GSP solves several classes of drafting design problems. It should be emphasized that the current version of GSP has been running for only a short time and its full capability is still being expanded. Also, GSP has not been developed as a production program for practical applications. Rather, it was developed to explore in depth many of the theoretical and practical problems inherent in the development of such systems. We expect later versions of GSP to have production efficiencies and characteristics.

In the following section is presented an overview of the GSP system, its general structure and modes of problem solving. Following the next section are more detailed descriptions of its techniques for representing objects, space, and design problems, and the methods it uses to solve certain arrangement problems. We conclude with several examples of the capabilities of GSP as an automated design and drafting system. This report is oriented toward future users of later versions of GSP and GSP-type systems, architects and designers. Technical details of the internal operation and structure of GSP have been presented elsewhere. [2]

Notes
[1] The breakdown of expenses recommended by the AIA are: schematic design 15%; design development 20%; construction documents 40%; review of bids 5%; and construction supervision 20%. See: AIA Handbook of Architectural Practice, the Octogon, Washington, D. C. In reality production of Construction Documents for architectural work averages about 45% and for engineering 71% of expenses. See: "The Economics of Architectural Practice" Report by the American Institute of Architects, Washington, D. C. 1968.
The spatial arrangement problem can be considered a scheduling problem involving the assignment of functions to areas of space over time. Little information is usually available concerning the scheduling of most temporary uses of space. Where such information is lacking, designers approximate scheduling by ignoring time as a dimension and assigning multiple temporary functions to a single space. The GSP overlap rules follow the same practice by allowing overlaps of Use Space.

The notation presented is the input language by which the user defines shapes to GSP. It translates these into equivalent representations in the form of variable domain arrays, as described in C. M. Eastman, "Representations for Space Planning", Communications of the Association for Computing Machinery, 13:4 (April, 1970) pp 242-250.

For a technical description of GSP's problem solving methods, see Eastman, op. cit (1971).


For a listing of all the formulas now used in Computing the Restrictiveness Ratings, see Eastman, op. cit. 1971.


INTRODUCTION
This paper describes RELATE, a computer program designed to assist the facilities design process. It is used to develop three-dimensional block plan layouts (form diagrams) economically and to help solve the "combinatorial" aspects of plan layouts.

Paper not received in time for publication
ABSTRACT
A unified procedure to find test patterns for detection and location of stuck-at-type single faults in the combinational circuits is described. A reduction algorithm to obtain a minimal set of detection patterns and a location algorithm to obtain complete location information obtainable by external observation; are presented. The circuit may consider AND, OR, NOT, NOR, NAND, EXCLUSIVE OR and LOGICAL EQUIVALENCE gates.

A basis for the article are the single and multi-dimensional path sensitization [1][2] and the graph theoretical approach for system diagnosis[3]. The main parts of this work is the construction of sensitization functions and the path analysis table, and the development of the reduction and location algorithms. The procedure is illustrated in detail by two examples.

REFERENCES
Application of a Logic Fault Analyzer to the Manufacture and Maintenance of the
Control Data 7600 Computer

Lionel C. Bening, Jr.
Control Data Corporation, Arden Hills, Minnesota

Summary
This paper describes the application of a sequential logic fault analyzer computer program to the
problem of logic circuit module tests for the CONTROL DATA[R] 7600 computer. A
description of the sequential fault analyzer is provided first. Next, the software system built
around this fault analyzer is outlined. The development of test sequences for 231 logic module
types used in the 7600 computer is considered. Block diagrams of the test fixtures used at the
manufacturing facility and to supplement field maintenance are provided and explained.
Preliminary results of the application of fault analyzer developed tests are reported.

References
SUMMARY
This presentation describes a method for generating functional test programs for Large Scale Integrated Circuits (LSI) and Logic Cards having hundreds of logic gates.

J. P. Roth's D-Algorithm, and S. Seshu's Logic Simulation Method have both been used extensively for generating logic circuit test patterns.

We have developed a more efficient method which incorporates both of these methods.

The new method solves diagnostic problems related to the logical circuits obtained by partitioning logic circuits making up the digital hardware.

In this presentation, we describe a test pattern generating method, and an Automatic Diagnostic Program Generator (ADP-3) using this method with a FACOM 230-60 computer.

References:
CLUSTR: A Program for Structuring Design Problems

Murray A. Milne
School of Architecture and Urban Planning, University of California at Los Angeles

Abstract
CLUSTR is a computer program which assists the designer in finding the structure inherent in his design problem. The designer supplies the list of elements which define the design problem, and then decides which of these elements are related. The computer decomposes the problem into subsets in which each element is related to every other element. In theory each of these subsets represents the smallest “structural” component of the problem: a coherent functional or behavioral sub-system. The most closely related subsets are then combined into larger clusters. This process continues until all clusters have been recomposed. The computer then draws a diagram to show how these subsets are combined to form the final problem structure. The computer also identifies the dominant elements at each node in the structure to assist the designer in finding the solutions to each sub-problem.

References
7. Others who have recently developed new computer based problem structuring algorithms include:
8. Alexander, Christopher, Ibid.
15. Ball, Geoffrey H., “Classification Analysis,” Office of Naval Research, Contract Number 4918 (00), SRI Number 5533 (lists over 500 methods).
17. For a discussion of form diagrams or patterns see:
The Three Dimensions of Architectural Design Automation

Harvey N. Lehrman
Martin Marietta Corporation, Orlando, FL

ABSTRACT

Now that the computer era has passed infancy, users of these machines are demanding that a better method be found for representing their output. Those using the computer as a design tool are not content with poring over page after page of uninteresting printout. Visual presentation has provided an answer to this problem, and plotters are now being used to present the designer’s thoughts as a series of pictures. These pictures readily present much more information than do many pages of printed output.

Instead of having to read columns of X, Y values, the plotter can present these values in graphical form as families of curves. These curves are automatically scaled by the computer and can be presented in various forms. Many items of interest can be read from these plots, such as curve intersections, slopes, ranges, etc.

For “real world” designs, capabilities can be provided for presenting outputs which are functions of more than two variables. To do this, perspective plotting routines can be developed which generate plots of three-dimensional objects on a two-dimensional surface. The user of these routines can specify the position of his eye in three-space and, in effect, view the object at any orientation and size. This provides him with information which he could never get from printed output.

In order to make perspective plots more meaningful, a technique can be developed which would, identify the lines which are hidden from view by other parts of the plot which are closer to the observer’s eye. Once identified, these lines could be eliminated from the plot so that the only lines plotted are those the observer would actually see. As the eye of the observer changes its position, some lines which were hidden come into view, and vice versa. The routines can keep track of the viewing of each line by checking to see if any surfaces are between the line and the observer’s eye. If there are any, the line is determined to be hidden.

With these tools for representing any object in perspective with the hidden lines removed, an architect could describe a structure he wished to build, and have the computer show it to him in perspective from any and all sides. By the use of “clipping”, just a portion of the structure could be “blown-up” and plotted. Also, the architect’s view as he walks through his structure could be calculated and displayed. With a library of standards available, the computer would be able to report on the amount of material needed to build the structure, as well as the amount of time and numbers of people needed.

Another feature of this system is the ability to calculate projection perimeters and areas. The amount of structure area exposed at some particular line of sight is an indication of the vulnerability of that structure. This area is a very important factor from a military point of view, also. By comparing the projected area of a target with the projected areas of the target’s most critical elements, a measure of vulnerability of the target can be obtained. By obtaining the areas projected from several selected viewing points, the expected vulnerability of the target can be calculated.
A Computer Simulation Approach To Elevator System Design

Alton J. Penz

ABSTRACT
Computer simulation methods enable the architect to accurately analyze the performance of specified elevator systems, thus facilitating more careful evaluation of alternative system designs than previously achievable. Simulation improves analysis because it involves replication of real-time elevator performance and does not rely on the rules of thumb and statistical assumptions incorporated into traditional analytical methods. The development of a computer simulation program for elevators requires caution to ascertain that the assumptions incorporated into the program are satisfactory and that the resulting program performs reliably and accurately. An acceptable simulation program, such as the one presented in this paper, facilitates a variety of research efforts encompassing sky lobby and express car layout strategies, dynamic peak control procedures, and cost/benefit studies of system design goals.

BIBLIOGRAPHY
Fault Detection and Diagnosis of Memory System Faults

A. R. Klayton, W. A. Barrett arid A. I. Larky
Lehigh University, Bethlehem, Pennsylvania

Introduction
Algorithms for the detection and diagnosis of faults in semiconductor random-access, word-organized memory systems are presented and evaluated. The memory system is conceptually partitioned into four major elements to facilitate analysis. During the testing of each element, previously tested elements are assumed to be fault-free, but untested elements may contain multiple faults. The problem of fault masking and interaction between the major elements of the system is investigated and considered in the development of the algorithms.

The four major elements of the memory system are: 1) a memory address register (MAR), 2) a decoder, 3) a two dimensional array of storage devices comprising the memory, and 4) a memory data register (MDR).

It is assumed that the memory system is generally inaccessible to test equipment other than through common control signals, power lines, the MAR and MDR. Thus the algorithms are applicable when it is not convenient or possible to access various parts of the system with test equipment, e.g., in systems fabricated using LSI techniques. This assumption complicates the testing problem due to the likelihood of fault masking between system elements.

The following analysis is not concerned with such problems as control signal timing and parameter tolerance, but rather with the detection and diagnosis of stuck-at 1 (SA1) and stuck-at 0 (SA0) solid faults within the system. It is useful to make a distinction between the concepts of a fault and a failure. Faults are logic gates, bit lines, etc., which are SA1 or SA0. Failures are deviations from correct system behavior which result from the presence of solid logic faults. For example, a gate SA1 in the decoder may result in the improper excitation of a decoder output line (memory word line). The decoder output malfunction is a failure. In general, a single fault may precipitate multiple failures.
Minimum Test Patterns for Residue Networks

D. C. Bossen, D. L. Ostapko, A. M. Patel, M. S. Schmookler
International Business Machines Corporation

Introduction
Residue networks are logic trees consisting of residue gates which calculate the modulo-m sum of two or more inputs. They are used most frequently for detection or correction of errors in digital operations. Binary or decimal arithmetic operations may be checked using modulo-3 networks as in Model 195 of System/360. Parity trees, consisting of modulo-2 gates, are frequently employed to detect errors in the transmission of digital data. This paper develops procedures for generating a minimum number of test patterns for detecting failures in residue networks. It extends the concepts introduced at the 1970 FJCC by Bossen, Ostapko, and Patel in their paper, "Optimum Test Patterns for Parity Trees." [1] It demonstrates easier methods for generating test pattern sequences and shows that they are applicable to all residue networks. The principal result of this paper is that for a single output residue network consisting of modulo-m gates having n or less inputs each, the required number of test patterns is \( m^{n} \), provided the gates are of a particular logical construction. This represents a minimum number of test patterns since that is exactly the number of test patterns required for a complete functional test of a single modulo-m gate with n inputs.

In the next section, an application of the method described in this paper is briefly illustrated. The properties of residue gates which are necessary and sufficient for the use of this method and various means for generating the test patterns are described in the remaining sections.

REFERENCES
INTRODUCTION
Louis C. Kingscott and Associates, Incorporated is an architectural-engineering firm with over forty years experience in the design of buildings and of their related mechanical and electrical systems. Since early 1967, LCK has utilized an IBM 1130 computing system to aid in the design process. This utilization, however, has not been limited to engineering calculations and problem-solving capabilities. Program modules, serving as management tools for project control, as well as various accounting systems, have also been developed. While the programs themselves are oriented to the unique problems and needs of LCK, the concept of a modular program system is applicable to organizations of many types and sizes…….
INTRODUCTION
This paper discusses an experimental integrated design system being developed at Cambridge University, aimed at exposing some principles of good practice in the construction of large ensembles of CAD programs. The paper discusses three aspects of this work: (1) the laying down of the philosophical foundation for the design of the Cad system, (2) system techniques used to implement these fundamental considerations, (3) the pursuit of a number of applications to evaluate the CAD system.

Paper not received in time for publication.
This paper describes a computer system to assist the logic hardware designer. It is interactive and graphical; it has been human engineered to be convenient for the designer. An elementary system has been implemented; plans for future expansions are described.
CIBOL - An Interactive Graphics Program Used in the Design of Printed Wiring Boards and Generation of Associated Artmasters

T. J. Kriewall, N. R. Miller

ABSTRACT

CIBOL (CIrcuit BOard Layout) is an application program that uses a computer graphics facility to aid in the design of low density discrete component printed wiring boards. There are three distinct program phases: creating the board outline, positioning the components, and routing the interconnections. A light pen and scope display are used interactively during the three program phases. The program directly generates manufacturing information on auxiliary peripherals in the form of parts lists, component layouts, drilling lists and artmasters. The use of the program has resulted in considerable savings in the cost of producing printed wiring board designs and reduction in the turnaround time from design inception to completion.

REFERENCES

ABSTRACT
The "On-Line Logical Simulation System" (OLLS) is a complete software package which permits a logic designer to interactively design, layout and simulate large digital systems using the IBM 2250 CRT. The user communicates with OLLS through a set of interactive displays using the light-pen, keyboard and programmed-function buttons. The major subsystems include file handling, device definition, drawing manipulation, simulation and input-output. OLLS is unique in its freedom of device definition, generality and adaptability. Its major contributions to the design process are fast, precise system documentation and accurate logic simulation.
INTRODUCTION

The Logic Design System (LDS) is a set of FORTRAN Computer programs that convert Boolean equations and component descriptions into logic diagrams, parts lists, board layouts, wiring lists, and diagnostic data. This system decreases the cost of digital design work and eliminates errors that could be introduced by manual methods. More important, it reduces lead time and frees the designer for creative tasks. It does not yet perform printed circuit routing or artwork generation, although these packages are presently under design.

The logic designer can use this tool without trying to describe his problem to a professional programmer. The system input is a free-form, problem oriented language that is readily learned by a designer, and is powerful enough to handle any design.

The Logic Design System has been constructed to avoid built in hardware obsolescence. The designer can select parts from a library of standard components and integrated circuits, or specify tailor-made modules consisting of integrated circuit configurations that may repeat many times in the same combination to suit the individual designer’s needs. The LDS is compatible with the use of both MSI/LSI logic elements.
**Summary**

Many programs for analyzing electronic circuits including nonlinear elements have been reported. It became difficult to evaluate circuits such as ICs by the breadboard method, because this method is not always suitable for simulation of high speed and high density circuits. Moreover lengthy experiments are involved. In the case of circuits composed of discrete parts, it was considered that many manual experiments could be replaced by computer aided analysis. We developed a general nonlinear network analysis program NONLISA to enable the use of medium scale computer system FACOM 270-30 for scientific and technical applications. This program is aimed mainly at the capability of circuit designers' easy treating and extending system functions without many modifications. Functions of NONLISA are DC analysis, transient analysis, sensitivity analysis, worst case analysis, Monte Carlo Simulation etc. for networks including nonlinear elements.

This paper deals with the structure and the functions of NONLISA.

**References**

INTRODUCTION
A data management facility has been designed to satisfy the specific needs of design automation, but the principles underlying it appear to be applicable to a variety of other situations. The system handles large amounts of engineering data related to computer systems design. These data are stored on direct-access devices and retained throughout the life of a computer system. Records representing the same information at different times in its history can be retained, as well as information reflecting different versions of the same computing system.

The facility, which we call the data management system (DMS), [1] provides multiple-indexing capabilities. Additional indexes can be added that allow data existing in the system to be retrieved in accordance with a theoretically unlimited variety of logical orderings.

The system can handle any data structure that can be represented by a suitable set of rules. These rules, which are specified by macro instructions, define the fields on which indexes are based, the characteristics of the data stored in the system, block sizes, and similar properties of files…

REFERENCES
Computer-Controlled Hardware Testing

Jean W. Sherman
International Business Machines Corporation, San Jose, California

ABSTRACT
Computer-controlled hardware testing can be accomplished safely, dependably, and economically with the four-phase plan presented in this paper. The four phases are product design, product testers, tester programming, and data output. Usefulness of the plan increases if the designer, programmer, and management understand each other's problems and do an adequate job of communicating.
Introduction
The Mask Shop Information System (MSIS) is a computer system which provides real-time control as well as information retrieval functions for a shop where integrated circuit masks are made. This system is now in use at two Bell Laboratories locations, one at Murray Hill, New Jersey, and the other at Allentown, Pennsylvania.....
Computer Expansion of Boolean Expressions

Y. H. Chuang, C. C. Kao
Washington University, St. Louis, Missouri

ABSTRACT
A computer oriented algorithm for symbolically expanding Boolean expressions containing "and", "or", and "complement" operators into their disjunctive norm form is presented. The algorithm consists of two parts in sequence. The first part applies the DeMorgan's laws, and the second part applies the distributive laws. Each part scans the expression only once, and performs the expansion directly while analysing the syntax. The one-pass feature is achieved through a syntax-oriented recursive approach. The most commonly used symbolic convention is assumed for the input and output expressions. The algorithm is believed to be machine-independent and efficient, and it can be easily implemented even on a small computer. The algorithm is found particularly useful in automation of switching circuit simplification and diagnostic test generation.

REFERENCES