acm

SIGDA NEWSLETTER

SPECIAL INTEREST GROUP ON DESIGN AUTOMATION

Vol. 2 No. 1 March 1972

Table of Contents Page

Present SIGDA Organization

Executive Committee:

Charles E. Radke - IBM, P. O. Box 390, Poughkeepsie, New York 12602 - Chairman

Lawrence Margol - Microdesign, Anaheim, California - Vice Chairman

John Hanne - Texas Instruments, Dallas, Texas - Secretary/Treasurer

Editorial Board

Walter Samek - Combustion Engineering, Inc., Windsor, Connecticut 06095 - Editor

Gerhard Paskusz - University of Houston, Houston, Texas - Associate Editor

Lawrence Margol - Executive Committee Representative

Technical papers appearing in this issue are unrefereed working papers.

For future SIGDA Newsletters, please send your contributions to Walter Samek.

Message from Chuck Radke

OUR NEW YEAR'S RESOLUTIONS

After one year of existence as the "new-SIGDA", I believe we have re-established SIGDA as an active professional organization. In one year we have seen 20% growth in membership and activities have gone from in 1970 no newsletter to three in 1971, from no SIGDA meetings at conferences to four informal meetings, from no SIGDA sponsored technical sessions to two. For 1972 we find ourselves planning for improved newsletters, a technical session at SJCC (chaired by me), active involvement in planning 1972 DA Workshop, a technical meeting at SJCC (on the subject of computer graphics in DA, by Bill Sass) and a technical session at ACM 1972 (chaired by Steve Krosner). Instead of informal SIGDA meetings at conferences I plan to have technical meetings covering some particular subject of interest to DA. (You are invited to submit subject and/or volunteer a presentation plus lead discussions.) Of course, we still need our business meetings, but I feel that this year, 1972, we need to put technical content into the "new-SIGDA".

Design Automation is a rapidly growing area, inviting enough for individuals to go-it-on-their-own in providing DA services in areas of building architecture and LSI (Large Scale Integration) chip design. One sees that companies are not content with just happened-stance development of a DA system but are beginning to plan for development and use of DA. This is particularly true in computer design use of DA, as exampled by my conversations with individuals at 1971 DA Workshop and at the Primary Seminar on LOGOS given at Case-Western Reserve University, Cleveland, Ohio, in October 1971.

One problem that I see is that universities are not as involved in DA system activities as one would like to see. I would like to especially extend an invitation to the academic members of SIGDA to become more involved in 1972. The theory of design, design methods and design procedures are an integral part of DA; I would encourage our academic members to sponsor a SIGDA technical meeting on this subject, design and DA, during 1972.

Herman van Beek (see abstracts for SJCC 1972 session on DA) in the introduction to his paper discusses the evolution to LSI (Large Scale Integration). He indicates that this evolution is being realized primarily because of process and engineering advancements - other than in DA. He indicates that DA has in reality played a small part in this evolution and asks the question "Why?". In industry, in computer design, electronic circuit design, building architecture, mechanical engineering, and aerospace design those in DA during 1972 should ask themselves, "is DA prepared to make use of advancements in state of arts of other engineering and programming disciplines?"

Finally, for a specific 1972 New Year's resolution let us, each one of us, make a resolution to submit at least one item this year for inclusion in the SIGDA Newsletter. Also in 1972, we'll see you at one of the (at least) four SIGDA meetings.

The Spring Joint Computer Conference

Abstracts of Session and Papers to be given at SJCC (Spring Joint Computer Conference) May 16, 17, and 18, 1972 in Atlantic City, New Jersey.

SESSION:

Design Automation: Design and Simulation

CHAIRMAN:

Charles E. Radke, IBM, Poughkeepsie, New York

COMMENT:

LSI utilization is forcing the change of Design Automation from a character of quite separate and isolated application programs to sets of programs working off a common design data base. This trend is clearly seen in development of chip design systems in which graphic I/O terminals are used for chip layout off a common design file created by design groups at terminals, extensively simulated at the chip level and evaluated for testing requirements.

In the LSI environment the role of simulation cannot stop at the logic gate level, but is seen to include high-level (algorithmic) simulation as well as a properly interfaced mixture of high-level, logic gate level and circuit analysis level models.

New constraints have been added such as difficulty of inserting test access points internal to large pieces of circuitry. New testing approaches need to be tried.

The conclusion to be drawn is that Design Automation (DA) in LSI is an integral and dependent part of the design process.

PAPER:

Computer Aided Design of MOS/LSI Circuits by H. W. vanBeek, Texas Instruments, Inc., Houston, Texas.

ABSTRACT:

This paper will review the role of computer aids, both hardware and software, used in the design of MOS (Metal Oxide Semiconductors) integrated circuits. The economic pressures on MOS designers to increase the functional capabilities of his designs will be placed in perspective. Specific emphasis will be placed on the parametric circuit analysis and circuit layout problems that must be solved by CAD (Computer Aided Design) programs. Within the latter area the verification and data handling problems will be discussed. The future of interactive-graphic programs for MOS design and layout will be reviewed. The challenge faced by CAD programs will be the containment of human errors without hindering the ingenuity of the MOS design engineer.

PAPER:

The Role of Simulation in LSI Design by J. J. Teets, IBM, Endicott, New York.

ABSTRACT:

Forecasts for Large Scale Integration (LSI) indicate that logic gate densities of 1000 gates per chip could possibly be obtained during the next decade. Supposedly, device and circuit innovations could be used to achieve even higher densities. There are, of course, a number of reasons why the industry is not yet manufacturing major LSI hardware on a large scale. One of these reasons is the lack of superior software support for computer-aided design, particularly simulation aids. For LSI, a major increase in logic design verification simulation will be required at the chip level to ensure correctness of design and to minimize the need for chip redesign.

The purpose of this paper is to review what has happened in the last five years, particularly in design verification, and to project what must be accomplished in the next five to ten years in the area of simulation in order to implement the promise of LSI.

PAPER:

Implementation of a Transient Macro-Model in Large Logic Systems, by N. B. Rabbat, Queen's University, Belfast, Northern Ireland.

ABSTRACT:

With the introduction of LSI, it has become important to predict propagation delays in a large arrangement of logic gates where complex factors such as input rise times, multiple inputs, loading capabilities, short input pulse width and the elemental components of the gates are generally ignored. The consideration of these factors in a system incorporating interfacing and feedback loops can be of considerable importance; also the effect of multiple reflections and crosstalk on the waveforms can cause functional errors in the system.

The contribution outlines a novel method of analysis for predicting waveforms in large systems taking into account the considerations outlined above. The analysis develops a macro-model for the transient input-output relationships of logic modules, and the response is determined by computation at fixed time intervals. For analyzing systems with several hundred modules of different kinds, a library of macro-models is proposed. A small library for gates exists at Queen's University, and an example is presented.

PAPER:

Functions for Improving Diagnostic Resolution in a Large Scale Integration Environment, by M. A. Mehta, Automatic Electric Laboratories, Inc., Northlake, Ill., N. P. Messinger, Illinois Institute of Technology, Chicago, Ill., W. B. Smith, Bell Telephone Laboratories, Holmdel, N.J.

ABSTRACT:

LSI implementation of digital circuitry opens the door to the consideration of dramatically new approaches to the design of system fault diagnosis. New constraints have been added, such as the difficulty of inserting test access points internal to large pieces of circuitry. At the same time, failure modes seem to be changing with bonding lead failures increasing in importance. This paper presents an approach that leans heavily on the assumption that adding additional logic to a circuit is of little consequence, whereas it is important to reduce the access provided for testing capability. As the practicality of the proposed approach has not been examined in detail, the concept is primarily presented to stimulate further study into the special problems and opportunities involved in diagnosis of LSI systems.

The approach discussed here is to incorporate on each least replaceable unit (LRU) a special combinational test circuit that will identify, by observation of the output alone, any stuck at "0" or "1" input failures to the particular input involved. Such a test circuit is termed an "Ambiguity Resolver" (AR) function, and proof of its existence for any number of inputs will be presented. The identification of a failure to a particular input in general actually only isolates the failure to the bus that is connected to that input because of the propagating tendency of failures on a bus. Thus, all LRU's connected to the bus implicated are equal candidates to have failures on their bus connections. A capability such as provided by AR functions would obviously be of little use in a situation in which buses tend to be common to a group of LRU's. However, when the observable points are restricted to the functional outputs, conventional methods usually do not provide resolution of input/output faults to a unique bus. This feature is conveniently provided by the usage of AR functions.

SIGDA TECHNICAL MEETING AT 1972 SJCC

Bill Sass, IBM, Kingston, N.Y., has accepted an invitation to give a talk and lead discussion on "Computer Graphics in DA" at SJCC 1972.

Chuck Radke, Chairman SIGDA, has proposed to Jon Mead, Chairman SIGGRAPH (ACM Special Interest Group on Computer Graphics) to hold a joint technical meeting on the subject.

In the past Bill has authored or co-authored several papers on the subject of the use of computer graphics in DA:

1. "Graphics in Design Engineering" at June 1966 SAE (Society of Automative Engineers) Conference on Advanced Launch Vehicles and Propulsion Systems.

2. "GLEAM/1130 - A Production System Base for Computer-Aided Design" at Fifth DA Workshop 1968.

3. "Computer Graphics Media" at 1969 Conference of American Society of Engineering Education.

4. "1130-Based Logic Layout and Evaluation System" at Seventh DA Workshop 1970.

5. "Revisiting an Operational Graphics Design System" at Eighth DA Workshop 1971.

INVITATION TO ATTEND THE SIGDA MEETINGS DURING 1972

SIGDA members are reminded that every year at FJCC, SJCC, ACM Conferences, and at the DA Workshops, a SIGDA meeting is held. Whatever your interests in DA, these meetings give you the opportunity to discuss areas of common interest with fellow SIGDA members. Non-members are also invited to attend.

To date these SIGDA meetings have been informal with members talking about items of interest to themselves, from piping algorithms to how to get their companies to support DA. We are now planning formal programs for 1972.

CALENDAR OF COMING EVENTS

Sixth Annual Princeton Conference on Information Sciences and Systems	- Princeton, N.J., March 23-24, 1972
Spring Joint Computer Conference	- Atlantic City, N.J., May 16-18, 1972
Tokyo Computer Conference	- Tokyo, Japan, June 8-13, 1972
SIAM 1972 National Meeting	- Philadelphia, Pa., June 12-14, 1972
1972 Summer Simulation Conference	- San Diego, Calif., June 13-16, 1972
1972 International Symposium on Fault Tolerant Computing	- Boston, Mass., June 19-21, 1972
Ninth Annual Design Automation Workshop	- Dallas, Texas, June 26-28, 1972
ACM 1972 Annual Conference	- Boston, Mass., August 14-16, 1972
1972 Fall Joint Computer Conference	- Las Vegas, Nev., November 13-16, 1972

SIGDA TO SPONSOR SESSION AT ACM 1972 IN BOSTON

Steve Krosner, IBM, Kingston, N.Y., has accepted to organize a SIGDA session at ACM 1972 Annual Conference, August 14-16, 1972, in Boston. Last year (1971) in Chicago, SIGDA sponsored two sessions: one chaired by Steve on subject of DA in Building Architecture and Urban Planning and the other by Chuck Radke on Electronic and Mechanical DA areas.

This year the emphasis is on a session which will include at least one tutorial paper. The session will be directed at developers of DA systems rather than users of DA systems. Hence the session should find more interest among the majority of attendees which are more programming, language, and systems oriented.

Areas, which are of interest include:

Design Languages and Processing

Design File Management and Control

Specially written Simulators, Command Languages, and Systems for Design Purposes.

Terminal Systems for Design Use

Graphic I/O and DA Systems

Shared Use of Central Computer Resources for Design Purposes.

SIGDA members and others are encouraged to submit a paper to Steve:

Mr. Steve Krosner
6 Parker Lane
Shokan, New York 12481

Papers should not exceed 7500 words, including a 100-150 word abstract, comprehensive biography, and a full set of illustrations properly keyed to the text. Five copies of the paper are required.

The Ninth Annual Design Automation Workshop

The preliminary program for this major event,as far as SIGDA is concerned, is given below. There are still some loose ends to be tied up, hence you may expect some minor changes here and there in this program. The final program will be sent out by the Workshop Committee, as soon as it is ready.

MONDAY, JUNE 26, 1972

9:00 A.M. Welcome to Dallas - J. R. Hanne
9:05 A.M. Introduction - H. Freitag
9:20 A.M. Keynote Speaker - R. Merwin, Department of Defense

10:20 A.M. Announcements - J. M. Galey

Session 1: Turnkey Systems
Chairman: C. Beardsley, IEEE Spectrum

11:00 A.M. "Interactive Graphics - An Approach to Concept Communications with Design Automation Systems", D. Albert, Systems, Science, and Software.

11:30 A.M. "Computer Generated Artwork - The Aide System", S. Newman, Algorex Data Corp.

A comprehensive system of compute programs which take raw logic diagrams or equations through design to printed circuit board artwork.

1:30 P.M. "A Tourguide to the AEDCAP Roadmap", R. Rohrer, Softech, Inc., Waltham, Mass.

A user oriented general use interactive circuit simulation program capable of performing nonlinear dc, small signal ac, or large signal transient analysis on any circuit file.

2:00 P.M. "Error Free MOS/LSI Design", A. V. Banes, Microdata Co., Chatsworth, Calif.

After an MOS layout is in digital form, a nodal analysis program, equation generation program, and crosstalk analysis program provide assurance of error free artwork.

3:00 P.M. "An Interactive, Graphical System for the Design of Photomasks", F. K. Richardson, Applicon, Inc., Burlington, Mass.

A stand-alone computer, a graphics input/output terminal and a comprehensive software system provide a rapid visual verification of digitized data from schematics to artwork.

3:30 P.M. "Automated Schematics and Printed Circuit Board Artwork", S. H. Zelinger, Calcomp, Inc., Anaheim, Calif.

A Computer Aided Production System (CAP) is used for the production of schematic drawings and printed circuit board artwork after design completion.

Session 2: Packaging
Chairman: B. Britt, IBM, San Jose

11:00 A.M. "Clustering and Linear Placement", D. Schuler and E. Ulhrich, GTE Laboratories.

Two algorithms are presented - one for clustering a set of interconnected nodes and the other for forming a linear placement of clustered interconnected nodes.

11:30 A.M. "A Proper Model for the Partitioning of Electrical Circuits", D. G. Schweikert, Bell Laboratories, Holmdel, N.J.

This paper discusses discrepancies in classical partitioning algorithms and suggests an alternative.

1:30 P.M. "Techniques for Assignment of Logic Elements to Modules", L. Steinberg, Temple University and Univac; L. Mah, Univac.

This paper discusses some criteria for the selection of a placement function for homogeneous modules and develops techniques for the automatic application of these criteria.

2:00 P.M. "Pin Assignment in Automated Printed Circuit Board Design", N. L. Koren, Univac, Blue Bell, Pa.

This paper describes a procedure for making pin assignment which has been empirically shown to substantially increase the number of successfully routed connections.

3:00 P.M. "Topological Class Routing for Printed Circuit Boards", L. Mah, Univac; L. Steinberg, Temple University and Univac.

This paper discusses some difficulties in some variations of Lee's algorithm and describes an alternate approach which considers both topology and capacity.

3:30 P.M. "A High Quality, Low Cost Router for MOS/LSI", R. L. Mattison, GTE Laboratories, Waltham, Mass.

This paper describes an implementation of Hightower's routing algorithm for MOS/LSI integrated circuits.

TUESDAY, JUNE 27, 1972
Session 3: Testing and Simulation
Chairman: S. Szygenda, SMU, Dallas, Texas

9:00 A.M. "Fault Simulation of Digital Logic Utilizing a Small Host Machine", R. McClure, Telpar, Inc., Dallas, Texas.

9:30 A.M. "Fault-Test Analysis Techniques Based on the Use of Logic Simulation", E. G. Ulrich, T. Baker, L. R. Williams, GTE Laboratories, Inc.

10:00 A.M. "TEGAS2 - Anatomy of a General Purpose Test Generation and Simulation System for Digital Logic", S. A. Szygenda, SMU, Dallas, Texas.

11:00 A.M. Another contribution from Bell Labs. Details to be announced.

11:30 A.M. "Applications of Logic Simulation in Design Automation at Texas Instruments", J. Robinson, Texas Instruments, Dallas, Texas.

1:30 P.M. "A Method for Rapid Testing of Beam Crossover Circuits", D. Hightower and B. Unger, Bell Laboratories, Holmdel, N.J.

This paper discusses a simple technique for testing Lepselter type crossover circuits for beam to path shorts.

2:00 P.M. "Hazard Detection by a Quinary Simulation of Logic Devices with Bounded Propagation Delays", D. Lewis.

A quinary algebra is presented which more accurately describes the physical operation of logic networks.

Session 4: Mechanical Design Automation
Chairman: W. Samek, Combustion Engineering

3:00 P.M. "The Graphically Accessed Interactive Design of Thermally Stressed Pipe Systems", M. R. Corley, University of Texas.

This paper discusses the implementation of a limited version of a general analytic method for the computation of thermal stresses in pipe systems using a storage tube computer graphics terminal with a tablet.

3:30 P.M. "The Value of Digital Printer Plotting in Ship Design", A. C. Landsburg, Dept. of Commerce, Washington, D.C.

A lack of available resources has prompted the exploration of the standard line printer to create various graphical outputs which would be helpful in the ship design process.

Session 5: Architecture
Chairman: W. Miller, Design Methods, Los Angeles, Calif.

9:00 A.M. "Lokat: A Spatial Allocation Procedure", A. Bernholtz, Ministry of State, Canada.

A series of two and three dimensional spatial allocation procedures are described.

9:30 A.M. "SYNARC: A Computer Aided Model for Architectural Design", J. Greenberg, California State University.

The SYNARC model provides the architect with a series of modular design programs.

10:00 A.M. "Combining Criteria in Space Allocation Problems", D. Grant, California State Polytechnic College.

Combining proximity criteria with nature-of-the-spot criteria in architectural space allocation problems with the use of a computer aided overlay model space allocation technique.

11:00 A.M. "Modeling the Site Planning of Homogeneous Uses", C. I. Yessios, Carnegie-Mellon University.

This paper discusses a computer implemented model for the solution of site planning problems.

11:30 A.M. "Optimization of Planning and Architectural Decisions Under Conditions of Uncertain Demand", E. Dudnik, University of Illinois.

1:30 P.M. "COMPUCON: Computer Aided Information System for Component Construction", R. D. Roberts, University of Washington.

A comparison of three-dimensional design configurations in graphical form is made in conjunction with pertinent physical, performance, and cost data in alphanumeric form.

2:00 P.M. "An Interactive Computer Graphics Space Allocation System", W. R. Spillars and S. Al-Banna, Columbia University.

This paper describes a computer graphics system which interactively generates layouts by enabling the designer to assign floor space of buildings to different rooms.

3:00 P.M. "Building Polyonimoes; An Approach to the Space Layout Problem", R. S. Frew, Yale University, and P. H. Roe, University of Waterloo.

A method of space layout is described which uses a pattern of square cells built from a given set of preferences of pairwise distances between all cells.

3:30 P.M. "Problem Areas and the Combination of Wall Units to Meet Relation Values Between Rooms", I. Oishi - University of Arkansas.

This paper describes a computer program which at a minimum produces a competent floor plan in which wall segments, window, fireplace, and other wall units are selected, located and displayed.

Session 6: Design Automation Systems
Chairman: D. J. Humcke, Bell Laboratories, Holmdel, New Jersey.

9:00 A.M. "A Logic and Signal Flow Diagram Subsystem", G. H. Sanderson and A. K. Milici, Sperry Rand Corp.

The addition of logic diagramming capability to an existing design automation system is discussed.

9:30 A.M. "Computer-Aided Logic Design System", F. W. Bliss, Bendix, Detroit, Michigan.

This paper describes a system which converts a state table description of a digital system into a logic diagram.

10:00 A.M. "A System for Automated Multilayer Board Layout", J. G. Harvey, Univac, St. Paul, Minn.

A description of a system which accepts designer specified logic equations and performs partitioning (to ICS), placement, routing, and conversion to numerical commands for production of multilayer boards.

Session 7: Microprogramming
Chairman: J. M. Galey, IBM, San Jose.

11:00 A.M. "A Survey of the Current Status of Microprogramming", C. V. Ramamoorthy, University of Texas.

This paper will discuss the current status as well as the tools, techniques and technologies required to further the use of microprogramming.

11:30 A.M. "Microprogramming, A Pedagogical Tool in Teaching Computer Science", R. E. Mervin, Dept. of Defense.

1:30 P.M. "The Role of Microprogramming in the Computer Science Curriculum", R. Rosin and G. Frieder, SUNY at Buffalo, Dept. of Computer Science.

2:00 P.M. "Microprogramming - Its Uses and Tradeoffs", J. M. Galey, IBM, San Jose.

An investigation is described, aimed at determining whether and when to use microprogramming in machine design.

WEDNESDAY, JUNE 28, 1972
Session 8: Interactive Graphics
Chairman: W. Sass, IBM, Kingston

9:00 A.M. "Interactive Textile Design: A Computer Tool for Designers and Manufacturers", J. R. Laurie, IBM Systems Science Institute, N.Y., N.Y.; N. Dooner, Fashion Institute of Technology of the State University of N.Y.

9:30 A.M. "Interactive Graphics for Schematic Editing", M. D. Mancusi and J. C. Wild, Bell Laboratories, Holmdel, N.J.

LOGIGRAF, an interactive graphics program to layout and edit schematic drawings is described.

10:00 A.M. "Interactive Computer Graphics for Design of Mechanical Parts and Production of N.C. Tapes", L. O. Sinkey and J. Z. Gingerich, Pennsylvania State University.

11:00 A.M. "Increasing Capabilities in Interactive Computer Graphics Terminals", C. W. Rosenthal, Bell Laboratories, Murry Hill, N.J.

Several trends in the architecture of terminal systems will be examined.

11:30 A.M. "Picture Generation for Interactive Graphics", L. J. French, RCA Laboratories.

A picture compiler is described that translates a data structure to a display file to drive the display.

Session 9: General Design Automation
Chairman: S. P. Krosner, IBM, Kingston

9:00 A.M. "Gate for Gate Modular Replacement of Combinational Switching Networks", D. Schmidt, Vanderbilt University.

Techniques are described to first design a network to meet a target Boolean function and then to map this design to a constrained set of modules.

9:30 A.M. "Optimal Design of Central Processor Data Paths", K. Irani and G. McClain, University of Michigan.

We explore the possibility of giving the user the ability of describing the desired computer architecture along with the building blocks desired.

10:00 A.M. "Design Automation in Network Computers", E. Bowden Sr., University of Illinois.

In this paper we present a brief description of a network computer currently under development and then propose an interactive design aid system for this distributed network.

11:00 A.M. "Design Automation and Queuing Networks", K. M. Chandy, University of Texas.

An interactive system for the evaluation of computer Queuing models.

11:30 A.M. "Adaptive Control - A Design Optimization Tool", J. Allen III and E. Reynolds, University of Texas.

This paper introduces the concept of design optimization using a real time digital adaptive controller.

1:30 P.M. Panel: "Future and History of Design Automation", Session Chairmen, D/A Workshop Committee.

3:00 P.M. "Critique of the Ninth Annual Design Automation Workshop", D/A Workshop Committee.

PROJECT LOGOS AT CASE WESTERN RESERVE UNIVERSITY

BY C. W. ROSE AND J. P. BARDEN

The descriptive title of Project LOGOS is Computer-Aided Design and Certification of Computer Systems. This account is based on unpublished papers presented at the Primary Seminar on Project LOGOS, 27-28 Octover 1971 at the University and on reports to the sponsor, The Advanced Research Projects Agency of the Department of Defense.

The principal investigator, Edward L. Glaser, arrived on the Case campus in 1967, the veteran of a number of large-scale design efforts, both hardware and software. Glaser had certain critical convictions about computer system design[1] from which he and his fellow investigators have created the motivations and objectives of Project LOGOS.

The Critique

The processes currently used to design, implement, stage, and test large-scale, general purpose computing systems will not be adequate for the "supersystems" of tomorrow. The "pencil and paper" design approach results in inability to enforce design disciplines, incomplete evaluation of interface conditions between separate modules, the introduction of coding errors in transcription of the design for computer-aided implementation, and the lack of adequate diagnostic programs, especially for the operating system.

The splitting of the design team into two insular design groups--hardware and software--causes serious and expensive problems for the integration, staging, and redesign cycle. No practical amount of simulation of the operating system and handchecking of the operating system-hardware interfaces will avoid inconsistencies, overconstraints, and underconstraints when implemented by two groups brought together for the first time. This is especially true when the software designers are dependent upon a description of the hardware which the logic designers only think they have implemented.

This situation is passed on to the user. The system is likely to be "buggy". It will probably be late, and, in these days of unbundling, the price of the software will surely reflect the length and difficulty of the labor.

There are systems in checkout containing approximately 300,000 gates in the processor alone, and larger ones are planned. The complexity of the software necessary to use this computer power efficiently, much of it in parallel or concurrent capability, will increase correspondingly. It is probably safe to speculate that these systems will never be completely checked out, or, if they are, the cost will be prohibitive.

The LOGOS Objectives

The ultimate goal is to create a design environment in which a small team of designers at interactive consoles will specify the total design, operating system and hardware, of a target system implemented by a combination of algorithms and user interaction to do precisely what it was designed to do.

The design process views the entire computer system as an entity. It analyzes, integrates, and tests the entire design history before separating the hardware and operating system components for implementation. It uses the design history to generate sufficient diagnostics to check the integrated system. The bookkeeping and analysis tasks are beyond the scope of any manual system; the entire design process must be computer-aided.

The design team must carry out the design on an interactive tool computer system where the design history is accrued in a machine-analyzable form on which implementation programs can operate directly. The design system must have several components: A data base and the necessary translators to and from the external representation of the target system; analysis programs which determine the consistency of the target system, simulate it, and evaluate its performance; transformation programs which manipulate and restructure the data base; and implementation processors which compile the software portions of the data base and implement the hardware portions in <u>micro</u> or <u>macro</u> logic elements

Finally, the target system must be certifiable. It must do exactly what its specifications on rigorous examination, say it will do. Certifiability in this sense depends upon the validity of the total design process and, in particular, precision as to what are computer processes and their interactions.

The Representational and Structural System

The heart of the LOGOS design system is its representational and structural form. Important leads on representation had been developed by Petri,[2] Holt,[3] Karp and Miller,[4] Slutz,[5] and Luconi.[6] Glaser and his fellow-investigators, however, laid down three crucial requirements:

(1) The representation must be declarative. It must define precisely the nature and structure of computer processes and their interactions;

(2) It must be a formal system in which the interactions and logical consistency can be analyzed algorithmically; and

(3) The target systems thus represented must be directly implementable in hardware and software by automated processes.

When the third requirement, a major objective, was established as feasible, the LOGOS investigators took off from the prior learning, and researches became exciting.

The representation system was developed in graph-theoretic form.[7] There are just two basic interconnected graphs, the data graphs and the control graphs. The data graph defines the data structure, transformations upon it, and data flows. The control graph sequences the transformations and defines the control flows. Any computational task can be characterized and represented by a data and control graph. An ALGOL-like block structure may be imposed on a task, or a set of related tasks, creating an environment tree allowing open and closed subroutines and procedures. Every block can be compressed as a matter of external representation and treated as a primitive function for purposes of higher level analysis. The internal representation is still there if the block itself ever has to be redesigned.

Inherent in this representation is a view of target system structure in which algorithms can analyze the representation, block by block and as a whole, for logical consistency, determinacy, race conditions, concurrency, and other performances as functions of time.[8]

The target system in view is a collection of facilities existing on a hierarchy of layers in which an activation of a facility creates a task. From the software viewpoint, a task is a program. The view from the top of the hierarchy is that the tasks on each layer are executed on virtual machines at lower layers having machine languages for the primitive operations called from higher layers. The inside view is that each layer has a group of facilities whose tasks simulate the machines necessary to execute the primitive operations requested from higher layers.

As the onion is peeled, the actual software primitives--machine language instructions--are reached. Consistency requires extension downward through the hardware-software interface layers to the hardware logic modules and LSI chips. The goal is to force the structure of the representation into exact correlation with the structure of the algorithms of the target system. Analysis of the representation then becomes analysis of the target system processes.[9]

Each facility may have a full set of elements. The elements are (1) a set of resources, (2) control thereof, (3) an interpreter of users' directives from upper layers, and (4) a set of algorithms for performing its task and directing facilities below to do others. No facility, however, need have the full set. A storage allocator, for instance, has resources but no algorithms to be activated from a higher layer.

Concurrency deserves a special note. Concurrency or parallel processing occurs whenever the temporal order of two events is unspecified. True parallelism occurs when operations are simultaneous. Apparent or logical parallelism occurs when the operations are ordered due to limited equipment, but the order is not fixed. Maximization of true or apparent concurrency is often a goal for multiprogrammed target systems. A major problem is determining whether the results of a computation are independent of the order in which unordered operations are performed. The LOGOS representation analysis allows algorithmic detection of pathological parallelisms and provides a means for the synchronization of system primitives on lower layers.

So much for the representation system incorporating the layered hierarchy view of target system structure. It has certain advantages, chiefly its application to users, software, and hardware.

Designers can pool total system resources. This allows the analysis of resources and their allocation in the target system to prevent system deadlocks. Subsets of resources, for instance, can be allocated to a facility, managed locally for a process, and returned to the system.

The layer discipline is consistent with stepwise decompositions and associated proofs of the correctness of algorithms.

Since the internal structure of each layer is fully known, performance at each layer is predictable, and so is the performance of the total system or any portion thereof. At any interface, the higher layer users' computational requests can be modeled with an arrival distribution at any lower interface defined with facilities and a distribution of service times.

Rigorous representation of the target system allows evaluations of design alternatives and redesign without committing the hardware to implementation or the operating system to code.

The location of the software-hardware interface in the representation is a postponable decision, allowing flexibility of implementation. When the interface is fixed, both hardware and software design information can be specified in detail by automated processes.[10]

Data Bases and Their Management

PDMS, the primitive data management system for LOGOS,[11] has been installed on the PDP-10 computer which has 42 million words of on-line disk storage. It implements a set of management procedures with the capability of creating and saving controlled-access data bases. There is a general set of access and manipulation operations.

PDMS incorporates a virtual memory addressing scheme with a paging system and is not limited to the 42 million words of disk storage. It has an extensible set of procedures for the on-line creation, deletion, and updating of the stored information. It operates in a timesharing environment with local (accessible to a unique user) and global (controlled sharing by many) data bases. Linking between local and global bases is provided. There is capability for internal data and data structure description that allows references to data elements without prior knowledge of their physical structure or format.

PDMS, too, has a layered structure. The PDP-10 monitor and file-handling facilities are at the bottom. The paging system is next up as the interface between the computer's file system and the PDMS data management procedures on the next layer. Since the compilers, interpreters, and applications programs reside at higher layers, they all can execute PDMS procedures. The LEAP facilities (a list-based associative data structure and manipulation facility) of SAIL (Stanford Artificial Intelligence Language) are being implemented in the PDMS primitives. Still higher layers contain application-oriented software so that several different language processors, separately or together, can access through PDMS the same or different files.

Designers do their own work in the local bases, with read-only access to global bases. They may link local files into global bases for evaluation of designs before incorporating them in the global bases. These have a unique data processor which, using PDMS primitives, checks the validity and resolves conflicts in all "write" requests made upon the global bases.

PDMS gives users capabilities of defining and accessing general network structures based on linked list concepts, binary tree structures, indexed sequential files, or inverted file organization in terms of attributes or properties.

Present Status

The representational and structural system has been implemented on the PDP-10 system in SAIL. PDMS has been installed and debugged. Research on the detailed syntax and semantics for data operators and the integration of data structure/data operator syntaxes in the representation system is nearing completion. Research has been completed in the main, and the development phase is under way.

Logical and structural consistency analysis algorithms are being defined, and many are now implemented on the PDP-10. A unique internal documentation system is being developed. Work is progressing on the generation of target software and hardware from the representations of target systems.

LOGOS principles have been successfully applied to the design problems of interfacing IMLAC terminals to the PDP-10 system; full implementation was accomplished in February 197

As it stands, LOGOS is an open-loop design system with performance information feeding back to the designers directly. A proposal to close the system with design decisions based on optimization techniques for performance is being seriously explored.[12] The close

LOGOS system should eventually produce, with human designers still making the crucial decisions, certifiable computing systems with design iteration occuring semi-automatically. The lead-times should be measured in months rather than years. The implementation on LOGOS of a primitive target computer system is about a year away.

References:

1. Glaser, E. L., Computer-Aided Design for Computing Systems, IEE Conf., Manchester, U.K., June 1969.

2. Petri, C. S., Kommunication mit Automaten, trans. by C. F. Greene, Applied Data Research, TM RADC-TR-65-377.

3. Holt, R. C., On Deadlock in Computer Systems, Doct. Dissertation, Cornell University, Ithaca, New York, January 1971.

4. Karp, R. M., and Miller, R. E., Parallel Program Schemata, J. Comp. and Syst. Sci. 3,147 (1969).

5. Slutz, D. R., The Flow Graph Schemata Model of Parallel Computation, Doct. Dissertation, M.I.T., Cambridge, Mass., September 1968.

6. Luconi, F.L., Asynchronous Computational Structures, Doct. Dissertation, M.I.T., Cambridge, Mass., September 1968.

7. Rose, C. W., A System of Representation for General Purpose Digital Computer Systems, Doct. Dissertation, CWRU, Cleveland, Ohio, August 1970.

8. Rose, C. W. and Bradshaw, F. T., The LOGOS Representation System, Primary Sem. Proj. LOGOS, CWRU, October 1971.

9. Bradshaw, F. T., Structure and Representation of Digital Computer Systems, Doct. Dissertation, CWRU, Cleveland, Ohio, January 1971.

10. Heath, F. G., and Rose, C. W., The Case for Integrated Hardware-Software, with CAD Implications, Primary Sem. Proj. LOGOS, CWRU, October 1971.

11. Pliner, M. S., PDMS--A Primitive Data Base Management System for Representing Structured Data in an Information Sharing Environment, Doct. Dissertation, CWRU, Cleveland, Ohio, September 1971.

12. Lynch, W. C., unpublished communication, CWRU, November 1971.

DA in Action: Highway Design in Ohio

The Ohio Highway Department has begun using a computer-based highway design system that enables engineers to quickly determine the social and economic impact of proposed road construction. The new system utilizes an IBM System/360 Model 50 and an IBM 1800 data acquisition and control system.

The IBM 1800 is linked with stereo plotting machines used in the terrain mapping and highway design process.

The system permits designers to assess such things as:

Alternate routes,

The amount of earth to be moved or added,

The number of people or businesses forced to relocate,

Relative construction costs, and

Safety factors governed by terrain or traffic considerations.

The department eventually plans to have engineers use TV-like computer terminals to evaluate highway location and design. Seated at the terminal, the engineer through a graphic terminal will be able to modify his plans instantly.