ILL

TN #: 238701

ILL #: 226416481

CUY

System ID: OCLC

Shipping Option: Article Exchange

Patron:

EFTS: No

CUY
UC Berkeley Libraries
University of California - Berkeley (CUY)

133 Doe Library - ILL Svcs. - Borrowing Unit
Berkeley, CA 94720-6000

Ariel:
Odyssey:
e-mail: cuy@library.berkeley.edu
E-MAIL: CUY@LIBRARY.BERKELEY.EDU
Fax: 510-643-8476
Phone: (510) 643-6152
Maxcost: 36.00IFM

Comments:
Notes: IDS_Logic: Chapter Availability - Not a book chapter
Need by: 09/16/2024

Journal Article

Journal Title: Proc. of IEEE International Conf. on Computer Design

Volume: 1 **Issue:**
Month/Year: 1983 **Pages:** 580-583

Article Author: HOWARD J. K HOWARD J. K

Article Title: Introduction to the IBM Los Gatos Logic Simulation Machine

Imprint:
Item #:
Call #: 39072028215774
Location: Libraries Annex: Non-Circulating

Copyright Charges:

Date Received: 7/19/2024 8:35:55 AM

Status:____________________

Date Cancelled:____________________
Reason Cancelled:____________________

Date Sent:____________________
Number of Pages:____________________

University at Buffalo Libraries, UC Berkeley Libraries
BUF NYUSBU
716-645-1334 libsharing@buffalo.edu

Lending - Lending

Introduction to the IBM Los Gatos Logic Simulation Machine

John K. Howard, Richard L. Malm, Larry M. Warren

IBM Los Gatos Laboratory, Los Gatos, California

ABSTRACT

The IBM Logic Simulation Machine, also known as the LSM, is a high speed hardware Logic Simulator that simulates up to 64,512 logic expressions at a rate of 640 x 10^6 expressions per second. Two LSMs have been in use at IBM's Los Gatos Laboratory since 1980 where they have revolutionized logic simulation and design. This paper introduces the LSM, discusses the history of its development, and describes other logic simulators that have been built or proposed.

INTRODUCTION

The Logic Simulation Machine (LSM) is a special purpose computer designed to perform logic simulation at speeds heretofore unattainable. Evaluating a maximum of 64,512 logic expressions at a rate of up to 640 million expressions per second, the LSM represents a significant advance in simulation technology. The LSM is used to simulate logic designs in an interactive fashion, allowing a designer to simulate a design, make a change, and re-simulate — all within minutes.

The LSM was conceived in 1977 by John Cocke of IBM's Thomas J. Watson Research Center in Yorktown Heights, New York, by Richard L. Malm of Los Gatos Laboratory, and by John Schedletsky, also of Yorktown. The LSM was built and enhanced at Los Gatos by a hardware and software team headed by Richard Malm. The LSM has been used regularly at Los Gatos since 1980 and represents, we believe, the first major use of a hardware logic simulator for VLSI design efforts.

IBM's Los Gatos Laboratory is a small development lab located at the southern end of Santa Clara Valley (also known as "Silicon Valley") in California. At Los Gatos, a VLSI design group is engaged in VLSI design, development of advanced design tools, and development of sophisticated testers.

This is the first of four integrated papers[1,2,3] that describe the LSM. This paper introduces the LSM by presenting a short history of logic design and the problems presented by VLSI design in particular, by discussing conceptually the design of the LSM and major decisions made during its development, and finally, by examining some other logic simulators. The second and third papers present an overview of the hardware and software that comprise the full LSM system. The fourth paper discusses Los Gatos' experience with the LSM: what has been learned, how the LSM is used, how the LSM has changed logic design and simulation at Los Gatos.

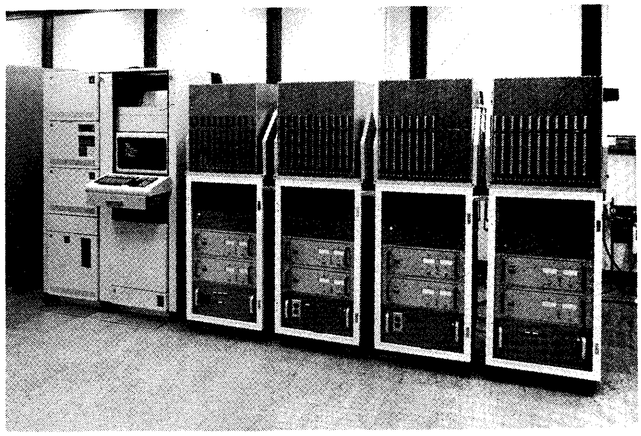

Figure 1. The IBM Los Gatos Logic Simulation Machine

LOGIC SIMULATION

The need for logic simulators has long been recognized. If the accuracy of a design can be verified before a real physical device is built, much time and money is saved. In the competitive field of VLSI design, wasted time can mean the loss of a competitive edge.

If a simulator can closely simulate a design as presented by the designer, design errors can be found before the chip is manufactured. The turnaround time for a design that repeatedly needs to be altered and rebuilt is just too long for present needs.

Of course, neither the problem nor the solution are new. As early as the 1970's, some modelling or simulation method was needed. The form simulation took then was the building of hardware prototypes, simple models that were supposed to function like

CH1935-6/83/0000/0580$01.00

the real design. Prototypes proved too difficult to maintain as design changes were made and usually fell behind in representing the current state of the design.

The next step in the development of simulators was the introduction of software simulators—large computer programs that run on general purpose computers and simulate the functions of a design. As noted by Barto and Syzgenda[4], the Von Neumann architecture was developed originally to solve differential equations and is really not suited to tasks such as simulation at all. The massive data structures that need to be moved, maintained and updated require complex and large computer programs. In turn, these programs are excessively slow and therefore not adequate for current simulation needs.

Still, software logic simulators are the primary simulation tools used in the industry today. To abrogate the problems inherent with software simulators, parts of a design are often simulated separately so that the size of the problem—and the time required to simulate it—are reduced. This ignores the connection between the pieces. Also, the long simulation times often prohibit thorough verification of a design, and design errors can be extremely expensive.

Obviously, a better solution to the simulation problem is needed.

THE LSM SYSTEM

The IBM Logic Simulation Machine was developed in response to the limitations of the software simulators. Specifically, it was felt that a simulator capable of meeting current and future needs should:

- Be fast enough to allow interactive use.
- Have a large capacity to accommodate large designs.
- Give meaningful and precise output in an easy to read form.
- Allow for easy input of designs and design changes.
- Provide a simple method of stimulating the model.
- Be flexible and general in what it will simulate.

The LSM design was initiated and guided by these requirements. This resulted in the following design features:

- Evaluation rate of up to 640 million gates per second. Typical large simulations take just a few minutes.
- Current capacity of 64,512 gates, with expansion possible.
- Graphic output at user terminal of user selected data.
- Easy interface to logic entry systems.
- Test patterns are described by Pascal algorithms or tabular data.
- Delay, and dotted logic are supported. Function tables are programmable.

The LSM is a high-speed, highly specialized computer designed for and dedicated to logic simulation. It is composed of 64 processors operating in parallel with a 100 nanosecond clock. One processor controls simulation, and loads the LSM memories. The other 63 processors simulate random logic by evaluating 5 input random logic gates. Signals are four valued: logical 1 and 0, undefined and uninitialized.

The LSM is supported by an extensive software package residing on a general purpose host computer. An IBM System 370, Model 3081, is currently being used at Los Gatos. The software handles all preprocessing and post-processing of design models and simulation data and provides the primary user interface to the LSM. LSM software is written in Pascal and IBM's System Product Interpreter[5], a control language.

The software does not perform any actual simulation; its purpose is solely to provide a straight-forward and easy implementation of the LSM hardware. Figure 2 shows the relationship between the LSM and the supporting software. Both the hardware and software are discussed fully in the second and third papers of this series.

OTHER HARDWARE LOGIC SIMULATORS

The LSM was not the first hardware logic simulator built. In June of 1970, the Boeing Company put into use what it called the **Boeing Computer Simulator**[6] Recognizing the limitations of software simulators, Boeing developed their simulator to aid the design of digital equipment used in space programs.

The Boeing Computer Simulator was an event driven hardware logic simulator with a maximum capacity of 48,000 gates. Boeing also developed an input language, the "Register Level Simulation Language," RLSL, with which a model was described. Their simulator offered Boeing tremendous logic design advantages at that time.

Both the LSM and the Boeing Computer Simulator are hardware devices, but beyond that common trait, their differences are many. A significant difference is that the Boeing simulator was an event driven simulator and the LSM is not. This difference sets the LSM aside from many proposed and actual hardware and software simulators.

An <u>event driven simulator</u> monitors all the gates in a design and from one cycle of simulation time to the next evaluates new outputs for only those gates that had a change of inputs. Many sophisticated software simulators use this simulation technique, and the concept is a good one: why evaluate gates for which the outputs will remain the same? Theoretically, better performance might be achieved with an event-driven simulator.

The disadvantage in building a hardware event-driven simulator is in the increased complexity and cost due in large part to the large amount of data that needs to be passed among various parts of the simulator. This soon results in switching problems and communication bottlenecks.

LSM	Software
simulation	define model define test patterns start simulation view sim. results change model

Figure 2. Functions of the LSM and its Supporting Software

It was realized early that the LSM could provide more than sufficient simulation speed by relying on fast hardware and parallel processing techniques. This assumption has been more than borne out by experience with the LSM. A highly parallel design also makes the system extremely modular, allowing low entry costs and easy system expansion.

There are other differences between the Boeing Computer Simulator and the LSM:

1. BCS had non-programmable logic functions, implemented by logic cards. LSM functions are fully programmable.

2. BCS test patterns were interpreted on a host computer as the simulation was run. LSM test patterns are compiled and loaded into the LSM as machine instructions. Then the simulation runs.

3. Nets to be examined had to be described in the BCS test patterns. With the LSM, nets can be chosen interactively and separately from test patterns.

Still other differences between the LSM are largely a result of the BCS being built in 1970 when many of today's automated design facilities were not available. Boeing developed a modelling language because at that time sophisticated logic entry systems did not exist. The LSM is able to take advantage of these logic entry systems, compiling a design description that will later be used for production so that it becomes the simulation model. The LSM uses interactive software with CRT terminals; the BCS used punched card input and line printer output.

Nevertheless, Boeing was a pioneer in hardware simulator technology and the Boeing Computer Simulator was certainly a great advance beyond the simulators available in the early 1970's.

Other hardware simulation methods have been proposed. R. Barto and S. A. Syzgenda have presented a detailed description of a possible simulator architecture.[7] They describe an event driven simulator capable of simulating up to 32,768 gates.

According to the estimates given, Barto's and Syzgenda's simulator would perform 197,000 gate evaluations per second. A straight rate comparison is not, however, an accurate comparison of speeds between an event-driven simulator and the LSM. The event simulator assumes that only a percentage of the total gates in a design are active at any given time, and thus gains its efficiency through simulating only those active gates.

Even so, the LSM evaluates gates at such a high speed that event driven simulation is not necessary.

One of the latest entries into the hardware simulation field has been made by **Zycad**.[8] Zycad offers two hardware simulators, the LE-1001 and the LE-1002. Both are event-driven simulators. The most powerful of the two, the LE-1002, simulates up to 65,535 three-input gates at a rate of approximately 1 million events per second.

IBM has continued development of hardware logic simulators with the **Yorktown Simulation Engine**, also known as the **YSE**. The YSE is a second generation LSM, with enhancements added as a result of experience with the LSM. It provides larger capacity and higher speed than the LSM, simulating up to 1 million gates at a rate of 3 billion gate simulations per second. The YSE has been developed and built at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York. It was presented at the IEEE 19th Design Automation Conference in 1982 and described in three papers. [9,10,11]

Where appropriate in the accompanying three papers, differences between the YSE and the LSM will be discussed. Presentation of the LSM in these papers gives a good historical perspective to the YSE, and because the LSM has been used extensively by IBM for over 2 years, that experience can be discussed.

CONCLUSION

The Logic Simulation Machine represents a new era in logic simulation and design. It is much faster and less cumbersome than other current simulation techniques, and IBM's use of the two LSMs at Los Gatos Laboratory has proven very successful. The LSMs are now used by IBM locations other than Los Gatos via IBM's computer communication network.

The LSM's speed frees the designer to spend more time creating and less time checking. Reworking due to design errors can be virtually eliminated. "What if" experimentation becomes possible with little time and hardware commitment. The power of a high speed hardware logic simulator like the LSM takes simulators beyond the verification role they have played in the past and makes them an integral part of VLSI design development.

ACKNOWLEDGEMENTS

As mentioned, the concept of the LSM was created by John Cocke (Yorktown), Richard Malm (Los Gatos), and John Schedletsky (Yorktown). Many others at Los Gatos and Yorktown contributed to the LSM project, and are not listed here. Richard Malm supervised development and implementation of the two LSMs at Los Gatos.

The authors thank Roy Russo of IBM for arranging Los Gatos Laboratory's participation in this conference and for his help with the papers and presentations given here.

REFERENCES

[1] Ted Burggraff, Al Love, Richard Malm, and Ann Rudy, "The IBM Los Gatos Logic Simulation Machine Hardware," Proceedings of the IEEE International Conference on Computer Design: VLSI in Computers , Rye, New York, October 31 - November 3, 1983.

[2] Jack Kohn, Richard Malm, Chuck Meiley, and Frank Nemec, "The IBM Los Gatos Logic Simulation Machine Software," Proceedings of the IEEE International Conference on Computer Design: VLSI in Computers , Rye, New York, October 31 - November 3, 1983.

[3] John K. Howard, Jack Kohn, Richard Malm, and Ann Rudy, "Using the IBM Los Gatos Logic Simulation Machine," Proceedings of the IEEE International Conference on Computer Design: VLSI in Computers , Rye, New York, October 31 - November 3, 1983.

[4] R. Barto and S. A. Szygenda, "A Computer Architecture for Digital Logic Simulation," Electronic Engineering , September 1980, 35-66.

[5] System Product Interpreter , International Business Machines Corporation, product number 5664-167.

[6] A. W. Van Ausdal, "Use of the Boeing Computer Simulator for Logic Design Confirmation and Failure Diagnostic Programs," Advanced Astronautics Science , 29 (June 1971), 573-594.

[7] Barto

[8] Zycad Logic Evaluator General Description , Zycad Corporation, November 1982, Order no. AA-040.

[9] Gregory F. Pfister, "The Yorktown Simulation Engine: Introduction," ACM IEEE 19th Design Automation Conference , June 1982, Paper 7.1, pp. 51-54.

[10] Monty M. Denneau, "The Yorktown Simulation Engine," ACM IEEE 19th Design Automation Conference , June 1982, Paper 7.2, pp. 55-59.

[11] E. Kronstadt and G. Pfister, "Software Support for the Yorktown Simulation Engine," ACM IEEE 19th Design Automation Conference , June 1982, Paper 7.3, pp. 60-64.