1

THEORY AND PRACTICE FOR LARGE-SCALE SYSTEMS

OK - DTIC

Semiannual Technical Report
for the period
April 1, 1983 to September 30, 1983

Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Principal Investigators:	Paul Penfield, Jr.	(617) 253-2506
	Dimitri A. Antoniadis	(617) 253-4693
	Clifton G. Fonstad	(617) 253-4634
	Lance A. Glasser	(617) 253-4677
	Ronald L. Rivest	(617) 253-5880
	Gerald J. Sussman	(617) 253-5874
	Richard E. Zippel	(617) 253-6028

This research was sponsored by Defense Advanced Research Projects Agency (DoD) under ARPA Order No. 3972, issued by Office of Naval Research under Contract N00014-80-C-0622.

[redacted]

Table of Contents

RESEARCH OVERVIEW

This report covers the period from April 1, 1983 through September 30, 1983. The research discussed here is described in more detail in several published and unpublished reports cited below.

A better understanding of the limiting mechanisms in heterojunction transistors is leading to improved devices. Installation of an MBE system will allow production of the first samples in December 1983.

Bipolar transistors have been fabricated on recrystallized silicon films for the first time. These are designed to study the properties of the film. A fully self-aligned JCMOS device has been fabricated with partial success. Another three-dimensional device structure, the staggered CMOS device, has been plagued by leaks in the oxide under the recrystallized layer.

A high-performance FIR filter has been designed at the logic level, as a test bed for retiming and size optimization algorithms. In a related effort, progress has been made on a technique for automatically testing adherence to a design methodology. Among the things checked are threshold drop limits, pullup network topology, information sources and sinks, charge-sharing faults, and races.

Several improvements to the PI placement and interconnect program have been made, including automatic power/ground routing. The program will receive study at an industrial test site. About a dozen systolic-array transformations have been identified as potentially important to designers. One, retiming, has been shown to be computationally feasible. Transformations between two-dimensional and three-dimensional interconnect structures have been derived.

After a disappointing production run of SCHEME-81 chips, with an abnormally high diffusion resistance, that yielded no working chips, a second production run produced some chips that appear to work almost completely. Reliable sequencing, conditional branching, dispatching, and memory references have been observed. These chips appear to be fast, supporting a clock rate of 1 MHz. Much of the user interface of the Schema system has been implemented, along with the basis for the DC analysis. The underlying data base appears to be reliable.

A list of some of the published and unpublished papers appears after the detailed descriptions of the various projects. Some of these are reprinted with this report.

HETEROSTRUCTURE-LOGIC TECHNOLOGY

Fabrication of triply implanted heterojunction npn bipolar transistors hes continued with some improvements in performance seen but gains remaining low. In particular, the excessive base-collector junction leakage, which has dominated the characteristics of all earlier devices, was eliminated in the latest devices by removing the sputtered SiO_2 dielectric film covering the devices and etching ~400 A off the exposed (In,Ga)As surfaces. At the same time, however, the gain was decreased to approximately 0.2. This severe degradation can only be explained by increased lateral emitter-base injection which results in a large decrease in emitter efficiency. An attempt will be made to confirm this explanation by etching a mesa around these devices. If this proves to be the correct explanation, an n^+-collar will be tried, to defeat this lateral injection while maintaining a planar structure. The elimination of the collector-base leakage is a very significant and important advance because it shows that the damage in this doubly implanted junction (n-implant into p-implant) is low enough to give low leakage diodes. It remains to be seen if the minority carrier diffusion length in the implanted base will be long enough to give the desired gains of 10 to 20.

Little progress has been made on improving the gain of the lateral pnp transistors, first reported about a year ago. We understand the importance of the surface better now, however, and feel it plays a crucial role in determining the gain of these devices. We are presently fabricating lateral pnp transistors having a field plate (MIS gate) over the base region. This electrode will allow control of the surface potential, and thereby the surface recombination velocity at the exposed base region surface. This should allow both a quantification of the role of surface recombination on the lateral transistor gain and control of this loss mechanism. If large surface recombination is the cause for the low gain (~4), a shallow n^+ implant may solve the problem as it will tend to shield the minority carrier holes from the surface.

During this report period fabrication of grown junction npn transistors has started. These devices are made by first growing an epitaxial n-InP, p^+-(In,Ga)As, n-(In,Ga)As heterostructure on an n^+-InP substrate. Liquid phase epitaxy is being used to grow the layers and manganese is being used as the base dopant. A multiple energy Be implantation is done to provide access to the p^+ base, ohmic contacts are applied, and mesas are etched to separate the devices. With heavily Mn-doped bases, low gains are seen on both the three-terminal heterojunction transistors just described and on phototransistors, devices lacking the base access implant. Recent devices grown from melts containing less Mn, show significantly better phototransistor gain (estimated at ~100) but the HJBT gain improves by very much less (still < 5). Present work directed at understanding this discrepancy is focusing on the bottom implanted emitter-base junction. This wide band gap homojunction is supposed to be blocking rather then conducting when in parallel with the E-B heterojunction; an attempt will be made to demonstrate whether this is or is not the case, and if it is not, to understand how to make it function as it must. This point also affects the lateral pnp transistors and is a critical issue.

The work on ion implantation, particularly on improving the effectiveness of the post-implant anneal, has led to continued study of the use of a high intensity arc lamp to do rapid isothermal anneals of implanted layers. This work began with GaAs, because the vast literature on this material gives a good benchmark with which to develop the basic arc lamp annealing procedures. Next, work centered on InP and now, arc lamp annealing of (In,Ga)As (see publications).

During this report period, an undergraduate thesis was completed (see publications) in which the first measurements of specific contact resistance on p-type (in,Ga)As were made. It was found that contact resistances in the low 10^{-6} could easily be obtained. At the same time the apparent specific resistance was smaller for smaller area contacts. Both of these preliminary observations are very important for devices; these measurements are now being repeated and extended.

Finally, the molecular beam epitaxy system was delivered in July and assembled in September and October. While there are still pieces back-ordered from the manufacturer, cell back-out procedures can now be started. It is hoped to have the cells baked-out and loaded by the end of November and the first GaAs and (Al,Ga)As layers grown in December. The work will move on to InGaAs after the first of the year. Dr. Al Cho of Bell Laboratories has agreed to provide several graduate students training on (In,Ga)As growth on his systems in January.

THREE-DIMENSIONAL DEVICES AND INTERCONNECTIONS

Two different self-aligned 3-D structures have been pursued. The first one is a stacked CMOS gate (JCMOS) and the second is a staggered CMOS latch. The first structure is primarily applicable in random logic, while the second is primarily suitable for static-RAM applications. The emphasis in this research at this point is on development of novel structures that take advantage of silicon-on-insulator (SOI) technology rather than the development of the SOI technology itself.

A partially successful fabrication of the JCMOS structure has been completed. The key novel feature is that the joint gate is self-aligned to both the upper and lower channel regions. Thus, the structure is scalable in lateral dimensions. Laser recrystallization of the topmost layer which forms the NMOS channel region was used. Since this project does not have a laser apparatus, the recrystallizations were done at the Sperry Research Center in Sudbury, MA. It appears that laser melting is the only practical technique for recrystallization of stacked structures at present. Recrystallizing the top polysilicon layer without affecting the integrity of the two underlying thin gate oxides remains a significant challenge. This step is by for the most important yield limiting step. Working JMOS inverters have been obtained, but the apparent mobility of the NMOS device which resides in the recrystallized layer is very low, of order 10 cm^2/V-sec. This effect is now being investigated, along with a modified version of the JCMOS which promises to yield a more planar surface than the current version.

Fully functional staggered CMOS devices have not yet been obtained. So far these devices have been plagued by leaky oxides which result from the thermal shock of the laser recrystallization. In fact, there is little doubt that melt-induced recrystallization is not going to be a practical technique for production of any kind of 3-D devices. However, it still is the only usable technique for our experiments. Certainly, considerable effort must be placed on the development of a practical non-melting single crystal SOI preparation technique.

During this reporting period bipolar transistors in zone-melt recrystallized filters have been demonstrated for the first time. This effort although not in the mainstream of 3-D devices was undertaken as a study of SOI films. Reasonably good recombination lifetimes were found (30-50 ns), but they are surprisingly lower then independently measured generation lifetimes from deep depletion capacitance recovery. These latter lifetimes were of order of 1 microsecond. At this point it is not clear whether this is a real property of these films or whether the recombination lifetime is lower because of the more involved process for bipolar devices.

VLSI CIRCUIT PERFORMANCE

Anne Park has completed the logic level design of a high speed digital FIR filter to be implemented in 3um cMOS. Each chip will contain several filter stages. A typical image processing application will require on the order of four chips. The throughput goal is several hundred megabytes per second. In addition to the fabrication of some fancy hardware, this project has several high level goals. The first goal is to try out several theories and tools, developed at MIT, in the context of a high performance circuit design. For instance, Leiserson's retiming techniques have been used to develop the initial logic level diagram. Mark Matson's optimizer for circuit level design will also be used. The other high level goal is to explore methodologies for the expeditious design of high performance MOS circuits.

Isaac Bain has been making good progress on a tool for methodology verification of hierarchically described VLSI circuits. The program is embedded in SCHEMA. The objective is the be able to specify a circuit methodology at the start of a design, and have the program check conformance to that methodology as the design progresses. This will relieve the designer of the need to write a new methodology checking program for each time the design style is changed. Wiring operators are used to help capture the designer's intent. The program implementation is now showing signs of life and can check simple circuits created in a wide range of methodologies. Typical checks include the number of threshold drops in a pass transistor network, pullup network topology, Vdd to ground shorts, information sources and sinks, multiple pullups, charge sharing faults, and races.

ROUTING AND COMPLEXITY

Professor Ronald Rivest has worked extensively on the problem of estimating required channel densities, on the average, given the number of nets known to be present in the channel. The primary motivation for this research is to estimate the magnitude of the "economies of scale" enjoyed by large channels relative to small ones. Suppose the "efficiency" of an instance of a channel routing problem is measured as the ratio of the channel density of the instance to the number of nets in that instance. For small channels, this ratio is nearly one on the average, whereas for large channels it approaches 0.5.

This has some surprising consequences for routing gate arrays and standard cell arrays: it may be more efficient to avoid the usual "divide and conquer" approach which tries to balance the number of nets appearing in each channel, and instead use a strategy of trying to maximize the number of nets in (say) the even channels and minimize the number of nets in the (say) odd channels, since the savings on the even channels may more than compensate for the lossage on the odd channels.

The combinatorics of this problem are believed to be previously unstudied. In this work (joint with Chuck Fiduccia) a channel is modeled as a set of 2n pins in a row, paired up into n two-pin nets. All the pins are assumed to be on the same side of the channel, so that there are no vertical conflicts. (This assumption is not unreasonable, given that channel density is being used as the measure of routing complexity.) It is also assumed that each possible pairing is equally likely to occur. While this assumption is clearly a bit unrealistic in the usual context where some care has been paid to module placement, it is a plausible first approximation that should be usable to help quantify the magnitude of the economies of scale enjoyed by the large channels. Preliminary results indicate that an n-net channel should have an expected channel density of

n/2 + O(n**c)

where c is a constant (yet to be determined) between 1/3 and 2/3.

The PI system is developing smoothly. The placement code has been completed and revised. The "min-cut" and "hardening" phases of this phase run quite quickly and give pleasing results. The special ad-hoc code for pad placement is running smoothly, and does a good job of determining the number of sides to use for pads.

The PI power/ground code is being revised to incorporate a new algorithm that runs the VDD tree inwards from a power ring placed just inside the ring of pads. This is expected to result in a substantial improvement in the quality of the resulting power/ground tree. The previous iteration of the power/ground algorithm was presented at the Design Automation Conference, June 1983, by Andy Moulton.

The stretching/resizing code is being substantially revised to generate better results. In particular, two new features have been added to this code: secondary constraints and biases. A secondary constraint is a constraint which is desired, but optional. The constraint-solving code will violate a secondary constraint if necessary to satisfy the primary constraints. This has a number of uses, among the most interesting of which may be the use of secondary constraints to preserve if possible the "nice alignment" of modules which have a number of pins facing each other across a channel. The notion of "bias" also has a number of uses, and quantifies the notion of resolving slack in a favorable direction. The node-elimination algorithm used to solve the constraint graph manipulates bias in an interesting way so that some global consideration can be given to minimizing wire resistance, etc.

The new crossing placement algorithm, developed by Chuck Fiduccia and Rivest, as described in the last progress report, is being implemented and included in PI. This code is still be developed; no experimental results to report yet.

The channel routing algorithm proposed by Burstein, which uses a hierarchical approach, is being implemented. This code is also under development; no experimental results to report yet.

A large number of test examples have been run through the portions of the PI system that are operational. The results are generally quite quickly produced and of good quality, although there is a perceived need to polish the channel routers and improve their overall success rate.

The PI system has been exported to the GE Research Lab in Schenectady, NY, where it is being studied for possible use.

Professor Charles Leiserson has been developing his ideas about systolic and semisystolic design. He has identified over a dozen transformation techniques that can aid in producing efficient systolic and semisystolic architectures. He has started developing models for understanding some of the phenomena. For example, he has added multiplexors to the register/combinational logic model used in his work on retiming.

Leiserson has also been experimenting with designs for complex arithmetic. With a student Ray Hirshfeld, he has designed a fixed-point complex multiplier that is more compact than one can get by routing together four normal multipliers. Now being studied is whether a three-multiplier version, which saves area, can be made to run as fast as a four-multiplier complex multiplier.

Leiserson has started a software project with Miller Maley and some undergraduates to build a layout compactor. The compactor will do one-dimensional compaction with automatic jog introduction. The jogs will be introduced optimally using a polynomial-time algorithm. The theoretical basis for the compactor is being developed and the algorithm improved. The reason a software project is necessary is because the real-world performance of the algorithm would seem to be much better than the current-best theoretical analysis.

With Sandeep Bhatt, Leiserson showed that a problem of orientating rectangles in slicing floorplans was polynomial-time solvable.

Professor F. Thompson Leighton is studying several VLSI-related problems, including: conversion of 2-dimensional layouts into 3-dimensional layouts, design of networks for very fast parallel computation, upper and lower bounds for sorting and packet routing, bounds and algorithms for channel routing, and the problem of decomposing a graph into a small number of stacks.

In the area of 3-dimensional placement and routing, Profs. Leighton and Rosenberg (Duke University) have developed algorithms for transforming a 2-dimensional layout with area A and maximum edge L2 into an H-level 1-active-layer layout with volume V=O((A/H)logA) and maximum edge length L3=O(L2/H). These bounds are close to the best possible. Moreover, the results indicate that, for many circuits, the added ability to locate transistors on multiple layers (as opposed to only wires) does not decrease the volume needed to embed a circuit in a chip.

In the area of parallel computation, Prof. Leighton has developed a simple network (the mesh of trees) that can solve a variety of problems (including sorting, Fourier transform and matrix multiplication) in O(log N) steps. Current research is directed towards finding additional applications for this network as well as for the related multidimensional mesh of trees and the shuffle-exchange graph. Particular effort is being devoted to finding fast, area-efficient networks for packet routing and sorting. As such networks can efficiently simulate an "ideal computer," their discovery could have important applications to the design of supercomputers. Initial work in this area has also led to the discovery of improved lower bounds for the communication complexity of sorting.

In the area of channel routing, Dr. Baker (Bell Labs), Sandeep Bhatt (a contract-supported graduate student) and Prof. Leighton have developed a linear-time approximation algorithm for Manhattan routing. Unlike the many heuristics previously discovered for Manhattan routing, the new algorithm is guaranteed to produce a routing that has channel width at most a constant factor times the optimal channel width. As part of the work, Baker, Bhatt and Leighton formalized the notion of channel flux which (like channel density) is a lower bound on channel width. Initial work with Dr. Pinter (Bell Labs) on the related problem of unrestricted 2-layer channel routing suggests that it may be possible to route any 2-point net problem with density D in a channel of width $D+O(D^{2/3})$. This is nearly a factor of 2 better than the best previous bound and seems to require only a few unit-length vertical wire overlaps per net.

In the area of graph decomposition, Dr. Chung (Bell Labs) and Profs. Leighton and Rosenberg (Duke) are studying methods for "embedding a graph into a book" so that the nodes of the graph are arranged in a line along the spine of the book and so that the edges are drawn without crossing on the pages of the book. The problem is to minimize the number of pages needed as well as the size of the pages. Advances on this problem (which is known to be NP-complete) will have direct application to a variety of wire routing problems for which no good solutions are known. In one application, each page of the book corresponds to a layer of interconnect on a chip. In two other applications, each page corresponds to an electrical last-in-first-out stack of transistors on a chip.

ENGINEERING OF INTEGRATED SYSTEMS

The SCHEME-81 chip has been tested and found to be functional.

The first chips were received from M27PA1 almost 9 months ago. Jon Taft made a beautiful interface for using a SCHEME chip as a computer, interfacing the scheme chip with a sophisticated clock generator, single stepping hardware, history capturing hardware, and main memory cards from the lab's production run of Lisp Machine memories. The first chips were tested with very discouraging results. There seemed to be little life in them and it was hard to tell what had happened. The voltages coming out of the chip were very low. The highest outputs were about 2.3 volts -- barely triggering the TTL circuitry. This made the testing very difficult. After much deduction and testing, it was hypothesized that the problem was that the high diffusion resistance (45 ohms per square in M27PA1) was the source of the problem. The high diffusion resistance problem was accentuated by the very good K′ of the transistors in the three micron process. This led to high currents in the inverters, and hence more voltage drop in the ratioed logic paths.

The pads, between the pad ring transistors and the pad logic, were stretched to accomodate the huge ground bus of the chip, but the pad electronics was powered by diffusions which cross the ground bus. Since two inverters occupy the pad logic, one or the other was always on. Simple calculations of voltage drop across the minimum width diffusion showed that this was the source of the low output voltage for the pads in the "high" logic state. The same source of voltage drop also affected the internal logic high levels on some critical tri-state pads outputs. For a while, it was thought that these bad internal levels were the reason the parts were nonfunctional.

The observed failures could not be correlated with this theory, however, and eventually the real problem was discovered. The diffusion resistance also affected the zero level on superbuffer drivers for the main microcode PLA OR-plane. The voltage drop through the "ground" diffusion was calculated at about .9 volts under normal conditions -- far above what is necessary to destroy the logic zero level needed in the PLA. The machine could be sequenced and would do simple things if Vbb was carefully adjusted to raise the threshold of PLA transistors, compensating for the bad zero levels, but this only worked marginally, since the one levels were also bad. (The same cells would have worked fine on the SCHEME-79 process where the diffusion resistance was only 10 Ohms per square and the transistors had a lower beta). The message here for other designers in the 3 micron process should be clear: Don′t ignore the parasitic diffusion resistances in power, and especially ground lines. Even small numbers of squares of diffusion can be enough to destroy the logic zero levels in the gates.

After this problem was discovered, the cells were redesigned to improve the performance on high diffusion-resistance processes. This is now complete, and simulation is in progress, in preparation for submission of the revised design.

This is where things stood until a few months ago when 11 bonded chips of the original design processed on M33MBA1 were received. The diffusion resistance on that process is about half of the resistance of M27PA1. These chips were tested and they appear to show vastly improved behavior, vindicating the hypothesis. The new chips have reasonable high levels (4 volts) and have reliable behavior. One of them almost works completely! It can sequence reliably, conditionally branch, dispatch on data stored in the registers, and make memory references. It fails in some states, though this is probably a random processing failure. The initial yield was expected to be low, so it is not surprising that no chip works perfectly, but in a large collection of such chips a few good parts would be found. The other good news is that the chips appeared to be fast, with a reliable clock time of 1 microsecond. This meets the original design spec of the chip and would admit a very impressive perfomance in running scheme programs.

In using RNL on the redesigned layout, the parasitic resistances of the clock distribution network became apparent as a major performance limiting factor. Others should be concerned about the parasitic clock distribution resistance in large designs if they are after performance.

A trick used in most industrial design, which has not migrated into the university community is to use the buried contact layer as a lower resistance interconnect in processes with shallow high resistance diffusion layerrs. The buried contact, since it is doped with phosphorous rather than arsenic, tends to diffuse much more deeply, and hence be much lower in diffusion resistance that the normal shallow arsenic diffusions. For power distribution, and for clock crossunders this can be a valuable technique in reducing the parasitic resistance. Expect 7-10 ohms per square for buried contact material, rather than the 40-60 ohms per square for shallow dif.

During the past six months work has accelerated on the Schema system. The overall specification has been completed and work has turned to implementing the system. The system is divided into three layers. The first layer consists of the databases and tools for interacting with the databases. The second layer contains analysis routines, while the third layer contains the synthesis software.

The first layer of software is essentially in place and usable. Besides the databases for dealing with topological specifications, there is a schematic capture system that allows the designer to graphically specify the topology. Appropriate mechanisms exist for representing waveforms, device models (including process corners) and interacting with different simulators. Work is proceeding on completing and polishing the databases and graphics system and enhancing it in several ways. Much of the polishing is being done by Jeff Eisen. Kent Pitman has begun studying the problem of dealing with a distributed design database. Doug Alan has been developing a procedural representation for simulation specification.

The design of the analysis layer software has begun. This layer is split into several components. The component currently being worked on is responsible for DC analysis. The purpose of DC analysis is to allow the designer to specify DC electrical parameters (voltages, currents, power, noise margins) which the system will use to deduce initial values for device sizes. In addition it will derive these DC parameters from a completely sized schematic (with the help of SPICE). Naturally, as the design is polished This has two advantages. First, topological specifications are insulated from process variations. And second, although the initial device sizes may be modified as the design is polished the Schema has available detailed electrical information about the intent of the designer which can be used to validate the final design.

The key problem in doing constraint based circuit analysis, as described by Stallman and Sussman, is solving the system of non-linear equations that results. Unlike bipolar devices, the first order device equations of MOS transistors are polynomials. This allows use of the Grobner basis algorithm for ideals to manipulate the constraint equations. This is a far more effective technique than the resultant techniques used by Sussman. Hand simulations indicate that these techniques should allow dealing effectively with circuits of a dozen or so transistors easily.

Completion and integration of DC analysis system will proceed this fall, and attention will then be turned towards AC analysis during the Spring as the simulation subsystem is completed.

PUBLICATIONS LIST

R. E. Zippel, "An Expert System for VLSI Design," Proceedings of International Symposium on Circuits and Systems, Newport Beach, CA, May 1983.

R. E. Zippel, "Capsules," Proceedings of SIGPLAN ´83, June 1983.

K. Tabatabaie-Alavi, A. N. M. M. Choudhury, N. J. Slater, J. Vlcek, W. Lee, and C. G. Fonstad, "Heterojunction Bipolar Transistor Technology for InP Based Compounds," Compendium of the Second NATO Workshop on "Materials Aspects of Indium Phosphide," Lancaster, England, March 28-30, 1983. In press.

B. S. Baker, S. Bhatt and F. T. Leighton, "An Approximation Algorithm for Manhattan Routing," Proceedings 15th Annual ACM Symposium on Theory of Computing, April 1983, pp 477-486.

F.R.K. Chung, F. T. Leighton, and A. L. Rosenberg, "Diogenes: A Methodology for Designing Fault-Tolerant VLSI Processor Arrays," Proceedings, International Conference on Fault Tolerant Computing, June 1983, pp 26-31, Italy.

Daniel Kleitman, Frank Thomson Leighton, Margaret Lepley, and Gary L. Miller, "An Asymptotically Optimal Layout for the Shuffle-Exchange Graph," Journal of Computer and System Sciences, Volume 23, pp 339-361 June 1983

F. T. Leighton, "Parallel Computation Using Meshes of Trees," Proceedings, 1983 International Workshop on Graph Theoretic Concepts of Computer Science, Osnabruck, West Germany, June, 1983.

M. Rodder and D. A. Antoniadis, "Silicon-On-Insulator Bipolar Transistors," IEEE Electron Device Letters EDL-4, No. 6, June 1983, pp. 193-195.

Thomas Knight, "Design of an Integrated Optical Sensor with On-Chip Preprocessing," Ph.D. Thesis, M.I.T. Department of Electrical Engineering and Computer Science, June, 1983.

Andrew S. Moulton, "Laying the Power and Ground Wires on a VLSI Chip," ACM IEEE 20th Design Automation Conference Proceedings, Miami Beach, FL, June 27-29, pp. 754-755.

J. Rubinstein, P. Penfield, Jr., and M. A. Horowitz, "Signal Delay in RC Tree Networks," IEEE Transactions on Computer-Aided Design, Vol CAD-2, No. 3, July 1983, pp. 202-211.

John L. Wyatt, Jr., "An New Approach to Timing Analysis of Digital MOS IC´s," Twenty-Sixth Midwest Symposium on Circuits and Systems, Puebla, Mexico, August 15-16, 1983.

A. N. M. M. Choudhury, W. Rowe, K. Tabatabaie-Alavi, C. G. Fonstad, K. Alavi, and A. Y. Cho, "Ion Implantation of Si and Be in $Al_{0.48}In_{0.52}As$," J. Applied Physics 54, No. 8, August 1983, pp. 4374-4377.

A. N. M. M. Choudhury, K. Tabatabaie-Alavi, C. G. Fonstad, and J. C. Gelpey, "Rapid Thermal Annealing of Se and Be Implanted in InP Using an Ultrahigh Power Argon Arc Lamp," Applied Physics Letters 43, No. 4, August 15, 1983, pp. 381-383.

F. T. Leighton, "Complexity Issues in VLSI: Optimal Layouts for the Shuffle-Exchange Graph and Other Networks," MIT Press, Cambridge, Massachusetts, September 1983

M. Rodder, "Silicon-on-Insulator Bipolar Transistors," M.S. Thesis, Department of Electrical Engineering and Computer Science, M.I.T., September 1983.

E. W. Maby, H. A. Atwater, and A. L. Keigler, "Electron-Beam Induced Current Measurements in Silicon-On-Insulator Films Prepared by Zone-Melting Recrystallization, Applied Physics Letters 43, No. 5, September 1, 1983, pp. 482-484.

K. Tabatabaie-Alavi, A. N. M. M. Choudhury, C. G. Fonstad, and J. C. Gelpey, "Rapid Thermal Annealing of Be, Si, and Zn implanted GaAs Using an Ultrahigh Power Argon Arc Lamp," Applied Physics Letters 43, No. 5, September 1, 1983, pp. 505-507.

Christopher J. Terman, "Simulation Tools for Digital Design," Ph.D. Thesis, Department of Electrical Engineering and Computer Science, M.I.T., September 1983.

K. Tabatabaie-Alavi, A. N. M. M. Choudhury, H. Kanabe, C. G. Fonstad, and J. C. Gelpey, "Electrical Characteristics of Be-Implanted GaAs Diodes Annealed with an Ultrahigh Power Argon Arc Lamp," Applied Physics Letters 43, No. 7, October 1, 1983, pp. 647-649.

J. L. Wyatt, C. Zukowski, and L. A. Glasser, "The Waveform Bounding Approach to Timing Analysis of Digital MOS IC's," to appear in 1983 IEEE International conference on Computer Design/VLSI in Computers, Port Chester, NY, Oct. 31-Nov. 3, 1983.

R. M. Karp, F. T. Leighton, R. L. Rivest, C. D. Thompson, U. Vazirani and V. Vazirani, "Global Wire Routing in Two-Dimensional Arrays," Proceedings, Foundations of Computer Science, Tucson, AZ, November, 1983, pp. 453-459.

REPRODUCTIONS OF SELECTED REPORTS

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

CAMBRIDGE, MASSACHUSETTS 02139

VLSI Memo No. 83-142 April 1983

Diogenes: A Methodology For Designing Fault-Tolerant VLSI Processor Arrays*

Fan R. K. Chung, F. Thomson Leighton, and Arnold L. Rosenberg**

ABSTRACT

In 1982 A. L. Rosenberg introduced by example the Diogenes approach to the design of fault-tolerant VLSI processor arrays. In this paper, we uncover the principles underlying the approach, and we derive from them a strategy for producing Diogenes designs for arbitrary interconnection networks. We use the strategy to derive optimal Diogenes designs of trees, grids, X-trees, and Boolean n-cubes, as well as surprisingly efficient designs of Benes permutation networks.

*F. T. Leighton was supported in part by a Bantrell fellowship, Defense Advanced Research Projects Agency Contract N00014-80-C-0622, and Air Force Contract OSR-82-0326. A. L. Rosenberg was supported at Duke University in part by NSF Grant MCS-8116522.

** Chung: Bell Laboratories, Murray Hill, NJ 07974. Leighton: Mathematics Department and Laboratory for Computer Science, M.I.T., Cambridge, MA, 02139; (617) 253-5876. Rosenberg: Department of Computer Science, Duke University, Durham, NC 27706.

DIOGENES:
A METHODOLOGY FOR DESIGNING FAULT-TOLERANT VLSI PROCESSOR ARRAYS

Fan R.K. Chung[*]
Bell Laboratories

F. Thomson Leighton[‡]
MIT

Arnold L. Rosenberg[§]
Duke University

ABSTRACT. In [19], Rosenberg introduced by example the Diogenes approach to the design of fault-tolerant VLSI processor arrays. In this paper, we uncover the principles underlying the approach, and we derive from them a strategy for producing Diogenes designs for arbitrary interconnection networks. We use the strategy to derive optimal Diogenes designs of trees, grids, X-trees, and Boolean n-cubes, as well as surprisingly efficient designs of Benes permutation networks.

1. INTRODUCTION

We study here one facet of the problem of designing fault-tolerant microcircuitry, in an environment tailored to a popular VLSI architecture: arrays of identical processing elements (PEs). Our specific problem is the following.

The P(A;p,u) Problem. We want to construct an n-node array A of identical PEs. By using conservative design rules, we may assume that we can fabricate wires and switches perfectly. But we wish to design PEs aggressively, to maximize density and speed. As a result, the PEs experience debilitating faults independently, with probability p. We want to design a fault-tolerant array of PEs, that

- can be configured to simulate the array A;
- utilizes at least the fraction u of the fault-free PEs (so we fabricate n/((1-p)u) PEs to get the desired n-PE array);
- admits an efficient layout; and
- utilizes a switching mechanism that is simple (in structure and in ease of configuration).

Our specific objective is to study and extend the *Diogenes* [19] approach to the *P(A;p,u)* problem. The qualities of the approach — notably, transparency to the PE designer, simplicity of configuration, and high utilization of fault-free PEs — suggest the desirability of studying the approach with an eye to applying it to a wide variety of interconnection networks. The first fruits of our study are reported here. Related theoretical issues occupy [8].

We proceed as follows. By analyzing a sample design, we uncover the principle underlying the Diogenes approach. We use that principle to derive a strategy for producing Diogenes designs of arbitrary interconnection networks. We illustrate the strategy by deriving optimal Diogenes designs of arbitrary trees, of grids, of X-trees, and of Boolean n-cubes, as well as a surprisingly efficient Diogenes design of Benes permutation networks [3]. The paper closes with research questions awaiting resolution.

Related Work. The techniques that have been proposed in the literature for solving the P(A;p,u) problem use one of two basic strategies. The schemes in [2,9,12,13,16,22] incorporate into each PE a switching element that can connect that PE to some fixed repertoire of potential neighbors. Appropriate switch setting in the fault-free PEs interconnects some fraction of the good PEs to realize the ideal array. The schemes in [4,11,15,18,19,21] posit a switching network disjoint from the PEs. PEs are constructed as if for the ideal array, but are interconnected through the switching network rather than directly. The scheme in [10] employs a hybrid strategy.

There have been a few papers that analyze rather than present design methods. [17] evaluates four approaches for designing fault-tolerant linear arrays. His main conclusion is that such evaluations cannot be absolute: one method may be preferred when designing small arrays of *large* PEs, whereas another is superior *for* large arrays of small PEs. [14] derives a model for assessing the cost of a given design strategy. [20] presents evidence that the internal-switch strategy produces designs that consume too much layout area to be considered for any but the smallest arrays.

2. THE DIOGENES DESIGN APPROACH

2.1. An Informal Description

The major aims of the Diogenes design approach are:

- to render the design of the fault-tolerant network transparent to the designer of the PEs;
- to construct a configuration mechanism that is reconfigurable and as *simple* as possible to "program" to the desired structure;
- to enhance testability at a *system* level by building into every array a scan-in/scan-out mechanism for isolating and accessing each PE;
- to utilize (to the extent allowed by array structure) all fault-free PEs.

The approach can be summarized as follows.

[*] Bell Laboratories, Murray Hill, NJ 07974 USA

[‡] Mathematics Department and Laboratory for Computer Science, MIT, Cambridge, MA 02139 USA

[§] Department of Computer Science, Duke University, Durham, NC 27706 USA

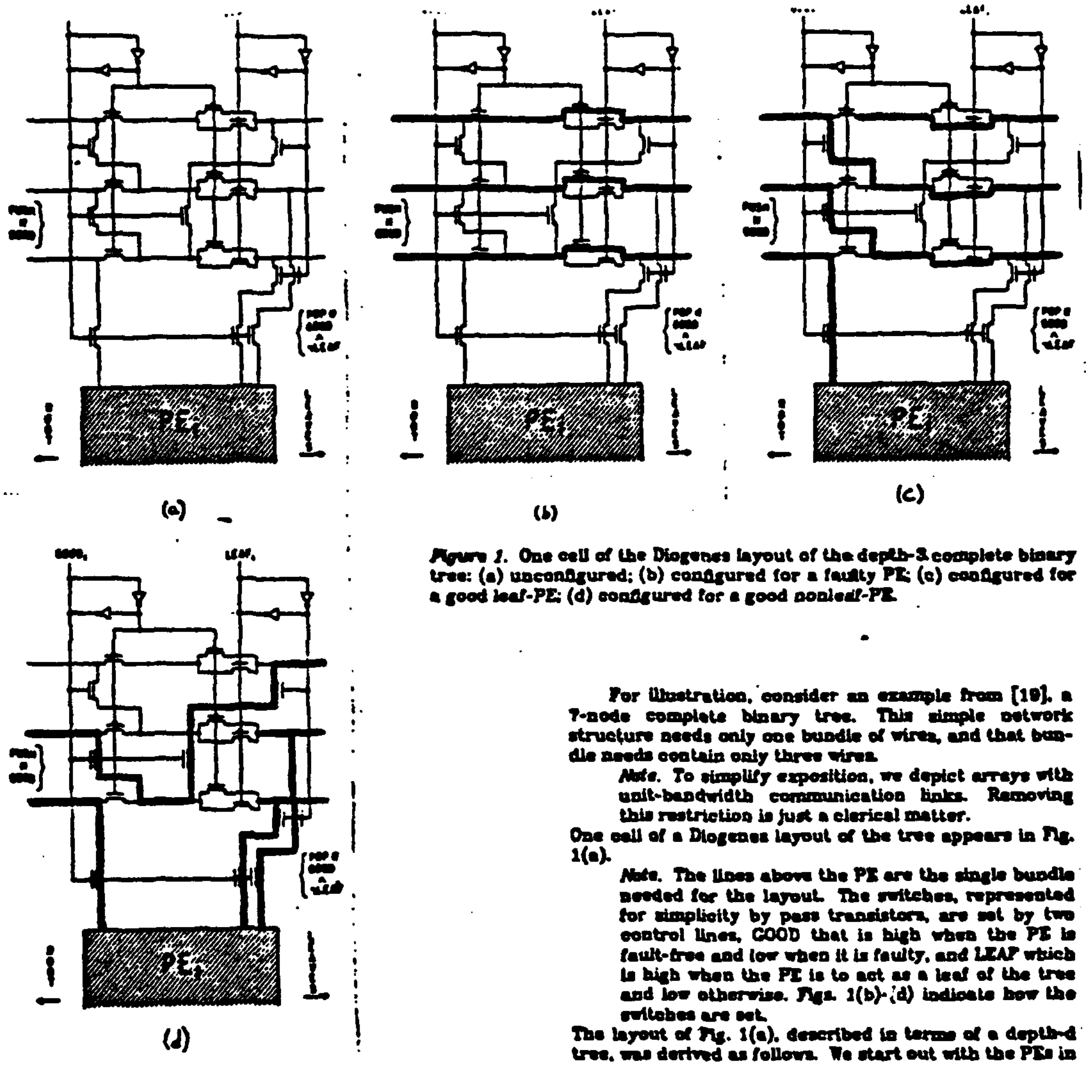


Figure 1. One cell of the Diogenes layout of the depth-3 complete binary tree: (a) unconfigured; (b) configured for a faulty PE; (c) configured for a good *leaf-PE*; (d) configured for a good *nonleaf-PE*.

One wishes to solve the P(A;p,u) problem for some given array A. One begins by fabricating |A|/((1-p)u) PEs in a (logical, if not physical) line, with some number of "bundles" of wires running above the line of PEs. One then scans along the line of PEs to determine which are faulty and which are *fault-free*. As each good PE is encountered, it is hooked into the bundles of wires through a network of switches, thereby connecting that PE to the fault-free PEs that have already been found and preparing it for eventual connection to those that will be found. *One stops looking for good PEs once |A|* have been found. (Alternatively, one could look for all the good PEs, and build the largest array of the given structure that one can.)

For illustration, consider an example from [19], a 7-node complete binary tree. This simple network structure needs only one bundle of wires, and that bundle needs contain only three wires.

> *Note.* To simplify exposition, we depict arrays with unit-bandwidth communication links. Removing this restriction is just a clerical matter.

One cell of a Diogenes layout of the tree appears in Fig. 1(a).

> *Note.* The lines above the PE are the single bundle needed for the layout. The switches, represented for simplicity by pass transistors, are set by two control lines, GOOD that is high when the PE is fault-free and low when it is faulty, and LEAF which is high when the PE is to act as a leaf of the tree and low otherwise. Figs. 1(b)-(d) indicate how the switches are set.

The layout of Fig. 1(a), described in terms of a depth-d tree, was derived as follows. We start out with the PEs in a line. We construct a single bundle with wires numbered 1,2,...,d. We test the PEs so that we know which are good and which are faulty. Next, we proceed down the line of PEs from right to left. As we encounter a good PE that is to be a leaf of the tree (a simple numerical formula tells us which should be leaves), we have it connect up to line 1 in the bundle (thereby preparing it to connect to its father in the tree), simultaneously having lines 1 through d-1 "shift up", to "become" lines 2 through d, respectively; switches disconnect the left parts of the lines from the right parts so that node-to-node connectivity remains correct. The bundle has thus behaved *like a stack being PUSHed; see the left side of* Figs. 1(c,d). When we encounter a good PE that is to be a nonleaf of the tree, we connect it to the stack/bundle in two stages. First, we have the PE connect up to lines

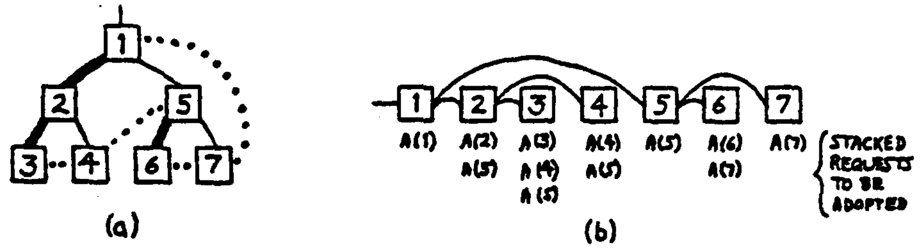


Figure 2. (a) The depth-3 complete binary tree. (b) The width-3 Diogenes (preorder) linearization of the tree.

1 and 2 of the bundle (thereby connecting the node to its sons in the tree), simultaneously having lines 3 through d "shift down" to "become" lines 1 through d-2, respectively; again switches ensure that proper node-to-node connectivity is maintained. The bundle has here behaved like a stack being POPped; see the right side of Fig. 1(d). Second, we have the PE *PUSH* a connection onto the stack, to prepare for eventual connection to its father in the tree. The process we have described here lays the tree out in *preorder* (cf. Fig. 2). Hence, a d-wire stack/bundle suffices to lay out a depth-d complete binary tree.

> *Note.* Our design strategy will require highlighting certain edg-s of the network being laid out, as well as *adding new edges* to it. Highlighted edges in Figs. 2-4 are represented by bold lines; added edges are represented by dotted lines. The significance of both kinds of special edges will be explained in Section 4.

The preceding example should suffice to introduce the Diogenes approach. The designs in [19] simplify the problem of configuring the network by organizing their wire bundles as either stacks (as our example) or queues. For example, two bits of information (= control lines) per PE suffice to configure a line of PEs into any-depth complete binary tree: one bit tells whether or not a PE is good; the other tells whether or not it's a leaf. A less structured bundle (e.g., a crossbar) would require a number of bits per PE proportional to the depth of the tree.

2.2. Stack-Induced Layouts

The Diogenes design approach is distinguished from other external-switch approaches (e.g., [4,18,21]) in its structuring switches so that wire bundles behave as stacks or queues. It is this organizing principle that we exploit to extend the approach to arbitrary interconnection networks.

We restrict attention here to Diogenes designs that organize wire bundles as stacks. Stack-bundles are (as pointed out in [19]) easier to implement than queue-bundles, thereby reinforcing our quest for easily applicable results. Moreover, it is our experience that learning to reason about stacks helps one to reason about queues.

Finally, an avenue for simplification: The Diogenes "recipe" has two parts: a faulty PE is passed by without hooking it into any stack/bundle; a fault-free PE is hooked into the bundles in some relatively complicated way. The former prescription is not interesting: one ignores bad PEs by straightforward use of the GOOD control lines that appear in every Diogenes design (cf. Fig. 1). The interesting aspect of Diogenes designs is how they use stacks to realize interconnections among the good PEs. Recognizing this, we simplify our study by ignoring the GOOD lines and their role in network configuration. This relegates to the background the fault-tolerating aspect of the motivating problem and concentrates solely on the problem of how to use stacks to configure a line of (fault-free) PEs into any desired array structure.

The essence of having a wire-bundle act as a stack is that inter-PE connections made using that bundle never cross.*(This is both necessary and sufficient.) Our topic of study thus reduces to the following. As is customary, we view arrays as undirected graphs (cf. [12]).

The Formal Layout Problem. To partition the edges of the graph G and to lay G out in the plane in such a way that:

- the vertices of G lie on a line;
- all edges of G lie above the line;
- no two edges in the same block of the partition cross.

In view of our earlier remark, it is clear that our problem of realizing arrays using stacks is equivalent to the formal problem just stated. A third formulation will be useful for insight.

The graph G is *outerplanar* if its vertices can be placed on a circle in such a way that the edges of G are noncrossing chords of the circle.

Proposition 1. [6] A graph is realizable with one stack if, and only if, it is outerplanar.

We are, thus, studying *multi-outerplanar* graphs:

> The graph G is *k-outerplanar* if it is the union of k outerplanar graphs whose outerplanarity is witnessed by the same layout of Vertices(G) on a circle.

Proposition 2. [6] A graph is realizable with k stacks if, and only if, it is k-outerplanar.

2.3. The Quality of a Diogenes Layout

Three parameters measure the quality of a Diogenes layout of a graph G:

1. the number of stacks employed in the layout;
2. the (a) *individual* and (b) *cumulative (stack)widths* (= number of lines) of the stacks used in the layout;
3. the number of control bits needed to configure the layout: given the layout, each vertex v of G has an associated vector of pairs of nonnegative integers, called its *type*.

$$T(v) = <<L_1,R_1>, <L_2,R_2>, \ldots>;$$

each L_i (resp., R_i) is the number of edges incident to v that connect via Stack i to vertices lying to the left (resp., right) of v. This measure is the base-2 logarithm of the number of distinct vertex-types in the layout of G, i.e., the number of "control" bits needed to configure the layout in the presence of faults.

We weight these measures in decreasing order of importance when "optimizing" layouts. In [8] we study tradeoffs among them.

3. DIOGENES LAYOUTS OF TREES

Our layout of the *n*-node complete binary tree is optimal with respect to all three quality measures: its one stack respects the outerplanarity of the tree (Prop. 1); its log n width respects the lower bound of [5]; and its two control bits per PE respects our insistence on fault tolerance. We can do almost as well with arbitrary trees.

Fact 1. (a) Any n-node k-ary tree admits a Diogenes layout with one stack of width $\le W(n) =_{def} k/2 \log n$.
(b) There is a fixed layout using a single width-W(n) stack and using $1+2\log_2(k+1)$ control bits per PE, that can be configured to any *k*-ary tree having n or fewer nodes.

Proof Sketch. Let G be a graph. One *adds a fringe* to a vertex v of G by appending to v a line of (possibly 0) vertices:

$$v - v_1 - v_2 - \ldots - v_r \quad r \ge 0.$$

A *fringing* of G is a graph obtained by adding a fringe to each vertex of G.

Concentrate on one vertex v of G. Say that when G is laid out, v is flanked by vertices u and w. Let v have two fringes, $v_1, \ldots, v_r$ and $v'_1, \ldots, v'_s$ (one or both can be empty). Lay the fringes out in the indicated order, between either u and v or v and w. To choose the sides and stacks, look at v's type. Put the first fringe on the side and the stack having the smallest *integer* entry in v's type; place the second fringe using the smallest entry in v's (now altered) type. This increases the cumulative stackwidth by at most 1, while leaving the stacknumber unchanged.

Fact 1 now follows by verifying that any k-ary tree T can be "built" by levels, by starting with a single vertex and "double"-*fringing* the *graph* $\le$ *k/2 log* |T| times. The number of control bits follows from counting the number of distinct vertex-types when all vertices have degree at most k+1. []

4. A GENERAL LAYOUT HEURISTIC

The layout technique of Fact 1 builds explicitly on the *structure of* the *graphs* being laid out. It would be useful to know what to look for in an interconnection network's structure to help one find efficient layouts of arrays of that structure. Experience from numerous Diogenes layouts has led us to the following heuristic.

The graph G is *hamiltonian* if there is a cycle in the graph that meets each vertex just once. The graph G' is an *augmentation* of the graph G if G' is obtained by adding $k \ge 0$ edges to G.

4.1. A Heuristic Layout Procedure.

To find a Diogenes layout for G:

1. Augment G (if necessary) so that it has a hamiltonian cycle.
2. Cut the cycle to obtain a layout of C in a line.
3. Assign edges to stacks using edge coloring as in [8]

As one example of the heuristic, our layout of complete binary trees results from applying the procedure to "preorder" augmentations of the trees. (See Fig. 2(a); the chosen cycle consists of bold and dotted lines.) Other one-stack layouts of trees exist (cf. [7]), but none *has smaller width or number of control bits.*

4.2. Origin of the Heuristic

The heuristic had two origins. First, the heuristic embodies the proof of Proposition 1. Second, it embodies the proof of the following generalization of Proposition 1 to a wide class of planar graphs.

A graph is *subhamiltonian* if it has a planar hamiltonian augmentation.

Proposition 3. [6] The graph G is two-stack realizable (= 2-outerplanar) if, and only if, it is subhamiltonian.

4.3. Applications of the Heuristic.

Square Grids

The augmented cycle formed by row-by-row sweeps in a square grid, as indicated in Fig. 3(a), leads to the layout of Fig. 3(b), which is optimal in number of stacks (the grid is planar but not outerplanar), stackwidth (see, e.g., [18]), and number of node types (the layout distinguishes only between east-to-west and west-to-east rows of the grid).

Fact 2. The nxn square grid admits a two-stack Diogenes layout with stackwidth n and with two node types. This realization is optimal in all three parameters.

X-Trees

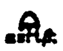
The *depth-d X-tree* X(d) is the augmentation of the depth-d complete binary tree that adds edges going across each level of the tree; see Fig. 4(a).

X(d) has cutwidth d and is subhamiltonian, but not outerplanar. Thus the best possible Diogenes layout would use two stacks of width d. It is very hard to find a two-stack layout of stackwidth smaller than roughly 2^d. (All obvious hamiltonian cycles lead to this enormous width.) The hamiltonian augmentation of X(d) of Fig. 4(a) leads to the stackwidth-3d two-stack layout of Fig. 4(b).

Fact 3. X(d) admits a Diogenes layout with two stacks, one of width 2d and one of width 3d. This reali-ation is optimal in stacknumber and within a factor of 5 of optimal in stackwidth.

The only subtlety here is to verify the claimed stackwidths. As part of our sketched verification, we describe the layout more formally. We proceed by induction. Say that we have a layout of X(d-1) with the claimed parameters and the following form. We depict the layout schematically by its linearization of the vertices, together with a few relevant edges. For simplicity, we draw edges in stack 1 above the line, those in stack 2 below the line.

Layout 1.



here r,s,t are, respectively, the root of X(d-1) and its left and right sons; α,β are the strings comprising the rest of the trees' vertices. Assume for induction that in Layout 1: (1) the left spine nodes [= leftmost nodes at each level] of X(d-1) appear in leaf-to-root order in α; the right spine nodes [the rightmost nodes at each level]

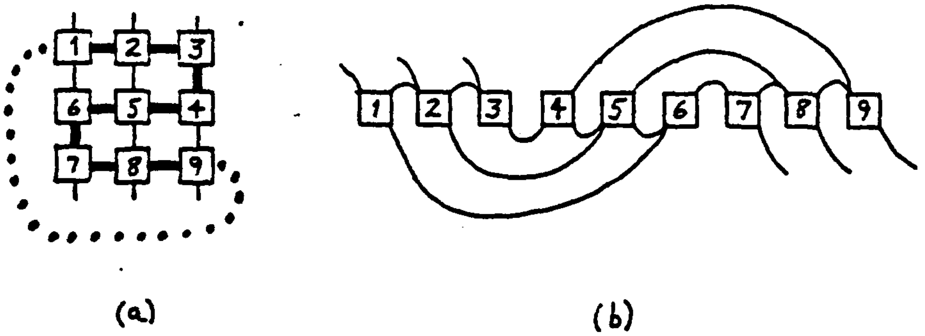
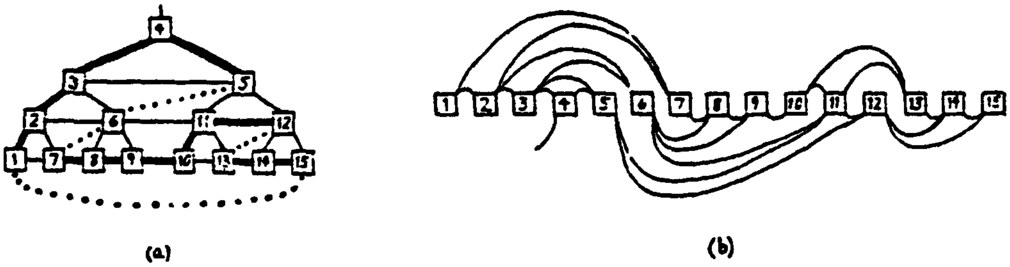


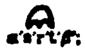
Figure 3. The square grid and its Diogenes linearization.

Figure 4. (a) The depth-4 X-tree X(4). (b) A two-stack, width-4 Diogenes linearization of X(4).



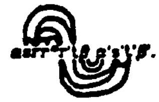
appear (nonconsecutively) in root-to-leaf order in *β*; (2) the nodes r,s,t, and all of the left and right spine nodes are *exposed* from the bottom, in the sense that no edges pass totally under them; (3) the width of stack 1 is ≤ 2d-2; (4) the width of stack 2 is 0 below the left spine nodes and is < 3k-3 to the right of the level-(d-k-1) spine nodes. Take a second copy of Layout 1:

Layout 2.



The layout of X(d) [whose vertex-set is the union of the vertex-sets of its two depth-(d-1) sub-X-trees, plus a root node r°] is obtained from the indicated layouts as follows:

Layout 3.



A careful analysis of the composite layout extends the induction. Analysis of small trees completes the induction, which establishes our claims.

Benes Permutation Networks

Let n be a power of 2. The *n-input Benes network* B(n) is defined inductively as follows; see Fig. 5(a).

- *B*(2) is the complete bipartite graph $K_{2,2}$ on two *input* vertices $i_{1,1}$, $i_{1,2}$ and two *output* vertices $o_{1,1}$, $o_{1,2}$.
- *B*(n) is obtained from two copies of B(n/2), n new input vertices, $i_{n,1}, i_{n,2}, \ldots, i_{n,n}$; and n new output vertices, $o_{n,1}, o_{n,2}, \ldots, o_{n,n}$. For each 1≤k≤n, one adds edges creating one copy of $K_{2,2}$ with "inputs" $i_{n,k}$ and $i_{n,k+n/2}$ and "outputs" $i_{n/2,k}$ and $i'_{n/2,k}$ and one copy of $K_{2,2}$ with "inputs" $o_{n/2,k}$ and $o'_{n/2,k}$ and "outputs" $o_{n,k}$ and $o_{n,k+n/2}$ (primed vertices come from the second copy of B(n/2)).

Benes networks are nonplanar, hence require at least three stacks. We have not yet achieved this bound, but we have found a six-stack realization, by means of the hamiltonian cycle that alternates running up and down the "columns" of inputs and outputs of B(n): see Fig. 5(a). We use three stacks to connect each "column" of vertices to the next; and we alternate sets of three stacks as we proceeed along the graph. It is surprising that any *family of graphs capable of realizing all permutations* can be laid out with a fixed number of stacks.

Fact 4. B(n) admits a Diogenes layout using six stacks, each of width n. This realization is within a factor of 2 of optimal in stacknumber and within a factor of 6 in stackwidth.

The same layout strategy yields layouts of comparable efficiency for structural relatives of B(n), including (log n)-stage cyclic shifters.

The Boolean n-Cube

The *Boolean n-cube* C(n) has as vertices the set of all length-*n* binary strings. String-vertices are adjacent just when they have unit Hamming distance. Thus C(n) has 2^n vertices and $n2^n$ edges. Since C(n) is hard to visualize for n>3, we describe its efficient layout in terms of strings rather than the graphical medium of hamiltonian cycles.

Fact 5. C(n) is n-stack realizable, with one stack of width 2^i for each 0≤i<n. This realization is within a factor of 2 of optimal in both stacknumber and cumulative

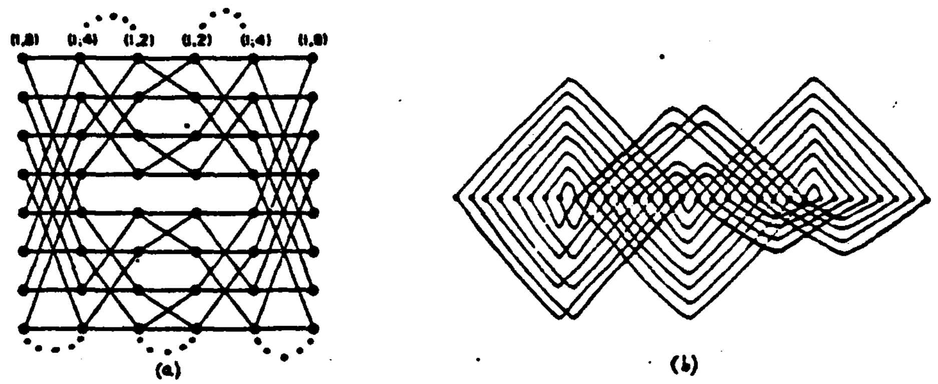


Figure 5. (a) The 8-input Benes network. (b) A six-stack layout of the first three levels of the network.

stackwidth.

The lower bound on stacknumber is immediate from three facts: (a) Stacknumber(C(n)) ≥ the number of outerplanar graphs into which C(n) can be decomposed; (b) an N-vertex outerplanar graph has at most 2N edges; (c) C(n) has $n2^n = N \log_2 N$ edges. The lower bound on cumulative stackwidth is easy to derive.

The upper bound is seen most easily by describing inductively the linearization of C(n)'s vertices.

- C(1)'s vertices are laid out as follows.

  $$0 \quad 1$$

  so one width-1 stack suffices.
- Assume that C(n) is realized with n stacks of widths $1,2,\ldots,2^{n-1}$, via the linearization (letting $N=2^n$)

  $$\beta_1\beta_2 \cdots \beta_N$$

  each β_i being a distinct length-n binary word. The following layout for C(n+1):

  $$0\beta_1 0\beta_2 \cdots 0\beta_N 1\beta_N \cdots 1\beta_2 1\beta_1$$

  uses just one more stack, of width $N=2^n$. This extends the induction.

5. OPEN PROBLEMS

1. Is there a fixed number S such that all planar graphs are S-stack realizable?
2. Is there a fixed number S such that all N-node outerplanar graphs can be realized with S stacks of width proportional to log N?

   The *depth-n ladder* L(n) is an n×2 grid. (a) L(n) is outerplanar, hence one-stack realizable. (b) Any one-stack realization of L(n) has stackwidth ≥ n/2. (c) There is a two-stack unit-width realization of L(n).
3. Is there a fixed number S such that all N-node planar subhamiltonian graphs can be realized with S stacks of width proportional to $N^{1/2}$?
4. Can Benes networks be realized with fewer than six stacks?

ACKNOWLEDGMENTS. The research of the second author (FTL) was supported in part by: a Bantrell fellowship, DARPA Contract N00014-80-C-0622, and Air Force Contract OSR-82-0326. The research of the third author (ALR) was supported in part by NSF Grant MCS-8116522.

The authors would like to thank Gary Miller for helpful and stimulating conversations on this topic.

REFERENCES

1. D.P. Agrawal (1982): Testing and fault tolerance of multistage interconnection networks. *Computer 15*, 41-53.
2. R.C. Aubusson and I. Catt (1978): Wafer-scale integration — a fault-tolerant procedure. *IEEE J. Solid-State Circuits, SC-13*, 339-344.
3. V.E. Benes (1964): Optimal rearrangeable multistage connecting networks. *Bell Syst. Tech. J. 43*, 1641-1656.
4. S.N. Bhatt and C.E. Leiserson (1982): How to assemble tree machines. *Proc. 14th ACM Symp. on Theory of Computing*, 77-84.
5. R.P. Brent and H.T. Kung *(1980): On the area of binary* tree layouts. *Inf. Proc. Let. 11*, 44-46.
6. F.R.K. Chung, F.T. Leighton, A.L. Rosenberg (1983): Multi-outerplanar graphs. In preparation.
7. D. Dolev and H. Trickey (1982): Embedding a tree on a line. IBM Report RJ-3368.
8. S. Even, A. Pnueli, A. Lempel (1972): Permutation graphs and transitive graphs. *J. ACM 19*, 400-410.
9. D. Fussell and P. Varman (1982): Fault-tolerant wafer-scale architectures for VLSI. *Proc. 9th Int'l Symp. on Computer Architecture.*
10. D. Gordon, I. Koren, G.M. Silberman (1982): Embedding tree structures in fault-tolerant VLSI hexagonal grids. typescript.
11. J.W Greene and A. El Gamal (1982): Area and delay penalties in restructurable wafer-scale arrays. *Proc. 3rd Caltech Conf. on VLSI.*
12. J.P. Hayes (1976): A graph model for fault-tolerant computing systems. *IEEE Trans. Comp., C-25*, 875-884.
13. I. Koren (1981): A reconfigurable and fault-tolerant VLSI multiprocessor array. *Proc. 8th Int'l Symp. on Computer Architecture*, 425-442.
14. I. Koren and M.A. Breuer (1982): On area and yield considerations for fault-tolerant VLSI processor arrays. USC Digital Integrated Systems Center Rpt. DISC/82-5.
15. F.T. Leighton and C.E. Leiserson (1982): Wafer-scale integration of systolic arrays. *Proc. 23rd IEEE Symp. on Foundations of Computer Science*, 297-311.
16. F. B. Manning (1977): An approach to highly integrated, computer-maintained cellular array. *IEEE Trans. Comp., C-26*, 536-552.
17. C.D. Rogers (1982): A critical survey of four techniques for the construction of fault-tolerant multiprocessor arrays. In Duke Tech. Rpt. CS-1982-17.
18. A.L. Rosenberg (1981): Routing with permuters: Toward reconfigurable and fault-tolerant networks. Duke Tech. Rpt. CS-1981-13.
19. A.L. Rosenberg (1982): The Diogenes approach to testable fault-tolerant arrays of processors. *IEEE Trans. Comp.*, to appear; Duke Tech. Rpt. CS-1982-6.
20. A.L. Rosenberg (1982): On designing fault-tolerant arrays of processors. Duke Tech. Rpt. CS-82-14; submitted for publication.
21. L. Snyder (1981): Overview of the CHiP computer. In *VLSI 81: Very Large Scale Integration* (ed. J. P. Gray) Academic Press, London, pp. 237-246.
22. W.W. Wong and C.K. Wong (1982): Minimum k-hamiltonian graphs. IBM Report RC-9254.

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

CAMBRIDGE. MASSACHUSETTS 02139

VLSI Memo No. 83-134 February 1983

An Expert System for VLSI Design*

Richard Zippel

ABSTRACT

This paper discusses the motivation and proposed funtionality of an expert system to aid the design of VLSI systems. We present some guiding principles for the construction of such a system and discuss the organization chosen in our particular system, SCHEMA.

*To appear in Proceedings, International Symposium on Circuits and Systems, Newport Beach, California, May, 1983. This work was supported by Defense Advanced Research Projects Agency Contract N00014-80-C-0622.

**Laboratory for Computer Science, Room NE43-721, M.I.T. Cambridge, MA, 02139; (617) 253-6028.

AN EXPERT SYSTEM FOR VLSI DESIGN

Richard Zippel

Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, Massachusetts

Abstract

This paper discusses the motivation and proposed functionality of an expert system to aid the design of VLSI systems. We present some guiding principles for the construction of such a system and discuss the organization chosen in our particular system, SCHEMA.

Introduction

Much of the current VLSI research in universities concentrates on those aspects of the design process that enable one to turn an idea into a set of masks as quickly as possible. Converting those masks into fully functional parts with adequate performance and acceptable yields is an arduous task that often requires at least as much, effort as the initial design. This task is often left undone, and the consequent degradation of functionality, performance and yield is chalked up as a cheap price to pay for participating in the VLSI revolution. In certain situations these costs are small when compared with size and reproducibility benefits over a comparable MSI implementation, but one cannot casually ignore a factor of 10 improvement in performance.

It is always easier to incorporate functionality at the beginning of a project than splicing it in after completion. This is surely true of performance and yield considerations also. It is our belief that with proper tools, designers will be willing and able to take these aspects of the design into account early in the design process. At first tape-out, the designer of an integrated circuit should not only be confident that the design has the desired functionality, but that it also meets the performance goals that have been set.

Furthermore, the designer should not need to relax the project's performance goals to use the design system. The design system must not sacrifice performance or area (yield) significantly. Otherwise, the system would be used only for the "unimportant" projects. In our opinion this design system should initially act as a designer's assistant, keeping track of the details the designer does not care about and performing the monotonous, repetitive operations that would be delegated to apprentice designers. Examples of monotonous, repetitive operations designers request of human apprentices are: Does this inverter have a trip point of 2.3 volts? Does the bootstrapped node boot? Or: Does this adder operate properly after a stretched clock cycle? Using tools that can answer these questions as a basis, more powerful and reliable design synthesis tools can be built.

Such a design system needs a more sophisticated model of the circuits and of the design than is incorporated in most VLSI design systems. It would embody a fair amount of "expertise" in circuit and system design. By virtue of "knowledge" contained in the system and the variety of ways that information can be used, such a system would be considered an *expert system* in VLSI (circuit) design.

Criteria for Expert Systems

We are building a system called SCHEMA that attempts to deal with these issues and provide the sort of environment just described. The following paragraphs describe some of the design criteria we are using to organize SCHEMA and some of its proposed functionality.

Much of SCHEMA's organization is based on our experiences with MACSYMA (a very large system for performing symbolic mathematical calculations) and its shortcomings. The following three design criteria have been suggested for knowledge-based systems in other areas, and we feel they are a good guide for what should be expected from current knowledge-based systems.

First, the system should provide an integrated, user-friendly environment. For instance, a schematic should only be entered once, and the system should be able to check it against the layout or any other constraints. Though this is mostly for the users' benefit, we have observed that any operation a user might want to perform eventually will be required by a program. Integrated facilities greatly ease later development of software and design tools.

Second, the system's internal semantics must match those of the final user as closely as possible. The internal routines must be able to deal with concepts like equilibrated differential signals, nodes that need to be bootstrapped and low impedance outputs. If these semantics are not used then it is very likely that inconsistencies will creep into the system. These inconsistencies will be hard to rationalize to human designers.

In the same way, it is important to let the designer provide information in whatever form is most appropriate for the problem. Often the way the information is specified can be used to guide later computational strategies (the system knows what the user feels is important). For instance, in specifying an inverter the designer should only need to specify a few of the following interrelated parameters: pullup/pulldown transistor width and length, pullup/pulldown shape, DC gain, inverter trip point, low output DC current, noise margins and rise/fall time for a given load. Existing systems require the designer to to specify transistor widths and lengths. Often the designer is more concerned with one of the electrical parameters, like DC current. In this case, the designer should not be required to specify transistor sizes to achieve the DC current parameter, but the DC current should be specified directly, while the system computes the transistor sizes. The inverter operates properly if these electrical parameters are achieved, something the system can check and maintain.

Third, the system must be able inform the designer of the basis for its results. It must be able to provide the designer with the reasons why circuits have the topological or physical structure they have and why certain structures were rejected. If the system generates a surprising or unusual circuit, it is important that the designer be able to determine the system's rationale if she/he is to use it confidently.

These principles, especially the last two, lead to a different sort of design system from that found in current silicon compilers or other synthesis tools that are being developed. When design fragments are synthesized, additional information must be produced to explain the purposes of the design's components. This additional information is nearly always missing in silicon compilers, which generally only produce a final layout or circuit diagram. Consequently it is difficult for designers to modify the design if it doesn't meet the desired performance levels or if it fails to meet the specification for some other reason. Since design is an evolutionary process, these sorts of modifications are inevitable.

Design Synthesis in SCHEMA

A large number of tools that help synthesize designs are already in common use. These range from simple PLA generators to datapath generators and silicon compilers. All of them convert a design specification into final artwork or a final circuit diagram. If the design does not behave as desired, then it is necessary to modify the original specification and apply the synthesis tool again. There are two major flaws with this approach.

First, designs evolve. The goals for a module are constantly changing, and the design must be modified. If the design is directly modified then all the expertise contained in the synthesis tool is lost. Instead, the change must be made to the design's specification. Consequently, the specification language must be able to deal with all conceivable types of constraints on the final design.

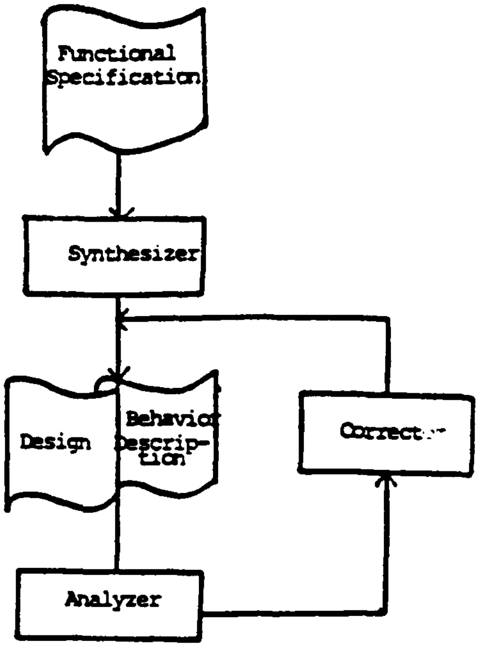


Figure 1: Proposed Synthesis Process in SCHEMA

Second, if the synthesis tools are to produce high performance designs they must incorporate the functional requirements provided in the specification and the low level details that affect performance (subthreshold conduction, junction leakages, etc.). All of these details must be juggled in producing the final design. Developing such an omniscient piece software can be quite difficult. We don't expect human experts' first try to be completely correct. They are expected to use simulators and other analysis tools to determine functional and performance problems. It is important that analysis tools be carefully coupled with the next generation of synthesis tools and silicon compilers.

The model of synthesis used in SCHEMA is shown in figure 1. The synthesis process uses two modules and relies on analysis tools contained in SCHEMA. The first module converts a design specification into a first cut at the design. This initial guess is analyzed and all deviations from specified behavior are noted and fed to a *correction* module. The correction module modifies the design, in an effort to improve its behavior. The modified design is then reanalyzed and recorrected, iteratively converging on an acceptable final design.

The designs produced as initial approximations by SCHEMA must contain more than just the topology and device sizes of the circuit. The correction module needs to know the circuit's desired behavior and each device's contribution. This information is the *behavioral description* of the circuit. The behavioral description is currently organized as a collection of voltage and current signals at different nodes of the topology and causal information connected to the parameters of these signals. The signals themselves are represented as sequences of levels and ramps, which seems adequate for describing the behavior of most digital circuits. The causal information is organized as constraints that relate signal parameters to parameters of the topology. Many of these constraints are easily derived from Kirchoff's current and voltage laws and the device models. Others capture cliches or "engineering models" that are useful for doing rough analyses of circuits. An example of this behavioral description and how it is used is given below.

Simulators produce all of the details of how a circuit works, but it takes time and effort to determine if the waveforms it produces are what the designer wanted. The analysis tools will need to be more than just simulators. The analysis tools planned for SCHEMA will make use of conventional simulators but they are intended to answer the questions the designer wants answered. They will either indicate that the design passes all the specified tests or, if it doesn't, what portions of the circuit fail and hopefully give some clues as to what the possible causes of the failure are. The tests that are applied to the design come from both the design specification and also the behavior specifications of the library modules that were used. For instance, the system should verify that the noise margins of the logic gates remain adequate under the test situation and that bootstrapped nodes actually do bootstrap as high as desired.

The correction module of figure 1 will initially be the human designer, but it is clear that if the analysis tools yield the type of information described above it will be easier to build correction modules to handler the simpler adjustments. In other situations circuit optimizers may prove to be the most effective means of adjusting the design.

Behavioral Circuit Descriptions

The bootstrapped AND gate shown in figure 2 is an example of a circuit that is extremely useful, but requires some care when used. We envision an inexperienced designer using the circuit as it comes from a library. SCHEMA's analysis tools notices when the circuit is being used improperly (based on its behavioral description) and warns the designer. The following paragraphs describe a portion of this circuit's behavior description.

The signals used for the behavioral description are constructed from simple levels and ramps as shown. Here we are only concerned with the portion of the model that deals with the final high voltage of the output. The parasitic capacitance on node B is denoted by C_p, and we denote the gate-to-channel capacitance by C_b. For simplicity, assume both capacitances are constant. Then the final output voltage will be

$$V_{OH} = \min(V_{\phi H}, V_{B2} - V_T).$$

B's bootstrap voltage is denoted by $\alpha = V_{B2} - V_{B1}$ and is controlled by the constraint

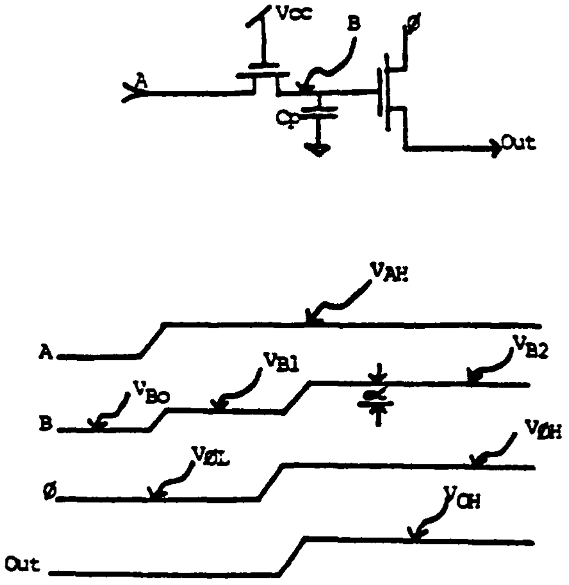


Figure 2: Dynamic AND Gate

$$\alpha = \frac{C_b}{C_b + C_p}(V_{\phi H} - V_{\phi L}),$$

and the voltage V_{B1} is constrained to be $V_{AH} - V_T$.

These constraints tie the circuit's expected internal voltages to the external stimuli, the transistor's threshold voltages and two capacitances C_p and C_b. If the output of this circuit is other than expected, the system can compare expected with observed internal waveforms, isolate differences and propagate these differences through the constraints to see what physical parameters are incorrect. If the output were lower than expected, and only V_{B2} is somewhat low, then only the last constraint (for α) would fail to hold. Since $V_{\phi H}$ and $V_{\phi L}$ are both as expected we must conclude that there is some problem with the capacitances.

This type of model and reasoning process seems to be adequate for describing most first-order phenomena, if care is taken when matching observed transient signals to the piece-wise linear signals used in the model. It narrows the circuit problems to manageable size and provides the designer with guidance in correcting the problem.

Conclusions

We have discussed some the ways in which a VLSI design system can help deal with problems of performance in design and have outlined some of SCHEMA's features that incorporate these ideas. Also, some general criteria for expert systems and how these criteria relate to VLSI design systems were discussed. The key guideline in this endeavor has been to try to ascertain what information the designer really wants to know and develop a system that can provide the information.

This work was supported by DARPA grant N0014-80-C-0622.

Capsules

Richard Zippel
Massachusetts Institute of Technology
Laboratory for Computer Science
Cambridge, Mass. 02139

Well organized large systems tend to consist of a large number of small pieces of code each of which captures a single semantic unit. These pieces of code are strung together to form larger semantic phrases, which are in turn components of even larger phrases. The smallest semantic units in a polynomial manipulation system might be the routines that add and multiply the coefficients of the polynomials. These routines are combined (used as subroutines) to form the routines that add and multiply polynomials, which are components in the factoring and greatest common divisor routines.

When a system is built in a top down manner, the larger phrases are formed first and are used to define the semantic components of the smaller phrases. Bottom up software design begins with the small phrases and generates the large phrases. In practice a combination of these two approaches is often used. The manner in which the routines are initially connected is usually simple, but in time the addition of new capabilities and features, and the necessities of performance enhancement generally cause the dependency structure to become quite complex.

A good example of this sort of complexity is when a new, "higher performance" representation of some data structure is introduced for critical uses (caching and hash tables examples of this optimization). In building an input/output system for a computer, we might initially specify a *stream* to be a simple, character at a time structure. When this structure is used for file operations and networking it becomes necessary to add buffering, but it would be unwise to use the buffered stream for terminal I/O. These two types of streams could share large amounts of code if the manner in which the lower level routines are "glued together" is sufficiently powerful. For instance, the only difference between the routines which close the stream is that the buffered stream must flush its buffers and return them to the buffer pool.

We are particularly interested in the "gluing together" process which is used to form large systems. In this paper we will describe a system called *Capsules* which we feel provides a more natural and more powerful combination mechanism than discussed previously. This system was originally an attempt to simplify the construction of an algebraic manipulation system, but we are now applying it to the development of a VLSI design system and investigating its utility in organizing the I/O system of a complex personal computer.

1. Philosophy

In most systems, when a piece of code is written it is given a name. In the earliest programming languages (Fortran, Basic, Lisp 1.5), When the user wants to perform some operation (like pushing an element on a stack or outputting a character), it is necessary to find a piece of code that implements the desired operation and refer to it using its name. In some systems (CLU [Lis77], Flavors [Wei81], Loops [Bob82], Smalltalk [Ing76, Xer81]) an extra level of indirection is introduced that allows the binding of the piece of code to an operation name to be delayed until after the code is written. This approach has been called *data abstraction*. In compile-time languages like CLU, the association of the code with the abstract operation is made at compile or link time. The Lisp and Smalltalk versions of these approach delay the binding until runtime. In either case, an extra level of indirection has been provided between the name representing an abstract operation and the piece of code that implements that abstraction. There is still no tie between the abstract operation as a semantic unit that the user wants to use and the piece that implements the operation.

In the Capsule system, the user specifies the desired behavior of the operation and the system is responsible for finding the piece of code that implements that operation and is compatible with previous constraints. If a more efficient piece of code is written that implements some operation, the system will use the more efficient code as long as it meets the user's specifications. This is a result of (1) the user referring to code fragments by their semantic purpose rather than their name, and (2) the system being responsible for matching the the semantic requests with the code fragments in the system.

2. Capsules

In the capsule system we have assumed that all actions occur by sending messages to objects—this is the *object oriented viewpoint*. Taking the dual point of view, where objects are passive and the correct function is chosen by the compiler, merely moves the mechanism we are discussing into the compiler. This is the fundamental difference between the Lisp Machine's flavor system which takes the object oriented viewpoint, and CLU which is function oriented.

By an *object* we will mean something to which a *message* can be sent. This will result in one (or more values) being returned and the internal state of the object being changed. The action caused when a message is sent to an object is called an *operation*. A message is a string used as the name of some operation. It contains no internal structure. The piece of code executed when a message is sent to an object is called a *method*. Objects may also contain internal state which is kept in *instance variables* that may be referenced by the methods.

Every object belongs to a class of equivalent objects that have the same methods and the same set of instance variables. This equivalence class is called a *collage*. Objects are created by calling the function MAKE-OBJECT on a collage. The methods are actually part of the collage, so as operations are added to the collage, the objects of the collage also acquire them.

The specification for how a method is to be constructed is kept in a structure called a *capsule*. When a capsule is added to a collage, the code within the capsule is incorporated in one or more of the methods of the collage. Capsules also contain information describing what their pieces of code expect of the collage to which they are added (what operations and instance variables there are, for instance).

The design of the capsule system was based on our experience with some very large software systems, and it is in the construction of large systems that its power is most apparent. The following paragraphs use a small example to explain the mechanisms and terminology of the Capsule system. As such, the Capsule mechanisms may seem to be overkill. The reader is asked to treat this small example as what it is, and to map the capsule mechanisms onto whatever large software system is familiar.

The small example we use is the implementation of a stack. A stack is an object that accepts two messages, PUSH and POP. These messages have the obvious meaning. We will additionally introduce an operation called TWIDDLE which interchanges the top two elements of a stack. Two implementations are given for stacks. One uses a list to implement the stack, while the other uses an array.

An *operation* is the specification of an action. It includes specifications for the number and type of arguments and return values as well as a specification of what the action will accomplish, called the *semantics* of the operation. The current implementation does not interpret the semantics in any manner. The semantics fields are checked for equality to ensure that two operations perform the same action.

To implement our stack example, the first thing we need to do is specify the operations that will be used, PUSH, POP and TWIDDLE.

```
    (DEFOPERATION PUSH
      (ARGUMENTS NIL)
      (RETURNS)
      (SEMANTICS PUSH-ELEMENT-ON-STACK)
      (DOCUMENTATION "Adds an element to the top
of a stack"))
```

The ARGUMENT field indicates that PUSH takes exactly one additional argument, its type is unspecified. No values are returned. The semantics field has the atom PUSH-ELEMENT-ON-STACK in it. Since the the semantics field is not really interpreted by the current version of this system, this atom is used as a place holder. We will leave out the semantics fields in the following examples. The documentation string is used by the run-time documentation system.

```
    (DEFOPERATION POP
      (RETURNS NIL)
      (DOCUMENTATION "Removes and returns the top
element of a stack"))

    (DEFOPERATION TWIDDLE
      (DOCUMENTATION "Exchanges the top two ele-
ments of a stack"))
```

The default assumptions are that an operation takes no arguments and returns no values. These assumptions are used in the specifications of POP and PUSH.

Protocol's are used to specify the characteristics of a collage. A protocol is a list of (1) operations (including their semantics), (2) axioms, which specify relationships among operations, (3) instance variables, and (4) attributes, which are other characteristics. The following protocol captures the notion of a stack.

```
    (DEFPROTOCOL BASIC-STACK
      (OPERATIONS PUSH POP)
      (AXIOM
       (STACK-PUSH-POP-AXIOM PUSH POP)))
```

That is, a stack accepts two operations, PUSH and POP (as described above) and these two operations obey the STACK-PUSH-POP-AXIOM, which means that a PUSH followed by a POP returns the value originally pushed and all the inductive variations of that statement. As with the semantics portions of operation specifications, the current system does not attempt to interpret the axioms in more primitive terms, but treats them atomically.

The following slightly more complex protocol illustrates how the mathematical abstraction of an algebraic ring may be specified.

```
(DEFPROTOCOL RING
  (OPERATIONS PLUS MINUS ZERO
              TIMES)
  (AXIOMS
   (COMMUTATIVE-LAW PLUS)
   (ASSOCIATIVE-LAW PLUS)
   (ASSOCIATIVE-LAW TIMES)
   (ALGEBRAIC-IDENTITY PLUS ZERO)
   (ALGEBRAIC-INVERSE PLUS MINUS)
   (DISTRIBUTIVE-LAW PLUS TIMES))
  (ATTRIBUTES
   (CHARACTERISTIC)))
```

It is assumed that the DEFOPERATIONs for the specified operations appear elsewhere. This specification indicates that there is a ZERO operation that returns the additive identity. An alternative implementation might require ZERO to be an instance variable. Any collage that adheres to this protocol is an *abstract ring*. Any piece of code that depends only on this protocol can be added to an abstract ring. This is somewhat closer to the mathematical understanding of abstraction than previous systems.

The DEFOPERATION form defines a protocol that contains a single operation and nothing else. The name of this protocol is the same as the name of the operation unless specified otherwise. This conveniently allows the use of operation names and protocols interchangeably.

All code is put into capsules. Capsules consist of four basic parts, (1) a required protocol, what the capsule expects of the collage it is to be added to, (2) an asserted protocol, things to add to the protocol of a collage when the capsule is added, (3) performance information about the algorithm contained in the capsule, and (4) the code itself. It often happens that more than one capsule could be added to a collage to satisfy some requirement. The performance information is used to break those deadlocks.

In order to allow incremental compilation and debugging, the specification of a capsule is separated into two pieces. A DEFCAPSULE form is used to indicate the first three parts of the specification while separate *DEFALGORITHM* forms are used for each piece of code. (In the terminology used in the flavor system, capsules are extensions of flavors, and algorithms are methods. Our algorithms can be more complex than the simple pieces of code that flavor methods must be, but it would take us too far afield to discuss these capabilities here.) The following implementation of the BASIC-STACK protocol illustrates this.

```
(DEFCAPSULE LIST-STACK
  (ASSERTS
   (PROTOCOL BASIC-STACK)
   (INSTANCE-VARIABLE (STACK ()))
   (ATTRIBUTE STACK-IMPLEMENTED-AS-LIST)))

(DEFALGORITHM (LIST-STACK PUSH) (ELEMENT)
  (SETQ STACK (CONS ELEMENT STACK)))

(DEFALGORITHM (LIST-STACK POP) ()
  (PROG1 (FIRST STACK)
         (SETQ STACK (REST STACK))))
```

The LIST-STACK capsule implements a stack in terms of a list of the elements of the stack. It makes no assumptions of the collage to which it is to be added. When it is added to a collage, the BASIC-STACK protocol is added, along with an instance variable STACK and the two pieces of code given in the DEFALGORITHM. In addition, an attribute is added to the collage that indicates the stack is implemented using a list.

The capsule that implements stacks in terms of arrays, is quite similar (we have ignored the problem of running off the end of the array for simplicity here).

```
(DEFCAPSULE ARRAY-STACK
  (ASSERTS
   (PROTOCOL BASIC-STACK)
   (INSTANCE-VARIABLES
     (STACK (MAKE-ARRAY '(100)))
     (INDEX 0))
   (ATTRIBUTE STACK-IMPLEMENTED-BY-ARRAY)))

(DEFALGORITHM (ARRAY-STACK PUSH) (ELEMENT)
  (SETF (AREF STACK INDEX) ELEMENT)
  (SETQ INDEX (+ INDEX 1)))

(DEFALGORITHM (ARRAY-STACK POP) ()
  (SETQ INDEX (- INDEX 1))
  (AREF STACK (+ INDEX 1)))
```

The function MAKE-COLLAGE is used to create collages. It takes an arbitrary number of arguments, each of which is either a capsule or the name of a capsule, or a protocol or the name of a protocol. Thus the following forms can be used to construct stack collages of the two type defined thus far.

```
(SETQ C1 (MAKE-COLLAGE 'LIST-STACK))
(SETQ C2 (MAKE-COLLAGE 'ARRAY-STACK))
```

The form (MAKE-COLLAGE 'BASIC-STACK) would result in an error because it is ambiguous. There are two capsules that can be used to create a collage with the BASIC-STACK protocol and there is no reason to prefer one over the other. (In this situation we have seriously considered just picking the first capsule. The user hasn't given any reason to prefer one capsule over the other so why not pick one at random?)

Once we have a couple of collages to work with, we can create stacks using the MAKE-OBJECT function. Its first argument is either a collage or the name of one. Additional arguments are passed to the initialization method if there is one. The following forms, create a stack from the collage C1 and push two elements onto it.

```
(SETQ STACK (MAKE-OBJECT C1))
(SEND STACK 'PUSH 1)
(SEND STACK 'PUSH 2)
```

Now, (SEND STACK 'POP) will return 2.

Though we have defined what is meant by the TWIDDLE operation, no capsule implements it. The following capsule provides an algorithms that "TWIDDLEs" the top two elements of an abstract stack.

```
(DEFCAPSULE BASIC-TWIDDLE
  (REQUIRES BASIC-STACK)
  (PERFORMANCE 1)
  (ASSERTS
   (PROTOCOL TWIDDLE)))

(DEFALGORITHM (BASIC-TWIDDLE TWIDDLE) ()
  (LET (TOP SECOND)
    (SETQ TOP (SEND SELF 'POP)
          SECOND (SEND SELF 'POP))
    (SEND SELF 'PUSH TOP)
    (SEND SELF 'PUSH SECOND)))
```

This capsule has a required protocol, BASIC-STACK. Thus it can only be combined with collages that already possess the PUSH and POP operations. It can be added to any abstract stack.

The routine ADD-PROTOCOLS is used to add protocols to collages. Its first argument is a collage and the rest of its arguments are protocols that the user wants the collage to meet. Thus the form

```
(ADD-OPERATION C1 'TWIDDLE)
```

adds a TWIDDLE operation to C1. More precisely, each collage contains a table that gives the relationships between message names and pieces of code. Each object constructed from a collage contains a pointer to this table. When an operation is added to a collage, the system isolates a capsule that both provides the desired operation and which can be added to the collage. The code portion of the capsule is then added to the collage's method table. Thus all objects of the collage are now extended with the new operation.

It is easy to define a slightly more efficient version of TWIDDLE for arrays. The ARRAY-TWIDDLE capsule does precisely this.

```
(DEFCAPSULE ARRAY-TWIDDLE
  (REQUIRES
    (ATTRIBUTE STACK-IMPLEMENTED-BY-ARRAY))
  (PERFORMANCE 2)
  (ASSERT
    (PROTOCOL TWIDDLE)))
(DEFALGORITHM (ARRAY-TWIDDLE TWIDDLE) ()
  (LET ((TEMP))
    (SETQ TEMP (AREF STACK (- INDEX 1)))
    (SETF (AREF STACK (- INDEX 1))
          (AREF STACK (- INDEX 2)))
    (SETF (AREF STACK (- INDEX 2)) TEMP)))
```

Notice that this capsule does not actually use the PUSH and POP operations. It only assumes that there are instance variables STACK and INDEX, and that they can be interpreted to form a stack. This is the purpose of the STACK-IMPLEMENTED-BY-ARRAY attribute.

With this capsule added to the system, adding the TWIDDLE operation to an ARRAY-STACK collage will get the new code, while previously it would have used the routine in BASIC-TWIDDLE.

As a final note, if a message is sent to an operation that does not possess a handler for that message then a *default-handler* is run. One of the default handlers with which we have been experimenting attempts to add the desired operation to the object's collage and then tries again. Thus if a stack did not have a TWIDDLE handler, a TWIDDLE message could be sent to it anyway, since one could be created for it and installed on the fly. If the stack was an ARRAY-STACK then the efficient ARRAY-TWIDDLE capsule would be added, otherwise the BASIC-TWIDDLE can be used.

There are two points to notice about this scenario. First, when new functionality was added to a collage, the user specified only what the desired semantics were and did not specify, directly or indirectly, a particular piece of code. Second, the functionality could be added dynamically while the system is running. In some domains, there are several algorithms for performing an operation and it can be very expensive to decide which to use. It is better not to pay that price until it is truly necessary.

3. Conclusions

When the capsule system was used to describe a portion of the stream code used in the LISP Machine, we noticed that the protocol specifications seemed more verbose than we would have liked. Closer examination revealed that many of the comments we had penciled into the version of the code that used flavors were being translated into protocols. This reinforces our impression that the capsule system is partially an attempt to force the programmer to make the code that is written more precise.

We feel that if the programmer makes this effort, the programming system will be in a much better position to aid in the development of large software systems. The Capsule system is a partially successful attempt to provide mechanism through which the programmer can truly express what the program is intended to do.

4. Acknowledgments

This work grew out of a long series of discussions with David Barton, Richard Jenks and Barry Trager on how to organize the algebraic algorithms of MACSYMA. Without this nearly insurmountable problem, none of this work would have come to fruition. Other discussions with Jeff Arnold, Carl Hoffman and Jon Sieber have been invaluable. This work was supported by DARPA contract N00014-80-C-0622.

References

1. Daniel G. Bobrow and Mark J. Stefik, "LOOPS: An Object Oriented Programming System for Interlisp," *Proceedings of the European AI Conference* (1982).
2. Daniel H. H. Ingalls, "The Smalltalk-76 Programming System: Design and Implementation," *Proceedings of the Principles of Programming Languages Symposium* (1976).
3. Barbara Liskov, Alan Snyder, Russel Atkinson, and Craig Schaffert, "Abstraction Mechanisms in CLU," *Communications of the ACM* **20**, (1977), 564-576.
4. Daniel L. Weinreb and David A. Moon, *Lisp Machine Manual*, MIT Artificial Intelligence Laboratory, Cambridge, MA, (1981).
5. Xerox Learning Research Group, "The Smalltalk-80 System," *Byte* **6**, 8 (1981), 36-48.

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

CAMBRIDGE, MASSACHUSETTS 02139

VLSI Memo No. 83-152 October 1983

Parallel Computation Using Meshes of Trees*

Tom Leighton**

ABSTRACT

The hypercube and its derivative networks(e.g., the shuffle-exchange graph and the cube-connected-cycles) have long been known to be powerful networks for parallel computation. Recently, an entirely different class of networks has been discovered that appears to rival the hypercube-based networks in usefulness. In this paper, we describe the new networks(which we call meshes of trees), and we show how they can be used to solve problems such as sorting, matrix-vector multiplication, discrete Fourier transform, transitive closure, minimum spanning tree, integer multiplication and matrix multiplication in O(logn) or $O(\log^2 n)$ steps.

*To appear in Proc. 1983 International Workshop on Graph Theoretic Concepts of Computer Science, Osnabruck, West Germany, June 1983. This work was supported by DARPA contract N00014-80-C-0622, by Air Force contract OSR-82-0326, and by a Bantrell Fellowship.

**Mathematics Department and Laboratory for Computer Science, Room NE43-836A, M.I.T., Cambridge, MA 02139; telephone (617) 253-5876.

Parallel Computation Using Meshes of Trees

(Extended Abstract)

Tom Leighton

Mathematics Department and
Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Abstract: The hypercube and its derivitive networks (e.g., the shuffle-exchange graph and the cube-connected-cycles) have long been known to be powerful networks for parallel computation. Recently, an entirely different class of networks has been discovered that appears to rival the hypercube-based networks in usefulness. In this paper, we describe the new networks (which we call *meshes of trees*), and we show how they can be used to solve problems such as sorting, matrix-vector multiplication, discrete Fourier transform, transitive closure, minimum spanning tree, integer multiplication and matrix multiplication in $O(logn)$ or $O(log^2n)$ steps.

1. Introduction

Graphs such as the hypercube, the shuffle-exchange graph [S71, S80] and the cube-connected-cycles [PV79] have long been known to be, very powerful networks for parallel computation. In fact, most of the fast parallel algorithms known for problems such as sorting and discrete Fourier transform are based on the unique structure of these networks.

Recently, an entirely different class of networks has been discovered that appears to rival the hypercube-based networks in their usefulness. The new networks are known by a variety of names (including the *orthogonal trees* and the *orthogonal forests*), but we call them *meshes of trees*. The structure inherent in meshes of trees can be found in algorithms that are up to ten years old, but the networks themselves have only recently been defined. Three groups of researchers are responsible for independently formalizing the definition of a mesh of trees:

1) Nath, Maheshwari and Bhatt [N82, NMB83] showed how the networks could be used for sorting, discrete Fourier transform, transitive closure and minimum spanning tree,

2) Cappello and Steiglitz [CS81] showed how the networks could be used for integer multiplication, and

3) Leighton [L81, L83] obtained optimal bo...ds for laying out the networks on a VLSI chip and showed how the networks could be used for sorting, discrete Fourier transform and matrix multiplication.

In this paper, we describe algorithms for all of these problems. In some cases (such as for matrix multiplication [PV80, L83] and integer multiplication [CS81]), the algorithms were known previously and we have included them for completeness. In other cases (such as for sorting and transitive closure), the algorithms are new and consume less of some resource (e.g., time, area or processor size) than did the previously known algorithms. Except for the graph problems (which take $O(log^2n)$ steps), all of the algorithms require only $O(logn)$ steps to execute.

The paper is divided into five sections. In Section 2, we define the 2-dimensional mesh of trees and attempt to provide some intuition as to why it is a good network for parallel computation. We also review the VLSI layout results that are known for the network. In Section 3, we show how the n-by-n mesh of trees can be used to sort n numbers in $O(logn)$ steps. We also show how pipelining can be used to decrease processor size, increase data rate, and decrease layout area. In Section 4, we describe algorithms for matrix-vector multiplication, Fourier transform, transitive closure, minimum spanning tree and integer multiplication. In Section 5, we discuss multidimensional meshes of trees and show how the n-by-n-by-n mesh of trees can be used to multiply two n-by-n matrices in $O(logn)$ steps. We also define the powerful shuffle-tree graph and explain why it can efficiently simulate algorithms designed for hypercube networks as well as those designed for meshes of trees.

Throughout the paper, we assume that nodes which are linked by an edge can communicate in a single time *step*. This assumption is made in many of the papers in the literature. (An exception is [NMB83], which assumes logarithmic communication time.) If longer communication times are required, then the number of steps calculated for algorithms in this paper must be scaled up accordingly.

2. The 2-Dimensional Mesh of Trees

2.1 Definition and Properties

The *2-dimensional mesh of trees* $M_{2,n}$ is constructed as follows. Starting with an n-by-n grid of nodes (where n is a power of two) and adding nodes and edges as specified, construct a complete binary tree in each row and column of the grid. The trees should be constructed so that the leaves in each tree are precisely the nodes in the corresponding row or column of the original grid. In particular, the (i, j) *grid node* (i.e., the node in the *ith* row and *jth* column of the grid) should double as the *ith* leaf of the *jth* column tree and the *jth* leaf of the *ith* row tree. (The *ith* leaf is determined by counting from left to right in the canonical drawing of the complete binary tree in the plane.)

As an example, we have drawn $M_{2,4}$ in Figure 1. The nodes in the original 4-by-4 grid are represented by dots. The nodes that were added to form row trees are drawn as small triangles while those added to form column trees are shown as small squares. The row tree edges are drawn with solid lines while dashed lines represent column tree edges.

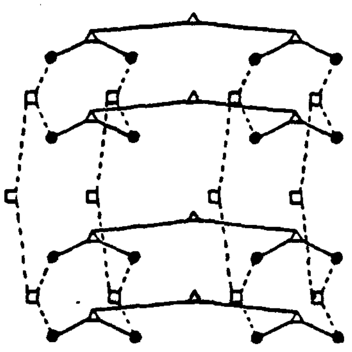


Figure 1: *The 4-by-4 mesh of trees $M_{2,4}$.*

It is not difficult to show that the n-by-n mesh of trees $M_{2,n}$ has $N = 3n^2 - 2n$ nodes, $4n^2 - 4n$ edges and maximum node degree three. More importantly, the graph has a small *diameter*; that is, every pair of nodes in the graph can be linked by a path of length $4 \log n = \Theta(\log N)$. This means that any pair of processors in a mesh of trees network can communicate in a short (logarithmic) amount of time, an indication that the network will be useful for parallel computation.

In addition to having a small diameter, the mesh of trees also has a nice recursive structure. For example, if we remove the roots of the row and column trees of $M_{2,4}$ and the edges incident to them, we are left with four copies of $M_{2,2}$, one in each quadrant of the original grid. (For example, see Figure 2.)

In general, if we remove the nodes and edges in the top k levels of the binary trees in $M_{2,n}$, we will be left with 2^{2k} copies of $M_{2,n/2^k}$. This property is important for two reasons. First, it means that the mesh of trees is particularly well suited for use with algorithms that are based on the divide-and-conquer paradigm. (This is another reason why the mesh of trees is such a powerful network for parallel computation.) Second, the fact that $M_{2,n}$ can be decomposed into four disjoint copies of $M_{2,n/2}$ by the removal of $O(n)$ nodes and edges means that $M_{2,n}$ has a $2^{1/2}$-bifurcator of size $O(n)$. (A graph is said to have a

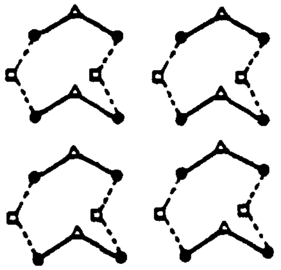


Figure 2: $M_{2,4}$ *with row and column roots removed.*

$2^{1/2}$*-bifurcator of size F* if it can be partitioned into disjoint subgraphs by the removal of F edges. The subgraphs, in turn, must have $2^{1/2}$-bifurcators of size $F/2^{1/2}$. The subgraphs need not be identical in size, but after $2\log F$ levels of recursion, the graph must be completely decomposed into isolated nodes. See [BL83, L82] for more information on bifurcators and their applications.) We will use the fact that $M_{2,n}$ has a $2^{1/2}$-bifurcator of size O(n) in Section 2.3 where we discuss VLSI layouts for the mesh of trees.

2.2 Relationship to the Complete Bipartite Graph

Just as the shuffle-exchange graph, cube-connected-cycles and related networks derive their computational power from the structure of the hypercube, the 2-dimensional mesh of trees can be seen to derive its power from the structure of the complete bipartite graph. The relationship between the n-by-n mesh of trees $M_{2,n}$ and the $2n$-node complete bipartite graph $K_{n,n}$ can best be explained by illustration. In Figure 3a, we have drawn $K_{4,4}$. In Figure 3b, we have drawn $M_{2,4}$ in a way that conforms to the structure of $K_{4,4}$. The nodes and edges of $M_{2,4}$ are drawn according to the same conventions followed in Figure 1.

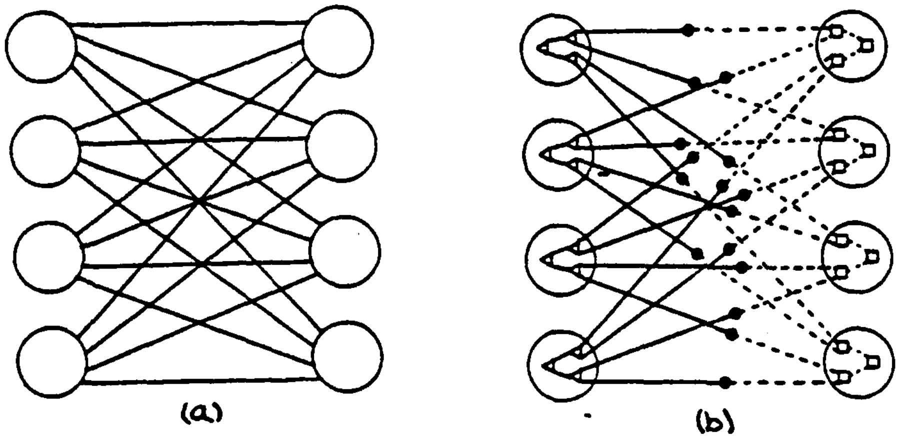


Figure 3: *a) the complete bipartite graph* $K_{4,4}$; *b) the 4-by-4 mesh of trees* $M_{2,4}$.

The illustration makes clear the correspondence between rows and columns of $M_{2,n}$ and nodes (left-hand side and right-hand side, respectively) of $K_{n,n}$, and between grid nodes of $M_{2,n}$ and edges of $K_{n,n}$. Given this correspondence, it is not at all surprising that the 2-dimensional mesh of trees is a powerful network for parallel computation.

2.3 VLSI Layouts

Although our primary emphasis in this paper is on the structural and computational properties of the mesh of trees, it is worth mentioning the results relevant to Thompson grid model layouts [T79, T80] for the mesh of trees.

A quick glance at Figure 1 suggests a natural $\Theta(n^2log^2n)$-area layout for $M_{2,n}$. This layout also has $\Theta(n^2log^2n)$ wire crossings and edges of length $\Theta(nlogn)$. For practical purposes, this layout is the simplest known and might well be optimal. Mathematically speaking, it is within a constant factor of optimal in area [L81, L83] but is suboptimal in other respects. For example, the fact that $M_{2,n}$ has a $2^{1/2}$-bifurcator of size $O(n)$ means that there is a recursively defined layout for $M_{2,n}$ that has area $O(n^2log^2n)$, $O(n^2logn)$ wire crossings and maximum edge length $O(nlogn/loglogn)$ [BL83, L82]. In [L81, L83], we prove that these bounds cannot be improved by more than a constant factor.

In addition to achieving the optimal area, crossing number and edge length bounds, the bifurcator-based layouts described in [BL83, L82] have a number of other useful properties. For example, the same bounds can be achieved for networks in which every node is replaced by an $O(logn)$-by-$O(logn)$-size processor. In fact, the area required for a layout with s-by-s-size processors is $\Theta(n^2(logn+s)^2)$. The layouts described in [BL83, L82] are also *fault-tolerant* in the sense that the same bounds can be achieved in the presence of faulty processors. Lastly, the layouts allow space for variable-size transistors to power signals across long wires. Without this feature, the assumption that communication across a long wire can be accomplished in unit time is less realistic.

Since any layout for $M_{2,n}$ has perimeter $\Omega(nlogn)$, it is possible (at least mathematically) to connect the $2n$ roots of $M_{2,n}$ to *pins* on the exterior of the layout without affecting any of the results mentioned above. This fact allows us to input data to the network through the row and column roots at a rate of $2n$ items per computational step. (For large values of n, this assumption may not be realistic given the current fabrication constraints on pincount. In such cases, the data must be entered at a slower rate and multiplexed to the row and column roots.)

3. Sorting Using the 2-Dimensional Mesh of Trees

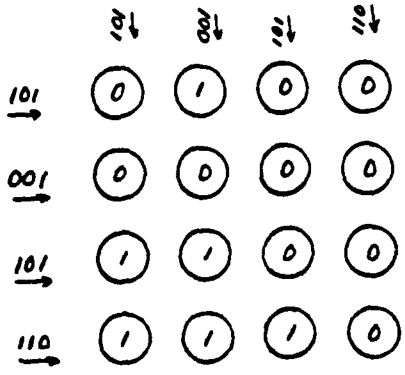
In what follows, we describe three algorithms for sorting using the mesh of trees or simple variants thereof. All three algorithms are based on a simple scheme described by Muller and Preparata in [MP75]. The algorithms essentially consist of comparing every number to every other number, computing ranks (i.e., positions in the sorted list), and then permuting the numbers according to rank. All three variants sort a list of n numbers in $O(logn)$ steps but each uses pipelining to reduce some other measure of complexity. In Section 3.1, we describe an implementation of the algorithm on the mesh of trees using processors that can only compute a constant number of 1-bit operations in a single time step (i.e., in one *bit step*). In Section 3.2, we show how to implement the algorithm to sort p lists of numbers in $O(logn+p)$ steps (but using $O(logn)$-bit-size processors), thereby increasing the data rate of the computation. In Section 3.3, we show how to implement the algorithm on a simplified mesh of trees in order to decrease the area necessary to sort n numbers in $O(logn)$ steps. The latter algorithm is also reported in [NMB83].

3.1 Pipelining to Reduce Processor Size

In many models of computation, it is assumed that processors can perform a constant number of $O(logn)$-bit word operations in a single step. In what follows, we restrict ourselves to consider processors that can perform only a constant number of 1-bit operations in a single step. In particular, we will show how to pipeline $M_{2,n}$ in order to sort n $O(logn)$-bit numbers in $O(logn)$ bit steps. For m-bit numbers, the algorithm requires $O(logn+m)$ steps, although this is suboptimal for $m >> \Omega(logn)$.

Let $w_1, \ldots, w_n$ denote the n numbers to be sorted and let r be the rank function for the list. (Formally, $r(i)$ is the position of w_i in the largest-first sorted ordering of the list.) Starting at the roots, input w_i bit by bit (leading order bit first) into the *ith* row and column trees for each i, $1 \leq i \leq n$. Pass the bits down each tree so that after *logn* steps, the leading bit of w_i has reached each leaf of the *ith* row and column trees. At this point, the (i, j) grid node sees the leading bits of w_i and w_j. If they differ, the grid node halts and stores the value 1 if $w_i > w_j$ and 0 otherwise. If the leading bits of w_i and w_j are identical, they are discarded and the (i, j) grid node next compares the 2nd leading bits. Comparison of w_i and w_j continues in this fashion until they are distinguished or until they are found to be identical on all the bits. If equal, then the grid node halts and stores the value 1 if $i > j$ and 0 otherwise.

After a total of $logn+m$ steps, every grid node has halted and stored a value indicating whether or not the number entered into its row tree was larger than the number entered into its column tree. For example, if the four numbers to be sorted on $M_{2,4}$ were (in binary) 101, 001, 101 and 110 (in that order), the values stored in the grid nodes would be as shown in Figure 4.



	101 ↓	001 ↓	101 ↓	110 ↓
101 →	0	1	0	0
001 →	0	0	0	0
101 →	1	1	0	0
110 →	1	1	1	0

Figure 4: *The values stored in the grid nodes of $M_{2,4}$ after all comparisons are completed.*

Notice that $r(j)$ can be found by summing the values stored in the *jth* column tree. (For reasons that will soon become apparent, we require that the values of $r(j)$ range from 0 to n-1 instead of from 1 to n.) This is because the *ith* leaf of the *jth* column tree contains a 1 precisely when $w_i > w_j$ or when $w_i = w_j$ and $i > j$. These values can in fact be summed in *2logn* steps as follows. At each step, each node in the tree stores a bit and transmits a bit to its father in the tree. The *transmitted bit* (or *parity bit*) is the least significant bit of the sum of the bit stored in the last step and the two bits being transmitted from its sons. The *stored bit* (or *carry bit*) is the most significant bit of the same sum. A node starts transmitting bits only after it receives transmitted bits, and stops once it has transmitted its carry bit and its sons have stopped transmitting. (The single exception to this rule is that the root never transmits its last carry bit, which is necessarily a 0.) Initially, the stored bits of non-leaf nodes are 0. The algorithm commences when the leaves transmit their stored values.

As an example, we have shown the sequence of steps taken by the 2nd column tree in Figure 5. Numbers inside the nodes indicate stored bits. Numbers on the edges indicate transmitted bits. Nodes are marked with X's after their last transmission. After *2logn* steps, the sum is output bit by bit (least significant bit first) at the root (where it may be stored in a *logn*-length stack of nodes).

After *3logn* + *m* steps, the *ith* column root contains w_i and $r(i)$. It remains only to route w_i to the $r(i)$*th* row root for each i in order to complete the sorting. The

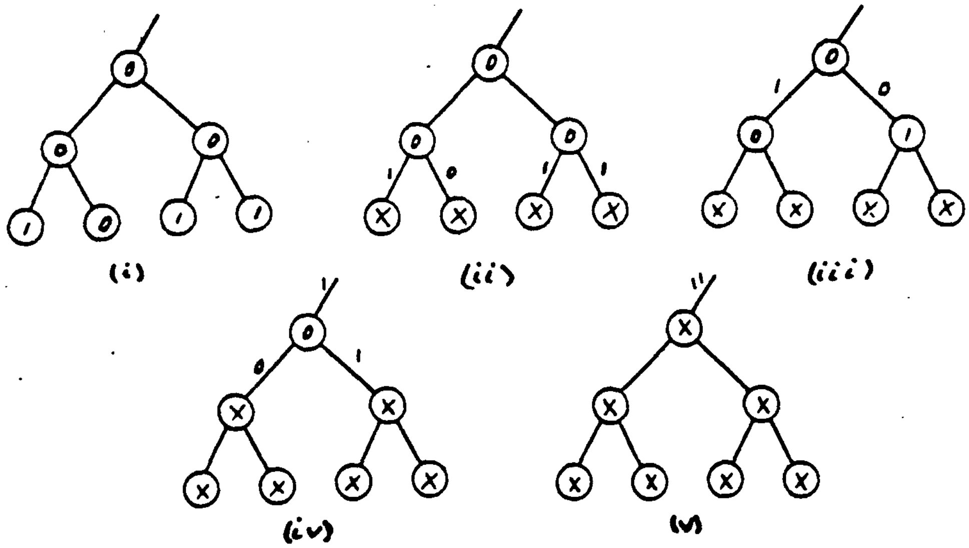


Figure 5: *Sequence of steps used :o sum the values in the leaves of the 2nd column tree.*

routing of w_i is accomplished by first paving a trail from the root of the *ith* column tree to the $r(i)$*th* leaf of that tree and then from that leaf (which is also in the $r(i)$*th* row tree) to the root of the $r(i)$*th* row tree.

The algorithm to accomplish this task is quite simple. In the first step, the root of the *ith* column tree observes the leading order bit of $r(i)$. If the bit is 0, then the node passes all remaining bits of $r(i)$ and w_i to its left son (i.e., the son that is closer to the top row). If the bit is 1, then the node passes all remaining bits to its right son. In either case, the leading bit is discarded (i.e., it is not passed on to either son). The nodes in lower levels of the tree act the same way. They observe (and discard) the first bit that they see. If it is 0, remaining bits are passed to the left son. Otherwise, the remaining bits are passed to the right son. It is not difficult to check that the $r(i)$*th* leaf (and no other leaf) of the *ith* column tree starts seeing w_i after *2logn* steps. (It should now be clear why we chose to express $r(i)$ in the range from 0 to n-*1*.) The leaf immediately passes the bits of w_i on to its father in the row tree, as do all the nodes in the row trees, until the last bit of w_i reaches the root of the $r(i)$*th* row tree. This happens after $logn+m$ additional steps. Since we have insured that $r(i) \neq r(j)$ for $i \neq j$, there will not be any conflicts in the row trees, and $w_1, \ldots, w_n$ will appear in sorted order in the row roots after a total of $6logn+2m$ steps. When $m=O(logn)$, the running time is thus $O(logn)$ bit steps.

3.2 Pipelining to Increase Data Rate

If the processors used in the mesh of trees are allowed to perform a constant number of $O(logn)$-bit word operations in a single time step, then the sorting algorithm described in Section 3.1 can be greatly simplified. In particular, it is no longer necessary to split up numbers into bits for pipelining. Rather, numbers from successive lists to be sorted can be pipelined so as to increase the number of lists that can be sorted in O(*logn*) steps. We describe how this can be done in what follows.

Let p denote the number of n-element lists to be sorted. At the first time step, enter the *ith* number of the first list into the roots of the *ith* row and column trees for all i. At the second time step, transmit these numbers to the sons of the roots and enter the numbers from the second list into the roots. At each subsequent step, continue processing the previously entered lists and enter the data from the next list to be sorted into the roots. At any time step, processors on any level will be handling data from at most three lists: one list that is being passed to the leaves for comparison, one list for which the ranks are being computed by summing, and one list for which the final routing is taking place. Once a list is entered into the network, it will be output after *4logn* steps. Hence p lists can be sorted in *4logn+p* steps. This is significantly faster than the O(*plogn*) steps required by the bit-based algorithm described in Section 3.1. (Of course, the processors are larger too.)

3.3 Pipelining to Reduce Layout Area

For some applications, decreasing the area of the layout is more important than increasing the number of problems that can be solved in O(*logn*) steps. In such cases, it is useful to sort using a simplified mesh of trees. The *n-by-n simplified mesh of trees* $S_{2,n}$ is constructed from $M_{2,n}$ by first removing the internal nodes and edges from all but every (*logn*)*th* row and column tree, and then inserting edges so that where there was once a copy of $M_{2,logn}$ in $M_{2,n}$, there is now a *logn*-by-*logn* mesh with a tree in the top row and leftmost column. For example, $S_{2,4}$ is shown in Figure 6. (This network has been discovered independently by many individuals including Nath, Maheshwari and Bhatt who call it the *orthogonal tree cycles* [NMB83].)

It is not difficult to construct a layout for $S_{2,n}$ that has $\Theta(n^2)$ area, $\Theta(n^2)$ crossings and edges of length O(n). Interestingly, the simplified mesh of trees can sort a list of n numbers in O(*logn*) steps. (We assume, of course, that the processors are capable of performing O(*logn*)-bit word operations in a single step.) Given a list of numbers to be sorted $w_1, \ldots, w_n$, the algorithm proceeds as follows.

At the first time step, w_{ilogn} is input to the roots of the *ith* row and column trees. In the next time step, these values are passed on to the sons of the roots and $w_{ilogn-1}$ is input to the roots of the *ith* row and column trees. This process continues for *logn* steps whereupon $w_{ilogn-k}$ is in every (*logn* - k)-level node of the

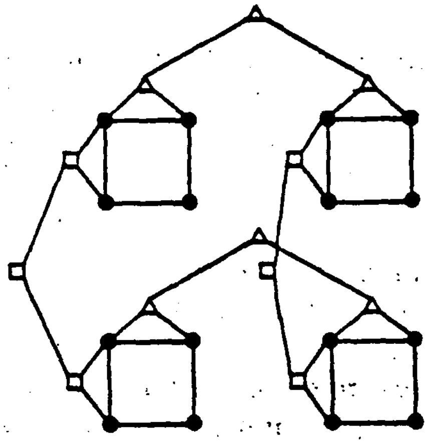


Figure 6: *The 4-by-4 simplified mesh of trees $S_{2,4}$.*

ith row and column trees. During the next *logn* steps, the data is passed to the grid nodes through the mesh edges. After a total of *2logn* steps, the *(i, j)* grid node will thus contain both w_i and w_j. All pairwise comparisons take place simultaneously and the values are passed through the meshes to the trees for summing. After *logn* additional steps, the rank of $w_{(i-1)logn+1}$ appears at the root of the *ith* column tree. The routing of these numbers can now commence. After another *logn* steps, every rank has been computed and the numbers are in various levels of the column trees. After *logn* more steps, all the numbers have reached the appropriate grid nodes. Starting simultaneously, the numbers begin moving in the other direction, first to leaves of the row trees and then on to the roots of the row trees. Eventually, the numbers will appear in sorted order in blocks of *logn* numbers at each of the *n/logn* row roots. The total running time is *8logn* word steps.

It is worth remarking that (like the bit-based algorithm described in Section 3.1) the sorting algorithm for the simplified mesh of trees cannot be further pipelined to sort several lists simultaneously.

4. Other Algorithms for the 2-Dimensional Mesh of Trees

In what follows, we briefly describe a variety of fast algorithms for the 2-dimensional mesh of trees. We commence with an O(*logn*)-step algorithm for matrix-vector multiplication in Section 4.1. This algorithm (which originally appeared in [L83, NMB83]) has applications to Fourier transform and convolution as well as to a variety of other problems. In Section 4.2, we describe algorithms for the transitive closure and minimum spanning tree problems. We conclude in Section 4.3 with an algorithm for integer multiplication on a mesh-of-trees-like structure that was developed by Cappello and Steiglitz [CS81].

4.1 Matrix-Vector Multiplication

Given a fixed n-by-n matrix $S=\{s_{ij} \mid 1 \leq i, j \leq n\}$, we will show how to use $M_{2,n}$ to compute the product of S and any input n-vector in $2logn$ steps. As S is fixed, it is not considered to be part of the on-line input. Rather, it is considered to be part of the program (in the form of off-line input) and we assume that the value of s_{ij} is initially stored in the (i, j) grid node for all i and j. The algorithm proceeds as follows.

Given any input vector $v = \{v_j \mid 1 \leq j \leq n\}$, input v_j into the root of the *jth* column tree for each j at the first time step. Pass the entries of v down the column trees so that after *logn* steps, each leaf in the *jth* column tree has received the value of v_j. Computation of the n^2 products $\{s_{ij}v_j \mid 1 \leq i, j \leq n\}$ can now take place simultaneously. Afterwards, we can find the values of the product vector Sv by summing the values of the leaves in each row tree. This summing operation takes an additional *logn* steps. Thus after a total of *2logn* steps, the values of the product are output at the roots of the row trees.

The form of the algorithm for matrix-vector multiplication is very similar to that for sorting. Hence, it should not be surprising that the algorithm just described can be pipelined in the three ways described in Section 3. For example, the product of S with p n-vectors can be computed in $2logn+p$ steps and the simplified mesh of trees can be used to compute a single product in *4logn* steps (thereby reducing the area). Reducing the size of the processors is slightly more complicated since multiplication is harder than comparison. The difficulty can be overcome by expanding each grid node into a small *logn*-bit multiplier.

The matrix-vector product algorithm has a number of useful applications. Most importantly, it can be used to compute discrete Fourier transforms by setting S to be the well-known discrete Fourier transform matrix. As a result, the mesh of trees can compute convolutions, interpolations and polynomial products as well as a variety of other tasks in $O(logn)$ steps.

4.2 Graph Problems

In [NMB83], Nath, Maheshwari and Bhatt, describe an $O(log^3n)$-step algorithm for the connected components problem. They also show how to modify the algorithm to obtain $O(log^3n)$-step algorithms for the transitive closure and minimum spanning tree problems. The algorithms are based on the Hirschberg-Chandra-Sarwate [HCS79] $O(log^2n)$-time algorithm for connected components, which uses a shared-memory model of parallel computation. In what follows, we show how to modify the [HCS79] algorithm so that it can be executed on an n-by-n mesh of trees in $O(log^2n)$ steps. (A similar modification is described in [HV83].) By following the techniques described in [N82, NMB83], the algorithm can then be modified to obtain $O(log^2 n)$-step algorithms for the transitive closure and minimum spanning tree problems.

Given and undirected n-node graph G, let $A = \{a_{ij} \mid 1 \leq i, j \leq n\}$ be the adjacency matrix for G and let D be an array such that $D(i)$ is the smallest number of a node in the connected component of G containing node i. Assume that $a_{ii}=1$ for all i. The [HCS79] algorithm for computing D proceeds as follows.

```
op1: set  $D(i) = i$  for all $i$
op2: do op2.1 through op2.4 for $logn$ iterations
    op2.1: set  $E(i) = \min_j \{D(j) \mid a_{ij}=1\}$  for all $i$
    op2.2: set  $C(i) = \min_j \{E(j) \mid D(j)=i\}$  for all $i$
    op2.3: do op2.3.1 for $logn$ iterations
        op2.3.1: set  $C(i) = C(C(i))$  for all $i$
    op2.4: set  $D(i) = C(D(i))$  for all $i$
op3: end
```

It is easily shown (see [NMB83], for example) that op2.1, op2.2, op2.3.1 and op2.4 can be implemented in $O(logn)$ steps on an n-by-n mesh of trees. The problem is that op2.3.1 is executed $log^2 n$ times by the preceding algorithm. This problem can be overcome by implementing the following algorithm on the mesh of trees.

```
op1: set  $D(i) = C(i) = i$  and  $S(i) =$ "active"  for all $i$
op2: do op2.1 through op2.6 for $10logn$ iterations
    op2.1: set  $E(i) = \min_j \{D(j) \mid a_{ij}=1$ and $S(D(j))=$"active"$\}$  for all $i$
    op2.2: if $S(C(i)) =$ "active", then set $C(i) = \min_j \{E(j) \mid D(j)=i\}$ for
           all $i$
    op2.3: if  $C(i) \neq i$, then set  $S(C(i)) = S(i) =$ "inactive"  for all $i$
    op2.4: set  $C(i) = C(C(i))$  for all $i$
    op2.5: if  $C(r) = r$  and no new values of $C(i)$ were set to $r$ in op2.4,
           then set  $S(r) =$ "active"  for all $r$
    op2.6: if  $S(C(D(i))) =$ "active", then set  $D(i) = C(D(i))$  for all $i$
op3: end
```

The key difference in the two algorithms is that the latter algorithm processes the necessary number of iterations of op2.3.1 in parallel with the rest of the operations. The $S(j)$ variables are included to insure that the functioning of op2.3.1 (op2.4 in the latter algorithm) doesn't conflict with the other operations. (In particular, $S(j)$ is "inactive" precisely when the label j is being processed by op2.4 and thus when it is not available for processing by the other operations.) The proof that this algorithm performs as claimed is somewhat delicate and is described in [HV83] so we have not included it here. The proof that the algorithm can be implemented on $M_{2,n}$ using $O(log^2n)$ steps is not difficult (especially given [NMB83]) and we leave it as an exercise for the reader.

As was the case with sorting and matrix-vector multiplication, these algorithms can be pipelined in the ways described in Section 3.

4.3 Integer Multiplication

In [CS81], Cappello and Steiglitz show how to use a variant of the mesh of trees structure to multiply two n-bit numbers in $O(logn)$ bit steps. We briefly summarize this result in what follows.

The network for integer multiplication is constructed from an n-by-$2n$ grid of nodes. Nodes and edges are added to form complete binary trees in the rows, columns and transverse diagonals of the grid. As before, the leaves of the trees should coincide with the nodes of the grid. As an example, we have indicated the location of the diagonal trees for $n=4$ in Figure 7.

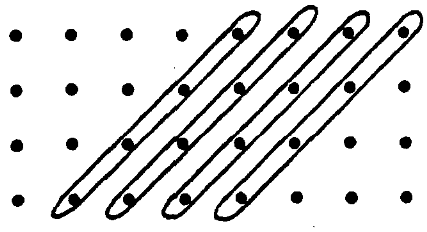


Figure 7: *Location of diagonal trees in a 4-by-8 grid of nodes.*

Let $a_1 \cdots a_n$ and $b_1 \cdots b_n$ be the binary representations of the two numbers to be multiplied. At the first step, input a_i to the root of the $(n-i+1)st$ row tree for each i and input b_i to the root of the ith diagonal tree for each i. Pass the values from the roots to the leaves of the trees so that after $logn$ steps, the $(n-i+1, n-i+1+j)$ grid node contains a_i and b_j. These values are then simultaneously multiplied and stored. For example, Figure 8 displays the stored values for the product of 1101 and 1011.

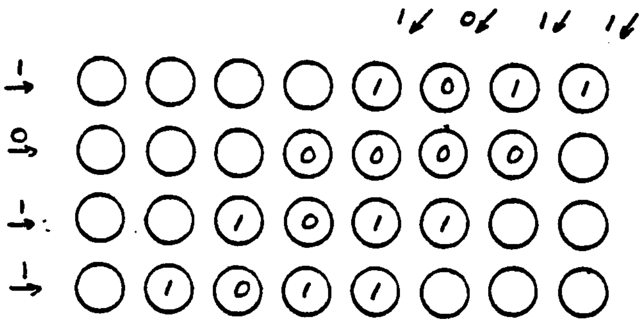


Figure 8: *Stored values for the product of 1101 and 1011.*

It is obvious from looking at Figure 8 how the algorithm multiplies two numbers. The algorithm is simply simulating the elementary school method of digital multiplication. At this point it remains only to sum the *n* $2n$-bit numbers contained in the rows. This is done in log^*n stages as described below.

In the first stage, the bits in each column are summed as described in Section 3.1 and then stored in a *logn*-length stack at the row root. This stage takes *2logn* steps and reduces the problem to that of adding *logn* $2n$-bit numbers. The new problem can be handled in a recursive fashion. For example, in the second stage, the algorithm takes *2loglogn* steps and leaves *loglogn* $2n$-bit numbers to be summed. After log^*n stages (O(*logn*) steps), we are left with 2 $2n$-bit numbers to be summed. This final sum is computed with a Brent-Kung adder [BK80] in O(*logn*) steps. (Note that the Brent-Kung network must be interconnected to the roots of the mesh of trees to form the multiplication network. By further modifying the network, the algorithm can be pipelined in the ways described in Section 3.)

5. Multidimensional Meshes of Trees

5.1 Definitions and Properties

The 2-dimensional mesh of trees can be easily generalized to higher dimensions. For example, the *3-dimensional n-by-n-by-n mesh of trees* $M_{3,n}$ can be constructed as follows. Starting with an *n*-by-*n*-by-*n* cube of nodes and adding nodes where indicated, construct a set of n^2 complete binary trees in each of the three dimensions of the cube. As before the tree should be constructed so that the leaves are precisely the nodes of the original cube and so that the subgraph induced on each octant of nodes is $M_{3,n/2}$. For example, we have drawn $M_{3,2}$ in Figure 9. The nodes in the original cube of nodes appear as dots while the internal nodes appear as squares.

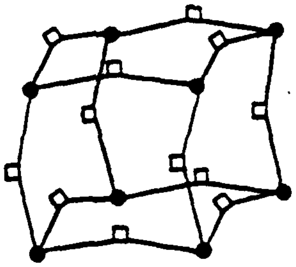


Figure 9: *The 2-by-2-by-2 mesh of trees $M_{3,2}$.*

The general *r-dimensional mesh of trees* $M_{r,n}$ is formed from an $\overbrace{n\text{-by-}n\text{-}\cdots\text{-}n}^{r}$ hypercube in a similar manner. In general, removal of the roots and edges that are in the top level of the binary trees will leave 2^r disjoint copies of $M_{r,n/2}$. It is easily shown that $M_{r,n}$ has $N = \Theta(rn^r)$ nodes, maximum degree r, diameter $2r\log n = \Theta(\log N)$, and a $2^{1/2}$-bifurcator of size $n^{r-1} = \Theta(N^{1-1/r})$. Optimal layouts for $M_{r,n}$ are discussed in [L81, L83].

5.2 Matrix Multiplication Using the 3-Dimensional Mesh of Trees

In what follows, we describe a $2\log n$-step algorithm for multiplying two n-by-n matrices using $M_{3,n}$. This algorithm was originally discovered by Preparata and Vuillemin [PV80], although the underlying network was not discovered to be the mesh of trees until recently. The algorithm can be pipelined in several ways (see [PV80]), including those ways described in Section 3. In what follows, we review the simplest version of the algorithm.

At the first time step, the two matrices to be multiplied are entered into the network via the roots of the trees in two of the dimensions (one dimension for each matrix). The entries are passed down through the trees so that after $\log n$ steps, the (i, j, k) grid node contains the (i, j) entry of the first martix and the (j, k) entry of the second matrix. All n^3 multiplications are then performed simultaneously. The entries of the product matrix are then calculated by summing the values of the leaves of each tree in the third (previously unused) dimension. The summing process takes an additional $\log n$ steps. The total computation takes $2\log n$ steps.

5.3 The Shuffle-Tree Graph

The r-dimensional mesh of trees was defined as a natural extension of the 2-dimensional mesh of trees. $M_{r,n}$ can also be viewed as a generalization of the r-cube, another powerful communications network. For example, $M_{r,2}$ is an r-cube with every edge replaced by a path of length two. (Glance at Figure 9 once again.) Viewed in this light, the r-dimensional mesh of trees motivates the definition of a *shuffle-tree graph* in much the same way that the r-cube motivates the definition of the 2^r-node shuffle-exchange graph. In what follows, we review the transformation of the hypercube into a shuffle-exchange graph and show how the same transformation can be applied to $M_{r,n}$ to form a shuffle-tree graph.

Because the r-cube has nodes of degree r, it is sometimes not appropriate for practical applications. In such cases, the shuffle-exchange graph is often used instead. The 2^r-node shuffle-exchange graph is formed from the (2^r-node) r-cube by removing all edges except those that link nodes differing in the last bit, and then inserting edges between nodes that are left or right cyclic 1-shifts of one another. The edges from the original r-cube are called *exchange edges* and the edges inserted between nodes that are 1-shifts of one another are called *shuffle edges* (owing to their ability to shuffle a deck of data in a single step [DGK81, LLM83]). Mathematically speaking, the shuffle edges are formed by rotating the nodes of the r-cube in r dimensions about the line between the all-0 node and the all-1 node. The rotation permutes the nodes in r-cycles which correspond to r-cycles of the shuffle edges. (When r is composite, degenrate cycles with fewer than r edges may appear.) As an example, we have included Figure 10. (Those readers who would like to know more about the properties of shuffle-exchange graphs might find [LLM83] of interest.)

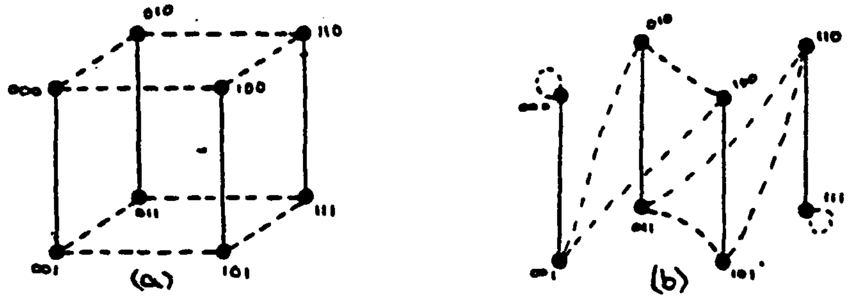


Figure 10: *a) the 3-cube (dashed edges are to be removed); b) the 8-node shuffle-exchange graph (dashed edges are shuffle edges).*

If the edges of the hypercube are used in only one dimension at a time by an algorithm and if the dimensions are used in the natural cyclic order (as they are for almost all hypercube algorithms), then the same algorithms can be implemented on the shuffle-exchange graph by passing the data along shuffle edges between each hypercube operation. Since passing the data along shuffle edges corresponds to a rotation of the underlying hypercube structure of the graph, the exchange edges of the shuffle-exchange graph effectively simulate the currently active dimension of edges in the hypercube. As a result, the time necessary to run the algorithm on the shuffle-exchange graph is at most double the time required by the hypercube. The advantage of the shuffle-exchange graph is that it has node degree at most three.

The (r, n)-*shuffle-tree graph* $T_{r,n}$ is constructed from $M_{r,n}$ by first removing all but one dimension of the trees, and then by adding edges that correspond to a rotation of $M_{r,n}$ in r-space about the line between nodes $(1, \ldots, 1)$ and $(n, \ldots, n)$. Two nodes $(u_1, \ldots, u_r)$ and $(v_1, \ldots, v_r)$ of the original n^r-node cube are leaves of the same tree in $T_{r,n}$ if $u_i = v_i$ for all $i < n$, and they are linked by a shuffle edge if one is a cyclic 1-shift of the other. For example, the tree and shuffle edges of $T_{2,4}$ are shown in Figures 11a and 11b, respectively. Notice that the shuffle edges correspond to a simple transposition of the nodes. That is because a transposition is simply a rotation in 2-space about the line between $(1,1)$ and (n,n).

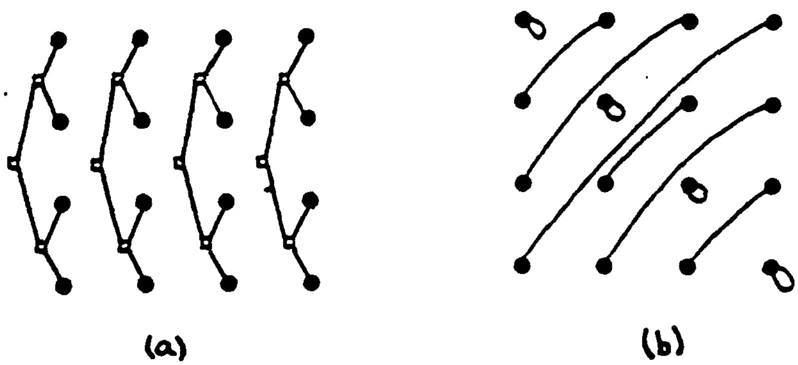


Figure 11: *a) the tree edges of* $T_{2,4}$; *b) the shuffle edges of* $T_{2,4}$.

It is worth noting that *all* of the algorithms described in this paper for $M_{r,n}$ can be run in twice the number of steps on $T_{r,n}$. This is due to the fact that all the algorithms perform operations along one dimension of trees at a time. When that dimension changes, the data is simply passed along the shuffle edges in $T_{r,n}$. As with the shuffle-exchange graph, however, $T_{r,n}$ has node degree at most three.

As we have just seen, the class of (r, n)-shuffle-tree graphs is very powerful. At one end of the spectrum $(n=2)$, the class includes the class of shuffle-exchange graphs, for which many fast algorithms are known [S71, S80]. At the other end of the spectrum $(r=2,3)$, the class includes the 2- and 3-dimensional meshes of trees, for which many good algorithms are also now known. Whether or not the graphs in the center of the spectrum are useful (we suspect that they are) is an interesting open question.

Acknowledgments

We are grateful to Sandeep Bhatt, Dan Hirschberg, Charles Leiserson, Margaret Lepley, Gary Miller, Franco Preparata and Clark Thompson for their helpful remarks.

References

[BK80] R. P. Brent and H. T. Kung, "The chip complexity of binary arithmetic," *Proceedings of the 12th Annual ACM Symposium on Theory of Computing*, April 1980, pp. 190-200.

[BL83] S. N. Bhatt and F. T. Leighton, "A framework for solving VLSI graph layout problems," *Journal of Computer and System Sciences*, to appear.

[CS81] P. R. Cappello and K. Steiglitz, "Area-efficient VLSI structures for multiplying at clock rate," Technical Report #289, Department of EECS, Princeton University, September 1981.

[DGK81] P. Diaconis, R. L. Graham and W. M. Cantor, "The mathematics of perfect shuffles," unpublished manuscript, Bell Labs, 1981.

[HCS79] D. S. Hirschberg, A. K. Chandra and D. V. Sarwate, "Computing connected components on parallel computers," *Communications of the ACM*, Vol. 22, August 1979, pp. 461-464.

[HV83] D. S. Hirschberg and D. Volper, "A parallel solution for the minimum spanning tree problem," *Proceedings of the 1983 Johns Hopkins Conference on Information Science and Systems*, pp. 680-684.

[L81] F. T. Leighton, "New lower bound techniques for VLSI," *Proceedings of the 22nd Annual IEEE Symposium on Foundations of Computer Science*, October 1981, pp. 1-12. (To appear in *Math Systems Theory.*)

[L82] F. T. Leighton, "A layout strategy for VLSI which is provably good," *Proceedings of the 14th Annual ACM Symposium on Theory of Computing*, May 1982, pp. 85-98. (To appear as [BL83].)

[L83] F. T. Leighton, *Complexity Issues in VLSI: Optimal Layouts for the Shuffle-Exchange Graph and Other Networks*, MIT Press, Cambridge Mass., to appear.

[LLM83] F. T. Leighton, M. Lepley and G. L. Miller, "Layouts for the shuffle-exchange graph based on the complex plane diagram," *SIAM Journal of Algebraic and Discrete Methods*, to appear.

[MP75] D. E. Muller and F. P. Preparata, "Bounds to complexities of networks for sorting and for switching," *Journal of the ACM*, Vol. 22, No. 2, April 1975, pp. 195-201.

[N82] D. Nath, "Efficient VLSI networks and parallel algorithms based on them," Ph.D. dissertation, Deptartment of Electrical Engineering, Indian Institute of Technology, New Delhi, February 1982.

[NMB83] D. Nath, S. N. Maheshwari and P. C. P. Bhatt, "Efficient VLSI networks for parallel processing based on orthogonal trees," *IEEE Transactions on Computers*, Vol. C-32, No. 6, June 1983, pp. 569-581.

[PV79] F. P. Preparata and J. E. Vuillemin, "The cube-connected-cycles: a versatile network for parallel computation," *Proceedings of the 20th Annual IEEE Symposium on Foundations of Computer Science*, October 1979, pp. 140-147.

[PV80] F. P. Preparata and J. E. Vuillemin, "Area-time optimal VLSI networks for matrix multiplication," *Proceedings of the 14th Princeton Conference on Information Science and Systems*, 1980.

[S71] H. S. Stone, "Parallel processing with the perfect shuffle," *IEEE Transactions on Computers*, Vol. C-20, No. 2, February 1971, pp. 153-161.

[S80] J. T. Schwartz, "Ultracomputers," *ACM Transactions on Programming Languages and Systems*, Vol. 2, No. 4, October 1980, pp. 484-521.

[T79] C. D. Thompson, "Area-time complexity for VLSI," *Proceedings of the 11th Annual ACM Symposium on Theory of Computing*, May 1979, pp. 81-88.

[T80] C. D. Thompson, *A Complexity Theory for VLSI*, Ph.D. Thesis, Department of Computer Science, Carnegie-Mellon University, 1980.

Silicon-On-Insulator Bipolar Transistors

M. RODDER AND D. A. ANTONIADIS, MEMBER, IEEE

Abstract—**Thin-film lateral n-p-n bipolar transistors (BJT) have been fabricated in moving melt zone recrystallized silicon on a 0.5-μm silicon dioxide substrate thermally grown on bulk silicon. Current-voltage characteristics of devices with different base widths (5 and 10 μm) have been analyzed. The use of a metal gate over oxide covering the base region has allowed the devices to be operated as n-channel MOSFET's as well thus surface effects on device characteristics have been investigated under varying gate-bias voltages. Maximum dc current gain values of 2.5 were achieved with a 5-μm base width and values around 0.5 with a 10-μm base width. Higher gain values were impeded by onset of high-level injection which occurred at low currents because of light base doping of these devices.**

TO DATE silicon-on-insulator (SOI) technology has found application only in MOSFET device fabrication. This is because of the simplicity of these devices and also because the extremely low-minority lifetimes in the dominant SOI films, namely silicon on sapphire (SOS), have made bipolar devices impractical. However, recent SOI technology based on moving melt-zone recrystallization has yielded films with significantly improved lifetimes [1]. SOI bipolar devices may be desirable, either by themselves or in combination with MOSFET's in the same circuit, because they exhibit higher transductance for a given area and bias current than MOSFET's. This paper reports the first realization of silicon-on-silicon-dioxide bipolar devices.

The samples were prepared using the nonseeded moving melt-zone recrystallization technique which has been reported elsewhere [2]. The insulator material underlying the 0.5-μm Si film was 0.5 μm of thermally grown SiO_2 on ⟨111⟩ oriented n-type doped silicon wafers. After recrystallization, the encapsulation layer was removed and a 500-Å buffer oxide layer was grown at 1000°C in 50 min. Next, a double boron implantation was performed over the entire sample with doses of 1.5×10^{12} cm^{-2} at 20 keV and 3.5×10^{12} cm^{-2} at 200 keV; this implant determined the base doping profile. Following the implant, the base regions were covered with a polyimide layer defined using a O_2 plasma etching process. A phosphorous implantation to form the emitter and collector regions was performed with a dose of 3.5×10^{15} cm^{-2} at 250 keV. The polyimide layer was then removed and an Al layer was patterned to be used at a mask during the subsequent SF_6 plasma etch of the recrystallized Si, so as to form individual Si islands. The 500-Å oxide was then removed and a new 1500-Å gate oxide was thermally grown at 900°C using a dry-wet-dry oxidation. Contact holes were then defined and, subsequently, Al was deposited and patterned to define the contact pads and gate electrodes. The samples were finally annealed in forming gas at 450°C for 30 min. A cross section and top view of the merged lateral BJT-MOSFET device is shown in Fig. 1.

The base (channel) dimension perpendicular to current flow was nominally 100 μm. Nominal base widths (channel lengths) were 5 and 10 μm. After measuring the actual base widths on the sample and accounting for lateral diffusion, the widths are estimated to be approximately 3.5 and 8.5 μm.

Measurements of base and collector currents (I_B and I_C) versus base-emitter voltage V_{BE} and as a function of top gate-to-base voltage V_{GB} were obtained using a two-channel Keithley 619 Electrometer incorporated into an automated data acquisition system. Values of oxide thickness and of the MOSFET threshold voltage were also obtained using the above

Manuscript received February 16, 1983; revised April 6, 1983. This work was supported by DARPA under Grant N0014-C-80-0622.

The authors are with the Massachusetts Institute of Technology, Cambridge, MA 02139.

0193-8587/83/0600-0193$01.00 © 1983 IEEE

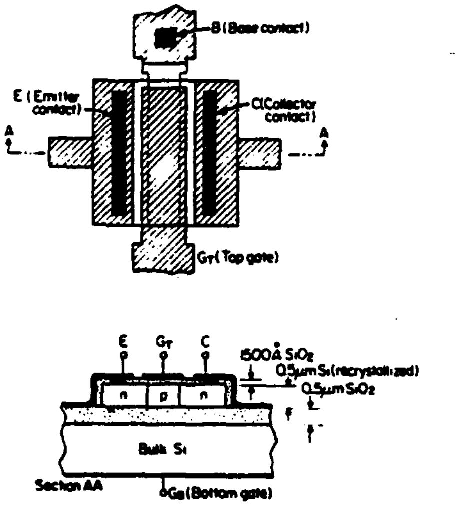


Fig. 1. Cross section and top view of the lateral bipolar-merged MOSFET device.

system and a capacitance meter; typical values are t_{ox} = 1500 Å, and V_T = 4 V at zero source-bulk (emitter-base) bias. Current-voltage characteristics are shown in Fig. 2 for typical 5- and 10-μm base-width devices. All measurements shown were performed with collector-base bias voltage V_{CB} = 0 V to avoid any base-width modulating effects, and back-gate (Si substrate) to base bias voltage equal to −10 V. A small number of devices were found to exhibit poorer characteristics than those shown in Fig. 2. Given the fact that the silicon recrystallization was unseeded, we attribute these poor characteristics to inevitable large-angle grain boundaries through those devices [3]. No significant performance difference between the majority of devices oriented parallel or perpendicular to the predominant sub-boundary direction was observed.

Two characteristics of the collector current can be seen from Fig. 2(a): (a) for sufficiently negative values of V_{GB} (−3 V), I_C is independent of variations of V_{GB}; (b) as V_{GB} increases, I_C shows a strong dependence on the gate-to-body bias voltage. Since we are primarily interested in the bipolar characteristics of the thin-film device, it is important to understand which of the I_C curves are strictly due to lateral bipolar action.

The distinguishing characteristics of a true (lateral) bipolar device are two-fold. First, if the carrier quasi-Fermi levels are assumed constant through the base-emitter junction depletion region, then the collected bipolar current exhibits a characteristic of the form $I_C \alpha \exp [qV_{BE}/nkT]$, where n = 1 for low, and n = 2 for high-level injection conditions. Hence, on a semi-log plot of I_C versus V_{BE}, for low-level injection the slope should correspond to the well-known 60-mV/decade at room temperature. Second, the collected bipolar current is due to the flow of injected minority carriers through a base region which satisfies the quasi-neutrality condition—i.e., the net charge density in the base ρ_b satisfies the condition $\rho_b \ll qN_A$ where N_A is the base-doping concentration. Therefore, if a component of collected current is due to injection of minority carriers into a region which does not satisfy the quasi-neutrality condition, then this component is not due to true bipolar transistor action but rather to field-effect action. Such a current component might be due to injection into a depleted region at the top or bottom Si-SiO_2 interface.

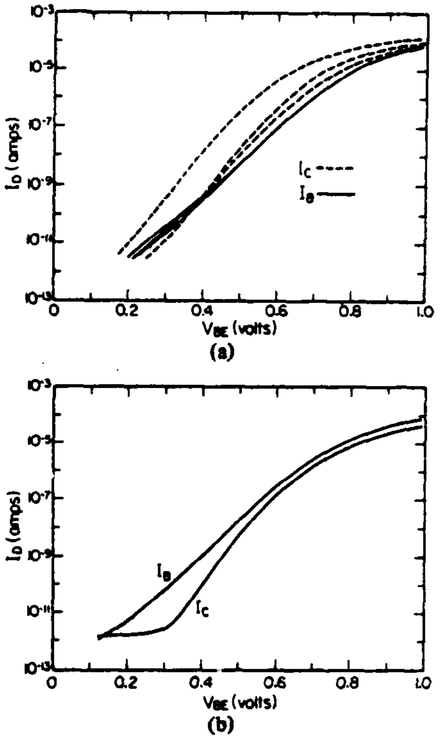


Fig. 2. (a) Typical I-V characteristic for a short-channel length device, (b) Typical I-V characteristic for a long-channel length device. All measurements at V_{CB} = 0 V and −10 V bias of back gate with respect to the emitter (source).

Thus to assure that the measured I_C is strictly due to bipolar action, it is important that the base near both the top and bottom Si-SiO_2 interface is biased into "flat-band" condition. If these regions are depleted then what we are observing is actually a surface MOSFET operating in weak inversion in parallel with the bipolar transistor. It is interesting to note that under these conditions, simple theory [4] predicts the slope of the drain current with respect to V_{BE} (this would be V_{BS} in MOSFET convention), is also 60 mV/decade as for the bipolar and thus as can be seen from Fig. 2(a), the slope cannot be used to distinguish between MOSFET and BJT operation. Since for electrons emitted into a depleted region there is very little recombination the apparent dc current gain β under this condition can be very high. Thus of the I_C curves in Fig. 2(a), the ones that correspond to true bipolar action are the ones for which V_{GB} assures neutral or accumulated surface condition. In our case this occurs around V_{GB} = −3 V. Below this value, the surface starts accumulating and the effect of the V_{GB} is screened out of the bulk base. Similarly the −10-V bias of the back gate with respect to the base assures that the back interface is accumulated, and, therefore, it does not present a favorable current path.

It is important to stress at this point that any other way of biasing our device would give erroneous results. For example, if the top gate was left floating or equivalently it was not there, as in most bulk lateral BJT, then fixed surface charges could deplete the surface and allow favorable surface current emission. Similarly, if the gate was biased with respect to the source rather than with respect to the bulk, a V_{BE}-dependent depletion modulation could occur with concommitant effects on the relationship between I_C and V_{BE}.

Although the collected current shows a strong dependence on V_{GB}, I_B shows only a slight dependence and only at low V_{BE} (Fig. 2(a)). This dependence is due to the fact that if the base region under the gate is depleted due to an applied bias V_{GB}, the base current will have a component arising from recombination in this depleted region. Since the magnitude of the recombination current is proportional to the depth of the depletion region, it is obvious that I_B will be a function of V_{GB} since the depth of the depleted region itself depends on V_{GB}. However, it is interesting to note that this surface effect is quite small indicating a low density of surface recombination centers at the top Si-SiO$_2$ interface.

Both the collector-current and the base-current characteristics exhibit considerable change in their respective slope at increased V_{BE}. Considering at first only the currents due to true lateral bipolar action, there are two effects both of which cause degradation of the current characteristics. First, the collector-current slope changes from 60 to 120 mV/decade due to the onset of high-level injection in the vicinity of V_{BE} = 0.7 V. From classical theory, for a uniformly doped base the onset condition is defined as

$$V_{BE} = 2\frac{kT}{q}\ln\frac{N_A}{n_i} \tag{1}$$

The value of $N_A = 1 \times 10^{16}$ cm^{-3} from the above definition corresponds reasonably well with the average base doping obtained from V_T versus V_{BE} observations and from SUPREM simulation; the fact that the base doping is not uniform perpendicularly to current flow does not significantly alter the validity of this discussion. On the other hand, the base-current slope degradation is not due to high-level injection; rather, a high series resistance of about 10 kΩ, from the edge of the base to the contact causes the actual base-emitter junction voltage to be less than the externally applied value of V_{BE}, for V_{BE} applied greater than approximately 0.75 V. Hence, for $V_{BE} > 0.75$ V, I_C also degrades because of this base series resistance effect.

Now, consider the current flow attributable to the merged MOSFET device associated with the top Si-SiO$_2$ interface. From Fig. 2(a), it can be seen that the degradation of current slope of the merged MOSFET begins not at a fixed V_{BE} but at a fixed value of collected current—interestingly, at approximately the value of I_C at which the true bipolar device enters the high-level injection region. It can be shown easily that this change of slope is due to onset of moderate inversion (i.e., end of the weak inversion) of the merged MOSFET device [5]; hence, the onset of moderate inversion for the MOSFET occurs at the same value of I_C as the onset of high-level injection for the BJT. This interesting observation can be easily shown to be predicatable from straightforward device theory [4].

It is now important to analyze the differences in I-V characteristics due to varying base widths. As can be seen by comparing Fig. 2(a) to Fig. 2(b), the values of I_B for a 5-μm base-width device is lower than the corresponding value for a 10-μm base-width device at the same V_{BE}. This is obviously due to less recombination in the narrower base. Accordingly, the dc common emitter current gain (β), increases from a value of 0.5 for the 10-μm device to above 2.5 for the 5-μm device. This indicates that the base current, for the most part, is due to bulk minority recombination. Using standard BJT theory and the two values of base width, a characteristic diffusion length (L_n) is determined to be approximately 5 μm. Assuming a value of D_n = 10 cm^2/s for the electron diffusion coefficient, we obtain an approximate value of $\tau_n = 3 \times 10^{-8}$ s for the minority-carrier lifetime (electrons).

It is not clear at this point whether the observed low lifetime is an inherent property of our thin films, or whether it is due to subsequent processing. Lifetimes in the microsecond range in SOI have been reported previously [1], although they were measured using a MOS deep-depletion capacitance recovery technique. Thus it appears that higher β's can be achieved in the future using these SOI films. Nevertheless, modifications in the device design to allow scaling of the base width to less than 1 μm, while maintaining a low base series resistance, can yield devices with β's of about 20 even with the present low lifetimes.

Acknowledgment

Many useful discussions with E. Maby and recrystallization of the samples by H. Atwater are gratefully acknowledged.

References

[1] B. Y. Tsaur, John C. C. Fan, and M. W. Geis, *Appl. Phys. Lett.*, vol. 40, p. 83, 1982.

[2] E. W. Maby, M. W. Geis, Y. L. LeCoz, D. J. Silversmith, R. W. Mountain, and D. A. Antoniadis, *Electron Device Lett.*, vol. EDL-2, p. 241, 1981.

[3] E. W. Maby and D. A. Antoniadis, *Appl. Phys. Lett.*, vol. 40, p. 691, 1982.

[4] S. M. Sze, *Physics of Semiconductors*, 1981, Ch. 8.

[5] Y. Tsividis, *Solid-State Electron.*, vol. 25, no. 11, p. 1099, 1982.

LAYING THE POWER AND GROUND WIRES ON A VLSI CHIP[*]

Andrew S. Moulton

Massachusetts Institute of Technology

Abstract

This paper presents the approach of MIT's Placement-Interconnect (PI) Project to routing noncrossing VDD and GND trees in single-layer metal. The input to the power-ground phase is a set of rectangular modules on a rectangular chip. There is one VDD pad, one GND pad, and each module has one VDD terminal, one GND terminal, and a current requirement. The power-ground phase calculates a cycle that passes through every module once, dividing the VDD terminals from the GND terminals. This splits the chip into a VDD region and a GND region. Signal-routing techniques find a short Steiner tree in the VDD region that connects the VDD terminals to the VDD pad. This Steiner tree consists of minimum width metal wires. The same techniques route the GND tree. Using each module's current requirement, tree-traveral determines the current requirement and required width of each wire of the trees. Techniques used to widen overcrowded channels widen the wires, producing the final VDD and GND trees.

Introduction

MIT's Placement-Interconnect (PI) Project, described by Prof. Ronald Rivest at the 19th DAC, is automating the placement and interconnect phases of IC design of custom NMOS or CMOS chips designed under the rules of Mead and Conway. Input to PI consists of a description of the rectangular modules and pads to appear on the chip and a set of nets, where each net is a set of terminals on modules and pads to be connected by wire. Each chip has one VDD net, one GND net, and many signal nets. To produce the description of the routed chip, PI goes through the following stages: placing the modules and pads on the chip, routing the VDD and GND nets, routing the signal nets, and compacting the layout of the modules, pads, and wires.

At the end of the placement phase, PI knows the current requirements of each terminal and knows the position of the one VDD pad, the one GND pad, the signal pads, the modules, and the terminals.

This paper describes PI's techniques to route the VDD and GND nets, which is the goal of PI's second phase, the power-routing phase. Power-routing lays wires to connect the VDD terminals to each other and the GND terminals to each other. These wires form a tree with its base at the pad and its leaves at the modules' terminals. Each wire of the tree must be wide enough to carry the current that might flow through it. Drawing both trees in the metal layer requires that they do not cross.

We want the total length of the two trees to be as small as possible. Using less of the metal layer for power wires leaves more metal for the signal-routing phase, enhancing its performance. The signal-routing phase divides the chip area not occupied by pads, modules, or power wires into rectangular regions called free channels and uses these free channels to determine where to lay signal wires. Short simple, regular power trees produce regular free channels that are more nearly square. Dealing with such free channels enhances the signal-router's performance.

History

Zahir Syed and Abbas El Gamal's[1] algorithm grows interdigitated trees. Applying "traffic rules" to the free channels between modules prevents the trees from crossing. Another possible approach routes one tree, minimizing its length, and then routes the other tree without crossing the first. This approach then rearranges separate branches of each tree, hoping to shorten the second tree without greatly lengthening the first. Other approaches grow the tree simultaneously. Rothermel and Mlynski[2] grow one tree from the left, the other from the right. Another approach grows, at each step, the branch of a tree that would least hinder the growing of the other tree.

Using a Hamiltonian cycle to divide the chip into regions

PI's power-routing phase divides the chip into a VDD region and a GND region and then routes each net within the appropriate region. To see the relationship between the layout of power trees and a cycle that passes through every module exactly once, consider a chip with the power trees already laid. Imagine standing on a module with the module's VDD terminal on your right and its GND terminal on your left. When you try to walk to another module, keeping the VDD wires on your right and the GND wires on your left will determine the next module you encounter. Continuing the walk takes you through every module and back to where you started. A layout of power trees thus determines a Hamiltonian cycle. PI's power-routing phase first draw a Hamiltonian cycle and lets this cycle determine the layout of the power trees.

Several characteristics of the cycle are closely related to the quality of the corresponding tree layout. To ensure that the chip is divided into two regions, the cycle must not cross itself. The two regions produced by a shorter cycle have simpler shapes, and routing in such regions results in better trees. These considerations lead us to find as short a Hamiltonian cycle as can be found in a reasonable amount of time.

Finding a short Hamiltonian cycle

When using an algorithm to find a short Hamiltonian cycle, we must define the distance from one module to another. In keeping with our notion of traveling with the VDD wires on

[*]This research we supported by DARPA grant N0014-80-C0622 and Air Force grant AFOSR-F49620-81-0054.

0738-100X/83/0000/0754$1.00 © 1983 IEEE

our right, we imagine that the Hamiltonian cycle will leave a module at a point, called an OUT point, on the perimeter of the module halfway between the terminals. The OUT point is counterclockwise from the VDD terminal and clockwise from the GND terminal. Every module also has an IN point (clockwise from the VDD terminal, counterclockwise from the GND terminal). The distance from Module A to Module B is the Manhattan distance (change in x-coordinate plus change in y-coordinate) from A's OUT point to B's IN point. Note that this definition of distance is not symmetric.

There are many algorithms to find a short Hamiltonian cycle.[3] PI currently uses Shen Lin's[4]:

Start with a random, directed Hamiltonian cycle.

Delete three edges.

Reconnect the segments to form a new Hamiltonian cycle.

If the new cycle is shorter than the original, look for three edges to delete from the new. If not, look for another set of three edges to delete from the original. If trying all possible sets of three edges fails to produce a shorter cycle, accept the original cycle as a reasonably short Hamiltonian cycle.

Routing the Hamiltonian cycle

The preceding algorithm gives the order in which the cycle traverses the modules but does not completely determine the cycle. We regard each edge of the cycle as a net of two "terminals" (one module's OUT point and the other's IN point) and route it using signal-routing techniques. These techniques minimize the wire lengths and the number of jogs, which is what we want. In routing the cycle's edges, no wire is allowed to cross a wire of a previously laid edge. This ensures that the final cycle will not cross itself.

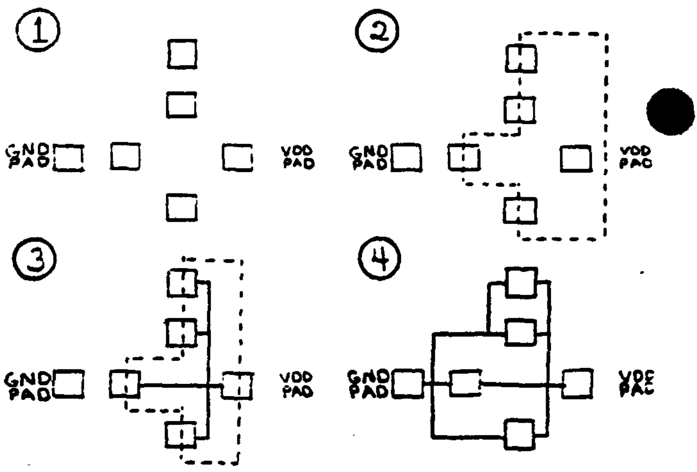
We must decide the order in which we route the segments of the Hamiltonian cycle. Laying the wires of one edge could block a path that would provide a short routing of a later edge. In routing this later edge, we may need to lay long wires to avoid crossing the previously laid wires. This effect is more noticeable with shorter edges. For a longer edge, when one module's OUT point is far from the other module's IN point, there is a greater choice of paths connecting these points using wires of approximately the same length, so cutting off one of these paths is less likely to significantly increase the final length of the tree. We therefore route the edges of the Hamiltonian cycle is ascending order according to the length of the edges.

Routing the VDD and GND nets

When the Hamiltonian cycle is completely routed, all terminals of one power net will be inside the cycle, all terminal of the other power net will be on the outside.

We first route the net inside the Hamiltonian cycle. To do this, we use signal-routing techniques that find a short Steiner tree that spans the terminals. We restrict this routing so that it doesn't cross the Hamiltonian cycle.

We then delete the Hamiltonian cycle and use signal-routing techniques to route the other net. We restrict this routing so that it doesn't cross the wires of the previously laid tree. After this routing has been accomplished, we have two noncrossing, interdigitated trees of minimum length wires. The following example shows the successive stages of input, drawing the Hamitonian cycle, routing the VDD tree, and erasing the Hamiltonian cycle and routing the GND tree.



Determining wire width

Tree traversal determines how much current might flow through each wire of the tree. We regard the layout of wires of each net as a tree with its base at the power pad and its leaves at the modules' terminals. User input gives each terminal's maximum current requirement. The maximum current of a wire ending in a terminal is the maximum current of the terminal. The maximum current of other wires is the sum of its children's maximum currents. After every wire's maximum current is known, multiplying by a design-rule constant gives every wire's minimum width.

Widening the wires

PI's stretcher widens each wire at least to its minimum width. The stretcher has algorithms to widen free channels when too many wires are squeezed into too little space. It uses these same algorithms to widen the power wires.

We regard each power wire as a long, thin rectangle and then regard each rectangle as a channel. We widen the channel, which means stretching the channel in the direction perpendicular to the current flow. Tree traversal shows the channel sides through which current enters and leaves the channel. If these sides are opposite each other, the direction of current flow is obvious and we stretch only in the direction perpendicular to this. If not, we stretch in both directions.

We stretch horizontally (in the x-direction) and then vertically (in the y-direction). Each direction requires a separate sequence of deciding whether or not this channel should be stretched in this direction, calculating the channel's minimum width, and then applying the stretching algorithms.

The result of stretching in both directions is the final power trees.

References

[1] Zahir A. Syed and Abbas El Gamal. "Single Layer Routing of Power and Ground Networks in Integrated Circuits," *Journal of Digital Systems*, Volume VI, Number 1, Spring, 1982, pages 53-63.

[2] H-J. Rothermel and D. A. Mlynski. "Computation of Power Supply Nets in VLSI Layout," *ACM IEEE Eighteenth Design Automation Conference Proceedings*, 1981, pages 37-47.

[3] See bibliography in Nicos Christofides. "The Traveling Salesman Problem," Chapter 6, *Combinatorial Optimization*, Wiley & Sons, 1979, pages 148-149.

[4] Shen Lin. "Some Computer Solutions of the Traveling-Salesman Problem," *BSTJ 44*, 2245-2269 (1965).

Signal Delay in *RC* Tree Networks

JORGE RUBINSTEIN, MEMBER, IEEE, PAUL PENFIELD, JR., FELLOW, IEEE,
AND MARK A. HOROWITZ, STUDENT MEMBER, IEEE

Abstract—In MOS integrated circuits, signals may propagate between stages with fanout. The exact calculation of signal delay through such networks is difficult. However, upper and lower bounds for delay that are computationally simpl≥ are presented in this paper. The results can be used 1) to bound the delay, given the signal threshold, or 2) to bound the signal voltage, given a delay time, or 3) certify that a circuit is "fast enough," given both the maximum delay and the voltage threshold.

I. Introduction

IN MOS INTEGRATED CIRCUITS, a given i.verter or logic node may drive several gates, some of them through long wires whose distributed resistance and capacitance may not be negligible. There does not seem to be reported in the literature any simple method for estimating signal propagation delay in such circuits, nor is there any general theory of the properties of *RC* trees, as distinct from *RC* lines. This paper presents a computationally simple technique for finding upper and lower bounds for the delay. The technique is of importance for VLSI designs in which the delay introduced by the interconnections may be comparable to or longer than active-device delay. This can be the case for wiring lengths as short as 1 mm, with 4-μm minimum feature size. The importance of this technique grows as the wiring lengths increase or the feature size decreases.

Consider the circuit of Fig. 1. The slowest transition (and therefore presumably the one of most interest) occurs when the driving inverter shuts off and its output voltage rises from a small value to V_{DD}. During this process, the various parasitic capacitances on the output are charged through the pullup transistor. Fig. 2 shows a simple model of this circuit for timing analysis. The pullup, which is nonlinear, is approximated by a linear resistor, and the transition is represented by a voltage source going from 0 (or a low value) to V_{DD} at time $t = 0$. (Later, for simplicity, a unit step will be considered instead.) The polysilicon lines are represented by uniform *RC* lines. The resistance of the metal line is neglected, but its parasitic capacitance remains. Capacitances associated with the pullup source diffusion, contact cuts, and the gates being driven are included. Any nonlinear capacitances are approximated by linear ones.

If all the resistances except the pullup can be neglected, then all the capacitors can be lumped together, and the circuit re-

Manuscript received April 20, 1982; revised June 14, 1983. This work was supported in part by Digital Equipment Corporation, in part by the Advanced Research Projects Agency of the Department of Defense and monitored by the Office of Naval Research under Contract N00014-C-80-0622, in part by the Air Force under Contract AFOSR 4-9620-80-0073, and in part by the Army Research Office under Contract DAAG-29-80-K-0046.

J. Rubinstein is with the Digital Equipment Corporation, 75 Reed Road, Hudson, MA 01749.

P. Penfield, Jr. is with the Department of Electrical Engineering and Computer Science and Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA 02139.

M. A. Horowitz is with the Department of Electrical Engineering, Stanford University, Palo Alto, CA 94305.

0028-0070/83/0700-0202$01.00 © 1983 IEEE

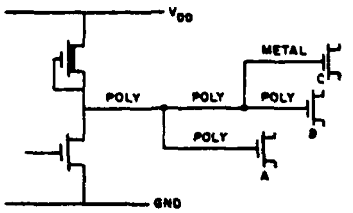


Fig. 1. Typical MOS signal-distribution network. The inverter is shown driving three gates, through a fanout network implemented in polysilicon and metal.

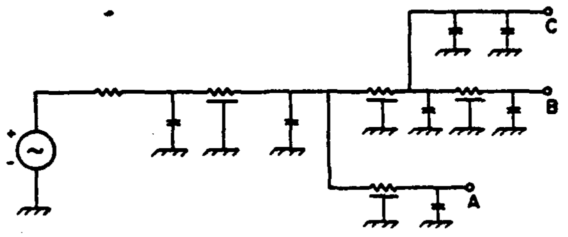


Fig. 2. Linear-circuit model for the network of Fig. 1. The voltage source is a step at time $t = 0$.

sponse may be found in closed form. The voltages at all the outputs are the same:

$$v_{\text{out}}(t) = V_{DD}(1 - e^{-t/RC_T}) \tag{1}$$

where R is the pullup resistance and C_T the total capacitance. Thus at a given time T, the output voltage $v_{\text{out}}(T)$ is given by (1), and the time at which $v_{\text{out}}(t)$ reaches some specified critical voltage V_{CR} is given by

$$T = RC_T \ln \frac{V_{DD}}{V_{DD} - V_{CR}}. \tag{2}$$

However, if the resistances of the lines are comparable to that of the pullup, this solution is not correct. The circuit response cannot generally be calculated in closed form. The results below can be used to calculate upper and lower bounds to the delay that are very tight in the case where most of the resistance is in the pullup. The theory as presented here does not explicitly deal with nonlinearities and therefore does not apply to signal propagation through pass transistors.

Previous work on distributed RC circuits is summarized in the extensive bibliographies of Ghausi and Kelly [1] and Kumar [2]. There does not appear to be any treatment of RC trees, as distinct from RC lines, in these bibliographies. Perhaps the most complete treatment of the properties of RC lines is that of Protonotarios and Wing [3], [4]; some (but not all) of the theorems proved there also apply to RC trees. Most of the work cited deals with techniques to approximate the response of such networks, rather than to find bounds; an exception is that of Singhal and Vlach [5], [6]. An important early analytical approximation to delay is that by Elmore [7], who called the first moment of the impulse response the delay. This definition is inadequate because it does not define delay in terms of signal threshold.

Preliminary, restricted versions of some of the results given below have been presented before by the two senior authors [8], [9], and utilized in at least two working timing analyzers [10]-[12]. The junior author simplified the derivation and tightened some of the bounds.

II. Statement of the Problem

An RC tree, as considered in this paper, is a generalization of the well-known RC lines [3], [4]. It may be defined recursively as follows. There are three primitive elements. First, a lumped capacitor between ground and another node is an RC tree. Second, a lumped resistor between two nonground nodes is an RC tree. Third, a (distributed) RC line, uniform or nonuniform, in the configuration with no dc path to ground, is an RC tree. Finally, any two RC trees with common ground, and one nonground node from each connected together, form a new RC tree. This definition does not permit resistor loops, so that the resistors (including those in the distributed RC lines) form a topological tree that does not include the ground node. All of the capacitors (including the distributed capacitances in the RC lines) are connected to ground. One of the nonground nodes of the final tree is assumed to be the input, and one or more nodes the outputs.

In many cases, each branch of the tree except the input terminates in an output; however, this is not required, and in this paper the outputs may be defined at any of the nonground nodes.

As a consequence of this definition, there is a unique path through the resistive part of the network from any nonground node to the input.

For simplicity, most of the theory below will be presented for the special case with only lumped resistors and capacitors. However, the generalization to include distributed RC lines (uniform or nonuniform) is straightforward. All the results apply in the form given, except that the summations in the formulas for T_P, T_{Di}, and T_{Ri} are replaced by a combination of summations and integrals. The easiest way to picture the result is to think of each RC line as represented by a finite number of lumped RC sections, so that the derivations apply, and then consider the limit as the number of sections used to represent each line goes to infinity. All the summations are well behaved in the limit. The required integrals are given explicitly in Appendix A for both uniform and nonuniform distributed lines.

The RC tree representing the signal path is, without loss of generality, assumed to be driven at the input with a unit step voltage (henceforth all voltages may be thought of as normalized to the magnitude of the step excitation). Gradually the voltages at all other nodes, and in particular at all the outputs, rise from 0 to 1 V. It is assumed that the output voltages cannot be calculated easily. The problem is to find simple upper and lower bounds for the output voltages, or, equivalently, to find upper and lower bounds for the delay associated with each output.

III. Analytical Theory

Consider any output node i (in this paper, i will be used as an index selecting an output node) and any lumped capacitor at node k with capacitance C_k. The resistance R_{ki} is defined as the resistance of th: portion of the (unique) path between

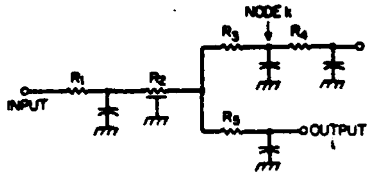


Fig. 3. Illustration of resistance terms. For this network, $R_{ki} = R_1 + R_2$, $R_{kk} = R_1 + R_2 + R_3$, and $R_{ii} = R_1 + R_2 + R_5$.

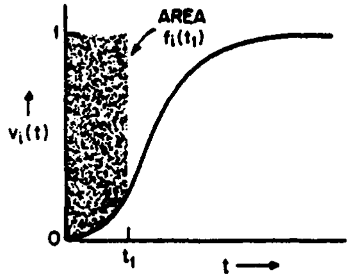


Fig. 4. Interpretation of $f_i(t)$ as the integral above the response $v_i(t)$. Note that $f_i(\infty) = T_{Di}$.

the input and i, that is common with the (unique) path between the input and node k. In particular, R_{ii} is the resistance between input and output i, and R_{kk} is the resistance between the input and node k. Thus $R_{ki} < R_{kk}$ and $R_{ki} < R_{ii}$. For an example, see Fig. 3.

The sum (over all the capacitors in the network)

$$T_P = \sum_k R_{kk} C_k \tag{3}$$

has the dimensions of time. Next, define for each output i two quantities that also have the dimensions of time

$$T_{Di} = \sum_k R_{ki} C_k \tag{4}$$

$$T_{Ri} = \left(\sum_k R_{ki}^2 C_k\right) \Big/ R_{ii}. \tag{5}$$

These quantities play a role in the final delay formulas but none of them is equal to the delay, although T_{Di} is equal to the first-order moment of the impulse response (see Appendix B), which has been called "delay" by Elmore [7]. Note that the network has one value of T_P, but each output of the network has a separate T_{Di} and T_{Ri}. It is easily shown from the definitions that

$$T_{Ri} < T_{Di} < T_P. \tag{6}$$

For RC trees without side branches, $T_{Di} = T_P$. An interpretation of T_P and T_{Di} in terms of the system function of the network appears in Appendix B.

The voltage at each output i (and in fact at each node) is a monotonic function of time during the transient, as proved in Appendix C. Also, the analog of the well-known fact that voltage along an RC line is a concave function of distance (suitably defined) is the following general result (proved in Appendix D):

$$R_{ii}[1 - v_k(t)] > R_{ki}[1 - v_i(t)]. \tag{7}$$

A similar result is found by interchanging i and k subscripts

$$R_{ki}[1 - v_k(t)] < R_{kk}[1 - v_i(t)]. \tag{8}$$

These results apply to any output i and any node k, whether the output is "upstream" or "downstream" from the node k.

At any instant of time, the voltage difference between the input and any output i may be calculated by summing the voltage drops along the (unique) path between input and output. Each such drop may be expressed as the resistance times the current feeding all "downstream" capacitors. Alternatively, this double sum may be expressed as a sum over all capacitors in the network, of the current through each capacitor times that portion of the "upstream" resistance that also happens to lie along the path to the output i. This resistance is what has been defined as R_{ki}, so

$$1 - v_i(t) = \sum_k R_{ki} C_k \frac{dv_k}{dt}. \tag{9}$$

Equation (9) is integrated between 0 and t, and the result denoted $f_i(t)$:

$$\begin{aligned} f_i(t) &= \int_0^t [1 - v_i(t')]\, dt' \\ &= \sum_k R_{ki} C_k v_k(t) \\ &= T_{Di} - \sum_k R_{ki} C_k [1 - v_k(t)]. \end{aligned} \tag{10}$$

This integral plays a central role in the derivation of the bounds. A graphical interpretation appears in Fig. 4, which shows a typical step response. The area above the response but below the unit input is $f_i(t)$. As t approaches infinity, this approaches T_{Di}.

If (7) and (8) are used in (10), the result is

$$T_{Ri}[1 - v_i(t)] < T_{Di} - f_i(t) < T_P[1 - v_i(t)] \tag{11}$$

which is equivalent to

$$\frac{T_{Di} - f_i(t)}{T_P} < \frac{df_i(t)}{dt} < \frac{T_{Di} - f_i(t)}{T_{Ri}} \tag{12}$$

which, when integrated between times t_1 and $t_2 > t_1$, yields

$$\begin{aligned} [T_{Di} - f_i(t_1)]\, e^{-(t_2 - t_1)/T_{Ri}} &< [T_{Di} - f_i(t_2)] \\ &< [T_{Di} - f_i(t_1)]\, e^{-(t_2 - t_1)/T_P}. \end{aligned} \tag{13}$$

Since $v_i(t)$ is monotonic nondecreasing

$$(t_4 - t_3)\,[1 - v_i(t_4)] < f_i(t_4) - f_i(t_3) \tag{14}$$

for any nonnegative t_3 and t_4.

The voltage bounds are now easily derived. Of course

$$v_i(t) > 0 \tag{15}$$

but, in addition, from (11) and (14) with $t_3 = 0$ and $t_4 = t$

$$v_i(t) > 1 - \frac{T_{Di}}{t + T_{Ri}} \tag{16}$$

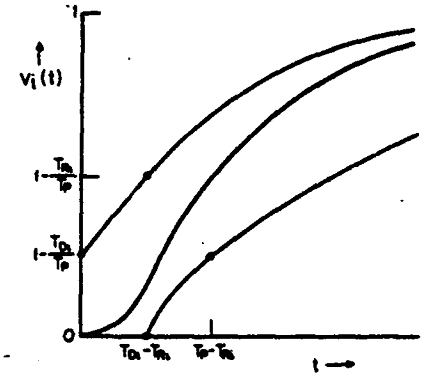


Fig. 5. Form of the bounds, with the distances from the exact solution exaggerated for clarity.

and, from the first inequality in (11), (14) with $t_3 = t - T_P + T_{Ri}$ and $t_4 = t$, and the second inequality in (13) with $t_1 = 0$ and $t_2 = t_3$

$$v_i(t) \geqslant 1 - \frac{T_{Di}}{T_P} e^{(T_P - T_{Ri})/T_P} e^{-t/T_P} \tag{17}$$

which holds only for $t \geqslant T_P - T_{Ri}$. The best lower bound is (15) for $t \leqslant T_{Di} - T_{Ri}$, (16) for $T_{Di} - T_{Ri} \leqslant t \leqslant T_P - T_{Ri}$, and (17) for $T_P - T_{Ri} \leqslant t$. The upper bounds on voltage are, from (11) and (14) with $t_3 = t$ and $t_4 = 0$

$$v_i(t) \leqslant 1 - \frac{T_{Di} - t}{T_P} \tag{18}$$

and, from the second inequality in (11), the first inequality in (13) with $t_1 = T_{Di} - T_{Ri}$ and $t_2 = t$, and (14) and $t_3 = T_{Di} - T_{Ri}$ and $t_4 = 0$

$$v_i(t) \leqslant 1 - \frac{T_{Ri}}{T_P} e^{(T_{Di} - T_{Ri})/T_{Ri}} e^{-t/T_{Ri}} \tag{19}$$

which holds only for $t \geqslant T_{Di} - T_{Ri}$. The best upper bound for voltage is (18) for $t \leqslant T_{Di} - T_{Ri}$ and (19) for $T_{Di} - T_{Ri} \leqslant t$.

Bounds for the time, given the voltage, are possible because the voltage is a monotonic function of time. Of course

$$t \geqslant 0 \tag{20}$$

and in addition, (18) and (19) can be inverted to yield

$$t \geqslant T_{Di} - T_P[1 -. v_i(t)] \tag{21}$$

$$t \geqslant T_{Di} - T_{Ri} + T_{Ri} \ln \frac{T_{Ri}}{T_P[1 - v_i(t)]} \tag{22}$$

and (16) and (17) yield

$$t \leqslant \frac{T_{Di}}{1 - v_i(t)} - T_{Ri} \tag{23}$$

$$t \leqslant T_P - T_{Ri} + T_P \ln \frac{T_{Di}}{T_P[1 - v_i(t)]} \tag{24}$$

where (22) applies only if $v_i(t) \geqslant 1 - T_{Ri}/T_P$. and (24) only if $v_i(t) \geqslant 1 - T_{Di}/T_P$. The general form of all these bounds is illustrated in Fig. 5.

These bounds, (15)-(19) for voltage, and (20)-(24) for time, constitute the major result of this paper.

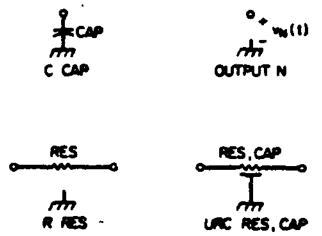


Fig. 6. Elements sufficient for describing *RC* trees. For simplicity, only uniform distributed *RC* lines are included. The parameters are *CAP*, *RES*, and *N*, and the functions which return networks are *C*, *OUTPUT*, *R*, and *URC*. The capacitor and output designation are one-port elements, and the resistor and uniform line are two-port elements.

IV. Practical Hierarchical Algorithms

Use of hierarchy is a powerful way to deal with complexity in design of large systems. Computation cost is usually less with hierarchical algorithms, and analysis of part of a design can be done before the rest is known. In this section, programs are given for calculating the voltage and time bounds of this paper hierarchically. Although intended for exposition, these programs are complete and do work. They may be used interactively, without any changes whatever, for small or moderate size networks, or, for large networks, they may be incorporated into systems that deal with machine-readable network descriptions.

One way to use the inequalities of this paper is to consider the overall *RC* tree, and compute for each capacitor the appropriate R_{ki} and R_{kk} so that T_P, T_{Di}, and T_{Ri} for each output can be found. Of course, for networks with distributed lines, the sums are augmented with integrals as discussed in Appendix A. In this approach, the calculations for each output require time proportional to the square of the number of elements.

An alternate scheme is to build up the network by construction, and calculate independently for each of the partially constructed networks enough information to permit the final calculation of T_P, T_{Di}, and T_{Ri}. A recursive definition of *RC* trees is given below, and if the network is expressed in these terms rather than in the form of a schematic diagram, the resulting expression can be used as a guide for the calculations. The computation time for each output is proportional to the number of elements, rather than the square of the number. Programs that implement this approach appear below.

Fig. 6 shows the four building blocks: lumped capacitor, lumped resistor, uniform *RC* line, and declaration of output. The capacitor and the output label are considered as two-terminal, or one-port networks. The *RC* line and the resistor are considered as two-port networks. If desired, particular nonuniform *RC* lines, such as exponentially or linearly tapered lines, can be included also. Fig. 7 shows the five permissible ways of wiring these building blocks, or previously wired subnetworks, together. Any *RC* tree can be denoted by an expression using only these wiring functions. The syntax shown is identical to APL syntax, and the programs below are written in APL. Note that Figs. 6 and 7 do not give a minimal set of elements or wiring functions, since some can be expressed in

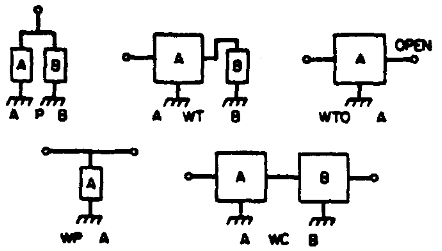


Fig. 7. Wiring functions for interconnecting elements or subtrees. The functions which return one-port networks are *P*, *WT*, and *WTO*, and those that return two-port networks are *WP* and *WC*. Here *A* and *B* are previously defined *RC* trees.

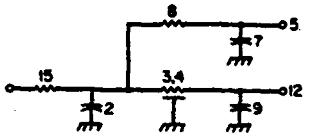


Fig. 8. Example network. Parameter values are in ohms and farads. The characteristic times (in sec.) are $T_P = 419$, $T_{D5} = 386$, $T_{R5} = 307.7$, $T_{D12} = 363$, and $T_{R12} = 335.2$.

terms of others. The names for the wiring functions are taken from the notation of the program MARTHA [13]-[16].

Example: The network shown in Fig. 8 may be denoted

$$(R\ 15)\ WT(C\ 2)\ P((R\ 8)\ WT(C\ 7)\ P\ OUTPUT\ 5)$$
$$P((URC\ 3\ 4)\ WC\ WP\ C\ 9)\ WT\ OUTPUT\ 12 \tag{25}$$

and is a one-port network, with two declared outputs.

For convenience, this notation allows a network with only one output to be expressed as a two-port with the second port an implicit output, without any explicit output declaration. The explicit declaration of outputs is handy because often side branches do not represent outputs of interest.

If an expression such as (25) is to be used as a guide for the calculations, then each function shown must correspond to the calculation of partial results which are sufficient to allow further calculations. The following information is adequate at each stage in the construction of the network:

(1) Total capacitance C_T.

(2) T_P of the network as constructed so far.

(3) For a two-port, considering port 2 as an implicit output, R_{22}, T_{D2}, and T_{R2}. (For convenience, the product $R_{22}T_{R2}$ is used in the programs below instead of T_{R2}.)

(4) For each declared output in a one-port or two-port network, R_{ii}, T_{Di}, and T_{Ri}. (For convenience, $R_{ii}T_{Ri}$ is used rather than T_{Ri}.)

(5) For each declared output in a two-port network, R_{2i}.

Each of the quantities identified above pertains to the particular subnetwork and can be calculated from a knowledge of that subnetwork alone, independent of how the subnetwork may later be wired together with other subnetworks. As an example of the use of these quantities during construction of the network, consider the cascade operation *WC*. The objective is to find C_T, T_P, R_{22}, T_{D2}, T_{R2}, and all R_{ii}, T_{Di}, T_{Ri}, and R_{2i} of the cascade *A WC B* from the corresponding quantities for its two arguments *A* and *B*. The formulas for calculating these are

```
    ∇  Z←C CAP
[1]    →'CAPACITANCE' ERRORIF 1≠ρCAP←,CAP
[2]    Z←1,CAP,0
    ∇

    ∇  Z←OUTPUT N
[1]    →'OUTPUT LABEL' ERRORIF 1≠ρN←,N
[2]    Z← 1 0 0 ,N, 0 0 0
    ∇

    ∇  Z←R RES
[1]    →'RESISTANCE' ERRORIF 1≠ρRES←,RES
[2]    Z← 2 0 0 ,RES, 0 0
    ∇

    ∇  Z←URC RC;RES;CAP
[1]    →'RES, CAP' ERRORIF 2≠ρRC←,RC
[2]    RES←1↑RC
[3]    CAP←1↓RC
[4]    Z←2,CAP,(CAP×RES÷2),RES,(CAP×RES÷2),CAP×RES×RES÷3
    ∇
```

Fig. 9. APL functions for the elements.

$$C_T = C_{TA} + C_{TB} \tag{26}$$

$$T_P = T_{PA} + T_{PB} + R_{22A}C_{TB} \tag{27}$$

$$R_{22} = R_{22A} + R_{22B} \tag{28}$$

$$T_{D2} = T_{D2A} + T_{D2B} + R_{22A}C_{TB} \tag{29}$$

$$\begin{aligned} T_{R2}R_{22} = {} & T_{R2A}R_{22A} + T_{R2B}R_{22B} \\ & + 2R_{22A}T_{D2B} + R_{22A}^2 C_{TB} \end{aligned} \tag{30}$$

$$R_{ii} = R_{iiA}, R_{iiB} + R_{22A} \tag{31}$$

$$\begin{aligned} T_{Di} = {} & (T_{DiA} + R_{2iA}C_{TB}), T_{DiB} \\ & + R_{22A}C_{TB} + T_{D2A} \end{aligned} \tag{32}$$

$$\begin{aligned} T_{Ri}R_{ii} = {} & (T_{RiA}R_{iiA} + R_{2iA}^2 C_{TB}), T_{RiB}R_{iiB} \\ & + T_{R2A}R_{22A} + 2R_{22A}T_{DiB} + R_{22A}^2 C_{TB} \end{aligned} \tag{33}$$

$$R_{2i} = R_{2iA}, R_{2iB} + R_{22A}. \tag{34}$$

The corresponding formulas for the other wiring functions are similar, but not as complicated.

A set of APL functions which implement this scheme appear in Figs. 9-12. The necessary data is passed around in the form of vectors. A one-port network is represented by the vector 1, C_T, T_P, followed by zero or more sets of four numbers, N_i, R_{ii}, T_{Di}, $T_{Ri}R_{ii}$. The number 1 starting off the vector is the number of ports, and N_i is the (numerical) label for each output. It is not necessary to pass along the number of declared outputs since that can be calculated from the length of the vector. In a similar way, a two-port network is represented by the vector 2, C_T, T_P, R_{22}, T_{D2}, $T_{R2}R_{22}$, followed by zero or more sets of five numbers, N_i, R_{ii}, T_{Di}, $T_{Ri}R_{ii}$, and R_{2i}, one set for each declared output.

The background functions in Fig. 12 provide 1) some error control (with automatic abort in case of error), 2) calculation of the number of ports of a network, 0 being returned for ill-formed arguments, 3) a matrix with the data for the declared outputs, and 4) extraction of T_P, T_{Di}, and T_{Ri}. The four elements appear in Fig. 9, and the wiring functions in Fig. 10. The listing of *WC*, for example, shows after error checking, the calculation of the required output, term by term, from the arguments. This function can be compared with (26)-(34).

Fig. 11 shows five functions intended to calculate the bounds for any network. The convention followed is that if the argument for any of these is a two-port network, the second port

```
      ∇ Z←A P B
[1]     →'1-PORT' ERRORIF(1≠NPORTS A)∨1≠NPORTS B
[2]     Z←1,(A[2 3]+B[2 3]),(3↓A),3↓B
      ∇

      ∇ Z←A WT B;E
[1]     →'2-PORT' ERRORIF 2≠NPORTS A
[2]     →'1-PORT' ERRORIF 1≠NPORTS B
[3]     Z←1,(A[2]+B[2]),A[3]+B[3]+B[2]×A[4]
[4]     E←OUTPUTS A
[5]     E[;3]←E[;3]+B[2]×E[;5]
[6]     E[;4]←E[;4]+B[2]×E[;5]÷2
[7]     Z←Z,, 0 ¯1 ↓E
[8]     E←OUTPUTS B
[9]     E[;2]←E[;2]+A[4]
[10]    E[;4]←E[;4]+A[6]+(2×A[4]×E[;3])+B[2]×A[4]÷2
[11]    E[;3]←E[;3]+A[5]+B[2]×A[4]
[12]    Z←Z,,E
      ∇

      ∇ Z←WTO A
[1]     →'2-PORT' ERRORIF 2≠NPORTS A
[2]     Z←1,A[2 3],, 0 ¯1 ↓OUTPUTS A
      ∇

      ∇ Z←A WC B;E
[1]     →'2-PORT' ERRORIF(2≠NPORTS A)∨2≠NPORTS B
[2]     Z←2,(A[2]+B[2]),(A[3]+B[3]+B[2]×A[4]),(A[4]+B[4]),A[5]+B[5]+B[2]×A[4]
[3]     Z←Z,A[6]+B[6]+(2×B[5]×A[4])+B[2]×A[4]÷2
[4]     E←OUTPUTS A
[5]     E[;3]←E[;3]+B[2]×E[;5]
[6]     E[;4]←E[;4]+B[2]×E[;5]÷2
[7]     Z←Z,,E
[8]     E←OUTPUTS B
[9]     E[; 2 5]←E[; 2 5]+A[4]
[10]    E[;3]←E[;3]+A[5]+B[2]×A[4]
[11]    E[;4]←E[;4]+A[6]+(2×B[5]×A[4])+B[2]×A[4]÷2
[12]    Z←Z,,E
      ∇

      ∇ Z←WP A
[1]     →'1-PORT' ERRORIF 1≠NPORTS A
[2]     Z←2,A[2 3], 0 0 0 ,,(OUTPUTS A),0
      ∇
```

Fig. 10. APL functions for the wiring functions.

```
      ∇ Z←VMIN A;II;TD;TP;TR
[1]     →T SETUP A
[2]     Z←(II≥TP-TR)×1-(TD÷TP)×*1-(II+TR)÷1E¯30⌈TP
[3]     Z←Z⌈1-TD÷1E¯30⌈II+TR
      ∇

      ∇ Z←VMAX A;II;TD;TP;TR
[1]     →T SETUP A
[2]     Z←TR××0⌊¯1+(TD-II)÷1E¯30⌈TR
[3]     Z←1-(Z⌈TD-II)÷1E¯30⌈TP
      ∇

      ∇ Z←TMIN A;II;TD;TP;TR
[1]     →V SETUP A
[2]     Z←TD-TP×1-II
[3]     Z←Z⌈(TR≥TP×1-II)×TD-TR×1-⍟TR÷1E¯20⌈TP×1-II
      ∇

      ∇ Z←TMAX A;II;TD;TP;TR
[1]     →V SETUP A
[2]     Z←(TD÷1E¯20⌈1-II)-TR
[3]     Z←Z⌊(TP-TR)+0⌈TP×⍟TD÷1E¯20⌈TP×1-II
      ∇

      ∇ Z←OK A
[1]     →'CIRCUIT' EPRORIF~(NPORTS A)∊ 1 2
[2]     Z←(T∘.≥TMAX A)-T∘.<TMIN A
      ∇
```

Fig. 11. Response functions. The very small numbers in the functions guard against errors for pathological networks and certain limiting values for voltage and time.

is taken as the desired output, and the declared outputs are ignored. If the argument is a one-port network, then the declared outputs are used. The two functions *TMIN* and *TMAX* calculate the lower and upper bounds for delay, and refer to a global variable named *V* which contains the threshold, a number (or array of numbers) between 0 and 1. The functions *VMIN* and *VMAX* calculate the lower and upper bounds for signal voltage and refer to a global variable *T* containing an ar-

```
      ∇ Z←N ERRORIF B
[1]     Z←⍳0
[2]     →(~1∊B)/0
[3]     N,' ERROR.'
[4]     Z←0
      ∇

      ∇ Z←NPORTS A
[1]     Z←0
[2]     →(1≠ρρA)/0
[3]     Z←1
[4]     →((1=1↑A)∧3=4|ρA)/0
[5]     Z←2
[6]     →((2=1↑A)∧(6≤ρA)∧1=5|ρA)/0
[7]     Z←0
      ∇

      ∇ E←OUTPUTS A
[1]     E← 0 0 ρ0
[2]     →(1 2 =NPORTS A)/L1,L2
[3]     →0
[4]    L1:E←((((ρA)-3)÷4),4)ρ3↓A
[5]     →0
[6]    L2:E←((((ρA)-6)÷5),5)ρ6↓A
      ∇

      ∇ Z←I SETUP A;E;N;S
[1]     N←NPORTS A
[2]     →Z←'CIRCUIT' ERRORIF~N∊ 1 2
[3]     →(N=2)/L2
[4]     E←OUTPUTS A
[5]     S←(ρI),1↑ρE
[6]     TP←A[3]
[7]     TD←SρE[;3]
[8]     TR←Sρ(E[;2]≠0)×E[;4]÷E[;2]
[9]     II←⍉(⌽S)ρ⍉I
[10]    →0
[11]   L2:TP←A[3]
[12]    TD←A[5]
[13]    TR←(A[4]≠0)×A[6]÷A[4]
[14]    II←I
      ∇
```

Fig. 12. APL background func..ons to support the functions in Figs. 9, 10, and 11.

```
⍝ EXAMPLE OF THE USE OF RC-TREE DELAY CALCULATIONS:

BRANCH1 ← (R 8) WT (C 7) P OUTPUT 5
BRANCH2 ← (URC 3 4) WT (C 9) P OUTPUT 12
NET ← (R 15) WT (C 2) P BRANCH1 P BRANCH2

⍝ NOW THE NETWORK IS DEFINED.

V ← 0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9

⍝ NOW THE VECTOR OF THRESHOLD VOLTAGES IS DEFINED.
⍝ NEXT TO FIND THE MINIMUM AND MAXIMUM BOUNDS FOR DELAY:

V, (TMIN NET), TMAX NET
0              0          0           78.261      27.833
0.1            8.9        0          121.03       68.167
0.2           50.8       27.8        170.39      117.22
0.3           93.05      72.555      226.34      173.17
0.4          140.49     124.22       290.92      237.76
0.5          196.6      185.33       367.32      314.15
0.6          265.27     260.12       460.81      407.65
0.7          353.8      356.54       581.35      528.18
0.8          478.57     492.44       751.24      698.07
0.9          691.88     724.76      1041.7       988.5
⍝ V          TMIN       TMIN        TMAX        TMAX
⍝            OUTPUT 5   OUTPUT 12   OUTPUT 5    OUTPUT 12

⍝ NOW TO DEFINE A DELAY VECTOR AND GET THE BOUNDS ON VOLTAGE:

T ← 0 20 40 60 80 100 200 300 400 500 1000 2000

T, (VMIN NET), VMAX NET
   0         0           0            0.078759    0.13365
  20         0           0            0.12649     0.18138
  40         0           0.03243      0.17422     0.2286
  60         0           0.0814       0.22196     0.27328
  80         0.0044853   0.12565      0.26968     0.31538
 100         0.053316    0.16644      0.31563     0.35503
 200         0.25459     0.34342      0.5055      0.52141
 300         0.41286     0.48283      0.64269     0.64487
 400         0.53752     0.59263      0.74182     0.73648
 500         0.63571     0.67913      0.81345     0.80446
1000         0.88954     0.90271      0.96326     0.95601
2000         0.98984     0.99105      0.99857     0.99777
 ⍝ T         VMIN        VMIN         VMAX        VMAX
 ⍝           OUTPUT 5    OUTPUT 12    OUTPUT 5    OUTPUT 12
```

Fig. 13. Example of the use of the fast calculation scheme to find upper and lower bounds on delay and response voltage.

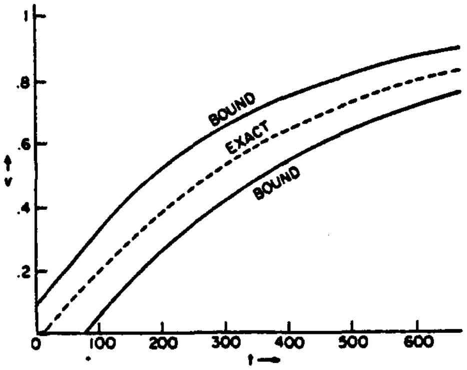


Fig. 14. Upper and lower bounds for output 5, as calculated in Fig. 13. The exact solution, found from circuit simulation, is shown also.

ray of (positive) delay times. The final function, *OK*, refers to both *V* and *T* and returns 1 if all is well, that is, if *TMAX* ≤ *T*, or -1 if the network definitely will fail, that is if *T* < *TMIN*, or 0 if the bounds are not tight enough to tell for sure, that is if *TMIN* ≤ *T* < *TMAX*. An example of the use of these functions to test the network in Fig. 8 is shown in Figs. 13 and 14.

V. Application to PLA Speed Estimates

These bounds are applied, as an example, to polysilicon lines driving the AND plane of a PLA, to determine whether or not the dominant delay occurs here. It is assumed that a strong superbuffer driver drives the line, and that every other minterm has a transistor present. The gates are assumed to be 4-microns square, separated by 24 μm of *RC* line. The poly resistance is assumed to be 30-Ω per square, the gate oxide thickness 400 Å, and the field-oxide thickness 3000 Å.

```
    ∇  Z←PLALINE N;A
        A←(URC 180 0.01)WC URC 30 0.013
       ⍝ A IS A SINGLE SECTION ACCOUNTING FOR TWO MINTERMS
        Z←(R 380)WC WP C 0.04
       ⍝ Z IS THE PULLUP R AND C FOR SUPERBUFFER DRIVER
       LOOP:→(N≤0)/0
        Z←Z WC A
        N←N-2
        →LOOP
    ∇
```

Fig. 15. APL function which returns a model of a PLA line with *N* minterms.

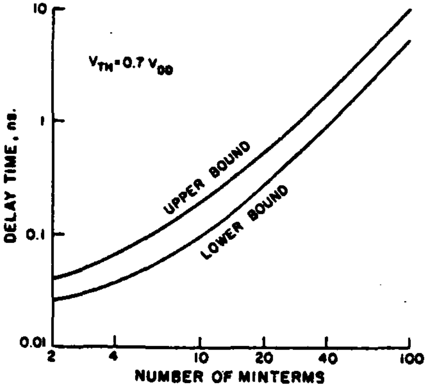


Fig. 16. Upper and lower bounds on response time of the network of Fig. 15, shown as a function of the number of minterms in the PLA.

These numbers lead to a capacitance of 0.01 pF and resistance 180 Ω between gates, and a resistance of 30 Ω and capacitance of 0.013 pF for each gate. The network is driven by a source resistance of 380 Ω and the effective capacitance of the output of the driver is estimated as 0.04 pF.

A function which returns a network with *N* minterms is shown in Fig. 15. The results of calculating the delay as a function of the number of minterms are shown in Fig. 16. The voltage threshold was taken to be 0.7 times V_{DD}. On this log-log plot the quadratic dependence of delay on number of minterms (as a measure of the length of the line) is evident. Also evident is the fact that even with as many as a hundred minterms, the delay is guaranteed to be no worse than 10 ns. This suggests that the dominant delay in a PLA occurs elsewhere.

VI. Conclusions

A computationally efficient method for calculating the signal delay through MOS interconnect lines with fanout has been described. Tight upper and lower bounds for the step response of *RC* trees have been presented, together with linear-time algorithms for these bounds from an algebraic description of the tree. Substantial computational simplicity is achieved even in the presence of *RC* distributed lines by representing the *RC* tree by a small set of suitably defined characteristic times, which can be calculated easily and used to generate the bounds.

Although only the step response is considered here, the results can be extended to upper and lower bounds for arbitrary excitation by use of the superposition integral. This extension is discussed in Appendix E. An example of this calculation appears in [9].

Extensions of the theory to *RC* trees with nonlinear elements (similar to the work of Glasser [17] for nonlinear MOS inverters) would be desirable for better modeling of MOS circuits. Investigations of *RC* trees with nonlinear capacitors and resistors are now under way, along with attempts to unify the modeling of gates and interconnects, and in particular to include pass transistors in the interconnects. Tighter bounds are also being looked for.

Appendix A

The results of this paper are valid for *RC* trees that contain distributed *RC* lines. All results apply without change, except that the definitions of T_P, T_{Di}, and T_{Ri} in (3)-(5) are replaced by (36)-(38) below.

The summations in (3)-(5) are for the case of lumped capacitors only, and the index k runs over all lumped capacitors in the network. The form for networks with distributed *RC* lines is similar; the index k runs over both lumped capacitors and *RC* lines. The terms for lumped capacitors are unchanged from (3)-(5), but for distributed lines additional terms appear in (36)-(38).

Each line k has a total capacitance C_k and appears in the network with one end (say the left-hand end) nearer the input of the network. Along the line, the capacitance is distributed, but the cumulative capacitance c is a function of position, and has a value between 0 (at the left end) and C_k (at the right end). For each value of c, there is a value of cumulative resistance $r(c)$ monotonically increasing with c; $r(0) = 0$ and $r(C_k)$ is the total resistance of the line. For uniform lines, $r(c)$ is a linear function, and for nonuniform lines $r(c)$ has other shapes. Define the series of integrals

$$I_k^{(n)} = \int_0^{C_k} [r(c)]^n \, dc. \tag{35}$$

Note that if the line k is interpreted as a simple *RC* tree without any additional elements, then its T_P and T_D are $I_k^{(1)}$ and its T_R is $I_k^{(2)}/r(C_k)$. For a uniform line, $I_k^{(1)} = r(C_k)\, C_k/2$ and $I_k^{(2)} = [r(C_k)]^2 C_k/3$.

For each distributed line k let R_{kk} be the resistance between the left-hand end of the line and the input of the network, and R_{ki} be the portion of that resistance that also lies on the (unique) path between the input and any output i. Then the expressions for T_P, T_{Di}, and T_{Ri} are

$$T_P = \sum_k R_{kk} C_k + \sum_k I_k^{(1)} \tag{36}$$

$$T_{Di} = \sum_k R_{ki} C_k + \sum_k I_k^{(1)} \tag{37}$$

$$T_{Ri} = \left(\sum_k R_{ki}^2 C_k + 2 \sum_k R_{ki} I_k^{(1)} + \sum_k I_k^{(2)} \right) \Big/ R_{ii} \tag{38}$$

where the first sum in all three expressions is over both lumped capacitors and distributed lines; the second sum in (36) is over all distributed lines; and the second sum in (37) and the second and third sums in (38) are over only these distributed lines which lie along the (unique) path between the input and output i.

Appendix B

From (3) it is evident that T_P is equal to the sum of all the open-circuit time constants of the network, a quantity that is well known in the analysis of multistage amplifiers, and that has been shown to be equal to the negative of the sum of the inverse of all the transmission poles [18]. That is, if the normalized system function $H_i(s)$ for the output i is

$$H_i(s) = \frac{1 + b_1 s + b_2 s^2 + \cdots}{1 + a_1 s + a_2 s^2 + \cdots} \tag{39}$$

then $T_P = a_1$. Also, it can be shown that $T_{Di} = a_1 - b_1$. To prove this, one starts from the equality [19] between $a_1 - b_1$ and the first-order time moment $\int_0^\infty h_i(t)\, t\, dt$ of the impulse response $h_i(t)$. Integrating by parts, one can show that

$$\int_0^\infty h_i(t)\, t\, dt = \int_0^\infty [1 - v_i(t)]\, dt,$$

and therefore is equal to $f_i(\infty) = T_{Di}$, as given by (10). This completes the proof, and shows that T_{Di} is equal to the first-order time-moment of the impulse response.

Appendix C

It is proved here that when a linear *RC* tree is excited with a step input from an initial rest condition, the voltage at each node is a monotonic function of time. This condition is identical to the condition that the impulse response of the same network is nonnegative. The proof is to assume that, with an impulse applied at the input, the voltage on one or more nodes is, at some instant of time, negative, and then show that this assumption leads to a contradiction.

It is assumed that distributed *RC* lines in the tree can be replaced by finite lumped ladder approximations with arbitrarily close impulse responses, so that if one or more nodes of the original network has a negative voltage, then so does one or more nodes of an approximate lumped network. For the remainder of this appendix, this lumped network is dealt with.

At time $t = 0+$, the voltages on all nodes are nonnegative. Let $v_{\min}(t)$ denote the lowest voltage of any node in the network at time t. Assume that at some time $t_0 > 0$, $v_{\min}(t_0) < 0$. Then there must be some prior time t_1 when $v_{\min}$ and its derivative with respect to time are both negative.

At time t_1, $v_{\min}(t_1)$ is achieved by at least one node. If it is achieved by more than one, at least one must have a negative derivative. This node is characterized by having the lowest voltage in the network, and also a negative derivative. The net current flowing into this node from other nodes in the *RC* tree is nonnegative, because the adjacent nodes, to which this node is directly connected through resistors, are not at a lower potential. This net current must flow into the capacitor at that node, and therefore the rate of change of the node voltages is nonnegative. This contradiction proves the impossibility of the assumption above that a node voltage is negative.

Appendix D

Equation (7) is to be proved. Note first, for any three nodes i, j, and k in the network, that R_{ii} is at least as large as both R_{ki} and R_{ji}. Also, R_{jk} is at least as large as the lesser of R_{ki} and R_{ji}. Thus

$$R_{ii}R_{jk} \geqslant R_{ki}R_{ji}. \tag{40}$$

Now note that, similar to (9)

$$1 - v_k(t) = \sum_j R_{jk}C_j \frac{dv_j}{dt} \tag{41}$$

$$1 - v_i(t) = \sum_j R_{ji}C_j \frac{dv_j}{dt} \tag{42}$$

so that, because of (40) and the fact that $v_j(t)$ is monotonic

$$\begin{aligned} &R_{ii}[1 - v_k(t)] - R_{ki}[1 - v_i(t)] \\ &\quad = \sum_j (R_{ii}R_{jk} - R_{ki}R_{ji})\, C_j \frac{dv_j}{dt} \\ &\quad \geqslant 0 \end{aligned} \tag{43}$$

which immediately implies (7).

Appendix E

Bounds for the response $y_i(t)$ of an RC tree to an arbitrary excitation $x(t)$ can be obtained from the upper and lower bounds $v_{ui}(t)$ and $v_{li}(t)$ derived for the unit step response $v_i(t)$.

First, the superposition integral can be used to obtain $y_i(t)$ as

$$\begin{aligned} y_i(t) &= \int_0^t v_i(t - t') \frac{dx(t')}{dt'} dt' \\ &= v_i(t) * dx/dt \end{aligned} \tag{44}$$

where * denotes time convolution. From

$$v_{li}(t) \leqslant v_i(t) \leqslant v_{ui}(t) \tag{45}$$

one obtains

$$v_{li}(t) * dx/dt \leqslant y_i(t) \leqslant v_{ui}(t) * dx/dt, \qquad dx/dt \geqslant 0 \tag{46}$$

$$v_{ui}(t) * dx/dt \leqslant y_i(t) \leqslant v_{li}(t) * dx/dt, \qquad dx/dt \leqslant 0 \tag{47}$$

where $v_{ui}(t)$ and $v_{li}(t)$ are known analytically. From (46) it can be seen that bounds for the ramp response can be obtained simply by integrating the unit step bounds. Equations (46) and (47) apply for monotonic inputs.

For the general case where the excitation $x(t)$ has both positive and negative slopes, one can define the following functions:

$$v_{\mathrm{MAX}i}(t, t') = \begin{cases} v_{ui}(t - t'), & dx/dt' > 0 \\ v_{li}(t - t'), & dx/dt' < 0 \\ 0, & dx/dt' = 0 \end{cases} \tag{48}$$

$$v_{\mathrm{MIN}i}(t, t') = \begin{cases} v_{li}(t - t'), & dx/dt' > 0 \\ v_{ui}(t - t'), & dx/dt' < 0 \\ 0, & dx/dt' = 0. \end{cases} \tag{49}$$

The response $y_i(t)$ is then bounded by

$$\begin{aligned} &\int_0^t v_{\mathrm{MIN}i}(t, t') \frac{dx(t')}{dt'} dt' \leqslant y_i(t) \\ &\qquad \leqslant \int_0^t v_{\mathrm{MAX}i}(t, t') \frac{dx(t')}{dt'} dt'. \end{aligned} \tag{50}$$

References

[1] M. S. Ghausi and J. J. Kelly, *Introduction to Distributed-Parameter Networks*. New York: Holt, Rinehart and Winston, 1968.

[2] U. Kumar, "A bibliography of distributed-RC networks," *IEEE Circuits Syst. Mag.*, vol. 2, no. 2, pp. 9-19, June, 1980.

[3] E. N. Protonotarios and O. Wing, "Theory of nonuniform *RC* lines, Part I: Analytic properties and realizability conditions in the frequency domain," *IEEE Trans. Circuit Theory*, vol. CT-14, no. 1, pp. 2-12; Mar. 1967.

[4] E. N. Protonotarios and O. Wing, "Theory of nonuniform *RC* lines, Part II: Analytic properties in the time domain," *IEEE Trans. Circuit Theory*, vol. CT-14, no. 1, pp. 13-20, Mar. 1967.

[5] K. Singhal and J. Vlach, "Calculation of time responses of nonuniform *RC*-distributed active networks," in *Dig. Tech. Pap., London 1971 IEEE Int. Symp. Electrical Network Theory*, (London, England), pp. 75-76, Sept. 6-10, 1971.

[6] K. Singhal and J. Vlach, "Approximation of nonuniform *RC*-distributed networks for frequency- and time-domain computations," *IEEE Trans. Circuit Theory*, vol. CT-19, no. 4, pp. 347-354, July 1972.

[7] W. C. Elmore, "The transient response of damped linear networks with particular regard to wide-band amplifiers," *J. Appl. Phys.*, vol. 19, no. 1, pp. 55-63, Jan. 1948.

[8] Paul Penfield, Jr. and Jorge Rubinstein, "Signal delay in *RC* tree networks," in *Proc. Second Caltech Conf. on Very Large Scale Integration*, (Pasadena, CA), pp. 269-283, Jan. 19-21, 1981.

[9] Paul Penfield, Jr. and Jorge Rubinstein, "Signal delay in *RC* tree networks," in *Proc. Eighteenth Design Automation Conf.*, (Nashville, TN), pp. 613-617, June 29-July 1, 1981.

[10] Rathin Putatunda, "AUTO-DELAY: A program for automatic calculation of delay in LSI/VLSI chips," in *Proc. Nineteenth Design Automation Conf.*, (Las Vegas, NV), pp. 616-621, June 14-16, 1982.

[11] Rathin Putatunda, "Automatic calculation of delay in custom generated LSI/VLSI chips," in *Proc. IEEE Int. Conf. Circuits and Computers*, (New York, NY), pp. 193-196, Sept. 28-Oct. 1, 1982.

[12] N. P. Jouppi, "TV: An nMOS timing analyzer," in *Proc. Third Caltech Conf. on Very Large Scale Integration*, R. Bryant, ed. Rockville, MD: Computer Science Press, 1983, pp. 71-85.

[13] P. Penfield, Jr., *MARTHA User's Manual*. Cambridge, MA: MIT Press, 1971.

[14] P. Penfield, Jr., "Description of electrical networks using wiring operators," *Proc. IEEE*, vol. 60, pp. 49-53, Jan. 1972.

[15] P. Penfield, Jr., *MARTHA User's Manual, 1973 Addendum*. Cambridge, MA: MIT Dep. of Elect. Eng., 1974.

[16] P. Penfield, Jr., "MARTHA," in *Handbook of Circuit Analysis Languages and Techniques*, R. W. Jensen and L. P. McNamee, eds. Englewood Cliffs, NJ: Prentice-Hall, 1976, pp. 496-535.

[17] L. A. Glasser, "The analog behavior of digital integrated circuits," in *Proc. Eighteenth Design Automation Conf.*, (Nashville, TN), pp. 603-612, June 29-July 1, 1981.

[18] R. D. Thornton, C. L. Searle, D. O. Pederson, R. B. Adler, and E. J. Angelo, Jr., *Multistage Transistor Circuits*. New York: Wiley, 1965, sec. 1.2.

[19] R. D. Thornton *et al.*, *Multistage Transistor Circuits*. New York: Wiley, 1965, sec. 8.2.6.

Jorge Rubinstein (M'81) was born in Mexico City. He received the degree Ingeniero Mecánico Electricista in 1977 from the Universidad Anáhuac in Mexico City. He received the S.M. degree from the Massachusetts Institute of Technology, Cambridge, MA in 1979.

Since 1980 he has been employed by Digital Equipment Corporation in Massachusetts.

Paul Penfield, Jr. (S'57-M'62-F'72) was born in Detroit, MI, on May 28, 1933. He received the B.A. degree in physics from Amherst College, Amherst, MA, in 1955, and the Sc.D. in electrical engineering from the Massachusetts Institute of Technology, Cambridge, MA, in 1960.

At present he is a Professor of Electrical Engineering at MIT, overseeing its interdisciplinary VLSI research program. Between 1974 and 1978 he served as Associate Head of the MIT Department of Electrical Engineering and Computer Science. His professional interests have included noise and thermodynamics, solid-state microwave circuits, electrodynamics of moving media, circuit theory, computer-aided design, APL language extensions, and integrated-circuit design automation.

He is a member of Sigma Xi, American Physical Society, Audio Engineering Society, and Association for Computing Machinery.

Mark A. Horowitz (S'78) received the B.S. and S.M. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1978. He is currently with the Department of Electrical Engineering at Stanford University, Palo Alto, CA, where his Ph.D. research is on computer aids for integrated circuit design.

Since 1979 he has been employed part time by Philips Research Labs, Sunnyvale, CA, working on device modeling and digital circuit design. Mr. Horowitz is a member of Eta Kappa Nu and Sigma Xi.

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

CAMBRIDGE, MASSACHUSETTS 02139

VLSI Memo No. 83-147 July 1983

A New Approach to Timing Analysis of Digital MOS IC's*

John L. Wyatt, Jr.**

ABSTRACT

The waveform bounding approach to fast timing analysis of MOS VLSI circuits is discussed. The idea is to compute rigorous closed-form expressions giving upper and lower bounds for transient voltage waveforms, rather than exact values. The goal is to enable rapid computation without sacrificing user confidence in the results.

*This is a preprint of a conference paper to be presented at the Twenty-Sixth Midwest Symposium on Circuits and Systems, Puebla, Mexico, August 15-16, 1983. This work was supported by the Motorola Research Grant Fund, by the Defense Advanced Research Projects Agency (DoD) under ARPA Order No. 3972, issued by the Office of Naval Research under Contract N00014-80-C-0622, and by the Air Force Office of Sponsored Research under Contract F29620-81-C-0054.

**Department of Electrical Engineering and Computer Science, Room 36-865, M.I.T., Cambridge, MA 02139; telephone (617) 253-6718.

A NEW APPROACH TO TIMING ANALYSIS OF DIGITAL MOS IC'S.

John L. Wyatt, Jr.
Department of Electrical Engineering
and Computer Science
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139
U.S.A.

ABSTRACT

The waveform bounding approach to fast timing analysis of MOS VLSI circuits is discussed. The idea is to compute rigorous closed-form expressions giving upper and lower bounds for transient voltage waveforms, rather than exact values. The goal is to enable rapid computation without sacrificing user confidence in the results.

I. Background and Objectives

Existing approaches to timing analysis and simulation of digital integrated circuits fall, roughly speaking, into three classes:

i) Methods such as SPICE 2 [1] and ASTAP [2], based on essentially exact numerical solutions of the network's differential equations, are accurate and reliable, but quite slow in terms of the needs of the VLSI era.

ii) Specialized MOS timing simulators like MOTIS-C [3] and SPLICE [4] rely on table lookup of device characteristics for speed, and save additional time by terminating a Newton-Raphson or similar interation before convergence is reached. SPLICE is in addition a mixed-mode circuit, timing and logic simulator and uses a selective trace algorithm to exploit latency. In both these programs the termination of an iterative step prior to convergence saves time at the cost of accuracy and, in some instances, of numerical stability [5]. The improvement in speed over SPICE 2 is typically one to two orders of magnitude for SPLICE [4] and about two orders of magnitude for MOTIS-C [6].

iii) More recently, some researchers are exploring an alternate approach to timing analysis and simulation based on a radically simplified electrical description of the network. RSIM [7], CRYSTAL [8], and TV [9] fall at the far end of the speed-accuracy tradeoff curve from SPICE 2. A MOSFET is typically represented in these programs by an extremely simplified model: a linear resistor in series with a switch. And a polysilicon or diffusion line is represented by a lumped capacitance in RSIM, or by a delay in CRYSTAL and TV obtained by simply averaging the upper and lower delay bounds obtained by Rubinstein, Penfield, and Horowitz [10]. These programs are potentially very fast and have a number of attractive user-oriented features. The drawback, of course, is that there are no absolute known limits to the error in their total delay estimates. The user can never be sure the answers they give are close enough.

The objective of the waveform bounding approach to timing analysis and simulation is to combine the computational speed that results from avoiding the numerical solution of differential equations with the user confidence in the result that comes from rigorous error bounds. Our attack on the timing analysis problem is based on a careful fundamental study of the differential equations describing the dynamics of gates, pass transistors, interconnect, and the standard digital circuits constructed from them.

The MIT group currently working on this project includes Profs. Paul Penfield, John Wyatt, and Lance Glasser and graduate students Charles Zukowski and Paul Bassett. In addition, Mark Horowitz [10, 11] is currently completing a dissertation on MOS timing analysis at Stanford.

II. Response Bounds for Interconnect

2.1) Linear Interconnect Models

This section summarizes the results obtained in [10]. In this work an MOS signal distribution network as shown in Fig. 1 is modelled as a branched linear RC line, i.e., an RC tree, as in Fig. 2.

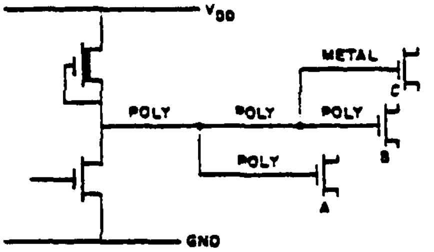


Figure 1. Typical MOS signal-distribution network. The inverter is shown driving three gates.

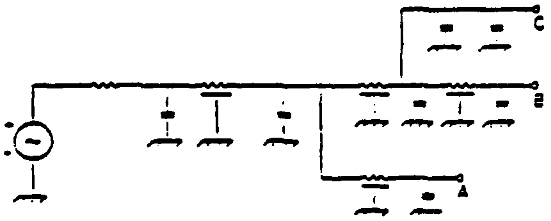


Figure 2. The linear RC tree shown above is a model for the network of Fig. 1. The voltage source is a unit step at time t = 0.

For any two nodes in the network, $R_{\ell m}$ is defined as the sum of the resistances along the route consisting of the intersection of the path from the input to node ℓ with the path from the input to node m, as illustrated in Fig. 3. The three time

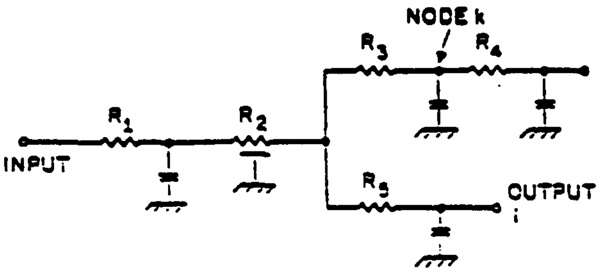


Figure 3. Illustration of resistance terms. For this network, $R_{ki} = R_1 + R_2$, $R_{kk} = R_1 + R_2 + R_3$, and $R_{ii} = R_1 + R_2 + R_5$.

constants used to derive response bounds are

$$T_p \triangleq \sum_k R_{kk}\, C_k \tag{1}$$

$$T_{Di} \triangleq \sum_k R_{ki}\, C_k \tag{2}$$

$$T_{Ri} \triangleq \sum_k (R_{ki}^2\, C_k)/R_{ii}\ , \tag{3}$$

where the summations are taken over all nodes of the network. The derivation in [10] shows that $\underline{v}_i(t) \le v_i(t) \le \bar{v}_i(t)$, for all $t \ge 0$, where $v_i(t)$ is the actual zero state step response at any terminal node i, and the bounds $\underline{v}_i(t)$ and $\bar{v}_i(t)$ are given by

$$\underline{v}_i(t) \triangleq \begin{cases} 0, \ 0 \le t < T_{Di} - T_{Ri} \\ 1 - \dfrac{T_{Di}}{t+T_{Ri}}\ , \ T_{Di}-T_{Ri} \le t < T_p-T_{Ri} \\ 1 - \dfrac{T_{Di}}{T_p} \exp[(T_p-T_{Ri}-t)/T_p], \ T_p-T_{Ri} \le t \end{cases} \tag{4}$$

$$\bar{v}_i(t) \triangleq \begin{cases} 1 + \dfrac{t-T_{Di}}{T_p}, \ 0 \le t < T_{Di}-T_{Ri} \\ 1 - \dfrac{T_{Ri}}{T_p} \exp[(T_{Di}-T_{Ri}-t)/T_{Ri}], \ T_{Di}-T_{Ri} \le t, \end{cases} \tag{5}$$

as illustrated in Fig. 4.

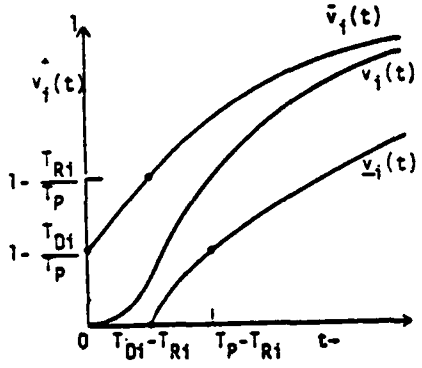


Figure 4. Form of the bounds, with the distance from the exact solution exaggerated for clarity.

The time required to compute these bounds grows only linearly with the number of elements in the network. Recent applications of this result include [8, 9, 12]. The ultimate goal of this portion of the project is to derive a hierarchy of such bounds, permitting the user to trade off accuracy for computation time.

2.2) Nonlinearities Affecting Interconnect

The linear circuit model in Fig. 2 fails to incorporate three types of nonlinearities present in Fig. 1 or related circuits: the nonlinear output resistance of the inverter, the nonlinear gate-to-channel capacitance of the MOSFET loads, and the nonlinear capacitance from any diffusion line to substrate. This section describes recent work [13-16] that allows the bounds for linear networks [10] to be applied to RC lines incorporating such nonlinearities. (Further research is needed for branched lines, i.e. RC trees.)

Using the notation and sign conventions illustrated in Fig. 5, the

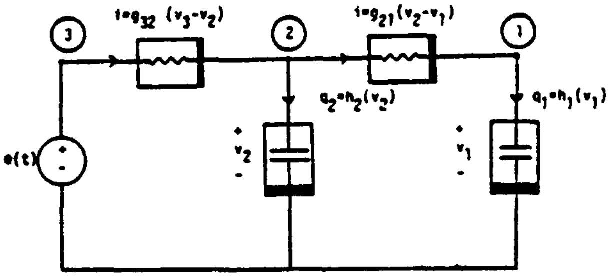


Figure 5. Two-capacitor example of a nonlinear, nonuniform RC line.

state equations for any nonuniform, nonlinear lumped RC line with N capacitors can be written in the form

$$\dot{v}_j = \frac{1}{C_j(v_j)}\left[g_{j+1,j}(v_{j+1}-v_j)-g_{j,j-1}(v_j-v_{j-1})\right], \quad 1 \le j \le N, \tag{6}$$

where $g_{10} \triangleq 0$, $v_{n+1} \triangleq e$, and the capacitor constitutive relations $q_j = h_j(v_j)$ are continuously differentiable with $C_j(v_j) \triangleq h_j'(v_j) > 0$ everywhere. We assume the resistor curves are continuously differentiable, strictly increasing, and pass through the origin.

The state space for (6) is the set of all vectors of capacitor voltages $\underline{v} \in \mathbb{R}^N$. The four subsets of the state space defined below play a key role in the theoretical development [15].

Def. 1

$$P \triangleq \{\underline{v} \in \mathbb{R}^N \mid v_j \ge 0,\ 1 \le j \le N\} \tag{7}$$

$$L(e) \triangleq \{\underline{v} \in \mathbb{R}^N \mid v_j \le e,\ 1 \le j \le N\} \tag{8}$$

$$S_I(e) \triangleq \{\underline{v} \in \mathbb{R}^N \mid v_j \le v_{j+1},\ 1 \le j \le N\} \tag{9}$$

$$T_I(e) \triangleq \{\underline{v} \in \mathbb{R}^N \mid \dot{v}_j \ge 0 \text{ in } (6),\ 1 \le j \le N\} \tag{10}$$

See Figs. 6-9. Roughly speaking, P is the set of all

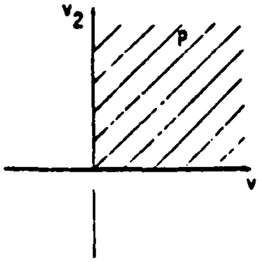


Figure 6. The set P for the network in Fig. 5.

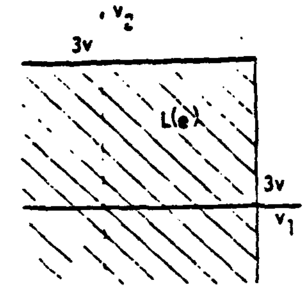


Figure 7. The set L(e) for the network in Fig. 5, when e=3v.

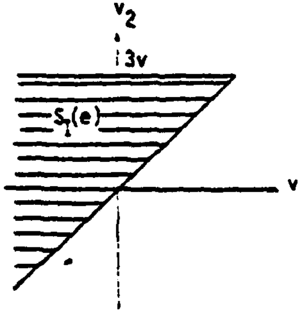


Figure 8. The set $S_I(e)$ for the network in Fig. 5, when e=3v.

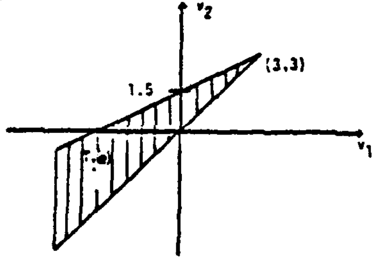


Figure 9. The set $T_I(e)$ for the network in Fig. 5, if both resistors are 1Ω and e=3v.

vectors of positive capacitor voltages, L(e) is the set of all vectors of capacitor voltages less than e, $S_I(e)$ consists of spatially increasing voltages as one traverses the line in the direction of the source, and $T_I(e)$ consists of temporally increasing (non-discharging) voltages.

Def. 2

For a vector ordinary differential equation $\dot{\underline{x}} = \underline{f}(\underline{x}, t)$, $\underline{x} \in \mathbb{R}^N$, a (possibly time-dependent) set of states $S(t) \subset \mathbb{R}^N$, is positive invariant iff for all $t_1 \geq 0$,

$$\underline{x}(t_1) \in S(t_1) \Rightarrow \underline{x}(t) \in S(t), \text{ for all } t \geq t_1.$$

Roughly speaking, a positive invariant set is a "cage" *that traps every trajectory that enters it*. Positive invariant sets give us information about the location of trajectories without requiring that we solve the network differential equations.

Lemma 1 [15]

For any nonlinear, nonuniform RC line, if e(t) is continuously differentiable, $e(t) \geq 0$, and $\dot{e}(t) \geq 0$ for $t \geq 0$, then P, L(e), $S_I(e)$ and $T_I(e)$ are all positive invariant.

From Lemma 1 we conclude, for example, that at any instant during an "up" transition from equilibrium, $v_j(t) \geq 0$, $v_j(t) \leq e$, $v_j(t) \leq v_{j+1}(t)$ and $\dot{v}_j(t) \geq 0$, $1 \leq j \leq N$.

The following partial orderings let us compare the resistances of nonlinear resistors and the capacitances of nonlinear capacitors [16].

Def. 3

For a collection of 2-terminal nonlinear resistors, with R_j characterized by $i=g_j(v)$, we say $R_j \leq R_k$ iff $[g_j(v) - g_k(v)]\, v \geq 0$, for all v. For nonlinear capacitors, with C_j characterized by $i = C_j(v)\dot{v}$, we say $C_j \leq C_k$ iff $C_1(v) \leq C_2(v)$ for each v.

Thus we compare large-signal (or "chord") resistances, but incremental capacitances. See Fig. 10. Using Lemma 1 and Def. 3, we can formulate and prove [16] the

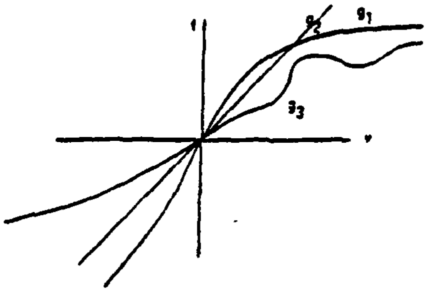


Figure 10. Resistor R_k is characterized by $i = g_k(v)$, k = 1,2,3. Resistors R_1 and R_2 pass current more easily than R_3, i.e. R_3 is a larger resistor than the other two. No such global comparison between R_1 and R_2 is possible.

Monotone Response Theorem for Nonlinear, Nonuniform RC Lines.

Given a nonlinear RC line as described above. *Suppose that (because of circuit parameter uncertainty, the use of linearized models for nonlinear elements, replacing the exact input by input bounds, etc.,)* we do one or more of the following:

a) overestimate the input e(t),

b) underestimate one or more R's,

c) underestimate one or more C's.

The resulting circuit model will then necessarily overestimate the output $v_j(t)$ at each instant t during "up" transitions (i.e., during transitions where $e \geq 0$, $\dot{e} \geq 0$ throughout.)

A similar result holds for "down" transitions and estimate errors of the opposite sign. Using part a) of the assumptions, this theorem allows us to computationally propagate upper and lower signal bounds through the network. Using parts b) and c), it allows us to replace a nonlinear line by two linear ones, one strictly faster and one strictly slower, to which the linear network bounds (4,5) in turn apply. We have not yet succeeded in finding a generalization of this result that will apply to nonlinear RC trees.

III. An Approach to Waveform Bounding for MOS Logic Gates

The results reported here are due to Charles Zukowski, and apply to MOS device models of the form

$$i_D = \frac{W}{L}\ f(v_{GB}, v_{DB}, v_{SB}), \qquad (11)$$

where D,G,S and B refer to drain, gate, source and substrate, respectively. For specificity we consider only n-channel devices in this paper. No special algebraic form for f is assumed, only that f is continuously differentiable and satisfies the natural monotonicity conditions

$$\frac{\partial f}{\partial v_{GB}} \geq 0, \frac{\partial f}{\partial v_{DB}} \geq 0, \frac{\partial f}{\partial v_{SB}} \leq 0 \qquad (12)$$

everywhere. Thus a wide variety of device models are allowed, with the exception that (11) does not allow for short-channel effects.

Our approach will be to reduce a multiple-input logic gate by steps to an "equivalent bounding inverter" and then to find bounds for the response of this inverter.

3.1) Reduction of Series-Parallel Transistor Network to "Equivalent Bounding Transistor"

We have developed a method for reducing any series-parallel transistor network to a single "equivalent bounding transistor." Using the technique recursively, one can replace the pullup or pulldown network of a multiple-input gate by a single transistor and have rigorous bounds for the error produced by this simplification.

For example, a parallel connection of N transistors, all identical except for widths, lengths and gate voltages, satisfies

$$i = f(\vec{v}_{GB}, v_{DB}, v_{SB}) \doteq \sum_{j=1}^{N} \frac{W_j}{L_j} f(v_{GBj}, v_{DB}, v_{SB}), \qquad (13)$$

where $\vec{v}_{GB}$ is the vector of gate voltages. We have proven that, because of the assumptions (12), there exist W_{eq}, L_{eq} independent of v_{GB}, and $\underline{v}_{GB}$ and $\bar{v}_{GB}$ that depend on v_{GB}, such that (13) can be replaced by the simpler bounds

$$\frac{W_{eq}}{L_{eq}} f(\underline{v}_{GB}, v_{DB}, v_{SB}) \leq i \leq$$

$$\frac{W_{eq}}{L_{eq}} f(\bar{v}_{GB}, v_{DB}, v_{SB}), \qquad (14)$$

for all $v_{DB} \geq v_{SB}$, describing a single transistor with a range of gate voltages. The function f is the same throughout (13) and (14). Figure 11 illustrates this process for N = 2.

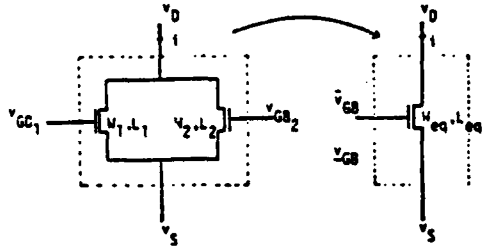


Figure 11. Replacing a parallel transistor network by an "equivalent bounding transistor". The cost of this simplification is that the exact value of i for the network on the left is replaced by a range of values in the simpler model corresponding to $\underline{v}_{GB} \leq v_{GB} \leq \bar{v}_{GB}$.

3.2) Reducing a Multiple-Input Gate to an "Equivalent Bounding Inverter"

A gate can be modelled as an "equivalent bounding inverter" by performing the reduction outlined in section 3.1 on both the pullup and pulldown networks, reducing each to a single transistor. Initial trials, comparing the result with SPICE simulations of the original network, indicate that the resulting bounds for i_{out} (v_{out}) differ from the exact values by only about ± 10% for practical circuits.

3.3) Bounding the Response of an Inverter and Load to Input Transitions

When applied to some multiple-input gates, the reduction procedure described in the previous two subsections may yield an inverter in which the pullup gate is externally driven. But for simplicity we consider here only the case of a standard NMOS depletion - load inverter as in Fig. 12.

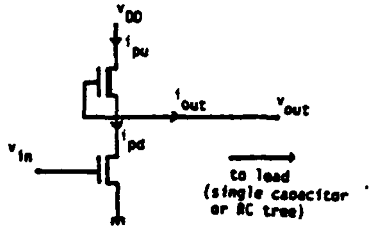


Figure 12. Depletion-load inverter.

To bound the response time of the loaded inverter we need simple bounds on the function i_{out} (v_{out}, v_{in}), which is the difference of the pullup and pulldown currents:

$$i_{out}(v_{out}, v_{in}) = i_{pu}(v_{out}) - i_{pd}(v_{out}, v_{in}). \quad (15)$$

Simple linear bounds on both the pullup and pulldown currents are shown in Fig. 13. The resulting bounds for the output curve $i_{out}(v_{out})$ depend on $v_{in}(t)$.

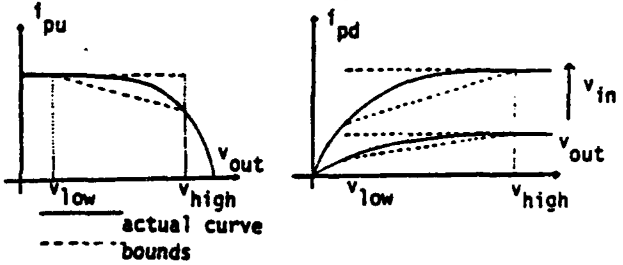


Figure 13. Simple linear bounds on the pullup and pulldown currents. The latter depend on v_{in}, and hence on t.

Initial simulations using this approach indicate that the delay bounds for these simplified models differ from the delays obtained from SPICE simulations by about ± 15%.

IV. Further Work in Progress

Much work remains to be done before the theoretical basis for the waveform bounding approach to timing analysis is complete. Among the larger remaining problems are:

1. extending the Penfield-Rubinstein bounds to incorporate time-varying source resistances, such as those modelling the pulldown current in Fig. 13,

2. finding bounds for the response of an RC tree containing pass transistors,

3. investigating the tolerance in the bounds obtained so far and finding tighter ones where necessary, and

4. incorporating effects of the Miller capacitance into bounds.

ACKNOWLEDGMENT

This project was supported by the Motorola Research Grant Fund.

REFERENCES

[1] L.W. Nagel, "SPICE 2, A Computer Program to Simulate Semiconductor Circuits," Memorandum No. ERL M520, Electronics Research Laboratory, U.C. Berkeley, Berkeley, CA, 94720, May, 1975

[2] W.T. Weeks, A.J. Jimenez, G.W. Mahoney, D. Mehta, H. Qassemzadeh, and T.R. Scott, "Algorithms for ASTAP-A Network Analysis Program," IEEE Trans. Circuit Theory, vol. CT-20, pp. 628-634, Nov. 1973.

[3] S.P. Fan, M.Y. Hsueh, A.R. Newton, and D.O. Pederson, "MOTIS-C: A New Circuit Simulator for MOS LSI Circuits," Proc. 1977 IEEE Int. Symp. on Circuits and Systems, pp. 700-703, April, 1977.

[4] A.R. Newton, "Techniques for the Simulation of Large Scale Integrated Circuits," IEEE Trans. Circuits and Systems, vol. CAS-26, no. 9, pp. 741-749, Sept. 1979.

[5] G. De Micheli and A. Sangiovanni-Vincentelli, "Characterization of Integration Algorithms for the Timing Analysis of MOS VLSI Circuits," Int. Jour. of Circuit Theory and Appl., vol. 10, pp. 299-309, 1982.

[6] M.Y. Hsueh, A.R. Newton, and D.O. Pederson, "New Approaches to Modelling and Electrical Simulation of LSI Logic Circuits," Journees d'Electronique, pp. 403-413, Lausanne, Switzerland, 1977.

[7] C.J. Terman, "User's Guide to NET, PRESIM, and RNL/NL," private communication, available from the author.

[8] J. Ousterhut, "CRYSTAL: A Timing Analyser for NMOS VLSI Circuits," Proc. 3rd Caltech Conf. on VLSI, March 1983.

[9] N. Jouppi, "TV: An NMOS Timing Analyser," Proc. 3rd Caltech Conf. on VLSI, March 1983.

[10] J. Rubinstein, P. Penfield, Jr., and M.H. Horowitz, "Signal Delay in RC Tree Networks," to appear in IEEE Trans. CAD.

[11] M. Horowitz, "Timing Models for MOS Pass Networks," Proc. 1983 IEEE Int. Symp. Circuits and Systems, pp. 198-201.

[12] R. Putanunda, "Auto Delay: A Program for Automatic Calculation of Delay in LSI/VLSI Chips," ACM IEEE 19th Design Autom Conf. Proc., pp. 616-621, 1982.

[13] J.L. Wyatt, Jr., "Inequality Theorems for Nonlinear Differential Equations," VLSI Memo 82-127, Dept. of Elect. Eng. and Comp. Sci., MIT, Cambridge, MA, 02139, Nov. 1982. All VLSI Memos are available from the Microsystems Program Office, Room 36-575, MIT.

[14] J.L. Wyatt, Jr., "Monotone Behavior of Nonlinear RC Meshes," VLSI Memo 82-128, Nov. 1982.

[15] J.L. Wyatt, Jr., and P. Bassett, "Spatial Monotonicity and Positive Invariance for Nonlinear RC Lines and Trees," VLSI Memo 83-140, April 1983.

[16] J.L. Wyatt, Jr., "Partial Ordering and Monotone Sensitivity for Nonlinear RC Meshes and Lines," VLSI Memo 83-145, June, 1983.

Ion implantation of Si and Be in $Al_{0.48}In_{0.52}As$

A. N. M. Masum Choudhury, W. Rowe, K. Tabatabaie-Alavi, C. G. Fonstad, K. Alavi,[a)]
and A. Y. Cho[a)]

Department of Electrical Engineering and Computer Science and Center for Materials Science and Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 20139

(Received 14 February 1983; accepted for publication 13 April 1983)

Electrical characterization of (Al,In)As ion implanted with Be and Si following low and high dose multiple energy implant schedules, and annealed between 740 and 815 °C with a pyrolytic silicon dioxide cap is reported. Only very low activation of Be is achieved (< 3%). Silicon activation is considerably higher (> 40%) and increases with increasing anneal temperature. However, a high concentration *n*-type surface layer is found on samples annealed at 815 °C. This surface layer is not found on similarly annealed samples which were not implanted, or which were implanted with phosphorus.

PACS numbers: 61.70.Tm, 72.80.Ey

INTRODUCTION

Recent interest in the III-V heterojunction structures for high performance microwave[1] and optoelectronic devices[2] and also for high speed transistor logic[3] has increased the need for lattice-matched compound semiconductor materials. Various systems, such as GaAs/(Al,Ga)As and InP/(In,Ga)(As,P), have been utilized. Another quaternary system is (Al,Ga,In)As which has a wider maximum band gap (1.53 eV[4]) than InP (1.35 eV) at 300 °K, and most importantly can be grown lattice matched to (In,Ga)As and to InP substrates by both molecular beam epitaxy (MBE)[5] and metal organic chemical vapor desposition (MOCVD)[9] as an attractive alternative to the phosphorus containing (In, Ga)(As,P) for the epitaxial layers in heterojunction structures. Since this quaternary contains only one group V element, As, which is much less volatile than phosphorus, it may also be preferred to InP because of its stability during high temperature processing. Doping of the wide band-gap ternary limit of this quarternary, $Al_{0.48}In_{0.52}As$ with silicon, tin, and beryllium during MBE growth has been reported.[6-8] Selective doping of this material by ion implantation or diffusion is, however, also required for forming *n*- and *p*-type regions in device work. In this letter, we discuss for the first time the implantation and annealing of Si and Be in $Al_{0.48}In_{0.52}As$.

PROCEDURES

Room temperature ion implantation was done on 1.8–2-μm-thick (Al,In)As undoped layers grown on ⟨100⟩ oriented S-doped and as well as Fe-doped InP substrates by MBE. The layers were semi-insulating, having resistivity greater than 10^7 Ω cm.

Two multiple energy Si implant schedules were used: a "low" dose implant for creating a low carrier concentration *n* layer and a "high" dose implant for a high carrier *n* layer. For Be, a single two-energy implant schedule was used. These schedules are detailed in Table I.

To anneal the samples, a thin layer (~ 40 nm) of pyrolytically grown SiO_2 was deposited on the implanted surface, and the samples were annealed in a forming gas atmosphere for 15–20 min at various temperatures ranging from 740 to 815 °C.

The carrier concentration and mobility of the silicon implanted layer was profiled using differential Hall measurements.[10] After each measurement ~ 22–88 nm of the material was removed by using an etching solution of H_2O, H_2SO_4, and H_2O_2 (145:4:1 by volume) for 10–40 sec (etch rate: 2.2 nm/sec) and then the measurement was repeated. *C-V* measurements on Au Schottky diodes were used to determine the implanted Be profile.[11] Here also the repeated etch-and-measure technique was used. Thickness measurements on a control sample etched simultaneously with the above samples were used to determine the thickness etched off and flatness of the surface.

EXPERIMENTAL RESULTS

The measured doping profiles of the silicon implants are shown in Fig. 1. The two dotted curves are the silicon profiles predicted by Lindhard–Scharff–Schiott (LSS) theory.[10] The upper dotted curve represents the LSS profile for the high dose silicon implant schedule. The upper three solid curves passing through the experimental points are the measured doping profiles for the high dose schedule annealed at three different temperatures, 740, 760, and 815 °C, for 15, 20, and 20 min, respectively. The lower solid curve presents

TABLE I. Silicon and beryllium multiple energy implant schedules used to dope (Al, In)As *n* and *p* type. All implants were performed with the substrates nominally at room temperature. The anneal procedures used are detailed in the text.

Schedule	Energy (keV)	Dose ($\times 10^{-13}$ cm^{-2})
Silicon low dose	100	1.06
	250	2.66
Silicon high dose	100	5.60
	250	14.0
Beryllium	50	5.35
	120	9.18

[a)] Also at Bell Laboratories, Murray Hill, NJ 07974.

0021-8979/83/084374-04$02.40

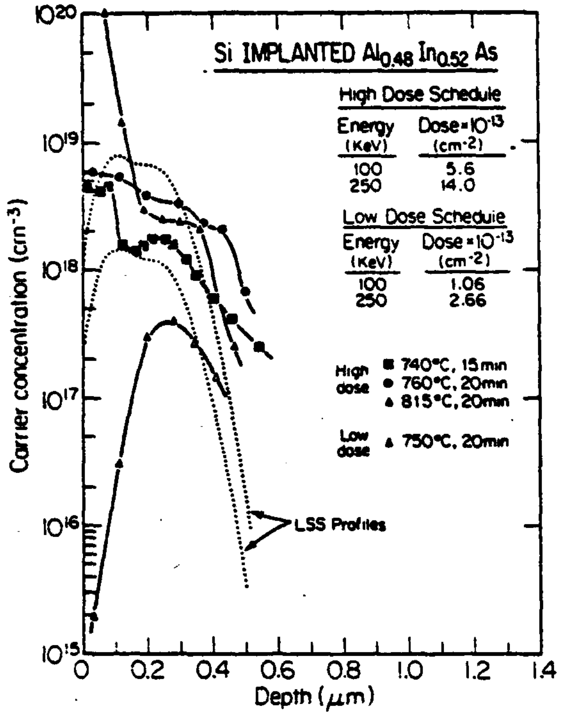


FIG. 1. Doping profiles of silicon implanted into (Al,In)As. The dotted curves indicate the profiles expected on the basis of LSS theory for the high and low dose schedules presented in Table I. The solid curves passing through the data points are the experimentally measured profiles for samples implanted with the high dose schedule annealed at 740 °C for 15 min (■), 760 °C for 20 min (●), and at 815 °C for 20 min (▲), and with the low dose schedule and annealed at 750 °C for 20 min (△). A pyrolytically deposited SiO_2 cap was used during all anneals.

the measured doping profile for the low dose schedule annealed at 750 °C for 20 min. The mobility profiles corresponding to the samples and data in Fig. 1 are presented in Fig. 2.

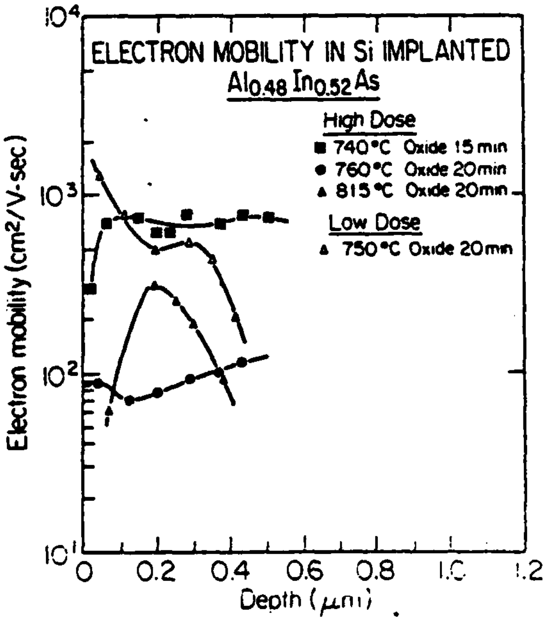


FIG. 2. Room temperature electron Hall mobility profiles corresponding to the silicon doping profiles presented in Fig. 1. The notation used in this figure is the same as in Fig. 1.

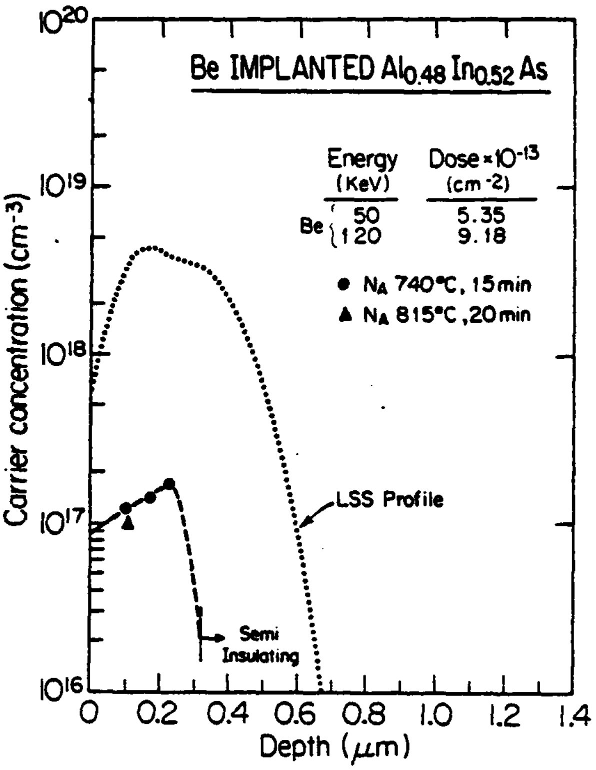


FIG. 3. Doping profiles of beryllium implanted into (Al,In)As. The dotted curve is the profile predicted by LSS theory and the solid and dashed curve is drawn through the experimental data. Measurements on two samples, one annealed 15 min at 740 °C (●), and the other annealed 20 min at 815 °C (▲), are presented.

The doping profiles measured by *C-V* measurements on the Be-implanted samples are presented in Fig. 3 along with the profile predicted by LSS theory (dotted line). Three data points for a sample annealed 15 min at 740 °C and one for a sample annealed 20 min at 815 °C are shown, along with a solid curve indicating the corresponding doping profile. A comparison of the areas under the measured profile and the LSS profile indicate an activation efficiency of approximately 3%.

As a result of the observation of the high carrier concentration near the surface of the Si-implanted samples annealed at 815 °C (see Fig. 1), additional annealing experiments were performed on unimplanted and phosphorous implanted samples. The results of these studies will be presented and discussed in the Discussion section which follows.

DISCUSSION

Looking first at Fig. 1, the Si implant profiles, and specifically the measured curves, it is seen that as the annealing temperature is increased, the profiles fall off more sharply, e.g., the tail of the implant profile annealed at 740 °C extends furthest into the material, whereas the one annealed at 815 °C does the least. This observation is consistent with similar phenomenon in (In,Ga)As reported elsewhere.[10]

The experimental profiles for the higher dose schedule close to the surface indicate a very high carrier concentration near the surface, similar to the conducting layer which has also been reported by Davies *et al.* in the case of furnace annealed silicon implanted InP.[12] This phenomenon artifically increases the apparent activation efficiency to more than 100%. In our case, the higher dose implant annealed at 815 °C has an apparent activation of about 400% while the samples annealed at 760 and 740 °C show 97% and 47% activation, respectively. The high activation efficiency in the first case is due to the surface layer.

To better understand the presence of this high concentration surface layer, an unimplanted sample of (Al,In)As coated with 400 Å of SiO_2 was annealed successively at 750, 785, and 815 °C for periods of 20 min and the sheet resistivity was measured after the anneal at each temperature. The sheet resistance was too high to be measured after the 20-min anneal at 750 °C. After the additional 20 min at 785 °C, the sheet resistance was 4.5×10^6 Ω per square, and after another 20 min at 815 °C the sheet resistance was 6.2×10^4 Ω per square. For comparison, the sheet resistance of a sample implanted following the high dose Si schedule and annealed at 815 °C was measured to be 372 Ω per square. This is $\simeq$200 times lower than the unimplanted sample annealed at 815 °C, and is $\simeq$5 times lower than the value of 1500–2000 Ω per square that would be expected if the conductivity was due solely to full activation of implanted Si. There is clearly a significant difference in the surface layer that forms on samples that have been implanted with silicon and on unimplanted samples.

To examine the role of the implantation damage in the creation of the high concentration surface layer, the same (Al,In)As sample was implanted with phosphorus following the high dose schedule. Phosphorus was chosen because it is adjacent to Si in the periodic table, and should be electronically relatively inert in (Al,In)As. A pyrolytic SiO_2 layer (40 nm thick) was deposited on the surface and the sample was annealed 20 min at 815 °C. After the SiO_2 was removed, the sheet resistance was measured to be 2×10^4 Ω per square. This is lower than before by approximately 3 times but still much higher than for the Si-implanted samples.

It should be noted that a high concentration *n*-type surface layer was not observed when the Be-implanted sample was annealed at 815 °C either. Thus, we can conclude that the damage caused to the surface during ion implantation is in itself not sufficient to result in a high concentration surface layer after an 815 °C anneal. It seems either that the presence of Si contributes to the surface change, or that the presence of P and/or Be aids in stabilizing the surface.

While the high dose Si implants all show a high surface concentration, there is no evidence of a high concentration surface layer on the sample implanted following the low dose schedule. The activation efficiency for this lower dose implant is only 23%. Although this activation is lower than the comparable activation of silicon implants in InP or GaAs, this may be related to the higher band gap of this material. Annealing at higher temperatures or implanting with the substrates heated, may consequently improve the activation efficiency.

Looking now at Fig. 2, the mobility data, if this figure is viewed along with Fig. 1, it is noted that the profiles having the lowest carrier concentration have the highest mobility and vice versa, with the exception of the high dose sample annealed at 740 °C. This sample has the highest mobility although the carrier concentration of the low dose sample is lower. This difference may simply reflect variations in the initial epitaxial layers. The mobility values at different concentration levels are in very good agreement with the reported mobility of tin[7] and silicon[6] doped MBE-grown (Al,In)As. The lower values of the mobility deeper into the material viewed along with the lower implant activation in that region is felt to be indicative of the fact that significant damage still remains after annealing.

Turning now to the Be implant data, Fig. 3, the activation was found to be only 2.7%. The depth of the profile is also very shallow. Annealing at a higher temperature (815 °C) did not help change the sheet resistance or concentration appreciably. It is reported that for (In,Ga)As,[11] InP,[13] and GaAs (Ref. 14) the activation of Be is always much less than that of Si. Possibly a similar phenomenon is occurring here. (Al,In)As, having higher band gap than any of those materials, would be expected to require still higher anneal tempertures and show even lower activation.

CONCLUSIONS AND SUMMARY

This work has demonstrated that it is possible to dope (Al,In)As *n*- and *p*-type with ion implantation. The general behavior of ion implanted silicon and beryllium is in many ways similar in this material to what it is in related III-V compounds, but as with other III-V's, the need for additional research on ion implantation and annealing technologies is also clear.

Activation in the case of Be implantation is very low, approximately 3%, and the activation of Si is also low, $\simeq$40%, although it is much higher than that of Be. Increasing the post-implant anneal temperature from 740 to 815 °C did not increase the Be activation but did increase slightly the Si activation and the associated carrier mobilities.

The most significant result of the higher anneal temperature was the appearance of a high concentration surface layer on the Si-implanted samples. This effect was not observed when unimplanted samples, and samples implanted with phosphorus (at the same doses and energies) and beryllium (similar dose but lower energies), were similarly annealed. Experiments directed at determining the reasons for this different behavior and thereby ways of stabilizing the surface, are planned. If, for example, implanting P does stabilize the surface, then no high concentration layer should be found when Si and P are implanted together. If it is Si itself which causes the problem, then a high concentration layer would be expected even when P and Si are both implanted. To help further determine the roles played by damage and the various ionic species in causing or preventing surface changes during the anneal, proton bombardment could also be used to damage the surface. The effect of reducing the implant energies must also be considered.

The observation of surface changes is important to the more general issue of increasing the activation efficiency of any species because the present results indicate that still higher anneal temperatures should be considered. This, and the use of substrate heating during implantation would appear to be logical steps to take in attempting to increase the activation and mobilities, reduce the sheet resistances, and remove implant damage.

ACKNOWLEDGMENTS

During the course of this work, extensive use was made of the facilities and services of the Microelectronics Laboratory and Ion Implantation Central Facility of the MIT Center for Materials Science and Engineering, a National Science Foundation Materials Research Laboratory supported in part by Grant No. DMR-78-24185; the authors would particularly like to acknowledge the assistance of Mark Rothman in performing the necessary ion implantations. The differential Hall measurements were performed using the equipment of Professor H. C. Gatos' Electronic Materials Research Group within the Center and the Department of Materials Science. MIT portion of this work was supported by the Defense Advanced Research Projects Agency and monitored by the Office of Naval Research, Grant No. N0014-80-C-0622 and the National Science Foundation through the MIT Center for Materials Science and Engineering.

[1] H. Ohno, J. Barnard, C. E. C. Wood, and L. F. Eastman, Electron Dev. Lett. EDL-1, 154 (1980).
[2] B. I. Miller, J. H. McFee, R. J. Martin, and P. K. Tien, Appl. Phys. Lett. 33, 44 (1978).
[3] K. Tabatabaie-Alavi and Clifton G. Fonstad, Electron Dev. Lett. EDL-3, 234 (1982).
[4] E. E. Matyas, Sov. Phys. Semicond. 13, 1194 (1979).
[5] D. Olego, T. Y. Chang, E. Silberg, E. A. Caridi, and A. Pinczuk, Appl. Phys. Lett. 41, 476 (1982).
[6] K. Y. Cheng and A. Y. Cho, J. Appl. Phys. 53, 4411 (1982).
[7] K. Y. Cheng, A. Y. Cho, and W. R. Wagner, J. Appl. Phys. 52, 6328 (1981).
[8] K. Y. Cheng, A. Y. Cho, and W. A. Bonner, J. Appl. Phys. 52, 4672 (1981).
[9] C. B. Cooper, III, M. J. Ludowise, W. T. Dietze, S. Fonte, V. M. Sardi, and R. R. Saxenz, "Growth of AlGaInAs by OM-VDE for Photovoltaic Applications," presented at the Fifth International Conference on Vapor Growth and Epitaxy, Coronado, CA, July 19–24, 1981 (unpublished).
[10] A. N. M. M. Choudhury, N. J. Slater, K. Tabatabaie-Alavi, and C. G. Fonstad, Appl. Phys. Lett. 40, 607 (1982).
[11] K. Tabatabaie-Alavi, A. N. M. M. Choudhury, N. J. Slater, and C. G. Fonstad, Appl. Phys. Lett. 40, 517 (1982).
[12] D. Eirug Davies, E. F. Kennedy, J. J. Comer, and J. P. Lorenzo, Appl. Phys. Lett. 36, 922 (1980).
[13] J. P. Donnelly and C. E. Hurwitz, Appl. Phys. Lett. 31, 418 (1977).
[14] W. V. McLevige, M. J. Helix, K. V. Vaidyanathan, and B. G. Streeman, J. Appl. Phys. 48, 3342 (1977).

Rapid thermal annealing of Se and Be implanted InP using an ultrahigh power argon arc lamp

A. N. M. Masum Choudhury, K. Tabatabaie-Alavi, and C. G. Fonstad
Department of Electrical Engineering and Computer Science and Center for Material Science and Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

J. C. Gelpey
Eaton Ion Beam Systems Division, Beverly, Massachusetts 01915

(Received 14 April 1983; accepted for publication 1 June 1983)

A 100-kW water-walled dc argon arc lamp has been used for the first time to post anneal ion-implanted InP samples. Temperatures as high as 925 °C and short cycle times (3 and 10 s) are used for the process. Se and Be were ion implanted into room temperature and hot substrate InP samples. A sputter deposited SiO_2 layer, 120 mm thick, covering all wafer surfaces was used as an encapsulant during the lamp anneal. Hot substrate Se implants (400 keV, 1.8×10^{14} cm^{-2}, 200 °C) show an average mobility of 1415 cm^2/Vs and an activation of ~63%, and room-temperature Be implants (50 keV, 3.35×10^{13} cm^{-2} and 150 keV, 5.74×10^{13} cm^{-2}) an average mobility of 88 cm^2/Vs and activation of ~45%. This annealing technique is straightforward and gives activations and mobilities comparable to, or better than, the best furnace anneals with sharp profiles and simplified surface encapsulation.

PACS numbers: 61.70.Tm, 72.80.Ey, 81.40.Ef, 81.40.Rs

The development of ion implantation technology for InP is very important because of its potential use in the fabrication of optoelectronic and microwave devices. Post-implant annealing of the implanted material to remove the unwanted damage introduced during ion implantation is perhaps the most important step in the entire process. Annealing of ion-implanted InP has previously been reported using a conventional furnace,[1,2] pulsed laser beams,[3] pulsed electron beams,[4] and a graphite strip heater.[5] In this letter, we present for the first time the mobility and carrier concentration profiles for room-temperature and hot (200 °C) Se and Be implants in InP annealed using an ultrahigh power water-walled dc argon arc lamp for very short periods: 3 and 10 s.

The InP samples used in this work were cut to 1 × 1 cm from ⟨100⟩ oriented iron doped, semi-insulating wafers. The surface was uncoated and was polished chemically prior to the implantation. For implantation into the heated InP, the sample was mounted on a copper plate with gallium. This plate holding the sample was heated by a carbon strip heater; a thermocouple was used to monitor the temperature. A dose of 1.8×10^{14} cm^{-2} at 400 keV of energy was used for Se implants. For Be, multiple energy implants of 50 and 150 keV with doses of 3.35×10^{13} and 5.74×10^{13} cm^{-2}, respectively, were used. 120 nm of SiO_2 was sputter deposited as a cap on both surfaces of the samples after implantation and prior to the lamp annealing.

The argon arc lamp used in the annealing system consists of a single quartz tube into which cold water is injected so that it spirals along the inside wall of the tube forming an inner water wall. The argon gas flows down the center of this water-walled tube. As much as 100 kW of power can be applied to the arc, which is 1 cm in diameter and 20 cm long. In addition to directly cooling the arc, the water prevents any material sputtered from the electrodes from depositing on the walls of the quartz tube. The samples are mounted in front of the tube and its reflector system on small quartz pins. The temperature is measured by an optical pyrometer viewing the back of the sample. A more detailed description of the arc lamp annealing system is published elsewhere.[6]

Anneal cycles of 3 and 10-s duration, like those shown in Fig. 1 were used and peak temperatures of 900 and 925 °C, respectively, were obtained as illustrated. The sample temperature is essentially uniform throughout its bulk during the anneal cycle. The rise and fall times of the temperature transient are determined by the sample thickness, whereas the peak temperature is determined by the lamp current, the thickness of the SiO_2 coating, and, in the case of the 3-s anneal where the system does not reach thermal steady state, the sample thickness. No change in either the thickness, etch

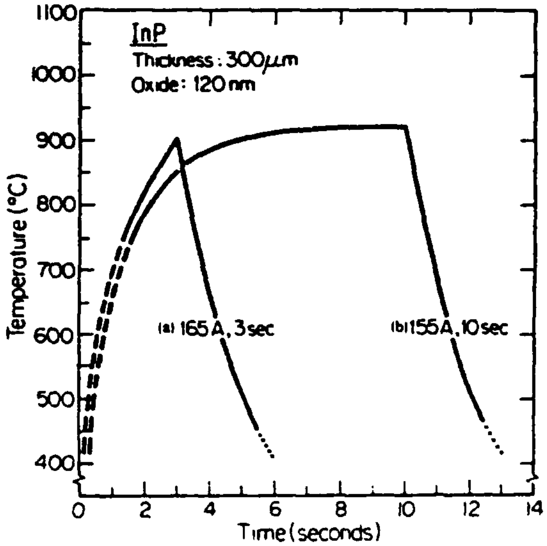


FIG. 1. Time-temperature cycles for the InP samples during the 3- and 10-s arc-lamp anneal sequences. The temperature is the reading of an optical pyrometer viewing the back of the wafer.

TABLE I. Summary of Hall characterization of arc-lamp annealed selenium implanted InP samples.

Selenium	Substrate during implant Cold (RT)	 Hot (200 °C)
Anneal cycle		
3 s 900 °C peak	185 Ω/□ $\bar{\mu}$ = 837 cm²/Vs 22.5%	52 Ω/□ $\bar{\mu}$ = 1373 cm²/Vs 48%
10 s 925 °C	77 Ω/□ $\bar{\mu}$ = 996 cm²/Vs 45%	40 Ω/□ $\bar{\mu}$ = 1415 cm²/Vs 61%

rate, or the refractive index of the SiO_2 film, nor in the appearance of the InP surface could be detected after the anneals.

Two sets of samples, each containing four InP pieces were implanted with Se and Be, respectively, following the schedules mentioned above. Two samples in each set were implanted at room temperature and the other two at 200 °C. Each set was then further divided and half were arc lamp annealed for 3 s and half for 10 s.

After annealing, the oxide was etched off and Van der Pauw patterns[7] were mesa etched on the material using conventional photolithography. For n-type material the contact pads were plated with indium and gold, and for p-type material they were plated with Zn and Au.[8] The contacts were sintered at 420 °C for 45 s in a forming gas atmosphere.

The samples were all characterized using room-temperature Hall measurements. Tables I and II summarize the results in terms of sheet resistance, average sheet mobility, and total percent activation achieved in each case for the Se and Be implants, respectively.

Referring first to the selenium implants (Table I), the hot substrate implants give consistently better results—lower sheet resistance, higher mobilities, and higher activation—than the room-temperature implants, and the 10-s anneal, which reaches the highest temperature at 925 °C is superior to the 3-s anneal. Comparing our lamp annealed results with the best known furnace annealed results[9] for the same dose of Se, we see that the activation for both room temperature and hot substrate implants is comparable, and the mobilities are slightly higher. Multiply scanned electron beam annealed Se implanted InP shows much lower activation (~36%) and mobility[10] than we see with lamp annealing. It has been shown in the literature that heavy ions like Se when implanted into InP or other III-V compounds at room temperature tend to amorphize the material and create much damage. Implanting into a hot substrate helps prevent amorphization and, thus, less damage is created and better electrical properties are achieved.[11] Our results with lamp annealing are consistent with these observations.

TABLE II. Summary of Hall characterization of arc-lamp annealed beryllium implanted InP samples. The hot substrate implants were too low concentration for satisfactory measurement.

Beryllium	Substrate during implant Cold (RT)	 Hot (200 °C)
Anneal cycle		
3 s 900 °C peak	1990 Ω/□ $\bar{\mu}$ = 81 cm²/Vs 42.5%	...
10 s 925 °C peak	1665 Ω/□ $\bar{\mu}$ = 88 cm²/Vs 47%	...

One each of the hot substrate and room-temperature Se implanted samples annealed at 925 °C for 10 s was depth profiled for carrier concentration and mobility using differential Hall measurements. The results are shown in Fig. 2. The peak carrier concentration reaches the same level, $\sim 1\times10^{19}$ cm^{-3}, as predicted by the Lindhard–Scharff–Schiott (LSS) theory, and there is no evidence for any dopant diffusion either into or out of the material. In contrast, similarly implanted, furnace annealed samples do not reach the peak carrier concentration predicted by the LSS theory and show an appreciable amount of in and out diffusion.[9]

The berylium implants into heated substrates showed very low concentration and were not studied further. This behavior appears to be similar to that seen in furnace annealed hot Be implants.[12] The results on the room-temperature Be implants were much better and the activation and mobility are similar to what is obtained from a furnace anneal for similar doses and energies.[12] As with Se, the 10-s anneal to 925 °C was superior to the slightly lower temperature 3-s anneal, but the difference between the two cycles is much less.

Both room-temperature Be implants were profiled us-

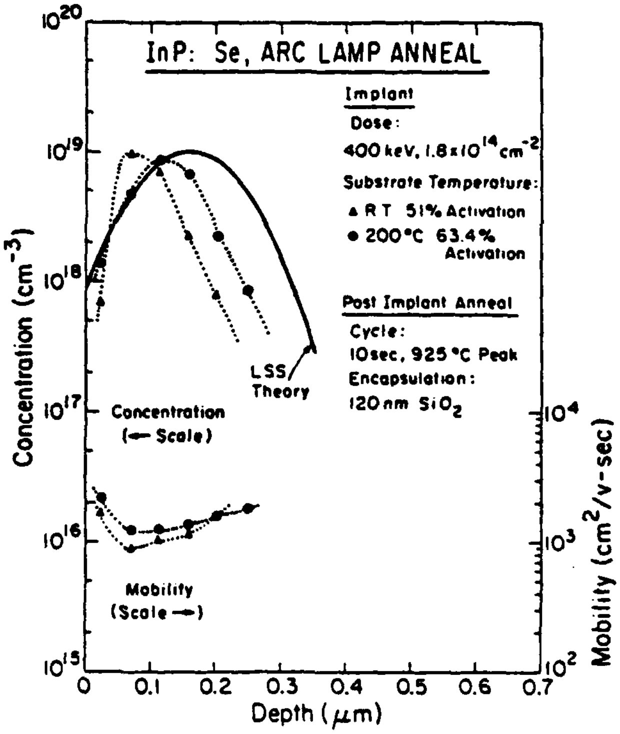


FIG. 2. Carrier concentration and mobility profiles measured by the differential Hall technique on two Se implanted samples; one implanted at room temperature and one at 200 °C, both arc-lamp annealed for 10 s to a peak temperature of 925 °C.

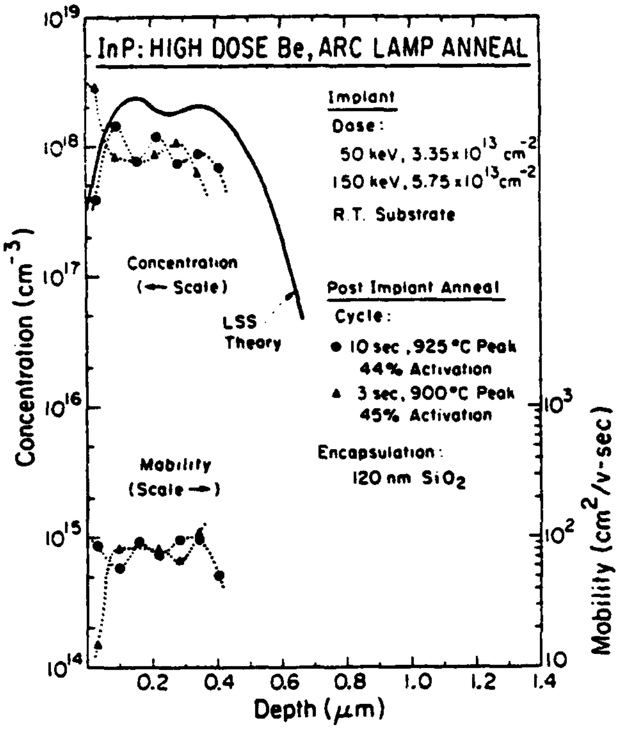


FIG. 3. Carrier concentration and mobility profiles measured by the differential Hall technique on two Be-implanted samples: one arc lamp annealed following the 3-s cycle, the other annealed following the 10-s cycle.

ing differential Hall measurements and the results are illustrated in Fig. 3. For furnace annealed samples appreciable diffusion of Be into the material is observed[12]; no such diffusion is found in the arc lamp annealed samples, as can be seen in Fig. 3. The activations for both annealing cycles are similar, but the mobility is slightly better for the higher temperature cycle.

In conclusion, the activation and carrier mobilities achieved in InP from Be and Se implantation using a high power arc lamp for the post-implant anneal are comparable to and sometimes better than those obtained from 10–15-min, 750 °C furnace anneals. At the same time, sharp profiles showing no dopant diffusion are obtained. Another significant feature of the lamp anneal technique is the relative ease with which the surface can be encapsulated and protected during a post-implant anneal. Compared to furnace annealing techniques, the arc lamp technique is straightforward, fast, and yields superior results. Work is presently in progress extending these studies to other dopants and still higher anneal temperatures.

The excellent assistance of P. Stump of Eaton with the arc lamp anneals is gratefully acknowledged. The authors also wish to thank Professor H. C. Gatos for the use of his laboratories for the Hall measurements. The ion implants were performed by M. Rothman in the Microelectronic Laboratory and Ion Implantation Central Facility of the MIT Center for Materials Science and Engineering, a National Science Foundation Materials Research Laboratory. The SiO_2 was sputter deposited by J. Walsh of the same facility. This work was supported by the Defense Advanced Research Agency of the Department of Defense and monitored by the Office of Naval Research under Contract N00014-C-80-0622.

[1] D. E. Davies, W. D. Potter, and J. P. Lorenzo, J. Electrochem. Soc. **125**, 1845 (1978).
[2] J. P. Donnelly and C. E. Hurwitz, Appl. Phys. Lett. **31**, 418 (1977).
[3] S. S. Gill, P. J. Topham, B. J. Sealy, and K. G. Stephens, J. Phys. D **14**, 2333 (1981).
[4] D. E. Davies, E. F. Kennedy, J. J. Comer, and J. P. Lorenzo, Appl. Phys. Lett. **36**, 922 (1980).
[5] S. S. Gill, B. J. Sealy, P. J. Topham, N. J. Barrett, and K. G. Stephens, Electron. Lett. **17**, 623 (1981).
[6] J. C. Gelpey and P. O. Stump, Proceedings of the First Canadian Semiconductor Technology Conference, Ottawa, 1982, p. 31.
[7] B. Baron, G. A. Shifrin, and O. J. Marsh, J. Appl. Phys. **40**, 3702 (1969).
[8] K. Tabatabaie-Alavi, A. N. M. M. Choudhury, N. J. Slater, and C. G. Fonstad, Appl. Phys. Lett. **40**, 398 (1982).
[9] J. P. Donnelly and C. E. Hurwitz, Solid State Electron. **23**, 943 (1980).
[10] Sukhdev S. Gill, Appl. Phys. Lett. **41**, 729 (1982).
[11] D. Eirug Davies, J. J. Comer, J. P. Lorenzo, and T. G. Ryan, Appl. Phys. Lett. **35**, 192 (1979).
[12] J. P. Donnelly and C. A. Armiento, Appl. Phys. Lett. **34**, 96 (1979).

Electron-beam-induced current measurements in silicon-on-insulator films prepared by zone-melting recrystallization

E. W. Maby,[a] H. A. Atwater,[a] and A. L. Keigler[b]
Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

N. M. Johnson
Xerox Palo Alto Research Center, Palo Alto, California 94304

(Received 5 May 1983; accepted for publication 10 June 1983)

Enhanced diffusion of arsenic along grain boundaries and subboundaries in zone-recrystallized silicon-on-insulator films has been measured by electron-beam-induced current analysis of lateral *pn* junctions fabricated in the films. A four-hour diffusion at 1100 °C resulted in protrusions of arsenic at the junction edges which measured approximately 3–5 μm along the grain boundaries and only 1–2 μm along the subboundaries. The results suggest that under more ordinary thermal processing conditions, field-effect transistors with channel lengths greater than about 1.5 μm can be randomly positioned with respect to the more numerous subboundaries, but grain boundaries should be avoided.

PACS numbers: 73.40.Lq, 61.70.Ng, 66.30.Jt, 81.10.− h

Zone-melting recrystallization by means of a movable-strip heat source can produce continuous device-worthy silicon films over thermally oxidized silicon when a suitable encapsulation layer is present to prevent silicon agglomeration.[1–5] The films consist of large grains, typically 1 mm wide and extending the length of the zone scan, which are seeded from a transition region where solidification begins.[2,5] Individual grains contain subboundaries, typically 25 μm apart, which are generally parallel to one another and to the direction of motion of the molten zone. The subboundaries consist of linear arrays of dislocations with angular deviations of the order of one degree or less; they originate at the interior corner of the solid-liquid interface during recrystallization.[5,6] Large-angle grain boundaries can be eliminated by seeding from the silicon substrate beneath the silicon dioxide layer[7,8] or by appropriate patterning of the silicon film prior to recrystallization.[9] The subboundaries are more difficult to control; however, entrainment techniques which force the subboundaries to be positioned at specific regular intervals have been reported.[6]

In a previous letter,[10] the electrical characteristics of both grain boundaries and subboundaries were examined by fabricating large (92×375 μm), phosphorus-doped (1×10^{17} cm^{-3}) resistors which were either parallel or perpendicular to the line defects. One or more transversely oriented grain boundaries provided significant ($\sim15\%$) bulk conductivity degradation; whereas, an average of 20 subboundaries provided only marginal ($\sim0.15\%$) bulk conductivity degradation. The surface conductivity at the interface between the silicon film which contained the resistors and the silicon dioxide below could be modulated by using the silicon substrate as a gate electrode. Transversely oriented grain boundaries induced peculiar "kinks" in the turn-on characteristics of these "upside-down" depletion-mode field-effect transistors. The influence of subboundaries on surface conductivity could not be detected. These results suggest that grain boundaries must be avoided when considering the placement of field-effect transistors on zone-recrystallized silicon-on-insulator films and that the degradation of electrical properties due to subboundaries can be ignored.

As the dimensions of field-effect transistors fabricated on zone-recrystallized silicon-on-insulator films are reduced, another potential problem arises, i.e., the enhanced diffusion of dopant impurities during high-temperature processing along line defects which form a connective path between source and drain. This phenomenon shortens the effective channel length of the transistor and ultimately leads to an abrupt increase in subthreshold leakage current. Enhanced diffusion of arsenic along grain boundaries in laser-recrystallized silicon-on-insulator films was shown to be significant when channel lengths are less than 3 μm, given a source-drain anneal at 900 °C for 90 min.[11]

In an earlier study,[12] enhanced diffusion of arsenic along grain boundaries in the vicinity of lateral *pn* junctions fabricated in laser-recrystallized silicon films was measured by electron-beam-induced current (EBIC) analysis, and the results were compared with scanning-electron-beam voltage-induced contrast as a check for measurement consistency. A five-hour arsenic diffusion at 1000 °C resulted in protrusions of arsenic at the junction edges extending up to 5 μm along the grain boundaries. The different lengths of the protrusions suggested significant variations in the microstructures or diffusivities of the grain boundaries.

This letter presents the results of EBIC analysis of lateral *pn* junctions fabricated in zone-recrystallized silicon-on-insulator films. In recognition of the electrical and crystallographic differences between grain boundaries and subboundaries, particular effort has been made to compare the enhanced diffusion along the two types of line defect. Earlier work suggested a difference in diffusion characteristics,[13] but the measurements were by a less accurate procedure in which the arsenic-rich regions of the lateral *pn* junctions were preferentially etched. If minimal diffusion along subboundaries could be demonstrated, it would further justify the random placement of devices with respect to the

[a] Department of Electrical Engineering and Computer Science.

[b] Department of Materials Science and Engineering.

subboundaries, thereby simplifying the processing requirements associated with zone-recrystallized silicon-on-insulator films.

Prior to recrystallization, samples were prepared by oxidizing (100) n-type silicon wafers at 1100 °C to obtain a silicon dioxide thickness of 0.5 μm. The thermal oxide was sequentially coated with a 0.5-μm-thick layer of low pressure chemical vapor deposition (LPCVD) polycrystalline silicon, a 2.0-μm-thick layer of CVD silicon dioxide, and a 0.03-μm-thick layer of sputtered silicon nitride. The details of the zone-melting recrystallization procedure have been reported elsewhere.[1-5] Briefly, the sample was placed within an argon ambient on a graphite strip which was resistively heated to 1000 °C. A movable upper strip, approximately 1 mm above the sample and 1 mm wide, was resistively heated until the polycrystalline silicon film melted, and the molten zone was moved across the sample at a speed of approximately 1 mm/s. After recrystallization, the silicon dioxide and silicon nitride encapsulation layers were removed in concentrated hydrofluoric acid. The silicon film was implanted with a fluence of 2×10^{12} cm^{-2} boron at 70 keV. The boron was uniformly redistributed throughout the silicon film at a later point in the process sequence to yield a concentration of 4×10^{16} cm^{-3}.

In order to distinguish the subboundaries from the grain boundaries, the films were decorated by a regular matrix of anisotropically etched pits.[14] These were prepared by depositing a 0.4-μm-thick layer of CVD silicon dioxide over the silicon film and then etching 5-μm-diam holes on 50-μm center spacing. The exposed silicon was etched in a potassium hydroxide solution until square pits revealed the local crystallographic orientations throughout the film. As in other experiments,[1-5] the texture of the film was (100).

The fabrication of lateral pn diodes began with the opening of large (up to 2×2 mm) holes in the silicon dioxide which had been used to define the matrix of etch pits. The holes defined the diode emitter (n^+) regions and were implanted with a fluence of 1×10^{15} cm^{-2} arsenic at 150 keV. The implant was annealed at 1100 °C for 10 min in dry oxygen to form a thin silicon dioxide cap and then for 4 h in dry nitrogen to exacerbate the degree of enhanced diffusion which was expected along the grain boundaries. The silicon dioxide over the diode emitter regions was removed following the anneal. Aluminum was deposited over the sample and then defined to form contact pads which were slightly recessed from the junction edges. The fabrication process concluded by etching the silicon dioxide which remained over the silicon film.

After processing, the samples were mounted on headers and aluminum wires were bonded to the diode n^+ contact pads. There was no direct electrical contact available to the diode p regions (field); the diodes were connected back-to-back in pairs without an applied bias voltage. A JSM-50 (Japanese Electron Optics Limited) scanning electron microscope was used for the EBIC analysis. With a 5-keV electron beam, the EBIC resolution was presumed to be consistent with that established in Ref. 12, approximately 0.5 μm.

Many grain boundaries were located near the transition region of the zone-recrystallized film. A typical EBIC image from this region is shown in Fig. 1. The arsenic protrusions extend approximately 3–5 μm from the edge of the lateral pn junction. A protrusion length is assumed to be roughly equal to a diffusion length $(Dt)^{1/2}$, where D is the coefficient of line

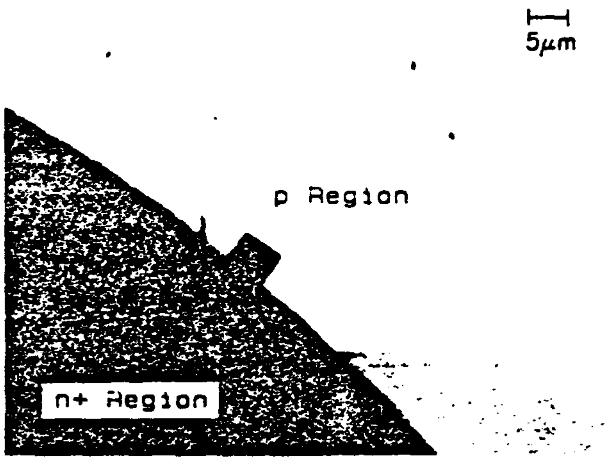


FIG. 1. EBIC image in the vicinity of a lateral pn junction which is intersected by grain boundaries. The large dark square which extends from the diode n^+ region is an anisotropically etched pit used to determine grain orientations in the recrystallized silicon film. A second dark square at the right indicates that a grain boundary is located between the two squares, and the misorientation of the squares indicates that the angle of the grain boundary is approximately 20°. Note the large protrusions where the grain boundaries intersect the junction edge.

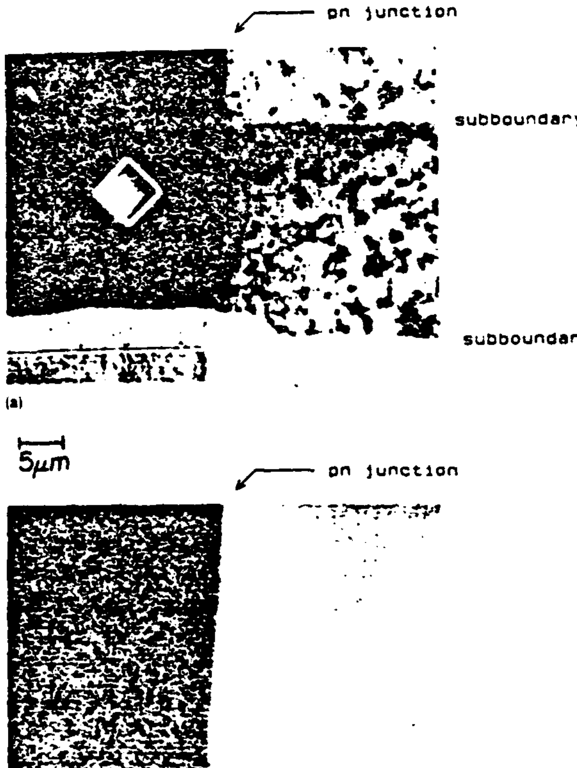


FIG. 2. (a) Secondary electron image in the vicinity of a lateral pn junction which is intersected by subboundaries. The distinct contrast is due to preferential electron channeling. (b) Corresponding EBIC image. Note the small protrusions along the subboundaries.

defect diffusion and t is the diffusion time. From this relation, the coefficient for arsenic diffusion along a grain boundary at 1100 °C is about 1×10^{-11} cm^2/s. The value which was estimated in Ref. 12 for the case of arsenic diffusion at 1000 °C was also about 1×10^{-11} cm^2/s. If the enhanced diffusion which brings about the transistor failure mode of Ref. 11 is assumed to correspond to a diffusion length of 1.5 μm for an arsenic diffusion time of 90 min at 900 °C, the estimated coefficient of grain boundary diffusion for this case is approximately 4×10^{-12} cm^2/s. Since these particular estimates vary as the square of a diffusion length and yet are comparable to within a factor of 2, the degree of enhanced arsenic diffusion along grain boundaries appears to be weakly dependent upon temperature over the range of 900–1100 °C. This conclusion is not consistent with that of earlier work[15] in which the enhanced diffusion of arsenic along grain boundaries in laser-recrystallized silicon-on-insulator films suggested a thermal activation energy of about 2.3 eV.

The EBIC analysis in the vicinity of subboundaries suggests a lesser degree of enhanced arsenic diffusion. Figure 2(a) shows two subboundaries separating regions of distinctly different contrast. The contrast results from preferential electron channeling along certain crystalline orientations.[5] The corresponding EBIC image of Fig. 2(b) shows small protrusions at the subboundary locations which extend approximately 1 μm beyond the lateral *pn* junction edge. The protrusion lengths were sometimes greater along other subboundaries within the sample, but in no case did a protrusion extend more than about 2 μm. For a nominal protrusion length of 1.5 μm, the estimated coefficient of arsenic diffusion along subboundaries has the approximate value of 1×10^{-12} cm^2/s, roughly one order of magnitude less than for the grain boundaries. Under more ordinary processing conditions, the subboundary diffusion length scales downward by a factor of 2 as the diffusion time is reduced to one hour, and some degree of scaling can be expected as the diffusion temperature is reduced. These results suggest that field-effect transistors can be randomly positioned with respect to the subboundaries provided that channel lengths are greater than about 1.5 μm.

In the work at MIT, E. W. Maby was supported by the Defense Advanced Research Projects Agency, contract N0014-C-80-0622, and H. A. Atwater was supported by the Department of Energy, contract DE-AC02-80-ER13019.

[1] E. W. Maby, M. W. Geis, Y. L. Le Coz, D. J. Silversmith, R. W. Mountain, and D. A. Antoniadis, IEEE Electron Device Lett. EDL-2, 241 (1981).

[2] M. W. Geis, H. I. Smith, B-Y. Tsaur, J. C. C. Fan, E. W. Maby, and D. A. Antoniadis, Appl. Phys. Lett. 40, 158 (1982).

[3] R. F. Pinizotto, H. W. Lam, and B. L. Vaandrager, Appl. Phys. Lett. 40, 388 (1982).

[4] T. J. Stultz and J. F. Gibbons, Appl. Phys. Lett. 41, 186 (1982).

[5] M. W. Geis, H. I. Smith, B-Y. Tsaur, J. C. C. Fan, D. J. Silversmith, and R. W. Mountain, J. Electrochem. Soc. 129, 2812 (1982).

[6] M. W. Geis, H I. Smith, D. J. Silversmith, R. W. Mountain, and C. V. Thompson, J. Electrochem. Soc. 130, 1178 (1983).

[7] B-Y. Tsaur, J. C. C. Fan, M. W. Geis, D. J. Silversmith, and R. W. Mountain, Appl. Phys. Lett. 39, 561 (1981).

[8] J. C. C. Fan, B-Y. Tsaur, R. L. Chapman, and M. W. Geis, Appl. Phys. Lett. 41, 186 (1982).

[9] H. A. Atwater, H. I. Smith, and M. W. Geis, Appl. Phys. Lett. 41, 747 (1982).

[10] E. W. Maby and D. A. Antoniadis, Appl. Phys. Lett. 40, 691 (1982).

[11] K. K. Ng, G. K. Celler, E. I. Povilonis, R. C. Frye, H. J. Leamy, and S. M. Sze, IEEE Electron Device Lett. EDL-2, 316 (1981).

[12] N. M. Johnson, D. K. Biegelsen, and M. D. Moyer, Appl. Phys. Lett. 38, 900 (1981).

[13] H. W. Lam, R. F. Pinizzotto, S. D. S. Malhi, and B. L. Vaandrager, Appl. Phys. Lett. 41, 1083 (1982).

[14] K. A. Bezjian, H. I. Smith, J. M. Carter, and M. W. Geis, J. J. Electrochem. Soc. 129, 1848 (1982).

[15] H. Baumgart, H. J. Leamy, L. E. Trimble, C. J. Doherty, and G. K. Celler, MRS Symposium on *Grain Boundaries in Semiconductors*, edited by H. J. Leamy, G. E. Pike, and C. H. Seager (North-Holland, New York, 1982), p. 311.

Rapid thermal annealing of Be, Si, and Zn implanted GaAs using an ultrahigh power argon arc lamp

K. Tabatabaie-Alavi, A. N. M. Masum Choudhury, and C. G. Fonstad
Department of Electrical Engineering and Computer Science, and Center for Materials Science and Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

J. C. Gelpey
Eaton Ion Beam Systems Division, Beverly, Massachusetts 01915

(Received 19 April 1983; accepted for publication 1 June 1983)

The use of a 100-kW water-walled dc argon lamp to anneal ion-implanted GaAs is reported. Annealing cycles of 3 and 10 s and peak temperatures from 950 to 1200 °C have been used to anneal Be, Si, and Zn implanted following representative implant schedules of technological importance. It is demonstrated that this technique is superior to conventional furnace anneal techniques in terms of the doping profiles, peak carrier concentrations, activation efficiencies (particularly at high doses), and mobilities achieved. The annealing technique should be applicable to large volume GaAs integrated circuit production and 100-mm-diam wafers can be annealed in a single exposure with better than 2% temperature uniformity (Si data).

PACS numbers: 81.40.Ef, 61.70.Tm, 72.80.Ey, 81.40.Rs

The use of rapid thermal annealing (RTA) techniques to anneal ion implanted GaAs, and related III–V compound semiconductors, promises to have a significant impact on the device technology of these materials because RTA should permit higher anneal temperatures with improved dopant activation and carrier mobilities, yield sharper junctions, and result in higher doping levels than can be achieved by furnace anneal techniques while simultaneously reducing capping requirements, and thereby significantly simplifying wafer processing. In this letter, we demonstrate that these objectives can be achieved when a high power, water-walled dc arc lamp, capable of processing 125-mm-diam wafers, is used to rapidly anneal GaAs.

Several RTA techniques have been applied to GaAs with good initial success, but each also has important limitations. Arai *et al.*[1] first reported using halogen lamps to anneal GaAs. The GaAs was placed face down on a Si wafer and annealed 5 s with a peak temperature of 950 °C. Nearly 100% activation of a 3×10^{12} cm^{-2}, 70-keV Si implant was achieved. More recently, Kuzuhara *et al.*[2] used the same technique to anneal a higher dose Si implant (5×10^{12} cm^{-2}, 100 keV) and obtained 75% activation and a sheet mobility of 3700-cm^2/V after a 2 s, 950 °C peak temperature anneal. They observed arsenic loss and the formation of gallium pits on the wafer surface if they used a 5-s anneal, however. The maximum rate of temperature increase in these experiments was 200 °C/min.

Davies *et al.*[3] have used filament lamps focused by elliptical mirrors to anneal GaAs implanted with zinc and silicon to 1000 °C. They achieved essentially 100% activation of a 2×10^{14} cm^{-2}, 200-keV Zn implant and 50% activation of a 4×10^{14} cm^{-2}, 200-keV Si implant. The system they used could heat the sample to 1000 °C in one second, but had a long fall time because of the nearly lossless optical cavity used.

Chapman *et al.*[4] have used a graphite strip heater to anneal GaAs to 1140 °C for 10 s. They obtained 18% activation of a 1×10^{15} cm^{-2}, 400-keV Se implant done into a hot (300 °C) substrate. They observed significant diffusion of Si into the GaAs from the Si_3N_4 cap used. With the graphite heater strip, the temperature rise was relatively slow.

As presently configured, the sources that have been used for the RTA of GaAs in the above studies, have important limitations. The problems of the rise and fall times of the heating cycles have already been mentioned. The latter two systems also were line sources and, thus, are unsuitable for annealing large area samples because the large thermal gradients created in samples as the line is scanned across them can cause the formation of slip planes. The graphite strip heater further has the problem that it must be operated in a vacuum and the possibility of contamination from the porous graphite also exists.

The system used in the present work overcomes these difficulties, and as well, achieves superior results to previously reported furnace and RTA anneal procedures.

The arc lamp annealing system used has been described previously by Gelpey and Stump.[5] Briefly, the light source is a 100-kW water-walled dc argon arc lamp which has a time constant of a few milliseconds and a color temperature of 5500 K, and produces a uniform power flux density of 80 W/cm^2 at full intensity (400-A lamp current) over the sample surface. Temperature uniformities of 2% have been measured over 100-mm-diam Si wafers.

The GaAs used in this study was semi-insulating undoped material. The ion implantation sequences (species, energy, and dose) are listed in columns 1–3 of Table I. All implants were performed into bare surfaces with the samples nominally at room temperature. After implantation and prior to annealing a 120-nm-thick layer of SiO_2 was sputter deposited on both surfaces of the wafer to serve as the encapsulation. During the anneal, the samples were held in small slotted quartz pins so that they are thermally (conductively) isolated form their surroundings. The sample temperature was monitored using an optical pyrometer viewing the back of the GaAs wafer. The temperature measurement in the present work is thought to be accurate to ± 20 °C.

TABLE I. Implant schedules and annealing cycles used in this study, and the results of Hall measurements of the surface layers including the sheet carrier concentration n_s, sheet Hall mobility μ_s, and sheet resistance R_s. All samples were undoped semi-insulating GaAs wafers.

				Anneal cycle				
Species	Energy (keV)	Dose (cm^{-2})	Thickness (μm)	Time (s)	Pk. temp. (°C)	n_s (cm^{-2})	μ_s (cm^2/Vs)	R_s ($\Omega/\Box$)
Zn	200	1.4×10^{15}	415	10	950	7.2×10^{14}	62	140
Zn	200	1.4×10^{15}	415	10	1020	6.8×10^{14}	70	130
Zn	200	1.4×10^{15}	415	3	1160	1.3×10^{15}	62	80
Si	200	4×10^{14}	415	10	950		no activation	
Si	200	4×10^{14}	415	10	1020	1.8×10^{13}	1850	90
Si	200	4×10^{14}	415	10	1100	1.15×10^{14}	1600	33
Si	200	4×10^{14}	415	3	1160	1.25×10^{14}	1500	35
Be	50, 150	4.4×10^{14}, 5.1×10^{14}	415	10	950	5.4×10^{14}	92	125
Be	50, 150	4.4×10^{14}, 5.1×10^{14}	415	10	1020	5.8×10^{14}	67	160
Be	50, 150	4.4×10^{14}, 5.1×10^{14}	415	10	1100	6.5×10^{14}	33	300
Be	50, 150	4.4×10^{14}, 5.1×10^{14}	415	3	1160	6.2×10^{14}	65	160
None	None	None	365	None	None	2.1×10^{6}	2300	1.3×10^{9}
None	None	None	365	10	1050	4.9×10^{6}	612	2.1×10^{9}
None	None	None	365	10	1100	2.3×10^{8}	75	3.8×10^{8}
None	None	None	365	3	1200	2×10^{8}	36	9.1×10^{8}
None	None	None	525	None	None	3.15×10^{5}	5500	3.6×10^{9}
None	None	None	525	10	1050	3.4×10^{7}	202	9.1×10^{8}
None	None	None	525	10	1090	2×10^{9}	10	3×10^{8}
None	None	None	525	3	1100	8.9×10^{7}	135	5.2×10^{8}
None	None	None		Furnace—see text		1×10^{8}	150	4×10^{8}

The time-temperature cycles used were similar to those recently used to anneal InP.[6] No change could be observed in the surface morphology of the GaAs after any of the anneal cycles used in the results reported in this study.

After the anneal, the SiO_2 was removed from the surface of the samples, they were patterned with a standard van der Pauw pattern, and their sheet resistivity, carrier concentration, and mobility were measured. The anneal cycle used and the results of this initial characterization are presented in columns 4–7 of Table I.

Several of the samples were further profiled using differential Hall measurements.[7] These profiles are presented in Figs. 1–3.

Zn implant. A peak hole concentration $\approx 9\times10^{19}$ cm^{-3} (see Fig. 1), sheet mobility of 62 cm^2/Vs, sheet resistance of 80 $\Omega/\Box$, and 93% electrical activation are obtained after a 3-s anneal to 1160 °C. To the best of our knowledge, this represents the highest doping level and electrical activation that has been reported for zinc implantation into GaAs (liquid-phase-epitaxial-regrowth annealing techniques are excluded). The mobility (Fig. 1) compares favorably with mobility data for zinc diffusion in GaAs at the same doping level.[8,9] The zinc concentration profile is consistent with the generally accepted interstitial-substitutional zinc diffusion theory in GaAs (Ref. 18), although the extent of diffusion is more than expected for such a short annealing time. In a related experiment, no enhanced lateral diffusion of zinc, implanted through photoresist into 10-μm-wide stripes, could be detected after a similar arc lamp anneal in spite of the presence of the SiO_2 capping layer.

Be implant. All cycles yielded similar results but the best results are obtained for the lower temperature anneal cycle 950 °C. A maximum hole concentration $\approx 2\times10^{19}$ cm^{-3}, and an electrical activation of 72% is obtained (see Fig. 2). In a related experiment, the junction depth in vapor phase epitaxially (VPE) grown GaAs ($N_D - N_A \approx 8\times10^{14}$ cm^{-3}), which had been identically Be implanted and annealed, was found to be 1–1.1 μm. These results are superior to the best of published furnace anneal data[10,11] in several aspects. In furnace annealing, which is usually done at 800–900 °C for 10–15 min, significant damage enhanced diffusion of Be occurs. This causes junction misplacement and limits the maximum Be concentration to the mid 10^{18} cm^{-3} range with $\approx$40% electrical activation at a 10^{15} cm^{-2} dose level. No Be diffusion takes place upon rapid thermal annealing

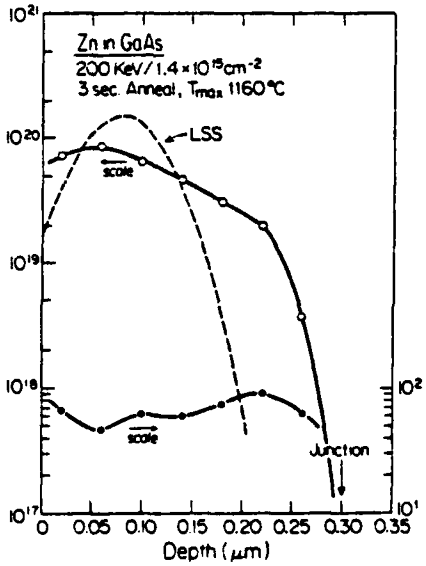


FIG. 1. Carrier concentration and mobility profiles measured by the differential Hall technique on arc lamp annealed zinc implanted (1.4×10^{15} cm^{-2}, 200 keV) GaAs samples annealed for 3 s to 1160 °C.

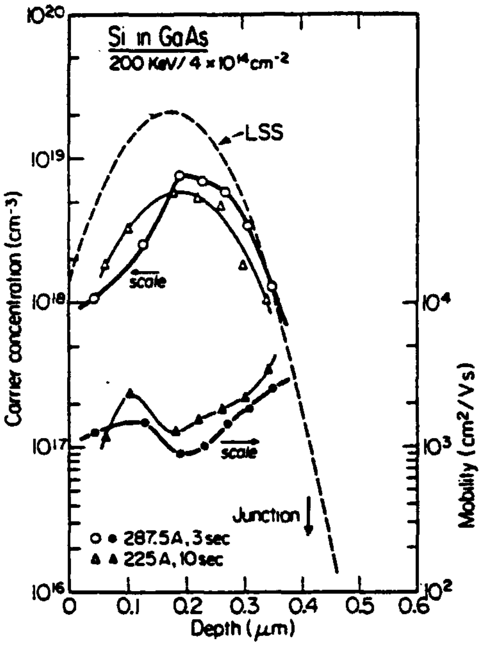


FIG. 2. Carrier concentration and mobility profiles measured on two silicon implanted (4×10^{14} cm^{-2}, 200 keV) GaAs samples: one annealed 3 s to 1160 °C, the other annealed 10 s to 1100 °C.

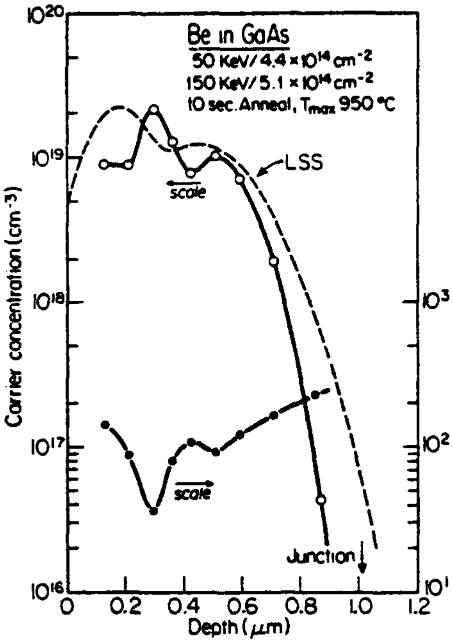


FIG. 3. Carrier concentration and mobility profiles measured on beryllium implanted (4.4×10^{14} cm^{-2}, 50 keV and 5.1×10^{14} cm^{-2}, 150 keV) GaAs samples annealed for 10 s to 950 °C.

and a high peak concentration is maintained. It is also found that mesa diodes fabricated in this VPE material are superior to the best published Be implanted furnace annealed diodes. These electrical results will be reported in a succeeding paper.

Si implant. Both 10-s and 3-s anneals at maximum temperatures of 1100 and 1160 °C, respectively, give similar results. A maximum carrier concentration of 7.5×10^{18} cm^{-3}, sheet resistance of 35 Ω/□, and 33% electrical activation are obtained (see Fig. 3). At the same implant energy and dose, Davies *et al.*[3] have reported 50% electrical activation and a sheet resistance of 23.9 Ω/□ after a 2.5-s anneal with a maximum temperature of 1000 °C. However, the temperature measurement in Ref. 3 may not be accurate because of the slow response time of the thermocouple, and we believe that the 50% electrical activation reported was obtained at temperatures much higher than 1000 °C.

Semi-insulating material: As is shown in Table I, the sheet resistance of unimplanted semi-insulating undoped GaAs after arc lamp annealing is either comparable to or higher than that obtained after 15 min of face to face proximity cap annealing (typical results of many runs) under argon atmosphere at 850 °C.[12] Results for furnace annealing under arsenic over pressure are similar to the proximity cap annealing. To our knowledge, this is the first quantitative report on the effect of rapid thermal annealing on the sheet resistance of semi-insulating undoped GaAs. If we use the sheet mobility as an indication of surface quality, the 1050 °C anneals are superior to furance annealing while the 1100 °C anneal is comparable. No surface conversion to *p*-type was observed even in anneals as high as 1200 °C.

In conclusion, it has been demonstrated that an ultrahigh intensity arc lamp can be used to rapidly thermal anneal ion-implanted GaAs with superior results in terms of activation efficiency, peak doping level, and sharpness of doping profile both laterally and vertically. The applicability of this system to large area wafers, the rapid response time of the lamp system, and the ease with which the sample surface can be encapsulated during processing are all attractive features and important advantages of this technique.

The authors acknowledge the help of Dr. David Bliss from Microwave Associates for the Hall measurements on semi-insulating GaAs. The assistance of M. Rothman in performing the ion implantation and J. Walsh for the SiO_2 deposition is also acknowledged. Extensive use was made of the MIT Center for Materials Science and Engineering, a National Science Foundation Materials Research Laboratory, during the course of this work. This work was supported by the National Science Foundation through Grant No. DMR 78-24185 and by the Defense Advanced Research Projects Agency and monitored by the Office of Naval Research under contract No. N00014-C-80-0622.

[1] Michio Arai, Kazuo Nishiyame, and Nanzo Watanbe, Jpn. J. Appl. Phys. 20, L124 (1981).
[2] Massaki Kuzuhara, Hideaki Kohzu, and Yoichiro Takayama, Appl. Phys. Lett. 41, 755 (1982).
[3] D. E. Davies, P. J. McNally, J. P. Lorenzo, and M. Julian, IEEE Electron. Devices Lett. EDL-3, 102 (1982).
[4] R. J. Chapman, J. C. C. Fan, J. P. Donnelly, and B-Y. Tsaur, Appl. Phys. Lett. 40, 805 (1982).
[5] J. C. Gelpey and P. O. Stump, in Proceedings of the Canadian Semiconductor Technology Conference, Ottawa, Aug. 1982, pp. 31–36.
[6] A. N. M. Masum Choudhury, K. Tabatabaie-Alavi, C. G. Fonstad, and J. C. Gelpy, Appl. Phys. Lett. 43, 381 (1983).
[7] B. Baron, G. A. Shifrin, and O. J. Marsh, J. Appl. Phys. 40, 3702 (1969).
[8] R. Jett Field and Sorab K. Ghandhi, J. Electrochem. Soc. 129, 1567 (1982).
[9] Yoshihisa Yamamoto and Hiroshi Kanbe, Jpn. J. Appl. Phys. 19, 121 (1982).
[10] J. P. Donnelly, F. J. Leonber, and C. O. Bozler, Appl. Phys. Lett. 28, 706 (1976).
[11] W. V. McLevige, M. J. Helix, K. V. Vaidyanathan, and B. G. Streetman, J. Appl. Phys. 48, 3342 (1977).
[12] David Bliss (private communication).

September, 1983

LIDS-P-1331

Nonlinear Dynamic Maximum Power Theorem, with Numerical Method

John L. Wyatt, Jr.

ABSTRACT

This paper considers the problem of maximizing the energy or average power transfer from a nonlinear dynamic n-port source. The main theorem includes as special cases the standard linear result $\underset{\sim}{Y}_{load} = \underset{\sim}{Y}^*_{source}$ and a recent finding for nonlinear resistive networks. An operator equation for the optimal output voltage $\hat{v}(\cdot)$ is derived, and a numerical method for solving it is given.

This research was supported by the National Science Foundation under Grant No. ECS 8006878, the Air Force Office of Sponsored Research under Contract #F29620-81-C-0054 and the Defense Advanced Research Projects Agency under contract #N00014-80-C-0622.

The author is with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139.

I. Introduction

This paper addresses the problem of extracting the maximum energy or average power from a source with the topology shown in Fig. 1. As in[1] [1], the problem is formulated as finding the optimal output voltage $\underset{\sim}{\hat{v}}(\cdot)$ for each current source waveform $\underset{\sim}{i}_s(\cdot)$ rather than finding a load that maximizes the power.

The central result is the operator equation (6) for $\underset{\sim}{\hat{v}}(\cdot)$. Theorem 1 gives conditions that guarantee uniqueness and global optimality of the solution: the standard result for linear systems [1] and recent work on resistive nonlinear systems [2] follow as special cases. Equation (11) defines a practical algorithm for solving (6), and Theorem 2 gives conditions that guarantee convergence.

The solution $\underset{\sim}{\hat{v}}(\cdot)$ can be of engineering value in two ways. First, the average power $\overline{P}(\underset{\sim}{\hat{v}})$ tells us the optimal performance that is possible in principle. Second, $\underset{\sim}{\hat{v}}(\cdot)$ itself is a concrete design goal. If the source admittance operator $\underset{\sim}{F}$ is continuous, a load for which the output approximates $\underset{\sim}{\hat{v}}(\cdot)$ (in the Hilbert space norm used in this work) will absorb an average power that approximates $\overline{P}(\underset{\sim}{\hat{v}})$.

[1] Reference,[1] actually deals with the dual network, where the source appears in Thevenin form.

II. Results

2.1) Notation and Definitions

Let L be any real inner product space and $\hat{L}$ any linear subspace of L. An operator F: $\hat{L} \rightarrow L$ is said to be

a) strictly increasing if

$$\langle F(y) - F(x), y-x \rangle > 0, \forall x \neq y \in \hat{L}, \quad (1)$$

b) uniformly increasing if for some $\delta > 0$,

$$\langle F(y) - F(x), y-x \rangle \geq \delta \| y-x \|^2, \forall x,y \in \hat{L}, \quad (2)$$

c) Lipschitz continuous if for some $K \geq 0$,

$$\| F(y) - F(x) \| \leq K \| y-x \| , \forall x,y \in \hat{L}. \quad (3)$$

Let L, L' be any real inner product spaces and $\mathcal{L}(L,L')$ denote t.e space of continuous linear maps from L to L', with the operator norm [3, p.53]. For $A \in \mathcal{L}(L,L')$, let A^{adj} denote the adjoint of A.

Given an operator F: L → L' and $x,h \in L$, suppose there exists an element denoted $\delta F(x,h)$ of L' such that

$$\lim_{t \to 0^+} \left\| \frac{F(x+th) - F(x)}{t} - \delta F(x,h) \right\|_{L'} = 0.$$

Then $\delta F(x,h)$ is called the Gâteaux variation of F at x for the increment h [4,p.251]. If $\delta F(x,h)$ exists for all $x,h \in L$, and if for each $x \in L$ the map $h \rightarrow \delta F(x,h)$ is an element of $\mathcal{L}(L,L')$, then F is said to be Gâteaux differentiable on L. In this case the map $x \rightarrow \delta F(x,\cdot)$ is called the Gâteaux derivative of F and denoted DF: $L \rightarrow \mathcal{L}(L,L')$ [4,pp.255-256]. Similarly

$\delta F(x,\cdot)$ is denoted $DF(x) \in L(L,L')$, and $\delta F(x,h)$ is denoted $(DF(x))h \in L'$. The value of using the Gâteaux derivative rather than the more restrictive Fréchet derivative [4,Chap.3] will become apparent in section 3.1.

The Hilbert space L_n^2 is the set of all measurable functions $x\colon \mathbb{R} \to \mathbb{R}^n$ such that the integral of $x_j^2(\cdot)$ over $\mathbb{R}$ is finite, $j=1,\ldots,n$, equipped with the usual inner product $\langle\cdot,\cdot\rangle$ and norm, $\|x\| \triangleq \langle x,x\rangle^{1/2}$.

For each $T > 0$, $L_{n,T}^2$ is the set of all periodic measurable functions $x\colon \mathbb{R} \to \mathbb{R}^n$ with period T such that the integral over one period of $x_j^2(\cdot)$ is finite, $j=1,\ldots,n$. It is a Hilbert space with the "average power" inner product

$$\langle x,y\rangle_T \triangleq \frac{1}{T}\int_0^T x(t)\cdot y(t)\,dt, \tag{4}$$

where $x\cdot y$ is the Euclidean inner product on $\mathbb{R}^n$. The norm on $L_{n,T}^2$ is denoted $\|x\|_T \triangleq \langle x,x\rangle_T^{1/2}$.

2.2) Main Theorem

Theorem 1 (Maximum Average Power in the Periodic Steady State)

Fix $T > 0$ and let the n-port N_1 in Fig. 1 be characterized by an admittance operator $F\colon L_T \to L_T$, where[2] L_T is any linear subspace of $L_{n,T}^2$. Suppose F is Gâteaux differentiable on L_T and that the associated operator $H\colon L_T \to L_T$, given by

$$H\colon v \mapsto F(v) + (DF_{(v)})^{\text{adj}}\, v \quad , \tag{5}$$

[2] Thus, if $v(\cdot)$ has period T and lies in $L_{n,T}^2$, the response $i(\cdot)$ of N_1 cannot have subharmonics.

is strictly increasing.

Then for each $i_s \varepsilon H(L_T)$ there is a unique solution $\hat{v}(i_s) \varepsilon L_T$ to

$$i_s = H(v), \tag{6}$$

and the average power[3] absorbed by the load,

$$\overline{P}(v) \triangleq \langle i_o, v \rangle_T = \langle i_s - F(v), v \rangle_T , \tag{7}$$

has a unique global maximum over L_T, which is attained at $v = \hat{v}(i_s)$.

Corollary (Maximum Total Energy for Transients)

Let L be a linear subspace of L^2_n, and substitute L for L_T in the assumptions of Theorem 1. Then the same conclusions[4] hold, but with $\hat{v}(i_s) \varepsilon L$ maximizing the total energy $E(v) \triangleq \langle i_s - F(v), v \rangle$ over L.

Note that in general F can be nonlinear and time-varying.

In applications one might wish to restrict attention to currents and voltages in $L^2_{n,T}$ with additional properties such as continuity or boundedness. This is the reason for introducing $L_T \subset L^2_{n,T}$ in the formulation of Theorem 1.

The essential idea behind the theorem is that a solution $\hat{v}(\cdot)$ of (6) is a stationary point of $\overline{P}: L_T \rightarrow \mathbb{R}$, and the monotonicity assumption on H guarantees that $\overline{P}$ is strictly concave. Details follow.

[3] A more explicit, but cumbersome, notation would be $\overline{P}(v, i_s)$. Using it, Theorem 1 states that $\forall v, i_s \varepsilon L_T$, $\overline{P}(v, i_s) < \overline{P}(\hat{v}(i_s), i_s)$ if $v \neq \hat{v}(i_s)$.

[4] For the Corollary, the adjoint is of course taken with respect to the inner product on L^2_n rather than $\langle\ ,\ \rangle_T$.

Proof of Theorem 1

Uniqueness of the solution to (6) follows from the fact that $\underset{\sim}{H}$ is strictly increasing. By the chain rule for the composition of Frechet- and Gâteaux-differentiable functions [4,p.253] $\bar{P}$ is Gâteaux differentiable and for all $\underset{\sim}{x},\underset{\sim}{h}\in L_T$,

$$(D\bar{P}(\underset{\sim}{x}))\underset{\sim}{h} = \left\langle \underset{\sim}{i}_s - \underset{\sim}{F}(\underset{\sim}{x}),\underset{\sim}{h}\right\rangle_T - \left\langle (D\underset{\sim}{F}(\underset{\sim}{x}))\underset{\sim}{h},\underset{\sim}{x}\right\rangle_T =$$
$$\left\langle \underset{\sim}{i}_s - \underset{\sim}{F}(\underset{\sim}{x}) - (D\underset{\sim}{F}(\underset{\sim}{x}))^{adj}\ \underset{\sim}{x},\underset{\sim}{h}\right\rangle_T =$$
$$\left\langle \underset{\sim}{i}_s - \underset{\sim}{H}(\underset{\sim}{x}),\underset{\sim}{h}\right\rangle_T. \tag{8}$$

Thus if $\underset{\sim}{i}_s\in\underset{\sim}{H}(L_T)$,

a) $D\bar{P}(\hat{\underset{\sim}{v}}(\underset{\sim}{i}_s)) = \underset{\sim}{0}\in \mathcal{L}(L_T,L_T)$,

b) given any $\underset{\sim}{x},\underset{\sim}{y}\in L_T$, the map $\lambda \to \bar{P}[\underset{\sim}{x}+\lambda(\underset{\sim}{y}-\underset{\sim}{x})]$ is differentiable at each $\lambda\in\mathbb{R}$, and

c) $\frac{d}{d\lambda}\bar{P}[\underset{\sim}{x}+\lambda(\underset{\sim}{y}-\underset{\sim}{x})] = \left\langle \underset{\sim}{i}_s - \underset{\sim}{H}[\underset{\sim}{x}+\lambda(\underset{\sim}{y}-\underset{\sim}{x})],\ (\underset{\sim}{y}-\underset{\sim}{x})\right\rangle_T$.

To show that $\hat{\underset{\sim}{v}}(\underset{\sim}{i}_s)$ globally optimizes $\bar{P}$, fix $\underset{\sim}{i}_s\in\underset{\sim}{H}(L_T)$, let $\hat{\underset{\sim}{v}} = \hat{\underset{\sim}{v}}(\underset{\sim}{i}_s)$, and choose any $\underset{\sim}{v}\in L_T$, $\underset{\sim}{v}\neq\hat{\underset{\sim}{v}}$. Then

$$\bar{P}(\underset{\sim}{v}) - \bar{P}(\hat{\underset{\sim}{v}}) =$$
$$\bar{P}[\hat{\underset{\sim}{v}}+\lambda(\underset{\sim}{v}-\hat{\underset{\sim}{v}})]\Big|_{\lambda=1} - \bar{P}[\hat{\underset{\sim}{v}}+\lambda(\underset{\sim}{v}-\hat{\underset{\sim}{v}})]\Big|_{\lambda=0} =$$
$$\int_0^1 \{\frac{d}{d\lambda}\ \bar{P}[\hat{\underset{\sim}{v}}+\lambda(\underset{\sim}{v}-\hat{\underset{\sim}{v}})]\}\ d\lambda. \tag{9}$$

Using c), the integrand above is

$$\left\langle \underline{i}_s - \underline{H}[\hat{\underline{v}}+\lambda(\underline{v}-\hat{\underline{v}})],\ \underline{v}-\hat{\underline{v}}\right\rangle_T =$$

(since $\underline{i}_s = \underline{H}(\hat{\underline{v}})$)

$$\frac{-1}{\lambda}\left\langle \underline{H}[\hat{\underline{v}}+\lambda(\underline{v}-\hat{\underline{v}})] - \underline{H}(\hat{\underline{v}}),\ [\hat{\underline{v}}+\lambda(\underline{v}-\hat{\underline{v}})] - [\hat{\underline{v}}]\right\rangle_T,\ \forall\lambda>0,$$

and the integrand vanishes at $\lambda=0$. The inner product above is strictly positive for $\lambda\neq 0$ since $\underline{H}$ is strictly increasing by assumption. Thus the integrand in (9) is negative for $\lambda>0$ and zero for $\lambda=0$, so $\overline{P}(\underline{v}) < \overline{P}(\hat{\underline{v}})$ as claimed. ■

The proof of the Corollary is essentially identical and will be omitted.

2.3) Relation to "Impedance Matching" Ideas

The emphasis in this paper is on finding the optimal output voltage $\hat{\underline{v}}(\cdot)$, not the optimal load. But the relation to impedance matching ideas deserves comment.

If the load in Fig. 1 is taken to be the (generally noncausal) admittance $\underline{G}_{opt}: L_T \to L_T$, defined by

$$\underline{G}_{opt}: \underline{v} \mapsto (D\underline{F}(\underline{v}))^{adj}\,\underline{v}\ , \qquad (10)$$

then the network is uniquely solvable given any $\underline{i}_s \in H(L_T)$, and the output voltage $\underline{v}(\cdot)$, which necessarily equals $\hat{\underline{v}}(\underline{i}_s)$, globally optimizes $\overline{P}$. This generally noncausal load is "matched" to the source for <u>all</u> inputs $\underline{i}_s \epsilon \underline{H}(L_T)$, and this result holds generally for a nonlinear, time-varying, even noncausal source admittance $\underline{F}$. The reader can easily verify that in the LTI case (10) reduces to the standard linear theorem $\underline{Y}_{load}(j\omega) = \underline{Y}^*_{source}(j\omega)$. More detail for the linear 1-port case is given in Section 3.1.

Of course in practice one has a causal load, usually predetermined, and wishes to couple it to the source through a lossless matching network designed to maximize the absorbed power over a range of inputs. In the linear case this important problem is called "broadband matching" [5-8]. We note that in both the linear and nonlinear cases the problem can be viewed as compensating or coupling to a predetermined load using lossless elements in such a way that the response approximates that of the noncausal exact match $\mathbf{G}_{opt}$ over the input range of interest.

For a particular drive $\mathbf{i}_s$, the situation is somewhat different. The optimal voltage $\hat{\mathbf{v}}(\cdot)$ is unique, but the optimal load is not: the only requirement on $\mathbf{G}$ is that $\mathbf{G}(\hat{\mathbf{v}}) = \mathbf{G}_{opt}(\hat{\mathbf{v}})$. In the linear case where $\mathbf{F}$ and $\mathbf{G}$ are respectively represented by admittance matrices $\mathbf{Y}_S(j\omega)$ and $\mathbf{Y}_L(j\omega)$, there are in general infinitely many optimal, positive semidefinite choices of $\mathbf{Y}_L$ at a given ω for which the network is uniquely solvable [9]. The problem of finding solutions in particular classes, such as the class of resistive loads, is studied in [10].

2.4) Numerical Algorithm

Equation (8) shows that $\mathbf{i}_s - \mathbf{H}(\mathbf{v})$ is the gradient [3,p.72], [4,p.196] of $\overline{P}$ at $\mathbf{v}$, $\forall\ \mathbf{i}_s, \mathbf{v} \in L_T$. This suggests that we attempt to maximize $\overline{P}$ by a simple "hill-climbing" algorithm of the form

$$\mathbf{x}_{j+1} = \lambda(\mathbf{i}_s - \mathbf{H}(\mathbf{x}_j)) + \mathbf{x}_j \overset{\Delta}{=} \mathbf{M}(\mathbf{x}_j) \tag{11}$$

for some $\lambda>0$. Note that under the assumptions of Theorem 1, if $\mathbf{x}_j \to \mathbf{x} \in L_T$ and $\mathbf{H}$ is continuous, then $\mathbf{i}_s = \mathbf{H}(\mathbf{x})$ and $\mathbf{x}$ globally maximizes $\overline{P}$. By tightening the assumptions a little further, we can guarantee convergence for all sufficiently small positive λ.

Theorem 2

Strengthen the assumptions of Theorem 1 by supposing further that L_T is closed and $\underset{\sim}{H}$ is uniformly increasing and Lipschitz continuous on L_T. (See (2), (3).) Then for any $\underset{\sim}{i}_s \varepsilon L_T$, any initial guess $\underset{\sim}{x}_o \varepsilon L_T$, and any $\lambda \varepsilon (0, 2\delta/K^2)$, the sequence generated by (11) converges to $\hat{\underset{\sim}{v}}(\underset{\sim}{i}_s)$.

Remark

Note that Theorem 2 also guarantees existence of a solution to (6) for all $\underset{\sim}{i}_s \; \varepsilon \; L_T$,i.e., $\underset{\sim}{H}(L_T) = L_T$.

Proof

Since $L_T \subset L^2_{n,T}$ is closed and $L^2_{n,T}$ is complete, L_T is complete [11]. It remains to show that $\underset{\sim}{M}$ is contractive, i.e., that for some $C < 1$,

$$\| \underset{\sim}{M}(\underset{\sim}{y}) - \underset{\sim}{M}(\underset{\sim}{x}) \|_T \leq C \; \| \underset{\sim}{y} - \underset{\sim}{x} \|_T, \; \forall \underset{\sim}{x}, \underset{\sim}{y} \; \varepsilon \; L_T, \tag{12}$$

to guarantee $\| \underset{\sim}{x}_n - \hat{\underset{\sim}{v}}(\underset{\sim}{i}_s) \|_T \rightarrow 0$ by the contraction mapping theorem [3,p.102], [12,p.28]. But

$$\| \underset{\sim}{M}(\underset{\sim}{y}) - \underset{\sim}{M}(\underset{\sim}{x}) \|^2_T =$$

$$\langle \underset{\sim}{y} - \underset{\sim}{x} - \lambda(\underset{\sim}{H}(\underset{\sim}{y}) - \underset{\sim}{H}(\underset{\sim}{x})), \underset{\sim}{y} - \underset{\sim}{x} - \lambda(\underset{\sim}{H}(\underset{\sim}{y}) - \underset{\sim}{H}(\underset{\sim}{x})) \rangle_T =$$

$$\| \underset{\sim}{y} - \underset{\sim}{x} \|^2_T - 2\lambda \langle \underset{\sim}{H}(\underset{\sim}{y}) - \underset{\sim}{H}(\underset{\sim}{x}), \underset{\sim}{y} - \underset{\sim}{x} \rangle_T + \lambda^2 \| \underset{\sim}{H}(\underset{\sim}{y}) - \underset{\sim}{H}(\underset{\sim}{x}) \|^2_T \leq$$

$$(1 - 2\lambda\delta + \lambda^2K^2)\ \|\underset{\sim}{y}-\underset{\sim}{x}\|_T^2 \triangleq c^2(\lambda)\ \|\underset{\sim}{y}-\underset{\sim}{x}\|_T^2 ,$$

and $$c^2(\lambda) < 1,\ \forall\lambda\varepsilon(0,2\delta/K^2).$$ ■

III. Examples

3.1) Linear Operators and Memoryless Operators

Consider the time-invariant scalar case for simplicity, and let L_T^2 stand for $L_{1,T}^2$.

If F_ℓ is the convolution operator: $v \mapsto a*b$ where $a: \mathbb{R} \to \mathbb{R}$ is absolutely integrable, then for each $T > 0$, F_ℓ is a continuous linear operator: $L_T^2 \to L_T^2$ and therefore Gâteaux (in fact, Fréchet) differentiable. Since F_ℓ is linear $DF_\ell(x) \equiv F_\ell$, and the reader can easily verify that $(DF_\ell(x))^{adj} = F_\ell^{adj}: v(\cdot) \mapsto a(-\cdot)*v(\cdot)$, i.e., the adjoint operation turns the impulse response around in time. Furthermore, $H_\ell: v(\cdot) \mapsto [a(\cdot) + a(-\cdot)]*v(\cdot)$ is strictly increasing on L_T^2 for each $T > 0$ iff $\mathrm{Re}\,\{\hat{a}(j\omega)\} > 0$ for all ω, where $\hat{a}$ is the Fourier transform of a. This follows from a slight modification of [12:pp.25, 174,235]. Similar results hold if $a(\cdot)$ contains impulse functions as well [12:pp.246-247]. Thus $G_{opt}: v(\cdot) \mapsto a(-\cdot)* v(\cdot)$, and G_{opt} is represented in the frequency domain by the complex admittance $\hat{a}^*(j\omega)$. Therefore Theorem 1 and equation (10) reduce to the standard result $Y_{load}(j\omega) = Y^*_{source}(j\omega)$ if F_ℓ is linear and time-invariant.

Suppose F_m is memoryless but possibly nonlinear, i.e., N_1 is a resistor with the constitutive relation $i=b(v)$. Assume that $b: \mathbb{R} \to \mathbb{R}$ is differentiable and its derivative $b'(\cdot)$ is bounded. Then b is Lipschitz continuous on $\mathbb{R}$ and hence for each $T > 0$ the operator $F_m: v(t) \mapsto b(v(t))$ maps L_T^2 into L_T^2. Using Prop. 13 of [13:p.85] and the Lebesgue Convergence Theorem [13:p.88], one can show that F_m is Gâteaux differentiable on L^2_T and that for all $x,y \in L^2_T$, $(DF_m(x))y = y(\cdot)b'(x(\cdot))\in L^2_T$. Furthermore if $h: v \mapsto b(v)+vb'(v)$ is a strictly increasing function on $\mathbb{R}$, then $H_m: v(t) \mapsto b(v(t)) + v(t)b'(v(t))$ is a strictly increasing operator: $L^2_T \to L^2_T$. Thus Theorem 1 reduces in this case to the result in [2].

The reader can easily check that $DF_m: L^2_T \to \mathcal{L}(L^2_T,L^2_T)$ is not continuous unless $b'(\cdot)$ is constant. Thus if $\mathcal{N}_1$ is a resistor with any nonlinearity (other than the trivial $i=gv+\tilde{i}$), F_m is not Fréchet differentiable [4,Chap.3] on L^2_T. This is the reason Theorem 1 was formulated in terms of the weaker Gâteaux derivative.

3.2) Positive Linear Combinations of Operators

The (noncausal) matched load (10) for the source admittance $\underset{\sim}{F}$ is related to $\underset{\sim}{F}$ by a mapping ℓ, $\ell(\underset{\sim}{F}) = \underset{\sim}{G}_{opt}: \underset{\sim}{v} \mapsto (\underset{\sim}{DF}(\underset{\sim}{v}))^{adj}\,\underset{\sim}{v}$. Note that ℓ is linear; i.e. $\ell(a\underset{\sim}{F}_1 + b\underset{\sim}{F}_2) = a\,\ell(\underset{\sim}{F}_1) + b\ell(\underset{\sim}{F}_2)$. Given $\underset{\sim}{F}_1$ and $\underset{\sim}{F}_2: L_T \to L_T$, consider $\underset{\sim}{F} \triangleq a\underset{\sim}{F}_1 + b\underset{\sim}{F}_2$. The reader can easily verify that if $\underset{\sim}{F}_1$, $\underset{\sim}{F}_2$ satisfy the conditions of Theorem 1 (resp. Theorem 2), then $\underset{\sim}{F}$ also satisfies Theorem 1 (resp. Theorem 2), provided $a \geq 0$, $b \geq 0$, $a+b > 0$.

For example, consider the source shown in Fig. 2, where $\mathcal{N}_1$ consists of the parallel connection of an LTI 1-port and a nonlinear resistor. If Y and g satisfy the conditions in section 3.1), then the (noncausal) matched load has the form shown in Fig. 2.

3.3) Circuit Example

Suppose the source takes the specific form in Fig. 3, with the resistor curves shown in Fig. 4. The convolution kernel $a(t) = e^{-t}$, $t \geq 0$, for the series connection of inductor and resistor satisfies the assumptions of section 3.1. The resistor curves g_K are differentiable everywhere and

$$h_k(v) \triangleq g_k(v) + v g_k{}'(v) = (k+1)v|v|^{k-1}, \quad k = 1,2,3, \qquad (13)$$

with $h_1(0) = 0$. All the assumptions of section 3.1 are satisfied except that the derivatives $g_2{}'(\cdot)$ and $g_3{}'(\cdot)$ are unbounded. (Since they are bounded on every bounded subset of $\mathbb{R}$, a more detailed argument, omitted here, shows that the solutions obtained below maximize $\overline{P}$ over $L_T^\infty \cap L_T^2$, which is certainly sufficient in practice.)

To find the optimal output $\hat{v}$ in the three cases, we carried out the iterative procedure (11), which becomes in this instance

$$x_{j+1}(t) =$$

$$\lambda \left[6\sin(t) - (k+1)x_j(t)|x_j(t)|^{k-1} - \int_{-\infty}^{\infty} e^{-|t-\tau|} x_j(\tau)d\tau \right] + x_j(t), \quad k = 1,2,3. \qquad (14)$$

Miss Pearl Yew of MIT has written a program in PASCAL to do the numerical solution. It was run on the DEC 20 in MIT's Research Laboratory of Electronics with an initial guess of $x_0(\cdot) \equiv 0$, and found to converge fairly rapidly for small positive values of λ. The results are shown in Fig. 5.

Since g_1 represents a linear resistor, it follows from the traditional linear theorem that $\hat{v}(t) = 2\sin(t)$ for $k=1$, in agreement with the numerical solution. Note that the instantaneous current drained by the nonlinear source resistor increases in magnitude with k for $|v| > 1$ but decreases for $|v| < 1$. Thus it is intuitively reasonable that the optimal output spends a progressively greater percentage of time in the region $|v| < 1$ as k increases.

REFERENCES

[1] C.A. Desoer, "The Maximum Power Transfer Theorem for n-Ports," IEEE Trans. Circuit Theory, vol. CT-20, no. 3, pp. 328-330, May 1973.

[2] J.L.Wyatt, Jr. and L.O. Chua, "Nonlinear Resistive Maximum Power Theorem with Solar Cell Application," to appear in IEEE Trans. Circuits and Systems, Nov. 1983.

[3] R. Abraham, J.E. Marsden and T. Ratiu, Manifolds, Tensor Analysis, and Applications, Addison Wesley, Reading, MA, 1983.

[4] T.M. Flett, Differential Analysis, Cambridge University Press, 1980.

[5] W.K. Chen, Theory and Design of Broadband Matching Networks, Pergamon Press, Cambridge, England, 1976.

[6] J.W. Helton, "Broadbanding: Gain Equalization Directly from Data," IEEE Trans. Circuits and Systems, vol. CAS-28, no. 12, pp. 1125-1137, Dec. 1981.

[7] H.J. Carlin and P. Amstutz, "On Optimum Broad-Band Matching," IEEE Trans. Circuits and Systems, vol. CAS-28, no. 5, pp. 401-405, May 1981.

[8] H.J. Carlin and B.S. Yarman, "The Double Matching Problem: Analytic and Real Frequency Solutions," IEEE Trans. Circuits and Systems, vol. CAS-30, no. 1, pp. 15-28, Jan. 1983.

[9] M. Vidyasagar, "Maximum Power Transfer in n-Ports with Passive Loads," IEEE Trans. Circuits and Systems, vol. CAS-21, no. 3, pp. 327-330, May 1974.

[10] H. Flanders, "On the Maximal Power Transfer Theorem for n-Ports," Int. J. Circuit Theory and Applications, vol. 4, pp. 319-344, 1976.

[11] J. Dieudonne, Foundations of Modern Analysis, Academic Press, New York, 1969, p. 53.

[12] C.A. Desoer and M. Vidyasagar, Feedback Systems: Input-Output Properties, Academic Press, New York, 1975.

[13] H.L. Royden, Real Analysis, 2nd ed., Macmillan, New York, 1968.

Figure Captions

Fig. 1 The solution of the operator equation (6), given a particular $\underset{\sim}{i}_s(\cdot)$, is the optimal output voltage $\underset{\sim}{\hat{v}}(\cdot)$. It can be achieved with a variety of loads.

Fig. 2 The optimal load admittance is obtained by a linear operator $\underset{\cdot}{\ell}$ on the source admittance. Thus the optimal load for a parallel connection of source admittances is the parallel connection of the optimal loads for each source separately.

Fig. 3 Theorems 1 and 2 let us numerically determine the optimal output voltage $\hat{v}(\cdot)$ for this circuit when the resistor curves are as shown in Fig. 4.

Fig. 4 The three resistor curves for the circuit in Fig. 3 are $g_k(v) \overset{\Delta}{=} v|v|^{k-1}$, $k=1,2,3$, with $g_1(0) \overset{\Delta}{=} 0$.

Fig. 5 One period of the optimal output voltages for the circuit in Fig. 3.

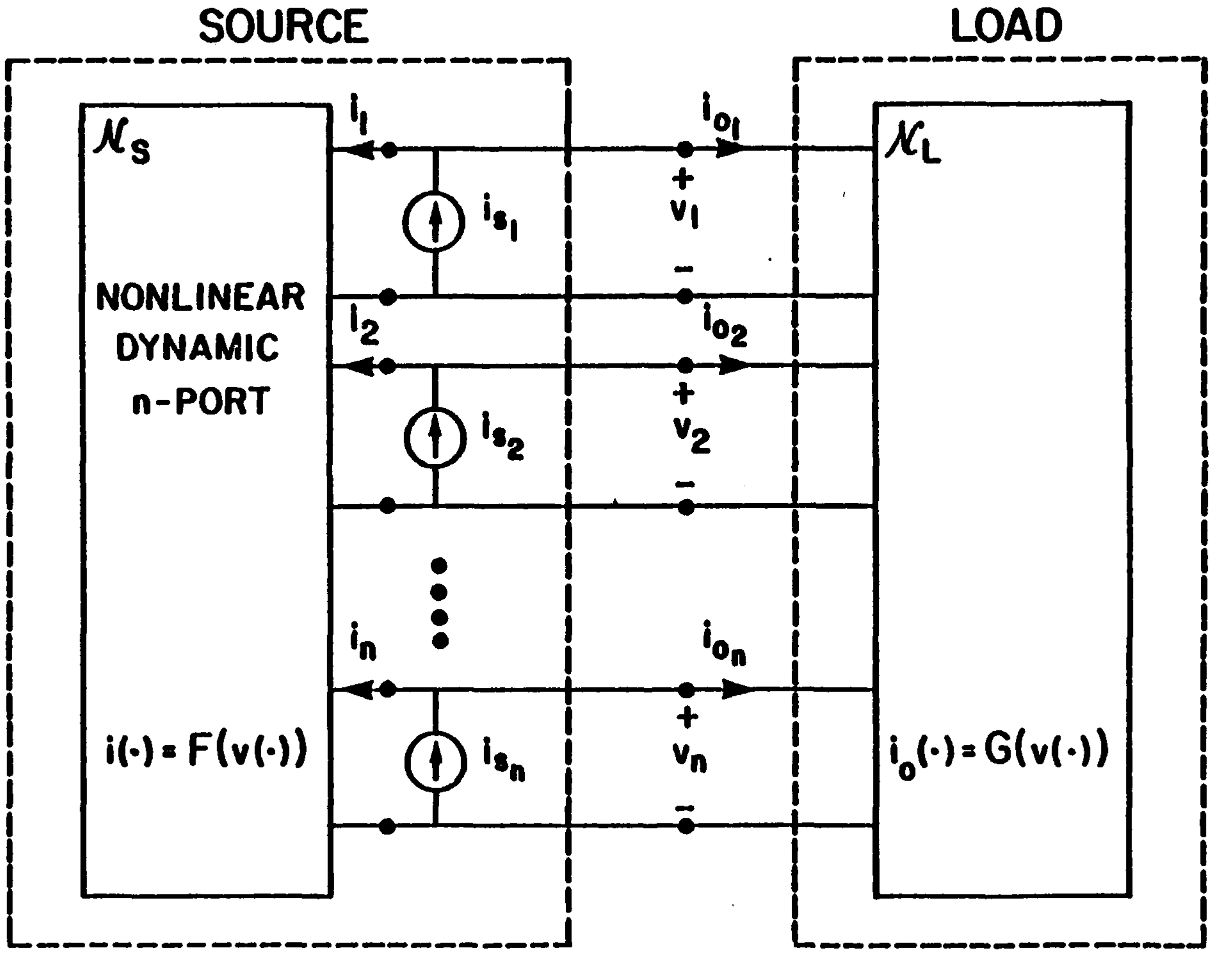

SOURCE
LOAD
$\mathcal{N}_S$
NONLINEAR
DYNAMIC
n-PORT
$i(\cdot) = F(v(\cdot))$
$\mathcal{N}_L$
$i_o(\cdot) = G(v(\cdot))$
i_1
i_2
i_n
i_{s_1}
i_{s_2}
i_{s_n}
i_{o_1}
i_{o_2}
i_{o_n}
+
v_1
−
+
v_2
−
+
v_n
−

Figure 1

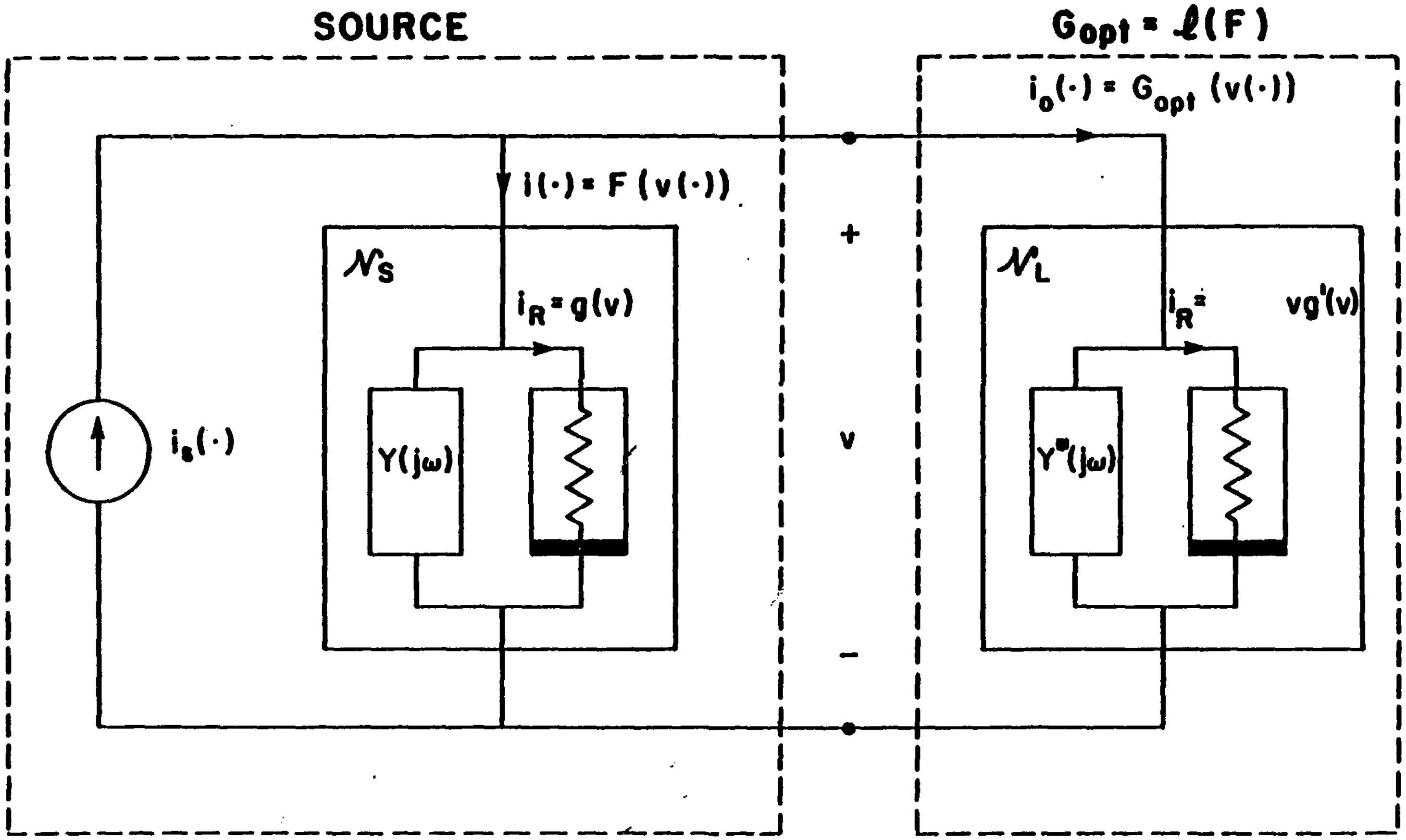

SOURCE
$G_{opt} = \mathcal{l}(F)$
$i_o(\cdot) = G_{opt}(v(\cdot))$
$i(\cdot) = F(v(\cdot))$
$\mathcal{N}_S$
$i_R = g(v)$
$Y(j\omega)$
$i_s(\cdot)$
+
v
−
$\mathcal{N}_L$
$i_R =$
$vg'(v)$
$Y^*(j\omega)$

Figure 2

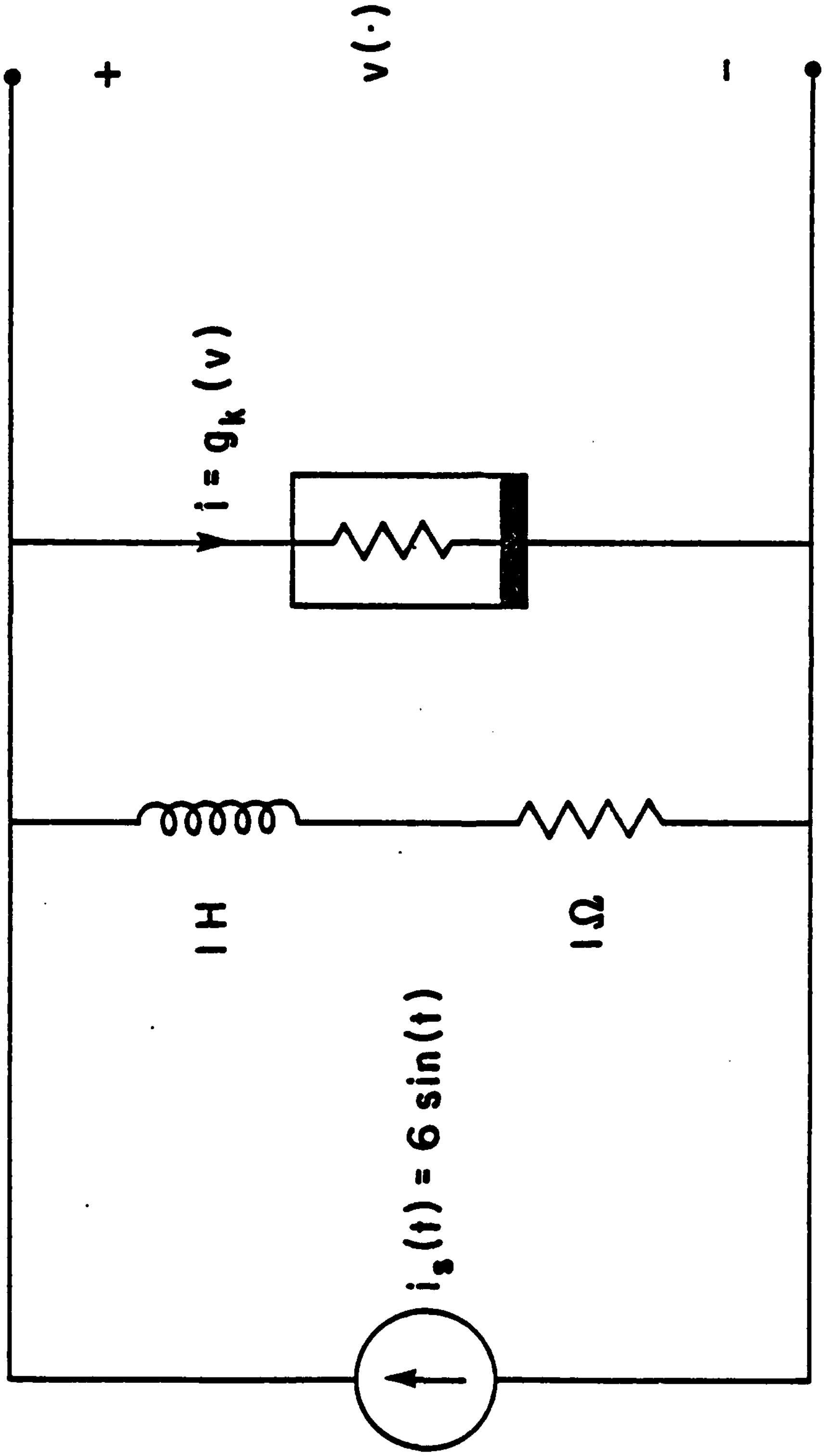


Figure 3

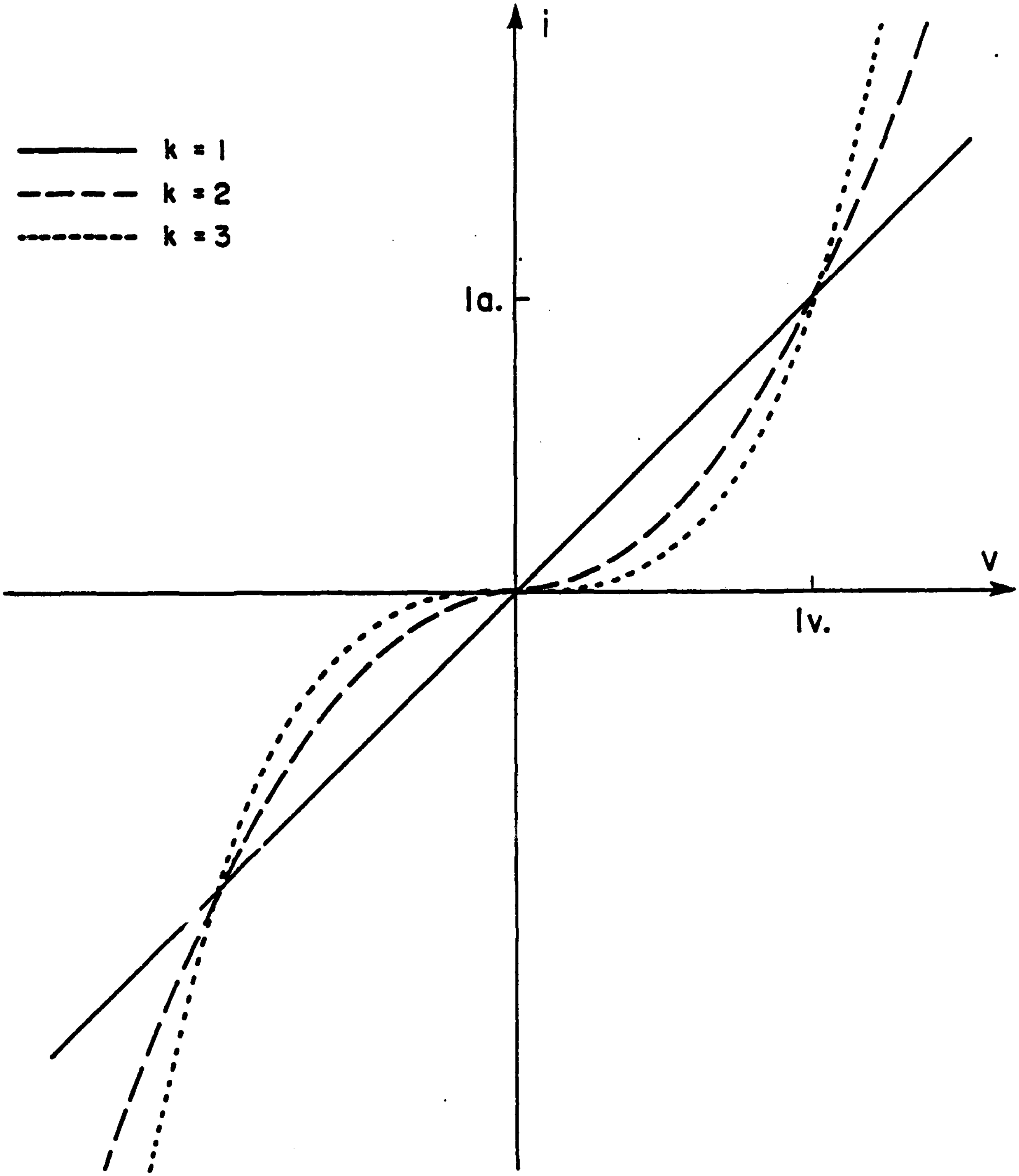


Figure 4

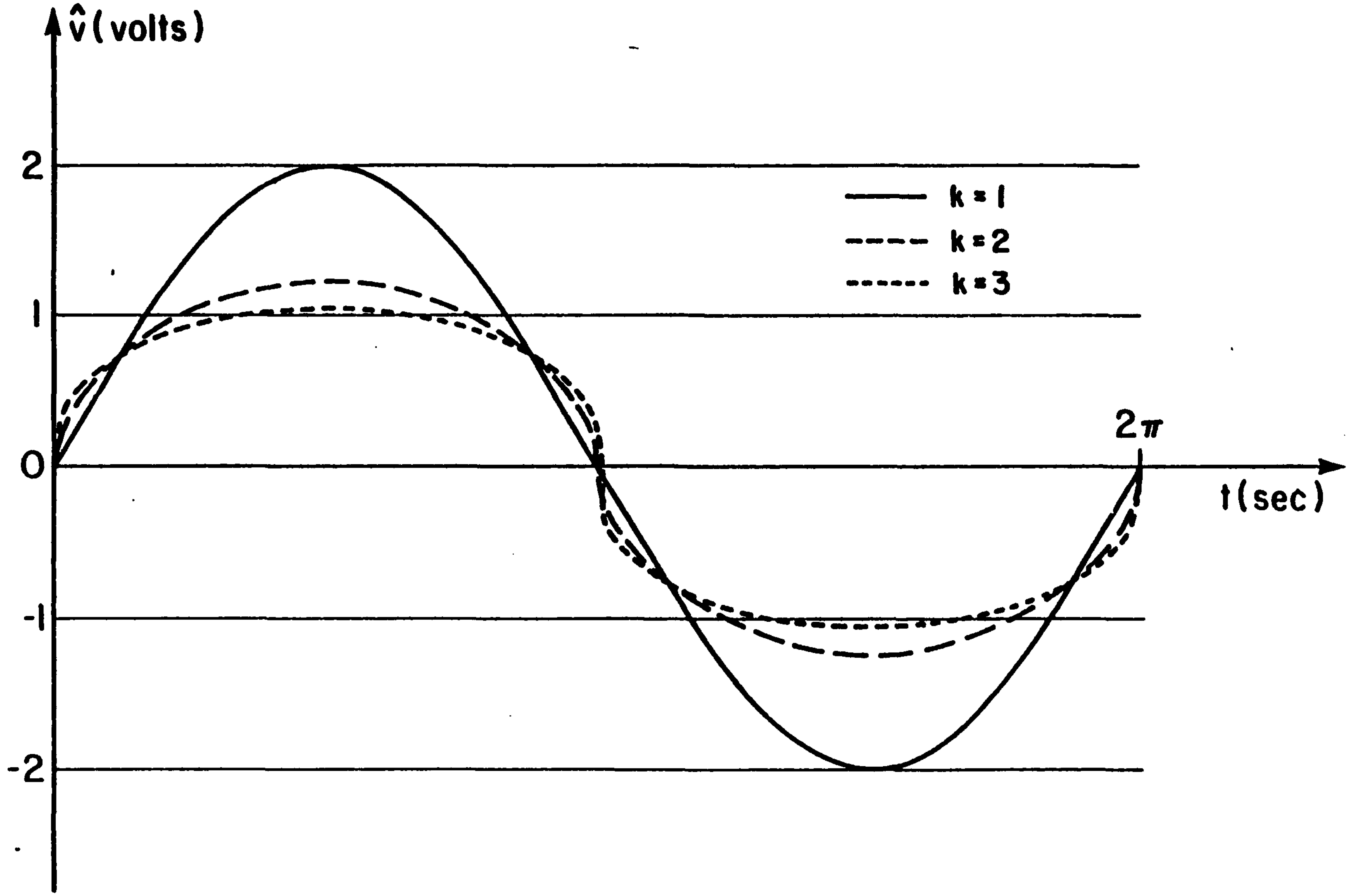


Figure 5

Electrical characteristics of Be-implanted GaAs diodes annealed with an ultrahigh power argon arc lamp

K. Tabatabaie-Alavi, A. N. M. Masum Choudhury, H. Kanbe,[a] and C. G. Fonstad
Department of Electrical Engineering and Computer Science and Center for Materials Science and Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

J. C. Gelpey
Eaton Ion Beam Systems Division, Beverly, Massachusetts 01915

(Received 5 May 1983; accepted for publication 8 July 1983)

The potential of arc lamp annealing techniques in GaAs device processing is demonstrated by the fabrication of Be-implanted mesa *pin* diodes. Implants were done at 50 and 120 keV with doses of 4.4×10^{14} and 5.1×10^{14} cm^{-2}, respectively (total dose $=9.5\times10^{14}$ cm^{-2}) into a 14-μm-thick undoped ($N_D-N_A\approx7.5\times10^{14}$ cm^{-3}) GaAs epitaxial layer grown by vapor phase epitaxy. Ten-second annealing cycles with peak temperatures of 950° and 1050 °C have been studied. The electrical characteristics of these diodes are superior to published furnace-annealed, Be-implanted GaAs diodes.

PACS numbers: 73.40.Lq, 81.40.Ef, 85.30.De, 61.70.Tm

Rapid thermal annealing (RTA) is emerging as a powerful technique in both Si and GaAs processing. Very short annealing periods, resulting in minimum dopant redistribution, and very high throughput are features of RTA ideally suited for large volume fabrication of very high speed integrated circuits (IC's). Several researchers have reported rapid thermal annealing of GaAs using halogen lamps, graphite strip heaters, and incandescent lamps.[1-3] We have previously reported annealing of Be-, Zn-, and Si-implanted GaAs using a 100-kW water-walled dc argon arc lamp.[4]

The characterization of these rapidly annealed implanted layers has so far been limited to depth profiling of carrier concentration and mobility. Although metal Schottky field-effect transistors (MESFET's) have been fabricated by rapid thermal annealing,[1] MESFET's are majority-carrier devices and their low-frequency characteristics depend primarily on mobility and carrier concentration. McLevige *et al.*[5] have shown that while good electrical activation and mobility can be obtained upon annealing of Be-implanted GaAs at 600 °C, the integrated photoluminescence intensity compared to unimplanted samples is very low unless 30 min of annealing at 900 °C is carried out. Photoluminescence intensity is, however, only indirectly related to electrical characteristics. For example, if we assume that ion implantation introduces a single Shockley–Read–Hall trap level, the nonradiative recombination lifetime $\tau_{\rm nr}$ will be inversely proportional to the trap level density N_T. The spontaneous emission lifetime $\tau_{\rm sp}$, which relates to photoluminescence intensity, is given by $(\tau_{\rm sp})^{-1}=(\tau_r)^{-1}+(\tau_{\rm nr})^{-1}$, where τ_r is the radiative recombination lifetime. On the other hand, the reverse leakage current of a GaAs or Si *p-n* diode is directly proportional to the trap density N_T in the space-charge region. Leakage current I_R, forward saturation current density I_S, ideality factor n, and maximum electric field at breakdown are generally accepted as being parameters that are very sensitive to the residual implant damage in ion-implanted *p-n* diodes.[6-7] In this letter we compare the characteristics of arc lamp annealed Be-implanted GaAs *pin* diodes with the best published furnace-annealed diodes. To our knowledge, detailed characterization of rapid-thermal-annealed GaAs *pin* diodes has not been previously reported.

Donnelly *et al.*[8] have fabricated Be-implanted diodes in vapor phase epitaxy (VPE) GaAs ($N_D-N_A=3\times10^{14}$ cm^{-3}). They obtained an average electric field of 1.5×10^5 V/cm at breakdown and observed clear Be diffusion. No more details of their diode characteristics were reported. Helix *et al.*[9] implanted Be into VPE GaAs ($N_D-N_A\approx3\times10^{15}$ cm^{-3}) at 250 keV with a 10^{14} cm^{-2} dose. Annealing was done at 900 °C for 30 min using a silicon nitride cap. They obtained an ideality factor of 1.6 and a saturation current density of 1.56×10^{-11} A/cm^2. However, the reverse leakage current increased exponentially with reverse voltage having a value of 19 nA at 0.9 times the breakdown voltage V_B for a 250-μm-diam diode. Milano *et al.*[10] made a detailed study of essentially the same diode structure as reported in Ref. 9 in both VPE and liquid phase epitaxially (LPE) grown material. Diodes made on VPE material had either a large leakage current and soft breakdown or were similar to the ones fabricated in Ref. 9. Although they obtained sharp breakdown and low leakage current for diodes made in LPE material, they observed a linear increase of reverse leakage current with reverse voltage and deduced an electron lifetime ≈6 ps in the Be-implanted p^+ region. Such a short electron lifetime is an indication of considerable residual implant damage after annealing.

Be ion implantation in this work was done at 50 and 120 keV at doses of 4.4×10^{14} and 5.1×10^{14} cm^{-2}, respectively. Implants were made into a VPE grown undoped ($N_D-N_A\approx8\times10^{14}$ cm^{-3}) GaAs layer with a nominal thickness of 14 μm. 120 nm of SiO$_2$ was sputter-deposited on both faces and samples were annealed for 10 s with maximum temperatures of 950 and 1050 °C following the time-temperature cycles illustrated in Fig. 1. The temperature plotted in Fig. 1 was obtained from an optical pyrometer viewing the back surface of the sample, i.e., the surfaces not facing the lamp. The Be

[a] Permanent address: Musashino Electrical Communication Laboratory, Nippon Telegraph and Telephone Public Corporation, Musashino-shi, Tokyo, 188 Japan.

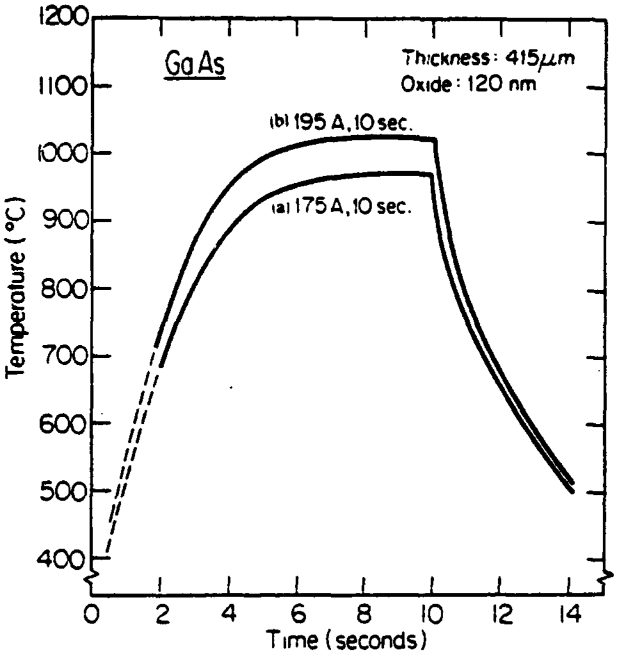


FIG. 1. Time temperature cycles used in the arc lamp annealing of the beryllium implants used in this study. The two curves shown corresponding to different peak lamp currents and thus to different peak temperatures: (a) 950 °C and (b) 1050 °C. The temperature value is the reading from an optical pyrometer viewing the back of the wafer.

concentration and mobility profiles in semi-insulating undoped GaAs that was implanted following the same implant schedule as that used for the diodes, and that was annealed following cycle A in Fig. 1 are shown in Fig. 2 (Ref. 4).

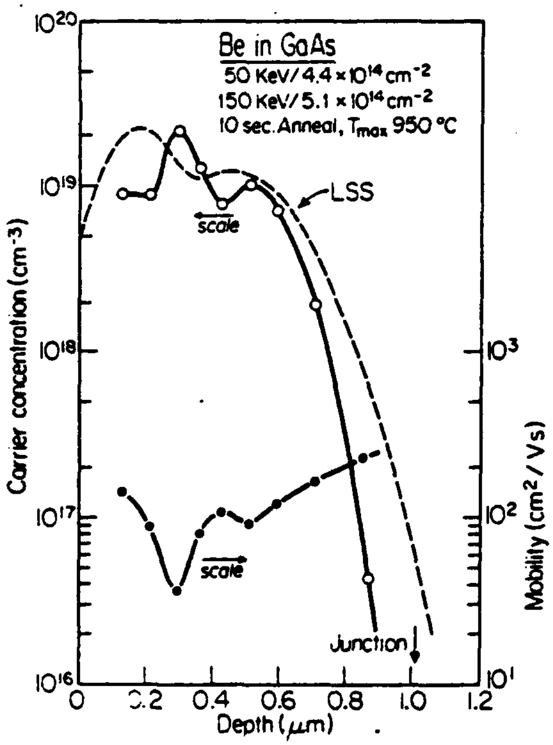


FIG. 2. Room-temperature carrier concentration and mobility profiles measured by differential Hall technique on Be-implanted (4.4×10^{14} cm^{-2}, 50 keV and 5.1×10^{14} cm^{-2}, 150 keV) GaAs samples arc lamp annealed for 10 s to 950 °C

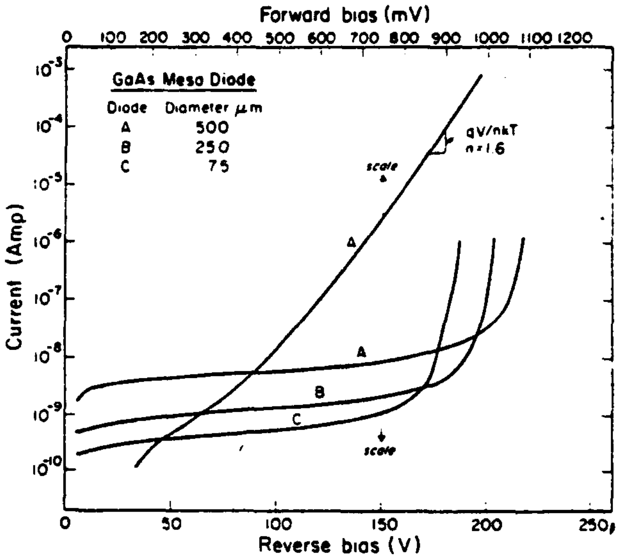


FIG. 3. Room-temperature current-voltage characteristics of three different diameter GaAs mesa *pin* diodes. The reverse bias characteristics are shown for 75-, 250-, and 500 μm diam mesas (voltage scale at bottom); the forward characteristic is shown for the 500-μm-diam mesa only (voltage scale at top).

Contact windows were opened through the oxide and Au/Zn/Au was electroplated and sintered at 420 °C to form Ohmic contact to the p^+ regions.[11] Mesa diodes with four different diameters, 75, 125, 250, and 500 μm, were fabricated. The junction depth in the VPE diodes was measured to be 1–1.1 μm using standard cleaving and staining with a 1:1:10 solution of $HF:H_2O_2:H_2O$ for 15 s with intense illumination.[12] This is consistent with the expected profile in Fig. 2, and indicates that no damage enhanced diffusion of Be, which is usually observed with furnace annealing of 10^{13}–10^{15} cm^{-2} dose levels,[8,9] has occurred.

Capacitance voltage *C-V* measurements on these diodes indicate an abrupt junction with a built-in voltage of 1.24 V. The electron carrier concentration deduced from these measurements varies from 6.4×10^{14} cm^{-3} at 1.64μm (zero bias) to 8.5×10^{14} at 3.8μm from the metallurgical junction (− 7 V bias). The epitaxial layer thickness (including the p^+ region) changes from 14 to 15 μm across the sample (13×13 mm). Such changes in thickness and doping are not unexpected for a thick VPE-grown layer.

The avalanche breakdown voltage V_B of all of the diodes was 200 ± 25 V across the sample. To calculate the punchthrough voltage and maximum electric field at avalanche breakdown, we assume an average epilayer thickness of 13.5 μm (excluding the p^+ region) and average doping level of 7.5×10^{14} cm^{-3}. Thus, the punchthrough voltage is ≈ 100 V. The maximum electric field at breakdown (200 V) is 2.2×10^5 V/cm which is close to the theoretically predicted value of 2.8×10^5 V/cm deduced from Ref. 13 for such a doping level.

The forward and reverse characteristics of 75-, 125-, 250-, and 500-μm-diam diodes annealed for 10 s at maximum temperature of 950 °C (cycle A in Fig. 1) are shown in Fig. 3. From the forward bias characteristics of a 500-μm-diam diode (0.75–1.0 V) we calculate an ideality factor of 1.6

TABLE I. Reverse leakage current at punchthrough (− 100 V) and 0.9 V_B.

Diode diameter (μm)	Annealing cycle (Fig. 1)	Leakage current − 100 V	(A) at 0.9 V_B
75	A	4.6×10^{-10}	1.6×10^{-9}
125	A	1.1×10^{-9}	4.0×10^{-9}
250	A	1.5×10^{-9}	4.8×10^{-9}
500	A	5.8×10^{-9}	2.5×10^{-8}
500	B	1.3×10^{-9}	2.7×10^{-9}

and a saturation current density of 1.4×10^{-11} A/cm^2. The room-temperature reverse leakage currents of four different diameter diodes at punchthrough (− 100 V) and $0.9V_B$ are listed in Table I. The temperature dependence of the reverse leakage current I_R, at punchthrough (− 100 V), and the avalanche breakdown voltage V_B have also been measured on a typical diode and are found to be given, respectively, by

$$I_R \propto \exp(-E_A/kT),$$

with $E_A = 0.354$ eV for 296 K $< T <$ 421 K, and

$$V_B = 137 + 1.5(T - 273)\ \text{V},$$

where k is the Boltzman constant and 296 K $< T <$ 395 K.

These diodes are superior to furnace-annealed diodes in several aspects. The reverse leakage current is low and has a normal behavior. No enhanced diffusion of Be occurs even at a dose level of 9.5×10^{14} cm^{-2}. Furthermore, the forward saturation current density and ideality factor are reasonable for a large band-gap material like GaAs and the maximum electric field at breakdown is close to the theoretically predicted value.

In conclusion, we have shown the potential of the arc lamp RTA technique in the fabrication of GaAs *pin* diodes, devices which are sensitive to residual implant damage.

The authors would like to acknowledge the help of Dr. Carl O. Bozler of MIT Lincoln Laboratory who generously provided the VPE grown GaAs material. The assistance of P. Stump of EATON with the arc lamp anneals and of M. Rothman and J. Walsh of MIT in performing the ion implantations and the SiO$_2$ deposition is also gratefully acknowledged. Extensive use was made of the Central Facilities of the MIT Center of Materials Science and Engineering, a National Science Foundation Materials Research Laboratory, during the course of this work. This work was supported by the National Science Foundation through the MIT Center for Materials Science and Engineering, grant No. DMR 78-24185, and by the Defense Advanced Research Projects Agency of the Department of Defense under Contract N00014-C-80-0622 monitored by the Office of Naval Research.

[1] Massaki Kuzuhara, Hideaki Kohzu, and Yoichiro Takayama, Appl. Phys. Lett. 41, 755 (1982).

[2] D. E. Davies, P. J. McNally, J. P. Lorenzo, and M. Julian, IEEE Electron. Devices, Lett. EDL-3, 102 (1982).

[3] R. L. Chapman, J. C. C. Fan, J. P. Donnelly, and B-Y. Tsaur, Appl. Phys Lett. 40, 805 (1982).

[4] K. Tabatabaie-Alavi, A. N. M. Masum Choudhury, C. G. Fonstad, and J C. Gelpey, Appl. Phys. Lett. 43, 505 (1983).

[5] W. V. McLevige, M. J. Helix, K. V. Vaidyanathan, and B. B. Streetman, J. Appl. Phys. 48, 3342 (1977).

[6] Peter Ashburn and Vernon Morgan, Solid State Electron. 17, 689 (1974).

[7] R. A. Powel and R. T. Fulks, J. Vac. Sci. Technol. 20, 33 (1982).

[8] J. P. Donnelly, F. J. Leonberg, and C. O. Bozler, Appl. Phys. Lett. 28, 706 (1976).

[9] Max J. Helix, K. V. Vaidyanathan, and Ben Streetman, IEEE J. Solid State Circuits SC-13, 426 (1978).

[10] R. A. Milano, M. J. Helix, T. H. Windhorn, B. G. Streetman, K. V. Vaidyanathan, and G. E. Stillman, in *Proceedings of the Seventh International Conference on Gallium Arsenide and Related Compounds*, Inst. Phys. Conf. Ser. 45, 411 (1978).

[11] K. Tabatabaie-Alavi, A. N. M. M. Choudhury, N. J. Slater, and C. G. Fonstad, Appl. Phys. Lett. 40, 398 (1982).

[12] J. C. Marinace, J. Electrochem. Soc. 110, 1153 (1963).

[13] Masanori Ito, Shuzo Kagawa, Takao Kaneda, and Toyoshi Yamaoka, J. Appl. Phys. 49, 4607 (1978).

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

CAMBRIDGE MASSACHUSETTS 02139

VLSI Memo No. 83-148 July 1983

The Waveform Bounding Approach to Timing Analysis of Digital MOS IC's*

John L. Wyatt, Jr., Charles Zukowski, Lance A. Glasser,

Paul Bassett, and Paul Penfield, Jr.**

ABSTRACT

The waveform bounding approach to fast timing analysis of MOS VLSI circuits is discussed. The idea is to compute rigorous closed-form expressions giving upper and lower bounds for transient voltage waveforms, rather than exact values. The goal is to enable rapid computation without sacrificing user confidence in the results.

*This is a preprint of a conference paper to be presented at the 1983 IEEE International Conference on Computer Design/VLSI in Computers, Port Chester, New York, Oct. 31 - Nov. 3, 1983. This work was supported by the Motorola Research Grant Fund, by the Defense Advanced Research Projects Agency (DoD) under ARPA Order No. 3972, issued by the Office of Naval Research under Contract N00014-80-C-0622, by the Air Force Office of Sponsored Research under Contract F29620-81-C-0054, by an IBM Fellowship, and by the National Science Foundation under Contract No. ECS 8118160.

**John L. Wyatt, Jr.: Department of Electrical Engineering and Computer Science, Room 36-865, M.I.T., Cambridge, MA 02139; telephone (617) 253-6716.

THE WAVEFORM BOUNDING APPROACH TO TIMING ANALYSIS OF DIGITAL MOS IC'S.

John L. Wyatt, Jr., Charles Zukowski, Lance A. Glasser,
Paul Bassett, and Paul Penfield, Jr.

Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

ABSTRACT

The waveform bounding approach to fast timing analysis of MOS VLSI circuits is discussed. The idea is to compute rigorous closed-form expressions giving upper and lower bounds for transient voltage waveforms, rather than exact values. The goal is to enable rapid computation without sacrificing user confidence in the results.

I. Background and Objectives

Existing approaches to timing analysis and simulation of digital integrated circuits fall, roughly speaking, into three classes:

i) Methods such as SPICE2 [1] and ASTAP [2], based on essentially exact numerical solution of the network's differential equations, are accurate and reliable. But even with the increase in speed afforded by the waveform relaxation method [3], exact numerical solution is too slow for the needs of the VLSI era.

ii) Specialized MOS timing simulators like MOTIS-C [4] and SPLICE [5] rely on table lookup of device characteristics for speed, and save additional time by terminating a Newton-Raphson or similar iteration before convergence is reached. SPLICE is in addition a mixed-mode circuit, timing and logic simulator and uses a selective trace algorithm to exploit latency. In both these programs the termination of an iterative step prior to convergence saves time at the cost of accuracy and, in some instances, of numerical stability [6]. The improvement in speed over SPICE2 is typically one to two orders of magnitude for SPLICE [5] and about two orders of magnitude for MOTIS-C [7].

iii) More recently, some researchers are exploring an alternate approach to timing analysis and simulation based on a radically simplified electrical description of the network. RSIM [8], CRYSTAL [9], and TV [10,11] fall at the far end of the speed-accuracy tradeoff curve from SPICE2. A MOSFET is typically represented in these programs by an extremely simplified model: a linear resistor in series with a switch. And a polysilicon or diffusion line is represented by a lumped capacitance in RSIM, or by a delay in TV obtained by simply averaging the upper and lower delay bounds obtained by Rubinstein, Penfield, and Horowitz [12]. These programs are potentially very fast and have a number of attractive user-oriented features. The drawback, of course, is that there are no absolute known limits to the error in their total delay estimates. The user can never be sure the answers they give are close enough.

The objective of the waveform bounding approach to timing analysis and simulation is to combine the computational speed that results from avoiding the numerical solution of differential equations with the user confidence in the result that comes from rigorous uncertainty bounds. Our attack on the timing analysis problem is based on a careful fundamental study of the differential equations describing the dynamics of gates, pass transistors, interconnect, and the standard digital circuits constructed from them.

In addition to the MIT group working on this project, Mark Horowitz [12,13] is currently completing a dissertation on MOS timing analysis at Stanford.

II. Response Bounds for Interconnect

2.1) Linear Interconnect Models

This section summarizes the results obtained in [12]. *In this work an MOS signal distribution* network as shown in Fig. 1 is modelled as a branched linear RC line, i.e., an RC tree, as in Fig. 2.

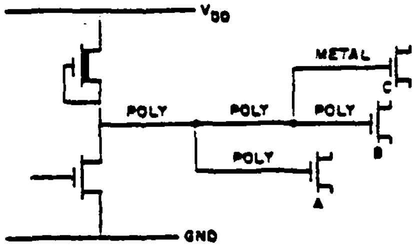


Figure 1. Typical MOS signal-distribution network. The inverter is shown driving three gates.

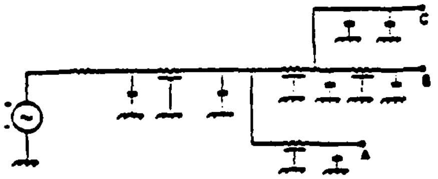


Figure 2. The linear RC tree shown above is a model for the network of Fig. 1. The voltage source is a unit step at time t = 0.

For any two nodes in the network, R_{km} is defined as the sum of the resistances along the route consisting of the intersection of the path from the input to node ℓ with the path from the input to node m, as illustrated in Fig. 3. The three time constants used to derive response bounds are

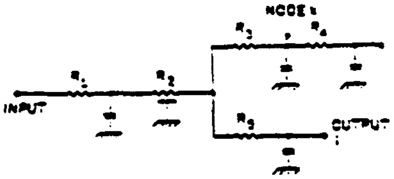


Figure 3. Illustration of resistance terms. For this network, $R_{ki} = R_1 + R_2$, $R_{kk} = R_1 + R_2 + R_3$, and $R_{ii} = R_1 + R_2 + R_5$.

$$T_P \triangleq \sum_k R_{kk} C_k \tag{1}$$

$$T_{Di} \triangleq \sum_k R_{ki} C_k \tag{2}$$

$$T_{Ri} \triangleq \sum_k (R_{ki}^2 C_k)/R_{ii}, \tag{3}$$

where the summations are taken over all nodes of the network. The derivation in [12] shows that $\underline{v}_i(t) \leq v_i(t) \leq \overline{v}_i(t)$, for all $t \geq 0$, where $v_i(t)$ is the actual zero state step response at any terminal node i, and the bounds $\underline{v}_i(t)$ and $\overline{v}_i(t)$ are given by

$$\underline{v}_i(t) \triangleq \begin{cases} 0, & 0 \leq t < T_{Di} - T_{Ri} \\ 1 - \dfrac{T_{Di}}{t+T_{Ri}}, & T_{Di}-T_{Ri} \leq t < T_P-T_{Ri} \\ 1 - \dfrac{T_{Di}}{T_P} \exp[(T_P-T_{Ri}-t)/T_P], & T_P-T_{Ri} \leq t \end{cases} \tag{4}$$

$$\overline{v}_i(t) \triangleq \begin{cases} 1 + \dfrac{t-T_{Di}}{T_P}, & 0 \leq t < T_{Di}-T_{Ri} \\ 1 - \dfrac{T_{Ri}}{T_P} \exp[(T_{Di}-T_{Ri}-t)/T_{Ri}], & T_{Di}-T_{Ri} \leq t, \end{cases} \tag{5}$$

as illustrated in Fig. 4.

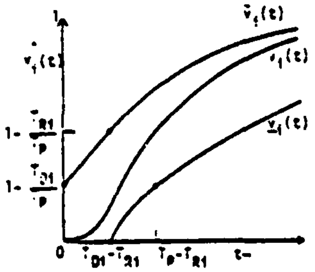


Figure 4. Form of the bounds with the distance from the exact solution exaggerated for clarity.

The time required to compute these bounds grows only linearly with the number of elements in the network. Recent applications of this result include [10,11,14,15 16]. The ultimate goal of this portion of the project is to derive a hierarchy of such bounds, permitting the user to trade off accuracy for computation time.

2.2) Nonlinearities Affecting Interconnect

The linear circuit model in Fig. 2 fails to incorporate three types of nonlinearities present in Fig. 1 or related circuits: the nonlinear output resistance of the inverter, the nonlinear gate-to-channel capacitance of the MOSFET loads, and the nonlinear capacitance from any diffusion line to substrate. This section describes recent work [17-20] that allows the bounds for linear networks [12] to be applied to RC lines incorporating such nonlinearities. (Furtner research is needed for branched lines, i.e. RC trees.)

Using the notation and sign conventions illustrated in Fig. 5, the state equations for any nonuniform, nonlinear lumped RC line with N capacitors can be written in the form

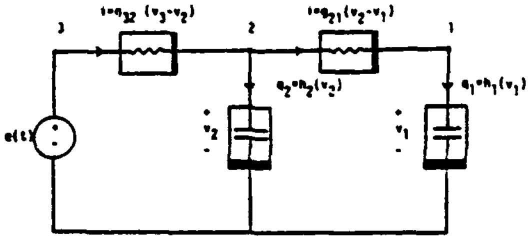


Figure 5. Two-capacitor example of a nonlinear, nonuniform RC line.

$$\dot{v}_j = \frac{1}{C_j(v_j)} \left[g_{j+1,j}(v_{j+1}-v_j) - g_{j,j-1}(v_j-v_{j-1}) \right], \quad 1 \leq j \leq N, \tag{6}$$

where $g_{10} \triangleq 0$, $v_{n+1} \triangleq e$, and the capacitor constitutive relations $q_i = h_j(v_j)$ are continuously differentiable with $C_j(v_j) \triangleq h_j'(v_j) > 0$ everywhere. We assume the resistor curves are continuously differentiable, strictly increasing, and pass through the origin.

Lemma 1 [19]

Consider any nonlinear, nonuniform RC line. At any instant during an "up" transition (i.e. $e \geq 0$, $\dot{e} \geq 0$) from equilibrium,

$$v_j(t) \geq 0,\ v_j(t) \leq e,\ v_j(t) \leq v_{j+1}(t)$$
$$\text{and } \dot{v}_j(t) \geq 0,\ 1 \leq j \leq N.$$

Lemma 1 is proved in [19]. Using it, we give a proof in [20] of the

Monotone Response Theorem for Nonlinear, Nonuniform RC Lines. Given a nonlinear RC line as described above. Suppose that (because of circuit parameter *uncertainty*, the use of linearized models for nonlinear elements, replacing the exact input by input bounds, etc.,) we do one or more of the following:

a) overestimate the input e(t),

b) underestimate one or more R's,

c) underestimate one or more C's.

The resulting circuit model will then necessarily overestimate the output $v_j(t)$ at each instant t during "up" transitions (i.e., during transitions where $e \geq 0$, $\dot{e} \geq 0$ throughout.)

A similar result holds for "down" transitions and estimate errors of the opposite sign. Using part a) of the assumptions, this theorem allows us to computationally propagate upper and lower signal bounds through the network. Using parts b) and c), it allows us to replace a nonlinear line by two linear ones, one strictly faster and one strictly slower, to which the linear network bounds (4,5) in turn apply. We have not yet succeeded in finding a generalization of this result that will apply to non-linear RC trees.

III. An Approach to Waveform Bounding for MOS Logic Gates

The results *reported here* apply to MOS device models of the form

$$i_D = \frac{W}{L} f(v_{GB}, v_{DB}, v_{SB}), \qquad (7)$$

where D,G,S and B refer to drain, gate, source and substrate, respectively. For specificity we consider only n-channel devices in this paper. No special algebraic form for f is assumed, only that f is continuously differentiable and satisfies the natural monotonicity conditions

$$\frac{\partial f}{\partial v_{GB}} \geq 0, \frac{\partial f}{\partial v_{DB}} \geq 0, \frac{\partial f}{\partial v_{SB}} \leq 0 \qquad (8)$$

everywhere. Thus a wide variety of device models are allowed, with the exception that (7) does not allow for snort-channel effects.

Our approach will be to reduce a multiple-input logic gate by steps to an "equivalent bounding inverter" and then to find bounds for the response of this inverter.

3.1) Reduction of Series-Parallel Transistor Network tc "Equivalent Bounding Transistor"

We have developed a method for reducing any series-parallel transistor network to a single "equivalent bounding transistor." Using the technique recursively, one can replace the pullup or pulldown network of a multiple-input gate by a single transistor and have rigorous bounds for the error produced by this simplification.

For example, a parallel connection of N transistors, ıll identical except for widths, lengths and gate voltages, satisfies

$$i = f(\vec{v}_{GB}, v_{DB}, v_{SB}) \triangleq \sum_{j=1}^{N} \frac{W_j}{L_j} f(v_{GB_j}, v_{DB}, v_{SB}), \qquad (9)$$

where $\vec{v}_{GB}$ is the vector of gate voltages. We have proven that, because of the assumptions (8), there exist W_{eq}, L_{eq} independent of $\vec{v}_{GB}$, and $\underline{v}_{GB}$ and $\bar{v}_{GB}$ that depend on $\vec{v}_{GB}$ such that (9) can be replaced by the simpler bounds

$$\frac{W_{eq}}{L_{eq}} f(\underline{v}_{GB}, v_{DB}, v_{SB}) \leq i \leq \frac{W_{eq}}{L_{eq}} f(\bar{v}_{GB}, v_{DB}, v_{SB}), \qquad (10)$$

for all $v_{DB} \geq v_{SB}$, describing a single transistor with a range of gate voltages. The function f is the same througnout (9) and (10). Figure 6 illustrates this process for N = 2.

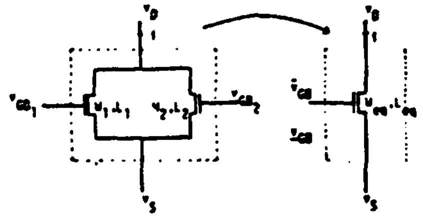


Figure 6. *Replacing a parallel transistor network by an "equivalent bounding transistor".* The cost of this simplification is that the exact value of i for the network on the left is replaced by a range of values in the simpler model corresponding to $\underline{v}_{GB} \leq v_{GB} \leq \bar{v}_{GB}$.

3.2) Reducing a Multiple-Input Gate to an "Equivalent Bounding Inverter"

A gate can be modelled as an "equivalent bounding inverter" by performing the reduction outlined in section 3.1 on both the pullup and pulldown networks, reducing each to a single transistor. Initial trials, comparing SPICE2 simulations of the original network with simulations of the "equivalent bounding inverter" indicate that the resulting bounds for $i_{out}(v_{out})$ differ from the exact values by only about ± 10% for practical circuits.

3.3) Bounding the Response of an Inverter and Load to Input Transitions

When applied to some multiple-input gates, the reduction procedure described in the previous two subsections may yield an inverter in which the pullup gate is externally driven. But for simplicity we consider *here only the case of a standard NMOS depletion - load* inverter as in Fig. 7.

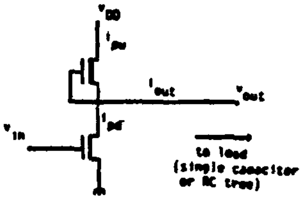


Figure 7. Depletion-load inverter.

To bound the response time of the loaded inverter we need simple bounds on the function $i_{out}(v_{out}, v_{in})$, which is the difference of the pullup and pulldown currents:

$$i_{out}(v_{out}, v_{in}) = i_{pu}(v_{out}) - i_{pd}(v_{out}, v_{in}). \qquad (11)$$

Simple linear bounds on both the pulluo and pulldown currents are shown in Fig. 8. The resulting bounds for the output curve $i_{out}(v_{out})$ depend on $v_{in}(t)$.

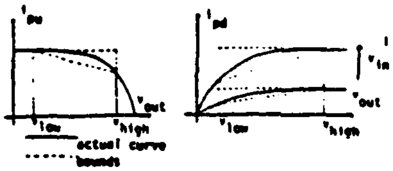


Figure 8. Simple linear bounds on the pullup and pull-down currents. The latter depend on v_{in}, and hence on t.

Initial simulations using this approach indicate that the delay bounds for these simplified models differ from the delays obtained from SPICE simulations by about ± 15%.

IV. Further Work in Progress

Much work remains to be done before the theoretical basis for the waveform bounding approach to timing analysis is complete. Among the larger remaining problems are:

1. extending the Penfield-Rubinstein bounds to incorporate time-varying source resistances, such as those *modelling tne pulldown current in Fig. 8,*

2. finding bounds for the response of an RC tree containing pass transistors,

3. investigating the tolerance in the bounds obtained so far and finding tighter ones where necessary, and

4. incorporating effects of the Miller capacitance into bounds.

ACKNOWLEDGMENT

This project was supported by the Motorola Research Grant Fund, an IBM Fellowship, and NSF Contract No. ECS8118160.

REFERENCES

[1] L.W. Nagel, "SPICE2, A Computer Program to Simulate Semiconductor Circuits," Memorandum No. ERL M520, Electronics Research Laboratory, U.C. Berkeley, Berkeley, CA, 94720, May, 1975.

[2] W.T. Weeks, A.J. Jimenez, G.W. Mahoney, D. Mehta, H. Qassemzadeh, and T.R. Scott, "Algorithms for ASTAP-A Network Analysis Program," IEEE Trans. Circuit Theory, vol. CT-20, pp. 628-634, Nov. 1973.

[3] E. Lelarasmee, A.E. Ruehli, and A.L. Sangiovanni-Vincentelli, "The Waveform Relaxation Method for Time-Domain Analysis of Large Scale Integrated Circuits," IEEE Trans. CAD, vol. CAD-1, no. 3, July 1982, pp. 131-145.

[4] S.P. Fan, M.Y. Hsueh, A.R. Newton, and D.O. Pederson, "MOTIS-C: A New Circuit Simulator for MOS LSI Circuits," Proc. 1977 IEEE Int. Symp. on Circuits and Systems, pp. 700-703, April, 1977.

[5] A.R. Newton, "Techniques for the Simulation of Large Scale Integrated Circuits," IEEE Trans. Circuits and Systems, vol. CAS-26, no. 9, pp. 741-749, Sept. 1979.

[6] G. De Micheli and A. Sangiovanni-Vincentelli, "Characterization of Integration Algorithms for the Timing Analysis of MOS VLSI Circuits," Int. Jour. of Circuit Theory and Appl., vol. 10, pp. 299-309, 1982.

[7] M.Y. Hsueh, A.R. Newton, and D.O. Pederson, "New Approaches to Modelling and Electrical Simulation of LSI Logic Circuits," Journees d'Electronique, pp. 403-413, Lausanne, Switzerland, 1977.

[8] C.J. Terman, "User's Guide to NET, PRESIM, and RNL/NL," private communication, available from the author.

[9] J. Ousterhut, "CRYSTAL: A Timing Analyzer for NMOS VLSI Circuits," Proc. 3rd Caltech Conf. on VLSI, March 1983, pp. 57-69.

[10] N. Jouppi, "TV: An NMOS Timing Analyzer," Proc. 3rd Caltech Conf. on VLSI, March 1983, pp. 71-85.

[11] N. Jouppi, "Timing Analysis for NMOS VLSI," ACM IEEE 20th Design Autom. Conf. Proc., June 1983, pp. 411-418.

[12] J. Rubinstein, P. Penfield, Jr., and M.H. Horowitz, "Signal Delay in RC Tree Networks," to appear in IEEE Trans. CAD.

[13] M. Horowitz, "Timing Models for MOS Pass Networks," Proc. 1983 IEEE Int. Symp. Circuits and Systems, pp. 198-201.

[14] R. Putatunda, "Auto Delay: A Program for Automatic Calculation of Delay in LSI/VLSI Chips," ACM IEEE 19th Design Autom. Conf. Proc., pp. 616-621, 1982.

[15] R. Putatunda, "Automatic Calculation of Delay in Custom Generated LSI/VLSI Chips," Proc. IEEE Int. Conf. on Circuits and Computers, Sept. 1982, pp. 193-196.

[16] E. Tamura, K. Ogawa, and T. Nakano, "Path Delay Analysis for Hierarchical Building Block Layout System," ACM IEEE 20th Design Autom. Conf. Proc., June 1983, pp. 403-410.

[17] J.L. Wyatt, Jr., "Inequality Theorems for Nonlinear Differential Equations," VLSI Memo 82-127, Dept. of Elect. Eng. and Comp. Sci., MIT, Cambridge,MA, 02139, Nov. 1982. All VLSI Memos are available from the Microsystems Program Office, Room 36-575, MIT.

[18] J.L. Wyatt, Jr., "Monotone Behavibr of Nonlinear RC Meshes," VLSI Memo 82-128, Nov. 1982.

[19] J.L. Wyatt, Jr., and P. Bassett, "Spatial Monotonicity and Positive Invariance for Nonlinear RC Lines and Trees," VLSI Memo 83-140, April 1983.

[20] J.L. Wyatt, Jr., "Partial Ordering and Monotone Sensitivity for Nonlinear RC Meshes and Lines," VLSI Memo 83-145, June, 1983.

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

CAMBRIDGE, MASSACHUSETTS 02139

VLSI Memo No. 83-153 October 1983

Global Wire Routing in Two-Dimensional Arrays*

R. M. Karp,** F. T. Leighton,*** R. L. Rivest,***

C. D. Thompson,** U. Vazirani,** V. Vazirani,**

ABSTRACT

We examine the problem of routing wires on a VLSI chip, where the pins to be connected are arranged in a regular rectangular array. We obtain tight bounds for the worst-case "channel-width" needed to route an n X n array, and develop provably good heuristics for the general case. An interesting "rounding algorithm" for obtaining integral approximations to solutions of linear equations is used to show the near-optimality of single-turn routings in the worst-case.

*To appear in Proc. Foundations of Computer Science, Tucson, AZ, November 1983. At M.I.T., this research was supported by funds from NSF grant MCS-80-06938, DARPA contract N00014-80-C-0622, Air Force contract OSR-82-0326, and a Bantrell Fellowship.

**Computer Science Division, U.C.Berkeley, Berkeley, CA 94720.

***Mathematics Department and Laboratory for Computer Science, M.I.T., Cambridge, MA 02139; Leighton: Room NE43-836A, telephone(617)253-5876.

Global Wire Routing in Two-Dimensional Arrays[†]

(Extended Abstract)

R. M. Karp[*]
F. T. Leighton[**]
R. L. Rivest[**]
C. D. Thompson[*]
U. Vazirani[*]
V. Vazirani[*]

[*]Computer Science Division
573 Evans Hall
U.C. Berkeley
California 94720

[**]Mathematics Department and
Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139

Very Brief Abstract

We examine the problem of routing wires on a VLSI chip, where the pins to be connected are arranged in a regular rectangular array. We obtain tight bounds for the worst-case "channel-width" needed to route an $n \times n$ array, and develop provably good heuristics for the general case. An interesting "rounding algorithm" for obtaining integral approximations to solutions of linear equations is used to show the near-optimality of single-turn routings in the worst-case.

Problem Definition

We use a classical model wherein the chip area is considered to be divided into a uniform $n \times n$ array of square *cells*. Each cell contains p *pins* (connection points for logic elements). Each instance of our routing problem specifies a collection of *nets* where each net is specified as a set of pins. (Each pin is on at most one net.) Each net is to be connected together by horizontal and vertical wires. Unless stated otherwise, we assume that $p = 1$ and that each net connects exactly two pins.

A global routing problem instance specifies a *pin placement*, so that the only remaining work is to route the wires between the pins. For this reason, the global routing problem is a special case of (and perhaps easier than) the general placement and routing problem studied in [T79, L80, L81, L82, BL83].

It is common to solve a global routing problem instance P in two steps:

(1) compute a *global routing* R specifying for each net the set of cells and cell edges to be traversed by the wiring for that net, and

(2) compute a *detailed routing* that specifies for each net the exact position of each wire, which follows the previously computed global routing and satisfies the usual separation constraints between wires, etc.

In this paper we are concerned exclusively with the problem of finding good global routings (which we henceforth call *routings*).

T-turn Routings

We are particularly concerned with *t-turn routings*, in which the path for each net contains at most t turns. A one-turn routing will have for each wire either a straight wire segment or an "L"-shaped wire-segment. The number of turns in a global routing is the least number of turns in any detailed routing consistent with the global routing. When horizontal and vertical wires are implemented on distinct layers, then the number of turns required is equal to the number of "vias" or "contact cuts" required to join the straight-line wire segments together. In the general case (e.g. when $p > 1$) we identify the number of "turns" with the number of vias required to implement the wiring pattern, or (equivalently) the sum for each net of the number of cells for which the global routing for that net crosses both a horizontal and a vertical side of the cell.

Notation: We denote the set of global routings for problem instance P by $\Gamma(P)$. The set of t-turn global routings are denoted by $\Gamma_t(P)$.

Example

Figure 1 presents an example of our global routing problem on a 4×4 grid with 8 nets. Figure 2 presents a typical solution to this problem, which happens to be in $\Gamma_1(P)$.

[†] This research was supported by funds from Semiconductor Research Corporation grant 1-44247-52055, NSF grant MCS 82-04506, NSF grant ECS 81-10684, NSF grant MCS-80-06938, NSF grant MCS-81-05217, DARPA contract N00014-80-C-0622, Air Force contract OSR-82-0326, a Bantrell Fellowship, and an IBM Graduate Fellowship.

Row \ Column	1	2	3	4
1	6	6	7	8
2	5	2	1	3
3	7	4	1	2
4	4	8	5	3

Figure 1

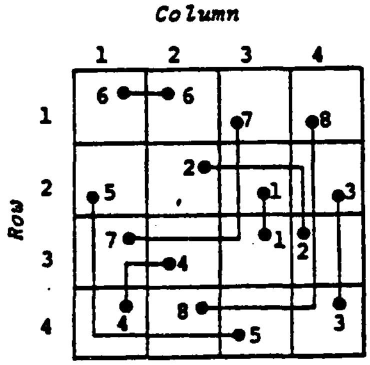


Figure 2

Channel Widths

Let P denote an instance of our global routing problem and let R denote a global routing solving P.

Notation: Let $w(R)$ denote the maximum number of wires passing from one cell into an adjacent one in the global routing R.

Remarks: Intuitively, $w(R)$ is the "channel width" which is needed to route the wires of the solution R, so we call w the "width" of the solution R. The one-turn routing R of Figure 2 has width 3 (there are three wires between cell (2,4) and cell (3,4)). Flipping either net 2 or net 8 to its other "L" configuration will reduce the width to 2. The reader can convince himself that no one-turn routing has width one by considering nets 1, 6, and 7.

Definition: An *optimum* global routing R is one that minimizes $w(R)$ over all global routings for the given problem instance (i.e. over all $R \in \Gamma(P)$).

Notation: *(Width of a problem instance P.)* We let $w(P)$ denote width of an optimal routing R for P.

Notation: *(Width of the best t-turn routing for P.)* We let $w_t(P)$ denote the least width of any t-turn routing R that solves P.

Notation: *(Worst-case width for n × n arrays.)* We let $w(n)$ denote the maximum width of any problem instance defined on an $n \times n$ array.

Notation: *(Restriction to t-turn routings.)* We let $w_t(n)$ denote the maximum of $w_t(P)$ for any problem instance P defined on an $n \times n$ array.

Notation: For $p \neq 1$, we use the notations $w(n,p)$ or $w_t(n,p)$.

Remarks: The reader will be able to distinguish the notations $w(R)$, $w(P)$, and $w(n)$ by the type of the argument.

Motivation

Our research was motivated by the following intriguing conjecture.

Conjecture (*Thompson*): $w(n) = w_1(n) = \lfloor \frac{n}{2} \rfloor + 1$.

This controversial-sounding conjecture states that in the worst-case we need only consider one-turn routings.

On the other hand, it is only requiring that for any problem instance P there exist a one-turn routing R for P such that $w(R) \leq w(n)$ and not that $w(R) \leq w(P)$. (It is not difficult to develop problem instances P for which $w(P) = 1$ but $w_1(P) = \Omega(n)$.)

Results

Our major theorems are listed here; proofs and proof sketches are generally given later.

Theorem 1. $\lfloor \frac{n}{2} \rfloor \leq w(n) \leq n$.

Proof: For the lower bound connect (i,j) to $(i,n-j)$ for $1 \leq j \leq \frac{n}{2}$, and consider the number of wires that must cross from column $\lfloor \frac{n}{2} \rfloor$ to $\lfloor \frac{n}{2} \rfloor + 1$. For the upper bound use any routing in $\Gamma_1(P)$ for a given instance P. ∎

Theorem 2. $\lfloor \frac{n}{2} \rfloor + 1 \leq w_1(n) \leq \lfloor \frac{n}{2} \rfloor + 2$.
Furthermore, a one-turn routing R with $w(R) \leq \lfloor \frac{n}{2} \rfloor + 2$ can be computed in time $O(n^3 \log(n))$.

Remarks: Theorem 2 very nearly proves part of Thompson's conjecture. We do not know how to resolve the small difference remaining in Theorem 2. The upper-bound proof involves the development of an elegant algorithm of independent interest for computing a good *integral* approximation to the solution of a set of linear equalities. The following theorem states the main result used.

Theorem 3. *[The Rounding Theorem]* Let $\mathbf{A}$ be a real-valued $r \times s$ matrix and let Δ be a positive real number such that in every column of $\mathbf{A}$,

(*i*) the sum of the positive elements is $\leq \Delta$, and

(*ii*) the sum of the negative elements is $\geq -\Delta$.

Let $\mathbf{x}$ be an s-vector and $\mathbf{b}$ and r-vector such that $\mathbf{A}\mathbf{x} = \mathbf{b}$. Then there exists an *integral* s-vector $\hat{\mathbf{x}}$ such that

(*i*) for all i, $1 \leq i \leq s$, either $\hat{\mathbf{x}}_i = \lfloor \mathbf{x}_i \rfloor$ or $\hat{\mathbf{x}}_i = \lceil \mathbf{x}_i \rceil$ (i.e. $\hat{\mathbf{x}}$ is a "rounded" version of $\mathbf{x}$).

(*ii*) $\mathbf{A}\hat{\mathbf{x}} = \hat{\mathbf{b}}$, where $\hat{\mathbf{b}}_i - \mathbf{b}_i \leq \Delta$ for $1 \leq i \leq r$.

Furthermore, $\hat{\mathbf{x}}$ can be computed from $\mathbf{A}$, $\mathbf{x}$, and $\mathbf{b}$ in time $O(r^3 \log(\frac{s}{r}))$.

Remarks: The Rounding Theorem says that when A has only a few small nonzero entries in each column, then we can effectively round x to a nearby *integral* point $\hat{x}$ while keeping $A\hat{x}$ from increasing very much over Ax.

Theorem 4. It is NP-complete to determine, given an instance P of our global routing problem, whether $w_1(P) \leq \lceil \frac{n}{2} \rceil - 2$.

Remarks: This result is perhaps surprising in view of Theorem 2; the approximation algorithm presented there is remarkably good. The result can also be improved, although we do not include the details in the abstract. In particular, it is also NP-complete to determine whether $w_1(P) \leq \lceil \frac{n}{2} \rceil - 1$. When p is even, it is NP-complete to determine whether $w_1(P) < \frac{pn}{2}$. Given the result proved in Theorem 5, this result is as tight as possible.

Theorem 5. When p is even, $w_1(n,p) = \frac{pn}{2}$.

Corollary. $\lfloor \frac{n}{2} \rfloor \leq w_3(n) \leq \lceil \frac{n}{2} \rceil + 1$.

Corollary. When p is odd, $p \cdot \lfloor \frac{n}{2} \rfloor \leq w_3(n,p) \leq \lceil \frac{pn}{2} \rceil + p$.

Remarks: Theorem 5 shows that three-turn routings can yield an improvement (by one). The upper bound proof uses an elegant argument based on finding Eulerian tours in an associated graph.

Theorem 6. There is a polynomial time approximation algorithm achieving

$$w(R) \leq O(w(P) \cdot \log(\frac{pn}{w(P)}))$$

for any problem instance P.

Remarks: The proof of theorem 6 involves a hierarchical bottom-up approach, using a recursion based on 2×2 subdivisions. We believe it is possible to reduce the logarithmic term to a constant, but have not yet been able to do so. The result is also valid for P containing multipoint nets.

Theorem 7. If $n \equiv 2 \pmod 4$ or $n \equiv 3 \pmod 4, w(n) \geq \lfloor \frac{n}{2} \rfloor + 1$.

Remarks: This lower bound extends that of Theorem 2 to handle routings having arbitrarily many turns, in the cases indicated.

Theorem 8. $w_2(n) \leq \lfloor \frac{n}{2} \rfloor + 1$.

Remarks: This theorem refines the techniques and results of Theorem 5 and its first corollary, moving from three-turn to two-turn nets and improving the upper bound for odd n by one.

Discussion of the Model

Chen *et. al* [CFKNS77] give an excellent overview of how IBM uses algorithms for solving this global routing problem to automatically wire master-slice logic arrays for their System/370 implementations. The model is particularly appropriate for gate-array technologies where each cell might contain a single NAND gate. Fabrication turn-around time can be very small here since wafers can be preprocessed to contain the array of gates and the only processing required once the logic design is finished is to produce horizontal and vertical wiring on the last two metal layers, to connect the gates together as desired. However, the preprocessing involved usually fixes an upper bound on the value of $w(R)$ that will be allowed – if all routing channels between gates have width 20 then the routing can not be realized if $w(R) > 20$.

As noted earlier, our concern is with "worst-case" values $w(n)$; in practice one would expect "typical" chips to have $w(P)$ substantially less than $w(n)$.

Related Work

Burstein and Pelavin [BP83] present an interesting recent "hierarchical" approach to this global routing problem. Much of the earlier algorithmic work (e.g. [HN83]) involved variations on standard shortest-path algorithms, used to route one net at a time. Some probabilistic models have been developed by El Gamal [EG81] to estimate $w(P)$ under various assumptions about the average distance between the pins on a net, *etc.*, in a typical instance P. Johnson [J82] gives an overview of the NP-completeness results known in this area.

Proof of Theorem 2: One-Turn Routing by Rounding

Theorem 2. $\lfloor \frac{n}{2} \rfloor + 1 \leq w_1(n) \leq \lfloor \frac{n}{2} \rfloor + 2$.

Proof: For $n = 2$, the lower bound example is easily constructed. (For example, see Figure 3.) For larger values of n, simply embed the 2-by-2 example in a "width-2 cross" of 0-turn vertical and horizontal nets.

The upper bound is proved using the Rounding Theorem. We first describe how to apply the Rounding Theorem to our routing problem, and then in the next section describe a surprisingly efficient "rounding algorithm".

We assume for convenience here that n is even. Let x_i be a $0-1$ valued variable associated with net i indicating which of the two L-shaped routes will be used. The interpretation is fixed but arbitrary. We assume here that each L-shaped route has *exactly* two wire segments. If both pins for a net lie in the same row or column we assume the two L-shaped routes are distinguished by the inclusion of different zero-length wire segments at their ends. (These are degenerate L-shapes with one leg of the L having zero length.) Each assignment of $0-1$ values to $\mathbf{x} = (x_1, \ldots, x_{n^2/2})$ places an easily computed number of wire segments in each row and column. For example, in the problem of Figure 3 the number of wire segments in column 1 is $(1 - x_1) + x_2$.

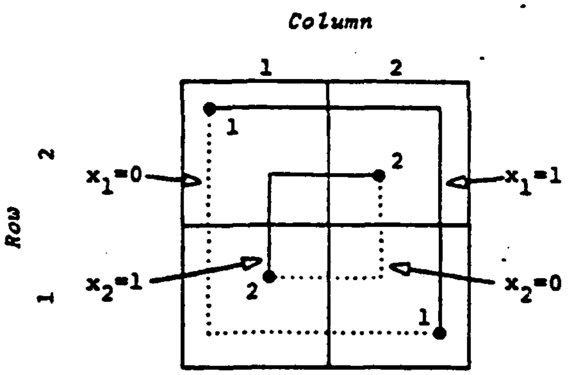


Figure 3

It is then simple to write a set of equations specifying that each row and column will contain exactly $\frac{n}{2}$ wire segments:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \qquad (*)$$

where A is a $(2 \times n) \times (\frac{n^2}{2})$ real-valued matrix. Each variable x_i will participate in at most four constraints, since its two L-routes affect the wire segmeht count in at most two rows and two columns. Furthermore, it is easy to check that A satisfies the conditions of the rounding theorem with $\Delta = 2$, since each x_i will enter two constraints positively and two negatively. Finally, it is easy to see that the vector $\mathbf{x} = (1/2, 1/2, 1/2, \ldots, 1/2)$ satisfies the equation $(*)$, since each net endpoint will then add 1/2 to the wire segment count for its row and column.

Applying the rounding theorem, we infer the existence of a $0-1$ valued vector $\hat{\mathbf{x}}$ such that

$$\mathbf{A}\hat{\mathbf{x}} \leq \mathbf{b} + (2, 2, \ldots, 2)$$

. Except for the claims regarding running times, this proves Theorem 2. ∎

Proof of Theorem 3: The Rounding Algorithm

Theorem 3. [*The Rounding Theorem*] Let A be a real-valued $r \times s$ matrix and let Δ be a positive real number such that in every column of A ,

(*i*) the sum of the positive elements is $\leq \Delta$, and

(*ii*) the sum of the negative elements is $\geq -\Delta$.

Let x be an s-vector and b and r-vector such that $\mathbf{A}\mathbf{x} = \mathbf{b}$. Then there exists an *integral* s-vector $\hat{\mathbf{x}}$ such that

(*i*) for all i, $1 \leq i \leq s$, either $\hat{\mathbf{x}}_i = \lfloor \mathbf{x}_i \rfloor$ or $\hat{\mathbf{x}}_i = \lceil \mathbf{x}_i \rceil$ (i.e. $\hat{\mathbf{x}}$ is a "rounded" version of x).

(*ii*) $\mathbf{A}\hat{\mathbf{x}} = \hat{\mathbf{b}}$, where $\hat{\mathbf{b}}_i - \mathbf{b}_i \leq \Delta$ for $1 \leq i \leq r$.

Furthermore, $\hat{\mathbf{x}}$ can be computed from A , x , and b in time $O(r^3 \log(\frac{s}{r}))$.

Proof: We now describe a "rounding algorithm" that efficiently computes the vector $\hat{\mathbf{x}}$ whose existence is assured by the Rounding Theorem. The input and output parameters are as described in that theorem:

Input: A , Δ, b , x .

Output: $\hat{\mathbf{x}}$

The execution time of our rounding algorithm is $O(r^3 \log(\frac{s}{r}))$. In our routing application, we have $r = 2n$ (one equality for each row or column) and $s = \frac{n^2}{2}$ (one variable for each net), so the execution time is $O(n^3 \log(n))$. This compares favorably with the more usual approach based on shortest paths, which runs in time $O(n^4)$ to route an $n \times n$ array.

The steps of our rounding algorithm are:

Step 1. [*Convert to* $0-1$ *problem.*] Replace x by $\mathbf{x} - \mathbf{x}'$, where $\mathbf{x}'_i = \lfloor x_i \rfloor$ for all i. Replace b by $\mathbf{b} - \mathbf{A}\mathbf{x}'$. Solve the modified problem (steps 2 to 3) and then convert back by adding $\mathbf{x}'$ to the $\hat{\mathbf{x}}$ computed and $\mathbf{A}\mathbf{x}'$ to the $\hat{\mathbf{x}}$ computed. Halt.

Step 2. [*Fast reduction in the number of variables.*] This step reduces the number of variables to $\leq r$ by $O(\log(\frac{s}{r}))$ passes through steps $2a - 2f$.

2a. [*Test if done.*] If $s \leq r$, go to step 3.

2b. [*Grouping.*] Divide the s variables into $(r+1)$ groups, each of roughly the same number of variables. Consider a new problem $\mathbf{C}\mathbf{y} = \mathbf{b}$ where y is an $(r+1)$-vector having one element for each group, and C is $r \times (r+1)$ matrix. C and y are obtained from A and x by adding the constraint that the within eacl group each variable will have the same value. For example, the first column of C is the sum of the columns of A corresponding to variables in the first group, and $\mathbf{y}_1$ is the sum of the $\mathbf{x}_i$'s from the first group.

2c. [*Reduce C to row-echelon form.*] Using elementary row operations, convert the $r \times (r+1)$ matrix C to row-echelon form, as in Figure 4, (if C has rank r). Note that this operation does not change the null space of C .

$$\mathbf{C} = \left[\begin{array}{cccccc|c} 1 & & & & & & \\ & 1 & & & 0 & & \\ & & 1 & & & & \alpha \\ & & & 1 & & & \\ 0 & & & & 1 & & \\ & & & & & 1 & \end{array} \right]$$

Figure 4

2d. [*Round.*] Let s be an $r+1$-vector in the null space of C. (This is easy to compute given step 2c.) Let $\lambda^* = \min\{\lambda \geq 0 \mid \mathbf{y} + \lambda \times s\text{has an integral component}\}$ and let

$$\mathbf{w} = \mathbf{y} + \lambda^* \times s$$

2e. [*Update.*] For each variable $\mathbf{x}_i$ in a group j where $\mathbf{w}_j$ is integral, fix $\hat{\mathbf{x}}_i$ at $\mathbf{w}_j$ and remove $\mathbf{x}_i$ from the problem (set $\mathbf{b} = \mathbf{b} - \mathbf{w}_j \cdot \mathbf{A}_{[,i]}$, where $\mathbf{A}_{[,i]}$ is the i-th column of A , delete $\mathbf{x}_i$ from x and delete the i-th column of A .) Set the remaining $\mathbf{x}_i$'s to their group values $\mathbf{w}_j$'s.

2f. [*Revise group structure.*] If now $s \leq r$, go to step 3. Else split the largest group into two smaller ones, update C to reflect the changes in steps 2d and 2e, and return to step 2d.

Step 3. [*One by one reduction in equalities and variables.*] If $s \leq r$ execute step 3a, else execute step 3b. Repeat step 3 until all variables have been fixed. Then halt; the desired solution has been found.

3a. [*$s \leq n$: Eliminate an equality.*] Find an i such that the elimination of equality i will not affect the final result (it can be proven that such an i always exists in this case). Eliminate this row from matrix A and from b .

3b. [*Eliminate a variable.*] This is much as in step 2, except we may only eliminate one variable; here each variable is in its own group.

This completes our description of the rounding algorithm. It is not too difficult to verify the claimed running time. ▮

Proof of Theorem 4: NP-Completes of Optimal One-Turn Routing

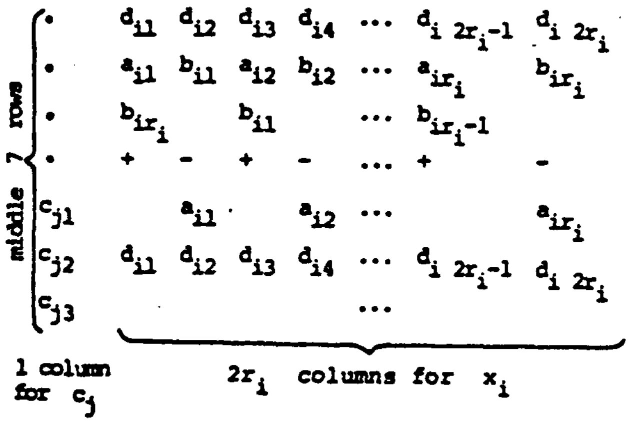
Theorem 4. It is NP-complete to determine, given an instance P of our global routing problem, whether $w_1(P) \leq \lceil \frac{n}{2} \rceil - 2$.

Proof: The reduction is from 3-SAT. Given an instance E of 3-SAT with variables $x_1, x_2, \ldots, x_q$ and clauses $c_1, c_2, \ldots, c_m$, set $n = 14m + 3$ and define the routing problem P as follows.

Pins in the rightmost $7m + 3$ columns of the grid are not included in any net. Any 1-turn routing of P can thus have row widths at most $7m = \lceil \frac{n}{2} \rceil - 2$. Therefore, we are only concerned with column widths in what follows.

Pins in the leftmost $7m$ columns but not in the middle 7 rows are paired so that the pin in the ith row of the jth column is linked to the pin in the $(n-i+1)$st row of the jth column. Each of these nets must be routed as a vertical wire, and the question of whether $w_1(P) \leq \lceil \frac{n}{2} \rceil - 2$ is equivalent to the question of whether the middle 7 rows can be routed with column width 2.

The middle 7 rows of the leftmost m columns are used to represent the clauses (one column for each clause). The middle 7 rows of the next $6m$ columns are used to represent the variables (*$2r_i$ columns for variable x_i where r_i is the number of times x_i* appears in E). The columns used to represent c_j and x_i are shown in Figure 5. The order of the columns from left to right is arbitrary.

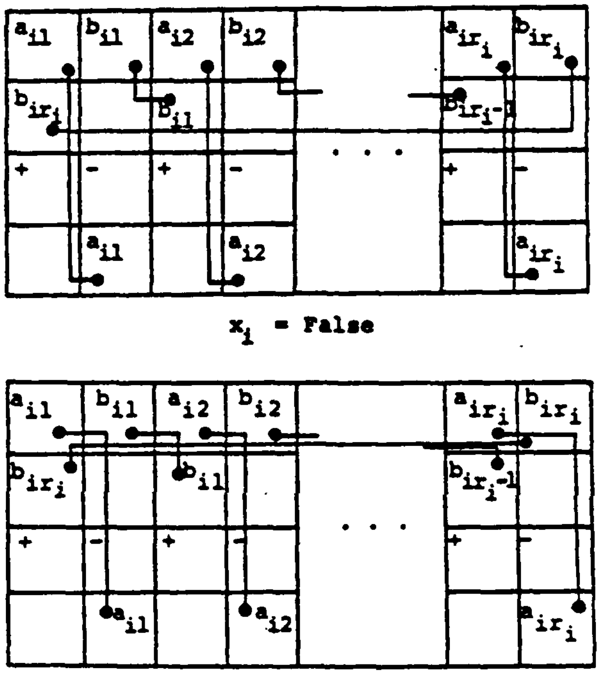


middle 7 rows
• d_{i1} d_{i2} d_{i3} d_{i4} … $d_{i\ 2r_i-1}$ $d_{i\ 2r_i}$
• a_{i1} b_{i1} a_{i2} b_{i2} … a_{ir_i} b_{ir_i}
• b_{ir_i} b_{i1} … b_{ir_i-1}
• + - + - … + -
c_{j1} a_{i1} a_{i2} … a_{ir_i}
c_{j2} d_{i1} d_{i2} d_{i3} d_{i4} … $d_{i\ 2r_i-1}$ $d_{i\ 2r_i}$
c_{j3} …
1 colum for c_j
$2r_i$ columns for x_i

Figure 5

If the kth term in clause c_j ($k = 1, 2$ or 3) is x_i, then any one of the $+$ symbols in the $2r_i$ columns for x_i is replaced by c_{jk}. If the kth term in clause c_j is $\overline{x_i}$, then any one of the $-$ symbols in the $2r_i$ columns for x_i is replaced by c_{jk}. Since x_i appears r_i times in E, there are always enough $+$'s and $-$'s for all the c_{jk}'s. The remaining $+$'s and $-$'s (as well as the dots) are not assigned to a net. In what follows, we show that the middle 7 rows can be routed with column width 2 if and only if E is satisfiable.

Clearly the nets labeled with d_{ik}'s must be routed as vertical wires. This leaves only two ways to route the nets labeled with a_{ik}'s and b_{ik}'s. The two routings correspond in a natural way to the truth value of the associated variable x_i. The routings are shown in Figure 6.



x_i = False

x_i = True

Figure 6

It remains to route the c_{jk}'s. It is easily shown that if c_{jk} corresponds to x_i where x_i has a true routing or to $\overline{x_i}$ where x_i has a false rout:n_ 'hen net c_{jk} can be safely routed without using a vertical wire segment in the column for c_j. This is not the case if c_{jk} corresponds to x_i where x_i has a false routing or to $\overline{x_i}$ where x_i has a true routing. In the latter cases, the net for c_{jk} must include a vertical wire segment in the column for c_j that passes through the top of the cell containing c_{j1}. Hence the middle 7 rows can be routed with column width 2 if and only if there is a k for each j such that c_{jk} corresponds to x_i where x_i has a true routing or to $\overline{x_i}$ where x_i has a false routing. This condition is equivalent to E being satisfiable. ▮

Proof of Theorem 5: Routing using Eulerian Tours

Theorem 5. When p is even, $w_1(n,p) = \frac{pn}{2}$.

Proof: Let each of the cells of the $n \times n$ array be the vertices of a graph, and connect any two vertices that are connected by a net.

Since p is even, every vertex will have an even degree.

Thus the edges can be organised into a directed path which is an Eulerian tour, traversing each edge exactly once. (The case that the graph is not connected arises but is easy to handle...)

For each edge $(i,j) \rightarrow (k,l)$ of the Eulerian tour, we use the an *L-shaped* route with the *horizontal arc first:*

$$(i,j) \rightarrow (i,l) \rightarrow (k,l)$$

Since each vertex will have $p/2$ horizontal arcs leaving it and $p/2$ vertical arcs entering it, we can route the entire chip with $pn/2$ tracks in each row or column.

To prove the first corollary ($p = 1$), we group the cells into 2×2 squares and apply the above construction for $p = 4$.

Then we may need to introduce small (length 1) jogs within each square to get the two horizontal arcs leaving on different rows. This we can do with only one extra track for each row or column, yielding $\lceil n/2 \rceil + 1$ tracks at most. Here each L-shaped route may have a little tail at each end so a net may have three turns total. ∎

Proof of Theorem 6: Provably Good Routing

Theorem 6. There is a polynomial time approximation algorithm achieving

$$w(R) \leq O(w(P) \cdot \log(\frac{pn}{w(P)}))$$

for any problem instance P.

Proof: (Sketch): Let $cut(P)$ denote the maximum, over all subsquares of the $n \times n$ array, of the number of nets which must cross the border of the square, divided by the perimeter of that square. It is easy to see that $cut(P)$ is a lower bound on $w(P)$.

Divide the chip into squares whose sides have length $\lambda = cut(P)/p$. Route these squares independently, in an arbitrary 1-turn manner in width at most $O(cut(P))$, routing nets that must leave a square arbitrarily to a point on the perimeter of that square. Then proceed through n/λ levels of bottom-up recursion, at each level pasting together four squares from the previous level in a 2×2 pattern, and using at most $O(cut(P))$ additional width to route all nets that leave the newly constructed square to the perimeter of that square. ∎

Proof of Theorem 7: Improved Lower Bound

Theorem 7. If $n \equiv 2 \pmod 4$ or $n \equiv 3 \pmod 4), w(n) \geq \lfloor \frac{n}{2} \rfloor + 1$.

Proof: Let $f = \lfloor \frac{n}{2} \rfloor$ and $c = \lceil \frac{n}{2} \rceil$. Consider dividing the chip as shown in Figure 7 into four quadrants A, B, C, D, where A is $f \times f$, B and C are $f \times c$, and D is $c \times c$.

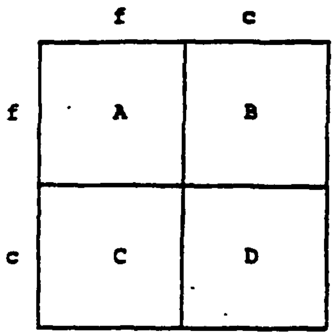


Figure 7

Consider a problem instance where each pin of A is to be connected to a corresponding pin in D, and each pin in B is connected with a pin in C. (If $c > f$, the remaining pins in D can be left unattached, or paired off.) Since $|A| = f^2$ is odd, at least $\lceil f^2/2 \rceil$ of the wires from A must run through B (without loss of generality – the case for C is symmetric). Thus the perimeter of B will be crossed at least $(f^2+1) + fc$ times: (f^2+1) times for the A-D nets and fc times for the B-C nets. Since the perimeter of B is crossed by only $f + c$ channels (rows or columns), at least one of these channels must contain at least

$$\lceil \frac{(f^2+1)+fc}{f+c} \rceil = \lceil f + \frac{1}{f+c} \rceil = f + 1 = \lfloor \frac{n}{2} \rfloor + 1$$

wires. ∎

Proof of Theorem 8: Good Two-Turn Routings

Theorem 8. $w_2(n) \leq \lfloor \frac{n}{2} \rfloor + 1$.

Proof: This is similar to the proof of the first corollary to Theorem 5, except that we group the cells regularly into 1×2 rectangles instead of 2×2 squares. The Eulerian theorem is applied as before. Finally, the L-shaped routings obtained will have to have at most one tail added to produce the final routing. When n is even it is easy to arrange the tails without increasing the number of tracks required per channel by more than one. When n is odd the argument is a little more delicate. Consider labelling each pin either "H" or "V" according to whether the route determined by the Eulerian tour would connect to that pin with a horizontal or vertical segment. Figure 8 shows a labelling that might result for a problem with $n = 7$.

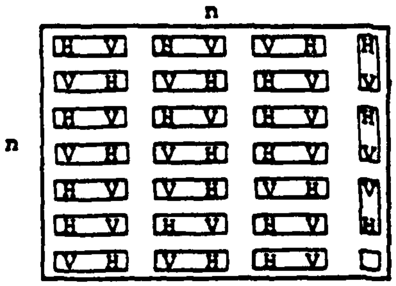


Figure 8

We are guaranteed that each 1 × 2 rectangle contains one "H" and one "V" by the use of the Eulerian tour, and we need to guarantee that each row has at most $\lfloor \frac{n}{2} \rfloor + 1$ "H"s and that each column has at most $\lfloor \frac{n}{2} \rfloor + 1$ "V"s. The rows are already OK, if the tiling pattern is like that of Figure 8. To adjust the columns we note that by running a short tail within a rectangle we can effectively move a V "on top of" its neighboring H. We can do this safely only in rows which have a "V" in the rightmost column; otherwise the tail might increase the required channel width. However, there are $\lfloor \frac{n}{2} \rfloor$ in the rightmost column, so we can always move as many as $\lfloor \frac{n}{2} \rfloor$ V's out of any column into a neighboring one. Thus we can use the tails to guarantee that no column will have more than $\lfloor \frac{n}{2} \rfloor + 1$ V's. ∎

Open Problems

We present here some open problems related to the above results. (We hope to be able to answer some of them in our final paper.)

Open Problem 1: Is there a constant c and a polynomial-time global routing algorithm A such that A will produce for any problem instance P a routing R with $w(R) \leq c \cdot w(P)$ (i.e. a routing whose width is within a constant factor of optimal)?

Open Problem 2: What is '$\frac{w_t(n)}{w(n)}$ for any fixed t? Is there a fixed t for which this ratio equals 1 for all n?

Open Problem 3: Can the logarithmic factor in the running time of the Rounding Algorithm be eliminated?

Open Problem 4: What are other applications of the Rounding Algorithm? (We do know of some ways of applying the algorithm for global routing applications that are more general than the techniques given in this abstract. We suspect that the algorithm may have a large number of useful applications.)

Open Problem 5: Let $cut(P)$ be defined as in the proof of Theorem 6. Is there a constant c such that $w(P) \leq c \cdot cut(P)$ for all problem instances P? (Note: we can prove that a similar measure computing wire-length within subsquares is linearly related to the *cut* measure.)

Open Problem 6: Can the additive "$+p$" term be improved in the second corollary to theorem 5?

Open Problem 7: Develop, empirically or otherwise, a good model of the wiring problem instances that arise in practice.

References

[BL83] Bhatt, S. N. and F. T. Leighton, "A Framework for Solving VLSI Graph Layout Problems," *Journal of Computer and System Sciences*, to appear.

[BP83] Burstein, M., and R. Pelavin, "Hierarchical Wire Routing," IBM T. J. Watson Research Center Report RC9861 (March 1983).

[CFKNS77] Chen, K. A., M. Feuer, K. H. Khokhani, N. Nan, and S. Schmidt, "The Chip Layout Problem: An Automatic Wiring Procedure," *Proceedings 14th Design Automation Conference* (IEEE, New Orleans, 1977), 298 - 302.

[EG81] El Gamal, A. A., "Two Dimensional Stochastic Model for Interconnections in Master Slice Integrated Circuits," *IEEE Trans. on Circuits and Systems* CAS-28 (February 1981), 127-138.

[HN83] Hong, S. J., and R. Nair, "Wire Routing Machines – New Tools for VLSI Physical Design," *Proc. IEEE* 71 (January 1983), 57 - 65.

[J82] Johnson, D. S., "The NP-Completeness Column: An Ongoing Guide," *Journal of Algorithms* 3 (1982), 381-395.

[L80] Leiserson, C. E., "Area-Efficient Graph Layouts (for VLSI)," *Proc. 21st FOCS Conference*, (IEEE, October 1980), 270-281.

[L81] Leighton, F. T., "New Lower Bound Techniques for VLSI," *Proc. 22nd FOCS Conference*, (IEEE, October 1981), 1-12.

[L82] Leighton, F. T., "A Layout Strategy for VLSI Which is Provably Good," *Proc. 14th ACM STOC Conference*, (ACM, May 1982), 85-98.

[T79] Thompson, C., "Area-Time Complexity for VLSI," *Proc. 11th ACM Conference*, (ACM, May 1979), 81-88.