

AD A138339

VLSI RESEARCH

SEMI-ANNUAL
TECHNICAL R&D STATUS REPORT
APRIL - OCTOBER 1983

PRINCIPAL INVESTIGATOR

R.W. BRODERSEN
(415) 542-1779

FACULTY RESEARCHERS

R.W. BRODERSEN
N. CHEUNG
A. DESPAIN
D.A. HODGES
C. HU
D. MESSERSCHMITT
R.S. MULLER
A. NEUREUTHER
A.P. NEWTON
W.G. OLDHAM
J. OUSTERHOUT
D.A. PATTERSON
A. SANGIOVANNI-VINCENTELLI
C. SEQUIN

DTIC

FEB 27 1984

This document has been approved
for public release and sale; its
distribution is unlimited.

SPONSORED BY
DEFENSE ADVANCED RESEARCH PROJECTS AGENCY
ARPA ORDER NO. 4631

MONITORED BY NAVAL ELECTRONIC SYSTEMS COMMAND
UNDER CONTRACT NO. N00039-83-0107

The views and conclusions contained in this
document are those of the authors and should
not be interpreted as representing the official
policies, either expressed or implied, of the
Defense Advanced Research Projects Agency or
the U.S. Government.

DTIC FILE COPY 84 01 13 023

ELECTRONICS RESEARCH LABORATORY
College of Engineering
University of California, Berkeley, CA 9472

VLSI RESEARCH
SEMI-ANNUAL
TECHNICAL R&D STATUS REPORT
APRIL - OCTOBER 1983

PRINCIPAL INVESTIGATOR

R.W. BRODERSEN
(415) 642-1779

FACULTY RESEARCHERS

R.W. BRODERSEN
N. CHEUNG
A. DESPAIN
D.A. HODGES
C. HU
D. MESSERSCHMITT
R.S. MULLER
A. NEUREUTHER
A.P. NEWTON
W.G. OLDHAM
J. OUSTERHOUT
D.A. PATTERSON
A. SANGIOVANNI-VINCENTELLI
C. SEQUIN

Executive Overview

This report covers the period from April to October 1983 on contract No. N00039-C-0107. A few of the highlights of this report follow.

In our architecture research a 41,000 transistor second generation 32 bit processor, which demonstrates the advantages of the reduced instruction set concept, has been successfully fabricated and tested. In addition a 48,000 transistor instruction cache was also found to be fully operational and it incorporated a new redundancy idea that tripled the yield. A new cad tool, a timing verifier called Crystal, was found to be extremely useful as it detected potential performance mistakes during the design process.

In the computer aids for design and layout research, 85 copies of the new tools tape have been distributed to university and industrial labs in the U.S. This tape contains about 25 programs, including several new programs (Lyra, Crystal, Peg and Tpack). A new improved version of the Waveform-Relaxation based simulator has been developed which utilizes new techniques to speed up convergence and for error control. A new algorithm called BLOSSOM has been developed, which can exploit the parallelism of VLSI for the solution of large-scale linear systems of algebraic equations.

The circuitry contained on one Multibus card has been developed to be able to recognize up to 1000 words of speech in real time. This board will have the equivalent of 118 MIPS of von Neumann equivalent operations. The algorithm uses dynamic time warping, which is the most successful of techniques at this time. The board has sufficient flexibility to allow considerable algorithmic enhancements, which should make it extremely useful for further developments in speech recognition research. The board uses two custom circuits, one of which was designed fully automatically from a high level description requiring no layout from the algorithm developer.

Models have been developed which explain the operation of transistors which have been scaled to near the ultimate limits in oxide thickness. The reliability of these devices has been investigated as well as performance degradation from hot and tunneling electrons.



letter on file

A-1

1. ARCHITECTURE

1.1. Testing Second Generation RISC's (D. Patterson)

The RISC II processor and the instruction cache chip have been received from fabrication and have been tested.

RISC II [1], a 32-bit NMOS microprocessor using 41,000 transistors, is an improved version of the RISC I chip. It is 25% smaller than its predecessor even though it has 75% more registers using the same design rules. The sharing of the bit lines for reading and writing, which made this size reduction possible, required, however, an extra pipe stage plus operand forwarding circuits.

Like RISC I, RISC II worked on the first silicon. This time, however, the performance was close to what we predicted in part because of careful design and extensive Spice simulation of critical data-path delays, and in part because of Crystal, a timing verifier developed by John Ousterhout. Crystal was used to find the time critical paths and to verify that the less regular parts of the circuitry (e.g. control) were matched in speed to the highly optimized data path. The predicted RISC II cycle time (i.e. execution of a register-to-register instruction) was 480 nsec. In the lab RISC II chips run at 500 ns per instruction (VDD=5V, VBB=VSS=0V, room temperature). The average power consumption is 1.25 Watts, a little less than we had anticipated.

Benchmark simulations showed that a 500 nsec RISC II runs integer C programs faster than a 3-MHz IAPX-286, 10-MHz NS 16032, 12-MHz 68000, or 18-MHz HP 9000 CPU.

We also tested the RISC II instruction cache [2], a 48,500 transistor NMOS chip (the largest chip we have built so far). The instruction cache also worked on first silicon. Again, Ousterhout's Crystal program was used to verify the timing of the cache, and again its usefulness was proven. It uncovered one performance mistake - the ratio on one gate was 4/1 instead of 1/4 - which would have stretched a 70 ns clock phase to 700 ns. When testing the corrected design, the fastest chip was found to have a 250 ns access time with a 480 ns cycle time (VDD=5V, VBB=VSS=0V, room temperature), comparing favorably with the projected 400 ns time. The instruction cache is thus compatible with the 500 ns RISC II CPU. The average power consumption was 1 Watt, less than we expected.

There were several new ideas in the cache chip, including one to improve yield. Caches already have a bit per cache block, used on power up, that indicates invalid data. We added an extra invalid bit to put blocks into a permanent "invalid" state if they were found defective. We tested 13 chips: the new idea tripled the yield.

In the last 6 months several papers sponsored by DARPA have been republished [3] [4] [5] [6]. We also saw the announcement of the Pyramid minicomputer, the first commercial computer to use ideas we developed at Berkeley.

References

- (1) M.G.H. Katevenis, R.W. Sherburne, D.A. Patterson, and C.H. Séquin: "The RISC II Micro-Architecture," *VLSI '83 Conference*, Trondheim, Norway, (August 1983).
- (2) D.A. Patterson, P. Garrison, M. Hill, D. Lioupis, C. Nyberg, T. Sippel, and K. Van Dyke: "Architecture of a VLSI Instruction Cache for a RISC," *Tenth*

Annual Symposium in Computer Architecture, June 14-15, 1983, pp. 108-118.

- (3) "Microprogramming." D.A. Patterson: *Scientific American Offprint 3132* W.H. Freeman and Company, New York, N.Y., 1983. (Appeared originally in *Scientific American*, Vol. 248, No. 3, March, 1983, pp. 36-43.)
- (4) P. Hansen, A. Linton, R. Mayo, M. Murphy, and D. Patterson: "A Performance Evaluation of the Intel iAPX 432," *Advanced Microprocessors*, (Edited by Amar Gupta and Hoo-min Toong) IEEE Press, New York, N.Y., 1983, pp. 242-252. (Appeared originally in *Computer Architecture News*, vol. 10, no. 4, June 15, 1982, pp. 17-26.)
- (5) D.A. Patterson and C.H. Sequin: "Design Considerations for Single-Chip Computers of the Future," *Advanced Microprocessors*, (Edited by Amar Gupta and Hoo-min Toong) IEEE Press, New York, N.Y., 1983, pp. 289-278. (Appeared originally in *IEEE Journal of Solid-State Circuits*, Vol. SC-15, no. 1, pp. 44-52, *IEEE Transactions on Computers*, Vol. C-29, no. 2, pp. 108-118, Joint Special Issue on Microprocessors and Microcomputers, February 1980.)
- (6) C.H. Sequin and D.A. Patterson: "Design and Implementation of RISC I," *VLSI Architecture* (Edited by B. Randell and P.C. Treleaven) Prentice Hall International, Hertfordshire, England (June 1983). (Appeared originally in *Proceedings Advanced Course on VLSI Architecture*, Univ. of Bristol, England (October 1982).)
- (7) Y. Tamir and C.H. Sequin: "Strategies for Managing the Register File in RISC", scheduled for publication in *IEEE Trans. on Computers*, Nov. 1983.

1.2 VLSI Communication Components For Multiprocessors (C. Sequin)

We have studied an approach to supercomputers that is particularly suited for implementation with VLSI technology. It relies on a large pool of single-chip computers that operate concurrently on many subtasks of a complex problem. Multiprocessors, in the past, have often fallen short of the expected performance because of a lack of interprocessor communication bandwidth. As the number of computers in the system is increased from a few tens to several hundreds, communication will become even more of a bottleneck unless the system is planned in such a manner that the available total bandwidth grows automatically with the number of processors in the system. This research produces guidelines for the construction of VLSI communication components for this framework.

The approach studied in detail concerns a "communications domain" of arbitrary topology created from VLSI components linked by high-speed dedicated links. It is shown that the power limitations of the individual chips make it advantageous to concentrate all available output bandwidth into a few ports with maximum bandwidth. These components would also have the built-in facilities for low-level functions such as handshaking, buffering, and flow control, so that the information packet or the block of data to be transmitted is the lowest primitive the systems builder needs to be concerned about. The design of such modular VLSI communications components is being investigated, cost-estimates in terms of the number of devices needed for their implementation are performed, and the trade-offs in the various design parameters are analyzed.

1.3 Novel, High-Performance Architectures (S. Baden, A. Despain)

The first phase of our investigation into the use of functional languages for array processing culminated in a presentation at COMPCON '83 [3]. At COMPCON we reported on our experiences with our own implementation of Rackus' functional language, FP [1]. (Berkeley FP is now available on the latest release of Berkeley UNIXTM [2]). We found that while FP's *Program Forming Operations* (PFO's) were a powerful facility for expressing concurrency, their generality was severely diminished owing to the lack of user-definable PFO's. As a result, user-defined program building blocks could not be re-used for applications related their original purpose. FP programs tend to be data structure intensive owing to a lack of data abstraction facilities, i.e., the programmer must pay undue attention to the low-level representation details. Since FP's semantics are strict, all intermediate results must be fully evaluated before being passed on to their enclosing expressions. This is unfortunate, as sometimes it is convenient to defer a complete evaluation: either to allow for infinite streams of data, e.g. I/O, (see for example Henderson and Morris' work on lazy evaluation [9]) to admit infinite expressions into the language [13] (a powerful programming technique), or to exploit some well-known optimization techniques (see Guibas and Wyatt's treatment of delayed evaluation in APL [8]). Finally, we found that FP's ambiguous treatment of errors coupled with its strict semantics made exception handling impossible and debugging difficult at best.

We have developed our own functional language, called XBA, to provide the abstraction and error handling facilities that a functional language must have to be practical. XBA has strict semantics, lending itself to a lazy implementation. The language is based in part on Turner's functional language KRC [13]. All information is treated uniformly in XBA; hence, the programmer may define his

own PFO's (they are just functions of functions) and error values may be manipulated as, say, numbers would be. A simple data typing facility is provided: Pascal-like structural type definitions and optional type declarations (XBA is not strongly typed but declarations help the compiler generate faster code). In addition, PROLOG-like argument pattern matching is provided.

We have found that XBA is better suited to writing concurrent software than FP owing to its abstraction facilities and to its uniform treatment of information. As a result machine independent code is easier to write and software is generally reusable. In addition, the adoption of non-strict semantics lends itself to a natural treatment of I/O and to optimizations that avoid temporary storage, unnecessary computations, or both.

As we have compiled XBA we have investigated applicative instruction sets based on the work of Turner and others [12,4]. Turner has reported on a combinatory representation for KRC programs, i.e., one that contains no bound variables. Besides constituting the first truly applicative instruction set the representation also brings into being the notion of compile-time in the implementation of functional languages (These languages tend to be executed interpretively). The combinatory representation blurs the distinction between compile time and run time; indeed, some function calls can be executed at compile time to reduce the number of run time procedure calls (traditionally, this is done with macros). Also, combinatory code is *dynamically self-improving* i.e., on-the-fly code generation is unnecessary. We conducted an extensive (and as of yet, unpublished) survey on the topic, concluding, that like data-flow implementations, combinatory ones impose a prohibitive overhead on task partitioning and scheduling, particularly on the sorts of array computations in which we are interested. Although some of this overhead can be removed using newly published data flow analysis techniques [10] it will be necessary to provide array facilities as primitives if such an implementation strategy is to be at all feasible. That brings us to our present research effort.

Currently we are investigating the design and implementation of array facilities for XBA. In this context XBA will be used as an abstract notation to describe the semantics of the facilities, rather than as a concrete language. This approach ensures that our results will be applicable to a wider variety of implementation styles than if we chose a concrete syntax for the language (e.g., the facilities could be implemented as ADA packages and a special preprocessor).

The use of the proposed array facilities, coupled with the availability of PFO's, encourages the programmer to favor *stylized*, e.g., patterned array accesses over random, low-level ones. This level of data access lends itself to a simple, efficient implementation strategy for generating and manipulating memory access patterns (see for example the work of Guibas and Wyatt on compiling APL [9]). In the first case, computations may be partitioned amongst replicated resources rapidly, and, in the second, data access patterns may be tuned to take advantage of particular hardware structures (e.g., vector registers in the Cray), obviating machine-dependent code. The memory system design supports multidimensional parallel accesses (see Lawrie and Vora's work on the prime memory system [11]) and localized address computations. In toto, the combination of low-overhead task partitioning and decentralized address computations make applicative architectures (e.g., either reduction [5] or dataflow [6]) much more attractive for array processing than they have been in the past [7].

Our final goal is to provide a complete specification of the array facilities along with the memory system design. The design of the array facilities will take precedent over the memory system design, so simple assumptions will be made

concerning the design of the processor-memory interconnection network. The final design will be evaluated through simulation to determine, for instance, the size and numbers of memories. We are meeting regularly with mathematicians and computational physicists to determine a reasonable set of benchmark programs with which to drive those simulations.

In sum, we have determined the appropriate features required of a practical functional language for array processing: concurrency, abstraction facilities, error recovery mechanisms, and non-strict semantics. We have developed a language that meets these requirements and an implementation strategy to overcome the "von Neumann bottleneck."

References

- (1) John Backus, "Can Programming be Liberated from the von Neumann Style? A Functional Style and its Algebra of Programs," *Comm. ACM* 21, 8 (Aug. 1978), 813-841. 1977 ACM Turing Award Lecture.
- (2) Scott B. Baden, *Berkeley FP User's Manual*, Rev. 4.1, Univ. of California, Berkeley, CA, Dec. 1982.
- (3) Scott B. Baden and Dorab R. Patel, "Berkeley FP - Experiences with a Functional Programming Language," *Conf. Rec. COMPCON '83*, San Francisco, Mar. 1983.
- (4) F. Warren Burton, "A Linear Space Translation of Functional Programs to Turner Combinators," *Info. Process. Letters* 14, 5 (23 Jul. 1982), 201-204.
- (5) John Darlington and Mike Reeve, "ALICE: A Multi-Processor Reduction Machine for the Parallel Evaluation of Applicative Languages," *Proc. Conf. Functional Languages and Computer Architecture*, Portsmouth, NH, 1981, 65-75.
- (6) Jack B. Dennis, Guang-Rong Gao and Kenneth W. Todd, "A Data Flow Supercomputer," MIT Laboratory for Computer Science, Mar. 1982.
- (7) D. D. Gajski, D. A. Padua and D. J. Kuck, "A Second Opinion on Data Flow Machines and Languages," *Computer*, Feb. 1982, 58-69.
- (8) Leo J. Guibas and Douglas K. Wyatt, "Compilation and Delayed Evaluation in APL," *Proc. 5th POPL*, 1978.
- (9) Peter Henderson and James Morris, "A Lazy Evaluator," *Proc. 3rd POPL*, Jan. 1978, 95-103.
- (10) Kent Karlsson, "An Outline of the Sky Reduction Machine," Report # 1981-02-03, Dept. of Comput. Sci., Univ. of Goeteborg, Goeteborg, Sweden, 1982.
- (11) D. Lawrie and C. Vora, "The Prime Memory System for Array Access," *Proc. Int'l. Conf. on Parallel Processing*, Aug. 1980, 81-87.
- (12) David A. Turner, "A New Implementation Technique for Applicative Languages," *Software - Practice and Experience* 9, (1979), 31-49.
- (13) D. A. Turner, "Recursion Equations as a Programming Language," in *Functional Programming and its Applications*, J. Darlington, P. Henderson and D. A. Turner (ed.), Cambridge Univ. Press, Cambridge, 1982.

1.4. Multiprocessor Circuit Simulation (D. G. Messerschmitt)

Code generators for LU decomposition in a SPICE circuit simulator running on a multiprocessor architecture have been completed. These have been run on a multiprocessor simulator, and their performance evaluated. As expected, because of the I/O bound nature of this computation, the performance was found to degrade significantly as the interprocessor communication delays increase. Subsequent work has concentrated on improving the performance by designing scheduling algorithms which take into account interprocessor communication delays. Two types of heuristic algorithms have been developed: local optimization and global optimization. While the latter take significantly more computer time to execute, and may therefore not be practical in a production circuit simulation environment, they show approximately double the speedup of the local optimization algorithm and thus serve as a basis of comparison. A Ph.D. thesis and a couple of papers are in preparation describing this work, and will be available in the next quarter.

An examination of digital filtering in a parallel computational environment has yielded a significant and surprising result. The goal has been to find an implementation approach for digital filters which would have the property that an arbitrarily high sampling rate could be achieved with a fixed speed of hardware by applying parallelism with only local interconnection. It is obvious how this can be done for non-recursive filters, but the surprising fact is that it can also be achieved for recursive filters, where the feedback destroys speedup by pipelining. A Ph.D. thesis and paper are also under preparation on this topic. Work is continuing on understanding how practical constraints (such as pin I/O limitations) limit the sampling rate.

Processor interconnection topology and routing algorithms are continuing to be pursued. Automatic generation of topology from traffic statistics has been implemented using a clustering algorithm. Measures of interconnection hardware complexity and speed performance are being developed in order to evaluate alternate interconnection topologies. Routing algorithms which efficiently exploit a given topology are also being investigated.

In cooperation with Professors Newton and Sangiovanni, ways of obtaining a hardware multiprocessor machine for experimental work are being pursued. This is important, as we expect that not all practical constraints can be uncovered by software simulations.

2. COMPUTER AIDS FOR DESIGN AND LAYOUT

2.1. 1983 VLSI Tools Distribution (J. Ousterhout)

Our new tools tape went into distribution on April 1. Since then, approximately 85 copies of the tape have been sent to university and industrial labs in the U.S. The new tape contains about 25 programs, including several new programs (Lyra, Crystal, Peg, and Tpack) as well as updated versions of older programs such as Caesar, Cifplot, and Mextra.

2.2. Crystal, Version2 (J. Ousterhout)

During the spring and summer of 1983, most of Crystal was re-written. The algorithms in the new version are simpler, cleaner, faster, and also more general than the version 1 algorithms (experience is a wonderful teacher!). Whereas version 1 had built-in notions about transistor types and could only handle nMOS, version 2 is table driven, so that users can define new transistor types. Version 2 has already been used for both nMOS and CMOS designs, and is slightly faster than version 1. In the summer of 1983, work was begun to upgrade the transistor models to include second-order effects due to waveform shape. An initial implementation has just been completed, but its accuracy has not yet been tested.

2.3. Caddy—A New IC Layout System (J. Ousterhout)

We have undertaken the development of a new IC layout system called Caddy. The system has three overall goals, based on problems experienced with our earlier systems. The first goal is to integrate design rule and circuit information into the layout editor in order to provide incremental design rule checking and circuit extraction. This additional expertise will permit interactive compaction and stretching of layouts. The second goal is to move away from fabrication details by eliminating the need for designers to specify implants and wells and contact details explicitly. These layers will be generated automatically by the system (the result is much like "sticks" except that it is fleshed out). The third goal is to provide interactive semi-automatic routing aids. In this respect, our goal is not to invent new algorithms and paradigms, but to find powerful ways of embedding existing techniques into an interactive design environment.

Initial discussions were held in the Spring and Fall of 1982, during which the underlying data structures and algorithms ("corner stitching") were developed. Between January and April of 1983 the basic structure of the system was designed and implementation was begun. A bare-bones system with about the functionality of Caesar (but with corner-stitching as the underlying data structure) became operational in early April 1983, and has been used since then in the layout of the nMOS SOAR microprocessor. During the spring and summer, design was completed for compaction, stretching, design-rule checking, routing, and multi-window. The window facilities are now in the final stage of debugging; compaction, stretching, and design-rule checking are partially implemented; and the routing implementation (based on an extended version of

Rivest's greedy channel router) is just beginning.

References

- (1) R.N. Mayo, J.K. Ousterhout, and W.S. Scott, eds: "1983 VLSI Tools," Technical Report No. UCB/CSD 83/115, Computer Science Division, University of California, Berkeley, 1983.
- (2) J.K. Ousterhout: "Corner Stitching: A Data Structuring Technique for VLSI Layout Tools," *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, to appear.
- (3) J.K. Ousterhout: "The User Interface and Implementation of Caesar," Technical Report No. UCB/CSD 83/131, Computer Science Division, University of California, Berkeley, 1983.

2.4. Finite-State Machine Synthesis (R. Newton, A. Sangiovanni-Vincentelli)

Sequential circuits play a major role in the control part of digital systems. We addressed the automated synthesis of sequential logic functions in a structured VLSI design methodology. We considered sequential logic functions implemented by synchronous deterministic Finite State Machines (FSM) consisting of two distinct components: a combinational circuit implemented by a Programmable Logic Array (PLA) and a memory implemented by Delay-type registers.

In particular we considered the problem of assigning binary codes to the internal states of a Finite State Machine. The literature is rich in papers dealing with the state-assignment problem. Here we refer to the major approaches only. Armstrong introduced a set of criteria for encoding states, aiming at the minimization of the number of gates used to implement the FSM and formulated the encoding problem as a graph embedding problem. Hartmanis, Stearns and Karp developed algebraic methods based on partition theory and on a reduced dependence criterion. Dolotta and McCluskey suggested a "column-based" procedure to code states. Note that despite these efforts, to the best of our knowledge no tool for designing FSM is in use today for a time-effective state encoding of industrial digital controllers.

Armstrong's approach can in principle handle rather large machines, but it has three serious drawbacks. The first is related to the fact that the criteria suggested by Armstrong do not take into account the techniques of fast heuristic logic minimizers such as MINI, PRESTO, or ESPRESSO-II in use today (Armstrong's paper appeared before the work on heuristic minimizers started). The second is that the state-assignment problem is transformed into a particular graph-embedding problem, which represents only partially the state coding problem. The third is that the graph embedding algorithm suggested by Armstrong was ineffective.

Our approach is based, as Armstrong's, on the use of distance relations among the codes of the internal states. We showed in [1] how the combinational logic can be reduced by requiring state codes to satisfy appropriate distances. Distance requirements are determined by predicting the effects of heuristic minimization of the combinational logic related to a symbolic description of the FSM, and are represented by a graph. In particular it is shown that a convenient reduction of the combinational logic is obtained if the distance between some state codes is large enough and appropriate states have adjacent codes.

We considered the problem of assigning codes which satisfy the distance relations. Adjacent code assignment can be seen as an embedding of an adjacency graph into a boolean hypercube. Armstrong and Saucier represented the state assignment problem as a subgraph isomorphism problem, where a one-to-one relation (coding) is sought between the set of the states (vertices of the adjacency graph) and a subset of the boolean hypercube vertices (codes). Note that even questioning the existence of a subgraph isomorphism is a hard problem: in particular it was shown to belong to the class of NP-complete problems. Since such an isomorphism may not exist, Armstrong and Saucier relaxed some adjacency requirements and proposed heuristic techniques to embed a subgraph of the adjacency graph into the boolean hypercube. Note that a distance-preserving embedding is not even guaranteed by augmenting the dimensions of the hypercube, i.e. increasing the length of the state codes.

Our approach exploits the use of dc conditions in state codes. In particular every state is coded by associating each vertex of the adjacency graph to a subcube of the boolean hypercube. This is equivalent to embed the adjacency graph into a squashed hypercube, i.e. a hypercube having appropriate faces squeezed

into vertices. Note that most of the state assignment techniques presented in the literature obtained a state coding using the minimum number of bits, because it was important to minimize the number of memory elements due to their cost. On the other hand, the area taken by the PLA is the major concern in a VLSI circuit implementation of a Finite State Machine. Minimal area PLA implementations of the FSM combinational component can be obtained by using non-minimal-length state codings i.e. fewer product-terms are often required to implement a logic function at the expense of an increased number of input/output columns. Therefore we allow non-minimal-length state codings when leading to minimal area PLAs. In this case, state coding corresponds to an embedding into a squashed hypercube of variable dimension. However bounds on code length can be enforced when required by a particular implementation.

Experimental results obtained using this approach on a number of industrial machines has been satisfactory. However, very recently a new technique based on multiple-valued logic minimization and on the ideas described above has been obtained in collaboration with Dr. R. Brayton of IBM. This technique has been very successful in coding Finite State Machines efficiently, improving over manual techniques and over the technique presented above by a sizable amount. We are presently studying its implications and developing new embedding algorithms.

References

- (1) G. De Micheli, A. Sangiovanni-Vincentelli, T. Villa, "Computer-aided Synthesis of Finite-State Machines" *Proc. of Int. Conf. on CAD 1983*, Santa Clara, CA Sept. 1983.

2.5. Relaxation Based Circuit Simulation (R. Newton, A. Sangiovanni-Vincentelli)

Recently, a new class of algorithms has been applied to the electrical IC simulation problem. New simulators have been developed at Berkeley (RELAX and RELAX2, SPLICE1.6 and SPLICE2) that use these methods and provide as accurate, or more accurate, waveforms than standard circuit simulators such as SPICE2 or ASTAP with up to two orders of magnitude speed improvement for large circuits. These simulators have been used for the analysis of both digital and analog MOS ICs. They use *relaxation* methods for the solution of the set of ordinary differential equations, ODEs which describe the circuit under analysis, rather than the direct, sparse-matrix methods on which standard circuit simulators are based.

During this period, we studied the numerical properties of the various methods for the analysis of MOS circuits and we presented them in a rigorous and unified framework in [1] and we improved our relaxation algorithms and their implementation in [2].

Relaxation-based Electrical Simulation has been written to provide a complete picture of the new methods for circuit analysis and we expect it to become the standard reference for new work on circuit simulation. Both the advantages and the limitations of these techniques for the analysis of large ICs are described. Some of the fundamental problems associated with conventional circuit simulation algorithms as circuit size increases are exposed and the mathematical basis for the relaxation approach is introduced. The special relaxation methods called *timing simulation* algorithms are described and their numerical properties are investigated. *Iterated Timing Analysis*, which applies relaxation techniques at the nonlinear equation level, is described and its convergence properties are proven. The *Waveform Relaxation* method, which applies relaxation techniques at the differential equation level, is presented and various techniques which can be used to improve its performance for electrical simulation are described. Future research directions including the use of special purpose hardware for the implementation of these algorithms are presented.

In [2], we described RELAX2, a new improved version of RELAX, a Waveform-Relaxation based simulator. In particular, we were able to characterize the convergence behaviour of the Waveform Relaxation Method on a class of circuits which required a large number of iterations to converge. The study of the convergence behaviour has led into the concept of "windowing", i.e., of breaking up the time interval over which analysis has to be performed, in sub-intervals so that the algorithm applied in these sub-intervals exhibits fast convergence. In addition, techniques for the adaptive error control of Waveform Relaxation have been tested and improved.

References

- (1) R. Newton and A. Sangiovanni-Vincentelli, "Relaxation-Based Electrical Simulation", *IEEE Trans. on Elec. Dev.*, to appear, *SIAM Journ. on Scientific and Statistical Computing*, to appear, *IEEE Trans. on CAD of IC and Syst.*, to appear.
- (2) J. White and A. Sangiovanni-Vincentelli, "RELAX2: A New Waveform-relaxation Approach for the Analysis of MOS LSI Circuits" *Proc. of Int. Symp. on Circ. and Syst.*, Newport Beach, Ca., May 1983, invited paper.

2.6. Special Purpose Architectures For The Solution of Large Scale Systems (R. Newton, A. Sangiovanni-Vincentelli)

The solution of Large-scale linear System of algebraic Equations(LSE) is needed in the analysis and simulation of many engineering systems.

New architectures, in particular vector computers such as the CRAY 1, have inspired the design of new algorithms to exploit parallelism in the solution process. An important example is the program CLASSIE for the simulation of electronic circuits. Along these lines, peripheral array processors, such as the FPS164, can also be used in conjunction with hosts such as the VAX11/780 to speed up the solution process. However, this speedup is not enough to cope with the problems to be solved in the VLSI era.

The advent of VLSI technology has made the cost-effective design of special purpose machines possible. Examples of these machines are the Yorktown Simulation Engine(YSE) for logic solution and Systolic Arrays. Special purpose machines have also been proposed for the solution of LSE. Most of these machines limit the size of the operand matrix. When no size limit is imposed, the operand matrix has to be partitioned into submatrices of equal sizes. Only Johnsson and Pottle treated the related numerical properties. However special matrix structures, such as the Bordered Block Diagonal Form (BBDF) or the Bordered Block Triangular Form(BBTF), commonly expected in engineering problem, have not been exploited. In [1], we proposed a new algorithm-architecture BLOSSOM for the solution of LSE.

This architecture supports other matrix operations used as subprocedures by block LU decomposition such as the multiplication and the inversion of submatrices. We described the hardware implementation of these matrix operations.

References

- (1) H. Ko and A. Sangiovanni-Vincentelli, "BLOSSOM: an Algorithm and Architecture for the Solution of Large-Scale Linear Systems" *Proceedings of the International Conference on Computer Design*, New York, Oct. 1983.

2.7. Incremental Wire Manipulation For VLSI Layout (C. Sequin)

Existing wire-based layout manipulation systems have tended to do operations such as compaction on a global basis, compacting an entire layout or module at a time. WICRD (Wire Incremental Compaction, Routing, and Displacement) is an experimental prototype system to explore algorithms for efficient incremental changes to an existing layout, while still allowing these global operations to proceed efficiently. The WICRD user model envisions a user sitting at a graphics terminal, making many small incremental changes to the layout and only occasionally invoking more global operations. A typical WICRD operation would be moving a single wire segment a short distance, changing the position of as few other objects as possible while maintaining the layout topology and connectivity. One of the conclusions demonstrated by WICRD is that global compaction may be viewed as a special case of incremental displacement, using the same conceptual framework and algorithms.

One of the issues investigated in WICRD is that of wire representation. Three classes of wire representations were investigated: connected skeletons, fleshed-out geometry, and directed finite-width wire segments. Skeletal representations model a wire as a chain of connected line segments, typically chosen to be the centerline of the wire, with attached attributes such as width and allowable separation from other wires. They correspond closely to the abstract semantics of wires, but have disadvantages when used in conjunction with circuit blocks of finite size because of their lack of explicit geometrical information.

Fleshed-out geometry models explicitly represent the physical space occupied by the wire material. They are a convenient representation for layout rule checking, but make it harder to keep track of connectivity and other semantic properties of the wiring pattern. Directed wire segments attempt to combine the best features of skeletal and fleshed representations by starting with a fleshed representation and adding explicit attributes to indicate the underlying wire structure. The wire representation implemented in WICRD uses this representation.

The various representations also lead to interesting trade-offs in the areas of automatic routing and compaction and in the user interface.

References

- (1) C.H. Sequin and R.M. Fujimoto, "X-tree and Y-components", in *VLSI Architecture*, B. Randell and P.C. Treleaven, editors, Prentice Hall, 1983, pp 299-328.

Thesis

- (2) M. Fujimoto, "VLSI Communication Components for Multicomputer Networks", PhD Thesis, U.C. Berkeley, Aug. 25, 1983.

3. INTEGRATED CIRCUIT DESIGN

3.1. A 1000 Word Speech Recognition System (R. W. Brodersen)

Circuitry contained on one multibus card will be able to recognize 1000 words in real time. On the card are 2 custom I.C.'s, an Intel 80186 microcomputer and memory. This card is in the final stages of construction.

One of the IC's is a filterbank chip which was generated fully automatically from software. This chip implements a 16 channel filterbank with a 112 poles of filtering, at an initial sample rate of 14 kHz.

The 2nd chip is an enhanced version of a previous design, which performs a dynamic programming algorithm. The new chip has more parallelism as well as a number of glue logic functions for memory control which were required to be able to fit the entire recognition system on one board.

The 186 performs the multibus interface and will be used in future research to implement such things as syntax direction, continuous speech algorithms and sophisticated training (learning) algorithms.

We are putting this card into a SUN workstation, and plan to use it to incorporate speech into a number of applications. The following two sections describe the two special purpose chips in more detail.

3.2. Dynamic Programming Integrated Circuit (R.W. Brodersen)

An integrated circuit capable of performing the pattern matching required to recognizing 1000 words of speech in real-time will has been designed and fabricated. The chip uses a parallel and pipelined architecture to compute the Euclidean distance between two 16 dimensional vectors (4 bits per dimension) while simultaneously executing a dynamic programming algorithm. The chip requires only a minimum amount of external circuitry: a clock, memory, and address latches.

The boundary conditions of the dynamic programming matrix can be controlled, so that the chip is compatible with both isolated and connected-word algorithms without additional hardware. No pruning or global slope constraints are used, thus each frame of the unknown word must be compared to each frame in the template word memory. Local slope constraints can be applied as a programmable option. For a one thousand word vocabulary, with 25 frames per word, at a 20ms frame rate, 25000 frames must be stored in the template memory. The processing of a single frame requires that a 16-dimensional Euclidean distance be computed, as well as a minimization. The chip can perform these computations in 800ns, thus all template frames can be processed in 20ms. The chip, therefore, not only runs in real time, but can process all unknown word frames before the endpoint is detected.

The chip has the following functional units, all running in parallel:

- (1) A distance processor that can compute a 4-dimensional Euclidean distance every clock cycle.
- (2) A pipeline accumulator that sums four 4-dimensional Euclidean distances into one 16 dimensional distance.

- (3) A dynamic programming processor that can compute one minimization every 4 clock cycles.
- (4) An addressing unit for the external template and scratch-pad memories.
- (5) A controller for each of the above processors.

The chip is implemented in a 4 micron NMOS process, has an active area of 20,000 square mils, and runs with a 5MHz clock.

3.2.1. Computer Generated Digital Filter Banks

The goal of this project is to allow the system designer to generate digital filter bank chips without performing any circuit design or layout. The circuit is entirely computer generated and requires about 5 to 10 minutes after the filter coefficients and filter organizations have been determined.

The project has been broken into two parts. First, a pipelined digital processor has been designed with a major focus on speed and small size. Also important is modularity, so that the processors can be easily assembled in different configurations with a minimum of signal routing. Second, programs have been developed to turn system information (filter types, coefficients, number of processor bits) into completed circuit layouts, ready for fabrication. To enhance flexibility, two separate programs are employed. One generates the micro-code for the processors from the filter descriptions, while the other generates the chip layout from the micro-code.

A completed 16 channel (each channel has a 4th order bandpass filter, rectifier and a 3rd order lowpass filter) operating at a 14KHz sample rate, has been designed and fabricated. Test results indicate that frequency response and dynamic range are identical to that expected. In addition, the circuits operated nearly 50% faster than required for the current application, indicating more complex systems are possible. The program for generating layouts from micro-code has been completed and chips generated by this program have been fabricated. The program for generating micro-code from filter descriptions is in development. The system now has the capability to use one or two processors and extension to allow 4 processor circuits is being added.

References

- (1) M. Lowy, H. Murviet, R. W. Brodersen, "A Large Vocabulary Speech Recognition System". *Proc. of VLSI in Modern Signal Processing USC*.
- (2) T. C. Choi, R.T. Kaneshiro, P.R. Gray, W. Jett, M. Wilcox, R. W. Brodersen, "High Frequency CMOS Switched-Capacitor Filters for Communications Applications" *1983 IEEE Intl. Solid State Circuits Conf. Tech. Digest*, pp. 246-297.
- (3) R.D. Fellman, R. W. Brodersen, "A Switched Capacitor Adaptive Lattice Filter" *Journal of Solid State Circuits*, February 1983, pp. 46-56.
- (4) T. Choi, P. R. Gray, R. Kaneshiro, R. W. Brodersen, "Circuits and Systems Paper.
- (5) H. Murviet, M. Lowy, D. Mintz, R. W. Brodersen, "An Architecture for a Speech Recognition System", *Tech. Digest of ISSCC, February 1983*, pp.118-119.
- (6) R.D. Fellman, R.J. Hurst, R. W. Brodersen, "Switched-Capacitor Circuits for Adaptive Filtering and Autocorrelation", *Tech. Digest of ISSCC, February 1983*, pp.126-127.

3.3. Rapid Design Methods for Signal Processing Circuits (R.W. Brodersen)

A rapid design system is being implemented for monolithic implementation of signal-processing systems. This interdisciplinary effort spans the fields of circuit design, CAD software development, computer architecture and digital systems design.

The circuit-design aspects of this project involve development of macrocell libraries in both NMOS and CMOS technologies. These libraries allow signal-processing circuits to be assembled rapidly, using proven architectural approaches to signal-processing problems. To a large extent, they are upgraded versions of cells already used successfully for implementing several signal-processing functions.

The CAD aspects of the project involve developing software that allows behavioral simulation and automatic layout generation of the target circuit functions based on a high-level description provided by the designer. The software system will allow signal-processing researchers to implement special-purpose circuits rapidly and efficiently, even if their background in circuit and system design is rather modest. The efficient silicon area utilization in this system far exceeds that of current and proposed "silicon compilers."

Target applications envisioned include filters, speech recognition, low-bit-rate speech coding, image processing, modems, line equalizers, and other telecommunications functions.

References

- (1) R. W. Brodersen, S. Pope, "Macrocell Design for Concurrent Signal Processing", *Computer Science Press*, pp. 395-412.
- (2) R. W. Brodersen, "Implications of VLSI Technology for Speech Processing", *Abstract in Proc. of COMPCON*, Los Angeles, CA, February 1983, pp. 645.

Theses

- (3) Sue Mellers, "A Preprocessor for a Speech Recognition System"
- (4) Alan Yuen, "Design of a Digital Spectra Analyzer"
- (5) David Mintz, "An Implementation of a Speech Recognition System"
- (6) Eric DXvies, "Endpoint Detection of Speech for Real Time Isolated Word Recognition"
- (7) Ramtin Ennail Beygin, "An LSI 7 Pole Lowpass Switched-Capacitor Filter"
- (8) Robert Kavalier, "Clustering Analysis Applied to Speech Recognition"

Ph.D. Theses

- (9) Tat Choi, "High Frequency Switched-Capacitor Filters"
- (10) Ron Fellman, "An MOS-LSI Adaptive Linear Prediction Filter for Speech Processing"
- (11) Menaham Lowey, "Design Considerations of a Speech Recognition System Using Special Purpose Integrated Circuits"
- (12) Hy Murviet, "A 1000 Word Real Time Speech Recognition System"

3.4. MOS Analog Circuits (D. Hodges)

3.4.1. High-Speed Monolithic Digital-to-Analog Converters

This project was aimed at improving the speed and linearity performance of fast digital-to-analog converters (DACs) for use in video and vector graphic display systems. Existing DAC designs typically employ R-2R resistive ladders and require trimming to eliminate differential nonlinearity. Also they often exhibit harmful "glitches" (transient output excursions) at major transition points due to time delay skews among decoder paths.

A new MOSLSI DAC technique has been developed and demonstrated to achieve excellent differential linearity and freedom from glitches. [1] A 8-bit experimental DAC was fabricated in our laboratory using 4 micron silicon gate NMOS technology to demonstrate the linearity and glitch immunity of the new technique. Overall die size is 2.5 mm by 3 mm. Output settling to 1% in less than 60 ns was observed. Glitch-free response is observed even with input skew greater than 20 ns. A doctoral dissertation on this work is in preparation.

Recent simulation studies have shown that through use of an advanced 3 micron CMOS process ("Berkeley CMOS") the settling time can be reduced to 40 ns. [2] Estimated die size with this process is 1.7 mm by 2.0 mm. Extension to 10 or even 12 bits resolution without sacrifice of differential linearity or glitch-free performance can be contemplated by using two levels of cascoding to combine output currents.

3.4.2. On-Board Frequency Reference

Analysis and simulation have been carried out for an on-chip CMOS active sinusoidal oscillator requiring no external components and having a frequency of oscillation which may be set in the range 1 MHz to 5 MHz by a single digital trimming operation. [3] The intended application of this oscillator is to provide a system clock for VLSI computing or communications components, eliminating the need for external RC or LC elements. Frequency stability comparable to crystal oscillators cannot be expected.

The proposed oscillator is composed of two CMOS operational amplifier integrators connected in a closed loop. Frequency trimming would be by means of a digitally-adjusted bias current source. Oscillation is initiated by normal thermal noise and reaches steady-state within 10-50 microseconds. Compared to relaxation oscillators, this sinusoidal oscillator may exhibit smaller timing jitter because the period of oscillation is determined by integration.

Simulation studies show that a 5 MHz oscillator with 10-15% harmonic distortion can be achieved with about 5 mW of power in a 3 micron CMOS process. Assuming that a suitable temperature-compensated bias current source can be designed, frequency stability better than $\pm 20\%$ over temperature and supply voltage variations should be achievable.

References:

- (1) V. W-K Shen & D. A. Hodges, "A 60 ns Glitch-Free NMOS DAC," *ISSCC Digest of Tech. Papers*, 1983, pp. 188-189.

Theses

- (2) K. Hirota, "CMOS 40 ns 8 bit Digital to Analog Converter," M.S. Plan II Report, University of California, Berkeley, Aug. 1983.
- (3) F-R Leu, "CMOS Sinusoidal Oscillator," M.S. Plan II Report, University of California, Berkeley, Aug. 1983.

4. TECHNOLOGY

4.1. Scaled Technology (C. Hu, R.S. Muller, R. W. Brodersen)

4.1.1. Thin Oxide Studies

In April, a paper on the theory of oxide tunneling conduction [1] was presented at the International Conference on Insulators on Silicon in Holland by a student. The conference paid for his travel expenses. Another paper on thin oxide device degradations was presented and published in the proceedings of Electrochemical Meeting in May [2]. Later in June at the Device Research Conference, we presented a paper on oxide charge transport [3]. Some novel phenomena of deep depletion regions being created or destroyed by charge tunneling through thin oxides appeared in the August issue of *Electron Device Letters* [10]. A comprehensive paper of our study on the 100 Å-gate MOSFET degradations was submitted [11]. Two abstracts have been submitted to the 1983 International Electron Devices Meeting - one dealing with impact ionization by tunneling electrons [14] and the other dealing with a comparison of device degradations induced by hot electrons and by tunneling electrons [15]. Finally, Professor C. Hu will present an invited paper on this series of studies at the 1983 IEEE Interface Specialist Conference in December 1983, in Florida.

In the next 6 months, emphasis is expected to shift to the phenomenon of time-dependent-dielectric breakdown - a leading reliability concern and challenging puzzle. Work will also begin on new dielectrics such as oxide-nitride combinations.

4.1.2. Hot Electron Studies

The pace of this fruitful research in this area continued. A paper on MOSFET characteristics near and beyond breakdown appeared in *IEEE Trans. Electron Devices* in June [4]. We published the first report on hot-electron currents in 0.14 μm channel MOSFET (and our model worked very well for that channel length) in June [5]. The study on punchthrough voltage appeared in September [6]. Another first was the successful photographing of light emission from Si MOSFETs and filamentary conduction under certain bias conditions [9]. A complete report on the channel hot-electron injection was submitted [12]. Light emission was also observed from forward-bias PN junctions [16]. The data and theory of bremsstrahlung radiation from Si MOSFET has been submitted for publication [17]. Finally, Professor C. Hu will give an invited paper on this research at the 1983 International Electron Devices Meeting. A spin-off of this research is a model of the feedthrough voltage in switch-capacitor circuits [8,13].

One area of future work will emphasize new structures that minimize hot-electron effects. Another project is concerned with obtaining a detailed picture of the effects of hot-carrier transfers on the valency and density of interface states. We are studying a modified charge-pumping technique that promises to permit the determination of several important interface-state properties.

References:

- (1) C. Chung, C. Hu, R.W. Brodersen, "Direct and Fowler-Nordheim Tunneling in Thin Gate Oxide MOS Structures", *International Conference on Insulating Films on Silicon (INFOS)*, April 1983. Proceedings to be published by North Holland Publishing Company.

- (2) C. Chang, M. S. Liang, C. Hu, R. W. Brodersen, "Charge Tunneling and Impact on Thin Oxide Device Reliability," *Electrochemical Society Meeting*, May 1983, pp. 609-610.
- (3) C. Chang, R.W. Brodersen, C. Hu, "Direct Tunneling in Thin Gate-Oxide Device Reliability," *Device Research Conference*, June 1983.
- (4) F. C. Hsu, R.S. Muller, C. Hu, "A Simplified Model of MOSFETs in the Break-down Mode," *IEEE Transactions on Electron Devices*, July 1983, pp. 571-576.
- (5) S. Tam, F.C. Hsu, C. Hu, R.S. Muller, P. Ko, "Hot Electron Currents in Very Short Channel MOSFETs," *IEEE Electron Devices Letters*, July 1983, pp. 249-251. F.C. Hsu, C. Hu, R.S. Muller, "A Model of Punchthrough Voltage," *IEEE Transactions on Electron Devices*, September 1983.
- (6) C. Hu, A. Neureuther, K. Terrill, "Computer Analysis of Alpha-Generated Charge," to appear in *International Journal of Solid State Electronics*, November 1983.
- (7) C. Hu, B. Sheu, "Modeling the Switch-Induced Error Voltage on a Switched Capacitor," *IEEE Circuit and Systems Letters*, September 1983.
- (8) F.C. Hsu, C. Hu, P. Ko, R.S. Muller, S. Tam, "Spatially-Resolved Observation of Light Emissions from Si MOSFETs," submitted to *IEEE Electron Device Letters*.
- (9) C. Chang, C. Hu, M.S. Liang, R.W. Brodersen, T. Yeow, "Creation and Termination of Deep Depletion by Charge Tunneling," *IEEE Electronic Device Letters*, August 1983.
- (10) C. Chang, C. Hu, M.S. Liang, R. W. Brodersen, "MOSFET Degradations Due to Stressing of Thin Oxide," submitted to *IEEE Transactions on Electronic Devices Meeting*.
- (11) C. Hu, P. Ko, S. Tam, "Lucky Electron Model of Channel Hot Electron Injection in MOSFETs," submitted to *IEEE Transactions on Electronic Devices*, C. Hu, B. Sheu, "Switch-Induced Error Voltage on a Switched Capacitor," submitted to *IEEE Journal of Solid State Circuits*.
- (12) R. W. Brodersen, C. Chang, C. Hu, M.S. Liang, "Carrier Tunneling- Related Effects in Thin Oxide MOSFETs," submitted to *International Electron Devices Meeting*.
- (13) R.W. Brodersen, C. Chang, C. Hu, M.S. Liang, "Hot-Carrier Induced Degradations in Thin Gate MOSFETs," submitted to *International Electron Devices Meeting*.
- (14) C. Hu, T. Ong, S. Tam, K. Terrill, "Light Emission from Forward-Biased PN Junctions," submitted to *IEEE Electron Device Letters*.
- (15) C. Hu, S. Tam, "Hot Electron Induces Photo-Carrier Generation in Silicon MOSFETs" submitted to *IEEE Transactions on Electronics Devices Letters*.
- (16) B. Sheu, C. Hu, "Switch-Induced Error Voltage on a Switched Capacitor", submitted to *IEEE JSSC*.

4.2. Berkeley Advanced CMOS (A. Neureuther)

The fabrication and testing of the first set of devices with the Advanced Berkeley CMOS process has been completed. The p-channel devices threshold was not sufficiently negative to turn the devices off. The n-channel devices worked as expected. The p-channel threshold problem was traced to a malfunction of the POC13 doping which had to be repeated after a cleaning etch. This etch reduced the poly thickness to the point that the source/drain implant was not adequately masked. This problem has been corrected on two runs in progress. Part of our present effort is to bring up this process on 4" wafers in our new microelectronics laboratory which is more suited to exploring design rule scaling.

4.3. Simulation Aids For Viewing Wafer Topography From Layout (A. Neureuther)

The object of this project is to develop an IC design aid, SIMPL (SIMulated Profiles from Layout), which gives the cross sectional view of the wafer topography and doping profile associated with the layout on a graphics editor. This simulator will complete the computer aided IC design sequence by linking the layout program to the topography simulator, SAMPLE, and its associated post processors for determining electrical parameters.

The first level simulator, SIMPL1 is now running in "C". Results for CMOS and Bipolar processes are described in the attached abstract which has been accepted for presentation at IEDM in December. SIMPL1 quickly calculates the approximate cross sectional view of the wafer at any step in the fabrication process. In order to make the simulation fast, SIMPL1 approximates all features as rectangles and has simple models of device physics. The simulations provide visual verification of the layout and the effects of each process step. The complete simulation of a CMOS inverter runs in 5 sec. on a VAX 11-780/UNIX.

The second level simulator, SIMPL2, will employ more process parameters and give a more realistic cross sectional view using the string model for profiles. The optional direct call of SAMPLE from SIMPL2 will be implemented for the accurate topography simulation. The conceptual design and implementation of SIMPL2 is now under study.

4.4. Aluminum/Silicon Contact Electromigration and Contact Resistivity Measurement (W.G. Oldham)

The failure of Al/Si contact due to current stress is being investigated. As device geometries scaled down, the density of current flow through conductor is increased, hence, aggravates the occurrence of electromigration. Two types of failure have been observed, one is contact leakage/spiking which is due to high silicon solubility in aluminum and electromigration, the other is aluminum line opening which is due to aluminum electromigration. Other refractory metal/silicide to silicon contact is also being considered.

Special attention has been paid to design the testing apparatus in such a way that chip temperature rise due to local heating is eliminated and constant temperature throughout the chip is assured. A IBM personal computer is also used as the controller to perform automatic stressing, resistance and leakage current measurement, data analysis, etc.

The location of contact failure site is dependent on the contact current distribution which, in turn, depends on contact resistivity and geometry. A study on contact resistivity measurement is also conducted. A model using conformal transformation is adapted to study current distribution along and below the contact. A novel technique which would yield uniform contact resistivity while preserves electromigration resistance is also being considered.

References

- (1) R. Kazerounian, W. G. Oldham, "Threshold Shift from As-Implantation of SiO₂", *Proceedings of the Electrochemical Society Meeting*, May 1983, San Francisco.
- (2) R. Kazerounian, W. G. Oldham, "CODMOS A Depletion Device Using Fixed Oxide Charge", *presented at Device Research Conference, June 1983, Burlington*.

4.5. Modification of Metal-Si Contacts With Ion Beams (N. Cheung)

The objective of this project is to investigate the effects of ion implantation on the electrical and metallurgical properties of metal-silicon contacts. Both silicide forming and non-silicide forming systems are being studied.

Arsenic ions have been implanted through thin metal films to doses on the order of 10E14 to 10E16 cm⁻². The high dose implant is used to simultaneously intermix the metal-Si interface and heavily dope the interface silicon. In the simple Al-Si eutectic system, we have observed that interface mixing enhances the uniform dissolution of Si into Al, which minimizes the Al pitting into shallow Si junctions due to subsequent annealing. In silicide forming systems, such as TaSi₂, ion beam mixing has been shown to promote silicide formation at reduced temperatures. The sheet resistance of the Si substrate decreases due the peaking of implanted dopants at the metal-Si interface. However, the implantation increases the contact resistance by a factor of five.

The effects of ion beam mixing on contact electrical properties are determined using a test vehicle that contains patterns for measuring the contact resistance and Schottky barrier height of contacts ranging from 20 to 1.5 microns square. Metallurgical reactions are monitored using Rutherford Back-scattering Spectrometry and SEM.