

VLSI CIRCUIT SIMULATION USING A VECTOR COMPUTER

**STEPHEN K. MCGROGAN
CONTROL DATA CORPORATION
OAKLAND, CALIFORNIA**

VLSI CIRCUIT SIMULATION USING A VECTOR COMPUTER

**Stephen K. McGrogan
Senior Staff Consultant**

**Control Data Corporation
1 Kaiser Plaza, Suite #301
Oakland, California, 94612**

ABSTRACT

Simulation of circuits having more than 2000 active devices requires the largest, fastest computers available. A vector computer, such as the CYBER 205, can yield great speed and cost advantages if efforts are made to adapt the simulation program to the strengths of the computer.

ASPEC and SPICE (1) are two widely used circuit simulation programs. ASPECV and VAMOS (5) are respectively vector adaptations of these two simulators. They demonstrate the substantial performance enhancements possible for this class of algorithm on the CYBER 205. ASPECV is in use at ISD. VAMOS is in daily production use at MOSTEK.

INTRODUCTION

Over the past decade, the design of integrated circuits has become increasingly complex. Manufacturers who once had special purpose circuits of only a few dozen components now have microprocessors and random access memory chips constructed of thousands of devices. While early circuits were readily designed and debugged by hand, the more complex circuits have necessitated computer assistance.

During one phase of computer aided design, circuit simulation programs are used. These programs are given circuit interconnection information (nodes) and device characterizations (models). After establishing initial current and voltage conditions at time zero, they simulate circuit operation by evaluating device conductances and node voltages over small increments of time. Due to the rapid response of microcircuitry to voltage changes, circuit simulation must often be performed at timesteps of a few hundred picoseconds. This small timestep may necessitate thousands of steps to simulate circuit performance for a given set of initial inputs. Many such simulations (which may each require hours on an IBM 3081 or CDC 176) are required to thoroughly explore a circuit's characteristics over a wide range of temperatures and input sets.

The speed of a supercomputer is valuable to engineers designing such large scale integrated (VLSI) circuits. These engineers are, however, unwilling to compromise simulation accuracy for speed. For this reason, various projects have investigated vector computers (2) (3) (4) for use in the transient analysis of VLSI circuits.

Two well-known and widely used circuit simulators are ASPEC, copyrighted by Mr. Frank Jenkins, and SPICE, copyrighted by the Regents of the University of California. ASPECV is the product of a technical team from the San Francisco District of Control Data Corporation Professional Services Division. This team spent approximately one man-year analyzing ASPEC in detail. Their effort included extensive conversations with the program's author and the rewriting of select areas of code for enhanced performance.

The program VAMOS was developed by Steven D. Hamm and Steven R. Beckerich of MOSTEK Corporation. VAMOS evolved from a simple installation of SPICE2 into a program in which 80 percent of the analysis routine code is vectorized. Many sections of code were radically changed due to the application of algorithmic, rather than simple syntactic, vectorization.

ARCHITECTURAL CONSIDERATIONS

ASPEC AND SPICE were initially developed for a type of computer similar to the Control Data Corporation 6400. Originally, the programs were designed to handle circuits with fewer than 600 devices. Intentional minimization of memory requirements increased central processor time. Many users modified ASPEC and SPICE for use with large-scale circuits, extending the programs into areas far beyond their design. When any design is so overextended, there are often undesirable consequences. One obvious consequence was long running time on circuits with more than 2,000 devices.

Optimum performance for both ASPEC and SPICE required retailoring program design to fit the architecture of the CYBER 205. The Cyber 205 used has two vector pipes, a 16 megabyte memory, and is capable of 200 million floating point operations per second (Megaflops) on 64 bit operands. To maximize performance, the characteristics of this hardware must be considered. Some major considerations are:

1. The CYBER 205 defines a vector as contiguous memory locations. While ASPEC has a compatible memory organization, SPICE2 linked list storage needs re-organization.
2. The scalar functional units on the CYBER 205 are pipelined. Code that cannot be vectorized can be optimized by taking advantage of inherent parallelism. Even so, the performance of scalar code will probably be substantially less than the theoretical maximum of 50 Magaflops.

3. The hardware can generate and use bit vectors, which are useful in vectorizing loops containing conditional statements. These bit vectors aid in producing routines that have no scalar code and run at full vector speed.

4. The virtual memory of the CYBER 205 provides over 2 trillion words of user memory space. Any program that repetitively uses more than the entire physical memory may, however, generate a great amount of paging delay. This fact constrains the choice of algorithms, as a fast algorithm may require additional memory.

PROGRAM DESIGN

Both ASPEC and SPICE perform their simulations by alternating modeling routines with a current matrix solution routine. The modeling routines calculate the new device conductances based on device operating points. There is one model for each type of device, such as diodes, jfets, mosfets, and bi-polar transistors. One model must simulate many different operating modes and consequently has many branches and special cases.

The matrix solution routine calculates branch currents based on the conductances calculated by the modeling routines. From these currents new node voltages are obtained. This routine uses sparse Gaussian Elimination techniques. The time required by this routine grows very rapidly and non-linearly with circuit complexity.

In SPICE, to best utilize the long vector capabilities of the CYBER 205, an interface routine was written between the vectorized analysis routines and the rest of SPICE2. This routine reorganized memory into contiguous vectors and established new element pointers. ASPEC was similarly treated. The task was less formidable as data was already in homogeneous arrays.

In both VAMOS and ASPECV, vectorization of device equations is done by long vector operations with conditional stores for the results. All devices are evaluated in all regions of operation and the results are masked together to form composite result vectors. This technique avoids the data motion overhead characteristic of other methods at a cost of extra operations in each region. For VAMOS, the data given in Table 1 shows the tremendous advantage vectorization provides. The small amount of scalar store code remaining in MOSFET contributes 19.4 of the total 25.5 seconds.

ROUTINE	SCALAR	VAMOS	RATIO
LOAD	19.9	1.8	11.1
DIODE	79.4	3.6	22.1
MOSFET	325.4	25.5	12.8

Table 1. VAMOS Routine Comparisons

In VAMOS, the vector startup time required by the CYBER 205 caused the rejection of a vectorized matrix solution method for subcircuits as used in the program CLASSIE (2). Instead, effort was expended in scalar code optimization to achieve maximum instruction overlap. As part of the preprocessing phase of the program, the row-column lookup is performed once and the indices are stored in an auxiliary array.

In addition to the VAMOS techniques, ASPECV's routine EQNSOL detects perfect alignment between rows in the matrix. As circuit size increases, the number of such rows increases dramatically. Full row-length linked triads are executed in this case.

PROGRAM PERFORMANCE

Table 2 illustrates a comparison between a scalar version and VAMOS. The scalar version was already heavily optimized. The circuit tested contained 2256 mosfets, 1312 diodes, 1774 resistors and capacitors, and had 1429 equations with 98.9 percent matrix sparsity. Overall VAMOS performance was 3 times scalar, with 4 times in transient analysis. VAMOS performed the analysis over 100 times faster than a VAX-11/780.

ROUTINES	SCALAR	VAMOS
READIN	68.4	51.9
SETUP	34.7	22.7
DC SOLUTION	47.8	19.0
TRANSIENT	503.8	126.4
OUTPUT	5.6	5.6
TOTAL	660.3	225.9

Table 2. VAMOS Program Performance Comparison

Table 3 shows the characteristics of a series of flexible circuits which can be made any size by repeating a basic circuit block. Resistors and capacitors are also present but are irrelevant to modeling time. Table 4 gives execution time for two processors running ASPEC, and the current version of ASPECV on the CYBER 205. It is projected that, with continued effort, for large circuits the CYBER 205 mosfet run times could be reduced by another factor of 2 to 3. Table 5 shows that the time to model a given device decreases with increasing circuit size, a very desirable characteristic for VLSI circuitry.

CIRCUIT	DIODES	MOSFETS	NODES	MATRIX
1	50	50	30	119
2	100	100	54	220
4	200	200	102	470
8	400	400	182	860
16	800	800	358	1718
32	1600	1600	718	3473

Table 3. Circuit Characteristics

CIRCUIT	TIME STEPS	UNIVAC 1182	CDC 176	CDC 205
1	420	30	6	3
2	622	82	16	6
4	869	208	42	15
8	1658	697	141	40
16	1658	1421	301	76
32	1658	TOO BIG	TOO BIG	158

Table 4. ASPEC/ASPECV Comparison

CIRCUIT	AVERAGE TIME (micro-secs)			VECTOR
	diode	mosfet	EFFECIENCY	
1	9.7	39		50
2	7.1	32		66
4	5.8	28		80
8	5.2	26		89
16	4.7	25		94
32	4.5	24		97

Table 5. ASPECV Size/Efficiency

Since most circuit simulation runs produce a great deal of printed output, current simulations using ASPECV spend the majority of their time in Fortran I/O. As an example, one ASPECV circuit containing 1000 devices and 950 nodes initially ran in 980 seconds on a UNIVAC 1182 and in 141 seconds on the CYBER 205. After optimizing everything but the diode and mosfet models, the same circuit required 72 seconds on the 205. Of the 72 seconds, 39 were spent in the models. ASPECV requires only 44 seconds to simulate the same circuit. Only 6.3 seconds are required in the models: 1.3 in diodes, 5.0 in mosfets. Although the mosfet model is still several times slower than theoretically possible, further effort would yield small returns indeed. The simulation mentioned spends over 66 percent of its time in Fortran I/O routines.

CONCLUSION

Program speedups of 3 to 4 were accomplished through vectorization. Future work directed at vectorization of the remaining scalar code may result in a similar speed increase. Fortran I/O provides an effective limit to maximum attainable speed.

REFERENCES

1. L. W. Nagel, "SPICE2: A computer Program to Simulate Semiconductor Circuits," Memorandum No. ERL-M520, University of California, Berkeley, May 1975.
2. A. Vladimirescu and D. O. Pederson, "Circuit Simulation on Vector Processors," Proceedings, IEEE International Conference of Circuits and Computers, New York, October 1982.
3. J. C. May, "A Device Clustering Algorithm for Vectorized Circuit Simulation," Proceedings, IEEE International Symposium on Circuits and Systems, Newport Beach, Calif., May 1983.
4. S. McGrogan and G. Tarsy, "Vector Enhancement of a Circuit Simulation Program," Proceedings, Symposium on CYBER 205 Applications, Colorado State University, Fort Collins, Colo., August 1982.
5. S. D. Hamm and S. R. Beckerich, "VAMOS: Circuit Simulation Program for a Vector Computer," Technical Paper, MOSTEK Corporation, Carrollton, Texas, August 1983.