Parallel Computing Using a Mixed-Level Device-Circuit Simulator

Shigeo Satoh and Noriaki Nakayama

Fujitsu Laboratories Ltd.

1. Introduction

A mixed-level (M-L) device-circuit simulator is the most accurate because transistor characteristics are expressed by the device simulator. It simultaneously solves device (Poisson and current continuity) and circuit (nodal) equations [1][2]. CPU time, however, is large in proportion to the number of transistors in the circuit. The size of the M-L simulator is, therefore, restricted.

Parallel computing has the potential to reduce the CPU time but this depends on the computation method. We propose a parallelization method for an M-L simulator. This paper shows the reduction in the CPU time of a parallel M-L simulator obtained using a multiple-instruction multiple-data (MIMD) parallel computer. We compare vector and parallel computing.

2. Parallel computation

To improve the Newton loop convergence of the nodal equation, the exact derivatives of the terminal currents for terminal voltages are needed. Our M-L simulator calculates the derivatives by changing the terminal voltages (ΔV). For a MOSFET, the device equation is solved three times in every Newton loop and the derivatives are calculated from these results (Fig. 1).

For single or scalar/vector processor machines, device equations are solved sequentially, then the circuit equation (Fig. 2(a)). Here, two MOSFETs are included in the circuit and Id1 and Id2 represent the MOSFET currents.

For multiprocessor machines (parallel computers), three processors are used for calculating the MOSFET derivatives and the circuit equation is solved by the host (Fig. 2(b)). Ideally, the calculation will be three times faster because the CPU time required for solving the circuit equation is negligible.

3. Evaluation

To examine the dependence of CPU times on circuit size, we simulated ringoscillator circuits with five to nine stages (Fig. 3) using a 2-GFLOPS vector computer, the Fujitsu VP2400, and a 16-MIPS-per-processor parallel computer, the Fujitsu AP1000 (Fig. 4). Here, "scalar" means CPU time using a scalar processor alone, "vector" means a scalar/vector processor, "single" means a single processor, and "parallel" means a 64-cell multiprocessor.

For the vector computer, CPU time increases with the number of stages. Circuit size is, therefore, limited by CPU time. For the parallel computer, CPU time is constant even when the number of stages increases and the circuit size is independent of the CPU time.

We compared the speedup factors for the vector (scalar/vector CPU time) and the parallel (Single/parallel CPU time) processor (Fig. 5). The vector processor's speedup factor does not depend on the circuit size because the vector length is independent of this size. The speedup factor for the parallel processor increases with the circuit size because the number of transistors calculated in the Newton loop increases. The parallelization efficiency (speedup factor/number of cell processors used) is not ideal, about 50%, because the host must wait until all the device equations converge. The parallel computer thus is suitable for large circuits.

4. Summary

We propose a parallelization method for calculating derivatives. A parallel computer is suitable for an M-L simulator because the circuit size is independent of CPU time. The exact cell library for a logic simulator is thus easily obtained.

References

[1] B. Meinerzhagen, J. M. J. Kruchen, K. H. Bach, F. M. Stecher, and W. L. Engl, "A modular approach to parallel mixed level device/circuit simulator," VLSI Process/Device Modeling Workshop, p 170, 1990

[2] H. Oka, S. Satoh, and N. Nakayama, "Bipolar circuit simulation system using a two-dimensional device simulator," Simulation of Semiconductor Devices and Processes (SISDEP), p 127, 1988

$$\frac{\partial I_D}{\partial V_{DS}} = \frac{I_D(V_D+\Delta V, V_G) - I_D(V_D, V_G)}{\Delta V}$$

$$\frac{\partial I_D}{\partial V_{GS}} = \frac{I_D(V_D, V_G+\Delta V) - I_D(V_D, V_G)}{\Delta V}$$

$I_D(V_D, V_G)$, $I_D(V_D+\Delta V, V_G)$, and $I_D(V_D, V_G+\Delta V)$ are obtained by solving the device equation.

Fig. 1 Current derivatives

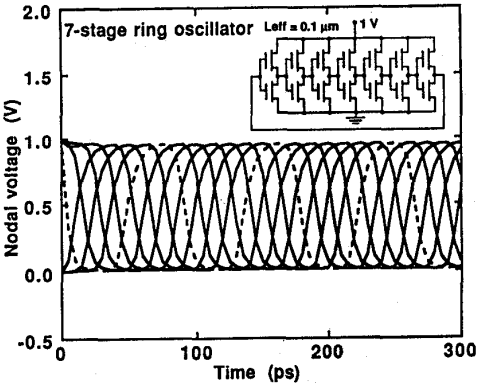
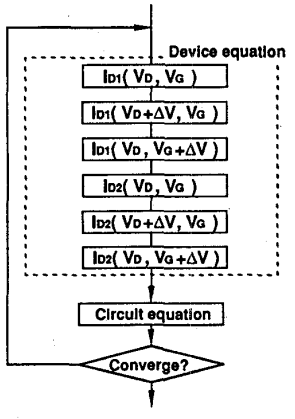
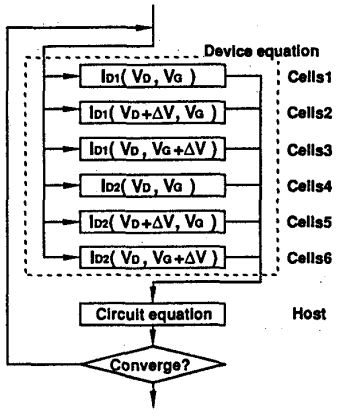


Fig. 3 Evaluation circuit



(a) Single processor



(b) Multiprocessor

Fig. 2 Newton loop of a mixed-level device-circuit simulator

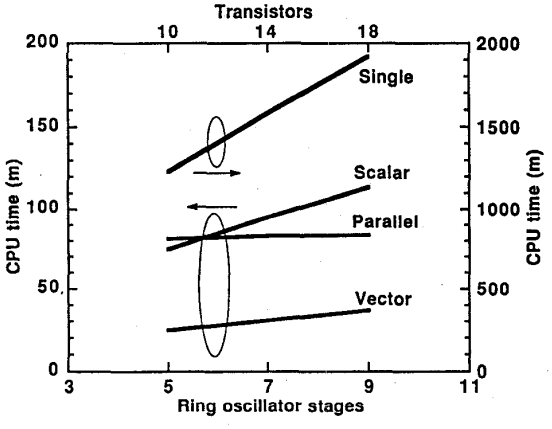


Fig. 4 Dependence of CPU time

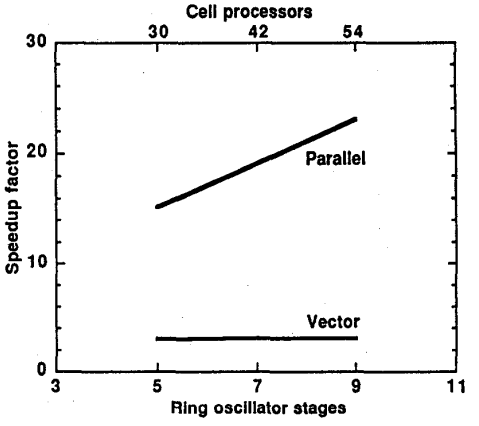


Fig. 5 Comparison of speedup factors