Parallelizing Electronic Circuit Simulation on Multicore Computer Cluster

Bojan Anđelković, Marko Dimitrijević,
Miona Andrejević Stošović and Vančo Litovski

Abstract - This paper presents an algorithm for parallelization of transistor-level analog circuit simulation. Basic information regarding modern simulation, leading to the need for parallel simulation is presented. Implementation of a new algorithm based on parallel equation formulation in a mixed-mode simulator is explained. Simulation performances are considered for two computer cluster configurations: one based on nodes with single core processors and the other based on nodes with multicore processors.

Keywords – Parallel circuit simulation, Distributed simulation.

I. Introduction

The simulation process of modern complex integrated electronic circuits may be characterized as memory and computationally intensive, and algorithmically complex. These properties pertain to the fact that a large number of ordinary (and, potentially, partial) nonlinear differential equations have to be solved for long running excitations. In addition, a huge amount of data may be created in a single simulation run, which needs to be processed and interpreted. Because of all these reasons simulation runtimes are very long. Having in mind that every design needs many simulation runs of the same design in order to get optimal solutions and satisfy the design requirements, it is obvious that long simulation runtimes lead to a slow design process. One possibility to reduce these runtimes is to parallelize the simulation algorithm and use computer clusters or multicore processors to execute simulations. In this approach complex calculations necessary during the simulation process can be distributed over different workstations/processors and performed simultaneously.

Over last decade, as personal computers performance has increased and prices have fallen steeply, both for the PCs themselves and the network hardware necessary to connect them, dedicated clusters of PC workstations have provided significant computing power at low cost.

Bojan Anđelković is with Fujitsu Microelectronics Europe, Germany. E-mail: abojan@gmail.com

Marko Dimitrijević, Miona Andrejević Stošović and Vančo Litovski are with the Department of Electronics, Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia. E-mail: (marko.dimitrijevic, miona.andrejevic, vanco.litovski)@elfak.ni.ac.rs.

There are several parallel simulators of electronic circuits that have been developed recently, such as Xyce [1], Titan [2] and SEAMS [3]. Titan and Xyce are parallel transistor level simulators that use SPICE as modeling language. Both simulators implement complex partitioning algorithms to split the circuit description and distribute generated partitions to different workstations/processors. These partitions are then simulated in parallel. Appropriate synchronization protocols should be applied to exchange necessary simulation data between the circuit partitions. The main goal of these parallel algorithms is to minimize communication between the workstations/ processors and achieve their equal load. SEAMS is a VHDL-AMS simulator that implements parallel digital simulation, while parallel mixed-signal simulation is under the development. A broad survey of various parallel simulator implementations and algorithms can be found in [4].

This paper presents the concept of development a parallel transistor level simulator of electronic circuits that executes on a computer cluster using MPI. Parallelization of equation formulation for nonlinear analog circuit elements is implemented in order to reduce simulation time.

II. Parallel Equation Formulation

In order to simulate complex mixed-signal electronic circuits at transistor level, they have to be modeled using algebraic equations and nonlinear Ordinary Differential Equations (ODE). ODEs are discretized in order to create sets of nonlinear algebraic equations. This generates a potentially large system of equations to be solved at a large number of time instants depending on the properties of the system under simulation and the stimulus signals. The system of nonlinear equations is solved iteratively with the help of linearization i.e. by application of Newton methods.

The algorithm for simulation of nonlinear dynamic electronic circuits in time domain is shown in listing 1 [5]. As it can be seen, at each iteration and at every time instant the matrix entries of the system of linear equations have to be recalculated. These entries are derivatives of the nonlinear equations and are computed within separate subroutines. Having in mind the number of matrix entries, the number of iterations and the number of time instants, it is necessary to provide an immense computational effort. It has been shown that even for small systems, equation

formulation takes more computational time than equation solution. Therefore, the calculation of matrix entries and equation formulation for non-linear circuit elements can be parallelized. The part of the simulation algorithm that can perform in parallel is highlighted by a rectangle in listing 1.

If we consider the circuit matrix as a sum of several matrices the number of which is equal to the number of processors implemented, we may create the whole matrix by creating its parts and then by summing them. That is illustrated in Fig. 1. There is no specific criterion for allocation of the circuit elements to specific processor (or submatrix). Simply the total number of non-linear circuit elements is divided by the number of processors and the list of elements is divided to equal parts and partitions are created. Such parallelization of the simulation algorithm is a new approach different from already developed solutions. It requires neither a sophisticated circuit and task partitioning algorithm nor synchronization protocols between these partitions, so it is easy to implement on a computer cluster using MPI routines.

```
generate node voltages and specified branch
      currents x^0 =[(v^0)^T (i^0)^T ]^T ;
choose h;
n = 0;

while(t < T) {                  /* time loop */
   m = 0;
   predict x^(n+1,0) ;
   until convergence {          /* iterative loop */
      generate descretized models;
      generate linearized models;
      formulate system of linear equations;
      solve the system and find x^(n+1,m+1) ;
      m++;
   }
   t = t + h;
   n++;
}
```

Listing 1. The simulation algorithm for nonlinear dynamic circuits in time domain

The generation of matrix entries for constant and linear dynamic elements is not performed in parallel, since these calculations may be performed outside of the iterative loop. Moreover, the matrix contributions for constant elements are calculated only once outside time and iterative loops, while entries for linear time dependent elements are calculated at every time instant outside iterative loop. All this reduces the overall time necessary for equation formulation. When parallel generation of matrix entries for various nonlinear elements is finished, the complete circuit matrix is formed (Fig. 1). Then system of linear equations is solved. That task can also be parallelized which should lead to further reduction in simulation time.

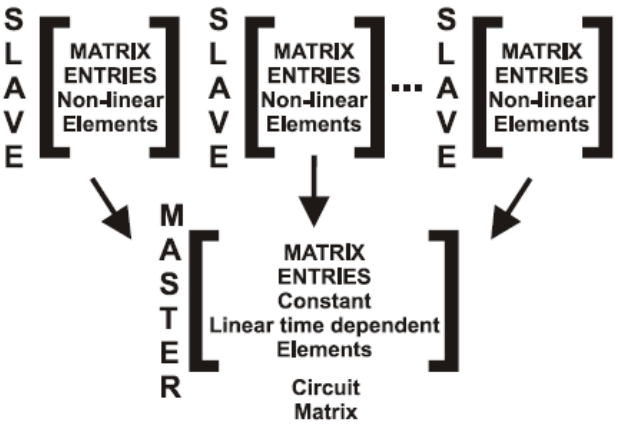

Fig. 1. Parallelization of equation formulation

III. PARALLEL SIMULATOR IMPLEMENTATION

The presented parallelization of equation formulation process is implemented in the simulator Alecsis [6]. It is a mixed-signal and mixed-domain simulator with proprietary hardware description language AleC++ [7] capable for modeling and simulation of complex systems containing different kinds of devices and subsystems [8]. The developed simulator with parallel simulation capability is called pAlecsis (*Parallel* Analog and Logic Electronic Circuits Simulation System).

The implementation of parallelization in the pAlecsis simulator on a computer cluster is shown in Fig. 2. Parallel equation formulation is implemented using one of the most common of parallel algorithm prototypes, master-slave algorithm [9]. In this algorithm calculation of matrix contributions for non-linear circuit elements at each time instant and iteration is distributed to multiple slave processes and they are calculated simultaneously. At the same time master process calculates matrix entries for specific number of nonlinear elements as well as for constant and linear time dependent elements. Master and slave processes execute on different cluster nodes (PC workstations).

Since multiple cluster nodes calculate contributions for different elements in parallel, the time necessary for equation formulation decreases.

In order to minimize communication between cluster nodes, appropriate data structures for all elements of the circuit are generated on all nodes simultaneously during compilation of the AleC++ model. In that way all cluster nodes have the information necessary to generate matrix contributions for all elements. Each node of the cluster performs equation formulation and calculation of matrix entries for equal number of nonlinear circuit elements. When entries for all elements on one slave are generated, they are sent to the master node using appropriate MPI routines (Fig. 2).

When the master node receives matrix entries from all slaves, it flushes them to the system matrix and performs one simulation step. In order to enable calculation of matrix

entries on slave nodes, the master node should send to the slaves' vectors of solutions of the system of equations for the two past time instants and previous iteration (denoted with *vp1*, *vp2* and *vi* respectively in Fig. 2). Appropriate MPI routines for transferring data are used to send and receive these vectors.

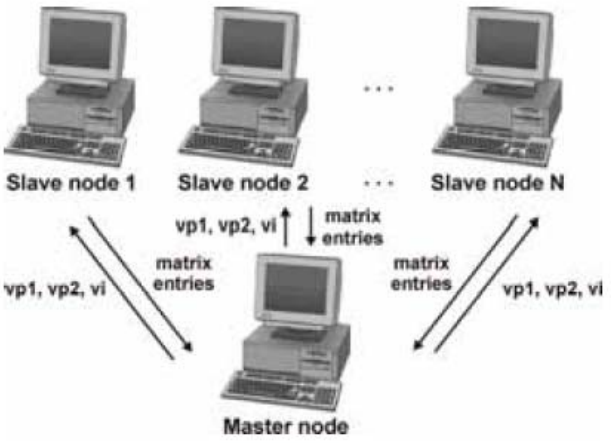

Fig. 2. Implementation of the pAlecsis simulator on a computer cluster

IV. Parallel Simulation Performances

Sequential simulation algorithms executing on a single workstation are tested for correctness usually by only seeing whether they give the right result. For parallel programs, that is not enough, but one wishes to reduce the simulation time. Therefore, measuring of simulation time is part of testing the parallel simulator to see whether it performs as intended. Usually performances of the parallel simulator are specified as speedup. If parallel simulation executes on N single processor cluster nodes, speedup is normally defined as [9]:

$$Speedup = \frac{Simulation\ time\ on\ 1\ node}{Simulation\ time\ on\ N\ nodes} \qquad (1)$$

Implemented parallel simulation algorithm reduces simulation time for bigger circuits when time necessary to calculate matrix entries for all elements at every time instant and every iteration exceeds time necessary to calculate matrix entries on slave nodes and send them to master node over the interconnecting network. For such circuits the parallel simulation on the cluster is faster than the simulation on a single processor workstation.

In order to determine the size of circuits in number of transistors for which there is a speedup in simulation on a cluster with two nodes, parallel simulations using the presented algorithm were performed on circuits consisting of various number of MOSFETs. These circuits are generated by successive replication of bilinear SC filter circuit with MOSFET operational amplifiers. Then speedup is calculated according to (1).

TABLE I
Beowulf cluster structure

Component	Specification
Master node	PC Pentium IV, 2.4GHz, 1GB RAM, 240GB HDD
Slave nodes	8× PC Pentium IV, 2.4GHz, 512MB RAM, 80GB HDD
LAN	1Gbit Ethernet

TABLE II
Speedup of parallel simulation in pAlecsis for Beowulf cluster structure

Number of MOSFETS	Simulation Speedup (2 cluster nodes)
740	1.1
1480	1.5

TABLE III
New computer cluster structure

Component	Specification
8 × Two quad-core Intel Xeon E5420, 2.5GHz, 4GB RAM, 250GB HDD	
1.4TB RAID5 network attached storage	
LAN	dual 1Gbit Ethernet

TABLE IV
Speedup of parallel simulation in pAlecsis for New computer cluster structure

Number of MOSFETS	Simulation Speedup
740	0.9
1480	1
1700	1.05

In this paper simulation results from two different computer clusters are compared. The first one is Beowulf cluster whose structure is given in Table 1. The generated simulation results are given in Table 2 [10].

New computer cluster structure is given in Table 3. This structure consists of 64 processors, and it was expected to give much better results than the first structure. The generated simulation results are given in Table 4. As it

can be seen there is no speedup for 740 and 1480 transistors, and there is a slight speedup for 1700 transistors. Total time simulation time is about 30% less. Given results are for simulations performed on 2 and 3 cluster nodes. For more cluster nodes simulation time is even greater.

VI. CONCLUSIONS

In this paper we compared simulation time results for two different computer clusters. We may conclude that clusters with better performances do not always give better results, because there is a problem in communication among cluster nodes. It is obvious here that communication time can exceed time necessary to calculate matrix entries on slave nodes. Our next step is to perform simulations on more complex circuit, because we then expect much better results.

REFERENCES

[1] http://www.cs.sandia.gov/xyce/

[2] Fröhlich, N., Riess, B. M., Wever, U., Zheng, Q., *A New Approach for Parallel Simulation of VLSI-Circuits on a Transistor Level*, IEEE Transactions on Circuits and Systems, Part I, Proc. Int. Conference on Parallel and Distributed Processing Techniques and Applications, pp. 601-613, Vol. 45, No. 6, June 1998.

[3] Martin, D. E., Radhakrishnan, R., Rao, D., Chetlur, M.,Subramani, K., Wilsey, P., *Analysis and Simulation of Mixed-Technology VLSI Systems*, Journal of parallel and distributed computing, vol. 62, No 3, pp. 468-493, 2002.

[4] Savić, M., Anđelković, B., Litovski, V., *Parallel Mixed-Mode Simulation – Preliminary Study*, Proc. INDEL 2004, Banja Luka, pp. 76-79, 2004.

[5] Litovski, V., Zwolinski, M., VLSI Circuit Simulation and Optimization, Chapman and Hall, London, 1997.

[6] Mrčarica, Ž., et al., *Alecsis 2.3, the simulator for circuits and systems. User's Manual*, Laboratory for Electronic Design Automation, Faculty of Electronic Engineering, University of Niš, Yugoslavia, LEDA-1/1998, http://leda.elfak.ni.ac.rs/projects/Alecsis/alecsis.htm

[7] Litovski, V., Maksimović, D., Mrčarica, Ž., Mixed-Signal Modeling with AleC++: Specific Features of the HDL, Simulation Practice and Theory 8, pp. 433-449, 2001.

[8] Mrčarica, Ž., Ilić, T., Glozić, D., Litovski, V., Detter, H., *Mechatronic Simulation Using Alecsis: Anatomy of the Simulator*, Proc. Eurosim'95, Vienna, Austria, pp. 651-656, 1995.

[9] Gropp, W., Lusk, E., and Skjellum, A., *Using MPI: Portable Parallel programming with the Message-Passing Interface*, second edition, MIT Press, 1999.

[10] Anđelković, B., Litovski, V., *Parallel Transistor Level Simulation based on Parallel Equation Formulation implemented on a Beowulf Cluster*, Simulation News Europe 17/3-4, pp. 55-58, December 2007.