Zbornik radova 51. Konferencije za ETRAN, Herceg Novi – Igalo, 4-8. juna 2007.
Proc. 51st ETRAN Conference, Herceg Novi – Igalo, June 4-8, 2007

IMPLEMENTATION AND PERFORMANCE ANALYSIS OF PARALLEL CIRCUIT SIMULATOR ON BEOWULF CLUSTER

Bojan Anđelković, Vančo Litovski, Predrag Petković, *Faculty of Electronic Engineering Niš*

Abstract – *Simulation process of modern complex integrated circuits is computationally and resource intensive leading to long simulation runtimes. Parallel simulation on a Beowulf cluster of PCs is an efficient way to cope with such high requirements. This paper presents an algorithm for parallel simulation that introduces parallelization in equation formulation phase and uses multiple workstations to simultaneously calculate matrix entries for nonlinear analog circuit elements. Performances of the implemented parallel simulator are tested and analyzed for different circuit complexity, number of cluster nodes and sizes of messages between the nodes.*

1. INTRODUCTION

Circuit simulation is one of the most important computer aided design methods for the analysis and validation of modern complex integrated circuits. The simulation process of such circuits involves intensive calculations requiring significant processing power, large operating memory, as well as big storage capacity. These requirements pertain to the fact that a large number of ordinary (and, potentially, partial) nonlinear differential equations have to be solved for long running excitations. Therefore, 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 slow the design process. One possibility to reduce these runtimes is to parallelize the simulation algorithm and use parallel computers to execute simulations. In this approach complex calculations necessary during the simulation process can be distributed over different workstations/processors and performed simultaneously.

The development of low-cost personal computers with high computing power and gigabit LAN network connections in past decade provided possibility for implementation of inexpensive distributed multiprocessor systems such as computer clusters. One particular implementation of this approach, involving open source system software and dedicated networks, has acquired the name "Beowulf" [1].

The development of software for running on computer clusters requires that a programmer himself implements the most appropriate parallel algorithm for the specific domain problem. There are several parallel computational models that define the types of operations available to the program. One of them that is the most common for applications on Beowulf clusters is message passing model of parallel computation, and in particular the Message Passing Interface (MPI) implementation of that model [2][3]. In the message-passing model the complete program is split into a set of processes that execute on different cluster nodes (PC workstations). The processes have only local memory and are able to communicate with each other by sending and receiving messages.

Several parallel simulators of electronic circuits have been developed recently, such as Xyce [4], Titan [5] and SEAMS [6]. Titan and Xyce use SPICE as modeling language for parallel transistor level simulations. 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. SEAMS is a VHDL-AMS simulator that implements parallel digital simulation. In order to perform parallel mixed-signal simulation, the parallel digital simulator is synchronized with analog simulation kernel executing on one processor. The drawback of such approach is the fact that analog simulation is often the slowest process in the simulated system, so it should be parallelized. A broad survey of various parallel simulator implementations and algorithms can be found in [7].

This paper describes the implementation of a parallel simulator of electronic circuits that executes on a Beowulf cluster using MPI. The next section presents parallelization of equation formulation for nonlinear analog circuit elements that is used to reduce long simulation runtimes. The third section introduces the results of performance analysis of the implemented simulator. The analysis is performed for different circuit complexity and number of cluster nodes involved in the simulation. Moreover, this section studies the influence of message sizes between cluster nodes to the overall simulation performances.

2. PARALLEL SIMULATOR IMPLEMENTATION

The algorithm for simulation of nonlinear dynamic electronic circuits in time domain is shown in Fig. 1 [8]. Complex mixed-signal electronic circuits at transistor level are modeled using algebraic equations and nonlinear Ordinary Differential Equations (ODE), in order to simulate their behavior in time domain. ODEs are discretized and that generates sets of nonlinear algebraic equations. The system of nonlinear equations is solved iteratively with the help of linearization i.e. by application of Newton methods.

As it can be seen in Fig. 1, 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 specific 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 nonlinear circuit elements should be parallelized. The part of the simulation algorithm that could be parallelized is highlighted by a rectangle in Fig. 1.

If one considers the circuit matrix as a sum of several matrices the number of which is equal to the number of processors used in the simulation, it is possible to create the whole matrix by creating its parts and then by summing them. That is illustrated in Fig. 2. Every submatrix contains entries for specific number of nonlinear circuit elements. There is no specific criterion for allocation of the circuit elements to specific processor (i.e. submatrix). The total list of nonlinear elements is partitioned into equal number of elements that is determined by simply dividing the total number of nonlinear circuit elements by the number of processors. Then every processor calculates submatrix entries for elements in one partition. 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 Beowulf cluster using MPI routines.

```
generate node voltages and
specified branch currents $x^0=[(v^0)^T (i^0)^T]^T$;
choose time step, h;
n = 0;
/* time loop */
while (t<T) {
  m = 0;
  predict $x^{n+1,0}$;
  generate discretized models;
  /* iterative loop */
  until convergence {
    generate linearized models;
    formulate system of linear equations;
    solve the system and find $x^{n+1,m+1}$;
    update $x^{n+1,m}$;
    m++;
  }
  t = t+h;
  update $x^n$;
  n++;
}
```

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

The generation of matrix entries for constant and linear dynamic elements is performed on one processor, 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. 2) and the system of linear equations can be solved.

The presented parallelization of equation formulation process is implemented in the simulator Alecsis [9]. It is a mixed-signal and mixed-domain simulator with proprietary hardware description language AleC++ [10] capable for modeling and simulation of complex systems containing different kinds of devices and subsystems [11]. 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 Beowulf cluster using MPI routines is shown in Fig. 3. The described parallel equation formulation is implemented using master-slave algorithm [2]. In this algorithm calculation of matrix entries for nonlinear circuit elements per time and per iteration is distributed to multiple cluster nodes (slave nodes) and they are calculated simultaneously. At the same time master node calculates matrix entries for specific number of nonlinear elements as well as for constant and linear time dependent elements. In order to minimize communication between cluster nodes, necessary data structures for all elements of the circuit are generated on all nodes simultaneously during compilation of the simulation model. In that way all cluster nodes have the information necessary to generate matrix contributions for all elements. To achieve equal load of all nodes, each node of the cluster performs equation formulation and calculation of matrix entries for equal number of nonlinear circuit elements. After generation of entries (for all elements) on one slave, they are sent to the master node using appropriate MPI routines for exchanging data (Fig. 3).

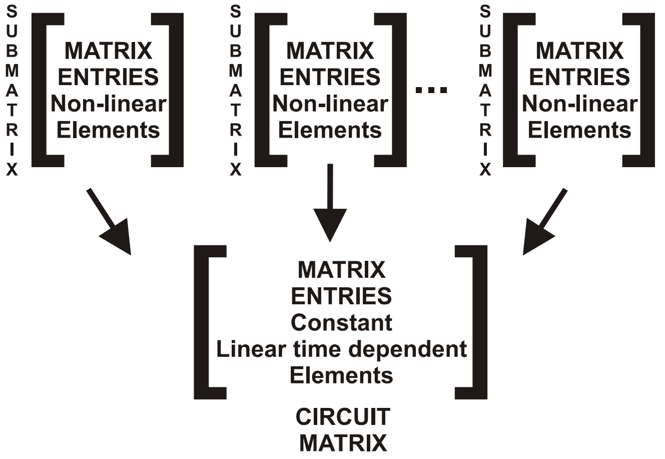
SUBMATRIX
MATRIX ENTRIES Non-linear Elements
SUBMATRIX
MATRIX ENTRIES Non-linear Elements
...
SUBMATRIX
MATRIX ENTRIES Non-linear Elements
MATRIX ENTRIES Constant Linear time dependent Elements
CIRCUIT MATRIX

Fig. 2. *Parallelization of equation formulation for nonlinear analog circuit elements*

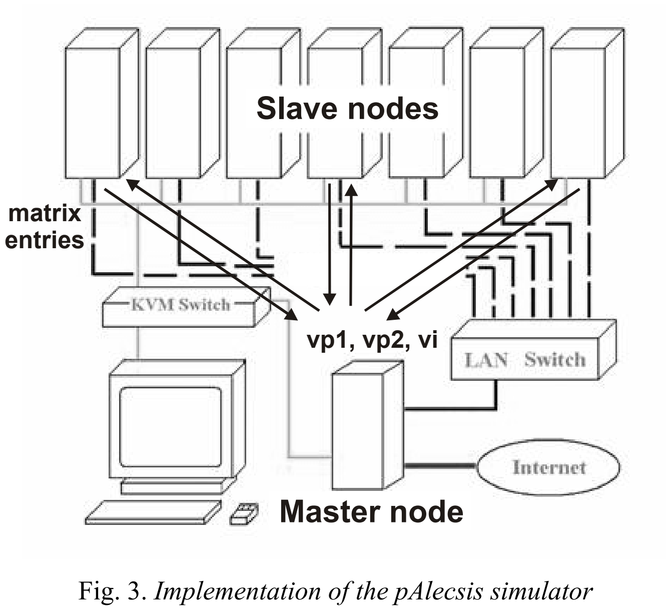

Fig. 3. *Implementation of the pAlecsis simulator on a Beowulf cluster*

When the master node receives matrix entries from all slaves, it flushes them to the circuit matrix and performs one iterative 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 by *vp1*, *vp2* and *vi*, respectively in Fig. 3. Appropriate MPI routines for transferring data are used to send and receive these vectors.

3. SIMULATION PERFORMANCES ANALYSIS

Performances of the implemented parallel simulator can be analyzed using simulation speedup. If parallel simulation executes on N single processor cluster nodes, simulation speedup is defined as:

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

The structure of Beowulf cluster used to perform simulations is given in Table 1.

Table 1. *Beowulf cluster structure*

Master node	PC Pentium IV, 2.4GHz, 1GB RAM, 240GB HDD
Slave nodes	4 X PC Pentium IV, 2.4GHz, 512GB RAM, 80GB HDD
LAN	1Gbit Ethernet

Fig. 4 shows parallel simulation speedup obtained on 2 cluster nodes for different circuit complexity. The circuit size is expressed in number of MOSFETs. As one can see, the implemented parallel simulation algorithm comes in effect (reduces simulation time, i.e. increases simulation speedup) for bigger circuits. Actually, in this case the time necessary to calculate matrix entries for all elements per time and per iteration exceeds time needed to calculate matrix entries on slave nodes and send them to the master node. For such circuits the parallel simulation on the cluster is faster than the simulation on a single processor workstation.

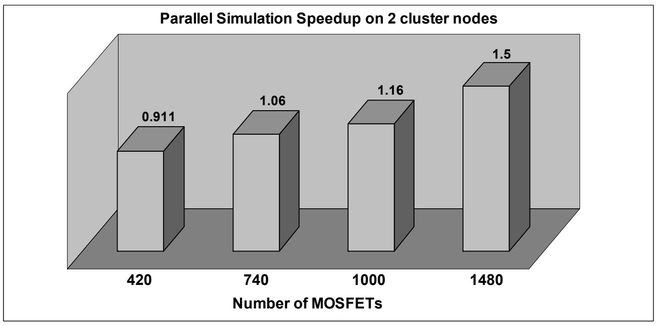

Fig. 4. *Parallel simulation performances for various circuit complexity and 2 cluster nodes*

Parallel simulation speedup for various number of cluster nodes involved in the simulation is shown in Fig. 5. The simulated circuit contains 1,000 MOSFETs. As it is shown, the simulation speedup on 4 cluster nodes is lower than on 3 cluster nodes. The decreased speedup relies on longer time needed for communication between the master node and higher number of slave nodes for this particular circuit size. Therefore, in order to achieve optimal speedup one needs to trade-off between the number of nodes and the circuit size.

Moreover, the parallel simulation performances are tested for various message sizes between the slaves and the master. The case of short messages was considered when each matrix entry is sent in a separate message. It gives many short messages that are sent from the slaves to the master. In the case of long messages, all matrix entries generated on one slave are sent at once, as one big message. Table 2 gives simulation runtimes for both short and long messages for the circuit with 80 MOSFETs. Obviously, sending less number of long messages gives much shorter simulation runtime than sending of many short messages. When one uses longer communication messages, the time necessary to setup the communication between the nodes is lower than if there are many short messages.

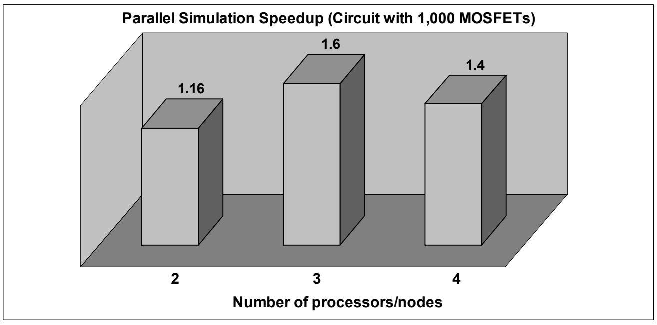

Fig. 5. *Parallel simulation performances for various numbers of cluster nodes*

Table 2. *Parallel simulation performances for different message sizes*

Type of messages	*Simulation Runtime (2 cluster nodes & 80 MOSFETs)*
Short messages	573.26 s
Long messages	5.3 s

4. CONCLUSION

This paper describes the implementation of parallel electronic circuit simulator on Beowulf cluster. The simulator introduces parallelization in equation formulation phase for nonlinear analog elements. It enables to distribute calculation of matrix entries for nonlinear circuit elements across different nodes of the cluster. Therefore, the time needed for equation formulation decreases, reducing the overall simulation time. Performances of the implemented parallel simulator are explored for different aspects of simulation. The simulation speedup for various circuit complexity is studied. The influence of communication overhead between the nodes to simulation speedup for different number of cluster nodes is analyzed. The relation between the size of communication messages and simulation runtime is considered, as well.

ACKNOWLEDGMENT

The work presented in this paper was supported by Serbian Ministry of Science and Environment Protection within the project with code number TR-6108B.

REFERENCES

[1] Sterling, T., *Beowulf Cluster Computing with Linux*, MIT Press, 2001.

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

[3] Gropp, W., Lusk, E., and Thakur, R., *Using MPI-2: Advanced Features of the Message-Passing Interface*, MIT Press, 1999.

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

[5] 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, Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications*, pp. 601-613, Vol. 45, No. 6, June 1998.

[6] 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.

[7] Dimitrijević, M., Anđelković, B., Savić, M., Litovski, V., “Gridification and parallelization of electronic circuit simulator”, in Proc. *INDEL* 2006, 2006, pp. 95-100

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

[9] 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.yu/projects/Alecsis/alecsis.htm

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

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

Sadržaj – Proces simulacije savremenih složenih integrisanih kola je računski i resursno zahtevan što dovodi do dugih vremena simulacije. Paralelna simulacija na Beowulf klasteru personalnih računara je efikasan način da se udovolji ovako visokim zahtevima. U ovom radu predstavljen je algoritam za paralelnu simulaciju koji paralelizuje fazu formulacije jednačina i koristi više radnih stanica za istovremeno izračunavanje elemenata matrice za nelinearne analogne elemente kola. Performanse implementiranog simulatora su testirane i analizirane za kola različitih veličina, različit broj nodova klastera i veličina poruka između nodova.

IMPLEMENTACIJA I ANALIZA PERFORMANSI PARALELNOG SIMULATORA KOLA NA BEOWULF KLASTERU

Bojan Anđelković, Vančo Litovski, Predrag Petković