Reprint from Proceedings of 11th IEEE/ACM/SCS Workshop on Parallel and Distributed Simulation (PADS'97), Lockenhaus, Austria, June 10 - 13, 1997

A New Approach for Partitioning VLSI Circuits on Transistor Level

Norbert Fröhlich, Rolf Schlagenhaft, and Josef Fleischmann

Institute of Electronic Design Automation,
Technical University of Munich, 80290 Munich, Germany
Email: {froehlich,schlagenhaft,fleischmann}@regent.e-technik.tu-muenchen.de

Abstract

For parallel simulation of VLSI circuits on transistor level a sophisticated partitioning of the circuits into subcircuits is crucial. Each net connecting the subcircuits causes additional communication and computation effort. As the slave processors simulating the subcircuits advance synchronously in time, the computation effort for each subcircuit should be approximately the same. In this paper a new approach for partitioning VLSI circuits on transistor level yielding a low number of interconnects between the subcircuits and balanced subcircuit sizes is presented. Simulation of industrial circuits using this partitioning is up to 41% faster than with other known partitioning approaches for parallel analog simulation.

1 Introduction

When designing VLSI circuits, time critical parts have to be simulated on transistor level. At Siemens research laboratories, Munich, (a cooperation partner of the Institute of Electronic Design Automation) a major simulation application on transistor level is the validation of critical paths with more than 10,000 transistors. These critical paths are extracted from dynamic memories and digital circuits.

Due to the increasing size of VLSI circuits and the required accuracy and reliability of the simulation, the computational effort for circuit simulation is rising. To keep pace with this trend the applied methods for analog simulation like implicit numerical integration are traditionally vectorized and implemented for supercomputers.

Nowadays the hardware environment is changing. The performance of modern workstations approaches those of small vector computers. As there is a much greater market for workstations than for supercomputers, parallel architectures built of ordinary workstation hardware are very cost effective.

Therefore, the analog simulator TITAN, which was developed at Siemens' research laboratories, was parallelized for distributed computers. The applied domain decomposition techniques [WZ96] split the circuit on transistor level into domains. Each domain is calculated by a separate process and the entire problem can be solved in parallel. The boundaries of the domains are synchronized by a master process running sequentially. In TITAN, numerical integration is performed by the multi-level Newton method with latency. For a test circuit with a highly parallel structure, a close to linear speedup of 8 on 8 processors is achieved. This demonstrates that there is only little overhead for parallel execution of the simulation.

For parallel simulation a partitioning is required which yields balanced domain, i.e. partition, sizes and only few interconnects. Each net interconnecting the partitions causes communication between the slave processes and the synchronizing master process. The master process calculates the interconnection network between all partitions, i.e. subcircuits, and each additional interconnect net rises the computation effort for this time critical calculation. As the slave processes simulating the subcircuits advance synchronously in time, maximum progress is achieved, if all slave processes have the same simulation load. This requires subcircuits of approximately the same size.

1.1 Previous Work

One of the first published algorithms for partitioning on transistor level is *Node Tearing* [SVCC77]. Further early approaches are some cluster algorithms as building *DC-Connected Components* and *Strongly Connected Components* or *Diagonal Dominance Norton Partitioning* [DOR85]. There are also some approaches combining the early methods with the classical *Fiduccia-Mattheyses method* [FM82] and *hierarchical methods*, which exploit the subcircuit information contained in the circuit description [CBE86][JRP91][KKN94]. Most of these approaches are only applicable on MOS-technology-circuits, others are highly focussed on the simulation tool they are developed for.

While developing TITAN, initially a simple straightforward *Clustering Algorithm* (CLA) for partitioning was implemented. Starting from an input voltage source, adjacent elements are grouped together until a specified partition size is reached. This approach is very fast but usually it has to be repeated several times

with different parameters to obtain an acceptable result.

In this paper a new concept for partitioning on transistor level is presented. In a first step the circuit is prepared in a way that arbitrary partitioning algorithms can be applied. Thus, it is possible to use the excellent gate level partitioning system SEAPART [RDJ94] for splitting circuits on transistor level. SEAPART is based on the *Analytical Partitioning Method* (APM) and yields subcircuits with a small number of interconnect signals and with very well balanced sizes.

In the next section the partitioning approach is described in detail. In Section 3 the partitioning results of the analytical partitioning method are compared with the results of the simple straightforward clustering algorithm. Simulation performance results for partitioning of industrial VLSI circuits with both methods are presented.

2 Partitioning Approach for Circuit Simulation

SEAPART is designed for partitioning large circuits to be implemented in multi-chip architectures and operates on a graph which normally represents a circuit on gate level. For applying SEAPART on transistor level the tool ATLAS was developed, which prepares the circuit description before partitioning and uses the SEAPART result after partitioning for creating a partitioned circuit description in TITAN's input format.

2.1 Circuit Conversion and Preparation for Partitioning

TITAN reads the circuit description in SPICE format [Nag75], which allows complex constructs containing a numerous variety of elements like transistors, capacitors, inductors, driven sources, etc.. SEAPART operates on an undirected graph, which is created from the PROUD input format containing only modules and the connections between them. Hence, a conversion is necessary when applying SEAPART to a circuit description given in SPICE format. Since there are some restrictions for the parallel simulation resulting in nets, which must not be cut, and some elements, which have to be grouped together, a simple 1:1 conversion of elements to modules is not desirable. ATLAS interprets the computational effort for the simulation of a circuit element as its corresponding physical module size. Thus, SEAPART's facility of creating balanced partitions in terms of area size can be exploited. Within SEAPART it is not possible to prevent certain nets from being cut. Therefore, these nets need special consideration before starting partitioning with SEAPART.

2.1.1 Coupled Modules

For simulation, controlled sources need information about their controlling elements. If the partitioning system assigns them to different partitions, the simulation system has to provide the controlled element's partition with information about the controlling element. ATLAS creates *coupled modules* to avoid this. This guarantees an assignment of the controlling element and the controlled source to the same partition by removing the controlling element from the netlist and packing it into the controlled module. Its pins become pins of the controlled module and the partitioning tool operates only on one module, which cannot be split. Mutual inductors are coupled in the same way.

2.1.2 Avoiding Illegal Cuts, Meta Modules

The nets between subcircuits, i.e. the interconnect nets, are cut while partitioning. The parallel analog simulator TITAN handles these nets as I/O-pins for the related subcircuits and connects virtual voltage sources to these I/O-pins, as shown in Figure 1.

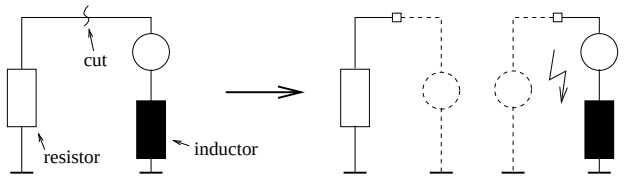

Figure 1: Cut Net

This is an important point, because loops in the circuitry containing only voltage sources and inductors cannot be simulated. It has to be ensured, that no paths from an I/O-pin to ground consisting only of voltage sources and inductors are generated, because the added virtual voltage sources are also connected to ground and thus, a loop invalid for simulation will be created as shown on the right side of Figure 1. To avoid this, we try to combine a critical module with an uncritical module. Thus, the combined *meta module* is no longer critical. If a critical module has a net connected in series to an uncritical module, we restrict cutting the connection by marking this net critical. Next, all elements connected to a critical net are packed into a meta module and the connecting net is eliminated from the netlist as shown in Figure 2. Thus, the partitioning tool operates on a single module and cannot cut this critical net.

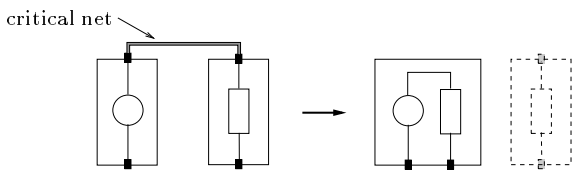

Figure 2: Building Meta Modules

The easiest way to avoid forbidden cuts would be to mark all nets connected to critical modules critical, but this would result in large meta modules. For fine granularity and well balanced partition sizes large meta modules have to be avoided.

The example in Figure 3 shows an input voltage source connected to net A and B.

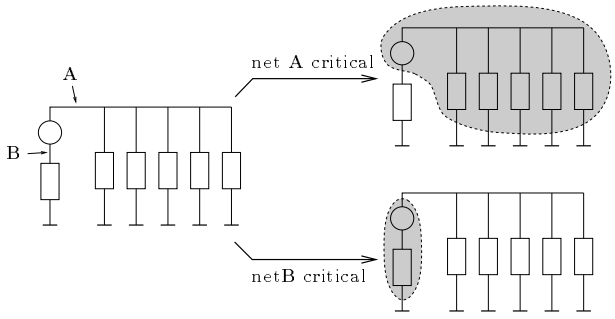

Figure 3: Example for Marking Critical Nets

In general, this source is connected to a lot of elements, maybe at net A. If A is marked critical, all the connected elements are packed into one meta module and a large meta module is created. But if it is detected, that B is connected in series to an uncritical element, e.g. a resistor, it is sufficient to mark net B critical and only a small meta module including the source and the resistor is created.

For small meta modules, but correct handling of the simulator restrictions it is crucial to find uncritical modules connected in series to a critical module and thus, to mark the right nets. This is performed by an algorithm examining each net of a critical module whether it is connected to another critical module. If there are nets of a critical module, which are connected only to uncritical modules, the net with the least connections is taken and marked critical. Now the critical module is no longer regarded critical, because in a later step it is combined with the selected net's uncritical modules to a meta module.

Preparing the netlist given on transistor level for partitioning in a way that the partitioning tool is totally unconstrained in distributing the elements has the great advantage, that arbitrary partitioning algorithms can be applied.

2.1.3 Creating Partial Circuit Descriptions

After partitioning, ATLAS uses the partitioning result and the original SPICE circuit description to create subcircuit descriptions in SPICE format.

This requires the correct unpacking of coupled modules and meta modules. Cut nets are detected and noted as I/O-pins for the subcircuits. As the subcircuit simulations are synchronized by the master process, a master description with special references to the cut nets is created additionally to the subcircuit descriptions.

2.2 Analytical Partitioning Method implemented in SEAPART

For partitioning, SEAPART divides a circuit iteratively until each subcircuit, i.e. partition, fits into a single chip of the multichip architecture [RDJ94], as shown in Figure 4.

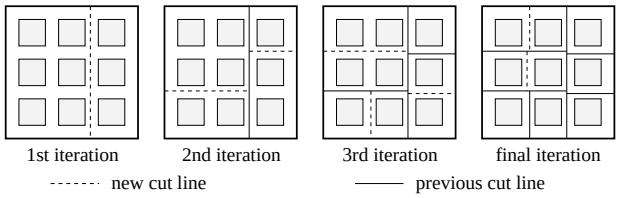

Figure 4: Partitioning Iterations

In each bipartitioning iteration three steps are performed. First, a 2-dimensional embedding of the modules with a minimal length of the connections between them is calculated. Second, each partition which is larger than the target partition size is initially partitioned by a rectangular cut, as demonstrated by the dashed lines in Figure 4. The initial cut uses an adapted ratio cut objective [HK91] yielding well balanced partitions in terms of module area. Third, the number of cut nets of the current partitioning is reduced by moving modules between all partitions applying a multi-way ratio cut method with a problem-specific objective function.

These three steps, calculation of the embedding, initial partitioning, and moving of modules, are done in each iteration, until the number of desired partitions is obtained. Calculation of the embedding and moving of modules is done over the whole circuit, which keeps the global view of the problem.

Compared to other partitioning tools SEAPART's results are excellent in terms of low number of cut nets and well balanced partitions [RS95], which both are crucial for efficient parallel simulation on transistor level.

3 Experimental Results

The performance of the new partitioning approach on transistor level is evaluated on the large industrial circuits shown in Table 1, which are state-of-the-art industry designs. Each circuit's size is characterized by the number of MOSFETs contained. *Industry1* is the critical path of a Dual Port RAM, whereas *Industry2* is the critical path of a 16 Megabit DRAM design.

All experiments are performed by a SGI Power Challenge (8 × R8000/90MHz). The parallel circuit simulator TITAN [WZ96] of Siemens' research laboratories is used for the simulations. Data communication

Circuit		1 Proc.	4 Proc.		8 Proc.	
			CLA	APM	CLA	APM
Industry1	real simulation CPU-time (h:m:s)	2:28:12	0:50:03	0:48:01	0:28:37	0:27:13
(13,800 MOS / 4,220 nets)	speedup	–	2.96	3.09	5.18	5.45
	# interconnect nets	–	343	294	371	361
Industry2	real simulation CPU-time (h:m:s)	12:31:36	5:47:26	4:51:33	4:17:15	2:31:50
(34,800 MOS / 18,161 nets)	speedup	–	2.16	2.58	2.92	4.95
	# interconnect nets	–	301	215	415	341

Table 1: Partitioning Results for the Clustering Algorithm (CLA) and the Analytical Method (APM)

between the CPUs is performed by the public domain software package *Parallel Virtual Machine* PVM.

The main objective for the partitioning is to achieve good speedup for parallel simulation. A low number of interconnect nets reduces the communication between the slaves and the master. Furthermore, the master process, which solves the equations describing the interconnect system, is accelerated. Table 1 compares the performance of the clustering partitioning algorithm CLA and the analytical partitioning algorithm APM. The table contains the real CPU-time for the simulation, speedups and the number of interconnect nets.

The analytical algorithm outperforms the clustering approach. Especially for eight partitions it obtains a low number of interconnect nets and increases the performance of parallel simulation. For *Industry2* the simulation time is reduced by 41%.

Furthermore, while developing the conversion tool to adapt the analytical method for partitioning on transistor level a small circuit with lots of controlled sources was created for testing purposes. The APM method handles this circuit without problems, whereas the CLA method is not able to find a valid partitioning for simulation.

4 Conclusions

In this paper we have presented a new approach for partitioning large electronic circuits on transistor level. Essential requirements for parallel simulation are a low number of interconnect nets and well balanced partitions. Therefore, we developed a conversion tool to apply an excellent analytical partitioning method on transistor level.

The efficiency of our new partitioning approach is demonstrated on industrial VLSI circuits. Compared to other known methods parallel simulation time is reduced up to 41%.

Acknowledgement

The authors would like to thank Prof. K. Antreich for continuous support, H. Spruth, P. Loibl and J. Spaderna for fruitful discussions and doing parts of the implementation of ATLAS. Furthermore we would like to thank B. Riess, who developed the partitioning tool SEAPART, and our cooperation partners at Siemens research laboratories U. Wever and Q. Zheng, who helped us to apply our partitioning knowledge to current industrial problems.

References

[CBE86] Paul Cox, Richard Burch, and Berton Epler. Circuit partitioning for parallel processing. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 186–189, 1986.

[DOR85] P. Debefve, F. Odeh, and A.E. Ruehli. Waveform techniques. In A.E. Ruehli, editor, *Circuit Analysis, Simulation and Design, Part 2*, volume 3 of *Advances in CAD for VLSI*, chapter 8, pages 41–127. North-Holland, 1985.

[FM82] C.M. Fiduccia and R.M. Mattheyses. A linear-time heuristic for improving network partitions. In *ACM/IEEE Design Automation Conference (DAC)*, volume 19, pages 175–181, 1982.

[HK91] L. Hagen and A. B. Kahng. Fast spectral methods for ratio cut partitioning and clustering. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 10–13, 1991.

[JRP91] W. John, W. Rissiek, and K.L. Paap. Circuit partitioning for waveform relaxation. In *European Conference on Design Automation (EDAC)*, pages 149–153, Feb 1991. Amsterdam.

[KKN94] T. Kage, F. Kawafuji, and J. Niitsuma. A circuit partitioning approach for parallel circuit simulation. *IEICE Transactions on Fundamentals*, E77-A(3):461–466, 1994.

[Nag75] L. Nagel. *SPICE2: A computer program to simulate semiconductor circuits.* Ph. D. dissertation, Univ. of California, Berkeley, 1975.

[RDJ94] Bernhard M. Riess, Konrad Doll, and Frank M. Johannes. Partitioning very large circuits using analytical placement techniques. In *ACM/IEEE Design Automation Conference (DAC)*, pages 646–651, June 1994.

[RS95] Bernhard M. Riess and Andreas A. Schoene. Architecture driven k-way partitioning for multichip modules. In *European Design and Test Conference (ED&TC)*, pages 71–75, Paris, France, March 1995.

[SVCC77] Alberto Sangiovanni-Vincentelli, Li-Kuan Chen, and Leon O. Chua. An efficient heuristic cluster algorithm for tearing large-scale networks. *IEEE Transactions on Circuits and Systems CAS*, CAS-24(12):709–717, Dec. 1977.

[WZ96] U. Wever and Q. Zheng. Parallel transient analysis for circuit simulation. In *Proceedings of the 29 Annual Hawaii International Conference on System Sciences*, volume 1, pages 442–447, 1996.