

USING NODE REPLICATION TO IMPROVE CIRCUIT'S PARTITION IN DISTRIBUTED LOGIC SIMULATION

Amar Guettaf, Pirouz Bazargan-Sabet

Université Pierre et Marie Curie (Paris VI), Laboratoire LIP6/ASIM
Tour 55-65, 2ème étage - 4, place Jussieu - 75252 Paris Cedex 05, France
E-mail: Amar.Guettaf@lip6.fr

KEYWORDS

Partitioning, Distributed simulation, Node replication, VLSI, Discrete event

ABSTRACT

Distributed simulation represents an attractive and smart way of improving the verification speed of large VLSI circuits. Unfortunately, this inexpensive approach suffers from the low performance of the communication networks used to connect local workstations. In this paper, we present a partitioning algorithm that attempt to find a suitable balance between the communication and the execution load to enhance the speedup. The main features of this method are the use of logic replication to reduce the communication overhead and a realistic cost function that takes into account the activity of signals. Signals' activity can be obtained through a probabilistic evaluation. The efficiency of this algorithm has been evaluated using a prototype simulator that implements a conservative synchronization method.

1 INTRODUCTION

Distributed simulation seems to be an appropriated solution to make the verification of high complexity circuits be faster. Several processors collaborate to perform the simulation of a single system. In this approach, which exceeds the technical bounds of a single machine, a great amount of memory, disk space and other computational resources are provided to enhance the simulation's speed.

The efficiency of a distributed simulation can be measured using a **speedup** value. The speedup is the ratio of the time spent in achieving some process using a standard program by the time required by an enhanced program. In distributed simulation, the speedup depends on 4 main parameters: (1) the intrinsic parallelism of the circuit, (2) the number of systems involved in the simulation process, (3) the simulator's synchronization strategy and (4) the efficiency of the partitioning.

The most popular method in logic simulation is the event-driven trace-selective approach. An event is associated with the transition of a signal and the value of a signal is re-evaluated each time an event occurs on, at least, one of its inputs. In addition to the simulation method, in a distributed simulator, a synchronization strategy must be adopted to maintain the global coherence of the system. Most of the researches carried out in this domain concern this last issue and few works have been done in partitioning applied to distributed simulation.

Partitioning is used in various domain to split a complex design into multiple smaller blocks. Making the partition, some important topics have to be considered: the respect of a certain balance between the different blocks in terms of execution and the reduction of the communication through the network. Finding an optimal partition for a given circuit is a tricky task due to the NP-completeness of the problem (Johnson and Garey 1979). A large number of heuristics has been proposed. In VLSI design tools, the move-based approaches have been largely adopted.

In this paper, we present a partitioning algorithm that belongs to the move-based approaches. We propose an improvement of the basic Fiduccia and Mattheyses algorithm by calling logic replication to exchange communication load against execution. Also, the concept of signal activity has been introduced to build a more realistic cost function for the event-driven simulation method. The next section gives a brief description of the original partitioning algorithm as well as the logic replication method. The concept of signal activity and the cost function is depicted in section 3. Sections 4 gives some results obtained using the proposed method within a distributed simulator. In section 5, a conclusion is given and future works are pictured.

2 MOVE BASED PARTITIONING METHODS

Despite the diversity of the domains where the partitioning is applied (place & root, testability, system prototyping, ...), it is possible to give a unique formulation for this problem. A VLSI circuit may be represented by a hyper-graph $H(V, E)$. V is the set of vertices representing the internal nodes and E the set of hyper-edges representing the internal signals that connect the nodes. The partitioning problem consists in defining k subsets - or blocs - V_1, V_2, \dots, V_k of V such as:

$$\bigcup_{i=1}^k V_i = V \quad \text{and} \quad \bigcup_{i=1}^k \left(\bigcup_{j=1, j \neq i}^k (V_i \cap V_j) \right) = \emptyset$$

The aim is to reach an optimized partition for some specific application. As it is unreasonable to run the application for each intermediate solution, a cost function is defined to give a simplified model of the target application. The cost function is supposed to have a behavior as close as possible to the target application. The cost function traditionally used to measure the quality of a partition is the **cutsize**. It represents the number of signals that cross the partition.

One of the most known algorithms, in partitioning, has been proposed by Kernigan and Lin (Lin and

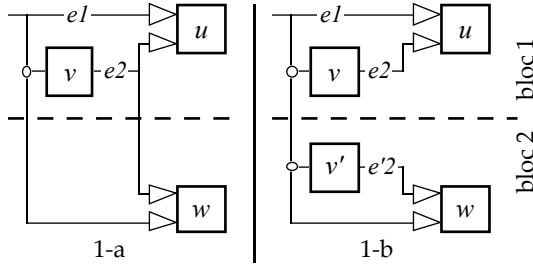


Figure 1: Node replication to improve the cutsizes

Kernigan 1970): a graph bisection technique. It starts with a random initial partition and uses pairwise swapping of vertices between partitions to get iteratively into a more efficient solution.

An improvement of the K-L algorithm which reduces the time complexity to $O(n)$ (where n is the number of vertices) was proposed by Fiduccia and Mattheyses (Mattheyses and Fiduccia 1982). In this algorithm, only one vertex is moved in a single move that allows the handling of unbalanced partitions. The concept of cutsizes is extended to hyper-graphs and a well suited data structure is defined to make a fast selection of the vertices to be moved.

A vertex replication technique has been proposed by Kring and Newton (Newton and Kring 1991) for circuit partitioning applied to mapping on FPGAs. The cutsizes is reduced by making some vertices be replicated in two or more partitions. Figure 1-a shows the partition of a circuit without vertex replication. However, when the node v is replicated, as in Figure 1-b, the cutsizes is reduced. When a node is replicated, it is present in both sub-circuits and its outputs do not contribute to the cutsizes.

Once a node has been replicated, it tends to remain so and nets connected to it remain in both sub-circuits. This may reduce the possibility of further improvements of the partition. For this reason, the number of replicated nodes must be limited to achieve an optimal partition. The experience shows that node replication can significantly reduce the cutsizes without increasing sensibly the size of the initial circuit.

3 THE PROPOSED IMPROVEMENT

As mentioned in the above section, the original K-L algorithm as well as the F-M and the K-N methods define a cost function based on the number of signals that cross the partition. However, in a distributed simulation, the cutsizes does not give a realistic representation of the simulation method. In particular, the execution load is not taken into account. Moreover, in K-N, to reduce the cutsizes by node replication, a maximum number of replicated nodes must be settled. Therefore, an inappropriate replication rate may lead to a sub-optimal simulation speed.

3.1 The cost function

In the node replication method, finding out the optimal value for replication rate may be hazardous, and the execution load of replicated nodes does not affect the

cost function. Clearly, in this method, the cost function has been defined to minimize the communication load and would look like:

$$\sum_i \|c_i\|$$

Where c_i is a signal crossing the cut and $\|c_i\|$ the number of blocs that read the signal c_i .

In fact, the key point underlying the replication method, is to obtain a smaller communication load by increasing the execution load. Knowing that sending a message over the network requires much more time than an execution, we propose a cost function that attempt to reduce the global load of a simulation: the execution and the communication load. For a conservative distributed simulation, the following cost function seems to be more convenient:

$$\text{Max}_k \left(\sum_i F_{out}(x_i) \right) + R_{ce} \sum_j \|c_j\|$$

$F_{out}(n)$ is the fanout of a node n . x_i are the internal nodes of a bloc and c_j are the hyper-edges that cross the cut. R_{ce} is the ratio of the time required for one message to be transmitted from one bloc to another by the time needed for the execution of one node.

The proposed algorithm is based on the K-N partitioning algorithm but uses a more realistic cost function. However, in the above cost function all signals continue to have the same weight. Actually, all the signals do not generate the same number of messages on the interface of the partitions and communications due to "silent" signals are not as important as the messages generated by "active" signals. Yet, a more appropriate cost function, that exploits the concept of signal's activity, is conceivable.

3.2 The concept of signal's activity

These recent years, a great interest has been given to the concept of signal's activity, in VLSI design. The most important application of this concept is certainly in the evaluation of the power consumption. In a CMOS digital circuit, the most part of the power consumption comes from the transitions of signals. Therefore, knowing the mean number of transitions of signals per second - or the transition density - it is possible the estimate of power consumption. To evaluate the transition densities, probabilistic approaches seem to be very promising since the calculation is done in a single-pass symbolic simulation.

However, these approaches have to face the problem of correlation between signals which can impair the probabilistic evaluation. We have developed a method based on symbolic simulation to detect potential sources of correlation. For each node, statistically independent inputs are identified and the Boolean expression of the node is expanded to these signals. Then, the transition density of signals is calculated in a second pass. The method of calculation is detailed in Dunoyer *et al.* 1995 and 1996. Briefly, the calculation uses the concept of

Boolean difference and computes the transition density of a signal from the transition density and the probability of its inputs (Najm 1991):

$$D(s) = \sum_j D(e_j) P\left(\frac{\partial s}{\partial e_j}\right)$$

Where s is the output signal of the node, $D(s)$ is the transition density of the signal s , e_j is an input of s , $\partial s / \partial e_j$ is the Boolean difference of s in regard of e_j and $P(F)$ the probability of the Boolean function F to take the value one.

However, in the partitioning problem, unlike the power evaluation tools, the activity of a signal is not directly related to its transition density. In a distributed event-driven simulator, the value of a signal is re-evaluated whenever, at least, one of its direct inputs has received an event or a transition. Thus, we propose a proper definition of the concept of **activity** of a signal, well suited to the partitioning problem. For a node i the activity A_i is defined as:

$$A_i = \sum_j D(e_{ij})$$

Where e_{ij} is a direct input of the node i and $D(x)$ is the transition density of a signal x .

Using the concept of signal's activity the proposed cost function becomes:

$$\text{Max}_k \left(\sum_i A(x_i) \right) + R_{ce} \sum_j A(c_j) \cdot \|c_j\|$$

4 SIMULATION ENVIRONMENT & RESULTS

A distributed simulator has been developed comprising two main parts: a central task and several sub-simulators. The central task reads the hierarchical description of the circuit produced by the partitioning tool. Each bloc is distributed to one simulation task and the communications between sub-simulators are defined. The simulation process can start once the test pattern file is loaded. The simulator implements a conservative synchronization strategy.

To measure the speedup, the time spent in a sequential simulator was compared to the time required by the distributed simulation. It has been measured on a

set of sequential logic circuits from the ISCAS89 benchmark suite. The distributed simulator has been implemented on a network of Sun ULTRA SPARC 1 connected through an Ethernet 10Mbits/sec.

Table 2 shows the gain of speedup on two machines using the proposed partitioning method. It should be noted that the results presented in this section take only into account the simulation time excluding the partitioning and the initialization time.

5 CONCLUSION AND FUTURE WORKS

A new partitioning method based on a realistic cost function has been investigated resulting in a reasonable speedup for a distributed simulator. A certain gain may be obtained over conventional partitioning technique even with a conservative synchronization method. This algorithm is able to reduce significantly the amount of messages on the network, but the speedup remains smaller than what would be expected. The difference can be explained by: (1) the usage of heuristics that may fall into local minima, (2) the overhead introduced by the synchronization and (3) the network's load.

Future works will focus on implementing other synchronization strategies such as Time Warp to study the relationship between synchronization and partitioning and its impact on the speedup. Also, the simulator will be implemented on a high performance 1 Gbyte/sec. network.

REFERENCES

- Dunoyer J., N. Abdallah, P. Bazargan-Sabet, 1995, "A New Generation of Digital CAD Tools Based on Probability", *27th Southeastern Symposium on System Theory*, pp. 348-352, 1995.
- Dunoyer J., N. Abdallah, P. Bazargan-Sabet, 1996, "A Symbolic Approach in Resolving Signal's Correlation", *29th Annual Simulation Symposium*, pp. 203-211.
- Johnson D.S., M. Garey, 1979, "Computer and Intractability: A Guide to the Theory of NP Completeness", San Fransisco, CA: Freeman.
- Lin S., W. Kernigan, 1970, "An Efficient Heuristic Procedure for Partitioning Graphs", *Bell System Technical Journal*, Vol 49, pp. 291-307.
- Mattheyses R.M., C.M. Fiduccia, 1982, "A Linear Time Heuristics for Improving Network Partitions", *Proceedings of the 19th Design Automation Conference*, pp. 175-181.
- Najm F., 1991, "Transition Density a Stochastic Measure of Activity in Digital Circuits", *28th Design Automation Conference*, pp. 644-649.
- Newton A.R., C. Kring, 1991, "A Cell-Replicating Approach to Mincut-Based Circuit Partitioning", *International Conference on Computer Aided Design*, pp. 2-5.

circuit	# of nodes	speedup by F-M	replicated nodes (%)	S.up new approach	gain in speedup
s5378	3 137	0.14	13.4	0.16	21%
s9234	6 019	0.17	12.1	0.20	19%
s13207	9 227	0.37	8.8	0.49	33%
s15850	10 840	0.38	7.2	0.45	20%
s35932	19 841	1.22	6.5	1.48	22%
s38584	22 105	1.31	4.6	1.63	25%
s38417	25 451	1.41	4.7	1.73	23%

Table 2: Improvement of the speedup