

**Comparing Edge-cuts to
Communications Volume in
Parallel VLSI Logic Simulation**

**Y. Chen
B. Noble
R. Chamberlain**

Y. Chen, B.L. Noble, and R.D. Chamberlain, "Comparing Edge-cuts to Communications Volume in Parallel VLSI Logic Simulation," in *Proc. of the 8th IASTED Int'l Conf. on Parallel and Distributed Computing and Systems*, October 1996, pp. 481-484.

Computer and Communications Research Center
Washington University
Campus Box 1115
One Brookings Dr.
St. Louis, MO 63130-4899

Comparing Edge-cuts to Communications Volume in Parallel VLSI Logic Simulation

Yuhua Chen*, Bradley L. Noble†, and Roger D. Chamberlain*

*Department of Electrical Engineering
Washington University
St. Louis, Missouri
{yhc,roger}@ccrc.wustl.edu

†Department of Electrical Engineering
Southern Illinois University at Edwardsville
Edwardsville, Illinois
bnoble@siue.edu

Abstract

The execution speed of parallel VLSI logic simulation can be greatly impacted by the quality of the partitioning algorithm used to assign components to processors. Graph partitioning algorithms have been commonly used for this task; however, the figure of merit optimized by these classical algorithms (edge-cut) is not precisely what is needed for a good simulation partitioning. This paper compares the edge-cuts that result from a top of the line graph partitioning algorithm to the communications volume that exists in the corresponding parallel simulation execution.

Keywords: parallel logic simulation, graph partitioning

1 Introduction

In parallel logic simulation, it is very important that the circuit be properly partitioned so that the communication volume among the processors is minimized, while maintaining an acceptable degree of load balance [1]. If we convert the logic circuit into a graph, where each component (gate) of the circuit is a vertex and the connections among the components are the edges of the graph, we can use existing graph partitioning algorithms to map each node to a specific processor. The implicit assumption with this technique is that the measure of a good graph partitioning (its edge-cut) is also a reasonable measure of the quality of a circuit partitioning. This paper will explicitly test this assumption.

Here, we apply the METIS graph partitioning system authored by Karypis and Kumar [5, 6] to the logic simulation partitioning problem and compare the edge-cuts generated by the graph partitioning to the actual

communication volume present in the resulting simulation execution. The initial application focus of the METIS system was the solution of the linear system of equations $A\mathbf{x} = \mathbf{b}$ for sparse matrices. The system, however, is directly applicable to any classic graph partitioning problem, where one wishes to partition a graph into K equal parts while minimizing the number of edges crossing partition boundaries. The difference between classic graph partitioning and logic simulation is that in the classic problem, the quality of the partition can be determined by measuring the edge-cuts of the partition, while in logic simulation, the minimum edge-cut partition might be a bad partition in terms of simulation runtime. This is due to the fact that the performance impact of a particular partitioning is not only a function of the graph topology, but also a function of the logical structure of the VLSI system and the input stimulus used to drive the circuit. In short, the communication volume implicit in a given circuit partitioning is not necessarily a strict function of the edge-cuts.

In this paper, we quantitatively compare the edge-cuts from a good partitioning algorithm with measured communication volume from simulation execution. In addition, we investigate the implications of communication volume on the execution time of a synchronous VLSI logic simulator.

2 Experiment Description

The purpose of these experiments is to investigate the correlation between edge-cuts in the partitioned graphs and communication volume in the resulting simulation. Our interest in communication volume is due to the fact that it is independent of the time synchronization algorithm used to coordinate the parallel simulation. In addition, we investigate the impact of edge-cuts on execution time for a globally-clocked time synchronization

¹This material is based upon work supported by the National Science Foundation under grant MIP-9309658 and Southern Illinois University at Edwardsville.

Circuit	Component Count	Edge Count
s5378	3,042	47,332
s9234	5,866	84,762
s15850	10,470	423,716

Table 1: ISCAS-89 benchmark circuits

algorithm.

In our experiments, we use the ISCAS-89 benchmarks [2], whose sizes vary from tens of components to over 10,000 components. Data reported here is from the three circuits summarized in Table 1. The METIS multi-level graph partitioning package is used to partition the circuits and the circuits are simulated with random input vectors. To measure the communication volume, the partitioned circuit is simulated using a serial VLSI logic simulator that counts the signal transitions crossing partition boundaries [4]. In addition, the simulations are executed on a Sun SPARCcenter 2000 multiprocessor running a globally-clocked version of the VLSI logic simulator. We perform the following experiments:

- We partition the original circuits without structural modification.
- We partition the circuits after removing the clock signal and reconstruct the circuit by adding clock sources in every partition.
- We simulate the partitioned circuits using an instrumented serial logic simulator.
- We compare the total communication volume measured in the simulation with the edge-cuts obtained in the graph partitioning.
- We compare the edge-cuts between each pair of partitions with the traffic volume across the boundary.
- We investigate the impact of communication volume on runtime and compare the runtimes on the multiprocessor with those on a uniprocessor.

3 Experimental Results

In our first set of experiments, the circuits are directly translated into graph form (circuit components forming the nodes and fanout components connected via edges) and partitioned using METIS. The resulting partitioned circuits are simulated using the instrumented serial VLSI logic simulator. We measure the total communication volume and the edge-cuts crossing partition boundaries.

In this set of experiments, the communication volume is consistently higher than the edge-cuts. The primary cause of this discrepancy is the clock signal. Since

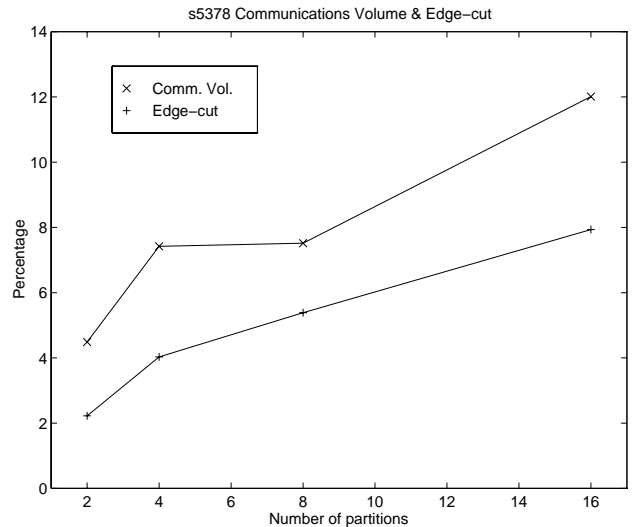


Figure 1: Circuit s5378

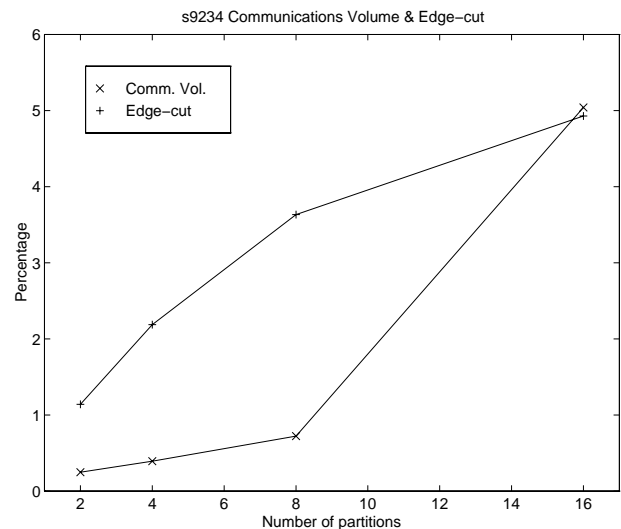


Figure 2: Circuit s9234

the clock signal is connected to every flip-flop, METIS tends to map these components to one or two partitions, effectively ignoring the other connections. However, during the logic simulation, such a partitioning will cause significant communication among the processors, which can significantly increase the execution runtime.

The solution is to treat the clock signal as a special case. The clock signal is removed from the circuit prior to translation into graph form. The circuit is partitioned as before, and the clock source is then replicated in each partition. This is a simplified version of the repeat message structure for large fanout nodes proposed by Briner et al. [3].

Once the clock signals are adequately dealt with, the results are very encouraging. The communication volume across processor boundaries is significantly reduced, especially for large numbers of partitions. Fig-

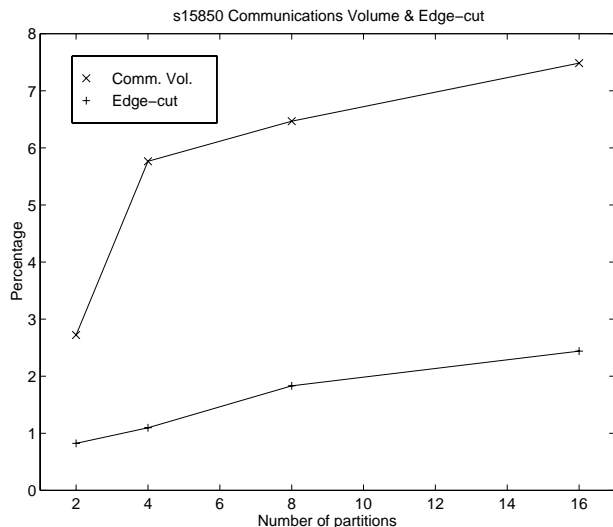


Figure 3: Circuit s15850

ures 1 through 3 compare the measured communications volume to the graph edge-cut for 2-way, 4-way, 8-way, and 16-way partitionings of the three benchmark circuits. To make communications volume and edge-cuts comparable, they are normalized as follows: edge-cuts are expressed as a percentage of total edges and communication volume is expressed as a percentage of total signal transitions (the communication volume that would exist if all the components were on distinct processors).

We can draw a number of conclusions from these plots. First, there clearly is a strong correlation between the edge-cuts of the graph and the communications volume present in the simulation execution. The normalized values are quite close for all 4 partitionings of all 3 circuits, with a maximum deviation of 5% (the 16-way partitioning of circuit s15850). Second, although there is a close correlation between the two values, it would be difficult to accurately predict one value given the other. For circuits s5378 and s15850, the communications volume is consistently higher than the edge-cuts. For circuit s9234, the edge-cuts are primarily higher than the communications volume; however, the values cross between 8 and 16 partitions.

The above results are only indicative of overall edge-cut and communications volume, they do not address the interaction between individual partitions. In the next set of results, we compare the edge-cut between each pair of partitions to the communications volume between the pair. In Figures 4 through 6, the pairwise message volume is plotted against the pairwise edge-cut for 2-way, 4-way, and 8-way partitionings of the 3 benchmark circuits. The sample cross-correlation coefficients, ρ , are also indicated in the figures, calculated for 2-way through 16-way partitionings. The overall aggregate sample cross-correlation coefficient is 0.70.

This data largely corroborates the overall edge-cut and communications volume data presented before.

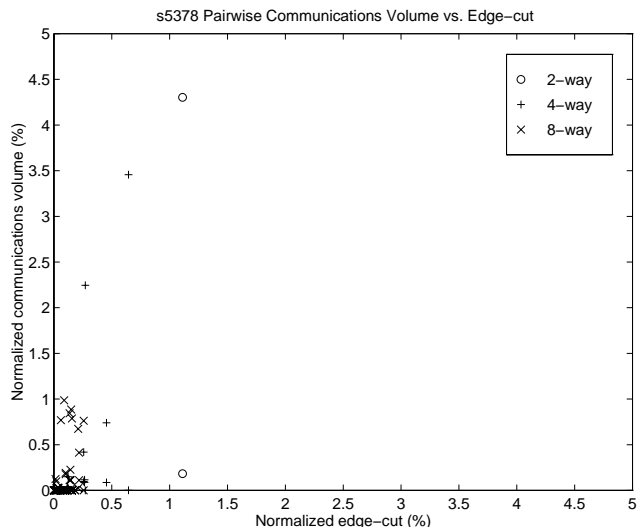


Figure 4: Circuit s5378, $\rho = 0.84$

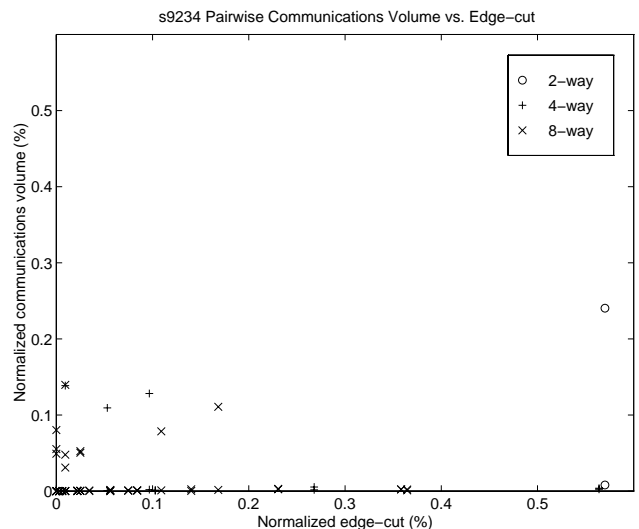


Figure 5: Circuit s9234, $\rho = 0.61$

There is a significant correlation between edge-cut data and communications volume; however, it would be erroneous to believe one could reliably predict one from the other. What we are looking for in these figures is for the collection of points to roughly form a straight line through the origin with a slope of one. Higher slopes indicate a higher communications volume (relative to edge-cut), and vice versa. Consistent with the earlier results, circuits s5378 and s15850 have a higher communication volume and an apparent slope greater than unity. Circuit s9234 has an apparent slope less than unity, corresponding to higher edge-cuts.

Although the communication volume data has the advantage that it is independent of the time synchronization algorithm used to coordinate simulated time across the processors, it does not directly correspond to the runtime needed to execute the parallel simulation. To

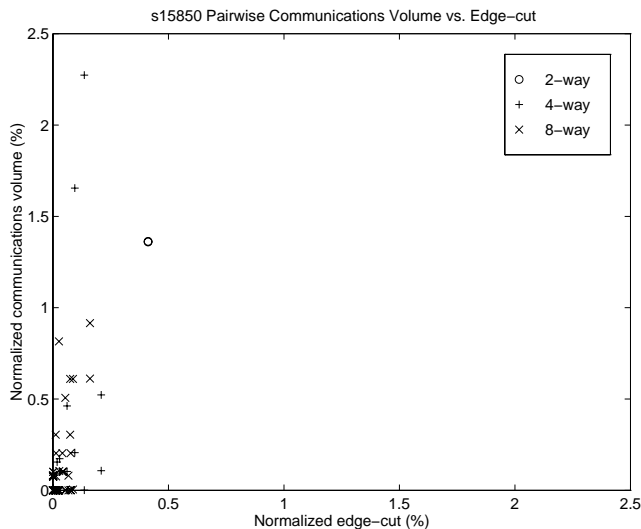


Figure 6: Circuit s15850, $\rho = 0.94$

help quantify this relationship, we execute several parallel simulation runs using a variety of partitioning algorithms. Circuit s5378 is run for 5,000 clock cycles both on one processor and on 4 processors of a Sun SPARC-center 2000 symmetric multiprocessor. The parallel executions use a globally-clocked time synchronization algorithm, and the partitioning of components to processors is accomplished using the following algorithms: (a) random assignment, (b) random assignment with replicated clock sources in each partition, and (c) METIS partitioning with replicated clock sources. The execution runtimes and speedups (relative to the uniprocessor case) are presented in Table 2. All of these timing results are the mean of 10 independent runs, to diminish the impact of such factors as variable operating system overhead, etc. Although the system is multiuser, other users were not allowed on the machine during the execution of the timing runs.

The results indicate a clear connection between parallel simulation performance and the communications volume present in the simulation. With the original random partitioning, the simulation exhibits a slowdown, executing slower on four processors than on one processor. Once the large fanout input (the clock signal) is distributed, the simulation at least shows a speedup, albeit a small one. Significant performance improvement is seen with the METIS partitioning, resulting in a parallel efficiency of over 50% (a speedup of greater than 2 on 4 processors).

4 Conclusions

To summarize, classic graph partitioning algorithms typically optimize edge-cuts between partitions. This figure of merit is not exactly what is desired in circuit partitioning for parallel logic simulation, and this

Proc.	Partitioning Algorithm	Runtime (ms)	Speedup
1	—	656,830	1.00
4	random	1,312,612	0.50
4	random w/ dist. clk.	517,590	1.27
4	METIS w/ dist. clk.	289,910	2.27

Table 2: Parallel execution of circuit s5378

paper investigates the connection between edge-cuts and communications volume (a more appropriate figure of merit for the parallel simulation problem).

What is observed is that there is a fairly good correlation between edge-cuts and communication volume, provided that high fanout signals are treated separately, as a special case. The two are not identical, however, and it is impractical to assume the possibility of reliably predicting one from the other.

The connection between communication volume and simulation runtime is shown fairly convincingly with a greater than four to one performance improvement of a 4 processor parallel simulation when the partitioning is altered from a high communications volume to a low volume. The strong impact of communications volume on parallel runtime indicates the importance of good partitioning algorithms. We are currently investigating the feasibility of developing edge weights for the graph, corresponding to signal transitions within the VLSI circuit, to further guide the graph partitioning algorithms.

References

- [1] M.L. Bailey, J.V. Briner, Jr., and R.D. Chamberlain, Parallel Logic Simulation of VLSI Systems. *ACM Computing Surveys*, Vol. 26, No 3, Sept. 1994.
- [2] F. Brglez, D. Bryan, and K. Kozminski, Combinational Profiles of Sequential Benchmark Circuits. In *Proc of the Int'l Symp. on Circuits and Systems*, pages 1929–1934, May 1989.
- [3] J.V. Briner, Jr., J.L. Ellis, and G. Kedem, Breaking the Barrier of Parallel Logic Simulation of Digital Systems. In *Proc. of the 28th ACM/IEEE Design Automation Conf.*, pages 223–226, June 1991.
- [4] R.D. Chamberlain and M.A. Franklin, Collecting Data About Logic Simulation. *IEEE Trans. on Computer-Aided Design*, Vol. CAD-5, No. 3, July 1986.
- [5] G. Karypis and V. Kumar, Multilevel Graph Partitioning and Sparse Matrix Ordering. In *Proc. of the Int'l Conf. on Parallel Processing*, 1995.
- [6] G. Karypis and V. Kumar, METIS Unstructured Graph Partitioning and Sparse Matrix Ordering System Version 2.0. <http://www.cs.umn.edu/~karypis>.